

BAB I

PENDAHULUAN

A. Latar Belakang

Pada era globalisasi ini kemajuan di bidang teknologi informasi semakin pesat memberikan pengaruh yang sangat besar terhadap berbagai aspek kehidupan manusia. Salah satu yang mendukung kemajuan teknologi informasi pada saat ini yaitu peningkatan kualitas pelayanan informasi dalam menyajikan informasi dan komunikasi secara luas. Banyak investasi waktu dan ruang dalam pengembangan aplikasi mobile sebagai salah satu penyajian informasi yang paling berkembang. Hal ini memberikan peluang bagi perusahaan untuk berkomunikasi secara efektif dengan pengguna dan menyampaikan informasi dengan cara yang lebih mudah diakses, interaktif, dan personal.

Aplikasi Mobile adalah sebuah aplikasi yang dirancang dan dikembangkan untuk proses kegiatan bisnis yang terjadi melalui perangkat mobile. Hal tersebut dapat memudahkan pengguna dalam mendapatkan berita atau hal lain yang mereka inginkan dengan cara yang lebih mudah dan cepat, yaitu dengan menginstal aplikasi komersial tersebut. Aplikasi ini bisa diunduh dan diinstal dari toko aplikasi resmi, seperti Google Play Store untuk Android.

Aplikasi mobile bagi suatu perusahaan dapat membantu perusahaan dalam meningkatkan produktivitas, kolaborasi tim, manajemen waktu, mobilitas kerja, dan pengumpulan data lapangan. PT. Cakra Agro Indonesia merupakan sebuah perusahaan swasta yang bergerak di bidang penjualan produk pertanian yang berlokasi di Indramayu, Jawa Barat. Proses pengumpulan data pada perusahaan tersebut masih dilakukan secara manual dan untuk produk yang ditawarkan oleh perusahaan pada website yang telah ada masih kurang menarik perhatian konsumen, sehingga diperlukan strategi penyelesaian yang efektif.

Sebagai penawaran solusi dalam permasalahan tersebut maka diibuatlah sebuah aplikasi mobile untuk perusahaan PT. Cakra Agro Indonesia. Pada aplikasi mobile ini terdapat informasi tentang perusahaan, deskripsi produk yang dibuat, laporan pengumpulan data di lapangan oleh beberapa staff ke staff yang dituju. Dari aplikasi mobile ini diharapkan dapat mempermudah staff perusahaan dalam

melaporkan data di lapangan dan menarik perhatian konsumen dalam melihat informasi mengenai produk.

B. Personel yang terlibat

Tim Manajemen Tugas Proyek “Pembuatan Aplikasi Komersial berbasis mobile” beranggotakan 7 orang dengan tugasnya masing-masing sesuai dengan tabel 1.1

Tabel 1. 1 Anggota Tim Brocode

Nama	NIM	Jabatan
Vintya Dewi Anjani	2000018209	Project Manager
Tatik Nafiaturhamah	2000018344	Sekretaris & Tester
Duwi Oktoviyanti	2000018206	Bendahara & Digital Marketing
Miftaqul Fiqi Firmansyah	2000018232	Programmer
Hikmatuz Zahra	2000018180	Business Analyst
Nizhaambiq Elbanny	2000018238	Programmer & Tester
Taufiq Aditya Putra	2000018201	UI/UX

C. Proses mendapatkan proyek

Adapun proses mendapatkan proyek untuk Manajemen Proyek Teknologi Informasi (MPTI) ini adalah didapatkan dari Bapak Fajar Anugrah yang merupakan pemilik dari PT. Cakra Agro Indonesia yang berlokasi di Indramayu, Jawa Barat. Pada saat itu perusahaan tersebut ingin mengembangkan website perusahaan menjadi aplikasi yang berbasis mobile. Kami mendapatkan informasi mengenai perusahaan tersebut dari rekan tim kami yang bertempat tinggal di Indramayu. Selanjutnya marketing dari tim kami mencoba untuk menghubungi dan mengajukan proposal kepada PT. Cakra Agro Indonesia, lalu tim kami melakukan diskusi terkait proposal yang kami ajukan kepada perusahaan dan proyek tersebut di setujui oleh pihak klien. Kemudian tim kami menghubungi kembali pihak klien dan berdiskusi secara online terkait dengan sistem dan proses dari system yang akan dibuat dapat selesai sesuai dengan waktu yang telah ditentukan. Tim kami menyetujui adanya Kerjasama proyek ini, maka dibuatlah aplikasi komersial PT. Cakra Agro Indonesia berbasis mobile.

D. Deskripsi perencanaan proyek

Berdasarkan rapat koordinasi yang telah dilakukan bersama klien secara online, maka “Pembuatan Aplikasi Komersial PT. Cakra Agro Indonesia Berbasis Mobile” dirancang dan dikembangkan untuk proses kegiatan bisnis yang terjadi melalui perangkat mobile. Pada pembuatan aplikasi ini terdapat beberapa fitur yang akan dibuat.

- a. Login
Login adalah suatu proses identifikasi diri untuk masuk ke dalam suatu aplikasi.
- b. CRUD (Create, Read, Update, dan Delete)
CRUD digunakan untuk membuat, membaca, menyisipkan, memanipulasi, mengedit, dan menghapus data.
- b. Searching
Searching adalah sebuah metode pencarian guna menemukan data atau informasi yang sedang dicari di dalam sebuah aplikasi.
- c. Logout
Logout adalah proses keluar dari akun atau layanan aplikasi tersebut.

BAB II

GAMBARAN MITRA/KLIEN

A. Gambaran Umum

1. Sejarah

PT. Cakra Agro Indonesia merupakan perusahaan formulator pestisida yang berkualitas dan terjangkau untuk mendukung kesuksesan para petani Indonesia.

2. Visi dan Misi

a) Visi

“Terwujudnya perusahaan PT. Cakra Agro Indonesia yang menjalin kerjasama agribisnis yang hebat dan berkesinambungan.”

b) Misi

- 1) Mewujudkan kemitraan yang saling menguntungkan antara perusahaan dengan seluruh Stakeholder.
- 2) Menjual produk yang inovatif solutif dan efisien bagi petani.
- 3) Menciptakan etos kerja yang disiplin dan penuh tanggung jawab.
- 4) Menerapkan Cashflow yang disiplin dan singkat dalam *Collection*.
- 5) Menerapkan pola efisiensi dan efektifitas dalam setiap aktivitas sales dan marketing.
- 6) Manajemen Marketing Development yang baik dan hebat.
- 7) Mampu beradaptasi dan berinovasi dalam kondisi bisnis apapun.
- 8) Membangun perusahaan *holding* yang berkesinambungan.

3. Alamat dan Kontak

Alamat : Jl. Raya Sanca Blok Manangga RT 002/001, Sanca, Kec.
Gantar, Kabupaten Indramayu, Jawa Barat 45264

Kontak : 081286888202

Peta Lokasi : <https://goo.gl/maps/FGQ19LtVBU1QcvMX9>



Gambar 2. 1 Lokasi PT. Cakra Agro Indonesia

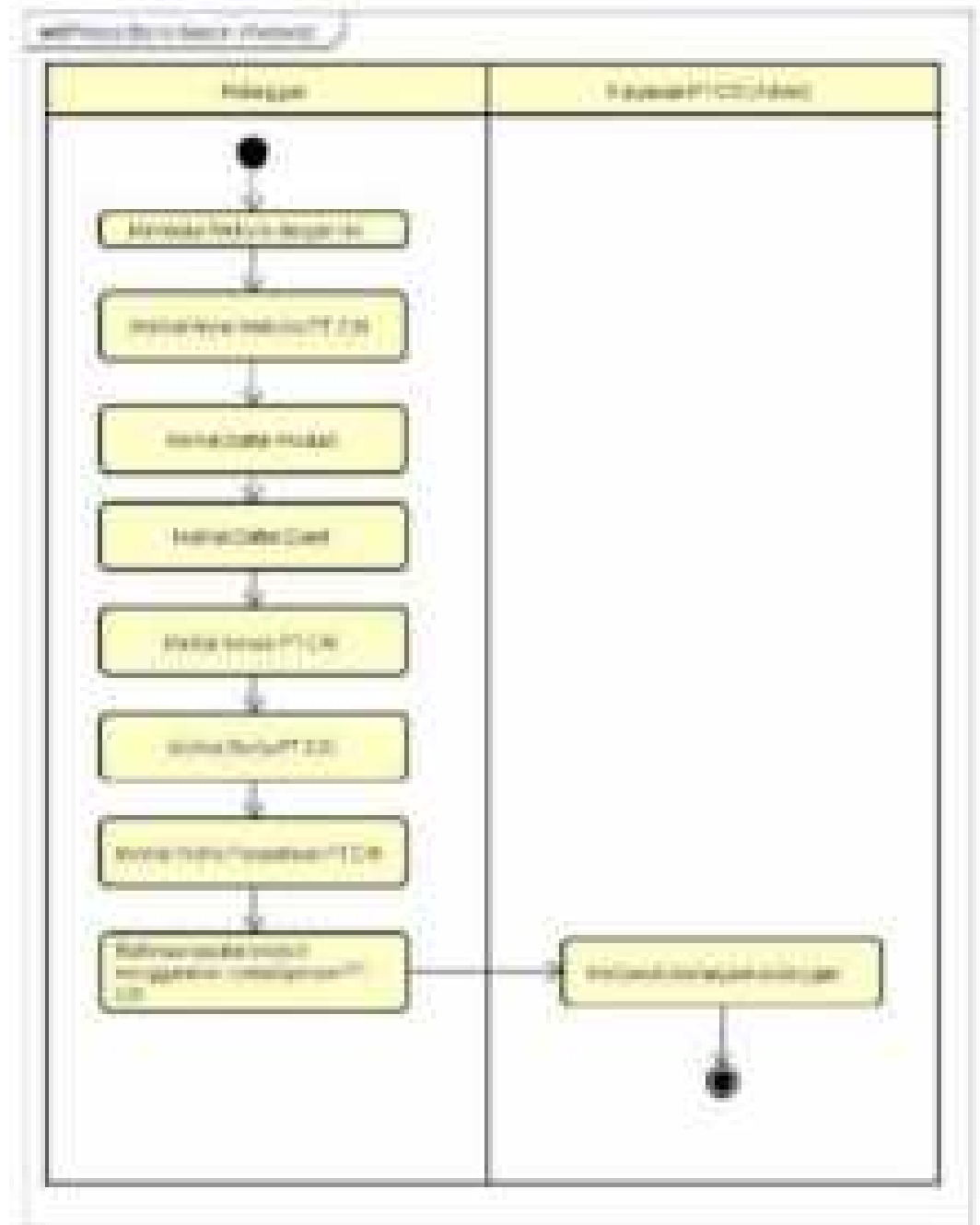
B. Struktur Organisasi Perusahaan



Gambar 2. 2 Struktur organisasi Perusahaan

C. Proses Bisnis pada Klien saat ini

Proses bisnis merupakan satu aktivitas yang terkait dengan menentukan bagaimana suatu pekerjaan tersebut dapat berjalan. Proses bisnis juga salah satu cara untuk mengkoordinasikan pekerjaan supaya berjalan dengan baik.



Gambar 2. 3 Proses Bisnis Saat Ini (Website)

BAB III

RENCANA MPTI

A. Rencana Pembagian Tugas

Berikut deskripsi rencana pembagian tugas pada anggota tim problem solver dalam proyek pembuatan sistem rekapitulasi data pasien adalah sebagai berikut :

1. *Project Manager*

Nama : Vintya Dewi Anjani

NIM : 2000018209

Tugas :

- a) Membentuk tim.
- b) Memimpin rapat.
- c) Membagi tugas setiap anggota.
- d) Memantau kerja anggota.
- e) Menyetujui dan menandatangani proposal dan surat perjanjian kerjasama.
- f) Bertanggung jawab terhadap tim.
- g) Mengawasi dan mengevaluasi pelaksanaan pengerjaan proyek.

2. Sekretaris dan *Tester*

Nama : Tatik Nafiaturohmah

NIM : 2000018344

Tugas :

- a) Menyusun proposal proyek.
- b) Menyusun surat perjanjian kerjasama antara tim dan klien.
- c) Menyiapkan logbook.
- d) Membuat notulensi rapat.
- e) Mengumpulkan data setiap anggota ke dalam *Drive* kelompok.
- f) Membuat laporan pengerjaan proyek.

3. Bendahara & *Digital Marketing*

Nama : Duwi Oktoviyanti

NIM : 2000018206

Tugas :

- a) Merancang anggaran dana proyek dalam proposal.
- b) Merancang gaji setiap anggota dan didiskusikan dengan *project manager*.
- c) Mengelola keuangan dalam pembuatan proyek.
- d) Membuat laporan pemasukan dan pengeluaran keuangan.
- e) Membuat strategi penawaran produk.
- f) Menawarkan proposal kepada klien.

4. *Programmer*

Nama : Miftaql Fiqi Firmansyah

NIM : 2000018232

Tugas :

- a) Membuat Mengimplementasi ui menjadi program.
- b) Membuat Membuat struktur program.
- c) Menghubungkan database ke program.
- d) Konsumsi API.
- e) Implementasi method GET.
- f) Bekerja sama dengan UI/UX designer.
- g) Bekerja sama dengan sesama programmer.

5. *Business Analyst*

Nama : Hikmatuz Zahra

NIM : 2000018180

Tugas :

- a) Melakukan wawancara dengan klien.
- b) Pembuatan proses bisnis.
- c) Menganalisis proses bisnis.
- d) Menganalisis kebutuhan fungsional dan non-fungsional.
- e) Membuat ERD, use case diagram, dan diagram activity.

6. *Programmer & Tester*

Nama : Nizhaambiq Elbanny

NIM : 2000018238

Tugas :

- a) Membuat database
- b) Bekerja sama dengan sesama *programmer*.
- c) Menyediakan API.
- d) Melakukan testing berkala.
- e) Menganalisa hasil testing.
- f) Membuat dokumentasi.

7. UI/UX

Nama : Taufiq Aditya Putra

NIM : 2000018201

Tugas :

- a) Pembuatan UI Desain
- b) Pembuatan *Prototype*
- c) Pembuatan Mockup
- d) Bekerja sama dengan programmer

B. Perancangan Sistem

Tujuan adanya perancangan sistem secara umum yaitu untuk memberikan gambaran umum kepada user tentang sistem yang akan di buat dan supaya meminimalisir kesalahpahaman terhadap pembuatan sistem. Berikut perancangan sistem pada aplikasi mobile yang akan dibuat :

1. Analisa

a. Kebutuhan Fungsional

Tabel 3. 1 kebutuhan Fungsional

No.	Kode	Deskripsi
1.	SKPL - F1	Aplikasi dapat menampilkan Foto-Foto Slide Produk di Homepage
2.	SKPL - F2	Aplikasi dapat menampilkan penjelasan singkat Produk di Homepage
3.	SKPL - F3	Aplikasi harus terdapat Menu Daftar Produk beserta deskripsi produk
4.	SKPL - F4	Aplikasi harus terdapat Menu Event yang menampilkan Submenu Aktivitas dan Demplot Produk
5.	SKPL - F5	Aplikasi harus terdapat Menu Inovasi yang menampilkan Submenu Produk baru dan Paket produk
6.	SKPL - F6	Aplikasi dapat menampilkan menu Event > Demplot atau video tutorial pemakaian produk yang dilakukan oleh PT. CAI
7.	SKPL - F7	Aplikasi dapat menampilkan menu Event > Aktivitas yang berisi daftar Kegiatan Aktivitas yang dilakukan PT. CAI
8.	SKPL - F8	Aplikasi dapat menampilkan menu Inovasi > Produk Baru dengan menampilkan produk terbaru dari PT.CAI
9.	SKPL - F9	Aplikasi dapat menampilkan menu Inovasi > Paket Produk berisi paket produk yang tersedia di PT. CAI
10.	SKPL - F10	Aplikasi dapat menampilkan menu FAQ dan form QnA berisi pertanyaan yang dapat diakses oleh Customer untuk bertanya dengan pihak PT. CAI
11.	SKPL - F11	Aplikasi dapat menampilkan menu Tentang Kami berisi Profil Perusahaan, Alamat, dan Nomor Kontak PT.CAI

b. Kebutuhan Non Fungsional

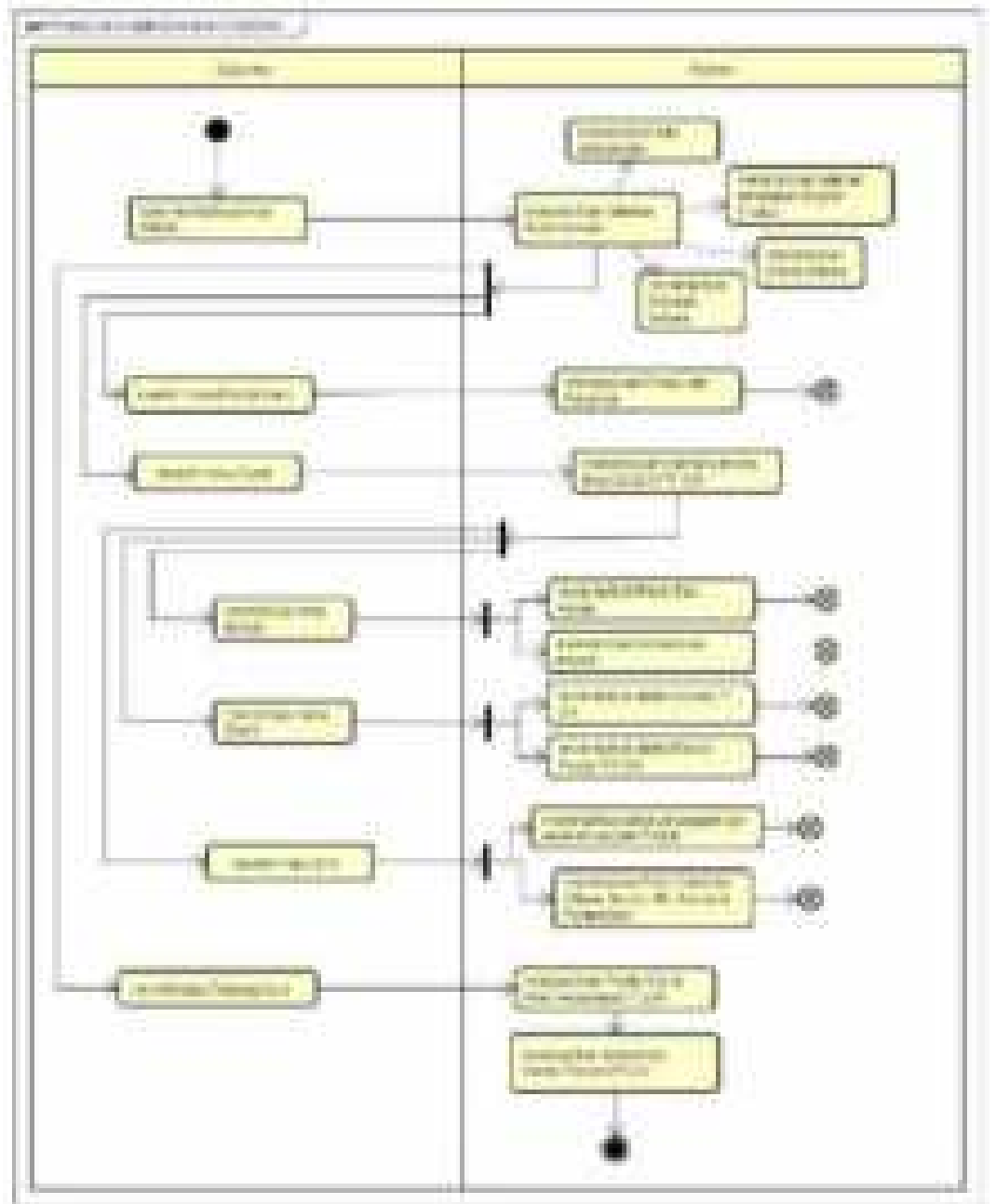
Tabel 3. 2 Kebutuhan Non Fungsional

No.	Kode	Parameter	Deskripsi
1.	SKPL - NF1	<i>Availability</i>	Aplikasi dapat diakses 24 jam
2.	SKPL - NF2	<i>Reliability</i>	<ul style="list-style-type: none"> Aplikasi dapat dioperasikan dan menampilkan data yang sesuai

			<p>dengan fungsinya termasuk fitur-fitur yang diinginkan</p> <ul style="list-style-type: none"> • Aplikasi dapat beroperasi dengan cepat
3.	SKPL - NF3	<i>Ergonomy</i>	Aplikasi memiliki interface yang memudahkan dan nyaman dimengerti pengguna (<i>User Friendly</i>)
4.	SKPL - NF4	<i>Portability</i>	Aplikasi hanya dapat diakses dengan perangkat Smartphone basis Android
5.	SKPL - NF5	<i>Memory</i>	Aplikasi memiliki penyimpanan atau database yang memadai
6.	SKPL - NF6	<i>Response Time</i>	Aplikasi dapat mengoperasikan dan menampilkan data dengan cepat sesuai fitur-fitur yang ada
7.	SKPL - NF7	<i>Safety</i>	<ul style="list-style-type: none"> • Aplikasi mempunyai keamanan yang dapat dipercaya • Data milik user hanya dapat diakses oleh Super Admin dan Admin
8.	SKPL - NF8	<i>Security</i>	Aplikasi memiliki keamanan tinggi agar data klien dan pengguna dapat terjamin dan terlindungi
9.	SKPL - NF9	<i>Communication</i>	Bahasa dalam aplikasi menggunakan Bahasa Indonesia

2. Perancangan

a. Proses Bisnis Aplikasi Mobile



Gambar 3. 1 Proses Bisnis Aplikasi Mobile (Customer)

b. Rancangan Entity Relationship Diagram (ERD)

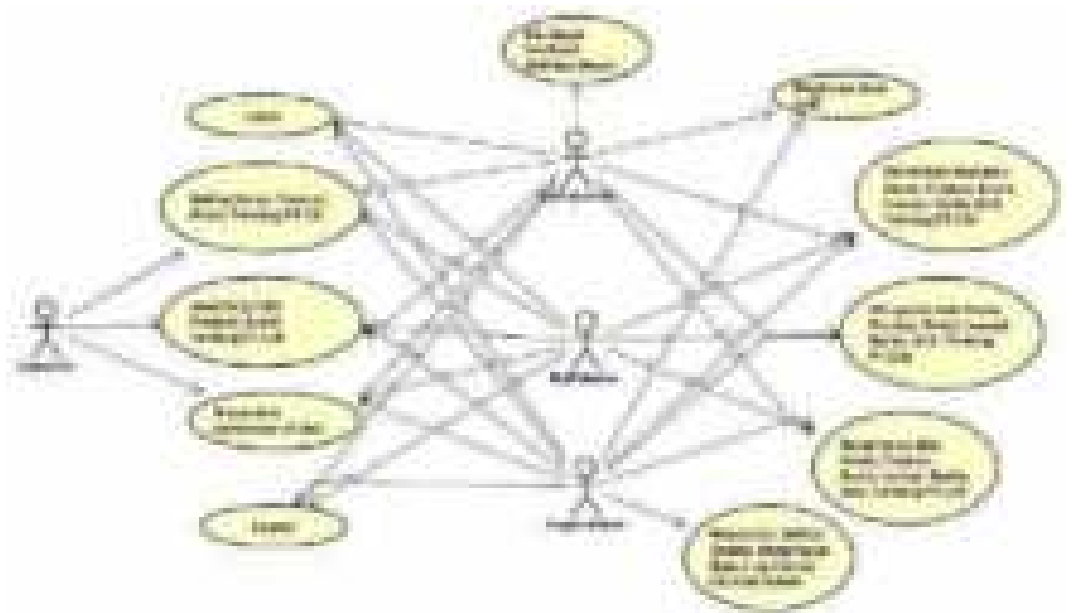
Entity Relationship Diagram (ERD) adalah suatu diagram yang digunakan untuk merancang suatu basis data, dipergunakan untuk memperlihatkan hubungan atau relasi antar entitas atau objek yang terlihat beserta atributnya.



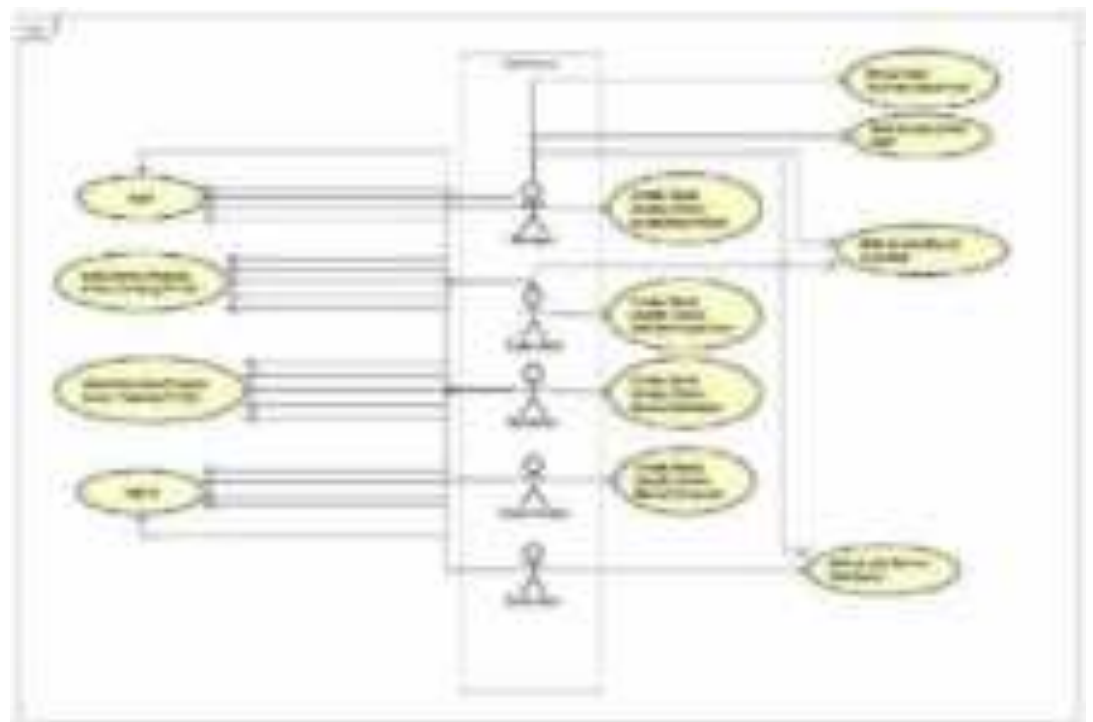
Gambar 3. 2 Entity Relationship Design (ERD)

c. Use Case Diagram

Use case digunakan untuk menggambarkan fungsi dasar dari sebuah sistem informasi. Use case mendeskripsikan cara sistem bisnis berinteraksi dengan lingkungannya.



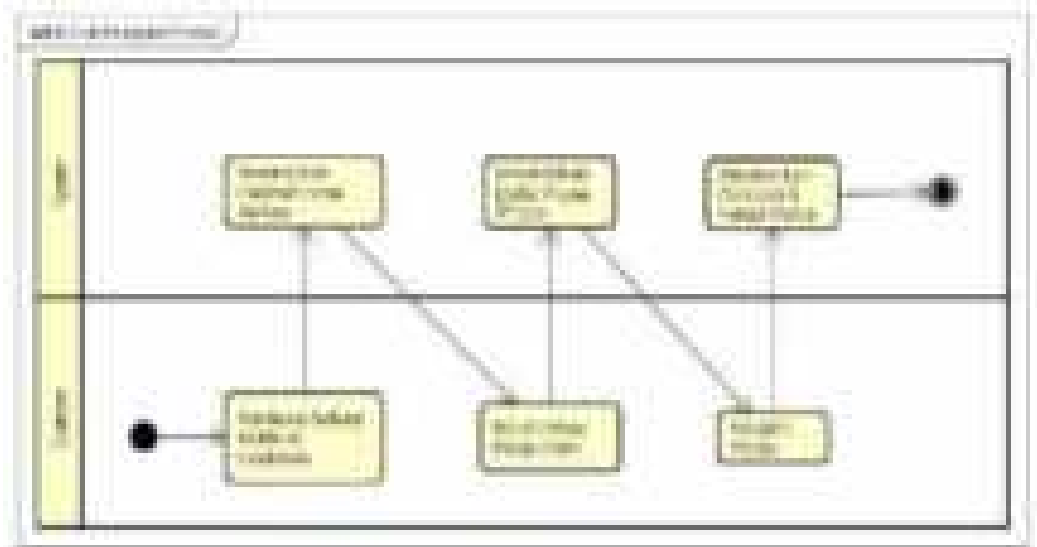
Gambar 3. 3 Use Case Diagram



Gambar 3. 4 Use Case Diagram

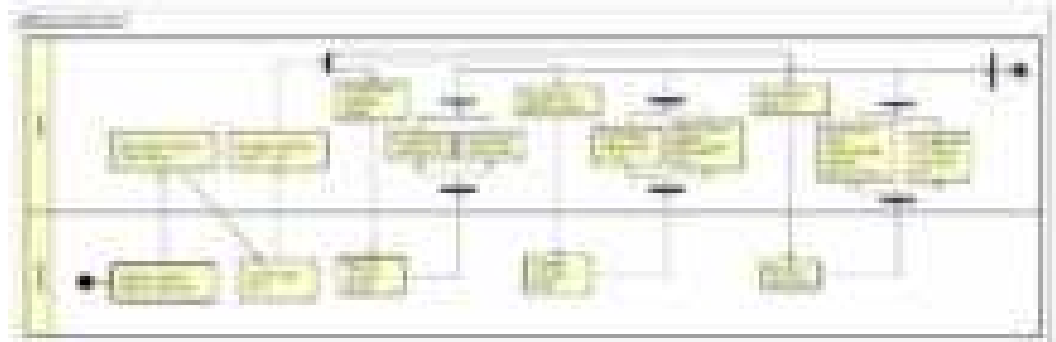
d. Activity Diagram

1) Activity Diagram Product



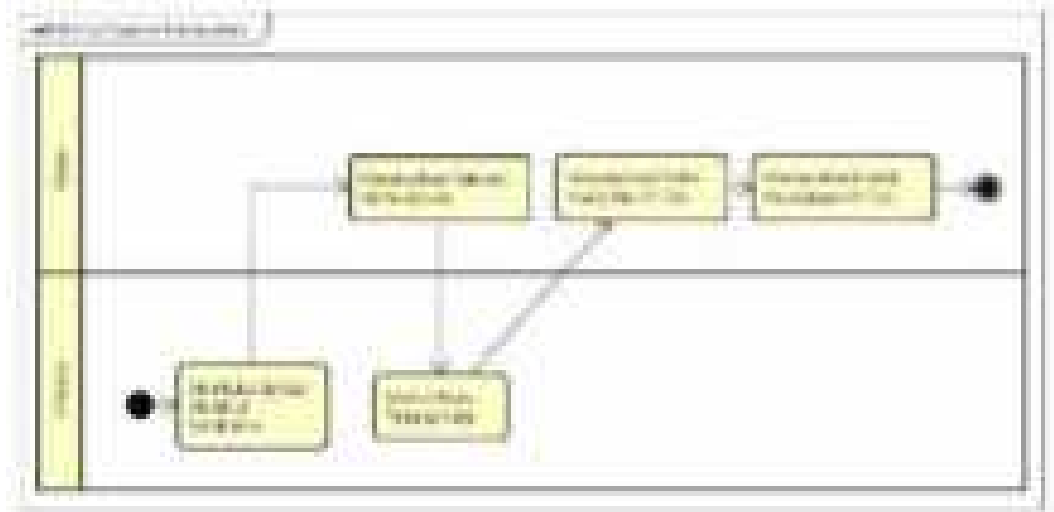
Gambar 3. 5 Activity Diagram Product

2) Activity Diagram Event



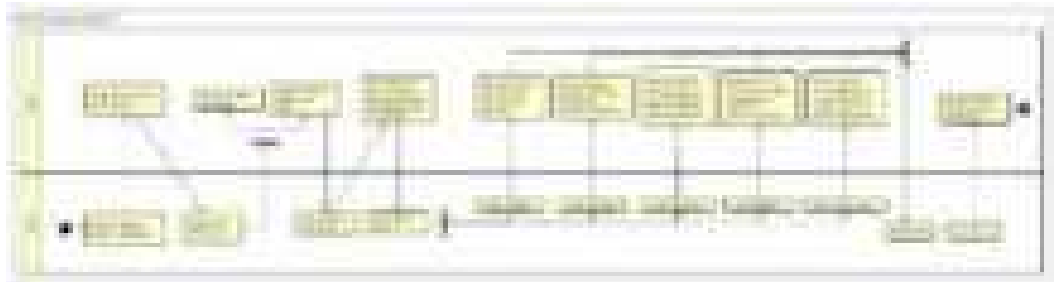
Gambar 3. 6 Activity Diagram Event

3) Activity Diagram Tentang Kami



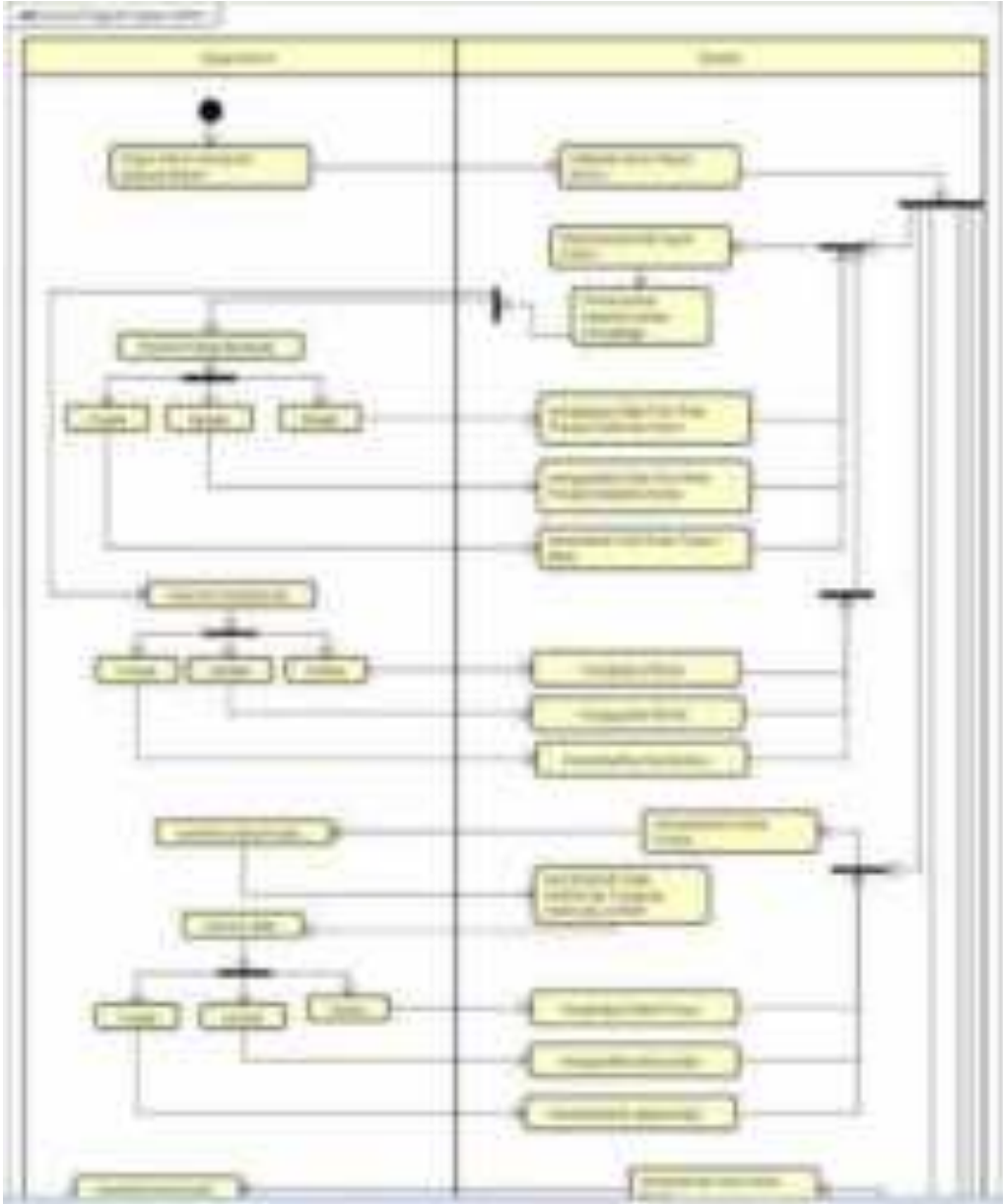
Gambar 3. 7 Activity Diagram Tentang Kami

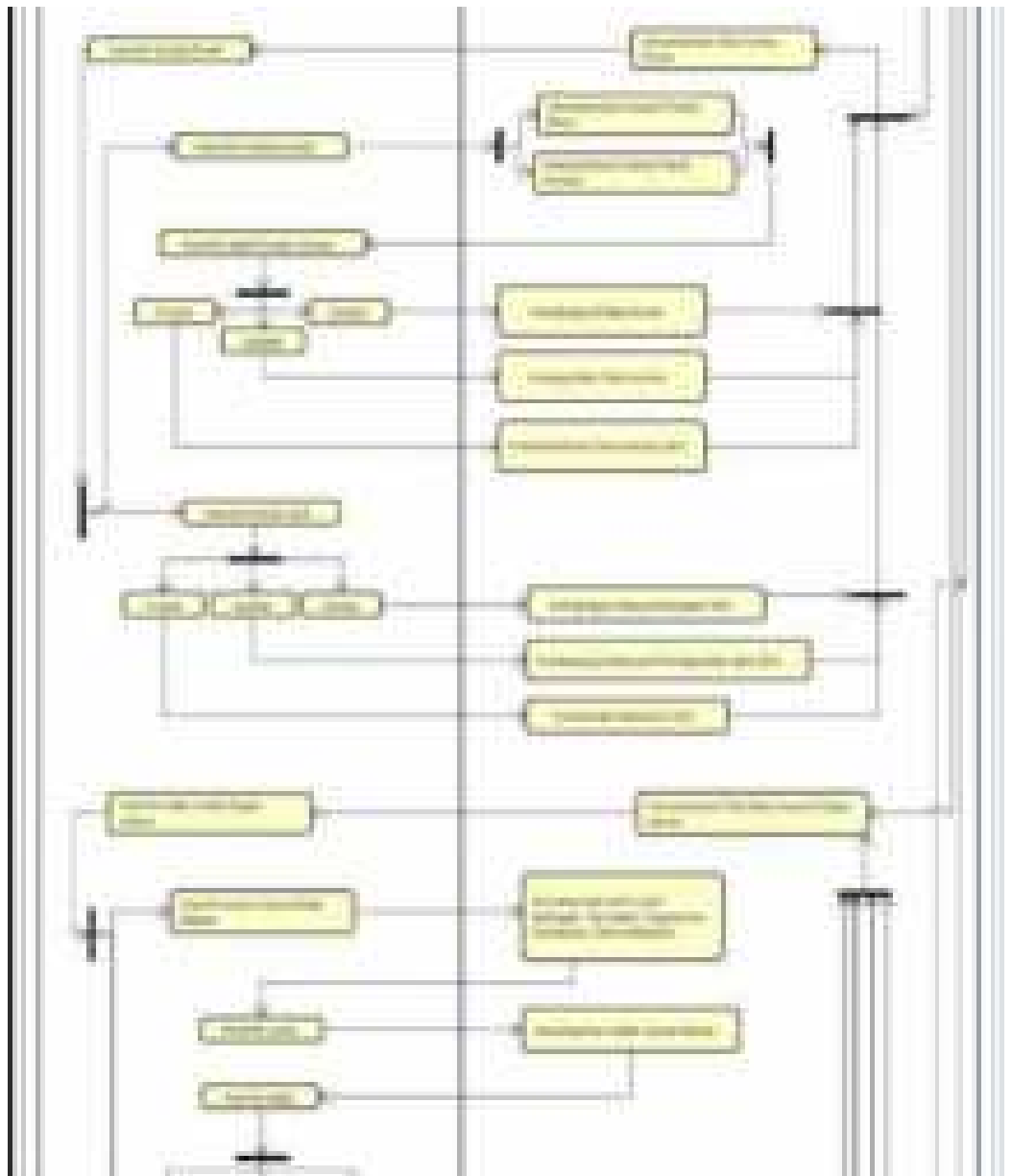
4) Activity Diagram Tambah Staff

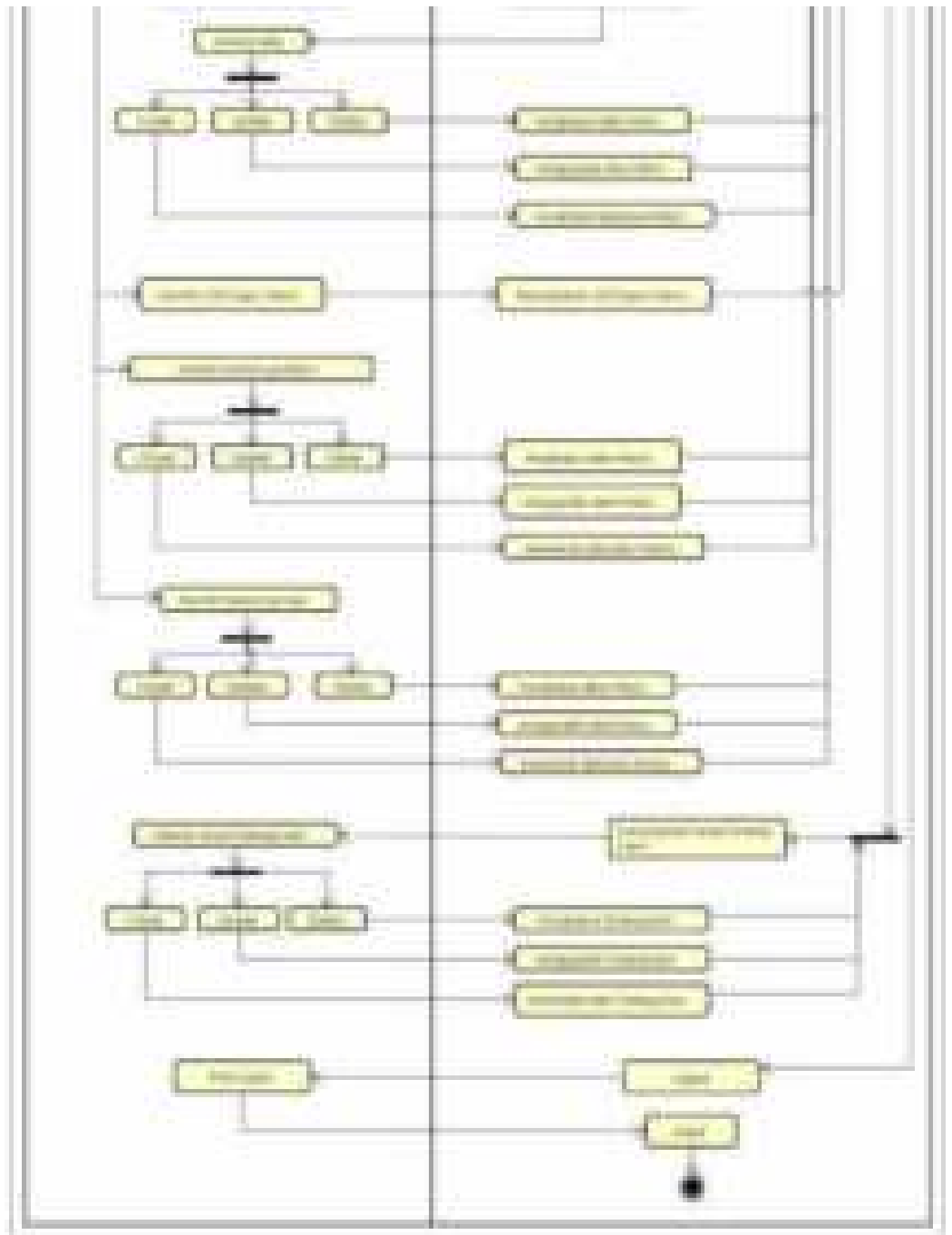


Gambar 3. 8 Activity Diagram Tambah Staff

5) Activity Diagram Super Admin

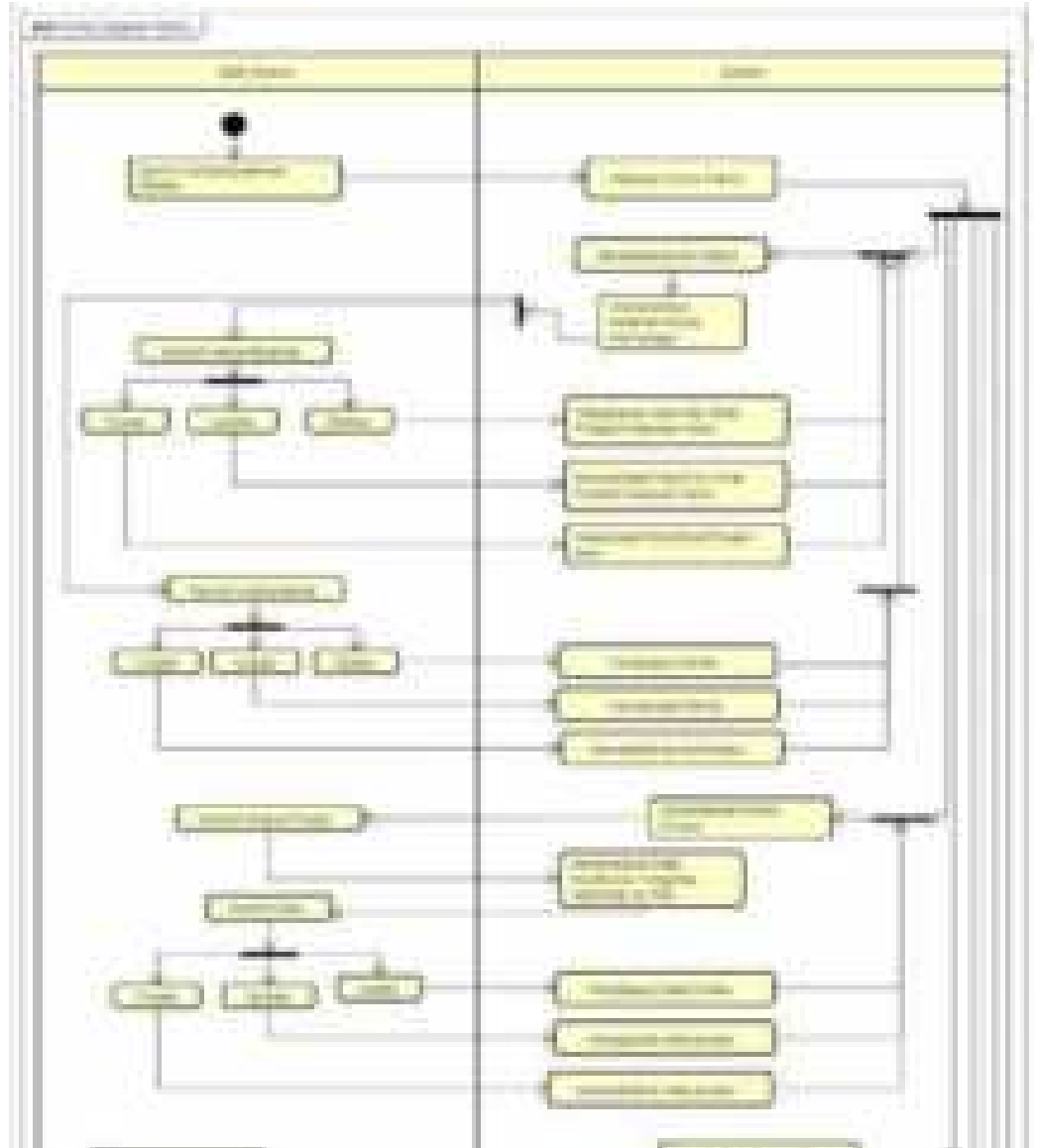


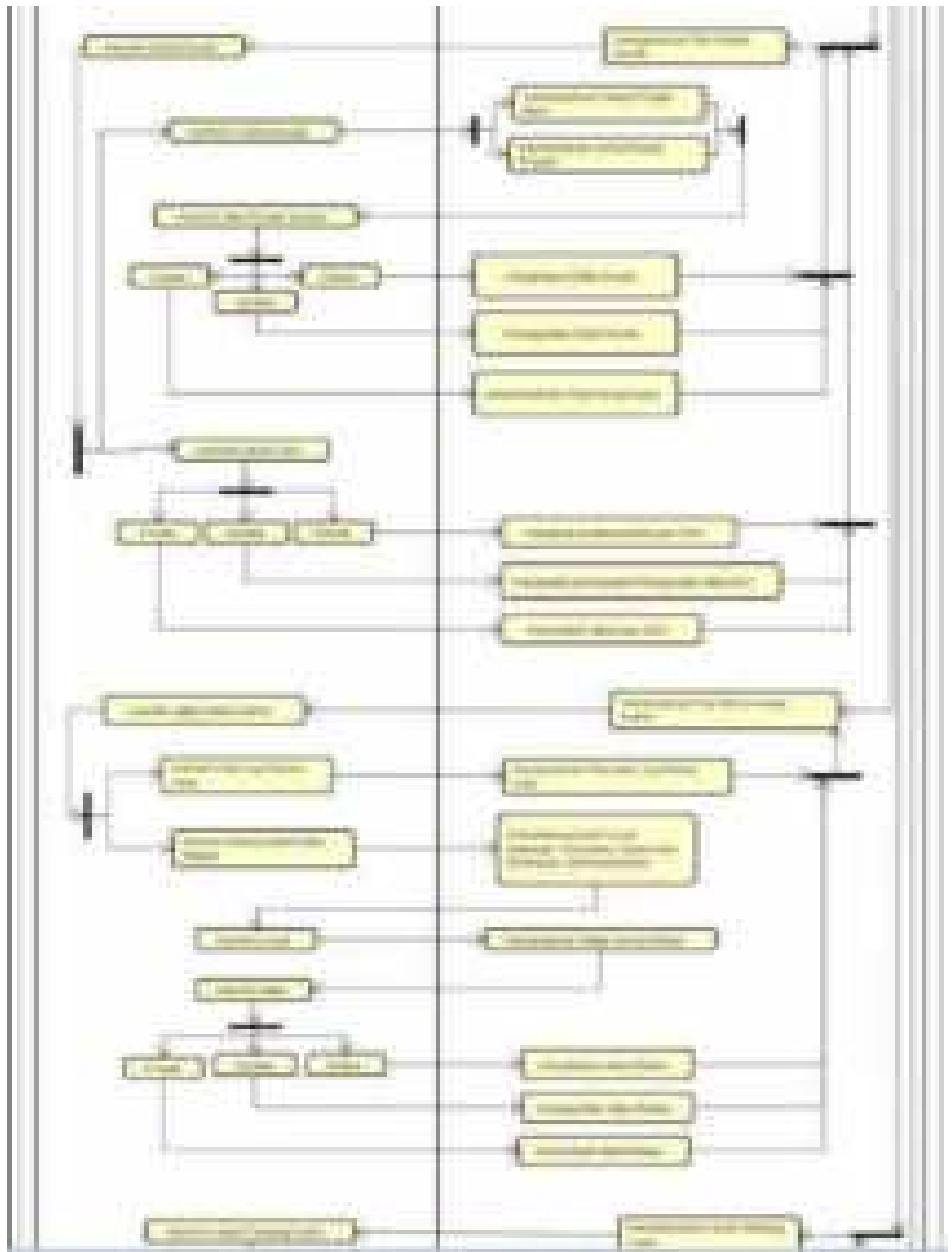


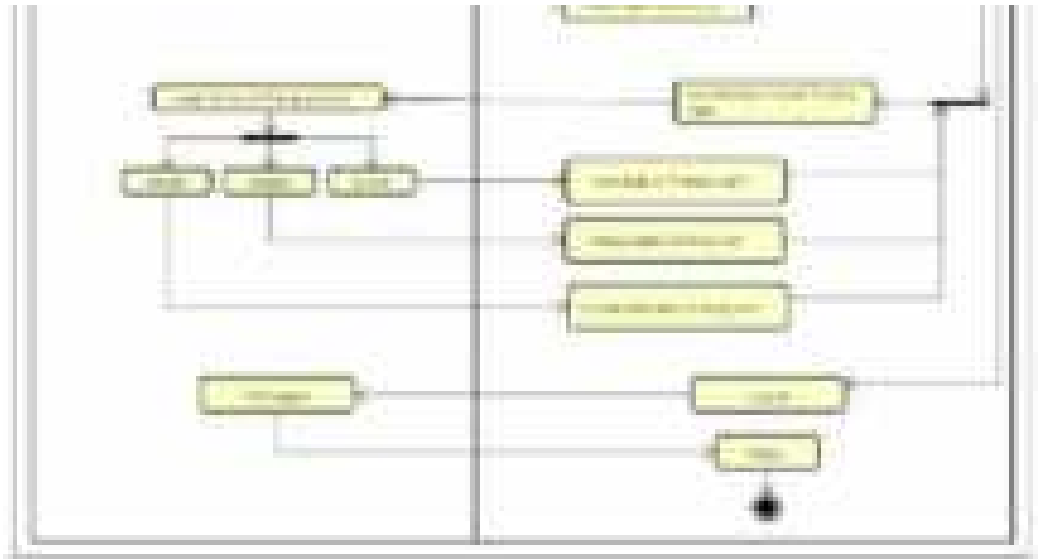


Gambar 3. 9 Activity Diagram Super Admin

6) Activity Diagram Admin

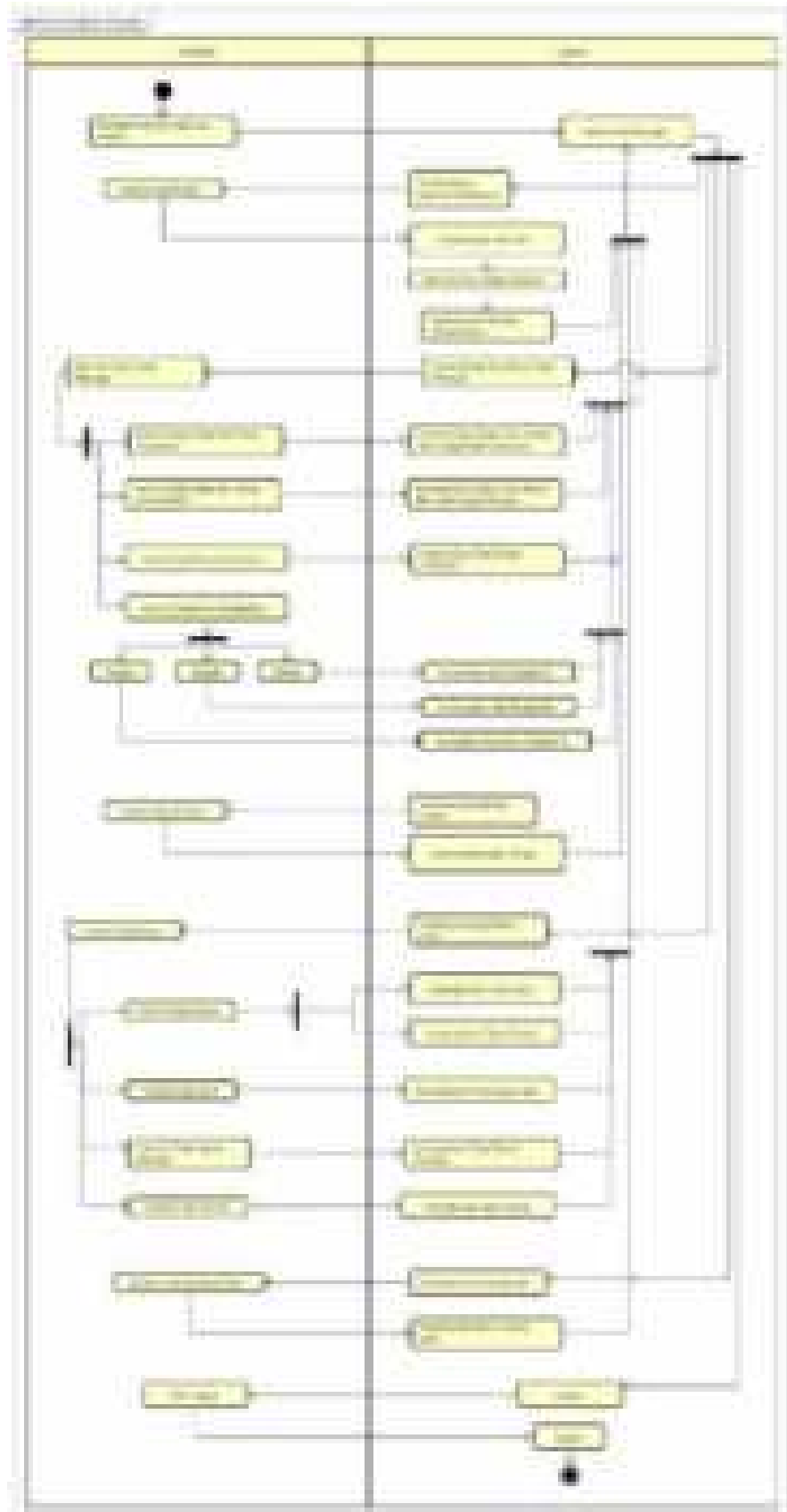






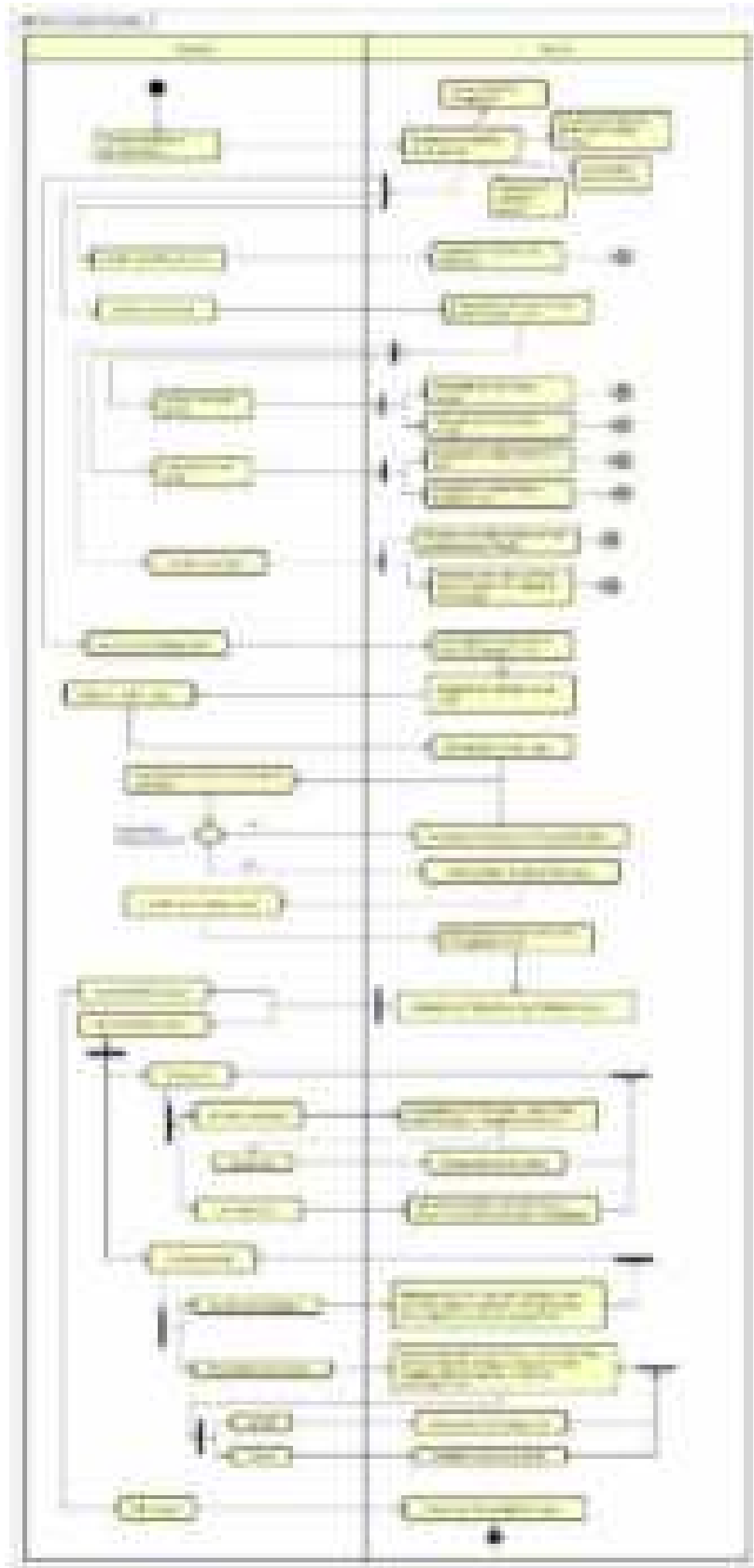
Gambar 3. 10 Activity Diagram Admin

7) Activity Diagram Manager



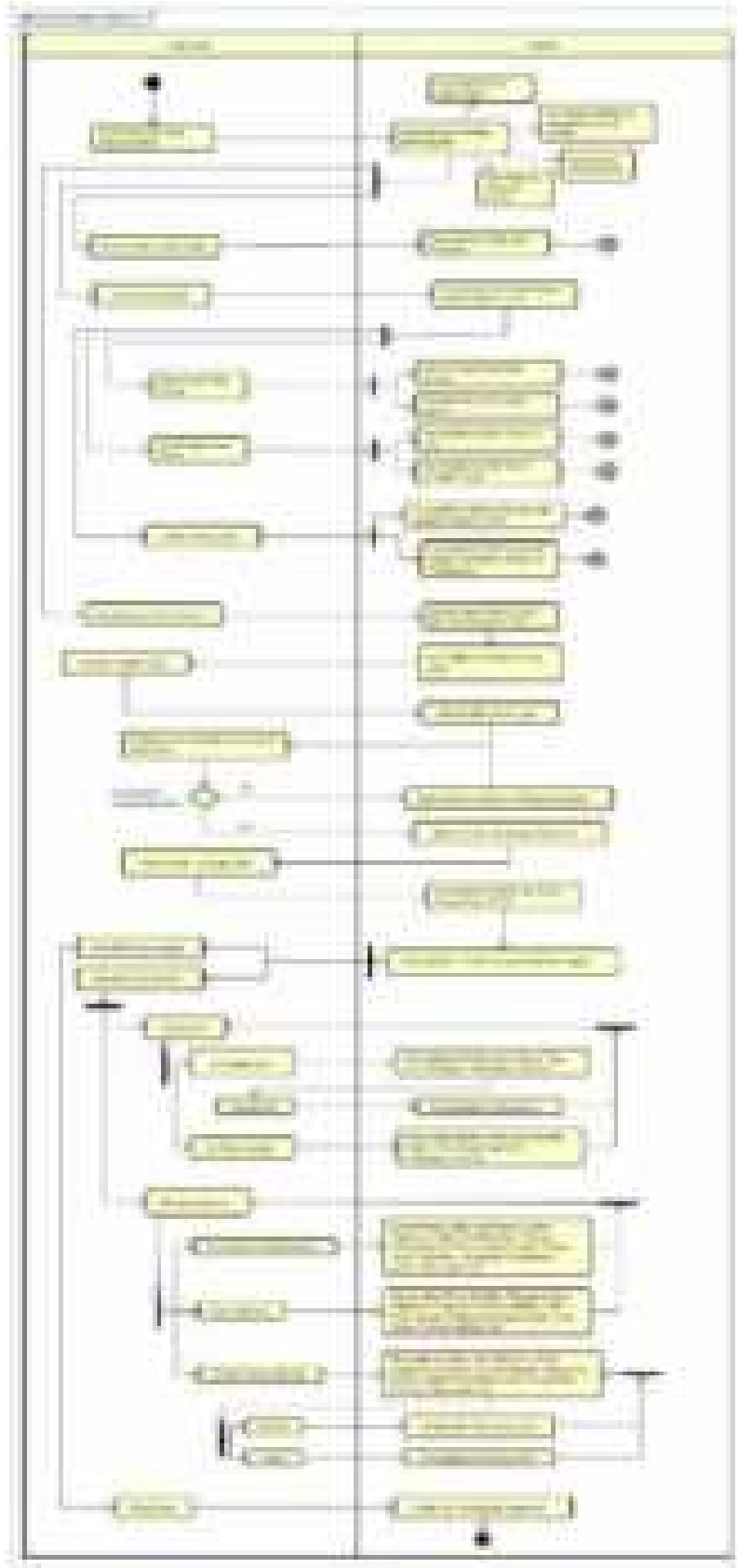
Gambar 3. 11 Activity Diagram Manager

8) Activity Diagram Secretary



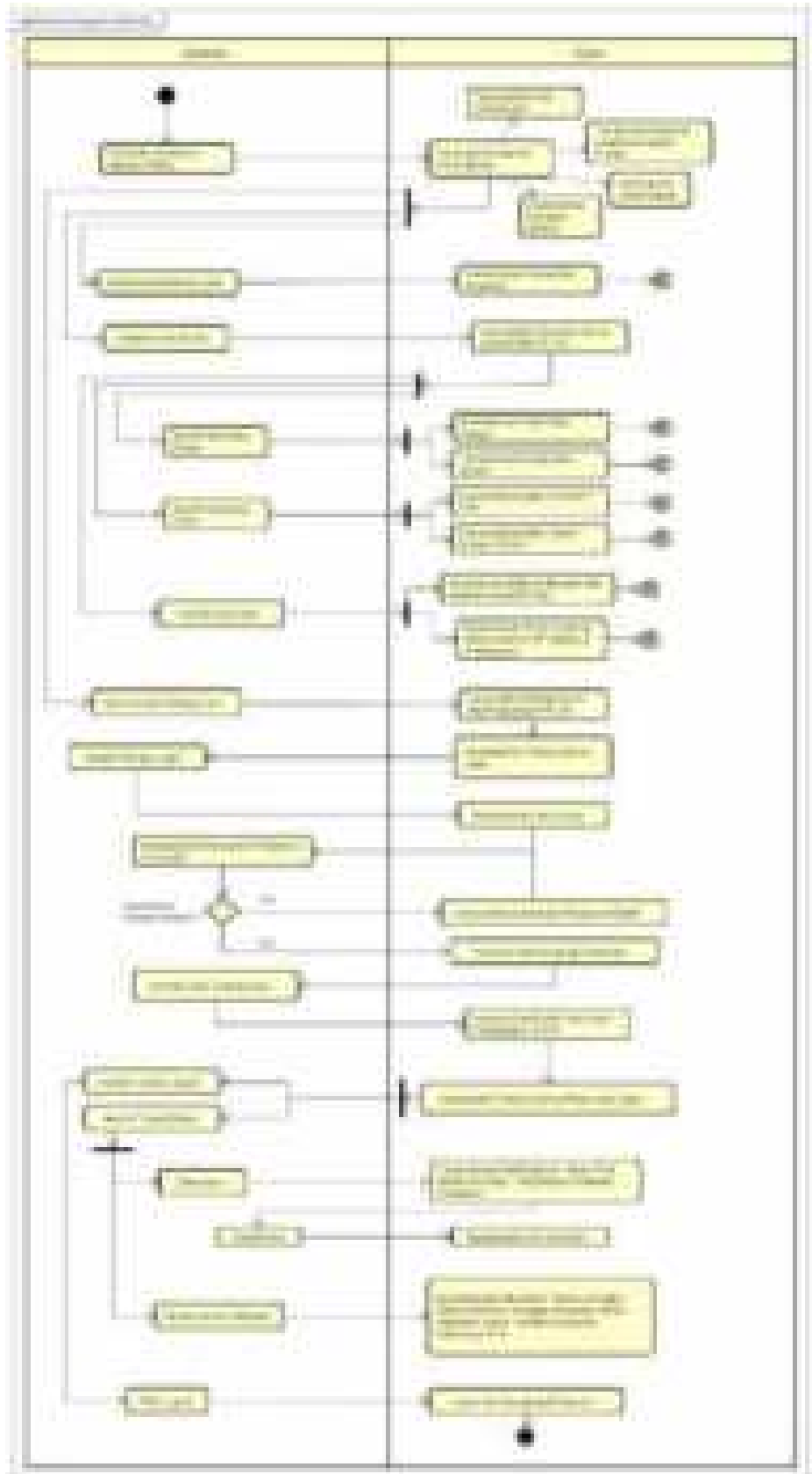
Gambar 3. 12 Activity Diagram Secretary

9) Activity Diagram Supervisor



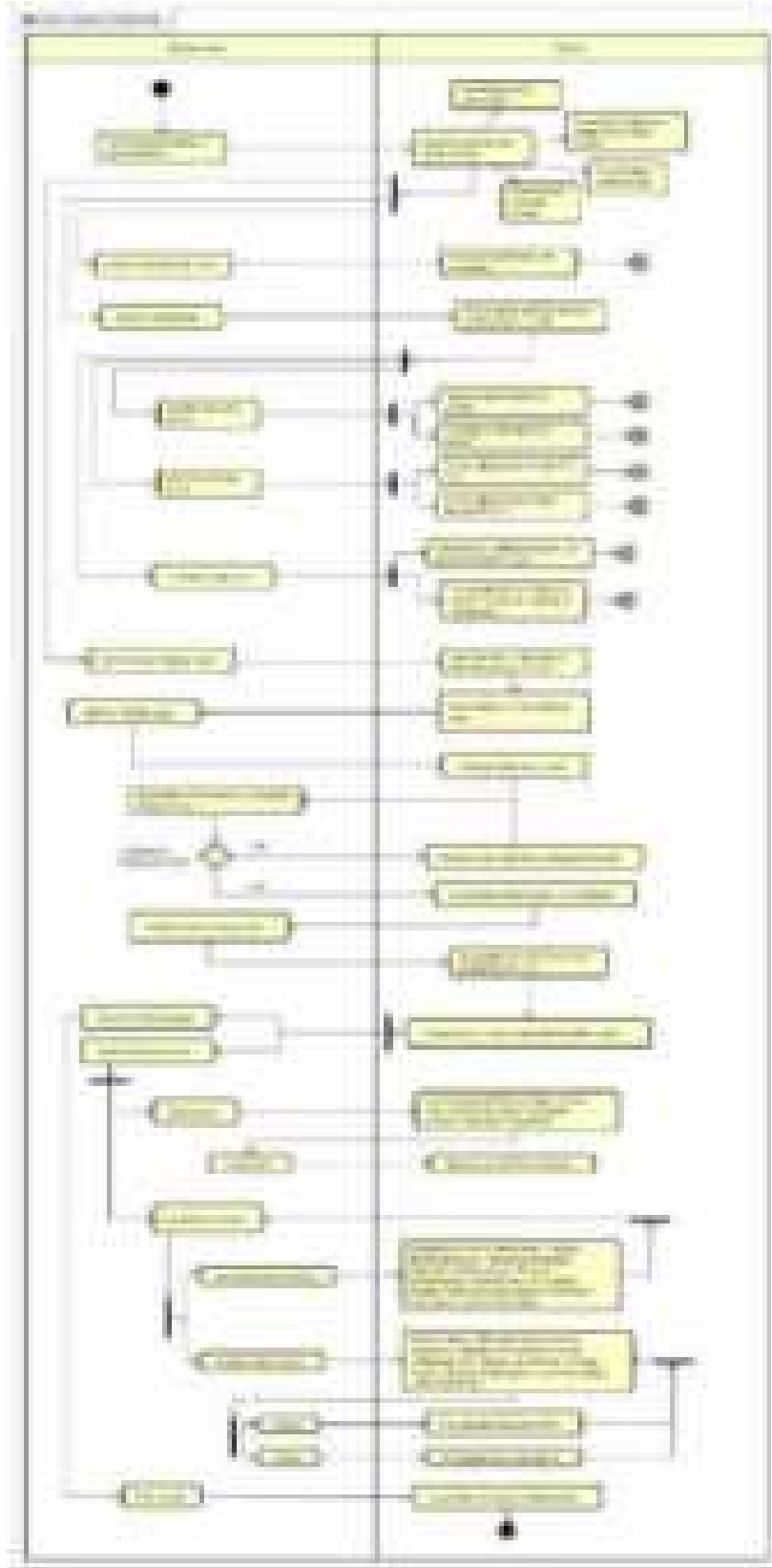
Gambar 3. 13 Activity Diagram Supervisor

10) Activity Diagram Distributor



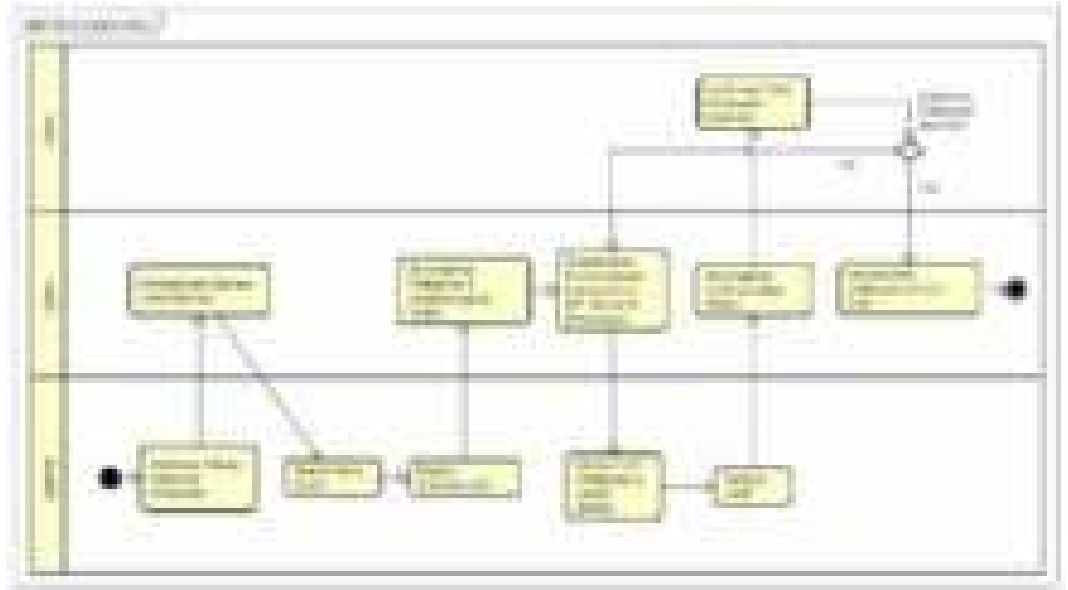
Gambar 3. 14 Activity Diagram Distributor

11) Activity Diagram Demonstrator



Gambar 3. 15 Activity Diagram Demonstrator

12) Activity Diagram QnA



Gambar 3. 16 Activity Diagram QnA

e. Rancangan Aplikasi Mobile

1) Tampilan Awal Aplikasi

Pada tampilan awal aplikasi bisa masuk menggunakan google atau email.
Bisa juga bisa daftar terlebih dahulu.



Gambar 3. 17 Tampilan Awal Aplikasi

2) Tampilan Daftar

Pada tampilan daftar, terdapat form untuk mengisi data.



Gambar 3. 18 Tampilan Daftar Akun

3) Tampilan Login

Pada halaman login terdapat masukkan email dan password untuk masuk ke dalam aplikasi.



Gambar 3. 19 Tampilan Login

4) Tampilan Lupa password

Jika sudah mempunyai akun tetapi lupa password maka bisa menggunakan email untuk mengatur ulang kata sandi.



Gambar 3. 20 Tampilan Lupa Password

5) Tampilan Home Page

Fitur yang terdapat di halaman ini adalah halaman depan dan deskripsi, beberapa tampilan slide produk dan deskripsi, beberapa contoh demplot dan sample berita.



Gambar 3. 21 Tampilan Home Page

6) Tampilan Jenis Produk

Fitur yang terdapat di halaman ini adalah beberapa tombol jenis produk.



Gambar 3. 22 Tampilan Jenis Produk

- 7) Macam Produk (halaman ini muncul saat salah satu jenis produk di klik)
Fitur yang terdapat di halaman ini adalah menampilkan macam produk dari jenisnya.



Gambar 3. 23 Tampilan Macam Produk

8) Detail Produk

Fitur yang terdapat di halaman ini adalah detail produk seperti gambar, nama produk, kandungan, deskripsi.



Gambar 3. 24 Tampilan Detail Produk

9) Tampilan Menu Aktivitas

Fitur yang terdapat di halaman ini adalah kolom Event yang berisikan tombol Aktivitas dan Demplot, Kolom Inovasi yang berisikan tombol produk baru dan paket produk, Tombol menuju FAQ.



Gambar 3. 25 Tampilan Menu Aktivitas

10) Demplot

Fitur yang terdapat di halaman ini adalah beberapa laporan demplot meliputi nama, kota, dan tanggal.



Gambar 3. 26 Tampilan Demplot

11) Halaman Kolom Pertanyaan

Fitur yang terdapat di halaman ini yaitu pertanyaan dari customer yang dapat di jawab oleh admin dan tombol ketik pertanyaan untuk menambahkan pertanyaan ke admin.



Gambar 3. 27 Tampilan Kolom pertanyaan

12) Tentang Perusahaan

Fitur yang terdapat di halaman ini yaitu Logo dan nama perusahaan, deskripsi, visi dan misi, foto dan alamat, tombol untuk menuju sosial media, dan tombol login untuk admin.



Gambar 3. 28 Tampilan Tentang Perusahaan

13) Halaman Logout

Pada halaman logout berisikan tampilan logout untuk keluar aplikasi.



Gambar 3. 29 Tampilan Logout

f. Pengujian

Tahap terakhir dalam proses pembuatan aplikasi komersial PT. Cakra Agro Indonesia adalah pengujian, dimana pada tahap ini pengguna atau user akan melakukan pengujian terhadap rancangan yang telah dibuat. Tujuan dari pengujian ini adalah untuk mengetahui apakah aplikasi yang telah dirancang sesuai dengan kebutuhan atau tidak.

C. Kesepakatan Proyek

1. Waktu Pelaksanaan Proyek

Waktu Mulai: 14 Mei 2023

Waktu Selesai: 7 Agustus 2023

2. Fitur Proyek

a. Login

Login adalah suatu proses identifikasi diri untuk masuk ke dalam suatu aplikasi.

b. CRUD (Create, Read, Update, dan Delete)

CRUD digunakan untuk membuat, membaca, menyisipkan, memanipulasi, mengedit, dan menghapus data.

CRUD pada aplikasi ini dibagi menjadi beberapa fitur yaitu:

a) Pendaftaran akun oleh admin

Admin melakukan pendaftaran akun berupa username dan password. Pendaftaran akun adalah tahap yang harus dilakukan user untuk dapat melakukan login, dengan cara mengisi data diri.

b) Upload

Upload adalah proses mengunggah file atau data (bisa berupa dokumen, foto, audio, video) ke server.

c) QnA (Questions and Answers)

QnA digunakan untuk menampilkan fitur tanya jawab dan menambahkan pertanyaan.

CRUD dalam aplikasi ini adalah sebagai berikut :

1) CRUD Super Admin

CRUD Super Admin dapat mengelola CRUD Table Admin dan CRUD Table LOG.

2) CRUD Admin

CRUD Admin dapat mengelola CRUD Table Staff, CRUD Table Produk, CRUD Table Event (Aktivitas dan Demplot), CRUD Table Inovasi (Produk Baru dan Paket Baru), CRUD Table Berita, CRUD Table Tentang Kami, CRUD Table QNA, dan CRUD Table Beranda.

3) CRUD Staff

CRUD Staff dibagi menjadi 5 bagian:

a. Manager

Manager dapat Read Report Secretary dan Supervisor, Read Bonus Distributor, Read Aktivitas Supervisor, dan CRUD Menu Budgeting.

b. Secretary

Secretary dapat Read Report dari Distributor dan CRUD Bonus Distributor.

c. Supervisor

Supervisor dapat Read Report dari Demonstrator dan CRUD Aktivitas.

d. Distributor

Distributor dapat Read Bonus Distributor.

e. Demonstrator

Demonstrator dapat CRUD Report.

4) CRUD Customer

CRUD Customer dapat melakukan Create dan Read QNA.

c. Searching

Searching adalah sebuah metode pencarian guna menemukan data atau informasi yang sedang dicari di dalam sebuah aplikasi.

d. Logout

Logout adalah proses keluar dari akun atau layanan aplikasi tersebut.

3. Biaya Proyek

Tabel 3. 3 Biaya Proyek

No	Uraian	Banyak	/fitur	Harga	SubTotal
----	--------	--------	--------	-------	----------

1	Project Management				
	Project Manager	1	875.000	875.000	1.175.000
	Project Team Member	1	300.000	300.000	
2	Bisnis Analyst				
	Usecase	17	60.000	1.020.000	1.185.000
	ERD	1	60.000	60.000	
	Analisa Proses Alur Data	7	15.000	105.000	
3	UI/UX	17	55.000	935.000	935.000
4	Programer				
	Penambahan Database	6	15.000	90.000	3.200.000
	Login	1	185.000	185.000	
	CRUD	10	260.000	2.600.000	
	Searching	1	150.000	150.000	
	LogOut	1	175.000	175.000	
5	Pengujian				
	Pengujian Aplikasi	7	100.000	700.000	950.000
		1	150.000	150.000	
	Panduan Pemakaian Aplikasi	1	100.000	100.000	
6	Upload ke Google PlayStore	1	400.000	400.000	400.000
7	Administrasi	1	100.000	100.000	100.000
8	Meeting				
	Kuota	1	100.000	100.000	200.000
	Konsumsi	1	100.000	100.000	
Total					8.145.000

D. Rencana Jadwal Pelaksanaan Proyek

Tabel 3. 4 Rencana Jadwal Pelaksanaan Proyek

No	Tahapan	Penjadwalan proyek											
		Mei			Juni				Juli				Agustus
		2	3	4	1	2	3	4	1	2	3	4	1
1	Requirements Analysis												
	a. Wawancara												

	b. Use Case																			
2	Design																			
	a. System & Software Design																			
3	Development																			
	a. Implementasi																			
	b. Unit Testing																			
4	Testing																			
	a. Uji Sistem																			
5	Maintenance																			
	a. Pengoperasian Aplikasi																			

BAB IV
PELAKSANAAN MPTI

A. Realisasi Pembagian Tugas

Realisasi pembagian tugas proyek tim brocode telah dilaksanakan oleh anggota tim sesuai dengan tugasnya masing-masing.

1. Vintya Dewi Anjani (*Project Manager*)

Tabel 4. 1 Realisasi Project Manager

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan
		Hari, Tanggal	Durasi (menit)		
1.	Melakukan pencarian dosen pembimbing	Selasa, 21/03/2023	10 menit	Mendapatkan dosen pembimbing	-
2.	Melaksanakan rapat ke-1 (kelompok)	Selasa, 14/03/2023	90 menit	Menetapkan jobdesc dan pencarian klien	-
3.	Bimbingan dengan dosen pembimbing terkait jobdesc	Kamis, 23/03/2023	10 menit	Dosen pembimbing mengarahkan untuk mengumpulkan tiap jobdesc pada Google Drive	-
4.	Melaksanakan rapat ke-2 (kelompok)	Jum'at, 24/03/2023	90 menit	Telah mendapatkan klien untuk membuat web company profile dan pembahasan proposal	-

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan
		Hari, Tanggal	Durasi (menit)		
5.	Melaksanakan rapat ke-3 (kelompok)	Rabu, 31/03/2023	90 menit	Revisi isi proposal	-
6.	Melaksanakan rapat ke-4 (kelompok)	Sabtu, 08/04/2023	40 menit	Perubahan proyek dari web company profile ke aplikasi mobile	Beresiko ganti klien
7.	Melakukan bimbingan dengan dosen pembimbing terkait perubahan proyek	Senin, 10/04/2023	15 menit	Dosen pembimbing menyarankan untuk diambil saja proyeknya, asalkan pembayarannya sesuai	Klien tidak jadi diganti
8.	Melakukan bimbingan dengan dosen pembimbing terkait isi proposal	Sabtu, 15/05/2023	60 menit	Revisi isi proposal dan penambahan biaya	-
9.	Melaksanakan rapat Bersama klien	Senin, 08/05/2023	60 menit	Penambahan fitur aplikasi dan perubahan judul menjadi aplikasi komersial	-
10.	Melaksanakan rapat ke-5 (kelompok)	Senin, 08/05/2023	60 menit	Penambahan fitur dan biaya	Perubahan jadwal pengerjaan
11.	Melakukan bimbingan dengan dosen pembimbing	Selasa, 09/05/2023	60 menit	Isi proposal diperinci dan biaya ditambah serta diperinci	Perubahan jadwal pengerjaan
12.	Melaksanakan rapat Bersama klien	Senin, 15/05/2023	45 menit	Pembahasan fitur lebih lanjut, jadwal pengerjaan, dan dana disetujui klien	-
13.	Melaksanakan rapat ke-6 (kelompok)	Selasa, 16/05/2023	120 menit	Revisi isi proposal, perubahan dana	Fitur yang diminta klien banyak, sehingga dana berubah

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan
		Hari, Tanggal	Durasi (menit)		
14.	Melakukan bimbingan dengan dosen pembimbing	Jum'at, 19/05/2023	15 menit	Dosen pembimbing menyetujui anggaran dana	-
15.	Melakukan demo web klien	Senin, 22/05/2023	90 menit	Penambahan fitur yang disesuaikan dengan struktur database web milik klien.	Perubahan jadwal pengerjaan dan anggaran biaya
16.	Melakukan bimbingan dengan dosen pembimbing terkait isi proposal	Rabu, 24/05/2023	15 menit	Penambahan analisis kebutuhan sistem	-
17.	Mengirim proposal ke klien	Kamis, 25/05/2023	60 menit	Klien menyetujui proposal	-
18.	Mengirim MoU ke klien	Kamis, 25/05/2023	5 menit	Klien memeriksa isi MoU terlebih dahulu	-
19.	Melakukan login website klien	Sabtu, 27/05/2023	30 menit	Fitur yang ada pada website berbeda dengan isi proposal	Perubahan jadwal pengerjaan dan anggaran biaya
20.	Melaksanakan rapat ke-7 (kelompok)	Senin, 29/05/2023	120 menit	Jadwal pengerjaan dan anggaran biaya berubah	-
21.	Mengirimkan fitur-fitur aplikasi yang akan dibuat ke klien	Rabu, 31/05/2023	5 menit	Klien menyetujui fitur-fitur yang akan dibuat	-
22.	Mengirim proposal kembali ke klien	Senin, 05/06/2023	15 menit	Klien meminta pengurangan dana dari Rp 9.650.000 ke Rp 6.600.000	Terdapat perubahan dana
23.	Melakukan bimbingan dengan dosen pembimbing terkait	Rabu, 07/06/2023	20 menit	Dosen pembimbing meminta tim berdiskusi terkait dana yang	Perubahan anggaran biaya

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan
		Hari, Tanggal	Durasi (menit)		
	pengajuan pengurangan dana oleh klien			diajukan klien, apakah dana sesuai dengan yang dikerjakan?	
24.	Melaksanakan rapat ke-8 (kelompok)	Rabu, 07/06/2023	45 menit	Anggaran biaya berkurang menjadi Rp 8.800.000	Anggaran biaya berkurang
25.	Mengirimkan proposal kembali ke klien	Rabu, 07/06/2023	7 menit	Klien meminta pengurangan dana kembali	-
26.	Melanjutkan rapat ke-8 (kelompok)	Rabu, 07/06/2023	45 menit	Anggaran biaya berkurang menjadi Rp 8.145.000	Anggaran biaya berkurang
27.	Mengirimkan proposal kembali ke klien	Rabu, 07/06/2023	10 menit	Klien menyetujui proposal	Klien meminta <i>soft launching</i> pada tanggal 10 Juli 2023
28.	Mengirim MoU ke klien	Rabu, 07/06/2023	5 menit	Klien memeriksa isi MoU terlebih dahulu	-
29.	Klien menandatangani MoU	Rabu, 14/06/2023	5 menit	MoU telah ditandatangani	-
30.	Mengirim <i>invoice</i> ke klien	Rabu, 14/06/2023	5 menit	<i>Invoice</i> telah dikirim	-
31.	Mengirimkan file .apk untuk kebutuhan <i>Soft launching</i> klien	Rabu, 12/07/2023	5 menit	File .apk diterima klien	-
32.	Melaksanakan rapat ke-9 (kelompok)	Selasa, 18/07/2023	120 menit	Pembahasan kendala-kendala yang terjadi	Terdapat kendala dalam penguploadan aplikasi ke Google PlayStore

2. Tatik Nafiaturohmah (Sekretaris dan Tester)

Tabel 4. 2 Realisasi Sekretaris dan Tester

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan
		Hari, Tanggal	Durasi (menit)		
1.	Notulensi rapat perdana kelompok	Selasa, 14/03/2023	90 menit	Menentukan jobdesc dan mencari klien	-
2.	Membuat tempat pengumpulan data untuk masing-masing jobdesc pada google drive	Jum'at, 24/03/2023	10 menit	Sebagai tempat pengumpulan data dalam google drive	-
3.	Notulensi rapat ke-2 kelompok	Jum'at, 24/03/2023	90 menit	Pembahasan proposal, menentukan fitur, jadwal, dan anggaran dana	-
4.	Pembuatan Proposal	Minggu, 26/03/2023	60 menit	Membuat proposal proyek company profile berbasis website	-
5.	Notulensi rapat kelompok ke-3	Rabu, 31/03/2023	90 menit	Pembahasan isi proposal lebih lanjut, melakukan perubahan fitur dan revisi jadwal	-
6.	Notulensi rapat kelompok ke-4	Sabtu, 08/04/2023	40 menit	Perubahan aplikasi sesuai permintaan klien dari website ke mobile	-
7.	Notulensi rapat Bersama klien ke-1	Senin, 08/05/2023	60 menit	Penambahan fitur dan perubahan judul proyek dari "pembuatan	-

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan
		Hari, Tanggal	Durasi (menit)		
				website company profile PT. Cakra Agro Indonesia” menjadi “pembuatan aplikasi komersial PT. Cakra Agro Indonesia berbasis mobile”	
8.	Notulensi rapat kelompok ke-5	Senin, 08/05/2023	60 menit	Terdapat penambahan fitur dan biaya, login untuk 3 jenis yaitu admin, user customer, dan user staff	Perubahan jadwal dikarenakan fitur bertambah
9.	Notulensi rapat Bersama dosen pembimbing (offline)	Senin, 09/05/2023	45 menit	Terdapat penambahan fitur dan perincian biaya	Terdapat penambahan fitur dan perincian biaya
10.	Membuat logbook kelompok	10/05/2023	20 menit	Mengisi logbook kelompok bimbingan	-
11.	Notulensi rapat Bersama klien ke-2	Senin, 15/05/2023	45 menit	Perubahan fitur pada login, untuk login harus ada verifikasi dari admin untuk masuk ke ruang diskusi	-
12.	Notulensi rapat kelompok ke-6	Selasa, 16/05/2023	120 menit	Revisi proposal, usecase, dana, fasilitas, CRUD lebih diperinci	-
	Notulensi rapat kelompok ke-7	Senin, 28/05/2023	120 menit	Revisi dana karena terdapat banyak penambahan fitur	-

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan
		Hari, Tanggal	Durasi (menit)		
	Notulensi rapat kelompok ke-8	Rabu, 08/06/2023	45 menit	Revisi dana karena klien meminta pengurangan dana sehingga terdapat penurunan harga per fitur	-
	Pembuatan MOU	Rabu, 08/06/2023	30 menit	MOU selesai dibuat	-
	Mulai membuat laporan MPTI	Minggu, 16/07/2023	30 menit	Mengerjakan bab 1	-
	Notulensi rapat kelompok ke-9	Selasa, 18/06/2023	120 menit	Pembahasan kendala-kendala yang terjadi	Terdapat kendala dalam penguploadan aplikasi ke google play store
	Melanjutkan laporan MPTI	rabu, 19/07/2023	48 menit	Mengerjakan bab 2 dan 3	-
	Notulensi Bimbingan kelompok Bersama dosen pembimbing	Jumat, 21/07/2023	20 menit	Konsultasi mengenai laporan	-
	Melanjutkan laporan MPTI	Minggu, 23/07/2023	60 menit	Mengerjakan bab 4	-
	Melengkapi laporan MPTI	Sabtu, 05/08/2023	60 menit	Melengkapi bab 1-5	
	Melengkapi laporan MPTI	Sabtu, 12/08/2023	80 menit	Melengkapi lampiran MPTI	

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan
		Hari, Tanggal	Durasi (menit)		
	Membuat test case untuk pengujian sistem	Jumat, 11/08/2023	2 jam	Membuat tabel test case customer	
	Membuat test case pengujian sistem	Sabtu, 12/08/2023	5 jam	Membuat tabel test case Super Admin	

3. Duwi Oktoviyanti (Bendahara dan *Digital Marketing*)

Tabel 4. 3 Realisasi Bendahara dan Digital Marketing

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan
		Hari, Tanggal	Durasi (menit)		
1.	Membuat rangkaian anggaran dana	Senin, 28/03/2023	120 menit	Membuat rangkaian dana	-
2.	Presentasi kepada klien	Senin, 08/05/2023	30 menit	Melakukan presentasi proposal kepada client	-
3.	Revisi rangkaian anggaran dana	Selasa, 09/05/2023	60 menit	Melakukan perubahan pada RAD menjadi lebih rinci	-
4.	Revisi rangkaian anggaran dana	Rabu, 10/05/2023	60 menit	Melakukan perbaikan pada RAD	-
5.	Revisi rangkaian anggaran dana	Senin, 15/05/2023	90 menit	Melakukan perubahan RAD karena pengurangan fitur	-
6.	Membuat template invoice	Selasa, 16/05/2023	120 menit	Membuat template invoice untuk diberikan kepada client	-
7.	Revisi invoice	Kamis, 19/05/2023	60 menit	Melakukan revisi pada template invoice	-
8.	Membuat invoice pembayaran tahap 2	senin, 31/06/2023	30	Invoice pembayaran tahap 2	-
9.	Melakukan pembayaran untuk upload ke google play store	Jumat, 14/07/2023	15	Melakukan upload apk pd google playstore	-

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan
		Hari, Tanggal	Durasi (menit)		
10.	Membayar gaji anggota mpti sesuai jobde pada pembayaran tahap pertama	Kamis, 12/08/2023	30	Membayar gaji anggota mpti sesuai jobdesk pada pembayaran tahap 1	-

4. Miftaql Fiqi Firmansyah (Programmer)

Tabel 4. 4 Realisasi Programmer

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan
		Hari, Tanggal	Durasi (Jam)		
1.	Implementasi UI menjadi program	Senin, 15/05/2023	2 Jam	UI/UX sudah terimplementasi	-
2.	Membuat arsitektur program	Senin, 22/05/2023	4 Jam	Menggunakan Retrofit sebagai API	-
3.	Implementasi UI superadmin kedalam program	Rabu, 12 Juli 2023	3 jam	Berhasil Terimplementasi	-
4.	consume API untuk membaca data pada database untuk super admin	Jumat, 14 Juli 2023 s.d sabtu 15 Juli 2023	1 hari	Berhasil Terimplementasi	-
5.	menyempurnakan UI superadmin	Senin, 17 Juli 2023	2 jam	Berhasil Terimplementasi	-
6.	membuat fungsi create data dari superadmin untuk masuk kedalam database	Kamis, 20 Juli 2023 s.d Minggu 23 Juli 2023	3 Hari	Berhasil Terimplementasi	-
7.	menyempurnakan UI input data pada super admin	Selasa, 25 juli 2023	1 jam	Berhasil Terimplementasi	-

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan
		Hari, Tanggal	Durasi (Jam)		
8.	membuat fungsi update pada superadmin	Rabu, 26 Juli 2023	1 hari	Berhasil Terimplementasi	-
9.	Implementasi ui admin	Jumat, 28 Juli 2023	2 jam	Berhasil Terimplementasi	-
10.	consume api untuk menampilkan data pada database ke tampilan	Minggu, 30 Juli 2023	1 hari	Berhasil Terimplementasi	-
11.	menyempurnakan ui admin	Senin, 31 Juli 2023	1 jam	Berhasil Terimplementasi	-
12.	membuat fungsi create data pada admin	Rabu, 2 Agustus 2023 s.d Jumat 4 Agustus 2023	2 hari	Berhasil Terimplementasi	-
13.	membuat fungsi update admin	Kamis, 3 Agustus 2023	3 Hari	Berhasil Terimplementasi	-
14.	membuat tampilan manager, supervisor, secretary	Sabtu, 5 Agustus 2023	3 jam	Berhasil Terimplementasi	-

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan
		Hari, Tanggal	Durasi (Jam)		
15.	menampilkan data manager, supervisor secretary	Senin, 7 Agustus 2023	1 hari	Berhasil Terimplementasi	-
16.	create data manager supervisor secretary	Rabu, 9 Agustus 2023	3 jam	Berhasil Terimplementasi	-
17.	implementasi UI demonstrator distributor	Kamis, 10 Agustus 2023	1 jam	Berhasil Terimplementasi	-
18.	membuat fungsi crud demonstrator distributor	Jumat, 11 Agustus 2023	4 jam	Berhasil Terimplementasi	-
19.	membuat fungsi delete	Sabtu, 12 Agustus 2023	1 jam	Berhasil Terimplementasi	-
20.	menyempurnakan crud	Sabtu, 12 Agustus 2023	1 hari	Berhasil Terimplementasi	-

5. Hikmatuz Zahra (*Business Analyst*)

Tabel 4. 5 Realisasi Business Analyst

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan
		Hari, Tanggal	Durasi (menit)		
1.	<ul style="list-style-type: none"> - Analisis Kebutuhan sistem company profile - Pembuatan usecase dan proses bisnis 	Senin, 03/04/2023	120 menit	<ul style="list-style-type: none"> - Klien PT. CAI dibuatkan web profile Perusahaan berbasis website - Terdiri dari 2 user dan customer - Customer dapat melihat Home, Produk, Event, Inovasi, Berita, Tentang kami PT. CAI 	<ul style="list-style-type: none"> - Direncanakan pembuatan website dinamis dimana admin memiliki akses CRUD memalui website sendiri - Admin perlu login terlebih dahulu - Akan dibuatkan database baru
2.	<ul style="list-style-type: none"> - Perubahan analisis dari company profile berbasis website menjadi berbasis mobile 	Rabu, 19/04/2023	60 menit	<ul style="list-style-type: none"> - Klien ingin tetap menggunakan website lama, sehingga data mobile dipindahkan ke basis mobile - Customer dapat masuk aplikasi mobile tanpa login dan searching data di aplikasi mobile 	<ul style="list-style-type: none"> - Diubah menjadi pembuatan aplikasi mobile untuk customer, tidak perlu admin - Aplikasi mobile dimasukkan ke google playstore

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan
		Hari, Tanggal	Durasi (menit)		
3.	<ul style="list-style-type: none"> - Analisis kebutuhan aplikasi diubah menjadi 3 jenis user - Perubahan usecase dan pembuatan activity diagram 	Minggu, 14/05/2023	120 menit	<ul style="list-style-type: none"> - User diubah menjadi 3 jenis user yaitu customer, admin. Dan staff FA - Customer, admin, dan staff harus login terlebih dahulu untuk masuk aplikasi - Menambahkan menu QnA bagi customer yang ingin bertanya terkait produk - Staff FA mendapatkan akses CRUD pada Homepage, Event, dan QnA 	<ul style="list-style-type: none"> - Aplikasi akan dibuat 3 jenis user : aplikasi customer, aplikasi admin, dan aplikasi staff FA - Admin akan diregistrasi akun staff FA
4.	<ul style="list-style-type: none"> - Perubahan usecase dan activity diagram - Analisis kebutuhan aplikasi diubah menjadi 4 user - Analisis database website milik klien PT. CAI 	Sabtu, 27/05/2023	120 menit	<ul style="list-style-type: none"> - User ditambahkan menjadi 4 jenis yaitu customer, admin, staff master, dan superadmin - SuperAdmin akan memiliki akses penuh aplikasi dan memiliki log history kegiatan admin - Admin memiliki akses CRUD data - Staff master akan dibedakan menjadi : manager, secretary, supervisor, distributor, dan demonstrator - Customer tidak perlu login aplikasi 	<ul style="list-style-type: none"> - Aplikasi akan dibuat 4 jenis tampilan : aplikasi customer, aplikasi superadmin, aplikasi admin, dan aplikasi staff master

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan
		Hari, Tanggal	Durasi (menit)		
				- Superadmin, admin, dan staff harus login terlebih dahulu	
5.	Mengupdate analisis database dengan menambah table baru	Kamis, 01/06/2023	120 menit	Penambahan table baru untuk Menu Log User dan Admin, dan tabel baru untuk menu QnA dan Kolom Pertanyaan	-

6. Nizhaambiq Elbanny (*Programmer dan Tester*)

Tabel 4. 6 Realisasi Programmer dan Tester

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan
		Hari, Tanggal	Durasi (Jam)		
1.	Menyiapkan tools yang akan digunakan berdasarkan proyek akan dikerjakan (rapat 1 dan 2)	Sabtu, 25/3/2023	2 Jam	Tools yang akan digunakan sudah ditentukan	-
2.	Menyiapkan kembali tools yang akan digunakan karena jenis proyek berbeda	Senin, 10/04/2023	3 Jam	Tools yang akan digunakan sudah ditentukan	Menggunakan tools yang berbeda dengan yang awal karena proyek berubah judul
3.	Membantu <i>business analyst</i> melakukan Analisa sistem	Jum'at, 26/05/2023	3 Jam	Diagram aktivitas sistem	-
4.	Membantu <i>business analyst</i> Melakukan Analisa untuk kebutuhan database	Kamis 01/06/2023	2 Jam	Analisa Kebutuhan Database	-
5.	Membuat Rancangan Database berdasarkan Analisa kebutuhan database yang sebelumnya dikerjakan	Sabtu 03/06/2023	3 Jam	Rancangan Database	-

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan
		Hari, Tanggal	Durasi (Jam)		
6.	Melakukan Implementasi rancangan databse ke localhost yang digunakan untuk pengujian database (MySQL)	Senin 05/06/2023	2 Jam	Database MySQL (localhost)	Database belum diuji
7.	Melakukan pengujian terhadap database (MySQL localhost)	Selasa 06/06/2023	2 Jam	Database MySQL sudah teruji (localhost)	-
8.	Membuat web hosting untuk database dummy yang akan digunakan untuk consume api tahapan pengujian	Kamis 08/06/2023	2 Jam	Web Hosting untuk consume api tahap pengujian	-
9.	Implementasi database mysql localhost ke web hosting yang digunakan untuk consume api tahap pengujian	Jumat 09/06/2023	2 Jam	Web Hosting untuk consume api tahap pengujian dengan database mysql	-
10.	Konfigurasi Akses Database di 000webhosting	Sabtu 10/06/2023	1 Jam	Akses database pada web hosting sudah dikonfigurasi	-
11.	Membuat Project CodeIgniter	Minggu 11/06/2023	2 Jam	Project Codeigniter	-
12.	Konfigurasi Database di CodeIgniter	Senin 12/06/2023	2 Jam	Project Codeigniter dengan database	-

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan
		Hari, Tanggal	Durasi (Jam)		
13.	Membuat Model, View, Controller di CodeIgniter	Selasa 13/06/2023	3 Jam	Project Codeigniter dengan MVC	-
14.	Membuat Fungsi CRUD pada tiap MVC yang sudah dibuat di Codeigniter	Kamis 15/06/2023	5 Jam	MVC Project Codeigniter dengan Fungsi CRUD	-
15.	Membuat Endpoint API di Controller	Selasa 20/06/2023	3 Jam	Endpoint API	-
16.	Menguji API di Lingkungan Pengembangan	Kamis 22/06/2023	3 Jam	API yang sudah teruji secara lokal	-
17.	Melakukan Deploy CodeIgniter ke 000webhosting	Sabtu 24/06/2023	3 Jam	Web Host dengan API	-
18.	Uji Endpoint API di Lingkungan Produksi	Senin 26/06/2023	2 Jam	API pada web host sudah teruji	-
19.	Memberikan Akses Endpoint API kepada aplikasi klien yang sedang dikembangkan programmer	Selasa 27/06/2023	1 Jam		-

7. Taufiq Aditya Putra (UI/UX)

Tabel 4. 7 Realisasi UI/UX

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan
		Hari, Tanggal	Durasi (Jam)		
1.	Membuat desain tampilan customer	Senin, 17/04/2023	3 Jam	Dapat membuat beberapa fitur tampilan yaitu Homepage, Jenis Produk, Kegiatan, Tentang Perusahaan	Memperbaiki Desain agar lebih mudah mudah digunakan user
2.	Membuat desain tampilan customer	Rabu, 26/04/2023	2 Jam	Dapat Melengkapi fitur yang sudah ditambah sebelumnya yaitu Homepage, Berita, Jenis Produk, Halaman produk, Kegiatan, Tentang Perusahaan	Mempersimpel desain tampilan
3.	Menambahkan fitur pada Costumer	Jum'at, 12/05/2023	1 Jam	Menampilkan fitur pertanyaan yaitu FAQ, Kolom Pertanyaan	Merubah desain agar lebih menarik
4.	Membuat desain tampilan admin	Senin, 01/05/2023	2 Jam	Dapat menambahkan beberapa fitur yaitu CRUD Homepage, CRUD Jenis Produk, CRUD Kegiatan, CRUD Tentang Perusahaan	Mempermudah pengguna dalam input data
5.	Membuat Prototype	Kamis, 13/07/2023	3 Jam	Prototype aplikasi mobile PT. Cakra Agro Indonesia	-

B. Realisasi Jadwal Pelaksanaan

Tabel 4. 8 Realisasi Jadwal Pelaksanaan

No.	Nama Kegiatan	Minggu Pelaksanaan																Realisasi			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Ya/Tidak	%		
1	Requirements Analysis																				
	a. Wawancara	■																		Ya	100
	b. Use Case		■																	Ya	100
2	Design																				
	a. System & Software Design			■	■															Ya	100
3	Development																				
	a. Implementasi				■	■	■	■	■	■										Ya	100
	b. Unit Testing				■	■	■	■	■	■										Ya	100
4	Testing																				
	a. Uji Sistem										■	■								Ya	100

No.	Nama Kegiatan	Minggu Pelaksanaan															Jumlah	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Ya/Tidak	W.
2.	Manajemen																	
	a. Pengoperasian Sistem																12	100

Ngoyok, 24 September 2022

Disetujui,

Dosen Pembimbing (DPP)



Dan Wawan S.T., M.Eng

Dosen Pembimbing (DPP)



Yusuf Rizki Anwar

C. Keberlanjutan Proyek

Keberlanjutan proyek Aplikasi Komersial PT. Cakra Agro Indonesia Berbasis Mobile ini tim kami akan melaksanakan:

1. Melanjutkan pemrograman aplikasi yang belum terealisasi sepenuhnya.
2. Melakukan hosting pada aplikasi.
3. Pihak mitra mendapatkan *free maintenance* selama 3 bulan.

BAB V

PENUTUP

A. Kesimpulan

Pada laporan Pembuatan Aplikasi PT. Cakra Agro Indonesia Berbasis Mobile, kami telah melampirkan proses pengembangan aplikasi yang masih dalam tahap pengerjaan, seperti melakukan analisis kebutuhan pengguna, merancang desain antarmuka yang responsif dan mengimplementasikan beberapa fitur inti. Namun, ada beberapa kendala yang dihadapi selama proses pengembangan.

Salah satu kendala utama adalah pengalaman dan keterbatasan waktu. Tim kami terdiri dari anggota yang masih menempuh pendidikan S1 Informatika dimana masih dalam tahap pembelajaran dan jadwal proyek yang berubah dikarenakan terdapat penambahan fitur serta menyesuaikan dengan database milik klien. Selain itu, adanya tantangan teknis dalam mengimplemtasikan komponen-komponen penting seperti sistem backend dengan antarmuka frontend yang juga mempengaruhi kemajuan proyek.

Untuk mengatasi kehambatan tersebut, kami telah meningkatkan koordinasi tim dan target penyelesaian untuk bagian-bagian yang belum terealisasi tersebut adalah dalam seminggu mendatang. Dalam rencana lanjutan, kami akan fokus pada menyelesaikan sisa pekerjaan serta melakukan uji coba menyeluruh untuk memastikan stabilitas dan kualitas aplikasi sebelum peluncuran resmi kepada pengguna.

Meskipun proses pengembangan belum selesai, kami telah belajar banyak dari tantangan-tantangan yang dihadapi. Pelajaran penting yang kami dapatkan adalah pentingnya perencanaan yang matang, komunikasi yang efektif, serta fleksibilitas dalam menyesuaikan dengan perubahan kebutuhan proyek.

B. Saran

Berikut terdapat beberapa saran dari tim BROCODE:

1. Memperbaiki perencanaan, mengevaluasi kembali perencanaan proyek untuk memastikan bahwa jadwal telah direncanakan dengan matang.
2. Tetap fleksibel yang berarti dalam proyek pengembangan perangkat lunak, perubahan adalah hal yang wajar. Jadilah fleksibel dan siap menyesuaikan diri dengan kebutuhan atau permintaan pelanggan serta beradaptasi dengan perkembangan teknologi terbaru.
3. Meninjau kembali rencana resiko yang telah disusun sebelumnya dan identifikasi potensi hambatan baru yang mungkin muncul selama sisa proses pengembangan aplikasi. Menyiapkan langkah-langkah pengurangan resiko untuk menghadapi resiko-resiko tersebut.

LAMPIRAN

i. Proposal MPTI

Praktikum Aplikasi Komersial PT. Citra Agri Indonesia
Berkas 1/2020





BARCODE

A. Pembuatan Aplikasi Komersial Berbasis Mobile

Untuk membuat suatu aplikasi yang sukses dan menguntungkan, Anda harus memahami pasar yang target aplikasi tersebut untuk. Hal tersebut dapat memastikan program Anda mendapatkan bentuk awal yang sangat penting. Langkah pertama yang harus Anda lakukan dan pilih, yaitu berpikir mengenai aplikasi komersial tersebut.

B. Urutan yang harus dilakukan

"Urutan dalam Komersial Berbasis Mobile"

1. Plan

a. Tim

Langkah awal Anda harus lakukan itu adalah memilih tim dalam Anda aplikasi.

b. F2U2 (User, Need, Location, dan Design)

F2U2 digunakan untuk memahami kebutuhan, keinginan, pemenuhan, perilaku, dan tanggapan user.

F2U2 akan membantu Anda untuk memilih kebutuhan dan user.

c. Profitability dan nilai bisnis

Untuk melakukan profitabilitas dan biaya komersial ini pemerintah.

Profitabilitas akan menunjukkan yang akan dihasilkan dan waktu dapat digunakan juga, dengan cara sangat akurat.

d. Usual

Usual adalah proses menggunakan UI user dan jika hanya dengan lebih awal, lebih terapan.

e. User Question and Answer

Ini digunakan untuk mengetahui dan juga untuk dan menambahkan pertanyaan.

F2U2 akan membantu untuk meningkatkan hal-hal.

2. F2U2 User Review

F2U2 User Review dapat membantu F2U2 User Review dan F2U2 User Review.

3. F2U2 User

F2U2 User dapat membantu F2U2 User Review dan F2U2 User Review.

F2U2 User dapat membantu F2U2 User Review dan F2U2 User Review.

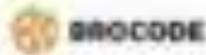
4. F2U2 User

F2U2 User dapat membantu F2U2 User.

a. Market

Market dapat membantu F2U2 User dan F2U2 User.

b. Strategy



Sebagai bagian dari Laporan dan Dashboard dan UEDD dalam Dashboard

- Laporan
- Laporan bagian dari Laporan dan Dashboard dan UEDD Admin
- Dashboard
- Dashboard bagian dari Dashboard
- Dashboard
- Dashboard bagian UEDD Report

- 1. UEDD Process
- UEDD Process dapat dilakukan secara manual

2. Reporting

Reporting adalah untuk melihat progress dan monitoring data yang dilakukan yang sedang dan akan dilakukan aplikasi

3. Logcat

Logcat adalah untuk melihat dan menganalisa log yang terjadi di aplikasi

2. Metode Aplikasi

a. Android Studio

3. Spesifikasi Sistem Aplikasi

- a. Jenis sistem operasi : Android 10.0
- b. Bahasa Pemrograman : Java
- c. Database : MySQL

4. Analisis Kelayakan Sistem

Kelayakan Finansial (FN)

No	Kode	Contoh
1	001 - F1	Aplikasi dapat memonitoring Puluhan ribu Produk di Homepage beserta manajemen setiap produk
2	001 - F2	Aplikasi dapat memonitoring Nama dan Demand setiap di Homepage
3	001 - F3	Aplikasi dapat melihat dan Update Produk beserta melihat profil dan pers produk
4	001 - F4	Aplikasi dapat melihat dan Update yang memonitoring volume, aktivitas dan Demand produk
5	001 - F5	Aplikasi dapat melihat dan Update yang memonitoring aktivitas Produk baru dan Detail produk
6	001 - F6	Aplikasi dapat memonitoring nama brand - Demand dan nama serta pemetaan produk yang dilakukan oleh PT. UN
7	001 - F7	Aplikasi dapat memonitoring nama, Detail - Aktivitas yang terjadi oleh Enginer Aplikasi yang dilakukan PT. UN



6	2025 - 45	Aplikasi dapat memformat nomor telepon + Nomor lain dengan menambahkan prefix sesuai dari PT CMI
7	2025 - 46	Aplikasi dapat memformat nomor telepon + Nomor Perusahaan untuk nomor yang berawak 0 PT CMI
14	2025 - 512	Aplikasi dapat memformat nomor FMS agar bisa (bisa) bisa: perbandingan yang dapat diakses oleh Customer untuk berlayar menggunakan PT CMI
15	2025 - 511	Aplikasi dapat memformat nomor Telepon yang bisa di call Perusahaan via web (WU, internal, Social Media dan Nomor PT CMI)

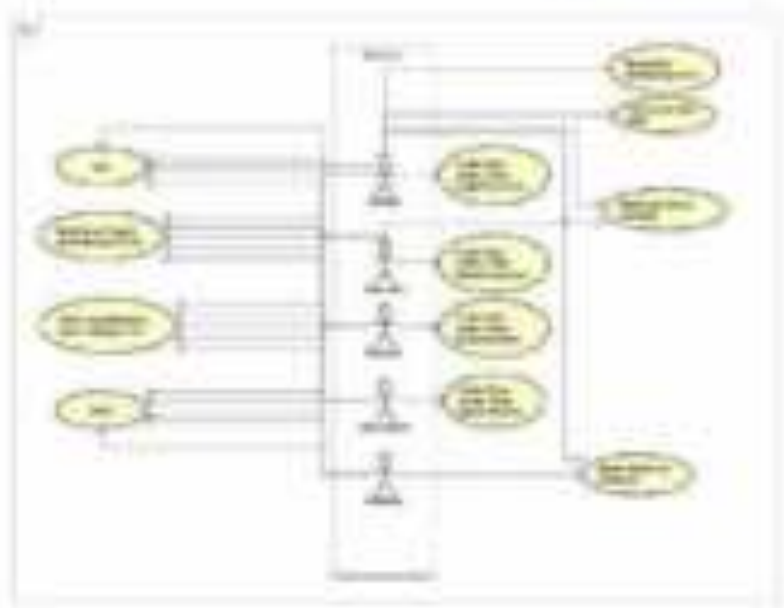
Estimasi Time Required (TRR)

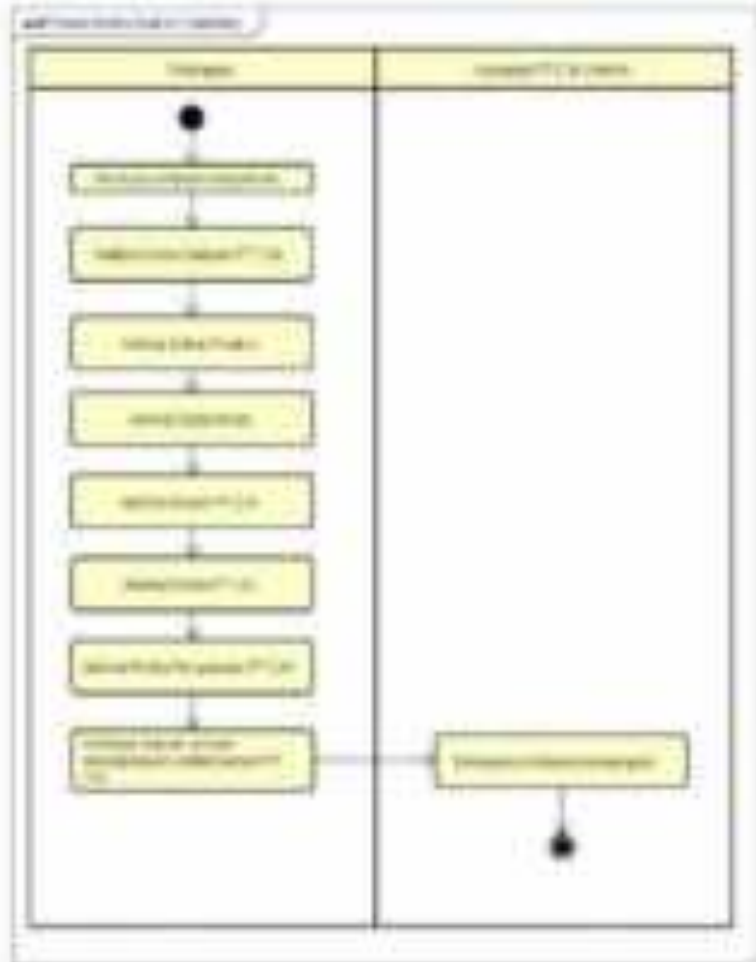
No.	Kode	Parameter	Detailnya
1	2025 - 51	Availability	Aplikasi dapat diakses 24 jam
2	2025 - 52	Flexibility	<ul style="list-style-type: none"> Aplikasi dapat dipindahkan dari menggunakan data yang sudah terdapat di server ke server lain yang sudah terdapat Aplikasi dapat menggunakan server lokal
3	2025 - 53	Efficiency	Aplikasi memiliki interface yang memudahkan dan menarik (mengerti) penggunaan (User Friendly)
4	2025 - 54	Portability	Aplikasi harus dapat dibawa dengan gampang ke perangkat lain (Android)
5	2025 - 55	Security	Aplikasi memiliki pengamanan data, informasi yang terakumulasi
6	2025 - 56	Response Time	Aplikasi dapat mengoptimalkan dan mempercepat data hingga output adalah UI/UX yang baik
7	2025 - 57	Scalability	<ul style="list-style-type: none"> Aplikasi memiliki kemampuan yang dapat digunakan Dapat lebih lanjut hingga dapat diakses oleh Super Admin dan Admin
8	2025 - 58	Reliability	Aplikasi memiliki kemampuan yang dapat diakses dan pengguna dapat berlayar dan berinteraksi
9	2025 - 59	Customization	Aplikasi dapat di custom menggunakan bahasa Indonesia

BROCODE

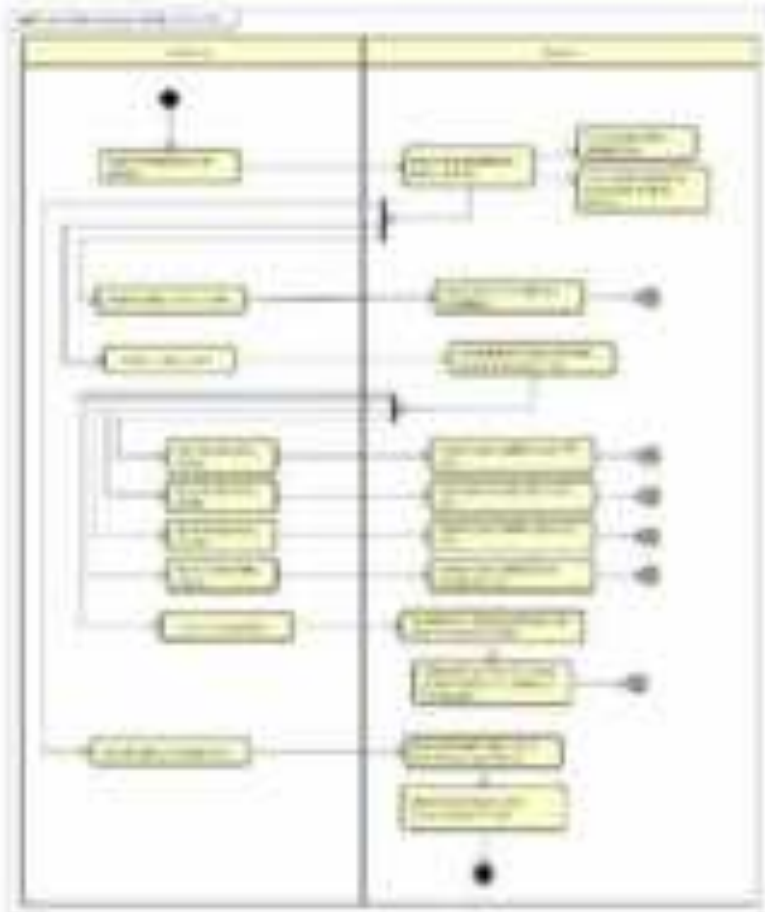
3. Buatlah ringkasan!
 - a. Apa itu kode QR? apa saja jenis-jenis kode QR?
 - b. Apa itu, kelebihan, dan kekurangan kode QR? apa saja jenis-jenis kode QR?
 - c. Apa itu kode QR? apa saja jenis-jenis kode QR?
 - d. Apa itu kode QR? apa saja jenis-jenis kode QR?
 - e. Apa itu, kelebihan, dan kekurangan kode QR? apa saja jenis-jenis kode QR?
 - f. Apa itu, kelebihan, dan kekurangan kode QR? apa saja jenis-jenis kode QR?
 - g. Apa itu, kelebihan, dan kekurangan kode QR? apa saja jenis-jenis kode QR?
 - h. Apa itu, kelebihan, dan kekurangan kode QR? apa saja jenis-jenis kode QR?
 - i. Apa itu, kelebihan, dan kekurangan kode QR? apa saja jenis-jenis kode QR?
 - j. Apa itu, kelebihan, dan kekurangan kode QR? apa saja jenis-jenis kode QR?
 - k. Apa itu, kelebihan, dan kekurangan kode QR? apa saja jenis-jenis kode QR?
 - l. Apa itu, kelebihan, dan kekurangan kode QR? apa saja jenis-jenis kode QR?
 - m. Apa itu, kelebihan, dan kekurangan kode QR? apa saja jenis-jenis kode QR?
 - n. Apa itu, kelebihan, dan kekurangan kode QR? apa saja jenis-jenis kode QR?
 - o. Apa itu, kelebihan, dan kekurangan kode QR? apa saja jenis-jenis kode QR?
 - p. Apa itu, kelebihan, dan kekurangan kode QR? apa saja jenis-jenis kode QR?
 - q. Apa itu, kelebihan, dan kekurangan kode QR? apa saja jenis-jenis kode QR?
 - r. Apa itu, kelebihan, dan kekurangan kode QR? apa saja jenis-jenis kode QR?
 - s. Apa itu, kelebihan, dan kekurangan kode QR? apa saja jenis-jenis kode QR?
 - t. Apa itu, kelebihan, dan kekurangan kode QR? apa saja jenis-jenis kode QR?
 - u. Apa itu, kelebihan, dan kekurangan kode QR? apa saja jenis-jenis kode QR?
 - v. Apa itu, kelebihan, dan kekurangan kode QR? apa saja jenis-jenis kode QR?
 - w. Apa itu, kelebihan, dan kekurangan kode QR? apa saja jenis-jenis kode QR?
 - x. Apa itu, kelebihan, dan kekurangan kode QR? apa saja jenis-jenis kode QR?
 - y. Apa itu, kelebihan, dan kekurangan kode QR? apa saja jenis-jenis kode QR?
 - z. Apa itu, kelebihan, dan kekurangan kode QR? apa saja jenis-jenis kode QR?







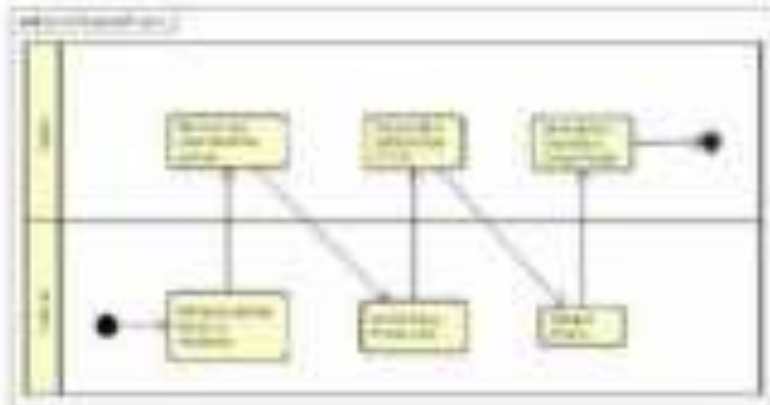
Process Model: System Model (UML)



Activity Diagram



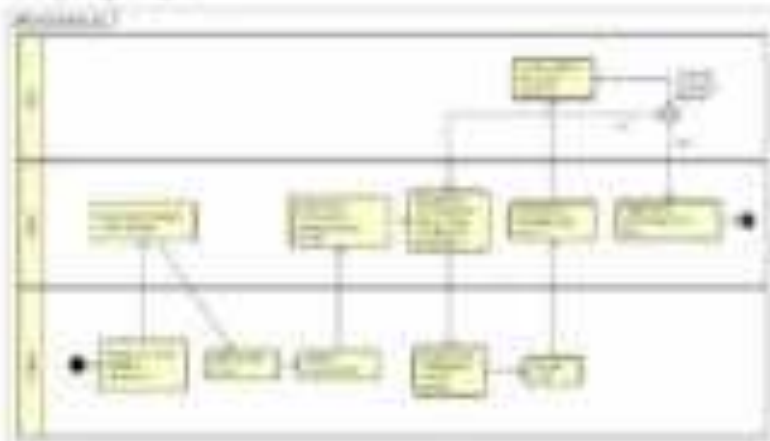
Activity Diagrams (Partial)



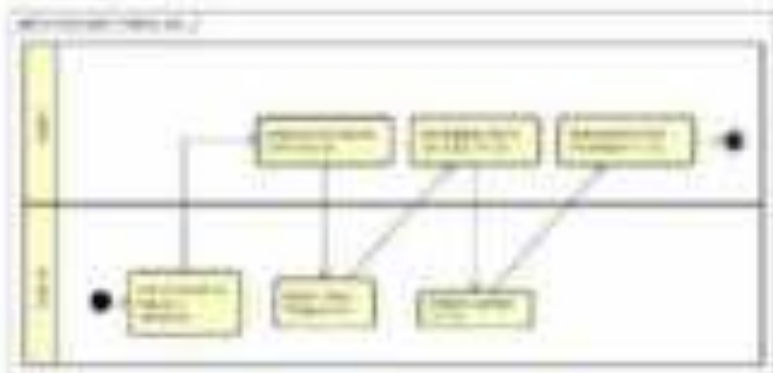
Activity Diagrams



Activity Diagrams

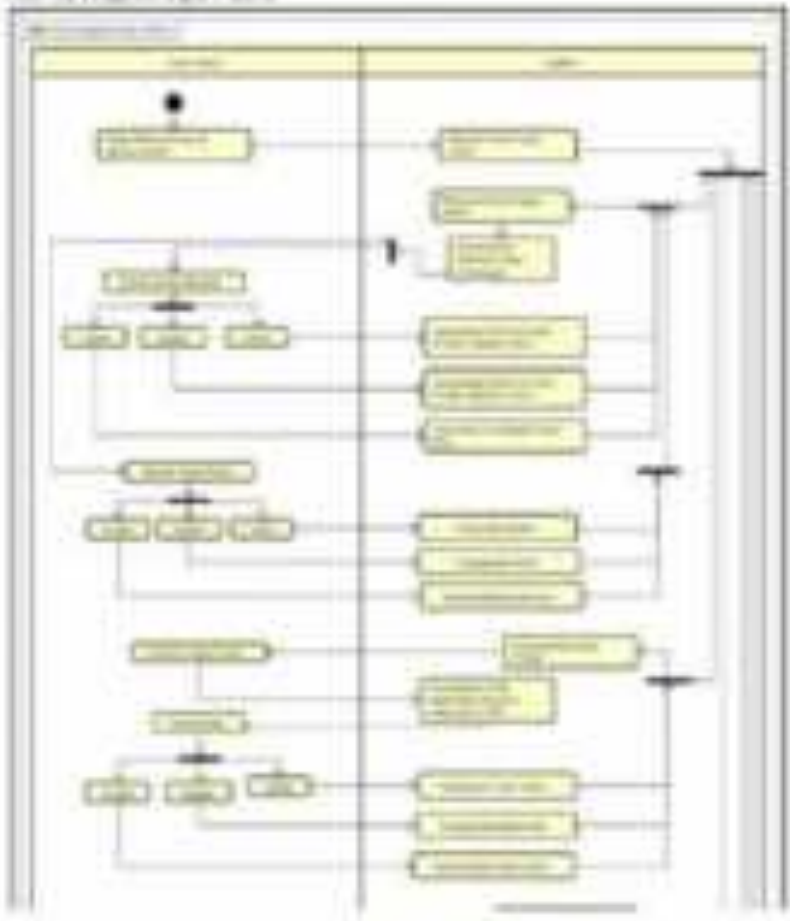


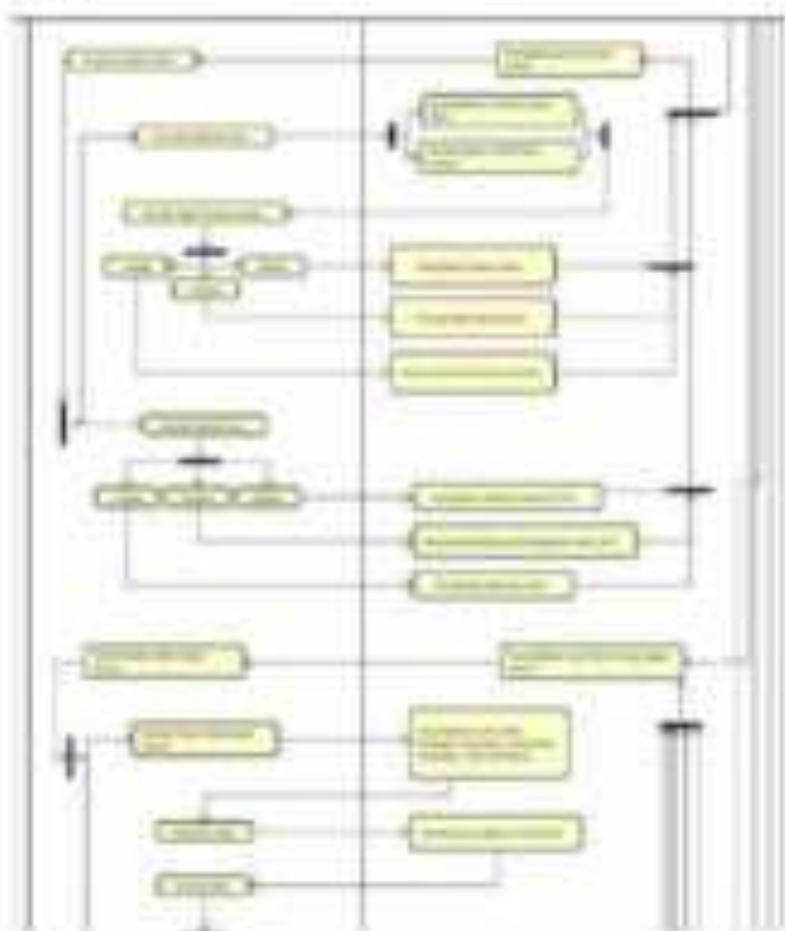
Activity Diagram: Finding Book




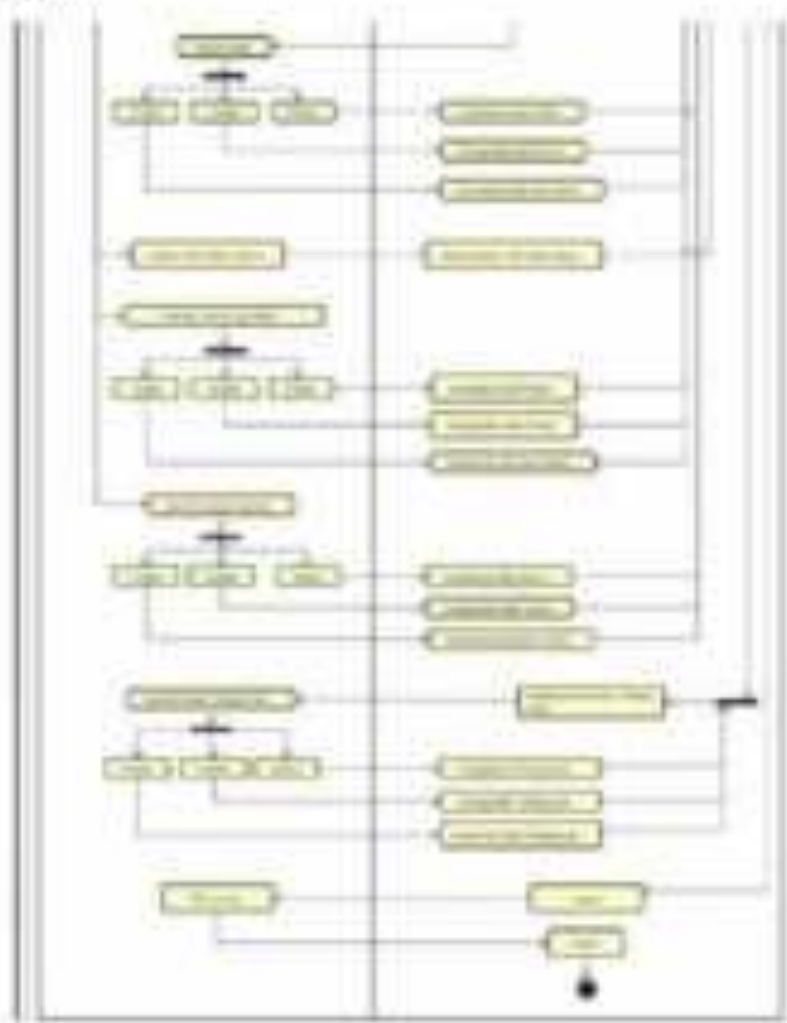
Activity Diagram: Finding Book





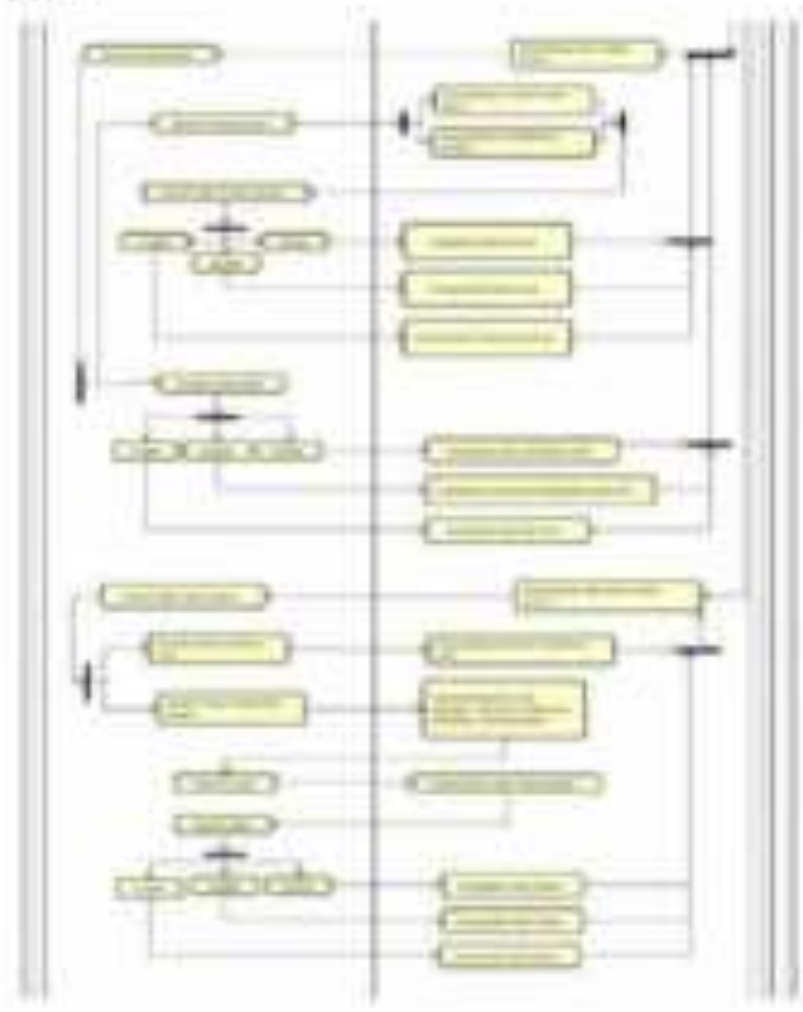


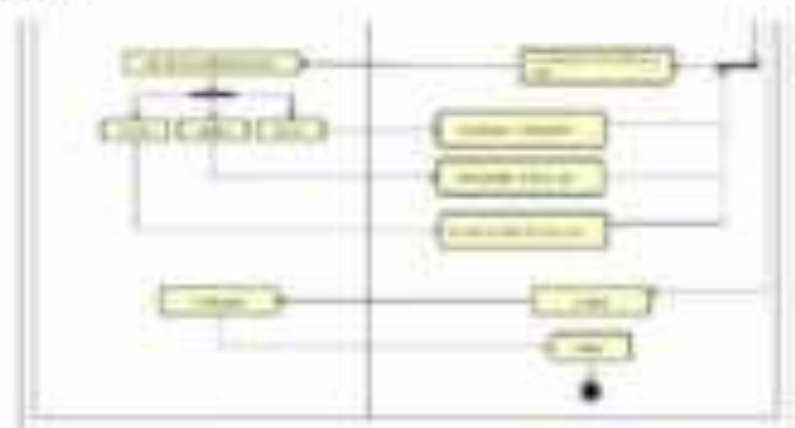
 BROCODE



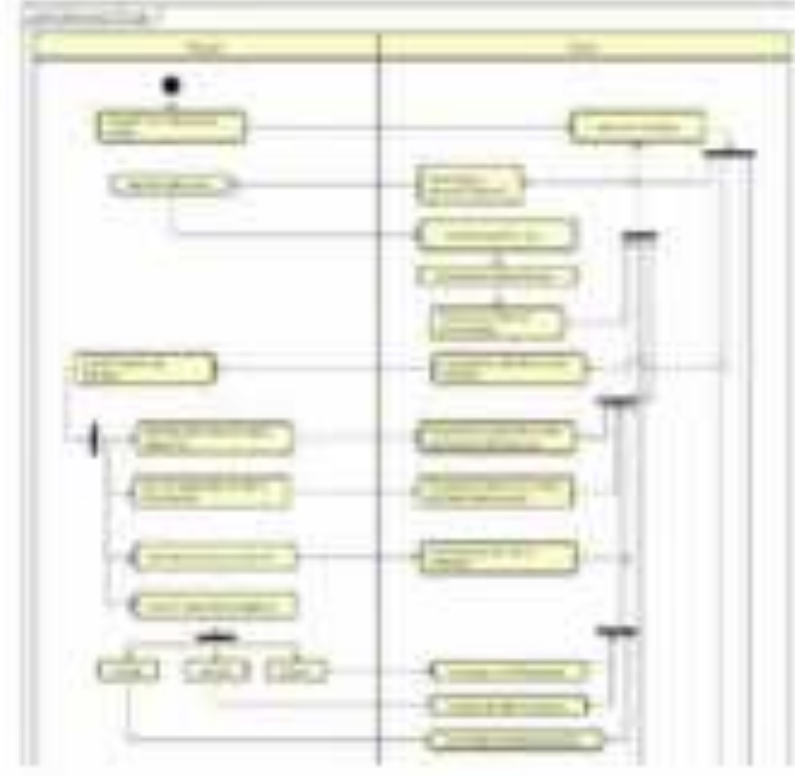
Activity Diagram: Action



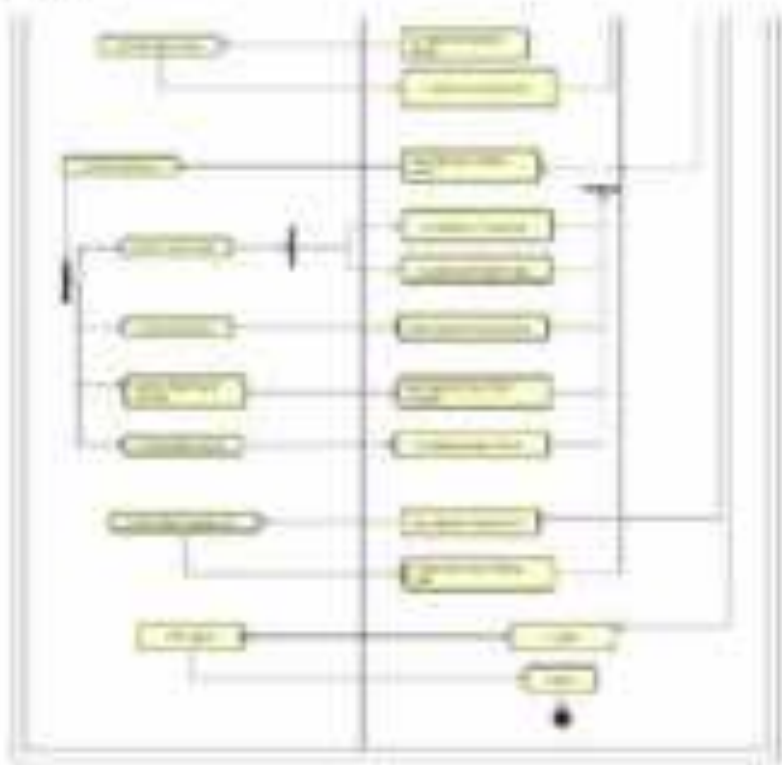


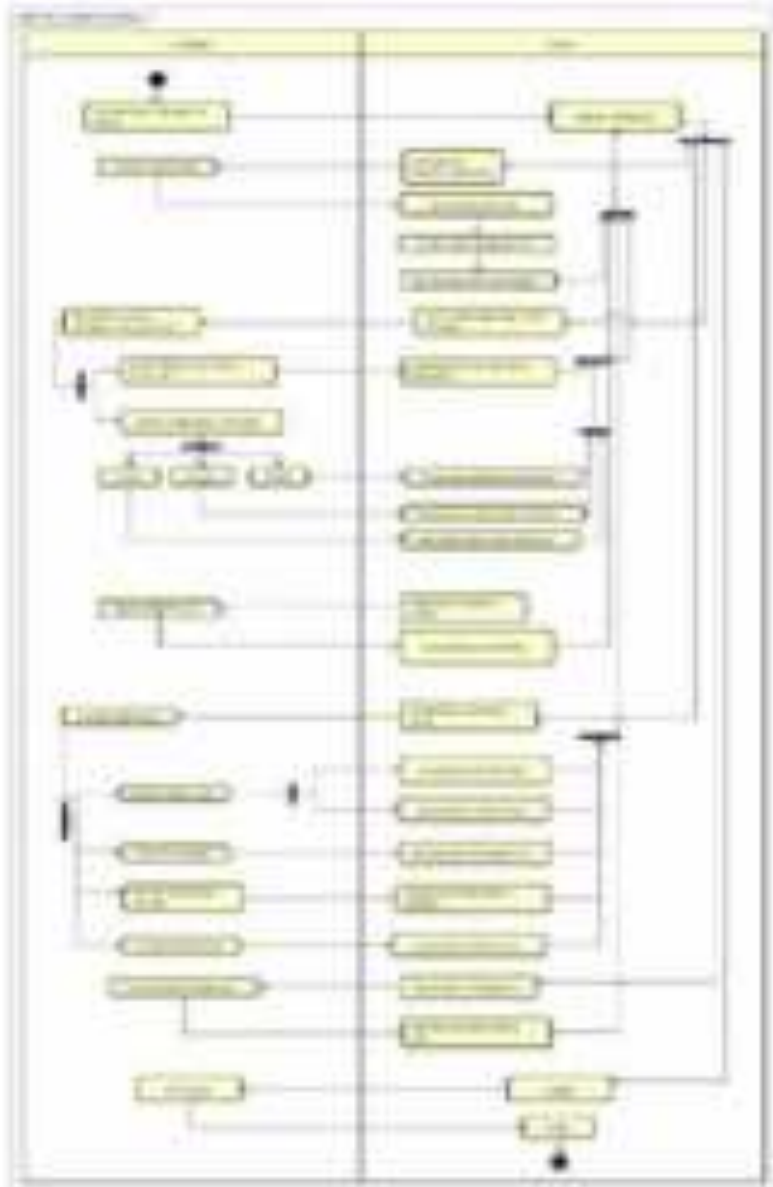


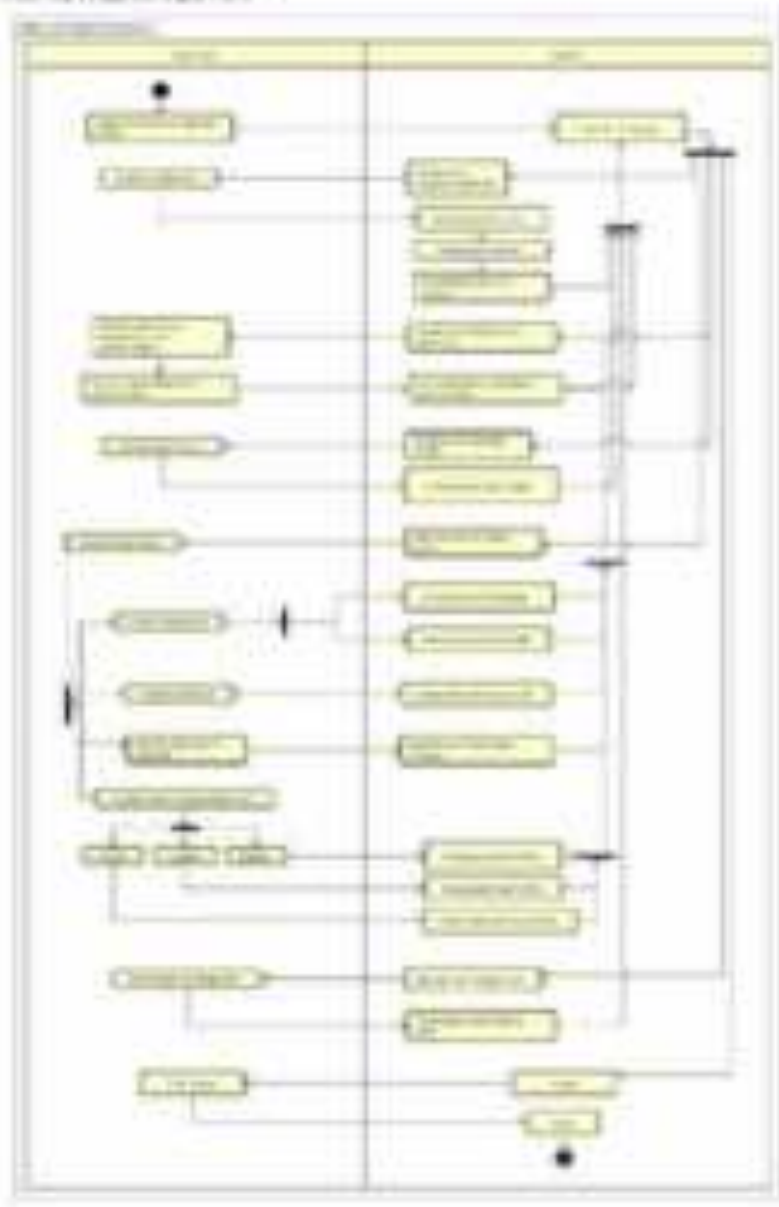
Leistungsübertragung

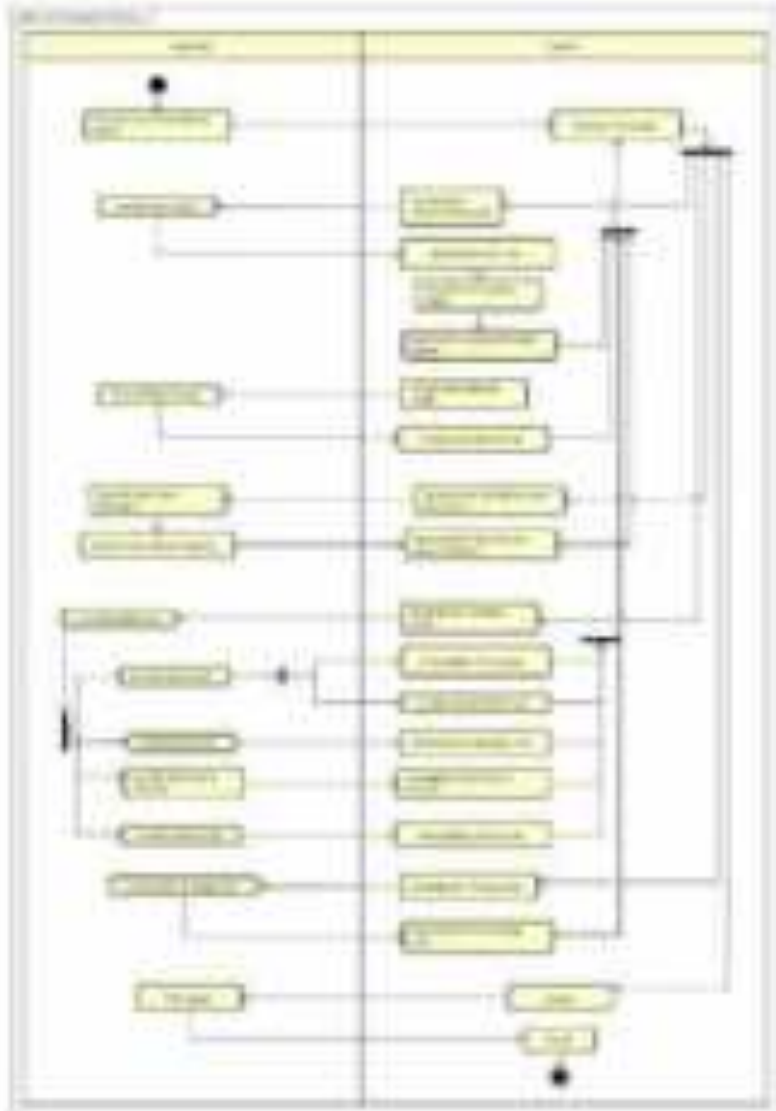


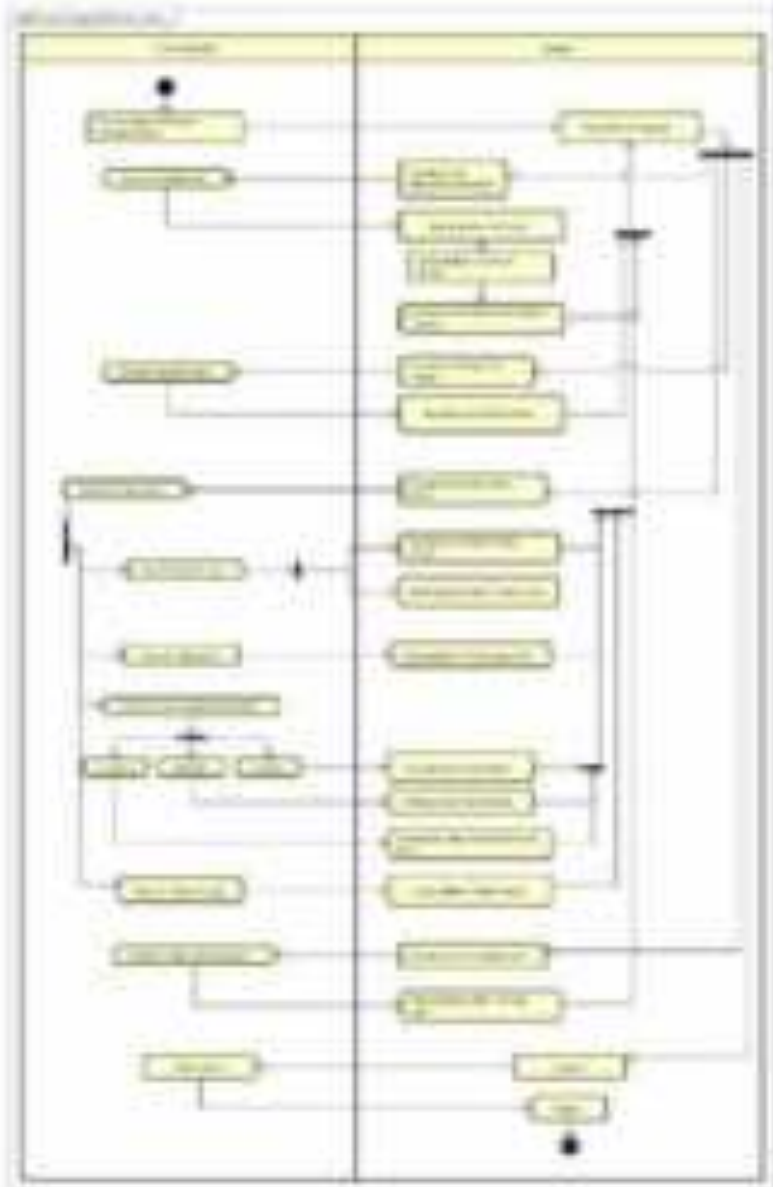
 BROCODE



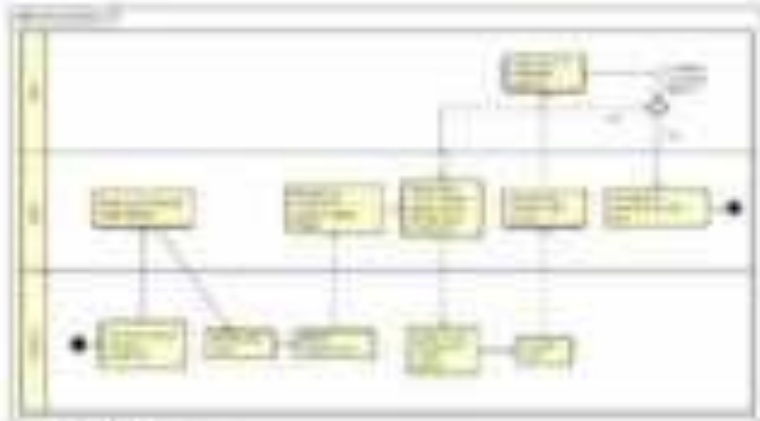








Activity Diagram



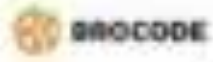
1. Design System Menu

a. Home Page

Fitur yang terdapat di halaman ini adalah halaman Home dan berlayar, sehingga pengguna dapat melihat dan klik, sehingga pengguna dapat melihat dan akses fitur



Detail Berlayar (lihat ini dari menu dan akses fitur di 116)
 Fitur yang terdapat di halaman ini adalah detail berlayar, sehingga pengguna dapat melihat dan akses fitur



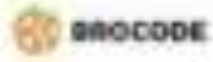
6. **Avak Prokha**

Filter yang mengatur & filter no print halaman web/ print print &



Menu Prokha (filter) to search and edit the page print & (10)

The yang filter & filter no print halaman web/ print print &



Final Product

Final product is a mobile application that provides information about products, including their names, descriptions, and images.



a. Menu Produk




Final Product

Final product of the application is a mobile application that can be used to track the quality of the product and the quality of the product.



a. Menu Aplikasi

 **BROCODE**

File yang terdapat di dalamnya akan berisi informasi data pribadi antara lain nomor ponsel, tanggal lahir, dan alamat.



Fileur Pribadi

File yang terdapat di dalamnya akan berisi informasi pribadi yang meliputi data hasil analisis PPT pada tanggal pengisian.



File yang terdapat di dalamnya akan berisi informasi pribadi yang meliputi data hasil analisis PPT pada tanggal pengisian.

BROCODE

Foto yang terlihat di halaman ini akan langsung terhubung ke semua profil pengguna dan terdapat di profil masing-masing pengguna lain.



Kolom Penawaran/Barang yang terlihat merupakan dari pengguna yang menggunakan aplikasi ini. Hal ini akan terhubung ke halaman masing-masing pengguna. Hal ini dapat di jual oleh siapa saja melalui aplikasi ini. Pengguna dapat menggunakan aplikasi ini untuk...



Foto yang terlihat di halaman ini akan langsung terhubung ke semua profil pengguna dan terdapat di profil masing-masing pengguna lain.

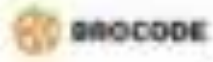
QR CODE

File yang terdapat di halaman ini sudah tersedia sebelum pengisian formulir di atas, tetapi untuk alasan keamanan, permohonan ini harus dipindai ulang untuk memastikan keabsahannya.



4. Scanning Procedures

File yang terdapat di halaman ini sudah terdapat di atas, tetapi untuk alasan keamanan, permohonan ini harus dipindai ulang untuk memastikan keabsahannya.



4. **Page Artikel**

Tentang PT Dahli SIA

Terdapat ringkasan di halaman ini yaitu tentang PT Dahli SIA, Sejarah Dahli SIA, Struktur PT Dahli SIA, dan lain-lain. Terdapat tombol untuk menambahkan



Terdapat Salaku Aktifitas (permainan), Terpadu Salaku (Data Statistik), Terpadu Salaku (QMS), Terpadu Salaku (Berkas), Terpadu Salaku (Pembelian), Terpadu Salaku (Jurnal), Terpadu Salaku (Yogyakarta), dan Terpadu Salaku (Kantor).



Hardware

Yang yang terdapat di dalamnya adalah: Salaku (yang terdapat di Salaku, Yogyakarta, Yogyakarta, dan Terpadu Salaku dan).

BARCODE



120) rapor perencanaan (tabel rencana masa depan)

File yang terdapat di dalam file yang sudah diinstal memiliki nama di bawah, Untuk file yang ada di dalam, maka akan terdapat laporan di bawah, dan file yang terdapat yang berbeda akan



Detail Report

File yang terdapat di dalam file yang sudah diinstal memiliki nama di bawah, Untuk file yang terdapat yang berbeda akan terdapat laporan di bawah, dan file yang terdapat yang berbeda akan



Halaman Utama
 Fitur yang terdapat di halaman ini adalah: Daftar barang, barang, Transaksi, dan juga ada Transaksi dan lain.



Halaman Utama
 Fitur yang terdapat di halaman ini adalah: Daftar barang, barang, Transaksi, dan juga ada Transaksi dan lain.

 BROCODE



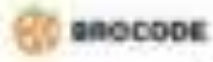
Example

Fill out the list of items in the order you want to add, before clicking on the green button.



Enter Photo

Fill out the list of items in the order you want to add, before clicking on the green button. Then click on the photo icon to add the photo.



Faktor Pemasok

Untuk dapat melakukan ini diperlukan beberapa data yang sudah tersedia. Untuk itu akan di berikan data tersebut dalam PDF.



Daftar Barang

Untuk dapat melakukan ini diperlukan beberapa data yang sudah tersedia. Untuk itu akan di berikan data tersebut dalam PDF.



C. Work Breakdown Structure (WBS)

Activity	Task Elements
A	Requirements Analysis <ul style="list-style-type: none">a. Requirementsb. User Use
B	Design <ul style="list-style-type: none">a. System & Software Design
C	Development <ul style="list-style-type: none">a. Implementationb. User Training
D	Testing <ul style="list-style-type: none">a. UAT/UBAT
E	Maintenance <ul style="list-style-type: none">a. Performance Support

a. Requirements Analysis

BROCODE

Pada proses ini program yang terdapat di brocode akan diunduh ke komputer pengguna sehingga informasi program tersebut akan tersimpan ke hard disk. Setelah pengunduhan selesai, brocode akan diinstal dengan berbagai nama antarmuka, seperti: *simulasi, virtual, simulasi, dan sebagainya*. Informasi yang diperoleh melalui brocode akan di simpan sebagai informasi dan akan disimpan yang dapat membuat kualitas informasi program dan program baik yang akan di simpan.

b) *Design, Develop & Software Design*

Pada proses ini akan melakukan *Development* dengan menggunakan bahasa yang ada untuk membuat dan mengkonstruksi serta desain yang akan digunakan. Program yang akan diinstal dengan tipe *software* menggunakan bahasa yang terdapat di brocode yang akan diinstal. Untuk itu juga akan membuat program yang akan diinstal. Informasi yang akan diinstal akan di simpan sebagai informasi dan akan di simpan.

c) *Design*

Pada proses ini akan melakukan *Design* yang akan diinstal dan akan diinstal dengan menggunakan bahasa yang ada untuk membuat dan mengkonstruksi serta desain yang akan digunakan. Program yang akan diinstal dengan tipe *software* menggunakan bahasa yang terdapat di brocode yang akan diinstal. Untuk itu juga akan membuat program yang akan diinstal. Informasi yang akan diinstal akan di simpan sebagai informasi dan akan di simpan.

d) *Design*

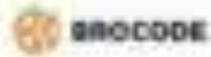
Pada proses ini akan melakukan *Design* yang akan diinstal dan akan diinstal dengan menggunakan bahasa yang ada untuk membuat dan mengkonstruksi serta desain yang akan digunakan. Program yang akan diinstal dengan tipe *software* menggunakan bahasa yang terdapat di brocode yang akan diinstal. Untuk itu juga akan membuat program yang akan diinstal. Informasi yang akan diinstal akan di simpan sebagai informasi dan akan di simpan.

e) *Design*

Pada proses ini akan melakukan *Design* yang akan diinstal dan akan diinstal dengan menggunakan bahasa yang ada untuk membuat dan mengkonstruksi serta desain yang akan digunakan. Program yang akan diinstal dengan tipe *software* menggunakan bahasa yang terdapat di brocode yang akan diinstal. Untuk itu juga akan membuat program yang akan diinstal. Informasi yang akan diinstal akan di simpan sebagai informasi dan akan di simpan.

II. *Aspek*

No	Uraian	Evaluasi hasil		
		Nilai	Nilai	Nilai



		1	2	3	4	5	6	7	8	9	10	11	12
1	Requirements Analysis												
	a. Requirements	■											
	b. Use Case		■										
2	Design												
	a. System & Software Design			■	■								
3	Development												
	a. Implementation				■	■	■	■	■				
	b. Unit Testing				■	■	■	■	■				
4	Testing												
	a. UAT User									■	■		
5	Deployment												
	a. Deployment to System												■

1. Budget Data

id	Item	Week	Plan	Usage	Var/Est
1	Project Management				
	Project Manager	1	175,000	175,000	0,000,000
	Project Team Member	1	40,000	40,000	
2	System Analyst				
	System	11	20,000	1,000,000	0,000,000
	UAT	1	40,000	40,000	
	Security Review after UAT	1	15,000	100,000	
3	System	11	20,000	400,000	0,000,000
4	Programmer				
	Development Software	9	15,000	90,000	
	Design	1	200,000	200,000	0,000,000
	Code	10	200,000	1,000,000	
	Deployment	1	10,000	10,000	
	Support	1	10,000	10,000	

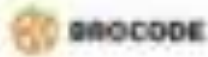
3. Program				
Propaganda dan Iklan	17	20.000	200.000	900.000
Penjualan Perseorangan				
Kontribusi	2	200.000	200.000	
4. Keuntungan Bersih				
Keuntungan	3	400.000	400.000	200.000
Keuntungan Bersih	1	200.000	200.000	200.000
A. Penghasilan				
Gaji	2	400.000	400.000	200.000
Distribusi	1	200.000	200.000	
Total				2.000.000

F. Tim Pengabdian

1. Yuzya Devi Anggraeni - Project Manager
2. Yully Adhira Putri - LPIS
3. SGG Diah Nurrahmah - Instruktur
4. Diani Hestianingsih - Koordinator dan Bookkeeper
5. Hana Nur Hafidza - Manajer Operasional
6. Rizki Nur Hafidza - Program Manager
7. Alifia Nur Hafidza - Program Manager

G. Foto Anggota





B Contact Person

Saluran AMI dapat dihubungi melalui:

Phone	: 0851 05000000
Web/Email	: www.amicode.com / info@amicode.com
Contact	: Gedung AMI Road 104/105
Email	: am@2001022001@amicode.com
Phone WP	: 0893 0751121

A Prinsip

Analisis proposal pribadi di bawah ini berdasarkan informasi yang sangat singkat tentang di lapangan. Semua konsep dalam proposal ini adalah hasil dari diskusi dengan pemilik di tingkat lokal, dan semua pertimbangan yang diperlukan untuk menilai kembali bahwa semua konsep tersebut mungkin. Semua konsep dalam dokumen proposal ini adalah hasil dari diskusi yang pribadi dan tidak dapat dipertanggungjawabkan secara hukum.

ii. Surat Kontrak/MoU Kerja MPTI dengan Mitra/Klien

SURAT PERJANJIAN KERJASAMA

Pada hari Rabu, tanggal 07, bulan Juni, tahun Dua Ribu Dua Puluh Tiga (07 Juni 2023) bertempat di tempat telah ditandatangani perjanjian kerjasama antara :

Nama : Fajar Anugrah
Jabatan : Petilik
No KTP : 32731510088600110
Alamat : Perum Mega Residence Blok A No.10

Bertindak atas nama pribadi, selanjutnya disebut sebagai **PIHAK PERTAMA.**

Nama : Vistya Dewi Anjani
Jabatan : Project Manager
No KTP : 2101175564021001
Alamat : Jl. Ringroad Selatan, Daerah Istimewa Yogyakarta

Bertindak atas nama pribadi, selanjutnya disebut sebagai **PIHAK KEDUA.**

Prosedur Pelaksanaan

1. PIHAK KEDUA akan melakukan pembuatan aplikasi komersial berbasis mobile untuk PIHAK PERTAMA sebagai dukungan dalam menjalankan usahanya.
2. PIHAK PERTAMA akan membayar harga jasa pembuatan sistem aplikasi mobile termasuk dengan proses upload ke Google Playstore kepada PIHAK KEDUA jika sistem aplikasi komersial berbasis mobile menjadi keputusan bersama.
3. PIHAK PERTAMA akan membayar kepada PIHAK KEDUA atau LANS DEVELOPMENT yang dibuktikan/ditunjukkan sebelum sistem yang telah dibuat.

Dengan terpa mengormati ketentuan yang telah disepakati dan atas dasar ketentuan hukum yang telah berlaku di wilayah Negara Kesatuan Republik Indonesia. Kedua belah pihak sepakat untuk membuat sebuah Perjanjian Kerjasama dengan ketentuan yang diatur dalam pasal-pasal berikut ini:

PASAL 1

MAKSUD DAN TUJUAN

Maksud dan tujuan perjanjian kerjasama ini adalah PARA PIHAK sepakat untuk melakukan kerjasama untuk pembuatan aplikasi komersial berbasis mobile.

PASAL 2

OBJEK PERJANJIAN

Objek Perjanjian kerjasama ini adalah pembuatan aplikasi komersial berbasis mobile untuk dikembangkan dengan website perusahaan yang telah ada sesuai dengan permintaan PIHAK PERTAMA.

PASAL 3

RUANG LINGKUP

Ruang lingkup perjanjian kerjasama ini adalah:

1. PIHAK PERTAMA sepakat bahwa dalam kerjasama ini menyediakan biaya sebesar Rp8.145.000,00 (delapan juta seratus empat puluh lima ribu rupiah).
2. Dua PIHAK KEDUA akan mengoptimalkan konsepnya dan pengalamannya.

PASAL 4

HAK DAN KEWAJIBAN PARA PIHAK

1. PIHAK PERTAMA berkewajiban:
 - a. Menyediakan dana sebesar Rp8.145.000,00 (delapan juta seratus empat puluh lima ribu rupiah) untuk pembuatan aplikasi komersial berbasis mobile.
 - b. Menyerahkan kewenangan kepada pihak kedua untuk mengatur pembayaran produksi dan pembuatan aplikasi komersial berbasis mobile.
2. PIHAK PERTAMA berhak :
 - a. Menuntut kepada PIHAK KEDUA untuk melakukan perbaikan perbaikan aplikasi komersial berbasis mobile selama 3 bulan secara gratis dan PIHAK PERTAMA akan dikenakan biaya administrasi setelah melewati batas waktu tersebut sesuai dengan

perjanjian yang telah disepakati, waktu mulai terhitungnya jumlah perbaikan adalah setelah aplikasi komersial selesai dibuat.

- b. Mendapatkan pedoman/tenor pembuatan aplikasi company profile berbasis mobile dan pelatihan kepada PIHAK KEDUA.
3. PIHAK KEDUA berkewajiban :
 - a. Membuat aplikasi komersial berbasis mobile yang akan diberikan kepada PIHAK PERTAMA sesuai dengan kesepakatan.
 - b. Melakukan pengujian terhadap aplikasi komersial sebelum diberikan kepada PIHAK PERTAMA.
 - c. PIHAK KEDUA wajib melaporkan perkembangan aplikasi secara berkala kepada PIHAK PERTAMA agar PIHAK PERTAMA bisa mengetahui apabila ada revisi pada produk atau tampilannya.
4. PIHAK KEDUA berhak :
 - a. Mendapat kesempatan terkait biaya operasional pembuatan aplikasi komersial berbasis mobile sebesar **Rp8.145.000,00** (delapan juta seratus empat puluh lima ribu rupiah).
 - b. Mendapatkan feedback dari PIHAK PERTAMA mengenai aplikasi komersial berbasis mobile.

Demikian surat ini dibuat dan disepakati kedua belah pihak.

Pihak Pertama,



Fajri Nugraha

Pihak Kedua,



(Viatya Devi Arjani)

iii. Log Book sudah terisi minimal 7x



PROYECTO DE LEY
DE

<p>Artículo 1.º El presente proyecto de ley tiene por objeto...</p>	<p>Artículo 2.º El presente proyecto de ley tiene por objeto...</p>
--	--

N.º	Descripción	Código	Unidad	Cantidad	Valor

N.º	Descripción	Código	Unidad	Cantidad	Valor

El presente proyecto de ley fue aprobado en el seno de la Comisión de Asesoramiento Jurídico el día...

El presente proyecto de ley fue aprobado en el seno de la Comisión de Asesoramiento Jurídico el día...

El presente proyecto de ley fue aprobado en el seno de la Comisión de Asesoramiento Jurídico el día...

El presente proyecto de ley fue aprobado en el seno de la Comisión de Asesoramiento Jurídico el día...

El presente proyecto de ley fue aprobado en el seno de la Comisión de Asesoramiento Jurídico el día...

iv. Foto Dokumentasi Kegiatan MPTI

Rapat #1 14 Maret 2023



Rapat #2 24 Maret 2023



Rapat #3 31 Maret 2023



Rapat #4 08 April 2023



Rapat bersama klien 8 Mei 2023



Rapat #5 8 Mei 2023



Rapat #6 16 Mei 2023



Rapat #7 29 Mei 2023



Rapat #8 07 Juni 2023



Rapat #9 18 Juli 2023



v. **Bukti Pembiayaan: pemasukan, pengeluaran, pendapatan/keuntungan & konsultan proyek**

Invoice

 BROCODE <small>PT. BROCODE INDONESIA</small>		INVOICE	
Bank: Bank Rakyat Indonesia Rekening: 014-0100000000000000 Alamat: Jl. Raya Cempaka No. 100, Jakarta Barat		Project: PT. Cahaya Raya Indonesia Tanggal Invoice: 14 Juni 2023	
DESKRIPSI	KUANTITAS	HARGA SATUAN	TOTAL
Manajemen Proyek	1 Paket	Rp. 200.000	Rp. 200.000
Keahlian Teknis			
- Manajemen	10 Paket	Rp. 100.000	Rp. 1.000.000
- IT	5 Paket	Rp. 80.000	Rp. 400.000
- Analisis Proses Bisnis	7 Paket	Rp. 15.000	Rp. 105.000
Keahlian Lainnya	10 Paket	Rp. 10.000	Rp. 100.000
Keahlian Lainnya			
- Keahlian Lainnya	5 Paket	Rp. 10.000	Rp. 50.000
- Keahlian Lainnya	3 Paket	Rp. 100.000	Rp. 300.000
- Keahlian Lainnya	2 Paket	Rp. 110.000	Rp. 220.000
Keahlian Lainnya	1 Paket	Rp. 400.000	Rp. 400.000
Keahlian Lainnya			
- Keahlian Lainnya	1 Paket	Rp. 100.000	Rp. 100.000
- Keahlian Lainnya	1 Paket	Rp. 100.000	Rp. 100.000
TOTAL HARGA			Rp. 4.000.000

Pembayaran dari klien (pemasukan)



Pembayaran ke Project Manager



Pembayaran ke Business Analyst



The image shows a payment slip from PT. JAYA BANGSA with the title "Transfer Berhasil". The slip lists the following details:

Item	Value
Transfer	10.000.000,00
Saldo Awal	10.000.000,00
Saldo Akhir	0,00
Saldo Transfer	10.000.000,00
Saldo Kas	0,00
Saldo Piutang	0,00
Saldo Hutang	0,00
Saldo Lain-lain	0,00
Saldo Total	0,00
Saldo Akhir	0,00
Saldo Awal	10.000.000,00
Saldo Akhir	0,00
Total	10.000.000,00

Pembayaran ke UI/UX



The image shows a payment slip from PT. JAYA BANGSA with the title "Transfer Berhasil". The slip lists the following details:

Item	Value
Transfer	10.000.000,00
Saldo Awal	10.000.000,00
Saldo Akhir	0,00
Saldo Transfer	10.000.000,00
Saldo Kas	0,00
Saldo Piutang	0,00
Saldo Hutang	0,00
Saldo Lain-lain	0,00
Saldo Total	0,00
Saldo Akhir	0,00
Saldo Awal	10.000.000,00
Saldo Akhir	0,00
Total	10.000.000,00

Pembayaran ke Programmer



Upload ke Google PlayStore





Kuota



Date	Detail	Qty
20/08/2024	Beras	10 kg
20/08/2024	Gula	5 kg
20/08/2024	Telur	10 butir
20/08/2024	Susu	1 liter
20/08/2024	Minyak	2 liter
20/08/2024	Garam	1 kg
20/08/2024	Gula Pasir	1 kg

Tambah Data

Konsumsi

No	Uraian	Jumlah	Satuan
1	Beras	10	kg
2	Gula	5	kg
3	Telur	10	butir
4	Susu	1	liter
5	Minyak	2	liter
6	Garam	1	kg
7	Gula Pasir	1	kg

Total: 100.000

vi. Wawancara Klien

TEKS WAWANCARA KLIEN PT. CAKRA AGRO INDONESIA

Tanggal/Waktu : Senin, 8 Mei 2023, Pukul 14.00 WIB

Narasumber : Bapak Fajar Anugrah

Jabatan : Pemilik PT. Cakra Agro Indonesia

Platform : Google Meet

No.	Pertanyaan	Jawaban Klien
1.	Assalamualaikum Warahmatullahi Wabarakatuh, dan Selamat Siang Pak Fajar dan Mas Taufik. Terimakasih sebelumnya sudah menghadiri Meet pada siang ini, dan sudah mengizinkan kelompok kami untuk melakukan wawancara project kepada Bapak.	Walaikumsalam Warahmatullahi Wabarakatuh. Iya sama-sama Mas dan Mbak.
2.	Sebelumnya izinkan kami bertanya pak, apakah PT. Cakra Agro Indonesia milik Bapak sudah memiliki website perusahaan dalam mengenalkan produk yang dimiliki PT Bapak kepada masyarakat, Pak?	Untuk Website sendiri kami sudah memiliki ya Mas Mbak. Mas dan Mbak bisa akses link https://cakraagroin.com/
3.	Baik Pak sebagai pembaruan pada Website Perusahaan PT Bapak, bagaimana tanggapan Bapak dengan rencana pembuatan Website Perusahaan PT. Cakra Agro Indonesia berbasis Mobile, Pak?	Saya kira itu suatu pembaharuan yang baik, dan akan lebih membantu dalam menyebarkan informasi kepada masyarakat karena berbasis Mobile mudah dijangkau masyarakat.
4.	Sebelumnya, kendala apa saja yang terjadi saat menggunakan website perusahaan produk PT. Cakra Agro Indonesia berbasis website ini ya Pak?	Untuk kendalanya sendiri, lebih kepada jangkauannya ya Mas Mbak. Saya melihat dengan membandingkan website sekarang akan lebih baik jika dibuatkan suatu Aplikasi Komersial berbasis Mobile yang bisa di download dan diinstal di Google Play. Selain untuk memudahkan jangkauan masyarakat lebih cepat dan praktis, juga dapat membantu dalam proses penjualan produk kami Mas Mbak. Sehingga masyarakat lebih mengetahui mengenai produk-produk pupuk yang kami miliki

		dan bisa melihat Profile Perusahaan kami ini.
5.	Untuk fitur-fitur di dalam Aplikasi Mobile nya sendiri apakah ada perubahan dari website yang sudah ada sebelumnya Pak?	Saya ingin ada penambahan dalam Aplikasi ini Mas Mbak agar menyempurnakan dari Website perusahaan yang sudah ada sebelumnya.
6.	Penambahan apa yang ingin ditambahkan dalam Aplikasi Komersial Mobile ini pak?	Saya ingin adanya Fitur Login, Logout, kemudian adanya Fitur QNA untuk Customer yang ingin bertanya terkait produk.
7.	Baik Pak, bagaimana tanggapan Bapak dengan Aplikasi Mobile dengan adanya fitur Login sebagai Admin di dalamnya?	Saya rasa itu hal yang bagus dilakukan Mas Mbak, agar mempermudah kami dalam mengisikan isi data dalam Aplikasi Mobile ini.
8.	Apakah dalam fitur Login ini Bapak memiliki permintaan untuk siapa saja yang bisa memiliki akses login ini Pak?	Saya ingin dalam fitur Login ini untuk Login Admin, dan Staff dalam Perusahaan Mas Mbak.
9.	Untuk Staff Perusahaan sendiri ada berapa bagian ya Pak yang diinginkan untuk bisa Login di Aplikasi ini?	Di dalam Perusahaan akan ada 5 bagian Staff Perusahaan, yaitu Manager, Secretary, Supervisor, Distributor dan Demonstrator.
10.	Untuk Struktur Perusahaannya bagaimana ya Pak tingkatan bagian Staff Perusahaan yang dimiliki PT. Cakra Agro Indonesia ini?	Struktur organisasi paling atas disini adalah Manajer, yang memiliki wewenang pada Secretary, Supervisor. Kemudian, Secretary memiliki wewenang atas Distributor. Sedangkan, Supervisor memiliki wewenang atas Demonstrator.
11.	Untuk sistem informasi yang akan diberikan, bagaimana pendapat Bapak dengan Aplikasi sistem 4 arah, yaitu Fitur Super Admin, Fitur untuk Customer, Fitur Admin dan Fitur Staf Perusahaan?	Saya rasa untuk hal bagus Mas Mbak.
12.	Dalam Aplikasi Komersial Mobile fitur Staf ini apakah bapak memiliki permintaan pada Staff dalam pengelolaan data di Aplikasi ini Pak?	Saya ingin terdapat fitur baru untuk tambah data, baca, edit data, hapus data dan Staff bisa upload File dan Foto. Pada Manajer dapat wewenang akses dalam pengelolaan Membaca Report Secretary dan Supervisor, membaca Bonus Distributor, membaca Aktivitas Supervisor, serta melakukan Tambah

		data, Baca, Edit dan Hapus data pada Menu Budgeting. Selanjutnya, untuk fitur Secretary terdapat wewenang akses membaca Report Distributor dan dapat Tambah data, Baca, Edit dan Hapus data pada Menu Bonus Distributor. Untuk Fitur Supervisor yaitu terdapat wewenang akses Membaca Report Demonstrator dan dapat Tambah data, Baca, Edit dan Hapus data pada Menu Aktivitas. Untuk Fitur Distributor yaitu dapat wewenang akses Membaca Bonus Distributor. Dan untuk Fitur Demonstrator dapat wewenang akses Tambah data, Baca, Edit dan Hapus data Report Demplot yang sudah dilakukan.
13.	Bagaimana tanggapan Bapak dengan Fitur Admin memiliki wewenang akses mengelola Tambah data, Baca, Edit dan Hapus data Tabel Staff; Tambah, Baca, Edit dan Hapus data Tabel Produk; Tambah, Baca, Edit dan Hapus data Tabel Event (Aktivitas dan Demplot); Tambah, Baca, Edit dan Hapus data Tabel Inovasi (Produk Baru dan Paket Baru); Tambah, Baca, Edit dan Hapus data Tabel Berita; Tambah, Baca, Edit dan Hapus data Tabel Tentang Kami; Tambah, Baca, Edit dan Hapus data Tabel QNA; dan Tambah, Baca, Edit dan Hapus data Tabel Beranda?	Saya rasa itu hal yang bagus Mas Mbak.
14.	Baik Pak terimakasih tanggapannya. Kemudian, bagaimana tanggapan Bapak dengan Fitur Super Admin memiliki wewenang akses dapat mengelola Tambah, Baca, Edit dan Hapus data Tabel Admin, dan dapat akses Tambah, Baca, Edit dan Hapus data Tabel LOG?	Saya rasa itu hal yang bagus juga Mas Mbak.
15.	Baik Pak terimakasih tanggapannya. Bagaimana tanggapan bapak dengan adanya	Saya rasa untuk Register Akun akan dilakukan sepenuhnya oleh Admin dan Super Admin, jadi proses daftar nama

	Register Akun untuk Staff Perusahaan yang akan Login Aplikasi ini?	Staff Perusahaan akan diisikan oleh Admin. Staff Perusahaan akan memberikan Nomor Telepon, Nama dan Bekerja pada bagian mana pada Admin, kemudian Admin akan mendaftarkannya dan Staff Perusahaan hanya menerima hasil Register Akun berupa Username dan Passwordnya dari Admin.
16.	Baik Pak terimakasih tanggapannya. Kemudian, bagaimana tanggapan Bapak dengan Register Akun Customer Pak?	Saya rasa Customer tidak perlu melakukan Login Mas Mbak, sehingga Aplikasi akan langsung terbuka ke Halaman Home tanpa perlu Login terlebih dahulu.
17.	Baik Pak. Selanjutnya, bagaimana proses yang diinginkan dalam Fitur baru QnA ini Pak?	Saya ingin dalam Fitur QnA ini Customer hanya perlu menjelaskan identitasnya seperti Nama, Alamat, dan Nomor telepon jika ingin menanyakan sesuatu pada Perusahaan melalui QnA ini, dan Admin dapat menjawab pertanyaan yang ditanyakan oleh Customer. Customer juga dapat melakukan Upload gambar dalam pertanyaan jika diperlukan.
18.	Baik Pak. Bagaimana tanggapan Bapak dengan Fitur Customer terdapat akses Membaca semua bagian tabel; serta akses Tambah, dan Baca pada bagian QnA?	Iyaa bisa seperti itu Mas Mbak.
19.	Baik Pak terimakasih. Selanjutnya, apakah dari Bapak dan Mas ada saran dan permintaan lain mengenai pembuatan Aplikasi Komersial berbasis Mobile ini pak?	Saya ingin fitur yang sudah ada di dalam Website Perusahaan yang lama dipindahkan ke dalam Aplikasi Mobile ini tanpa mengubah data-datanya. Kemudian, ditambahkan dengan Fitur Login, Fitur Super Admin, Fitur Admin, dan Fitur Staff-Staff Perusahaan, serta penambahan fitur QnA untuk Customer seperti tadi.
20.	Baik Pak terima kasih untuk semua saran dan pendapatnya, dan terimakasih juga sudah bersedia untuk melakukan wawancara dengan kelompok kami.	Baik Mas Mbak terima kasih kembali.

Dokumentasi



vii. Tools: source code, hosting, pemegang password, user manual

1. Activity

a) Aktivitas

1) Kelola Aktivitas

```
package com.example.cakraagroindonesia.Activity.Aktivitas;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Adapter.DataAktivitas;
import com.example.cakraagroindonesia.Interface.InterfaceAktivitas;
import com.example.cakraagroindonesia.Model.ModelAktivitas;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;
```

```

public class KelolaAktivitas extends AppCompatActivity {
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;

    TextView btnTambah;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_daftar_aktivitas);

        btnTambah = findViewById(R.id.btntambahaktivitas);
        btnTambah.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaAktivitas.this,
TambahAktivitas.class));
            }
        });
    }

    @Override
    protected void onResume() {
        super.onResume();
        recyclerView = findViewById(R.id.rv_kelolaaktivitas);
        layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL, false);
        recyclerView.setLayoutManager(layoutManager);

        InterfaceAktivitas interfaceAktivitas =
RetroServer.KonesiAPI().create(InterfaceAktivitas.class);
        Call<ModelAktivitas> tampil =
interfaceAktivitas.getAktivitas();
        tampil.enqueue(new Callback<ModelAktivitas>() {
            @Override
            public void onResponse(Call<ModelAktivitas> call,
Response<ModelAktivitas> response) {
                ArrayList<ModelAktivitas.aktivitas> listData =
response.body().getAktivitas();
                adapter = new DataAktivitas(KelolaAktivitas.this,
listData);
                recyclerView.setAdapter(adapter);
                adapter.notifyDataSetChanged();
            }
        });
    }
}

```

```
        @Override
        public void onFailure(Call<ModelAktivitas> call,
    Throwable t) {
            }
        });
    }
}
```

SourceCode 1. 1 Kelola Aktivitas

2) Tambah Aktivitas

Nn

```
package com.example.cakraagroindonesia.Activity.Aktivitas;

import android.Manifest;
import android.app.DatePickerDialog;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Interface.InterfaceAktivitas;
import com.example.cakraagroindonesia.Interface.InterfaceSupervisor;
import com.example.cakraagroindonesia.Model.ModelAktivitas;
import com.example.cakraagroindonesia.Model.ModelSupervisor;
import com.example.cakraagroindonesia.R;

import java.io.File;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.List;
import java.util.Locale;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
```

```

import retrofit2.Callback;
import retrofit2.Response;

public class TambahAktivitas extends AppCompatActivity {

    private TextView btnPilihFoto,btnsubmit;
    private EditText tanggalaktivitas, namakegiatan,statusaktivitas,
    namakios, produkaktivitas, jumlahpartisipan, budget, lokasi,
    areasales, provinsi, statusapps;
    private Calendar calendar;

    private String TanggalAktivitas, NamaKegiatan, StatusAktivitas,
    NamaKios, Produktivitas, JumlahPartisipan, Budget, Lokasi,
    Areasales, Provinsi, StatusApps;
    private String namaSupervisorTerpilih, kodeSupervisorTerpilih,
    kodeManagerTerpilih, namaManagerTerpilih;

    private Spinner spinner;
    private ArrayAdapter spinnerAdapter;

    private ImageView foto;
    private Uri selectedImageUri;

    private static final int REQUEST_CODE_PICK_IMAGE = 101;
    private static final int PERMISSION_REQUEST_CODE = 102;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_daftar_aktivitas_baru);

        calendar = Calendar.getInstance();

        spinner = findViewById(R.id.spinner);
        tanggalaktivitas = findViewById(R.id.tanggalaktivitas);
        btnPilihFoto = findViewById(R.id.btnPilihFoto);
        foto = findViewById(R.id.foto);
        namakegiatan = findViewById(R.id.namakegiatan);
        namakios = findViewById(R.id.namakios);
        produkaktivitas = findViewById(R.id.produkaktivitas);
        jumlahpartisipan = findViewById(R.id.jumlahpartisipan);
        budget = findViewById(R.id.budget);
        lokasi = findViewById(R.id.lokasi);
        areasales = findViewById(R.id.areasales);
        provinsi = findViewById(R.id.provinsi);
        btnsubmit = findViewById(R.id.btnsubmit);
        statusaktivitas = findViewById(R.id.statusaktivitas);

```

```

        statusapps = findViewById(R.id.statusapps);

        tanggalaktivitas.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                showDatePickerDialog();
            }
        });

        InterfaceSupervisor interfaceSupervisor =
RetroServer.KonesiAPI().create(InterfaceSupervisor.class);
        Call<ModelSupervisor> Spinner =
interfaceSupervisor.getSupervisor();
        Spinner.enqueue(new Callback<ModelSupervisor>() {
            @Override
            public void onResponse(Call<ModelSupervisor> call,
Response<ModelSupervisor> response) {
                ModelSupervisor modelSupervisor = response.body();
                ArrayList<ModelSupervisor.supervisor> supervisorList
= modelSupervisor.getSupervisor();

                List<String> namaSupervisor = new ArrayList<>();
                for (ModelSupervisor.supervisor nama :
supervisorList){
                    namaSupervisor.add(nama.getNama_supervisor());
                }
                List<String> kodeSupervisor = new ArrayList<>();
                for (ModelSupervisor.supervisor kode :
supervisorList){
                    kodeSupervisor.add(kode.getKode_sv());
                }
                List<String> kodeManager = new ArrayList<>();
                for (ModelSupervisor.supervisor kodeMG :
supervisorList){
                    kodeManager.add(kodeMG.getKode_mg());
                }
                List<String> namaManager = new ArrayList<>();
                for (ModelSupervisor.supervisor namaMG :
supervisorList){
                    namaManager.add(namaMG.getNama_manager());
                }

                spinnerAdapter = new
ArrayAdapter<>(TambahAktivitas.this,
android.R.layout.simple_spinner_item, namaSupervisor);

```

```

        spinnerAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spinner.setAdapter(spinnerAdapter);

        spinner.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?>
adapterView, View view, int i, long l) {
                namaSupervisorTerpilih =
namaSupervisor.get(i);
                kodeSupervisorTerpilih =
kodeSupervisor.get(i);
                kodeManagerTerpilih = kodeManager.get(i);
                namaManagerTerpilih = namaManager.get(i);
            }
            @Override
            public void onNothingSelected(AdapterView<?>
adapterView) {
            }
        });
        @Override
        public void onFailure(Call<ModelSupervisor> call,
Throwable t) {
        }
    });

    Log.d("MyTag", "onCreate:
"+kodeManagerTerpilih+namaManagerTerpilih+kodeSupervisorTerpilih+namaManagerTerpilih);

    btnPilihFoto.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            checkAndRequestPermission();
        }
    });
    btnsubmit.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

            TanggalAktivitas =
tanggalaktivitas.getText().toString().trim();
            NamaKegiatan =
namakegiatan.getText().toString().trim();

```

```

        StatusAktivitas =
statusaktivitas.getText().toString().trim();
        NamaKios = namakios.getText().toString().trim();
        Produktivitas =
produkaktivitas.getText().toString().trim();
        JumlahPartisipan =
jumlahpartisipan.getText().toString().trim();
        Budget = budget.getText().toString().trim();
        Lokasi = lokasi.getText().toString().trim();
        Areasales = areasales.getText().toString().trim();
        Provinsi = provinsi.getText().toString().trim();
        StatusApps = statusapps.getText().toString().trim();

        if(!TanggalAktivitas.isEmpty() &&
!NamaKegiatan.isEmpty() && !StatusAktivitas.isEmpty() &&
!NamaKios.isEmpty() && !Produktivitas.isEmpty() &&
!JumlahPartisipan.isEmpty() && !Budget.isEmpty() &&
!Lokasi.isEmpty() && !Provinsi.isEmpty() && !StatusApps.isEmpty()){
            if (selectedImageUri != null){
                String imagePath =
getRealPathFromURI(selectedImageUri);

                tambahData(imagePath, TanggalAktivitas,
NamaKegiatan, NamaKegiatan, NamaKios, Produktivitas,
JumlahPartisipan, Budget, Lokasi, Areasales, Provinsi, StatusApps);
            }else{
                Toast.makeText(TambahAktivitas.this, "Pilih
gambar terlebih dahulu", Toast.LENGTH_SHORT).show();
            }
        } else {
            Toast.makeText(TambahAktivitas.this, "Semua
field harus diisi", Toast.LENGTH_SHORT).show();
        }
    }
});
}

private void showDatePickerDialog() {
    DatePickerDialog datePickerDialog = new DatePickerDialog(
TambahAktivitas.this,
(view, year, month, dayOfMonth) -> {
        calendar.set(year, month, dayOfMonth);
        String myFormat = "yyyy-MM-dd"; // Format
tanggal yang diinginkan

```

```

        SimpleDateFormat sdf = new
SimpleDateFormat(myFormat, Locale.getDefault());
        tanggalaktivitas.setText(sdf.format(calendar.get
Time()));
    },
    calendar.get(Calendar.YEAR),
    calendar.get(Calendar.MONTH),
    calendar.get(Calendar.DAY_OF_MONTH)
);
//
// // Batasi tanggal yang bisa dipilih (opsional)
// datePickerDialog.getDatePicker().setMaxDate(System.current
TimeMillis()); // Hanya tanggal hari ini atau sebelumnya
// //
datePickerDialog.getDatePicker().setMinDate(System.currentTimeMillis
()); // Hanya tanggal hari ini atau setelahnya
//
    datePickerDialog.show();
}

private void checkAndRequestPermission() {
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
        openGallery();
    } else {
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
PERMISSION_REQUEST_CODE);
    }
}

private void openGallery() {
    Intent intent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
    startActivityForResult(intent, REQUEST_CODE_PICK_IMAGE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_CODE_PICK_IMAGE && resultCode ==
RESULT_OK && data != null) {
        selectedImageUri = data.getData();
        foto.setImageURI(selectedImageUri);
    }
}

@Override

```

```

    public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        if (requestCode == PERMISSION_REQUEST_CODE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                openGallery();
            } else {
                Toast.makeText(this, "Izin akses galeri dibutuhkan
untuk memilih foto", Toast.LENGTH_SHORT).show();
            }
        }
    }

    private String getRealPathFromURI(Uri contentUri) {
        String[] projection = {MediaStore.Images.Media.DATA};
        Cursor cursor = getContentResolver().query(contentUri,
projection, null, null, null);
        if (cursor == null) {
            return contentUri.getPath();
        } else {
            int columnIndex =
cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
            cursor.moveToFirst();
            String filePath = cursor.getString(columnIndex);
            cursor.close();
            return filePath;
        }
    }

    private void tambahData(String imagePath, String
STanggalAktivitas, String SNamaKegiatan, String SStatusAktivitas,
String SNamaKios, String SProduktivitas, String SJumlahPartisipan,
String SBudget, String SLokasi, String SAreasales, String SProvinsi,
String SStatusApps) {
        File imageFile = new File(imagePath);
        RequestBody imageRequestBody =
RequestBody.create(MediaType.parse("image/*"), imageFile);
        MultipartBody.Part Foto =
MultipartBody.Part.createFormData("foto_kegiatan",
imageFile.getName(), imageRequestBody);
        RequestBody Tgl =
RequestBody.create(MediaType.parse("text/plain"),
STanggalAktivitas);
        RequestBody Namakegiatan =
RequestBody.create(MediaType.parse("text/plain"), SNamaKegiatan);

```

```

        RequestBody Status =
RequestBody.create(MediaType.parse("text/plain"), SStatusAktivitas);
        RequestBody Namakios =
RequestBody.create(MediaType.parse("text/plain"), SNamaKios);
        RequestBody Produktivitas =
RequestBody.create(MediaType.parse("text/plain"), SProduktivitas);
        RequestBody Jumlahpartisipan =
RequestBody.create(MediaType.parse("text/plain"),
SJumlahPartisipan);
        RequestBody Budget =
RequestBody.create(MediaType.parse("text/plain"), SBudget);
        RequestBody Lokasi =
RequestBody.create(MediaType.parse("text/plain"), SLokasi);
        RequestBody Areasales =
RequestBody.create(MediaType.parse("text/plain"), SAreasales);
        RequestBody Provinsi =
RequestBody.create(MediaType.parse("text/plain"), SProvinsi);
        RequestBody kodeMG =
RequestBody.create(MediaType.parse("text/plain"),
kodeManagerTerpilih);
        RequestBody namaMG =
RequestBody.create(MediaType.parse("text/plain"),
namaManagerTerpilih);
        RequestBody kodeSV =
RequestBody.create(MediaType.parse("text/plain"),
kodeSupervisorTerpilih);
        RequestBody namaSV =
RequestBody.create(MediaType.parse("text/plain"),
namaSupervisorTerpilih);
        RequestBody Statusapps =
RequestBody.create(MediaType.parse("text/plain"), SStatusApps);

        Log.d("MyTag", "onResponse: " + imagePath +
STanggalAktivitas + SNamaKegiatan + SNamaKios + SProduktivitas +
SJumlahPartisipan + SBudget + SLokasi + SAreasales + SProvinsi +
kodeManagerTerpilih + namaManagerTerpilih + kodeSupervisorTerpilih +
namaSupervisorTerpilih);

        InterfaceAktivitas interfaceAktivitas =
RetroServer.KonesiAPI().create(InterfaceAktivitas.class);
        Call<ModelAktivitas> simpan =
interfaceAktivitas.setAktivitas(Foto,namaSV,namaMG,kodeMG,Areasales,
Provinsi,Namakegiatan,Tgl,Jumlahpartisipan,Produktivitas,Namakios,Bu
dget,kodeSV,Lokasi,Status,Statusapps);
        simpan.enqueue(new Callback<ModelAktivitas>() {
            @Override

```



```

import com.example.cakraagroindonesia.Interface.InterfaceAktivitas;
import com.example.cakraagroindonesia.Interface.InterfaceSupervisor;
import com.example.cakraagroindonesia.Model.ModelAktivitas;
import com.example.cakraagroindonesia.Model.ModelSupervisor;
import com.example.cakraagroindonesia.R;

import java.io.File;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.List;
import java.util.Locale;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class UbahAktivitas extends AppCompatActivity {

    private String varId, varTgl, varStatus, varNamaKegiatan,
varNamaKios, varProduk, varJumlahPartisipan, varBudget, varLokasi,
varAreaSales, varProvinsi, varStatusApps, varTFoto, varFoto;

    private TextView btnPilihFoto,btnsubmit;
    private EditText tanggalaktivitas, namakegiatan,statusaktivitas,
namakios, produkaktivitas, jumlahpartisipan, budget, lokasi,
areasales, provinsi, statusapps;
    private Calendar calendar;

    private String TanggalAktivitas, NamaKegiatan, StatusAktivitas,
NamaKios, Produktivitas, JumlahPartisipan, Budget, Lokasi,
Areasales, Provinsi, StatusApps;
    private String namaSupervisorTerpilih, kodeSupervisorTerpilih,
kodeManagerTerpilih, namaManagerTerpilih;

    private Spinner spinner;
    private ArrayAdapter spinnerAdapter;

    private ImageView foto;
    private Uri selectedImageUri;

    private static final int REQUEST_CODE_PICK_IMAGE = 101;
    private static final int PERMISSION_REQUEST_CODE = 102;

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_daftar_aktivitas_ubah);

    spinner = findViewById(R.id.spinner);
    tanggalaktivitas = findViewById(R.id.tanggalaktivitas);
    calendar = Calendar.getInstance();
    btnPilihFoto = findViewById(R.id.btnPilihFoto);
    foto = findViewById(R.id.foto);
    namakegiatan = findViewById(R.id.namakegiatan);
    namakios = findViewById(R.id.namakios);
    produkaktivitas = findViewById(R.id.produkaktivitas);
    jumlahpartisipan = findViewById(R.id.jumlahpartisipan);
    budget = findViewById(R.id.budget);
    lokasi = findViewById(R.id.lokasi);
    areasales = findViewById(R.id.areasales);
    provinsi = findViewById(R.id.provinsi);
    btnsubmit = findViewById(R.id.btnsubmit);
    statusaktivitas = findViewById(R.id.statusaktivitas);
    statusapps = findViewById(R.id.statusapps);

    varId = getIntent().getStringExtra("xId");
    varTgl = getIntent().getStringExtra("xTgl");
    varStatus = getIntent().getStringExtra("xStatus");
    varNamaKegiatan =
getIntent().getStringExtra("xNamaKegiatan");
    varNamaKios = getIntent().getStringExtra("xNamaKios");
    varProduk = getIntent().getStringExtra("xProduk");
    varJumlahPartisipan =
getIntent().getStringExtra("xJumlahPartisipan");
    varBudget = getIntent().getStringExtra("xBudget");
    varLokasi = getIntent().getStringExtra("xLokasi");
    varAreaSales = getIntent().getStringExtra("xAreaSales");
    varProvinsi = getIntent().getStringExtra("xProvinsi");
    varStatusApps = getIntent().getStringExtra("xStatusApps");
    varTFoto = getIntent().getStringExtra("xTFoto");
    varFoto = getIntent().getStringExtra("xFoto");

    tanggalaktivitas.setText(varTgl);
    statusaktivitas.setText(varStatus);
    namakegiatan.setText(varNamaKegiatan);
    namakios.setText(varNamaKios);
    produkaktivitas.setText(varProduk);
    jumlahpartisipan.setText(varJumlahPartisipan);
    budget.setText(varBudget);
    lokasi.setText(varLokasi);

```

```

        areasales.setText(varAreaSales);
        provinsi.setText(varProvinsi);
        statusapps.setText(varStatusApps);

        tanggalaktivitas.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                showDatePickerDialog();
            }
        });

        InterfaceSupervisor interfaceSupervisor =
RetroServer.KonesiAPI().create(InterfaceSupervisor.class);
        Call<ModelSupervisor> Spinner =
interfaceSupervisor.getSupervisor();
        Spinner.enqueue(new Callback<ModelSupervisor>() {
            @Override
            public void onResponse(Call<ModelSupervisor> call,
Response<ModelSupervisor> response) {
                ModelSupervisor modelSupervisor = response.body();
                ArrayList<ModelSupervisor.supervisor> supervisorList
= modelSupervisor.getSupervisor();

                List<String> namaSupervisor = new ArrayList<>();
                for (ModelSupervisor.supervisor nama :
supervisorList){
                    namaSupervisor.add(nama.getNama_supervisor());
                }
                List<String> kodeSupervisor = new ArrayList<>();
                for (ModelSupervisor.supervisor kode :
supervisorList){
                    kodeSupervisor.add(kode.getKode_sv());
                }
                List<String> kodeManager = new ArrayList<>();
                for (ModelSupervisor.supervisor kodeMG :
supervisorList){
                    kodeManager.add(kodeMG.getKode_mg());
                }
                List<String> namaManager = new ArrayList<>();
                for (ModelSupervisor.supervisor namaMG :
supervisorList){
                    namaManager.add(namaMG.getNama_manager());
                }
            }
        });
    }
}

```

```

        spinnerAdapter = new
ArrayAdapter<>(UbahAktivitas.this,
android.R.layout.simple_spinner_item, namaSupervisor);
        spinnerAdapter.setDropDownViewResource(android.R.lay
out.simple_spinner_dropdown_item);
        spinner.setAdapter(spinnerAdapter);

        spinner.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?>
adapterView, View view, int i, long l) {
                namaSupervisorTerpilih =
namaSupervisor.get(i);
                kodeSupervisorTerpilih =
kodeSupervisor.get(i);
                kodeManagerTerpilih = kodeManager.get(i);
                namaManagerTerpilih = namaManager.get(i);
            }
            @Override
            public void onNothingSelected(AdapterView<?>
adapterView) {
            }
        });
        @Override
        public void onFailure(Call<ModelSupervisor> call,
Throwable t) {
        }
    });

    Log.d("MyTag", "onCreate:
"+kodeManagerTerpilih+namaManagerTerpilih+kodeSupervisorTerpilih+nam
aManagerTerpilih);

    btnPilihFoto.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            checkAndRequestPermission();
        }
    });
    btnsubmit.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

```

```

        TanggalAktivitas =
tanggalaktivitas.getText().toString().trim();
        NamaKegiatan =
namakegiatan.getText().toString().trim();
        StatusAktivitas =
statusaktivitas.getText().toString().trim();
        NamaKios = namakios.getText().toString().trim();
        Produktivitas =
produkaktivitas.getText().toString().trim();
        JumlahPartisipan =
jumlahpartisipan.getText().toString().trim();
        Budget = budget.getText().toString().trim();
        Lokasi = lokasi.getText().toString().trim();
        Areasales = areasales.getText().toString().trim();
        Provinsi = provinsi.getText().toString().trim();
        StatusApps = statusapps.getText().toString().trim();

        if(!TanggalAktivitas.isEmpty() &&
!NamaKegiatan.isEmpty() && !StatusAktivitas.isEmpty() &&
!NamaKios.isEmpty() && !Produktivitas.isEmpty() &&
!JumlahPartisipan.isEmpty() && !Budget.isEmpty() &&
!Lokasi.isEmpty() && !Provinsi.isEmpty() && !StatusApps.isEmpty()){
            if (selectedImageUri != null){
                String imagePath =
getRealPathFromURI(selectedImageUri);
                varFoto =
getRealPathFromURI(selectedImageUri);
                varTFoto =
getRealPathFromURI(selectedImageUri);

                tambahData(imagePath, TanggalAktivitas,
NamaKegiatan, NamaKegiatan, NamaKios, Produktivitas,
JumlahPartisipan, Budget, Lokasi, Areasales, Provinsi, StatusApps);
            }else{
                Toast.makeText(UbahAktivitas.this, "Pilih
gambar terlebih dahulu", Toast.LENGTH_SHORT).show();
            }
        } else {
            Toast.makeText(UbahAktivitas.this, "Semua field
harus diisi", Toast.LENGTH_SHORT).show();
        }
    }
});
}
}

```

```

private void showDatePickerDialog() {
    DatePickerDialog datePickerDialog = new DatePickerDialog(
        UbahAktivitas.this,
        (view, year, month, dayOfMonth) -> {
            calendar.set(year, month, dayOfMonth);
            String myFormat = "yyyy-MM-dd"; // Format
tanggal yang diinginkan
            SimpleDateFormat sdf = new
SimpleDateFormat(myFormat, Locale.getDefault());
            tanggalaktivitas.setText(sdf.format(calendar.get
Time()));
        },
        calendar.get(Calendar.YEAR),
        calendar.get(Calendar.MONTH),
        calendar.get(Calendar.DAY_OF_MONTH)
    );
    //
    // // Batasi tanggal yang bisa dipilih (opsional)
    // datePickerDialog.getDatePicker().setMaxDate(System.current
TimeMillis()); // Hanya tanggal hari ini atau sebelumnya
    // //
    datePickerDialog.getDatePicker().setMinDate(System.currentTimeMillis
()); // Hanya tanggal hari ini atau setelahnya
    //
    datePickerDialog.show();
}

private void checkAndRequestPermission() {
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
        openGallery();
    } else {
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
PERMISSION_REQUEST_CODE);
    }
}

private void openGallery() {
    Intent intent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
    startActivityForResult(intent, REQUEST_CODE_PICK_IMAGE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
}

```

```

        if (requestCode == REQUEST_CODE_PICK_IMAGE && resultCode ==
RESULT_OK && data != null) {
            selectedImageUri = data.getData();
            foto.setImageURI(selectedImageUri);
        }
    }
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        if (requestCode == PERMISSION_REQUEST_CODE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                openGallery();
            } else {
                Toast.makeText(this, "Izin akses galeri dibutuhkan
untuk memilih foto", Toast.LENGTH_SHORT).show();
            }
        }
    }
    private String getRealPathFromURI(Uri contentUri) {
        String[] projection = {MediaStore.Images.Media.DATA};
        Cursor cursor = getContentResolver().query(contentUri,
projection, null, null, null);
        if (cursor == null) {
            return contentUri.getPath();
        } else {
            int columnIndex =
cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
            cursor.moveToFirst();
            String filePath = cursor.getString(columnIndex);
            cursor.close();
            return filePath;
        }
    }

    private void tambahData(String imagePath,String
STanggalAktivitas,String SNamaKegiatan,String SStatusAktivitas,
String SNamaKios, String SProduktivitas, String SJumlahPartisipan,
String SBudget, String SLokasi, String SAreasales, String SProvinsi,
String SStatusApps) {
        File imageFile = new File(imagePath);
        RequestBody imageRequestBody =
RequestBody.create(MediaType.parse("image/*"), imageFile);

```



```

        MultipartBody.Part Foto =
MultipartBody.Part.createFormData("foto_kegiatan",
imageFile.getName(), imageRequestBody);
        RequestBody Tgl =
RequestBody.create(MediaType.parse("text/plain"),
STanggalAktivitas);
        RequestBody Namakegiatan =
RequestBody.create(MediaType.parse("text/plain"), SNamaKegiatan);
        RequestBody Status =
RequestBody.create(MediaType.parse("text/plain"), SStatusAktivitas);
        RequestBody Namakios =
RequestBody.create(MediaType.parse("text/plain"), SNamaKios);
        RequestBody Produktivitas =
RequestBody.create(MediaType.parse("text/plain"), SProduktivitas);
        RequestBody Jumlahpartisipan =
RequestBody.create(MediaType.parse("text/plain"),
SJumlahPartisipan);
        RequestBody Budget =
RequestBody.create(MediaType.parse("text/plain"), SBudget);
        RequestBody Lokasi =
RequestBody.create(MediaType.parse("text/plain"), SLokasi);
        RequestBody Areasales =
RequestBody.create(MediaType.parse("text/plain"), SAreasales);
        RequestBody Provinsi =
RequestBody.create(MediaType.parse("text/plain"), SProvinsi);
        RequestBody kodeMG =
RequestBody.create(MediaType.parse("text/plain"),
kodeManagerTerpilih);
        RequestBody namaMG =
RequestBody.create(MediaType.parse("text/plain"),
namaManagerTerpilih);
        RequestBody kodeSV =
RequestBody.create(MediaType.parse("text/plain"),
kodeSupervisorTerpilih);
        RequestBody namaSV =
RequestBody.create(MediaType.parse("text/plain"),
namaSupervisorTerpilih);
        RequestBody Statusapps =
RequestBody.create(MediaType.parse("text/plain"), SStatusApps);

        Log.d("MyTag", "onResponse: " + imagePath +
STanggalAktivitas + SNamaKegiatan + SNamaKios + SProduktivitas +
SJumlahPartisipan + SBudget + SLokasi + SAreasales + SProvinsi +
kodeManagerTerpilih + namaManagerTerpilih + kodeSupervisorTerpilih +
namaSupervisorTerpilih);

```

```

        InterfaceAktivitas interfaceAktivitas =
RetroServer.KonesiAPI().create(InterfaceAktivitas.class);
        Call<ModelAktivitas> simpan =
interfaceAktivitas.setAktivitas(Foto,namaSV,namaMG,kodeMG,Areasales,
Provinsi,Namakegiatan,Tgl,Jumlahpartisipan,Produktivitas,Namakios,Bu
dget,kodeSV,Lokasi,Status,Statusapps);
        simpan.enqueue(new Callback<ModelAktivitas>() {
            @Override
            public void onResponse(Call<ModelAktivitas> call,
Response<ModelAktivitas> response) {
                Toast.makeText(UbahAktivitas.this, "Data Berhasil
Disimpan", Toast.LENGTH_SHORT).show();
                finish();
            }
            @Override
            public void onFailure(Call<ModelAktivitas> call,
Throwable t) {
                Toast.makeText(UbahAktivitas.this, "Data Gagal
Disimpan", Toast.LENGTH_SHORT).show();
            }
        });
    }
}

```

SourceCode 1. 3 Ubah Aktivitas

b) Beranda

1) Homepage

```

package com.example.cakraagroindonesia.Activity.Beranda;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;

```

```

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Interface.InterfaceBeranda;
import com.example.cakraagroindonesia.Model.ModelBeranda;
import com.example.cakraagroindonesia.R;

import java.io.File;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class Homepage extends AppCompatActivity {
    private EditText judulBeranda, deskripsiBeranda;
    private TextView btnUbah, btnPilihFoto;
    private String judul, deskripsi;

    private ImageView foto;
    private Uri selectedImageUri;

    private static final int REQUEST_CODE_PICK_IMAGE = 101;
    private static final int PERMISSION_REQUEST_CODE = 102;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_homepage);

        judulBeranda = findViewById(R.id.judulberanda);
        deskripsiBeranda = findViewById(R.id.deksripsiberanda);
        btnUbah = findViewById(R.id.bnubah);
        foto = findViewById(R.id.foto);
        btnPilihFoto = findViewById(R.id.btnPilihFoto);

        InterfaceBeranda interfaceBeranda =
RetroServer.KonesiAPI().create(InterfaceBeranda.class);
        Call<ModelBeranda> call = interfaceBeranda.getBeranda(3);
        call.enqueue(new Callback<ModelBeranda>() {
            @Override

```

```

        public void onResponse(Call<ModelBeranda> call,
Response<ModelBeranda> response) {
            ModelBeranda modelBeranda = response.body();
            judulBeranda.setText(modelBeranda.getJudul());
            deskripsiBeranda.setText(modelBeranda.getDeskripsi(
));
            Glide.with(Hompage.this).load(modelBeranda.getFoto(
)).into(foto);

            selectedImageUri =
Uri.parse(modelBeranda.getFoto());

        }

        @Override
        public void onFailure(Call<ModelBeranda> call, Throwable
t) {

        }

    });

    btnPilihFoto.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            checkAndRequestPermission();
        }
    });

    btnUbah.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            judul = judulBeranda.getText().toString().trim();
            deskripsi =
deskripsiBeranda.getText().toString().trim();

            if (!judul.isEmpty() && !deskripsi.isEmpty()){
                if (selectedImageUri != null){
                    String imagePath =
getRealPathFromURI(selectedImageUri);
                    uploadImage(imagePath, judul, deskripsi);
                }else {
                    Toast.makeText(Hompage.this, "Pilih Foto
Terlebih dahulu", Toast.LENGTH_SHORT).show();
                }
            }
        }
    }
}

```

```

    });
}
private void checkAndRequestPermission() {
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
        openGallery();
    } else {
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
PERMISSION_REQUEST_CODE);
    }
}

private void openGallery() {
    Intent intent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
    startActivityForResult(intent, REQUEST_CODE_PICK_IMAGE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_CODE_PICK_IMAGE && resultCode ==
RESULT_OK && data != null) {
        selectedImageUri = data.getData();
        foto.setImageURI(selectedImageUri);
    }
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
    if (requestCode == PERMISSION_REQUEST_CODE) {
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            openGallery();
        } else {
            Toast.makeText(this, "Izin akses galeri dibutuhkan
untuk memilih foto", Toast.LENGTH_SHORT).show();
        }
    }
}
}

```

```

private String getRealPathFromURI(Uri contentUri) {
    String[] projection = {MediaStore.Images.Media.DATA};
    Cursor cursor = getContentResolver().query(contentUri,
projection, null, null, null);
    if (cursor == null) {
        return contentUri.getPath();
    } else {
        int columnIndex =
cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
        cursor.moveToFirst();
        String filePath = cursor.getString(columnIndex);
        cursor.close();
        return filePath;
    }
}

private void uploadImage(String imagePath, String Judul, String
Deskripsi) {
    File imageFile = new File(imagePath);

    RequestBody imageRequestBody =
RequestBody.create(MediaType.parse("image/*"), imageFile);

    MultipartBody.Part Foto =
MultipartBody.Part.createFormData("foto", imageFile.getName(),
imageRequestBody);
    RequestBody judul =
RequestBody.create(MediaType.parse("text/plain"), Judul);
    RequestBody deskripsi =
RequestBody.create(MediaType.parse("text/plain"), Deskripsi);

    Log.d("MyTag", "uploadImage: "+imageFile+Judul+Deskripsi);
    InterfaceBeranda interfaceBeranda =
RetroServer.KonesiAPI().create(InterfaceBeranda.class);
    Call<ModelBeranda> call = interfaceBeranda.setBeranda(3,
Foto, judul, deskripsi);
    call.enqueue(new Callback<ModelBeranda>() {
        @Override
        public void onResponse(Call<ModelBeranda> call,
Response<ModelBeranda> response) {
            Toast.makeText(Homepage.this, "Data Berhasil Disimpan
", Toast.LENGTH_SHORT).show();
            finish();
        }
        @Override
        public void onFailure(Call<ModelBeranda> call, Throwable
t) {

```

```

        Toast.makeText(Homepage.this, "Data Gagal Disimpan",
Toast.LENGTH_SHORT).show();
    }
    });
}
}
}

```

SourceCode 1. 4 Homepage

2) Kelola Beranda

```

package com.example.cakraagroindonesia.Activity.Beranda;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.R;

public class KelolaBeranda extends AppCompatActivity {
    TextView btnHomepage, btnProdukHomepage, btnBerita;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_kelola_beranda);

        btnHomepage = findViewById(R.id.btnHomepage);
        btnProdukHomepage = findViewById(R.id.btnProdukHomepage);
        btnBerita = findViewById(R.id.btnBerita);

        btnHomepage.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaBeranda.this,
Homepage.class));
            }
        });
        btnProdukHomepage.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaBeranda.this,
KelolaProdukHomepage.class));
            }
        });
    }
}

```

```

    }
    });
    btnBerita.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            startActivity(new Intent(KelolaBeranda.this,
KelolaBerita.class));
        }
    });
}
}
}

```

SourceCode 1. 5 Kelola Beranda

3) Kelola Berita

```

package com.example.cakraagroindonesia.Activity.Beranda;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetrofitServer;
import com.example.cakraagroindonesia.Adapter.DataKelolaBerita;
import com.example.cakraagroindonesia.Interface.InterfaceBerita;
import com.example.cakraagroindonesia.Model.ModelBerita;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;
import java.util.Collections;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class KelolaBerita extends AppCompatActivity {
    TextView btnTambahBerita;
    private RecyclerView rvBerita;
    private RecyclerView.Adapter adBerita;
    private RecyclerView.LayoutManager lmBerita;
    @Override
    protected void onCreate(Bundle savedInstanceState) {

```



```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_berita);

btnTambahBerita = findViewById(R.id.btntambah);

btnTambahBerita.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startActivity(new Intent(KelolaBerita.this,
TambahBerita.class));
    }
});

//TAMPIL BERITA
//Deklarasi
rvBerita = findViewById(R.id.rv_kelolaberita);
lmBerita = new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL,false);
rvBerita.setLayoutManager(lmBerita);

}

@Override
protected void onResume() {
    super.onResume();
    InterfaceBerita interfaceBerita =
RetroServer.KonesiAPI().create(InterfaceBerita.class);
    Call<ModelBerita> tampilBerita =
interfaceBerita.getBerita();
    tampilBerita.enqueue(new Callback<ModelBerita>() {
        @Override
        public void onResponse(Call<ModelBerita> call,
Response<ModelBerita> response) {
            ArrayList<ModelBerita.data_berita> listBerita=
response.body().getData_berita();
            adBerita = new
DataKelolaBerita(KelolaBerita.this,listBerita);
            Collections.reverse(listBerita);
            rvBerita.setAdapter(adBerita);
            adBerita.notifyDataSetChanged();
        }
    });

    @Override
    public void onFailure(Call<ModelBerita> call, Throwable
t) {}
});

```

```
}  
}
```

SourceCode 1. 6 Kelola Berita

4) Kelola Produk Homepage

```
package com.example.cakraagroindonesia.Activity.Beranda;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.TextView;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.example.cakraagroindonesia.API.RetroServer;  
import  
com.example.cakraagroindonesia.Adapter.DataKelolaProdukHomepage;  
import  
com.example.cakraagroindonesia.Interface.InterfaceProdukHomepage;  
import com.example.cakraagroindonesia.Model.ModelProdukHomepage;  
import com.example.cakraagroindonesia.R;  
  
import java.util.ArrayList;  
import java.util.Collections;  
  
import retrofit2.Call;  
import retrofit2.Callback;  
import retrofit2.Response;  
  
public class KelolaProdukHomepage extends AppCompatActivity {  
    private RecyclerView recyclerView;  
    private RecyclerView.Adapter adapter;  
    private RecyclerView.LayoutManager layoutManager;  
  
    TextView btnTambah;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_produk_homepage);  
  
        btnTambah = findViewById(R.id.btntambah);  
        btnTambah.setOnClickListener(new View.OnClickListener() {
```

```

        @Override
        public void onClick(View view) {
            startActivity(new Intent(KelolaProdukHomepage.this,
TambahProdukHomepage.class));
        }
    });
}

@Override
protected void onResume() {
    super.onResume();
    recyclerView = findViewById(R.id.rv_kelolaprodukhomepage);
    layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL,false);
    recyclerView.setLayoutManager(layoutManager);

    InterfaceProdukHomepage interfaceProdukHomepage =
RetroServer.KonesiAPI().create(InterfaceProdukHomepage.class);
    Call<ModelProdukHomepage> tampil =
interfaceProdukHomepage.getProdukHomepage();
    tampil.enqueue(new Callback<ModelProdukHomepage>() {
        @Override
        public void onResponse(Call<ModelProdukHomepage> call,
Response<ModelProdukHomepage> response) {
            ArrayList<ModelProdukHomepage.produk_beranda>
listData = response.body().getProduk_beranda();
            adapter = new
DataKelolaProdukHomepage(KelolaProdukHomepage.this, listData);
            Collections.reverse(listData);
            recyclerView.setAdapter(adapter);
            adapter.notifyDataSetChanged();
        }
    });

    @Override
    public void onFailure(Call<ModelProdukHomepage> call,
Throwable t) {

    }
});
}
}
}

```

SourceCode 1. 7 Kelola Produk Homepage

5) Tambah Berita

```
package com.example.cakraagroindonesia.Activity.Beranda;

import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Interface.InterfaceBerita;
import com.example.cakraagroindonesia.Model.ModelBerita;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class TambahBerita extends AppCompatActivity {
    private TextView btnTambah;
    private EditText judulBerita, isiBerita;
    private String judul,isi;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tambah_berita);

        btnTambah = findViewById(R.id.btntambah);
        judulBerita = findViewById(R.id.judulberita);
        isiBerita = findViewById(R.id.isiberita);

        btnTambah.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                judul = judulBerita.getText().toString();
                isi = isiBerita.getText().toString();

                tambahData();
            }
        });
    }
}
```

```

private void tambahData(){
    ModelBerita.data_berita newData = new
ModelBerita.data_berita();
    newData.setJudul_berita(judul);
    newData.setIsi_berita(isi);

    ArrayList<ModelBerita.data_berita> listData = new
ArrayList<>();
    listData.add(newData);

    ModelBerita modelBerita = new ModelBerita();
    modelBerita.setData_berita(listData);

    InterfaceBerita interfaceBerita =
RetroServer.KonesiAPI().create(InterfaceBerita.class);
    Call<ModelBerita> simpan =
interfaceBerita.setBerita(judul,isi);
    simpan.enqueue(new Callback<ModelBerita>() {
        @Override
        public void onResponse(Call<ModelBerita> call,
Response<ModelBerita> response) {
            Toast.makeText(TambahBerita.this, "Berhasil Simpan
Data", Toast.LENGTH_SHORT).show();
            finish();
        }

        @Override
        public void onFailure(Call<ModelBerita> call, Throwable
t) {
            Toast.makeText(TambahBerita.this, "Gagal Simpan
Data", Toast.LENGTH_SHORT).show();
        }
    });
}
}
}

```

SourceCode 1. 8 Tambah Berita

6) Tambah Produk Homepage

```
package com.example.cakraagroindonesia.Activity.Beranda;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.API.RetroServer;
import
com.example.cakraagroindonesia.Interface.InterfaceProdukHomepage;
import com.example.cakraagroindonesia.Model.ModelProdukHomepage;
import com.example.cakraagroindonesia.R;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.IOException;
import java.io.InputStream;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class TambahProdukHomepage extends AppCompatActivity{

    private EditText merkproduk, deskripsiproduk;
    private TextView btnPilihFoto, btnsubmit;
```

```

private ImageView foto;

private String merk,deskripsi;

private Uri selectedImageUri;

private static final int REQUEST_CODE_PICK_IMAGE = 101;
private static final int PERMISSION_REQUEST_CODE = 102;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_tambah_produk_homepage);

    merkproduk = findViewById(R.id.merkproduk);
    deskripsiproduk = findViewById(R.id.deskripsiproduk);
    btnPilihFoto = findViewById(R.id.btnPilihFoto);
    btnsubmit = findViewById(R.id.btnsubmit);
    foto = findViewById(R.id.foto);

    btnPilihFoto.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            checkAndRequestPermission();
        }
    });

    btnsubmit.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            merk = merkproduk.getText().toString();
            deskripsi = deskripsiproduk.getText().toString();

            if (!merk.isEmpty() && !deskripsi.isEmpty()){
                File imagePath = new
File(getRealPathFromURI(selectedImageUri));

                tambahData(imagePath, merk,deskripsi);
            }else{
                Toast.makeText(TambahProdukHomepage.this, "Semua
field harus diisi", Toast.LENGTH_SHORT).show();
            }
        }
    });
}

```

```

private void checkAndRequestPermission() {
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
        openGallery();
    } else {
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
PERMISSION_REQUEST_CODE);
    }
}

private void openGallery() {
    Intent intent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
    intent.setType("image/png");
    startActivityForResult(intent, REQUEST_CODE_PICK_IMAGE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_CODE_PICK_IMAGE && resultCode ==
RESULT_OK && data != null) {
        selectedImageUri = data.getData();
        Glide.with(this).load(selectedImageUri).into(foto);
//        tampilfoto.setImageURI(selectedImageUri);
        if (selectedImageUri != null){
            Toast.makeText(this, "Berhasil Memilih Foto",
Toast.LENGTH_SHORT).show();
        }else{
            Toast.makeText(this, "Gagal Memilih Foto",
Toast.LENGTH_SHORT).show();
        }
    }
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
    if (requestCode == PERMISSION_REQUEST_CODE) {
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            openGallery();
        } else {

```



```

        Toast.makeText(this, "Izin akses galeri dibutuhkan
untuk memilih foto", Toast.LENGTH_SHORT).show();
    }
}

private String getRealPathFromURI(Uri contentUri) {
    String[] projection = {MediaStore.Images.Media.DATA};
    Cursor cursor = getContentResolver().query(contentUri,
projection, null, null, null);
    if (cursor == null) {
        return contentUri.getPath();
    } else {
        int columnIndex =
cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
        cursor.moveToFirst();
        String filePath = cursor.getString(columnIndex);
        cursor.close();
        return filePath;
    }
}

private byte[] getBytesFromInputStream(InputStream inputStream)
throws IOException {
    ByteArrayOutputStream byteBuffer = new
ByteArrayOutputStream();
    int bufferSize = 1024;
    byte[] buffer = new byte[bufferSize];
    int len;
    while ((len = inputStream.read(buffer)) != -1) {
        byteBuffer.write(buffer, 0, len);
    }
    return byteBuffer.toByteArray();
}

private void tambahData(File imgFile, String merk, String
deskripsi){
    try {
        InputStream inputStream =
getContentResolver().openInputStream(selectedImageUri);

        byte[] imageByte = getBytesFromInputStream(inputStream);

        RequestBody imageRequestBody =
RequestBody.create(MediaType.parse("image/png"), imageByte);

```


7) Ubah Berita

```
package com.example.cakraagroindonesia.Activity.Beranda;

import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Interface.InterfaceBerita;
import com.example.cakraagroindonesia.Model.ModelBerita;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class UbahBerita extends AppCompatActivity {

    private int varId;
    private String varJudul, varDeskripsi;

    private TextView btnTambah;
    private EditText judulBerita, isiBerita;
    private String judul,isi;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tambah_berita);

        btnTambah = findViewById(R.id.btntambah);
        judulBerita = findViewById(R.id.judulberita);
        isiBerita = findViewById(R.id.isiberita);

        varId = getIntent().getIntExtra("xId", -1);
        varJudul = getIntent().getStringExtra("xJudul");
        varDeskripsi = getIntent().getStringExtra("xDeskripsi");

        judulBerita.setText(varJudul);
        isiBerita.setText(varDeskripsi);
    }
}
```

```

        btnTambah.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                judul = judulBerita.getText().toString();
                isi = isiBerita.getText().toString();

                tambahData();
            }
        });
    }

    private void tambahData(){
        ModelBerita.data_berita newData = new
ModelBerita.data_berita();
        newData.setJudul_berita(judul);
        newData.setIsi_berita(isi);

        ArrayList<ModelBerita.data_berita> listData = new
ArrayList<>();
        listData.add(newData);

        ModelBerita modelBerita = new ModelBerita();
        modelBerita.setData_berita(listData);

        InterfaceBerita interfaceBerita =
RetroServer.KonesiAPI().create(InterfaceBerita.class);
        Call<ModelBerita> simpan =
interfaceBerita.updateBerita(varId,judul,isi);
        simpan.enqueue(new Callback<ModelBerita>() {
            @Override
            public void onResponse(Call<ModelBerita> call,
Response<ModelBerita> response) {
                Toast.makeText(UbahBerita.this, "Berhasil Simpan
Data", Toast.LENGTH_SHORT).show();
                finish();
            }

            @Override
            public void onFailure(Call<ModelBerita> call, Throwable
t) {
                Toast.makeText(UbahBerita.this, "Gagal Simpan Data",
Toast.LENGTH_SHORT).show();
            }
        });
    }
}
}

```

8) Ubah Produk Homepage

```
package com.example.cakraagroindonesia.Activity.Beranda;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.API.RetroServer;
import
com.example.cakraagroindonesia.Interface.InterfaceProdukHomepage;
import com.example.cakraagroindonesia.Model.ModelProdukHomepage;
import com.example.cakraagroindonesia.R;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.IOException;
import java.io.InputStream;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class UbahProdukHomepage extends AppCompatActivity{
```

```

private int varId;
private String varMerk, varDeskripsi;

private EditText merkproduk, deskripsiproduk;
private TextView btnPilihFoto, btnsubmit;
private ImageView foto;

private String merk,deskripsi;

private Uri selectedImageUri;

private static final int REQUEST_CODE_PICK_IMAGE = 101;
private static final int PERMISSION_REQUEST_CODE = 102;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_ubah_produk_homepage);

    merkproduk = findViewById(R.id.merkproduk);
    deskripsiproduk = findViewById(R.id.deskripsiproduk);
    btnPilihFoto = findViewById(R.id.btnPilihFoto);
    btnsubmit = findViewById(R.id.btnsubmit);
    foto = findViewById(R.id.foto);

    varId = getIntent().getIntExtra("xId", -1);
    varMerk = getIntent().getStringExtra("xMerk");
    varDeskripsi = getIntent().getStringExtra("xDeskripsi");

    merkproduk.setText(varMerk);
    deskripsiproduk.setText(varDeskripsi);

    btnPilihFoto.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            checkAndRequestPermission();
        }
    });

    btnsubmit.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            merk = merkproduk.getText().toString();
            deskripsi = deskripsiproduk.getText().toString();

            if (!merk.isEmpty() && !deskripsi.isEmpty()){

```

```

        File imagePath = new
File(getRealPathFromURI(selectedImageUri));

        tambahData(imagePath, merk, deskripsi);
    }else{
        Toast.makeText(UbahProdukHomepage.this, "Semua
field harus diisi", Toast.LENGTH_SHORT).show();
    }
}
});
}

private void checkAndRequestPermission() {
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
        openGallery();
    } else {
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
PERMISSION_REQUEST_CODE);
    }
}

private void openGallery() {
    Intent intent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
    intent.setType("image/png");
    startActivityForResult(intent, REQUEST_CODE_PICK_IMAGE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_CODE_PICK_IMAGE && resultCode ==
RESULT_OK && data != null) {
        selectedImageUri = data.getData();
        Glide.with(this).load(selectedImageUri).into(foto);
//        tampilfoto.setImageURI(selectedImageUri);
        if (selectedImageUri != null){
            Toast.makeText(this, "Berhasil Memilih Foto",
Toast.LENGTH_SHORT).show();
        }else{
            Toast.makeText(this, "Gagal Memilih Foto",
Toast.LENGTH_SHORT).show();
        }
    }
}

```

```

    }
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
    if (requestCode == PERMISSION_REQUEST_CODE) {
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            openGallery();
        } else {
            Toast.makeText(this, "Izin akses galeri dibutuhkan
untuk memilih foto", Toast.LENGTH_SHORT).show();
        }
    }
}

private String getRealPathFromURI(Uri contentUri) {
    String[] projection = {MediaStore.Images.Media.DATA};
    Cursor cursor = getContentResolver().query(contentUri,
projection, null, null, null);
    if (cursor == null) {
        return contentUri.getPath();
    } else {
        int columnIndex =
cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
        cursor.moveToFirst();
        String filePath = cursor.getString(columnIndex);
        cursor.close();
        return filePath;
    }
}

private byte[] getBytesFromInputStream(InputStream inputStream)
throws IOException {
    ByteArrayOutputStream byteBuffer = new
ByteArrayOutputStream();
    int bufferSize = 1024;
    byte[] buffer = new byte[bufferSize];
    int len;
    while ((len = inputStream.read(buffer)) != -1) {
        byteBuffer.write(buffer, 0, len);
    }
    return byteBuffer.toByteArray();
}
}

```



```

    private void tambahData(File imgFile, String merk, String
deskripsi){
        try {
            InputStream inputStream =
getContentResolver().openInputStream(selectedImageUri);

            byte[] imageByte = getBytesFromInputStream(inputStream);

            RequestBody imageRequestBody =
RequestBody.create(MediaType.parse("image/png"), imageByte);

            MultipartBody.Part Foto =
MultipartBody.Part.createFormData("foto",imgFile.getName(),
imageRequestBody);
            RequestBody Merk =
RequestBody.create(MediaType.parse("text/plain"), merk);
            RequestBody Deksripsi =
RequestBody.create(MediaType.parse("text/plain"), deskripsi);

            Log.d("MyTag", "tambahData INSEKTISIDA:
"+merk+deskripsi);

            InterfaceProdukHomepage interfaceProdukHomepage =
RetroServer.KonesiAPI().create(InterfaceProdukHomepage.class);
            Call<ModelProdukHomepage> tambah =
interfaceProdukHomepage.updateProdukHomepage(varId,Foto,Merk,Deksrip
si);

            tambah.enqueue(new Callback<ModelProdukHomepage>() {
                @Override
                public void onResponse(Call<ModelProdukHomepage>
call, Response<ModelProdukHomepage> response) {
                    Toast.makeText(UbahProdukHomepage.this,
"Berhasil Tambah Data", Toast.LENGTH_SHORT).show();
                    finish();
                }

                @Override
                public void onFailure(Call<ModelProdukHomepage>
call, Throwable t) {
                    Toast.makeText(UbahProdukHomepage.this, "Gagal
Tambah Data", Toast.LENGTH_SHORT).show();
                }
            });
        } catch (IOException e){
            e.printStackTrace();
        }
    }

```

```

        Toast.makeText(this, "Gagal Membaca Gambar",
Toast.LENGTH_SHORT).show();
    }
}
}

```

SourceCode 1. 11 Ubah Produk Homepage

c) Inovasi

1) Kelola Inovasi

```

package com.example.cakraagroindonesia.Activity.Inovasi;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.R;

public class KelolaInovasi extends AppCompatActivity {
    TextView btnProdukBaru, btnPaketProduk;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_kelola_inovasi);

        btnProdukBaru = findViewById(R.id.btnKelolaProdukBaru);
        btnPaketProduk = findViewById(R.id.btnKelolaPaketProduk);

        btnProdukBaru.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaInovasi.this,
KelolaProdukBaru.class));
            }
        });
        btnPaketProduk.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaInovasi.this,
KelolaPaketProduk.class));
            }
        });
    }
}

```

```
}  
});  
  
}  
}
```

SourceCode 1. 12 Kelola Inovasi

2) Kelola Paket Produk

```
package com.example.cakraagroindonesia.Activity.Inovasi;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.TextView;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.example.cakraagroindonesia.API.RetroServer;  
import com.example.cakraagroindonesia.Adapter.DataKelolaPaketProduk;  
import  
com.example.cakraagroindonesia.Interface.InterfacePaketProduk;  
import com.example.cakraagroindonesia.Model.ModelPaketProduk;  
import com.example.cakraagroindonesia.R;  
  
import java.util.ArrayList;  
  
import retrofit2.Call;  
import retrofit2.Callback;  
import retrofit2.Response;  
  
public class KelolaPaketProduk extends AppCompatActivity {  
    TextView btnTambah;  
    private RecyclerView recyclerView;  
    private RecyclerView.Adapter adapter;  
    private RecyclerView.LayoutManager layoutManager;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_kelola_paket_produk);  
  
        btnTambah = findViewById(R.id.btntambah);  
        btnTambah.setOnClickListener(new View.OnClickListener() {
```

```

        @Override
        public void onClick(View view) {
            startActivity(new Intent(KelolaPaketProduk.this,
TambahPaketProduk.class));
        }
    });
}

@Override
protected void onResume() {
    super.onResume();
    recyclerView = findViewById(R.id.rv_kelolapaketproduk);
    layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL,false);
    recyclerView.setLayoutManager(layoutManager);

    InterfacePaketProduk interfacePaketProduk =
RetroServer.KonesiAPI().create(InterfacePaketProduk.class);
    Call<ModelPaketProduk> tampil =
interfacePaketProduk.getPaketProduk();
    tampil.enqueue(new Callback<ModelPaketProduk>() {
        @Override
        public void onResponse(Call<ModelPaketProduk> call,
Response<ModelPaketProduk> response) {
            ArrayList<ModelPaketProduk.paket_produk> listData =
response.body().getPaket_produk();
            adapter = new
DataKelolaPaketProduk(KelolaPaketProduk.this, listData);
            recyclerView.setAdapter(adapter);
            adapter.notifyDataSetChanged();
        }
    });
    @Override
    public void onFailure(Call<ModelPaketProduk> call,
Throwable t) {}
});
}
}

```

SourceCode 1. 13 Kelola Paket Produk

3) Kelola Produk Baru

```

package com.example.cakraagroindonesia.Activity.Inovasi;

import android.content.Intent;
import android.os.Bundle;

```

```

import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Adapter.DataKelolaProdukBaru;
import com.example.cakraagroindonesia.Interface.InterfaceProdukBaru;
import com.example.cakraagroindonesia.Model.ModelProdukBaru;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class KelolaProdukBaru extends AppCompatActivity {
    TextView btnTambah;

    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_kelola_produk_baru);

        btnTambah = findViewById(R.id.btntambah);
        btnTambah.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaProdukBaru.this,
TambahProdukBaru.class));
            }
        });
    }

    @Override
    protected void onResume() {
        super.onResume();
        recyclerView = findViewById(R.id.rv_kelolaprodukbaru);
    }
}

```

```

        layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL, false);
        recyclerView.setLayoutManager(layoutManager);

        InterfaceProdukBaru interfaceProdukBaru =
RetroServer.KonesiAPI().create(InterfaceProdukBaru.class);
        Call<ModelProdukBaru> tampil =
interfaceProdukBaru.getProdukBaru();
        tampil.enqueue(new Callback<ModelProdukBaru>() {
            @Override
            public void onResponse(Call<ModelProdukBaru> call,
Response<ModelProdukBaru> response) {
                ArrayList<ModelProdukBaru.produk_baru> listData =
response.body().getProduk_baru();
                adapter = new
DataKelolaProdukBaru(KelolaProdukBaru.this, listData);
                recyclerView.setAdapter(adapter);
                adapter.notifyDataSetChanged();
            }

            @Override
            public void onFailure(Call<ModelProdukBaru> call,
Throwable t) {}
        });
    }
}

```

SourceCode 1. 14 Kelola Produk Baru

4) Tambah Paket Produk

```

package com.example.cakraagroindonesia.Activity.Inovasi;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.Display;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

```

```

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.example.cakraagroindonesia.API.RetroServer;
import
com.example.cakraagroindonesia.Interface.InterfacePaketProduk;
import com.example.cakraagroindonesia.Model.ModelPaketProduk;
import com.example.cakraagroindonesia.R;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.IOException;
import java.io.InputStream;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class TambahPaketProduk extends AppCompatActivity {
    EditText namapaket, tanaman, deskripsipaketproduk, hasil;
    TextView uploadppt, btntambah;
    String Ppt, Nama, Tanaman, Deskripsi, Hasil;

    private Uri pptUri;

    private static final int REQUEST_PICK_PPT_FILE = 1;
    private static final int PERMISSION_REQUEST_CODE = 102;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tambah_paket_produk);

        namapaket = findViewById(R.id.namapaket);
        tanaman = findViewById(R.id.tanaman);
        deskripsipaketproduk =
findViewById(R.id.deskripsipaketproduk);
        hasil = findViewById(R.id.hasil);
        uploadppt = findViewById(R.id.ppt);
        btntambah = findViewById(R.id.btntambah);
        TextView textView = findViewById(R.id.textView);

```

```

uploadppt.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        checkAndRequestPermission();
    }
});

btntambah.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Nama = namapaket.getText().toString();
        Tanaman = tanaman.getText().toString();
        Deskripsi =
deskripsipaketproduk.getText().toString();
        Hasil = hasil.getText().toString();

        if (!Nama.isEmpty() && !Tanaman.isEmpty() &&
!Deskripsi.isEmpty() && !Hasil.isEmpty()) {
            if (pptUri != null) {
                File ppt = new
File(getFilePathFromUri(pptUri));

                uploadData(ppt, Nama, Tanaman, Deskripsi,
Hasil);
            } else {
                Toast.makeText(TambahPaketProduk.this,
"Pilih PPT terlebih dahulu", Toast.LENGTH_SHORT).show();
            }
        } else {
            Toast.makeText(TambahPaketProduk.this, "Semua
field harus diisi", Toast.LENGTH_SHORT).show();
        }
    }
});
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == REQUEST_PICK_PPT_FILE && resultCode ==
RESULT_OK && data != null) {
        pptUri = data.getData();

        if (pptUri != null) {

```



```

        // Ubah URI menjadi File
        Toast.makeText(TambahPaketProduk.this, "Berhasil
Memilih PPT", Toast.LENGTH_SHORT).show();
    }else {
        Toast.makeText(TambahPaketProduk.this, "Gagal
Memilih PPT", Toast.LENGTH_SHORT).show();
    }
}

private void checkAndRequestPermission() {
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
        openExplorer();
    } else {
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
PERMISSION_REQUEST_CODE);
    }
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
    if (requestCode == PERMISSION_REQUEST_CODE) {
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            openExplorer();
        } else {
            Toast.makeText(this, "Izin akses dibutuhkan untuk
memilih file", Toast.LENGTH_SHORT).show();
        }
    }
}

private void openExplorer() {
    Intent intent = new Intent(Intent.ACTION_GET_CONTENT);
    intent.setType("application/*"); // Filter hanya file PPT
startActivityForResult(intent, REQUEST_PICK_PPT_FILE);
}

private String getPathFromUri(Uri contentUri) {
    String filePath;
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.Q) {
        String[] projection =
{MediaStore.Files.FileColumns.DISPLAY_NAME};
    }
}

```

```

        try (Cursor cursor =
getContentResolver().query(contentUri, projection, null, null,
null)) {
            if (cursor != null && cursor.moveToFirst()) {
                int columnIndex =
cursor.getColumnIndexOrThrow(MediaStore.Files.FileColumns.DISPLAY_NAME);
                filePath = cursor.getString(columnIndex);
            } else {
                filePath = contentUri.getPath();
            }
        }
    } else {
        filePath = contentUri.getPath();
    }
    Log.d("MyTag", "Selected URI: " + pptUri.toString());
    Log.d("MyTag", "getFileFromUri: " + filePath);
    return filePath;
}

private byte[] getBytesFromInputStream(InputStream inputStream)
throws IOException {
    ByteArrayOutputStream byteBuffer = new
ByteArrayOutputStream();
    int bufferSize = 1024;
    byte[] buffer = new byte[bufferSize];
    int len;
    while ((len = inputStream.read(buffer)) != -1) {
        byteBuffer.write(buffer, 0, len);
    }
    return byteBuffer.toByteArray();
}

private void uploadData(File pptFile, String Snama,String
Stanaman,String Sdeskripsi,String Shasil){

    try {
        InputStream inputStream =
getContentResolver().openInputStream(pptUri);
        // Membaca file menjadi byte array
        byte[] fileBytes = getBytesFromInputStream(inputStream);

        RequestBody requestBody =
RequestBody.create(MediaType.parse("application/"), fileBytes);

```

```

        MultipartBody.Part Ppt =
MultipartBody.Part.createFormData("ppt", pptFile.getName(),
requestFile);
        RequestBody tanaman =
RequestBody.create(MediaType.parse("text/plain"), Stanaman);
        RequestBody paket_produk =
RequestBody.create(MediaType.parse("text/plain"), Snama);
        RequestBody iterasi =
RequestBody.create(MediaType.parse("text/plain"), Sdeskripsi);
        RequestBody hasil =
RequestBody.create(MediaType.parse("text/plain"), Shasil);

        Log.d("MyTag", "uploadData:
"+fileBytes+Stanaman+Stanaman+ Sdeskripsi+Shasil);

        InterfacePaketProduk interfacePaketProduk =
RetroServer.KonesiAPI().create(InterfacePaketProduk.class);
        Call<ModelPaketProduk> tambah =
interfacePaketProduk.setPaketProduk(Ppt,tanaman,paket_produk,iterasi
,hasil);
        tambah.enqueue(new Callback<ModelPaketProduk>() {
            @Override
            public void onResponse(Call<ModelPaketProduk> call,
Response<ModelPaketProduk> response) {
                Toast.makeText(TambahPaketProduk.this, "Berhasil
Simpan Data", Toast.LENGTH_SHORT).show();
                finish();
            }

            @Override
            public void onFailure(Call<ModelPaketProduk> call,
Throwable t) {
                Toast.makeText(TambahPaketProduk.this, "Gagal
Simpan Data", Toast.LENGTH_SHORT).show();
            }
        });

        } catch (IOException e) {
            e.printStackTrace();
            Toast.makeText(TambahPaketProduk.this, "Gagal
membaca file", Toast.LENGTH_SHORT).show();
        }
    }
}

```

SourceCode 1. 15 Tambah Paket Produk

5) Tambah Produk Baru

```
package com.example.cakraagroindonesia.Activity.Inovasi;

import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Interface.InterfaceProdukBaru;
import com.example.cakraagroindonesia.Model.ModelProdukBaru;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class TambahProdukBaru extends AppCompatActivity {
    private TextView btntambah;
    private EditText namabahan, formulasi, tanaman;
    private String NamaBahan, Formulasi, Tanaman;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tambah_produk_baru);

        namabahan = findViewById(R.id.namabahan);
        formulasi = findViewById(R.id.formulasi);
        tanaman = findViewById(R.id.tanaman);
        btntambah = findViewById(R.id.btntambah);

        btntambah.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                NamaBahan = namabahan.getText().toString();
                Formulasi = formulasi.getText().toString();
                Tanaman = tanaman.getText().toString();

                tambahData();
            }
        });
    }
}
```

```

    }
    });
}

private void tambahData(){
    ModelProdukBaru.produk_baru newData = new
ModelProdukBaru.produk_baru();
    newData.setNama_bahan>NamaBahan);
    newData.setFormulasi(Formulasi);
    newData.setTanaman(Tanaman);

    ArrayList<ModelProdukBaru.produk_baru> listData = new
ArrayList<>();
    listData.add(newData);

    ModelProdukBaru modelProdukBaru = new ModelProdukBaru();
    modelProdukBaru.setProduk_baru(listData);

    Log.d("MyTag", "tambahData: "+NamaBahan+Formulasi+Tanaman);

    InterfaceProdukBaru interfaceProdukBaru =
RetroServer.KonesiAPI().create(InterfaceProdukBaru.class);
    Call<ModelProdukBaru> simpan =
interfaceProdukBaru.setProdukBaru>NamaBahan,Formulasi,Tanaman);
    simpan.enqueue(new Callback<ModelProdukBaru>() {
        @Override
        public void onResponse(Call<ModelProdukBaru> call,
Response<ModelProdukBaru> response) {
            Toast.makeText(TambahProdukBaru.this, "Berhasil
Simpan Data", Toast.LENGTH_SHORT).show();
            finish();
        }

        @Override
        public void onFailure(Call<ModelProdukBaru> call,
Throwable t) {
            Toast.makeText(TambahProdukBaru.this, "Gagal Simpan
Data", Toast.LENGTH_SHORT).show();
        }
    });
}
}
}

```

SourceCode 1. 16 Tambah Produk Baru

6) Ubah Paket Produk

```
package com.example.cakraagroindonesia.Activity.Inovasi;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.example.cakraagroindonesia.API.RetroServer;
import
com.example.cakraagroindonesia.Interface.InterfacePaketProduk;
import com.example.cakraagroindonesia.Model.ModelPaketProduk;
import com.example.cakraagroindonesia.R;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.IOException;
import java.io.InputStream;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class UbahPaketProduk extends AppCompatActivity {

    private String varNamaPaket, varDeskripsi, varTanaman, varHasil;
    private int varId;

    EditText namapaket, tanaman, deskripsipaketproduk, hasil;
```

```

TextView uploadppt, btntambah;
String Nama, Tanaman, Deskripsi, Hasil;

private Uri pptUri;

private static final int REQUEST_PICK_PPT_FILE = 1;
private static final int PERMISSION_REQUEST_CODE = 102;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_tambah_paket_produk);

    namapaket = findViewById(R.id.namapaket);
    tanaman = findViewById(R.id.tanaman);
    deskripsipaketproduk =
findViewById(R.id.deskripsipaketproduk);
    hasil = findViewById(R.id.hasil);
    uploadppt = findViewById(R.id.ppt);
    btntambah = findViewById(R.id.btntambah);

    varId = getIntent().getIntExtra("xId", -1);
    varNamaPaket = getIntent().getStringExtra("xNamaPaket");
    varTanaman = getIntent().getStringExtra("xDeskripsi");
    varDeskripsi = getIntent().getStringExtra("xDeskripsi");
    varHasil = getIntent().getStringExtra("xHasil");

    namapaket.setText(varNamaPaket);
    tanaman.setText(varTanaman);
    deskripsipaketproduk.setText(varDeskripsi);
    hasil.setText(varHasil);

    uploadppt.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            checkAndRequestPermission();
        }
    });

    btntambah.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Nama = namapaket.getText().toString();
            Tanaman = tanaman.getText().toString();
            Deskripsi =
deskripsipaketproduk.getText().toString();
            Hasil = hasil.getText().toString();

```

```

        if (!Nama.isEmpty() && !Tanaman.isEmpty() &&
!Deskripsi.isEmpty() && !Hasil.isEmpty()) {
            if (pptUri != null){
                File ppt = new
File(getFilePathFromUri(pptUri));

                uploadData(ppt, Nama, Tanaman, Deskripsi,
Hasil);
            } else {
                InterfacePaketProduk interfacePaketProduk =
RetroServer.KonesiAPI().create(InterfacePaketProduk.class);
                Call<ModelPaketProduk> ubah =
interfacePaketProduk.update(varId, Tanaman, Nama, Deskripsi, Hasil);
                ubah.enqueue(new
Callback<ModelPaketProduk>() {
                    @Override
                    public void
onResponse(Call<ModelPaketProduk> call, Response<ModelPaketProduk>
response) {
                        Toast.makeText(UbahPaketProduk.this,
"Berhasil Simpan Data", Toast.LENGTH_SHORT).show();
                        finish();
                    }
                    @Override
                    public void
onFailure(Call<ModelPaketProduk> call, Throwable t) {
                        Toast.makeText(UbahPaketProduk.this,
"Gagal Simpan Data", Toast.LENGTH_SHORT).show();
                    }
                });
            }
        } else {
            Toast.makeText(UbahPaketProduk.this, "Semua
field harus diisi", Toast.LENGTH_SHORT).show();
        }
    });
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == REQUEST_PICK_PPT_FILE && resultCode ==
RESULT_OK && data != null) {

```



```

        pptUri = data.getData();

        if (pptUri != null) {
            // Ubah URI menjadi File
            Toast.makeText(UbahPaketProduk.this, "Berhasil
Memilih PPT", Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(UbahPaketProduk.this, "Gagal Memilih
PPT", Toast.LENGTH_SHORT).show();
        }
    }
}

private void checkAndRequestPermission() {
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
        openExplorer();
    } else {
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
PERMISSION_REQUEST_CODE);
    }
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
    if (requestCode == PERMISSION_REQUEST_CODE) {
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            openExplorer();
        } else {
            Toast.makeText(this, "Izin akses dibutuhkan untuk
memilih file", Toast.LENGTH_SHORT).show();
        }
    }
}

private void openExplorer() {
    Intent intent = new Intent(Intent.ACTION_GET_CONTENT);
    intent.setType("application/*"); // Filter hanya file PPT
    startActivityForResult(intent, REQUEST_PICK_PPT_FILE);
}

private String getPathFromUri(Uri contentUri) {
    String filePath;
}

```

```

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.Q) {
            String[] projection =
{MediaStore.Files.FileColumns.DISPLAY_NAME};
            try (Cursor cursor =
getContentResolver().query(contentUri, projection, null, null,
null)) {
                if (cursor != null && cursor.moveToFirst()) {
                    int columnIndex =
cursor.getColumnIndexOrThrow(MediaStore.Files.FileColumns.DISPLAY_NA
ME);
                    filePath = cursor.getString(columnIndex);
                } else {
                    filePath = contentUri.getPath();
                }
            }
        } else {
            filePath = contentUri.getPath();
        }
        Log.d("MyTag", "Selected URI: " + pptUri.toString());
        Log.d("MyTag", "getFileFromUri: " + filePath);
        return filePath;
    }

    private byte[] getBytesFromInputStream(InputStream inputStream)
throws IOException {
        ByteArrayOutputStream byteBuffer = new
ByteArrayOutputStream();
        int bufferSize = 1024;
        byte[] buffer = new byte[bufferSize];
        int len;
        while ((len = inputStream.read(buffer)) != -1) {
            byteBuffer.write(buffer, 0, len);
        }
        return byteBuffer.toByteArray();
    }

    private void uploadData(File pptFile, String Snama,String
Stanaman,String Sdeskripsi,String Shasil){

        try {
            InputStream inputStream =
getContentResolver().openInputStream(pptUri);
            // Membaca file menjadi byte array
            byte[] fileBytes = getBytesFromInputStream(inputStream);

            RequestBody requestFile =
RequestBody.create(MediaType.parse("application/+"), fileBytes);

```

```

        MultipartBody.Part Ppt =
MultipartBody.Part.createFormData("ppt", pptFile.getName(),
requestFile);
        RequestBody tanaman =
RequestBody.create(MediaType.parse("text/plain"), Stanaman);
        RequestBody paket_produk =
RequestBody.create(MediaType.parse("text/plain"), Snama);
        RequestBody iterasi =
RequestBody.create(MediaType.parse("text/plain"), Sdeskripsi);
        RequestBody hasil =
RequestBody.create(MediaType.parse("text/plain"), Shasil);

        Log.d("MyTag", "uploadData:
"+fileBytes+Stanaman+Stanaman+ Sdeskripsi+Shasil);

        InterfacePaketProduk interfacePaketProduk =
RetroServer.KonesiAPI().create(InterfacePaketProduk.class);
        Call<ModelPaketProduk> tambah =
interfacePaketProduk.updatePaketProduk(varId,Ppt,tanaman,paket_produk,iterasi,hasil);
        tambah.enqueue(new Callback<ModelPaketProduk>() {
            @Override
            public void onResponse(Call<ModelPaketProduk> call,
Response<ModelPaketProduk> response) {
                Toast.makeText(UbahPaketProduk.this, "Berhasil
Simpan Data", Toast.LENGTH_SHORT).show();
                finish();
            }

            @Override
            public void onFailure(Call<ModelPaketProduk> call,
Throwable t) {
                Toast.makeText(UbahPaketProduk.this, "Gagal
Simpan Data", Toast.LENGTH_SHORT).show();
            }
        });

        } catch (IOException e) {
            e.printStackTrace();
            Toast.makeText(UbahPaketProduk.this, "Gagal membaca
file", Toast.LENGTH_SHORT).show();
        }
    }
}

```

SourceCode 1. 17 Ubah Paket Produk

7) Ubah Produk Baru

```
package com.example.cakraagroindonesia.Activity.Inovasi;

import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Interface.InterfaceProdukBaru;
import com.example.cakraagroindonesia.Model.ModelProdukBaru;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class UbahProdukBaru extends AppCompatActivity {
    private int varId;
    private String varNamaBahan, varFormulasi, varTanaman;

    private TextView btntambah;
    private EditText namabahan, formulasi, tanaman;
    private String NamaBahan, Formulasi, Tanaman;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ubah_produk_baru);

        namabahan = findViewById(R.id.namabahan);
        formulasi = findViewById(R.id.formulasi);
        tanaman = findViewById(R.id.tanaman);
        btntambah = findViewById(R.id.btntambah);

        varId = getIntent().getIntExtra("xId", -1);
        varNamaBahan = getIntent().getStringExtra("xNamaBahan");
        varFormulasi = getIntent().getStringExtra("xFormulasi");
        varTanaman = getIntent().getStringExtra("xTanaman");
    }
}
```

```

        namabahan.setText(varNamaBahan);
        formulasi.setText(varFormulasi);
        tanaman.setText(varTanaman);

        btntambah.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                NamaBahan = namabahan.getText().toString();
                Formulasi = formulasi.getText().toString();
                Tanaman = tanaman.getText().toString();

                ubahData();
            }
        });
    }

    private void ubahData(){
        ModelProdukBaru.produk_baru newData = new
ModelProdukBaru.produk_baru();
        newData.setNama_bahan>NamaBahan);
        newData.setFormulasi(Formulasi);
        newData.setTanaman(Tanaman);

        ArrayList<ModelProdukBaru.produk_baru> listData = new
ArrayList<>();
        listData.add(newData);

        ModelProdukBaru modelProdukBaru = new ModelProdukBaru();
        modelProdukBaru.setProduk_baru(listData);

        Log.d("MyTag", "tambahData: "+NamaBahan+Formulasi+Tanaman);

        InterfaceProdukBaru interfaceProdukBaru =
RetroServer.KonesiAPI().create(InterfaceProdukBaru.class);
        Call<ModelProdukBaru> simpan =
interfaceProdukBaru.updateProdukBaru(varId>NamaBahan,Formulasi,Tanam
an);
        simpan.enqueue(new Callback<ModelProdukBaru>() {
            @Override
            public void onResponse(Call<ModelProdukBaru> call,
Response<ModelProdukBaru> response) {
                Toast.makeText(UbahProdukBaru.this, "Berhasil Simpan
Data", Toast.LENGTH_SHORT).show();
                finish();
            }
        })
    }
}

```

```

        @Override
        public void onFailure(Call<ModelProdukBaru> call,
        Throwable t) {
            Toast.makeText(UbahProdukBaru.this, "Gagal Simpan
        Data", Toast.LENGTH_SHORT).show();
        }
    });
}
}
}

```

SourceCode 1. 18 Ubah Produk Baru

d) Kelola Data Master

1) Kelola Data Admin

```

package com.example.cakraagroindonesia.Activity.KelolaDataMaster;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Adapter.DataAdmin;
import com.example.cakraagroindonesia.Interface.InterfaceAdmin;
import com.example.cakraagroindonesia.Model.ModelAdmin;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class KelolaDataAdmin extends AppCompatActivity {
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;

    TextView btnTambah;
    @Override

```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_data_admin);

    btnTambah = findViewById(R.id.btntambah);
    btnTambah.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            startActivity(new Intent(KelolaDataAdmin.this,
TambahDataAdmin.class));
        }
    });
}

@Override
protected void onResume() {
    super.onResume();
    recyclerView = findViewById(R.id.rv_admin);
    layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL, false);
recyclerView.setLayoutManager(layoutManager);

    InterfaceAdmin interfaceAdmin =
RetroServer.KonesiAPI().create(InterfaceAdmin.class);
    Call<ModelAdmin> tampil = interfaceAdmin.getAdmin();
    tampil.enqueue(new Callback<ModelAdmin>() {
        @Override
        public void onResponse(Call<ModelAdmin> call,
Response<ModelAdmin> response) {
            ArrayList<ModelAdmin.data_admin> listData =
response.body().getData_admin();
            adapter = new DataAdmin(KelolaDataAdmin.this,
listData);
            recyclerView.setAdapter(adapter);
            adapter.notifyDataSetChanged();
        }

        @Override
        public void onFailure(Call<ModelAdmin> call, Throwable
t) {

        }
    });
}
}

```

SourceCode 1. 19 Kelola Data Admin

2) Kelola Data Demonstrator

```
package com.example.cakraagroindonesia.Activity.KelolaDataMaster;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Adapter.DataDemonstrator;
import
com.example.cakraagroindonesia.Interface.InterfaceDemonstrator;
import com.example.cakraagroindonesia.Model.ModelDemonstrator;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class KelolaDataDemonstrator extends AppCompatActivity {
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;

    TextView btnTambah;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_data_demonstrator);

        btnTambah = findViewById(R.id.btntambah);
        btnTambah.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new
Intent(KelolaDataDemonstrator.this, TambahDataDemonstrator.class));
            }
        });
    }
}
```



```

@Override
protected void onResume() {
    super.onResume();
    recyclerView = findViewById(R.id.rv_demonstrator);
    layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL,false);
    recyclerView.setLayoutManager(layoutManager);

    InterfaceDemonstrator interfaceDemonstrator =
RetroServer.KonesiAPI().create(InterfaceDemonstrator.class);
    Call<ModelDemonstrator> tampil =
interfaceDemonstrator.getDemonstrator();
    tampil.enqueue(new Callback<ModelDemonstrator>() {
        @Override
        public void onResponse(Call<ModelDemonstrator> call,
Response<ModelDemonstrator> response) {
            ArrayList<ModelDemonstrator.demonstrator> listData =
response.body().getDemonstrator();
            adapter = new
DataDemonstrator(KelolaDataDemonstrator.this, listData);
            recyclerView.setAdapter(adapter);
            adapter.notifyDataSetChanged();
        }

        @Override
        public void onFailure(Call<ModelDemonstrator> call,
Throwable t) {

        }
    });
}
}
}

```

SourceCode 1. 20 Kelola Data Demonstrator

3) Kelola Data Distributor

```

package com.example.cakraagroindonesia.Activity.KelolaDataMaster;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

```

```

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Adapter.DataDistributor;
import
com.example.cakraagroindonesia.Interface.InterfaceDistributor;
import com.example.cakraagroindonesia.Model.ModelDistributor;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class KelolaDataDistributor extends AppCompatActivity {
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;

    TextView btnTambah;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_data_distributor);

        btnTambah = findViewById(R.id.btntambah);
        btnTambah.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaDataDistributor.this,
TambahDataDistributor.class));
            }
        });
    }

    @Override
    protected void onResume() {
        super.onResume();
        recyclerView = findViewById(R.id.rv_distributor);
        layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL, false);
        recyclerView.setLayoutManager(layoutManager);

        InterfaceDistributor interfaceDistributor =
RetroServer.KonesiAPI().create(InterfaceDistributor.class);
        Call<ModelDistributor> tampil =
interfaceDistributor.getDistributor();

```

```

        tampil.enqueue(new Callback<ModelDistributor>() {
            @Override
            public void onResponse(Call<ModelDistributor> call,
Response<ModelDistributor> response) {
                ArrayList<ModelDistributor.distributor> listData =
response.body().getDistributor();
                adapter = new
DataDistributor(KelolaDataDistributor.this, listData);
                recyclerView.setAdapter(adapter);
                adapter.notifyDataSetChanged();
            }

            @Override
            public void onFailure(Call<ModelDistributor> call,
Throwable t) {

            }
        });
    }
}

```

SourceCode 1. 21 Kelola Data Distributor

4) Kelola Data Manager

```

package com.example.cakraagroindonesia.Activity.KelolaDataMaster;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Adapter.DataManager;
import com.example.cakraagroindonesia.Interface.InterfaceManager;
import com.example.cakraagroindonesia.Model.ModelManager;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

```

```

public class KelolaDataManager extends AppCompatActivity {
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;

    TextView btnTambah;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_data_manager);

        btnTambah = findViewById(R.id.btntambah);
        btnTambah.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaDataManager.this,
TambahDataManager.class));
            }
        });
    }

    @Override
    protected void onResume() {
        super.onResume();
        recyclerView = findViewById(R.id.rv_manager);
        layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL, false);
        recyclerView.setLayoutManager(layoutManager);

        InterfaceManager interfaceManager =
RetroServer.KonesiAPI().create(InterfaceManager.class);
        Call<ModelManager> tampil = interfaceManager.getManager();
        tampil.enqueue(new Callback<ModelManager>() {
            @Override
            public void onResponse(Call<ModelManager> call,
Response<ModelManager> response) {
                ArrayList<ModelManager.manager> listData =
response.body().getManager();
                adapter = new DataManager(KelolaDataManager.this,
listData);
                recyclerView.setAdapter(adapter);
                adapter.notifyDataSetChanged();
            }
        });
    }
}

```

```

        public void onFailure(Call<ModelManager> call, Throwable
t) {
            }
        });
    }
}

```

SourceCode 1. 22 Kelola Data Manager

5) Kelola Data Master

```

package com.example.cakraagroindonesia.Activity.KelolaDataMaster;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.R;

public class KelolaDataMaster extends AppCompatActivity {
    TextView btnDataadmin, btnDataSalesSecretary, btnDataSupervisor,
    btnDataDemonstrator, btnDataDistributor, btnDataManager;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_data_master);

        deklarasi();
        btnDataadmin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaDataMaster.this,
KelolaDataAdmin.class));
            }
        });
        btnDataSalesSecretary.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaDataMaster.this,
KelolaDataSecretary.class));
            }
        });
    }
}

```

```

        btnDataSupervisor.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaDataMaster.this,
KelolaDataSupervisor.class));
            }
        });
        btnDataDemonstrator.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaDataMaster.this,
KelolaDataDemonstrator.class));
            }
        });
        btnDataDistributor.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaDataMaster.this,
KelolaDataDistributor.class));
            }
        });
        btnDataManager.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaDataMaster.this,
KelolaDataManager.class));
            }
        });
    }

    private void deklarasi() {
        btnDataadmin = findViewById(R.id.btndataadmin);
        btnDataSalesSecretary =
findViewById(R.id.btndatasalessecretary);
        btnDataSupervisor = findViewById(R.id.btndatasupervisor);
        btnDataDemonstrator =
findViewById(R.id.btndatademonstrator);
        btnDataDistributor = findViewById(R.id.btndatadistributor);
        btnDataManager = findViewById(R.id.btndatamanager);
    }
}

```

SourceCode 1. 23 Kelola Data Master

6) Kelola Data Secretary

```
package com.example.cakraagroindonesia.Activity.KelolaDataMaster;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Adapter.DataSecretary;
import com.example.cakraagroindonesia.Interface.InterfaceSecretary;
import com.example.cakraagroindonesia.Model.ModelSecretary;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class KelolaDataSecretary extends AppCompatActivity {
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;

    TextView btnTambah;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_data_secretary);
        btnTambah = findViewById(R.id.btntambah);
        btnTambah.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaDataSecretary.this,
TambahDataSecretary.class));
            }
        });
    }

    @Override
    protected void onResume() {
```

```

        super.onResume();
        recyclerView = findViewById(R.id.rv_kelolasecretary);
        layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL,false);
        recyclerView.setLayoutManager(layoutManager);

        InterfaceSecretary interfaceSecretary =
RetroServer.KonesiAPI().create(InterfaceSecretary.class);
        Call<ModelSecretary> tampil =
interfaceSecretary.getSecretary();
        tampil.enqueue(new Callback<ModelSecretary>() {
            @Override
            public void onResponse(Call<ModelSecretary> call,
Response<ModelSecretary> response) {
                ArrayList<ModelSecretary.secretary> listData =
response.body().getSecretary();
                adapter = new
DataSecretary(KelolaDataSecretary.this, listData);
                recyclerView.setAdapter(adapter);
                adapter.notifyDataSetChanged();
            }

            @Override
            public void onFailure(Call<ModelSecretary> call,
Throwable t) {

            }
        });
    }
}

```

SourceCode 1. 24 Kelola Data Secretary

7) Kelola Data Supervisor

```

package com.example.cakraagroindonesia.Activity.KelolaDataMaster;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;

```



```

import com.example.cakraagroindonesia.Adapter.DataSupervisor;
import com.example.cakraagroindonesia.Interface.InterfaceSupervisor;
import com.example.cakraagroindonesia.Model.ModelSupervisor;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class KelolaDataSupervisor extends AppCompatActivity {
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;

    TextView btnTambah;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_data_supervisor);
        btnTambah = findViewById(R.id.btntambah);
        btnTambah.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaDataSupervisor.this,
TambahDataSupervisor.class));
            }
        });
    }

    @Override
    protected void onResume() {
        super.onResume();
        recyclerView = findViewById(R.id.rv_supervisor);
        layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL, false);
        recyclerView.setLayoutManager(layoutManager);

        InterfaceSupervisor interfaceSupervisor =
RetroServer.KonesiAPI().create(InterfaceSupervisor.class);
        Call<ModelSupervisor> tampil =
interfaceSupervisor.getSupervisor();
        tampil.enqueue(new Callback<ModelSupervisor>() {
            @Override
            public void onResponse(Call<ModelSupervisor> call,
Response<ModelSupervisor> response) {

```



```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_tambah_data_admin);

    btnSubmit = findViewById(R.id.btnsubmit);
    Nama = findViewById(R.id.nama);
    Username = findViewById(R.id.username);
    Password = findViewById(R.id.password);

    btnSubmit.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            nama = Nama.getText().toString();
            username = Username.getText().toString();
            password = Password.getText().toString();
            level = "admin";

            if (!nama.isEmpty() && !username.isEmpty() &&
!password.isEmpty()){

                ModelAdmin.data_admin newData = new
ModelAdmin.data_admin();
                newData.setNama(nama);
                newData.setUsername(username);
                newData.setPassword(password);
                newData.setLevel(level);

                ArrayList<ModelAdmin.data_admin> listData = new
ArrayList<>();
                listData.add(newData);

                ModelAdmin modelAdmin = new ModelAdmin();
                modelAdmin.setData_admin(listData);

                InterfaceAdmin interfaceAdmin =
RetroServer.KonesiAPI().create(InterfaceAdmin.class);
                Call<ModelAdmin> simpan =
interfaceAdmin.setAdmin(nama,username,password,level);
                simpan.enqueue(new Callback<ModelAdmin>() {
                    @Override
                    public void onResponse(Call<ModelAdmin>
call, Response<ModelAdmin> response) {
                        Toast.makeText(TambahDataAdmin.this,
>Data Berhasil Disimpan", Toast.LENGTH_SHORT).show();

```



```

import androidx.core.content.ContextCompat;

import com.example.cakraagroindonesia.API.RetroServer;
import
com.example.cakraagroindonesia.Interface.InterfaceDemonstrator;
import com.example.cakraagroindonesia.Interface.InterfaceSupervisor;
import com.example.cakraagroindonesia.Model.ModelDemonstrator;
import com.example.cakraagroindonesia.Model.ModelSupervisor;
import com.example.cakraagroindonesia.R;

import java.io.File;
import java.util.ArrayList;
import java.util.List;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class TambahDataDemonstrator extends AppCompatActivity {

    private TextView btnsubmit, btnPilihFoto;
    private EditText namademonstrator, provinsi, kabupaten,
    usernamedemonstrator, passworddemonstrator;
    private String NamaDemonstrator, Provinsi, Kabupaten,
    UsernameDemonstrator, PasswordDemonstrator;
    private String namaSupervisorTerpilih, kodeSupervisorTerpilih,
    kodeManagerTerpilih;
    private String level;

    private ImageView foto;
    private Uri selectedImageUri;

    private Spinner spinner;
    private ArrayAdapter spinnerAdapter;

    private static final int REQUEST_CODE_PICK_IMAGE = 101;
    private static final int PERMISSION_REQUEST_CODE = 102;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tambah_data_demonstrator);

        namademonstrator = findViewById(R.id.namademonstrator);

```

```

        provinsi = findViewById(R.id.provinsi);
        kabupaten = findViewById(R.id.kabupaten);
        usernamedemonstrator =
findViewById(R.id.usernamedemonstrator);
        passworddemonstrator =
findViewById(R.id.passworddemonstrator);
        btnsubmit = findViewById(R.id.btnsubmit);
        spinner = findViewById(R.id.spinner);
        btnPilihFoto = findViewById(R.id.btnPilihFoto);
        foto = findViewById(R.id.foto);

        InterfaceSupervisor interfaceSupervisor =
RetroServer.KonesiAPI().create(InterfaceSupervisor.class);
        Call<ModelSupervisor> Spinner =
interfaceSupervisor.getSupervisor();
        Spinner.enqueue(new Callback<ModelSupervisor>() {
            @Override
            public void onResponse(Call<ModelSupervisor> call,
Response<ModelSupervisor> response) {
                ModelSupervisor modelSupervisor = response.body();
                ArrayList<ModelSupervisor.supervisor> supervisorList
= modelSupervisor.getSupervisor();

                List<String> namaSupervisor = new ArrayList<>();
                for (ModelSupervisor.supervisor nama :
supervisorList){
                    namaSupervisor.add(nama.getNama_supervisor());
                }
                List<String> kodeSupervisor = new ArrayList<>();
                for (ModelSupervisor.supervisor kode :
supervisorList){
                    kodeSupervisor.add(kode.getKode_sv());
                }
                List<String> kodeManager = new ArrayList<>();
                for (ModelSupervisor.supervisor kodeMG :
supervisorList){
                    kodeManager.add(kodeMG.getKode_mg());
                }

                spinnerAdapter = new
ArrayAdapter<>(TambahDataDemonstrator.this,
android.R.layout.simple_spinner_item, namaSupervisor);
                spinnerAdapter.setDropDownViewResource(android.R.lay
out.simple_spinner_dropdown_item);
                spinner.setAdapter(spinnerAdapter);
            }
        });
    }
}

```

```

        spinner.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?>
adapterView, View view, int i, long l) {
                namaSupervisorTerpilih =
namaSupervisor.get(i);
                kodeSupervisorTerpilih =
kodeSupervisor.get(i);
                kodeManagerTerpilih = kodeManager.get(i);
            }
            @Override
            public void onNothingSelected(AdapterView<?>
adapterView) {
            }
        });
    }

    @Override
    public void onFailure(Call<ModelSupervisor> call,
Throwable t) {
    }
});

    Log.d("MyTag", "onCreate:
"+kodeManagerTerpilih+kodeManagerTerpilih+kodeSupervisorTerpilih);

//      InterfaceDemonstrator interfaceDemonstrator =
RetroServer.KonesiAPI().create(InterfaceDemonstrator.class);
//      Call<ModelDemonstrator> newid =
interfaceDemonstrator.getDemonstrator();
//      newid.enqueue(new Callback<ModelDemonstrator>() {
//          @Override
//          public void onResponse(Call<ModelDemonstrator> call,
Response<ModelDemonstrator> response) {
//              ArrayList<ModelDemonstrator.demonstrator> id =
response.body().getDemonstrator();
//              String lastId = new
DataDemonstrator(TambahDataDemonstrator.this, id).getLastId();
//              String numericPart = lastId.substring(2);
//              int lastNumber = Integer.parseInt(numericPart);
//              int newNumber = lastNumber + 1;
//              newId = "DS" + String.format("%03d", newNumber);

```

```

//      }
//      @Override
//      public void onFailure(Call<ModelDemonstrator> call,
Throwable t) {
//
//      }
//      });

    btnPilihFoto.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            checkAndRequestPermission();
        }
    });

    btnsubmit.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

            level = "demonstrator";
            NamaDemonstrator =
namademonstrator.getText().toString().trim();
            Provinsi = provinsi.getText().toString().trim();
            Kabupaten = kabupaten.getText().toString().trim();
            UsernameDemonstrator =
usernameDemonstrator.getText().toString().trim();
            PasswordDemonstrator =
passworddemonstrator.getText().toString().trim();

            if(!NamaDemonstrator.isEmpty() &&
!Provinsi.isEmpty() && !Kabupaten.isEmpty() &&
!UsernameDemonstrator.isEmpty() && !PasswordDemonstrator.isEmpty()){
                if (selectedImageUri != null){
                    String imagePath =
getRealPathFromURI(selectedImageUri);

                    tambahData(imagePath,
NamaDemonstrator,Provinsi,Kabupaten,UsernameDemonstrator,PasswordDem
onstrator);
                }else{
                    Toast.makeText(TambahDataDemonstrator.this,
"Pilih gambar terlebih dahulu", Toast.LENGTH_SHORT).show();
                }
            } else {
                Toast.makeText(TambahDataDemonstrator.this,
"Semua field harus diisi", Toast.LENGTH_SHORT).show();
            }
        }
    });

```



```

    }
    });
}

private void checkAndRequestPermission() {
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
        openGallery();
    } else {
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
PERMISSION_REQUEST_CODE);
    }
}

private void openGallery() {
    Intent intent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
    startActivityForResult(intent, REQUEST_CODE_PICK_IMAGE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_CODE_PICK_IMAGE && resultCode ==
RESULT_OK && data != null) {
        selectedImageUri = data.getData();
        foto.setImageURI(selectedImageUri);
    }
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
    if (requestCode == PERMISSION_REQUEST_CODE) {
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            openGallery();
        } else {
            Toast.makeText(this, "Izin akses galeri dibutuhkan
untuk memilih foto", Toast.LENGTH_SHORT).show();
        }
    }
}

private String getRealPathFromURI(Uri contentUri) {
    String[] projection = {MediaStore.Images.Media.DATA};
}

```

```

        Cursor cursor = getContentResolver().query(contentUri,
projection, null, null, null);
        if (cursor == null) {
            return contentUri.getPath();
        } else {
            int columnIndex =
cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
            cursor.moveToFirst();
            String filePath = cursor.getString(columnIndex);
            cursor.close();
            return filePath;
        }
    }
}

private void tambahData(String imagePath, String
S>NamaDemonstrator, String SProvinsi, String SKabupaten, String
SUsernameDemonstrator, String SPasswordDemonstrator) {
    File imageFile = new File(imagePath);
    RequestBody imageRequestBody =
RequestBody.create(MediaType.parse("image/*"), imageFile);
    MultipartBody.Part Foto =
MultipartBody.Part.createFormData("fotodemonstrator",
imageFile.getName(), imageRequestBody);
    RequestBody Nama =
RequestBody.create(MediaType.parse("text/plain"),
S>NamaDemonstrator);
    RequestBody Provinsi =
RequestBody.create(MediaType.parse("text/plain"), SProvinsi);
    RequestBody Kabupaten =
RequestBody.create(MediaType.parse("text/plain"), SKabupaten);
    RequestBody Username =
RequestBody.create(MediaType.parse("text/plain"),
SUsernameDemonstrator);
    RequestBody Password =
RequestBody.create(MediaType.parse("text/plain"),
SPasswordDemonstrator);
    RequestBody Supervisor =
RequestBody.create(MediaType.parse("text/plain"),
namaSupervisorTerpilih);
    RequestBody KDSupervisor =
RequestBody.create(MediaType.parse("text/plain"),
kodeSupervisorTerpilih);
    RequestBody KDManager =
RequestBody.create(MediaType.parse("text/plain"),
kodeManagerTerpilih);
    RequestBody Level =
RequestBody.create(MediaType.parse("text/plain"), level);
}

```

```

        Log.d("MyTag", "onResponse: " + imagePath +
        SNamaDemonstrator + SProvinsi + SKabupaten + SUsernameDemonstrator +
        SPasswordDemonstrator + namaSupervisorTerpilih +
        kodeSupervisorTerpilih + kodeManagerTerpilih + level);

        InterfaceDemonstrator interfaceDemonstrator =
        RetroServer.KonesiAPI().create(InterfaceDemonstrator.class);
        Call<ModelDemonstrator> simpan =
        interfaceDemonstrator.setDemonstrator(Foto, Nama, Username, Password, Su
        pervisor, KDsupervisor, Kabupaten, KDManager, Provinsi, Level);
        simpan.enqueue(new Callback<ModelDemonstrator>() {
            @Override
            public void onResponse(Call<ModelDemonstrator> call,
            Response<ModelDemonstrator> response) {
                Toast.makeText(TambahDataDemonstrator.this, "Data
                Berhasil Disimpan", Toast.LENGTH_SHORT).show();
                finish();
            }

            @Override
            public void onFailure(Call<ModelDemonstrator> call,
            Throwable t) {
                Toast.makeText(TambahDataDemonstrator.this, "Data
                Gagal Disimpan", Toast.LENGTH_SHORT).show();
            }
        });
    }
}

```

SourceCode 1. 27 Tambah Data Demonstrator

10) Tambah Data Distributor

```

package com.example.cakraagroindonesia.Activity.KelolaDataMaster;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;

```

```

import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.example.cakraagroindonesia.API.RetroServer;
import
com.example.cakraagroindonesia.Interface.InterfaceDistributor;
import com.example.cakraagroindonesia.Interface.InterfaceSecretary;
import com.example.cakraagroindonesia.Model.ModelDistributor;
import com.example.cakraagroindonesia.Model.ModelSecretary;
import com.example.cakraagroindonesia.R;

import java.io.File;
import java.util.ArrayList;
import java.util.List;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class TambahDataDistributor extends AppCompatActivity {
    private TextView btnsubmit, btnPilihFoto;
    private EditText
namadistributor,perusahaan,usernameistributor,passworddistributor;
    private String NamaDistributor, Perusahaan, Username, Password,
KDmg;

    private String namaSecretaryTerpilih, kodeSecretaryTerpilih,
kodeManagerTerpilih, newId, level;

    private Spinner spinner;
    private ArrayAdapter spinnerAdapter;

    private ImageView foto;
    private Uri selectedImageUri;

```

```

private static final int REQUEST_CODE_PICK_IMAGE = 101;
private static final int PERMISSION_REQUEST_CODE = 102;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_tambah_data_distributor);

    namadistributor = findViewById(R.id.namadistributor);
    perusahaan = findViewById(R.id.perusahaan);
    usernamedistributor =
findViewById(R.id.usernamedistributor);
    passworddistributor =
findViewById(R.id.passworddistributor);
    spinner = findViewById(R.id.spinner);
    foto = findViewById(R.id.foto);
    btnsubmit = findViewById(R.id.btnsubmit);
    btnPilihFoto = findViewById(R.id.btnPilihFoto);

    InterfaceSecretary interfaceSecretary =
RetroServer.KonesiAPI().create(InterfaceSecretary.class);
    Call<ModelSecretary> Spinner =
interfaceSecretary.getSecretary();
    Spinner.enqueue(new Callback<ModelSecretary>() {
        @Override
        public void onResponse(Call<ModelSecretary> call,
Response<ModelSecretary> response) {
            ModelSecretary modelSecretary = response.body();
            ArrayList<ModelSecretary.secretary> secretaryList =
modelSecretary.getSecretary();

            List<String> namaSecretary = new ArrayList<>();
            for (ModelSecretary.secretary nama : secretaryList){
                namaSecretary.add(nama.getNama_secretary());
            }
            List<String> kodeSecretary = new ArrayList<>();
            for (ModelSecretary.secretary kode : secretaryList){
                kodeSecretary.add(kode.getKode_sc());
            }
            List<String> kodeManager = new ArrayList<>();
            for (ModelSecretary.secretary kodeMg :
secretaryList){
                kodeManager.add(kodeMg.getKode_mg());
            }

```

```

        spinnerAdapter = new
ArrayAdapter<>(TambahDataDistributor.this,
android.R.layout.simple_spinner_item, namaSecretary);
        spinnerAdapter.setDropDownViewResource(android.R.lay
out.simple_spinner_dropdown_item);
        spinner.setAdapter(spinnerAdapter);

        spinner.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?>
adapterView, View view, int i, long l) {
                namaSecretaryTerpilih =
namaSecretary.get(i);
                kodeSecretaryTerpilih =
kodeSecretary.get(i);
                kodeManagerTerpilih = kodeManager.get(i);
            }
            @Override
            public void onNothingSelected(AdapterView<?>
adapterView) {
            }
        });
    }

    @Override
    public void onFailure(Call<ModelSecretary> call,
Throwable t) {
    }
});

//      InterfaceDistributor interfaceDistributor =
RetroServer.KonesiAPI().create(InterfaceDistributor.class);
//      Call<ModelDistributor> newid =
interfaceDistributor.getDistributor();
//      newid.enqueue(new Callback<ModelDistributor>() {
//          @Override
//          public void onResponse(Call<ModelDistributor> call,
Response<ModelDistributor> response) {
//              ArrayList<ModelDistributor.distributor> id =
response.body().getDistributor();
//              String lastId = new
DataDistributor(TambahDataDistributor.this, id).getLastId();
//              String numericPart = lastId.substring(2);
//              int lastNumber = Integer.parseInt(numericPart);

```

```

//          int newNumber = lastNumber + 1;
//
//          newId = "DT" + String.format("%03d", newNumber);
//      }
//      @Override
//      public void onFailure(Call<ModelDistributor> call,
Throwable t) {
//
//      }
//  });

    btnPilihFoto.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            checkAndRequestPermission();
        }
    });

    btnsubmit.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            level = "distributor";
            NamaDistributor =
namadistributor.getText().toString().trim();
            Perusahaan = perusahaan.getText().toString().trim();
            Username =
usernameistributor.getText().toString().trim();
            Password =
passworddistributor.getText().toString().trim();

            if(!NamaDistributor.isEmpty() &&
!Perusahaan.isEmpty() && !Username.isEmpty() &&
!Password.isEmpty()){
                if (selectedImageUri != null){
                    String imagePath =
getRealPathFromURI(selectedImageUri);

                    tambahData(NamaDistributor,Username,Password
,level,kodeSecretaryTerpilih,namaSecretaryTerpilih,imagePath,kodeMan
agerTerpilih,Perusahaan);
                }else{
                    Toast.makeText(TambahDataDistributor.this,
"Pilih gambar terlebih dahulu", Toast.LENGTH_SHORT).show();
                }
            } else {

```



```

private String getRealPathFromURI(Uri contentUri) {
    String[] projection = {MediaStore.Images.Media.DATA};
    Cursor cursor = getContentResolver().query(contentUri,
projection, null, null, null);
    if (cursor == null) {
        return contentUri.getPath();
    } else {
        int columnIndex =
cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
        cursor.moveToFirst();
        String filePath = cursor.getString(columnIndex);
        cursor.close();
        return filePath;
    }
}

private void tambahData(String SNamaDistributor, String
SUsername, String SPassword, String Slevel, String
SkodeSecretaryTerpilih, String SnamaSecretaryTerpilih, String
imagePath, String SkodeManagerTerpilih, String SPerusahaan) {
    File imageFile = new File(imagePath);
    RequestBody imageRequestBody =
RequestBody.create(MediaType.parse("image/*"), imageFile);
    MultipartBody.Part Foto =
MultipartBody.Part.createFormData("fotodistributor",
imageFile.getName(), imageRequestBody);
    RequestBody Nama =
RequestBody.create(MediaType.parse("text/plain"), SNamaDistributor);
    RequestBody Username =
RequestBody.create(MediaType.parse("text/plain"), SUsername);
    RequestBody Password =
RequestBody.create(MediaType.parse("text/plain"), SPassword);
    RequestBody Level =
RequestBody.create(MediaType.parse("text/plain"), Slevel);
    RequestBody KodeSC =
RequestBody.create(MediaType.parse("text/plain"),
SkodeSecretaryTerpilih);
    RequestBody NamaSC =
RequestBody.create(MediaType.parse("text/plain"),
SnamaSecretaryTerpilih);
    RequestBody KodeMG =
RequestBody.create(MediaType.parse("text/plain"),
SkodeManagerTerpilih);
    RequestBody Perusahaan =
RequestBody.create(MediaType.parse("text/plain"), SPerusahaan);

```

```

        Log.d("MyTag", "onResponse: " + imageRequestBody +
        SNamaDistributor + SPerusahaan + SUsername + SPasssword +
        SnamaSecretaryTerpilih + SkodeSecretaryTerpilih + Slevel +
        SkodeManagerTerpilih);

        InterfaceDistributor interfaceDistributor =
        RetroServer.KonesiAPI().create(InterfaceDistributor.class);
        Call<ModelDistributor> simpan =
        interfaceDistributor.setDistributor(Foto, Nama, Username, Password, Level,
        KodeSC, NamaSC, Perusahaan, KodeMG);
        simpan.enqueue(new Callback<ModelDistributor>() {
            @Override
            public void onResponse(Call<ModelDistributor> call,
            Response<ModelDistributor> response) {
                Toast.makeText(TambahDataDistributor.this, "Data
                Berhasil Disimpan", Toast.LENGTH_SHORT).show();
                finish();
            }

            @Override
            public void onFailure(Call<ModelDistributor> call,
            Throwable t) {
                Toast.makeText(TambahDataDistributor.this, "Data
                Gagal Disimpan", Toast.LENGTH_SHORT).show();
            }
        });
    }
}

```

SourceCode 1. 28 Tambah Data Distributor

11) Tambah Data Manager

```

package com.example.cakraagroindonesia.Activity.KelolaDataMaster;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageView;

```

```

import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Interface.InterfaceManager;
import com.example.cakraagroindonesia.Model.ModelManager;
import com.example.cakraagroindonesia.R;

import java.io.File;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class TambahDataManager extends AppCompatActivity {
    private EditText namamanager, usernamemanager, passwordmanager;
    private String NamaManager, UsernameManager,
    PasswordManager, level, newId;
    private TextView btnPilihFoto, btnsubmit;

    private ImageView foto;

    private Uri selectedImageUri;

    private static final int REQUEST_CODE_PICK_IMAGE = 101;
    private static final int PERMISSION_REQUEST_CODE = 102;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tambah_data_manager);

        namamanager = findViewById(R.id.namamanager);
        usernamemanager = findViewById(R.id.usernamemanager);
        passwordmanager = findViewById(R.id.passwordmanager);
        btnPilihFoto = findViewById(R.id.btnPilihFoto);
        btnsubmit = findViewById(R.id.btnsubmit);
        foto = findViewById(R.id.foto);

```

```

//      InterfaceManager interfaceManager =
RetroServer.KonesiAPI().create(InterfaceManager.class);
//      Call<ModelManager> newid = interfaceManager.getManager();
//      newid.enqueue(new Callback<ModelManager>() {
//          @Override
//          public void onResponse(Call<ModelManager> call,
Response<ModelManager> response) {
//              ArrayList<ModelManager.manager> id =
response.body().getManager();
//              String lastId = new
DataManager(TambahDataManager.this, id).getLastId();
//              String numericPart = lastId.substring(2);
//              int lastNumber = Integer.parseInt(numericPart);
//              int newNumber = lastNumber + 1;
//
//              newId = "MG" + String.format("%03d", newNumber);
//          }
//          @Override
//          public void onFailure(Call<ModelManager> call,
Throwable t) {
//
//          }
//      });

    btnPilihFoto.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            checkAndRequestPermission();
        }
    });

    btnsubmit.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            NamaManager =
namamanager.getText().toString().trim();
            UsernameManager =
usernamemanager.getText().toString().trim();
            PasswordManager =
passwordmanager.getText().toString().trim();
            level = "manager";

            if (!NamaManager.isEmpty() &&
!UsernameManager.isEmpty() && !PasswordManager.isEmpty()) {
                if (selectedImageUri != null) {

```

```

        String imagePath =
getRealPathFromURI(selectedImageUri);

        tambahData(imagePath, NamaManager,
UsernameManager, PasswordManager);
    } else {
        Toast.makeText(TambahDataManager.this,
"Pilih gambar terlebih dahulu", Toast.LENGTH_SHORT).show();
    }
    } else {
        Toast.makeText(TambahDataManager.this, "Semua
field harus diisi", Toast.LENGTH_SHORT).show();
    }
    }
});
}
private void checkAndRequestPermission() {
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
        openGallery();
    } else {
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
PERMISSION_REQUEST_CODE);
    }
}

private void openGallery() {
    Intent intent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
    startActivityForResult(intent, REQUEST_CODE_PICK_IMAGE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_CODE_PICK_IMAGE && resultCode ==
RESULT_OK && data != null) {
        selectedImageUri = data.getData();
        foto.setImageURI(selectedImageUri);
    }
}

@Override

```

```

    public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        if (requestCode == PERMISSION_REQUEST_CODE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                openGallery();
            } else {
                Toast.makeText(this, "Izin akses galeri dibutuhkan
untuk memilih foto", Toast.LENGTH_SHORT).show();
            }
        }
    }

    private String getRealPathFromURI(Uri contentUri) {
        String[] projection = {MediaStore.Images.Media.DATA};
        Cursor cursor = getContentResolver().query(contentUri,
projection, null, null, null);
        if (cursor == null) {
            return contentUri.getPath();
        } else {
            int columnIndex =
cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
            cursor.moveToFirst();
            String filePath = cursor.getString(columnIndex);
            cursor.close();
            return filePath;
        }
    }

    private void tambahData(String imagePath, String SNamaManager,
String SUsernameManager, String SPasswordManager) {
        File imageFile = new File(imagePath);
        RequestBody imageRequestBody =
RequestBody.create(MediaType.parse("image/*"), imageFile);
        MultipartBody.Part Foto =
MultipartBody.Part.createFormData("fotomanager",
imageFile.getName(), imageRequestBody);
        RequestBody Nama =
RequestBody.create(MediaType.parse("text/plain"), SNamaManager);
        RequestBody Username =
RequestBody.create(MediaType.parse("text/plain"), SUsernameManager);
        RequestBody Password =
RequestBody.create(MediaType.parse("text/plain"), SPasswordManager);
        RequestBody Level =
RequestBody.create(MediaType.parse("text/plain"), level);
    }

```

```

        Log.d("MyTag", "onResponse: " + imagePath + SNamaManager +
SUsernameManager + SPasswOrdManager + newId + level);

        InterfaceManager interfaceManager =
RetroServer.KonesiAPI().create(InterfaceManager.class);
        Call<ModelManager> simpan =
interfaceManager.setManager(Foto, Nama, Username, Password, Level);
        simpan.enqueue(new Callback<ModelManager>() {
            @Override
            public void onResponse(Call<ModelManager> call,
Response<ModelManager> response) {
                Toast.makeText(TambahDataManager.this, "Data
Berhasil Disimpan", Toast.LENGTH_SHORT).show();
                finish();
            }

            @Override
            public void onFailure(Call<ModelManager> call, Throwable
t) {
                Toast.makeText(TambahDataManager.this, "Data Gagal
Disimpan", Toast.LENGTH_SHORT).show();
            }
        });
    }
}

```

SourceCode 1. 29 Tambah Data Manager

12) Tambah Data Secretary

```

package com.example.cakraagroindonesia.Activity.KelolaDataMaster;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;

```

```

import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Adapter.DataSecretary;
import com.example.cakraagroindonesia.Interface.InterfaceManager;
import com.example.cakraagroindonesia.Interface.InterfaceSecretary;
import com.example.cakraagroindonesia.Model.ModelManager;
import com.example.cakraagroindonesia.Model.ModelSecretary;
import com.example.cakraagroindonesia.R;

import java.io.File;
import java.util.ArrayList;
import java.util.List;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class TambahDataSecretary extends AppCompatActivity {
    private TextView btnSubmit, btnPilihFoto;
    private EditText Nama, Username, Password;
    private String newId, nama, username, password, level;
    private String namaManagerTerpilih, kodeManagerTerpilih;

    private Spinner spinner;
    private ArrayAdapter spinnerAdapter;

    private ImageView foto;
    private Uri selectedImageUri;

    private static final int REQUEST_CODE_PICK_IMAGE = 101;
    private static final int PERMISSION_REQUEST_CODE = 102;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

```



```

        setContentView(R.layout.activity_tambah_data_secretary);

        btnSubmit = findViewById(R.id.btnsubmit);
        Nama = findViewById(R.id.nama);
        Username = findViewById(R.id.username);
        Password = findViewById(R.id.password);
        foto = findViewById(R.id.foto);
        btnPilihFoto = findViewById(R.id.btnPilihFoto);
        spinner = findViewById(R.id.spinnermanager);

        InterfaceManager interfaceManager =
RetroServer.KonesiAPI().create(InterfaceManager.class);
        Call<ModelManager> call = interfaceManager.getManager();
        call.enqueue(new Callback<ModelManager>() {
            @Override
            public void onResponse(Call<ModelManager> call,
Response<ModelManager> response) {
                ModelManager modelManager = response.body();
                ArrayList<ModelManager.manager> managerList =
modelManager.getManager();

                List<String> namaManager = new ArrayList<>();
                for (ModelManager.manager nama : managerList) {
                    namaManager.add(nama.getNama_manager());
                }
                List<String> kodeManager = new ArrayList<>();
                for (ModelManager.manager kode : managerList){
                    kodeManager.add(kode.getKode_mg());
                }

                spinnerAdapter = new
ArrayAdapter<>(TambahDataSecretary.this,
android.R.layout.simple_spinner_item, namaManager);
                spinnerAdapter.setDropDownViewResource(android.R.lay
out.simple_spinner_dropdown_item);
                spinner.setAdapter(spinnerAdapter);

                spinner.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
                    @Override
                    public void onItemSelected(AdapterView<?>
adapterView, View view, int position, long id) {
                        namaManagerTerpilih =
namaManager.get(position);
                        kodeManagerTerpilih =
kodeManager.get(position);
                    }
                })
            }
        });

```

```

        @Override
        public void onNothingSelected(AdapterView<?>
adapterView) {
            }
        });
    }

    @Override
    public void onFailure(Call<ModelManager> call, Throwable
t) {
        }
    });

    InterfaceSecretary interfaceSecretary =
RetroServer.KonesiAPI().create(InterfaceSecretary.class);
    Call<ModelSecretary> newid =
interfaceSecretary.getSecretary();
    newid.enqueue(new Callback<ModelSecretary>() {
        @Override
        public void onResponse(Call<ModelSecretary> call,
Response<ModelSecretary> response) {
            ArrayList<ModelSecretary.secretary> id =
response.body().getSecretary();
            String lastId = new
DataSecretary(TambahDataSecretary.this, id).getLastId();
            String numericPart = lastId.substring(2);
            int lastNumber = Integer.parseInt(numericPart);
            int newNumber = lastNumber + 1;

            newId = "SC" + String.format("%03d", newNumber);
        }
        @Override
        public void onFailure(Call<ModelSecretary> call,
Throwable t) {
        }
    });

    btnPilihFoto.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            checkAndRequestPermission();
        }
    });
});

```

```

        btnSubmit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                nama = Nama.getText().toString();
                username = Username.getText().toString();
                password = Password.getText().toString();
                level = "secretary";

                if (!nama.isEmpty() && !username.isEmpty() &&
!password.isEmpty() && !newId.isEmpty() &&
!namaManagerTerpilih.isEmpty() && !kodeManagerTerpilih.isEmpty() &&
!level.isEmpty()) {
                    if (selectedImageUri != null){
                        String imagePath =
getRealPathFromURI(selectedImageUri);
                        tambahData(imagePath,nama,username,password,
newId,namaManagerTerpilih,kodeManagerTerpilih,level);
                    } else {
                        Toast.makeText(TambahDataSecretary.this,
"Pilih Foto terlebih dahulu", Toast.LENGTH_SHORT).show();
                    }
                    } else {
                        Toast.makeText(TambahDataSecretary.this, "Semua
field harus diisi", Toast.LENGTH_SHORT).show();
                    }
                }
            });
        }

        private void checkAndRequestPermission() {
            if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
                openGallery();
            } else {
                ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
PERMISSION_REQUEST_CODE);
            }
        }

        private void openGallery() {
            Intent intent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
            startActivityForResult(intent, REQUEST_CODE_PICK_IMAGE);
        }
    }

```

```

    @Override
    protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == REQUEST_CODE_PICK_IMAGE && resultCode ==
RESULT_OK && data != null) {
            selectedImageUri = data.getData();
            foto.setImageURI(selectedImageUri);
        }
    }
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        if (requestCode == PERMISSION_REQUEST_CODE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                openGallery();
            } else {
                Toast.makeText(this, "Izin akses galeri dibutuhkan
untuk memilih foto", Toast.LENGTH_SHORT).show();
            }
        }
    }
    private String getRealPathFromURI(Uri contentUri) {
        String[] projection = {MediaStore.Images.Media.DATA};
        Cursor cursor = getContentResolver().query(contentUri,
projection, null, null, null);
        if (cursor == null) {
            return contentUri.getPath();
        } else {
            int columnIndex =
cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
            cursor.moveToFirst();
            String filePath = cursor.getString(columnIndex);
            cursor.close();
            return filePath;
        }
    }

    private void tambahData(String imagePath, String Snama,String
Susername,String Spassword,String SnewId, String
SnamaManagerTerpilih, String SkodeManagerTerpilih,String Slevel) {
        File imageFile = new File(imagePath);
    }

```

```

        RequestBody imageRequestBody =
RequestBody.create(MediaType.parse("image/*"), imageFile);
        MultipartBody.Part Foto =
MultipartBody.Part.createFormData("fotosecretary",
imageFile.getName(), imageRequestBody);
        RequestBody Nama =
RequestBody.create(MediaType.parse("text/plain"), Snama);
        RequestBody Username =
RequestBody.create(MediaType.parse("text/plain"), Susername);
        RequestBody Password =
RequestBody.create(MediaType.parse("text/plain"), Spassword);
        RequestBody KDmanager =
RequestBody.create(MediaType.parse("text/plain"),
SkodeManagerTerpilih);
        RequestBody NamaManager =
RequestBody.create(MediaType.parse("text/plain"),
SnamaManagerTerpilih);
        RequestBody Level =
RequestBody.create(MediaType.parse("text/plain"), Slevel);
        Log.d("MyTag", "onClick: "+newId+ " "+kodeManagerTerpilih+
"+namaManagerTerpilih+ " "+level+ " "+Susername+ " "+Spassword+
"+Snama+ " "+imagePath);

        InterfaceSecretary interfaceSimpan =
RetroServer.KonesiAPI().create(InterfaceSecretary.class);
        Call<ModelSecretary> simpan =
interfaceSimpan.setSecretary(Foto,Nama,Username>Password,Level,KDman
ager,NamaManager);
        simpan.enqueue(new Callback<ModelSecretary>() {
            @Override
            public void onResponse(Call<ModelSecretary> call,
Response<ModelSecretary> response) {
                Toast.makeText(TambahDataSecretary.this, "Data
Berhasil Disimpan", Toast.LENGTH_SHORT).show();
                finish();
            }

            @Override
            public void onFailure(Call<ModelSecretary> call,
Throwable t) {
                Toast.makeText(TambahDataSecretary.this, "Data Gagal
Disimpan", Toast.LENGTH_SHORT).show();
            }
        });
    }
}

```

SourceCode 1. 30 Tambah Data Secretary

13) Tambah Data Supervisor

```
package com.example.cakraagroindonesia.Activity.KelolaDataMaster;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Interface.InterfaceManager;
import com.example.cakraagroindonesia.Interface.InterfaceSupervisor;
import com.example.cakraagroindonesia.Model.ModelManager;
import com.example.cakraagroindonesia.Model.ModelSupervisor;
import com.example.cakraagroindonesia.R;

import java.io.File;
import java.util.ArrayList;
import java.util.List;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class TambahDataSupervisor extends AppCompatActivity {
    private TextView btnSubmit, btnPilihFoto;
```

```

    private EditText Nama, Budget, AreaSales, Provinsi, Username,
    Password;
    private Spinner spinner;
    private ArrayAdapter spinnerAdapter;
    private String newId,namaSupervisor,budget, areasales,
    provinsi,username,password,level;
    private String namaManagerTerpilih,kodeManagerTerpilih;

    private ImageView foto;
    private Uri selectedImageUri;

    private static final int REQUEST_CODE_PICK_IMAGE = 101;
    private static final int PERMISSION_REQUEST_CODE = 102;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tambah_data_supervisor);

        spinner = findViewById(R.id.spinnermanager);
        Nama = findViewById(R.id.namasupervisor);
        Budget = findViewById(R.id.budget);
        AreaSales = findViewById(R.id.areasales);
        Provinsi = findViewById(R.id.provinsi);
        Username = findViewById(R.id.username);
        Password = findViewById(R.id.password);
        btnSubmit = findViewById(R.id.btnsubmit);
        foto = findViewById(R.id.foto);
        btnPilihFoto = findViewById(R.id.btnPilihFoto);

        InterfaceManager interfaceManager =
        RetroServer.KonesiAPI().create(InterfaceManager.class);
        Call<ModelManager> call = interfaceManager.getManager();
        call.enqueue(new Callback<ModelManager>() {
            @Override
            public void onResponse(Call<ModelManager> call,
            Response<ModelManager> response) {
                ModelManager modelManager = response.body();
                ArrayList<ModelManager.manager> managerList =
                modelManager.getManager();

                List<String> namaManager = new ArrayList<>();
                for (ModelManager.manager nama : managerList) {
                    namaManager.add(nama.getNama_manager());
                }
                List<String> kodeManager = new ArrayList<>();

```

```

        for (ModelManager.manager kode : managerList){
            kodeManager.add(kode.getKode_mg());
        }

        spinnerAdapter = new
ArrayAdapter<>(TambahDataSupervisor.this,
android.R.layout.simple_spinner_item, namaManager);
        spinnerAdapter.setDropDownViewResource(android.R.lay
out.simple_spinner_dropdown_item);
        spinner.setAdapter(spinnerAdapter);

        spinner.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?>
adapterView, View view, int position, long id) {
                namaManagerTerpilih =
namaManager.get(position);
                kodeManagerTerpilih =
kodeManager.get(position);
            }

            @Override
            public void onNothingSelected(AdapterView<?>
adapterView) {
                // Handle jika tidak ada yang dipilih
            }

        });
    }

    @Override
    public void onFailure(Call<ModelManager> call, Throwable
t) {

    }

});

//      InterfaceSupervisor interfaceSupervisor =
RetroServer.KonesiAPI().create(InterfaceSupervisor.class);
//      Call<ModelSupervisor> newid =
interfaceSupervisor.getSupervisor();
//      newid.enqueue(new Callback<ModelSupervisor>() {
//          @Override
//          public void onResponse(Call<ModelSupervisor> call,
Response<ModelSupervisor> response) {

```



```

//          ArrayList<ModelSupervisor.supervisor> id =
response.body().getSupervisor();
//          String lastId = new
DataSupervisor(TambahDataSupervisor.this, id).getLastId();
//          String numericPart = lastId.substring(2);
//          int lastNumber = Integer.parseInt(numericPart);
//          int newNumber = lastNumber + 1;
//
//          newId = "SV" + String.format("%03d", newNumber);
//      }
//      @Override
//      public void onFailure(Call<ModelSupervisor> call,
Throwable t) {
//
//      }
//  });

    btnPilihFoto.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            checkAndRequestPermission();
        }
    });

    btnSubmit.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            namaSupervisor = Nama.getText().toString().trim();
            budget = Budget.getText().toString().trim();
            areasales = AreaSales.getText().toString().trim();
            provinsi = Provinsi.getText().toString().trim();
            username = Username.getText().toString().trim();
            password = Password.getText().toString().trim();
            level = "supervisor";

            if (!namaSupervisor.isEmpty() && !budget.isEmpty()
&& !areasales.isEmpty() && !provinsi.isEmpty() &&
!username.isEmpty() && !password.isEmpty()) {
                if (selectedImageUri != null) {
                    String imagePath =
getRealPathFromURI(selectedImageUri);

                    tambahData(imagePath, namaSupervisor,
budget, areasales, provinsi, username, password);
                } else {

```

```

        Toast.makeText(TambahDataSupervisor.this,
"Pilih gambar terlebih dahulu", Toast.LENGTH_SHORT).show();
    }
    } else {
        Toast.makeText(TambahDataSupervisor.this, "Semua
field harus diisi", Toast.LENGTH_SHORT).show();
    }
    });
}

private void checkAndRequestPermission() {
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
        openGallery();
    } else {
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
PERMISSION_REQUEST_CODE);
    }
}

private void openGallery() {
    Intent intent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
    startActivityForResult(intent, REQUEST_CODE_PICK_IMAGE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_CODE_PICK_IMAGE && resultCode ==
RESULT_OK && data != null) {
        selectedImageUri = data.getData();
        foto.setImageURI(selectedImageUri);
    }
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
    if (requestCode == PERMISSION_REQUEST_CODE) {

```

```

        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            openGallery();
        } else {
            Toast.makeText(this, "Izin akses galeri dibutuhkan
untuk memilih foto", Toast.LENGTH_SHORT).show();
        }
    }
}

private String getRealPathFromURI(Uri contentUri) {
    String[] projection = {MediaStore.Images.Media.DATA};
    Cursor cursor = getContentResolver().query(contentUri,
projection, null, null, null);
    if (cursor == null) {
        return contentUri.getPath();
    } else {
        int columnIndex =
cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
        cursor.moveToFirst();
        String filePath = cursor.getString(columnIndex);
        cursor.close();
        return filePath;
    }
}

private void tambahData(String imagePath, String
SnamaSupervisor, String Sbudget, String Sareasales, String
Sprovinsi, String Susername, String Spassword) {
    File imageFile = new File(imagePath);
    RequestBody imageRequestBody =
RequestBody.create(MediaType.parse("image/*"), imageFile);
    MultipartBody.Part Foto =
MultipartBody.Part.createFormData("fotosupervisor",
imageFile.getName(), imageRequestBody);
    RequestBody Nama =
RequestBody.create(MediaType.parse("text/plain"), SnamaSupervisor);
    RequestBody Budget =
RequestBody.create(MediaType.parse("text/plain"), Sbudget);
    RequestBody Sales =
RequestBody.create(MediaType.parse("text/plain"), Sareasales);
    RequestBody Provinsi =
RequestBody.create(MediaType.parse("text/plain"), Sprovinsi);
    RequestBody Username =
RequestBody.create(MediaType.parse("text/plain"), Susername);
    RequestBody Password =
RequestBody.create(MediaType.parse("text/plain"), Spassword);
}

```

```

        RequestBody KDmanager =
RequestBody.create(MediaType.parse("text/plain"),
kodeManagerTerpilih);
        RequestBody NamaManager =
RequestBody.create(MediaType.parse("text/plain"),
namaManagerTerpilih);
        RequestBody Level =
RequestBody.create(MediaType.parse("text/plain"), level);

        Log.d("MyTag", "onResponse: " + imagePath + SnamaSupervisor
+ Sbudget + Sareasales + Sprovinsi + Susername + Spassword);

        InterfaceSupervisor interfaceSimpan =
RetroServer.KonesiAPI().create(InterfaceSupervisor.class);
        Call<ModelSupervisor> simpan =
interfaceSimpan.setSupervisor(Foto, KDmanager, NamaManager, Nama, Sales,
Budget, Provinsi, Username, Password, Level);
        simpan.enqueue(new Callback<ModelSupervisor>() {
            @Override
            public void onResponse(Call<ModelSupervisor> call,
Response<ModelSupervisor> response) {
                Toast.makeText(TambahDataSupervisor.this, "Data
Berhasil Disimpan", Toast.LENGTH_SHORT).show();
                finish();
            }

            @Override
            public void onFailure(Call<ModelSupervisor> call,
Throwable t) {
                Toast.makeText(TambahDataSupervisor.this, "Data
Gagal Disimpan", Toast.LENGTH_SHORT).show();
            }
        });
    }
}

```

SourceCode 1. 31 Tambah Data Supervisor

14) Ubah data Admin

```

package com.example.cakraagroindonesia.Activity.KelolaDataMaster;

import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

```

```

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Interface.InterfaceAdmin;
import com.example.cakraagroindonesia.Model.ModelAdmin;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class UbahDataAdmin extends AppCompatActivity {
    private String varId, varNama, varUsername, varPassword;

    private TextView btnSubmit;
    private EditText Nama, Username, Password;
    private String nama, username, password, level;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ubah_data_admin);

        btnSubmit = findViewById(R.id.btnsubmit);
        Nama = findViewById(R.id.nama);
        Username = findViewById(R.id.username);
        Password = findViewById(R.id.password);

        varId = getIntent().getStringExtra("xId");
        varNama = getIntent().getStringExtra("xNama");
        varUsername = getIntent().getStringExtra("xUsername");
        varPassword = getIntent().getStringExtra("xPassword");

        Nama.setText(varNama);
        Username.setText(varUsername);
        Password.setText(varPassword);

        btnSubmit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                nama = Nama.getText().toString();
                username = Username.getText().toString();
                password = Password.getText().toString();
                level = "admin";
            }
        });
    }
}

```


15) Ubah Data Secretary

```
package com.example.cakraagroindonesia.Activity.KelolaDataMaster;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Interface.InterfaceManager;
import com.example.cakraagroindonesia.Interface.InterfaceSecretary;
import com.example.cakraagroindonesia.Model.ModelManager;
import com.example.cakraagroindonesia.Model.ModelSecretary;
import com.example.cakraagroindonesia.R;

import java.io.File;
import java.util.ArrayList;
import java.util.List;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
```

```

import retrofit2.Response;

public class UbahDataSecretary extends AppCompatActivity {
    private String
varId, varNama, varNamaManager, varUsername, varPassword, varTFoto, varFoto;

    private TextView btnSubmit, btnPilihFoto;
    private EditText Nama, Username, Password;
    private String newId, nama, username, password, level;
    private String namaManagerTerpilih, kodeManagerTerpilih;

    private Spinner spinner;
    private ArrayAdapter spinnerAdapter;

    private ImageView foto;
    private Uri selectedImageUri;

    private static final int REQUEST_CODE_PICK_IMAGE = 101;
    private static final int PERMISSION_REQUEST_CODE = 102;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ubah_data_secretary);

        btnSubmit = findViewById(R.id.btnsubmit);
        Nama = findViewById(R.id.nama);
        Username = findViewById(R.id.username);
        Password = findViewById(R.id.password);
        foto = findViewById(R.id.foto);
        btnPilihFoto = findViewById(R.id.btnPilihFoto);
        spinner = findViewById(R.id.spinnermanager);

        varId = getIntent().getStringExtra("xId");
        varNama = getIntent().getStringExtra("xNama");
        varNamaManager = getIntent().getStringExtra("xNamaManager");
        varUsername = getIntent().getStringExtra("xUsername");
        varPassword = getIntent().getStringExtra("xPassword");
        varTFoto = getIntent().getStringExtra("xTFoto");
        varFoto = getIntent().getStringExtra("xFoto");

        Nama.setText(varNama);
        Username.setText(varUsername);
        Password.setText(varPassword);
        Glide.with(this).load(varTFoto).into(foto);

```



```

        InterfaceManager interfaceManager =
RetroServer.KonesiAPI().create(InterfaceManager.class);
        Call<ModelManager> call = interfaceManager.getManager();
        call.enqueue(new Callback<ModelManager>() {
            @Override
            public void onResponse(Call<ModelManager> call,
Response<ModelManager> response) {
                ModelManager modelManager = response.body();
                ArrayList<ModelManager.manager> managerList =
modelManager.getManager();

                List<String> namaManager = new ArrayList<>();
                for (ModelManager.manager nama : managerList) {
                    namaManager.add(nama.getNama_manager());
                }
                List<String> kodeManager = new ArrayList<>();
                for (ModelManager.manager kode : managerList){
                    kodeManager.add(kode.getKode_mg());
                }

                spinnerAdapter = new
ArrayAdapter<>(UbahDataSecretary.this,
android.R.layout.simple_spinner_item, namaManager);
                spinnerAdapter.setDropDownViewResource(android.R.lay
out.simple_spinner_dropdown_item);
                spinner.setAdapter(spinnerAdapter);

                spinner.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
                    @Override
                    public void onItemClick(AdapterView<?>
adapterView, View view, int position, long id) {
                        namaManagerTerpilih =
namaManager.get(position);
                        kodeManagerTerpilih =
kodeManager.get(position);
                    }

                    @Override
                    public void onNothingSelected(AdapterView<?>
adapterView) {
                    }

                });
            }
        });
    }
}

```

```

        @Override
        public void onFailure(Call<ModelManager> call, Throwable
t) {

        }

    });

    //      InterfaceSecretary interfaceSecretary =
RetroServer.KonesiAPI().create(InterfaceSecretary.class);
    //      Call<ModelSecretary> newid =
interfaceSecretary.getSecretary();
    //      newid.enqueue(new Callback<ModelSecretary>() {
    //          @Override
    //          public void onResponse(Call<ModelSecretary> call,
Response<ModelSecretary> response) {
    //              ArrayList<ModelSecretary.secretary> id =
response.body().getSecretary();
    //              String lastId = new
DataSecretary(UbahDataSecretary.this, id).getLastId();
    //              String numericPart = lastId.substring(2);
    //              int lastNumber = Integer.parseInt(numericPart);
    //              int newNumber = lastNumber + 1;
    //
    //              newId = "SC" + String.format("%03d", newNumber);
    //          }
    //          @Override
    //          public void onFailure(Call<ModelSecretary> call,
Throwable t) {
    //
    //          }
    //      });

    btnPilihFoto.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            checkAndRequestPermission();
        }
    });

    btnSubmit.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

            nama = Nama.getText().toString();
            username = Username.getText().toString();
            password = Password.getText().toString();
            level = "secretary";

```

```

        if (!nama.isEmpty() && !username.isEmpty() &&
!password.isEmpty()) {
            if (selectedImageUri != null){
                String imagePath =
getRealPathFromURI(selectedImageUri);
                tambahData(imagePath,nama,username,password,
newId,namaManagerTerpilih,kodeManagerTerpilih,level);
            } else {
                Toast.makeText(UbahDataSecretary.this,
"Pilih Foto terlebih dahulu", Toast.LENGTH_SHORT).show();
            }
        } else {
            Toast.makeText(UbahDataSecretary.this, "Semua
field harus diisi", Toast.LENGTH_SHORT).show();
        }
    }
});
}

private void checkAndRequestPermission() {
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
        openGallery();
    } else {
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
PERMISSION_REQUEST_CODE);
    }
}

private void openGallery() {
    Intent intent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
    startActivityForResult(intent, REQUEST_CODE_PICK_IMAGE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_CODE_PICK_IMAGE && resultCode ==
RESULT_OK && data != null) {
        selectedImageUri = data.getData();
        foto.setImageURI(selectedImageUri);
    }
}
}

```

```

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
    if (requestCode == PERMISSION_REQUEST_CODE) {
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            openGallery();
        } else {
            Toast.makeText(this, "Izin akses galeri dibutuhkan
untuk memilih foto", Toast.LENGTH_SHORT).show();
        }
    }
}

private String getRealPathFromURI(Uri contentUri) {
    String[] projection = {MediaStore.Images.Media.DATA};
    Cursor cursor = getContentResolver().query(contentUri,
projection, null, null, null);
    if (cursor == null) {
        return contentUri.getPath();
    } else {
        int columnIndex =
cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
        cursor.moveToFirst();
        String filePath = cursor.getString(columnIndex);
        cursor.close();
        return filePath;
    }
}

private void tambahData(String imagePath, String Snama,String
Susername,String Spassword,String SnewId, String
SnamaManagerTerpilih, String SkodeManagerTerpilih,String Slevel) {
    File imageFile = new File(imagePath);
    RequestBody imageRequestBody =
RequestBody.create(MediaType.parse("image/*"), imageFile);
    MultipartBody.Part Foto =
MultipartBody.Part.createFormData("fotosecretary",
imageFile.getName(), imageRequestBody);
    RequestBody Nama =
RequestBody.create(MediaType.parse("text/plain"), Snama);
    RequestBody Username =
RequestBody.create(MediaType.parse("text/plain"), Susername);
    RequestBody Password =
RequestBody.create(MediaType.parse("text/plain"), Spassword);

```

```

        RequestBody KDmanager =
RequestBody.create(MediaType.parse("text/plain"),
SkodeManagerTerpilih);
        RequestBody NamaManager =
RequestBody.create(MediaType.parse("text/plain"),
SnamaManagerTerpilih);
        RequestBody Level =
RequestBody.create(MediaType.parse("text/plain"), Slevel);
        Log.d("MyTag", "onClick: "+newId+ " "+kodeManagerTerpilih+
"+namaManagerTerpilih+ " "+level+ " "+Susername+ " "+Spassword+
"+Snama+ " "+imagePath);

        InterfaceSecretary interfaceSimpan =
RetroServer.KonesiAPI().create(InterfaceSecretary.class);
        Call<ModelSecretary> simpan =
interfaceSimpan.updateSecretary(varId,Foto>Nama,Username>Password,Le
vel,KDmanager>NamaManager);
        simpan.enqueue(new Callback<ModelSecretary>() {
            @Override
            public void onResponse(Call<ModelSecretary> call,
Response<ModelSecretary> response) {
                Toast.makeText(UbahDataSecretary.this, "Data
Berhasil Disimpan", Toast.LENGTH_SHORT).show();
                finish();
            }

            @Override
            public void onFailure(Call<ModelSecretary> call,
Throwable t) {
                Toast.makeText(UbahDataSecretary.this, "Data Gagal
Disimpan", Toast.LENGTH_SHORT).show();
            }
        });
    }
}

```

SourceCode 1. 33 Ubah Data Secretary

16) Ubah Data Supervisor

```

package com.example.cakraagroindonesia.Activity.KelolaDataMaster;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.net.Uri;

```

```

import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Interface.InterfaceManager;
import com.example.cakraagroindonesia.Interface.InterfaceSupervisor;
import com.example.cakraagroindonesia.Model.ModelManager;
import com.example.cakraagroindonesia.Model.ModelSupervisor;
import com.example.cakraagroindonesia.R;

import java.io.File;
import java.util.ArrayList;
import java.util.List;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class UbahDataSupervisor extends AppCompatActivity {
    private String varId, varNama, varNamaManager, varBudget,
varAreaSales, varProvinsi, varUsername, varPassword, varTFoto,
varFoto;

    private TextView btnSubmit, btnPilihFoto;
    private EditText Nama, Budget, AreaSales, Provinsi, Username,
Password;
    private Spinner spinner;
    private ArrayAdapter spinnerAdapter;

```

```

    private String newId,namaSupervisor,budget, areasales,
provinsi,username,password,level;
    private String namaManagerTerpilih,kodeManagerTerpilih;

    private ImageView foto;
    private Uri selectedImageUri;

    private static final int REQUEST_CODE_PICK_IMAGE = 101;
    private static final int PERMISSION_REQUEST_CODE = 102;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ubah_data_supervisor);

        spinner = findViewById(R.id.spinnermanager);
        Nama = findViewById(R.id.namasupervisor);
        Budget = findViewById(R.id.budget);
        AreaSales = findViewById(R.id.areasales);
        Provinsi = findViewById(R.id.provinsi);
        Username = findViewById(R.id.username);
        Password = findViewById(R.id.password);
        btnSubmit = findViewById(R.id.btnsubmit);
        foto = findViewById(R.id.foto);
        btnPilihFoto = findViewById(R.id.btnPilihFoto);

        InterfaceManager interfaceManager =
RetroServer.KonesiAPI().create(InterfaceManager.class);
        Call<ModelManager> call = interfaceManager.getManager();
        call.enqueue(new Callback<ModelManager>() {
            @Override
            public void onResponse(Call<ModelManager> call,
Response<ModelManager> response) {
                ModelManager modelManager = response.body();
                ArrayList<ModelManager.manager> managerList =
modelManager.getManager();

                List<String> namaManager = new ArrayList<>();
                for (ModelManager.manager nama : managerList) {
                    namaManager.add(nama.getNama_manager());
                }
                List<String> kodeManager = new ArrayList<>();
                for (ModelManager.manager kode : managerList){
                    kodeManager.add(kode.getKode_mg());
                }
            }
        });
    }

```

```

        spinnerAdapter = new
ArrayAdapter<>(UbahDataSupervisor.this,
android.R.layout.simple_spinner_item, namaManager);
        spinnerAdapter.setDropDownViewResource(android.R.lay
out.simple_spinner_dropdown_item);
        spinner.setAdapter(spinnerAdapter);

        spinner.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?>
adapterView, View view, int position, long id) {
                namaManagerTerpilih =
namaManager.get(position);
                kodeManagerTerpilih =
kodeManager.get(position);
            }

            @Override
            public void onNothingSelected(AdapterView<?>
adapterView) {
                // Handle jika tidak ada yang dipilih
            }

        });

        @Override
        public void onFailure(Call<ModelManager> call, Throwable
t) {

        }

    });

//      InterfaceSupervisor interfaceSupervisor =
RetroServer.KonesiAPI().create(InterfaceSupervisor.class);
//      Call<ModelSupervisor> newid =
interfaceSupervisor.getSupervisor();
//      newid.enqueue(new Callback<ModelSupervisor>() {
//          @Override
//          public void onResponse(Call<ModelSupervisor> call,
Response<ModelSupervisor> response) {
//              ArrayList<ModelSupervisor.supervisor> id =
response.body().getSupervisor();
//              String lastId = new
DataSupervisor(TambahDataSupervisor.this, id).getLastId();
//              String numericPart = lastId.substring(2);

```



```

//          int lastNumber = Integer.parseInt(numericPart);
//          int newNumber = lastNumber + 1;
//
//          newId = "SV" + String.format("%03d", newNumber);
//      }
//      @Override
//      public void onFailure(Call<ModelSupervisor> call,
Throwable t) {
//
//      }
//  });

varId = getIntent().getStringExtra("xId");
varNama = getIntent().getStringExtra("xNama");
varNamaManager = getIntent().getStringExtra("xNamaManager");
varBudget = getIntent().getStringExtra("xBudget");
varAreaSales = getIntent().getStringExtra("xAreaSales");
varProvinsi = getIntent().getStringExtra("xProvinsi");
varUsername = getIntent().getStringExtra("xUsername");
varPassword = getIntent().getStringExtra("xPassword");
varTFoto = getIntent().getStringExtra("xTFoto");
varFoto = getIntent().getStringExtra("xFoto");

Nama.setText(varNama);
Budget.setText(varBudget);
AreaSales.setText(varAreaSales);
Provinsi.setText(varProvinsi);
Username.setText(varUsername);
Password.setText(varPassword);
Glide.with(this).load(varTFoto).into(foto);

btnPilihFoto.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        checkAndRequestPermission();
    }
});

btnSubmit.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        namaSupervisor = Nama.getText().toString().trim();
        budget = Budget.getText().toString().trim();
        areasales = AreaSales.getText().toString().trim();
        provinsi = Provinsi.getText().toString().trim();
        username = Username.getText().toString().trim();
        password = Password.getText().toString().trim();
    }
});

```

```

        level = "supervisor";

        if (!namaSupervisor.isEmpty() && !budget.isEmpty()
&& !areasales.isEmpty() && !provinsi.isEmpty() &&
!username.isEmpty() && !password.isEmpty()) {
            if (selectedImageUri != null) {
                String imagePath =
getRealPathFromURI(selectedImageUri);

                tambahData(imagePath, namaSupervisor,
budget, areasales, provinsi, username, password);
            } else {
                Toast.makeText(UbahDataSupervisor.this,
"Pilih gambar terlebih dahulu", Toast.LENGTH_SHORT).show();
            }
            } else {
                Toast.makeText(UbahDataSupervisor.this, "Semua
field harus diisi", Toast.LENGTH_SHORT).show();
            }
        }
    });
}

private void checkAndRequestPermission() {
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
        openGallery();
    } else {
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
PERMISSION_REQUEST_CODE);
    }
}

private void openGallery() {
    Intent intent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
    startActivityForResult(intent, REQUEST_CODE_PICK_IMAGE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
}

```

```

        if (requestCode == REQUEST_CODE_PICK_IMAGE && resultCode ==
RESULT_OK && data != null) {
            selectedImageUri = data.getData();
            foto.setImageURI(selectedImageUri);
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        if (requestCode == PERMISSION_REQUEST_CODE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                openGallery();
            } else {
                Toast.makeText(this, "Izin akses galeri dibutuhkan
untuk memilih foto", Toast.LENGTH_SHORT).show();
            }
        }
    }

    private String getRealPathFromURI(Uri contentUri) {
        String[] projection = {MediaStore.Images.Media.DATA};
        Cursor cursor = getContentResolver().query(contentUri,
projection, null, null, null);
        if (cursor == null) {
            return contentUri.getPath();
        } else {
            int columnIndex =
cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
            cursor.moveToFirst();
            String filePath = cursor.getString(columnIndex);
            cursor.close();
            return filePath;
        }
    }

    private void tambahData(String imagePath, String
SnamaSupervisor, String Sbudget, String Sareasales, String
Sprovinsi, String Susername, String Spassword) {
        File imageFile = new File(imagePath);
        RequestBody imageRequestBody =
RequestBody.create(MediaType.parse("image/*"), imageFile);
    }

```

```

        MultipartBody.Part Foto =
MultipartBody.Part.createFormData("fotosupervisor",
imageFile.getName(), imageRequestBody);
        RequestBody Nama =
RequestBody.create(MediaType.parse("text/plain"), SnamaSupervisor);
        RequestBody Budget =
RequestBody.create(MediaType.parse("text/plain"), Sbudget);
        RequestBody Sales =
RequestBody.create(MediaType.parse("text/plain"), Sareasales);
        RequestBody Provinsi =
RequestBody.create(MediaType.parse("text/plain"), Sprovinsi);
        RequestBody Username =
RequestBody.create(MediaType.parse("text/plain"), Susername);
        RequestBody Password =
RequestBody.create(MediaType.parse("text/plain"), Spassword);
        RequestBody KDmanager =
RequestBody.create(MediaType.parse("text/plain"),
kodeManagerTerpilih);
        RequestBody NamaManager =
RequestBody.create(MediaType.parse("text/plain"),
namaManagerTerpilih);
        RequestBody Level =
RequestBody.create(MediaType.parse("text/plain"), level);

        Log.d("MyTag", "onResponse: " + imagePath + SnamaSupervisor
+ Sbudget + Sareasales + Sprovinsi + Susername + Spassword);

        InterfaceSupervisor interfaceSimpan =
RetroServer.KonesiAPI().create(InterfaceSupervisor.class);
        Call<ModelSupervisor> simpan =
interfaceSimpan.setSupervisor(Foto,KDmanager,NamaManager,Nama,Sales,
Budget,Provinsi,Username>Password,Level);
        simpan.enqueue(new Callback<ModelSupervisor>() {
            @Override
            public void onResponse(Call<ModelSupervisor> call,
Response<ModelSupervisor> response) {
                Toast.makeText(UbahDataSupervisor.this, "Data
Berhasil Disimpan", Toast.LENGTH_SHORT).show();
                finish();
            }

            @Override
            public void onFailure(Call<ModelSupervisor> call,
Throwable t) {
                Toast.makeText(UbahDataSupervisor.this, "Data Gagal
Disimpan", Toast.LENGTH_SHORT).show();
            }
        });

```

```

    });
}
}

```

SourceCode 1. 34 Ubah Data Supervisor

17) Ubah Demonstrator

```

package com.example.cakraagroindonesia.Activity.KelolaDataMaster;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Interface.InterfaceDemonstrator;
import com.example.cakraagroindonesia.Interface.InterfaceSupervisor;
import com.example.cakraagroindonesia.Model.ModelDemonstrator;
import com.example.cakraagroindonesia.Model.ModelSupervisor;
import com.example.cakraagroindonesia.R;

import java.io.File;
import java.util.ArrayList;
import java.util.List;

import okhttp3.MediaType;

```

```

import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class UbahDemonstrator extends AppCompatActivity {
    private String varId, varNama, varProvinsi, varKabupaten,
varUsername, varPassword, varTFoto, varFoto;

    private TextView btnsubmit, btnPilihFoto;
    private EditText namademonstrator, provinsi, kabupaten,
username, passworddemonstrator;
    private String NamaDemonstrator, Provinsi, Kabupaten,
UsernameDemonstrator, PasswordDemonstrator;
    private String namaSupervisorTerpilih, kodeSupervisorTerpilih,
kodeManagerTerpilih;
    private String level;

    private ImageView fotox;
    private Uri selectedImageUri;

    private Spinner spinner;
    private ArrayAdapter spinnerAdapter;

    private static final int REQUEST_CODE_PICK_IMAGE = 101;
    private static final int PERMISSION_REQUEST_CODE = 102;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ubah_demonstrator);

        namademonstrator = findViewById(R.id.namademonstrator);
        provinsi = findViewById(R.id.provinsi);
        kabupaten = findViewById(R.id.kabupaten);
        username =
findViewById(R.id.username);
        passworddemonstrator =
findViewById(R.id.passworddemonstrator);
        btnsubmit = findViewById(R.id.btnsubmit);
        spinner = findViewById(R.id.spinner);
        btnPilihFoto = findViewById(R.id.btnPilihFoto);
        fotox = findViewById(R.id.fotox);

        varId = getIntent().getStringExtra("xId");
        varNama = getIntent().getStringExtra("xNama");

```

```

varProvinsi = getIntent().getStringExtra("xProvinsi");
varKabupaten = getIntent().getStringExtra("xKabupaten");
varUsername = getIntent().getStringExtra("xUsername");
varPassword = getIntent().getStringExtra("xPassword");
varTFoto = getIntent().getStringExtra("xTFoto");
varFoto = getIntent().getStringExtra("xFoto");

Glide.with(this).load(varTFoto).into(fotox);
namademonstrator.setText(varNama);
provinsi.setText(varProvinsi);
kabupaten.setText(varKabupaten);
username demonstrator.setText(varUsername);
passworddemonstrator.setText(varPassword);

InterfaceSupervisor interfaceSupervisor =
RetroServer.KonesiAPI().create(InterfaceSupervisor.class);
Call<ModelSupervisor> Spinner =
interfaceSupervisor.getSupervisor();
Spinner.enqueue(new Callback<ModelSupervisor>() {
    @Override
    public void onResponse(Call<ModelSupervisor> call,
Response<ModelSupervisor> response) {
        ModelSupervisor modelSupervisor = response.body();
        ArrayList<ModelSupervisor.supervisor> supervisorList
= modelSupervisor.getSupervisor();

        List<String> namaSupervisor = new ArrayList<>();
        for (ModelSupervisor.supervisor nama :
supervisorList){
            namaSupervisor.add(nama.getNama_supervisor());
        }
        List<String> kodeSupervisor = new ArrayList<>();
        for (ModelSupervisor.supervisor kode :
supervisorList){
            kodeSupervisor.add(kode.getKode_sv());
        }
        List<String> kodeManager = new ArrayList<>();
        for (ModelSupervisor.supervisor kodeMG :
supervisorList){
            kodeManager.add(kodeMG.getKode_mg());
        }

        spinnerAdapter = new
ArrayAdapter<>(UbahDemonstrator.this,
android.R.layout.simple_spinner_item, namaSupervisor);
        spinnerAdapter.setDropDownViewResource(android.R.lay
out.simple_spinner_dropdown_item);

```

```

        spinner.setAdapter(spinnerAdapter);

        spinner.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?>
adapterView, View view, int i, long l) {
                namaSupervisorTerpilih =
namaSupervisor.get(i);
                kodeSupervisorTerpilih =
kodeSupervisor.get(i);
                kodeManagerTerpilih = kodeManager.get(i);
            }
            @Override
            public void onNothingSelected(AdapterView<?>
adapterView) {

            }
        });
    }

    @Override
    public void onFailure(Call<ModelSupervisor> call,
Throwable t) {

    }
});

    btnPilihFoto.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            checkAndRequestPermission();
        }
    });

    btnsubmit.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

            level = "demonstrator";
            NamaDemonstrator =
namademonstrator.getText().toString().trim();
            Provinsi = provinsi.getText().toString().trim();
            Kabupaten = kabupaten.getText().toString().trim();
            UsernameDemonstrator =
usernameDemonstrator.getText().toString().trim();

```



```

        PasswordDemonstrator =
passworddemonstrator.getText().toString().trim();

        if(!NamaDemonstrator.isEmpty() &&
!Provinsi.isEmpty() && !Kabupaten.isEmpty() &&
!UsernameDemonstrator.isEmpty() && !PasswordDemonstrator.isEmpty()){
            if (selectedImageUri != null){
                String imagePath =
getRealPathFromURI(selectedImageUri);

                tambahData(imagePath,
NamaDemonstrator,Provinsi,Kabupaten,UsernameDemonstrator,PasswordDem
onstrator);
            }else{
                Toast.makeText(UbahDemonstrator.this, "Pilih
gambar terlebih dahulu", Toast.LENGTH_SHORT).show();
            }
        } else {
            Toast.makeText(UbahDemonstrator.this, "Semua
field harus diisi", Toast.LENGTH_SHORT).show();
        }
    });
}

private void checkAndRequestPermission() {
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
        openGallery();
    } else {
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
PERMISSION_REQUEST_CODE);
    }
}

private void openGallery() {
    Intent intent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
    startActivityForResult(intent, REQUEST_CODE_PICK_IMAGE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_CODE_PICK_IMAGE && resultCode ==
RESULT_OK && data != null) {
        selectedImageUri = data.getData();
    }
}

```

```

        fotox.setImageURI(selectedImageUri);
    }
}
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
    if (requestCode == PERMISSION_REQUEST_CODE) {
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            openGallery();
        } else {
            Toast.makeText(this, "Izin akses galeri dibutuhkan
untuk memilih foto", Toast.LENGTH_SHORT).show();
        }
    }
}
private String getRealPathFromURI(Uri contentUri) {
    String[] projection = {MediaStore.Images.Media.DATA};
    Cursor cursor = getContentResolver().query(contentUri,
projection, null, null, null);
    if (cursor == null) {
        return contentUri.getPath();
    } else {
        int columnIndex =
cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
        cursor.moveToFirst();
        String filePath = cursor.getString(columnIndex);
        cursor.close();
        return filePath;
    }
}

private void tambahData(String imagePath, String
SNamaDemonstrator, String SProvinsi, String SKabupaten, String
SUsernameDemonstrator, String SPasswordDemonstrator) {
    File imageFile = new File(imagePath);
    RequestBody imageRequestBody =
RequestBody.create(MediaType.parse("image/*"), imageFile);
    MultipartBody.Part Foto =
MultipartBody.Part.createFormData("fotodemonstrator",
imageFile.getName(), imageRequestBody);
    RequestBody Nama =
RequestBody.create(MediaType.parse("text/plain"),
SNamaDemonstrator);
}

```

```

        RequestBody Provinsi =
RequestBody.create(MediaType.parse("text/plain"), SProvinsi);
        RequestBody Kabupaten =
RequestBody.create(MediaType.parse("text/plain"), SKabupaten);
        RequestBody Username =
RequestBody.create(MediaType.parse("text/plain"),
SUsernameDemonstrator);
        RequestBody Password =
RequestBody.create(MediaType.parse("text/plain"),
SPasswordDemonstrator);
        RequestBody Supervisor =
RequestBody.create(MediaType.parse("text/plain"),
namaSupervisorTerpilih);
        RequestBody KDSupervisor =
RequestBody.create(MediaType.parse("text/plain"),
kodeSupervisorTerpilih);
        RequestBody KDManager =
RequestBody.create(MediaType.parse("text/plain"),
kodeManagerTerpilih);
        RequestBody Level =
RequestBody.create(MediaType.parse("text/plain"), level);

        Log.d("MyTag", "onResponse: " + imagePath +
S>NamaDemonstrator + SProvinsi + SKabupaten + SUsernameDemonstrator +
SPasswordDemonstrator);

        InterfaceDemonstrator interfaceDemonstrator =
RetroServer.KonesiAPI().create(InterfaceDemonstrator.class);
        Call<ModelDemonstrator> simpan =
interfaceDemonstrator.updateDemonstrator(varId, Foto, Nama, Username, Pa
ssword, Supervisor, KDSupervisor, Kabupaten, KDManager, Provinsi, Level);
        simpan.enqueue(new Callback<ModelDemonstrator>() {
            @Override
            public void onResponse(Call<ModelDemonstrator> call,
Response<ModelDemonstrator> response) {
                Toast.makeText(UbahDemonstrator.this, "Data Berhasil
Disimpan", Toast.LENGTH_SHORT).show();
                finish();
            }

            @Override
            public void onFailure(Call<ModelDemonstrator> call,
Throwable t) {
                Toast.makeText(UbahDemonstrator.this, "Data Gagal
Disimpan", Toast.LENGTH_SHORT).show();
            }
        });
    });

```

```
}  
}
```

SourceCode 1. 35 Ubah Demonstrator

18) Ubah Distributor

```
package com.example.cakraagroindonesia.Activity.KelolaDataMaster;  
  
import android.Manifest;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.database.Cursor;  
import android.net.Uri;  
import android.os.Bundle;  
import android.provider.MediaStore;  
import android.util.Log;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.AdapterView.Adapter;  
import android.widget.AdapterView.OnItemClickListener;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.Spinner;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
import androidx.core.content.ContextCompat;  
  
import com.bumptech.glide.Glide;  
import com.example.cakraagroindonesia.API.RetroServer;  
import  
com.example.cakraagroindonesia.Interface.InterfaceDistributor;  
import com.example.cakraagroindonesia.Interface.InterfaceSecretary;  
import com.example.cakraagroindonesia.Model.ModelDistributor;  
import com.example.cakraagroindonesia.Model.ModelSecretary;  
import com.example.cakraagroindonesia.R;  
  
import java.io.File;  
import java.util.ArrayList;  
import java.util.List;  
  
import okhttp3.MediaType;  
import okhttp3.MultipartBody;
```

```

import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class UbahDistributor extends AppCompatActivity {
    private String
varId, varNama, varUsername, varPassword, varLevel, varKodesc, varNamasv, v
arTFoto, varFoto, varKodemg, varPerusahaan;

    private TextView btnsubmit, btnPilihFoto;
    private EditText
namadistributor, perusahaan, usernamedistributor, passworddistributor;
    private String>NamaDistributor, Perusahaan, Username, Password;

    private String namaSecretaryTerpilih, kodeSecretaryTerpilih,
kodeManagerTerpilih, level;

    private Spinner spinner;
    private ArrayAdapter spinnerAdapter;

    private ImageView foto;
    private Uri selectedImageUri;

    private static final int REQUEST_CODE_PICK_IMAGE = 101;
    private static final int PERMISSION_REQUEST_CODE = 102;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ubah_distributor);

        namadistributor = findViewById(R.id.namadistributor);
        perusahaan = findViewById(R.id.perusahaan);
        usernamedistributor =
findViewById(R.id.usernamedistributor);
        passworddistributor =
findViewById(R.id.passworddistributor);
        spinner = findViewById(R.id.spinner);
        foto = findViewById(R.id.foto);
        btnsubmit = findViewById(R.id.btnsubmit);
        btnPilihFoto = findViewById(R.id.btnPilihFoto);

        varId = getIntent().getStringExtra("xId");
        varNama = getIntent().getStringExtra("xNama");
        varUsername = getIntent().getStringExtra("xUsername");
        varPassword = getIntent().getStringExtra("xPassword");

```

```

varLevel = getIntent().getStringExtra("xLevel");
varKodesc = getIntent().getStringExtra("xKodesc");
varNamasv = getIntent().getStringExtra("xNamasv");
varTFoto = getIntent().getStringExtra("xTFoto");
varFoto = getIntent().getStringExtra("xFoto");
varKodemg = getIntent().getStringExtra("xKodemg");
varPerusahaan = getIntent().getStringExtra("xPerusahaan");

Glide.with(this).load(varTFoto).into(foto);
namadistributor.setText(varNama);
perusahaan.setText(varPerusahaan);
usernameistributor.setText(varUsername);
passworddistributor.setText(varPassword);

InterfaceSecretary interfaceSecretary =
RetroServer.KonesiAPI().create(InterfaceSecretary.class);
Call<ModelSecretary> Spinner =
interfaceSecretary.getSecretary();
Spinner.enqueue(new Callback<ModelSecretary>() {
    @Override
    public void onResponse(Call<ModelSecretary> call,
Response<ModelSecretary> response) {
        ModelSecretary modelSecretary = response.body();
        ArrayList<ModelSecretary.secretary> secretaryList =
modelSecretary.getSecretary();

        List<String> namaSecretary = new ArrayList<>();
        for (ModelSecretary.secretary nama : secretaryList){
            namaSecretary.add(nama.getNama_secretary());
        }
        List<String> kodeSecretary = new ArrayList<>();
        for (ModelSecretary.secretary kode : secretaryList){
            kodeSecretary.add(kode.getKode_sc());
        }
        List<String> kodeManager = new ArrayList<>();
        for (ModelSecretary.secretary kodeMg :
secretaryList){
            kodeManager.add(kodeMg.getKode_mg());
        }

        spinnerAdapter = new
ArrayAdapter<>(UbahDistributor.this,
android.R.layout.simple_spinner_item, namaSecretary);
        spinnerAdapter.setDropDownViewResource(android.R.lay
out.simple_spinner_dropdown_item);
        spinner.setAdapter(spinnerAdapter);

```

```

        spinner.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?>
adapterView, View view, int i, long l) {
                namaSecretaryTerpilih =
namaSecretary.get(i);
                kodeSecretaryTerpilih =
kodeSecretary.get(i);
                kodeManagerTerpilih = kodeManager.get(i);
            }
            @Override
            public void onNothingSelected(AdapterView<?>
adapterView) {

            }
        });
    }

    @Override
    public void onFailure(Call<ModelSecretary> call,
Throwable t) {

    }
});

//      InterfaceDistributor interfaceDistributor =
RetroServer.KonesiAPI().create(InterfaceDistributor.class);
//      Call<ModelDistributor> newid =
interfaceDistributor.getDistributor();
//      newid.enqueue(new Callback<ModelDistributor>() {
//          @Override
//          public void onResponse(Call<ModelDistributor> call,
Response<ModelDistributor> response) {
//              ArrayList<ModelDistributor.distributor> id =
response.body().getDistributor();
//              String lastId = new
DataDistributor(UbahDistributor.this, id).getLastId();
//              String numericPart = lastId.substring(2);
//              int lastNumber = Integer.parseInt(numericPart);
//              int newNumber = lastNumber + 1;
//
//              newId = "DT" + String.format("%03d", newNumber);
//          }
//          @Override
//          public void onFailure(Call<ModelDistributor> call,
Throwable t) {

```

```

//
//     }
//     });

    btnPilihFoto.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            checkAndRequestPermission();
        }
    });

    btnsubmit.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            level = "distributor";
            NamaDistributor =
namadistributor.getText().toString().trim();
            Perusahaan = perusahaan.getText().toString().trim();
            Username =
usernameistributor.getText().toString().trim();
            Password =
passworddistributor.getText().toString().trim();

            if(!NamaDistributor.isEmpty() &&
!Perusahaan.isEmpty() && !Username.isEmpty() &&
!Password.isEmpty()){
                if (selectedImageUri != null){
                    String imagePath =
getRealPathFromURI(selectedImageUri);

                    tambahData(NamaDistributor,Username,Password
,level,kodeSecretaryTerpilih,namaSecretaryTerpilih,imagePath,kodeMan
agerTerpilih,Perusahaan);
                }else{
                    Toast.makeText(UbahDistributor.this, "Pilih
gambar terlebih dahulu", Toast.LENGTH_SHORT).show();
                }
            } else {
                Toast.makeText(UbahDistributor.this, "Semua
field harus diisi", Toast.LENGTH_SHORT).show();
            }
        }
    });
}

private void checkAndRequestPermission() {

```



```

        if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
            openGallery();
        } else {
            ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
PERMISSION_REQUEST_CODE);
        }
    }
    private void openGallery() {
        Intent intent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
        startActivityForResult(intent, REQUEST_CODE_PICK_IMAGE);
    }
    @Override
    protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == REQUEST_CODE_PICK_IMAGE && resultCode ==
RESULT_OK && data != null) {
            selectedImageUri = data.getData();
            foto.setImageURI(selectedImageUri);
        }
    }
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        if (requestCode == PERMISSION_REQUEST_CODE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                openGallery();
            } else {
                Toast.makeText(this, "Izin akses galeri dibutuhkan
untuk memilih foto", Toast.LENGTH_SHORT).show();
            }
        }
    }
    private String getRealPathFromURI(Uri contentUri) {
        String[] projection = {MediaStore.Images.Media.DATA};
        Cursor cursor = getContentResolver().query(contentUri,
projection, null, null, null);
        if (cursor == null) {
            return contentUri.getPath();
        } else {

```

```

        int columnIndex =
cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
        cursor.moveToFirst();
        String filePath = cursor.getString(columnIndex);
        cursor.close();
        return filePath;
    }
}

private void tambahData(String SNamaDistributor, String
SUsername, String SPassword, String Slevel, String
SkodeSecretaryTerpilih, String SnamaSecretaryTerpilih, String
imagePath, String SkodeManagerTerpilih, String SPerusahaan) {
    File imageFile = new File(imagePath);
    RequestBody imageRequestBody =
RequestBody.create(MediaType.parse("image/*"), imageFile);
    MultipartBody.Part Foto =
MultipartBody.Part.createFormData("fotodistributor",
imageFile.getName(), imageRequestBody);
    RequestBody Nama =
RequestBody.create(MediaType.parse("text/plain"), SNamaDistributor);
    RequestBody Username =
RequestBody.create(MediaType.parse("text/plain"), SUsername);
    RequestBody Password =
RequestBody.create(MediaType.parse("text/plain"), SPassword);
    RequestBody Level =
RequestBody.create(MediaType.parse("text/plain"), Slevel);
    RequestBody KodeSC =
RequestBody.create(MediaType.parse("text/plain"),
SkodeSecretaryTerpilih);
    RequestBody NamaSC =
RequestBody.create(MediaType.parse("text/plain"),
SnamaSecretaryTerpilih);
    RequestBody KodeMG =
RequestBody.create(MediaType.parse("text/plain"),
SkodeManagerTerpilih);
    RequestBody Perusahaan =
RequestBody.create(MediaType.parse("text/plain"), SPerusahaan);

    Log.d("MyTag", "onResponse: " + imageRequestBody +
SNamaDistributor + SPerusahaan + SUsername + SPassword +
SnamaSecretaryTerpilih + SkodeSecretaryTerpilih + Slevel +
SkodeManagerTerpilih);

    InterfaceDistributor interfaceDistributor =
RetroServer.KonesiAPI().create(InterfaceDistributor.class);

```

```

        Call<ModelDistributor> simpan =
interfaceDistributor.updateDistributor(varId,Foto>Nama,Username,Pass
word,Level,KodeSC>NamaSC,Perusahaan,KodeMG);
        simpan.enqueue(new Callback<ModelDistributor>() {
            @Override
            public void onResponse(Call<ModelDistributor> call,
Response<ModelDistributor> response) {
                Toast.makeText(UbahDistributor.this, "Data Berhasil
Disimpan", Toast.LENGTH_SHORT).show();
                finish();
            }

            @Override
            public void onFailure(Call<ModelDistributor> call,
Throwable t) {
                Toast.makeText(UbahDistributor.this, "Data Gagal
Disimpan", Toast.LENGTH_SHORT).show();
            }
        });
    }
}

```

SourceCode 1. 36 Ubah Distributor

19) Ubah Manager

```

package com.example.cakraagroindonesia.Activity.KelolaDataMaster;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

```

```

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Interface.InterfaceManager;
import com.example.cakraagroindonesia.Model.ModelManager;
import com.example.cakraagroindonesia.R;

import java.io.File;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class UbahManager extends AppCompatActivity {

    private String
varId, varNama, varUsername, varPassword, varTFoto, varFoto;

    private EditText namamanager, usernamemanager, passwordmanager;
    private String NamaManager, UsernameManager,
PasswordManager, level, newId;
    private TextView btnPilihFoto, btnsubmit;

    private ImageView foto;

    private Uri selectedImageUri;

    private static final int REQUEST_CODE_PICK_IMAGE = 101;
    private static final int PERMISSION_REQUEST_CODE = 102;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ubah_manager);

        namamanager = findViewById(R.id.namamanager);
        usernamemanager = findViewById(R.id.usernamemanager);
        passwordmanager = findViewById(R.id.passwordmanager);
        btnPilihFoto = findViewById(R.id.btnPilihFoto);
        btnsubmit = findViewById(R.id.btnsubmit);
        foto = findViewById(R.id.foto);

        varId = getIntent().getStringExtra("xId");

```

```

varNama = getIntent().getStringExtra("xNama");
varUsername = getIntent().getStringExtra("xUsername");
varPassword = getIntent().getStringExtra("xPassword");
varTFoto = getIntent().getStringExtra("xTFoto");
varFoto= getIntent().getStringExtra("xFoto");

namamanager.setText(varNama);
usernamemanager.setText(varUsername);
passwordmanager.setText(varPassword);
Glide.with(this).load(varTFoto).into(foto);

btnPilihFoto.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        checkAndRequestPermission();
    }
});

btnsubmit.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
       >NamaManager =
namamanager.getText().toString().trim();
        UsernameManager =
usernamemanager.getText().toString().trim();
        PasswordManager =
passwordmanager.getText().toString().trim();
        level = "manager";

        if (!NamaManager.isEmpty() &&
!UsernameManager.isEmpty() && !PasswordManager.isEmpty()) {
            if (selectedImageUri != null) {
                String imagePath =
getRealPathFromURI(selectedImageUri);

                tambahData(imagePath,>NamaManager,
UsernameManager, PasswordManager);
            } else {
                Toast.makeText(UbahManager.this, "Pilih
gambar terlebih dahulu", Toast.LENGTH_SHORT).show();
            }
        } else {
            Toast.makeText(UbahManager.this, "Semua field
harus diisi", Toast.LENGTH_SHORT).show();
        }
    }
});

```

```

    }
    private void checkAndRequestPermission() {
        if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
            openGallery();
        } else {
            ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
PERMISSION_REQUEST_CODE);
        }
    }

    private void openGallery() {
        Intent intent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
        startActivityForResult(intent, REQUEST_CODE_PICK_IMAGE);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == REQUEST_CODE_PICK_IMAGE && resultCode ==
RESULT_OK && data != null) {
            selectedImageUri = data.getData();
            foto.setImageURI(selectedImageUri);
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        if (requestCode == PERMISSION_REQUEST_CODE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                openGallery();
            } else {
                Toast.makeText(this, "Izin akses galeri dibutuhkan
untuk memilih foto", Toast.LENGTH_SHORT).show();
            }
        }
    }
}

```

```

private String getRealPathFromURI(Uri contentUri) {
    String[] projection = {MediaStore.Images.Media.DATA};
    Cursor cursor = getContentResolver().query(contentUri,
projection, null, null, null);
    if (cursor == null) {
        return contentUri.getPath();
    } else {
        int columnIndex =
cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
        cursor.moveToFirst();
        String filePath = cursor.getString(columnIndex);
        cursor.close();
        return filePath;
    }
}

private void tambahData(String imagePath, String SNamaManager,
String SUsernameManager,String SPasswOrdManager) {
    File imageFile = new File(imagePath);
    RequestBody imageRequestBody =
RequestBody.create(MediaType.parse("image/*"), imageFile);
    MultipartBody.Part Foto =
MultipartBody.Part.createFormData("fotomanager",
imageFile.getName(), imageRequestBody);
    RequestBody Nama =
RequestBody.create(MediaType.parse("text/plain"), SNamaManager);
    RequestBody Username =
RequestBody.create(MediaType.parse("text/plain"), SUsernameManager);
    RequestBody Password =
RequestBody.create(MediaType.parse("text/plain"), SPasswOrdManager);
    RequestBody Level =
RequestBody.create(MediaType.parse("text/plain"), level);

    Log.d("MyTag", "onResponse: " + imagePath + SNamaManager +
SUsernameManager + SPasswOrdManager + newId + level);

    InterfaceManager interfaceManager =
RetroServer.KonesiAPI().create(InterfaceManager.class);
    Call<ModelManager> simpan =
interfaceManager.updateManager(varId,Foto,Nama,Username,PasswOrd,Level);
    simpan.enqueue(new Callback<ModelManager>() {
        @Override
        public void onResponse(Call<ModelManager> call,
Response<ModelManager> response) {
            Toast.makeText(UbahManager.this, "Data Berhasil
Disimpan", Toast.LENGTH_SHORT).show();

```

```

        finish();
    }

    @Override
    public void onFailure(Call<ModelManager> call, Throwable
t) {
        Toast.makeText(UbahManager.this, "Data Gagal
Disimpan", Toast.LENGTH_SHORT).show();
    }
});
}
}

```

SourceCode 1. 37 Ubah Manager

e) Log

1) Log

```

package com.example.cakraagroindonesia.Activity.Log;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.R;

public class Log extends AppCompatActivity {
    TextView btnlogsuperadmin, btnlogadmin, btnlogmaster,
btnloguser;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_log);
        deklarasi();

        btnlogsuperadmin.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(Log.this,
LogSuperAdmin.class));
            }
        });
    }
}

```



```

    });

    btnlogadmin.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            startActivity(new Intent(Log.this, LogAdmin.class));
        }
    });

    btnlogmaster.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            startActivity(new Intent(Log.this,
LogMaster.class));
        }
    });

    btnloguser.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            startActivity(new Intent(Log.this, LogUser.class));
        }
    });
}

private void deklarasi() {
    btnlogsuperadmin = findViewById(R.id.btnlogsuperadmin);
    btnlogadmin = findViewById(R.id.btnlogadmin);
    btnlogmaster = findViewById(R.id.btnlogmaster);
    btnloguser = findViewById(R.id.btnloguser);
}
}
}

```

SourceCode 1. 38 Log

2) Log Admin

```

package com.example.cakraagroindonesia.Activity.Log;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.R;

public class LogAdmin extends AppCompatActivity {

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_log_admin);
}
}

```

SourceCode 1. 39 Log Admin

3) Log Master

```

package com.example.cakraagroindonesia.Activity.Log;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.R;

public class LogMaster extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_log_master);
    }
}

```

SourceCode 1. 40 Log Master

4) Log Super Admin

```

package com.example.cakraagroindonesia.Activity.Log;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.R;

public class LogSuperAdmin extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_log_super_admin);
    }
}

```

```
}
```

SourceCode 1. 41 Log Super Admin

5) Log User

```
package com.example.cakraagroindonesia.Activity.Log;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.R;

public class LogUser extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_log_user);
    }
}
```

SourceCode 1. 42 Log User

f) Notifikasi

```
package com.example.cakraagroindonesia.Activity.Notifikasi;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Adapter.DataNotifikasi;
import com.example.cakraagroindonesia.Interface.InterfaceNotifikasi;
import com.example.cakraagroindonesia.Model.ModelNotifikasi;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;
```

```

public class Notifikasi extends AppCompatActivity {

    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_notifikasi);
    }

    @Override
    protected void onResume() {
        super.onResume();
        recyclerView = findViewById(R.id.rv_notifikasi);
        layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL, false);
        recyclerView.setLayoutManager(layoutManager);

        InterfaceNotifikasi interfaceNotifikasi =
RetroServer.KonesiAPI().create(InterfaceNotifikasi.class);
        Call<ModelNotifikasi> tampil =
interfaceNotifikasi.getNotifikasi();
        tampil.enqueue(new Callback<ModelNotifikasi>() {
            @Override
            public void onResponse(Call<ModelNotifikasi> call,
Response<ModelNotifikasi> response) {
                ArrayList<ModelNotifikasi.notifikasi> listData =
response.body().getNotifikasi();
                adapter = new DataNotifikasi(Notifikasi.this,
listData);
                recyclerView.setAdapter(adapter);
                adapter.notifyDataSetChanged();
            }

            @Override
            public void onFailure(Call<ModelNotifikasi> call,
Throwable t) {}
        });
    }
}

```

SourceCode 1. 43 Notifikasi

g) Pertanyaan

1) Kelola FAQ

```
package com.example.cakraagroindonesia.Activity.Pertanyaan;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Adapter.DataKelolaFaq;
import com.example.cakraagroindonesia.Interface.InterfaceFaq;
import com.example.cakraagroindonesia.Model.ModelFaq;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;
import java.util.Collections;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class KelolaFAQ extends AppCompatActivity {
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;
    TextView btnTambah;

    int Id;

    private float currentRotation = 0;

    private static final int VIEW_TYPE_ITEM = 0;
    private static final int VIEW_TYPE_ITEM_LAST = 1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_data_faq);

        btnTambah = findViewById(R.id.btntambah);
        btnTambah.setOnClickListener(new View.OnClickListener() {
```

```

        @Override
        public void onClick(View view) {
            startActivity(new Intent(KelolaFAQ.this,
TambahFAQ.class));
        }
    });

    recyclerView = findViewById(R.id.rv_kelolafaq);
    layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL, false);
    recyclerView.setLayoutManager(layoutManager);
}

@Override
protected void onResume() {
    super.onResume();

    InterfaceFaq interfaceFaq =
RetroServer.KonesiAPI().create(InterfaceFaq.class);
    Call<ModelFaq> tampil = interfaceFaq.getFaq();
    tampil.enqueue(new Callback<ModelFaq>() {
        @Override
        public void onResponse(Call<ModelFaq> call,
Response<ModelFaq> response) {
            ArrayList<ModelFaq.data_faq> listData =
response.body().getData_faq();
            adapter = new DataKelolaFaq(KelolaFAQ.this,
listData);

            Collections.reverse(listData);
            recyclerView.setAdapter(adapter);
            adapter.notifyDataSetChanged();
        }
        @Override
        public void onFailure(Call<ModelFaq> call, Throwable t)
{

        }
    });
}
}
}

```

SourceCode 1. 44 Kelola FAQ

2) Kelola Pertanyaan

```
package com.example.cakraagroindonesia.Activity.Pertanyaan;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.R;

public class KelolaPertanyaan extends AppCompatActivity {
    TextView btnDataQNA, btnDataFAQ;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_kelola_pertanyaan);
        btnDataQNA = findViewById(R.id.btnDataQNA);
        btnDataFAQ = findViewById(R.id.btnDataFAQ);

        btnDataQNA.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaPertanyaan.this,
KelolaQNA.class));
            }
        });
        btnDataFAQ.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaPertanyaan.this,
KelolaFAQ.class));
            }
        });
    }
}
```

SourceCode 1. 45 Kelola Pertanyaan

3) Kelola QnA

```
package com.example.cakraagroindonesia.Activity.Pertanyaan;

import android.os.Bundle;
```

```

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Adapter.DataKelolaQna;
import com.example.cakraagroindonesia.Interface.InterfaceQna;
import com.example.cakraagroindonesia.Model.ModelQna;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;
import java.util.Collections;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class KelolaQNA extends AppCompatActivity {
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_data_qna);
    }

    @Override
    protected void onResume() {
        super.onResume();
        recyclerView = findViewById(R.id.rv_kelolaqna);
        layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL, false);
        recyclerView.setLayoutManager(layoutManager);

        InterfaceQna interfaceQna =
RetroServer.KonesiAPI().create(InterfaceQna.class);
        Call<ModelQna> tampil = interfaceQna.getQna();
        tampil.enqueue(new Callback<ModelQna>() {
            @Override
            public void onResponse(Call<ModelQna> call,
Response<ModelQna> response) {
                ArrayList<ModelQna.data_qna> listData =
response.body().getData_qna();
                adapter = new DataKelolaQna(KelolaQNA.this,
listData);

```



```

        Collections.reverse(listData);
        recyclerView.setAdapter(adapter);
        adapter.notifyDataSetChanged();
    }

    @Override
    public void onFailure(Call<ModelQna> call, Throwable t)
{
    }
});
}
}

```

SourceCode 1. 46 Kelola QnA

4) Tambah FAQ

```

package com.example.cakraagroindonesia.Activity.Pertanyaan;

import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.API.RetrofitServer;
import com.example.cakraagroindonesia.Interface.InterfaceFaq;
import com.example.cakraagroindonesia.Model.ModelFaq;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class TambahFAQ extends AppCompatActivity {
    private TextView btnSubmit;
    private EditText pertanyaanFaq, jawabanFaq;
    private String pertanyaan, jawaban;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}

```

```

setContentView(R.layout.activity_tambah_data_faq);

btnSubmit = findViewById(R.id.btnsubmit);
pertanyaanFaq = findViewById(R.id.pertanyaan);
jawabanFaq = findViewById(R.id.jawaban);

btnSubmit.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        pertanyaan = pertanyaanFaq.getText().toString();
        jawaban = jawabanFaq.getText().toString();

        tambahData();
    }
});
}

private void tambahData(){
    ModelFaq.data_faq newData = new ModelFaq.data_faq();
    newData.setPertanyaan(pertanyaan);
    newData.setJawaban(jawaban);

    ArrayList<ModelFaq.data_faq> listData = new ArrayList<>();
    listData.add(newData);

    ModelFaq modelFaq = new ModelFaq();
    modelFaq.setData_faq(listData);

    InterfaceFaq interfaceFaq =
RetroServer.KonesiAPI().create(InterfaceFaq.class);
    Call<ModelFaq> simpan =
interfaceFaq.setFaq(pertanyaan,jawaban);
    simpan.enqueue(new Callback<ModelFaq>() {
        @Override
        public void onResponse(Call<ModelFaq> call,
Response<ModelFaq> response) {
            Toast.makeText(TambahFAQ.this, "Data Gagal
Disimpan", Toast.LENGTH_SHORT).show();
        }

        @Override
        public void onFailure(Call<ModelFaq> call, Throwable t)
{
            Toast.makeText(TambahFAQ.this, "Data Berhasil
Disimpan", Toast.LENGTH_SHORT).show();
            finish();
        }
    }
}

```

```

    });
}
}

```

SourceCode 1. 47 Tambah FAQ

5) Tambah QnA

```

package com.example.cakraagroindonesia.Activity.Pertanyaan;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.Nullable;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Interface.InterfaceQna;
import com.example.cakraagroindonesia.Model.ModelQna;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class TambahQna extends Activity {
    private TextView btnKirim;
    private EditText Nama, Notelp, Pertanyaan;
    private String nama, notelp, pertanyaan;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.popup_tambahpertanyaan);

        btnKirim = findViewById(R.id.btnkirimpertanyaan);
        Nama = findViewById(R.id.namapenanya);
        Notelp = findViewById(R.id.notelppenanya);
        Pertanyaan = findViewById(R.id.pertanyaanpenanya);

        btnKirim.setOnClickListener(new View.OnClickListener() {
            @Override

```

```

        public void onClick(View view) {
            nama = Nama.getText().toString();
            notelp = Notelp.getText().toString();
            pertanyaan = Pertanyaan.getText().toString();

            tambahData();
        }
    });
}

private void tambahData(){
    ModelQna.data_qna newData = new ModelQna.data_qna();
    newData.setNama(nama);
    newData.setTelepon(notelp);
    newData.setPertanyaan(pertanyaan);

    ArrayList<ModelQna.data_qna> listData = new ArrayList<>();
    listData.add(newData);

    ModelQna modelQna = new ModelQna();
    modelQna.setData_qna(listData);

    InterfaceQna interfaceQna =
RetroServer.KonesiAPI().create(InterfaceQna.class);
    Call<ModelQna> simpan =
interfaceQna.setQna(nama,notelp,pertanyaan);
    simpan.enqueue(new Callback<ModelQna>() {
        @Override
        public void onResponse(Call<ModelQna> call,
Response<ModelQna> response) {
            Toast.makeText(TambahQna.this, "Data Berhasil
Disimpan", Toast.LENGTH_SHORT).show();
            finish();
        }

        @Override
        public void onFailure(Call<ModelQna> call, Throwable t)
{
            Toast.makeText(TambahQna.this, "Data Gagal
Disimpan", Toast.LENGTH_SHORT).show();
            finish();
        }
    });
}
}
}

```

SourceCode 1. 48 Tambah QnA

6) Ubah FAQ

```
package com.example.cakraagroindonesia.Activity.Pertanyaan;

import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Interface.InterfaceFaq;
import com.example.cakraagroindonesia.Model.ModelFaq;
import com.example.cakraagroindonesia.R;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class UbahFAQ extends AppCompatActivity {

    private int varId;
    private String varPertanyaan, varJawaban;

    private TextView btnSubmit;
    private EditText pertanyaanFaq, jawabanFaq;
    private String pertanyaan, jawaban;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ubah_faq);

        btnSubmit = findViewById(R.id.btnsubmit);
        pertanyaanFaq = findViewById(R.id.pertanyaan);
        jawabanFaq = findViewById(R.id.jawaban);

        varId = getIntent().getIntExtra("xId", -1);
        varPertanyaan = getIntent().getStringExtra("xPertanyaan");
        varJawaban = getIntent().getStringExtra("xJawaban");

        btnSubmit.setOnClickListener(new View.OnClickListener() {
            @Override
```

```

        public void onClick(View view) {
            pertanyaan = pertanyaanFaq.getText().toString();
            jawaban = jawabanFaq.getText().toString();

            InterfaceFaq interfaceFaq =
RetroServer.KonesiAPI().create(InterfaceFaq.class);
            Call<ModelFaq> call = interfaceFaq.updateFaq(varId,
pertanyaan, jawaban);
            call.enqueue(new Callback<ModelFaq>() {
                @Override
                public void onResponse(Call<ModelFaq> call,
Response<ModelFaq> response) {
                    Toast.makeText(UbahFAQ.this, "Berhasil Ubah
Data", Toast.LENGTH_SHORT).show();
                    finish();
                }

                @Override
                public void onFailure(Call<ModelFaq> call,
Throwable t) {
                    Toast.makeText(UbahFAQ.this, "Gagal Ubah
Data", Toast.LENGTH_SHORT).show();
                }
            });
        }
    }
}

```

SourceCode 1. 49 Ubah FAQ

7) Ubah QnA

```

package com.example.cakraagroindonesia.Activity.Pertanyaan;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Interface.InterfaceQna;
import com.example.cakraagroindonesia.Model.ModelQna;
import com.example.cakraagroindonesia.R;

```

```

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class UbahQNA extends Activity {

    private int varId;
    private String varNama, varNotelp, varPertanyaan;

    private ImageView fotox;
    private TextView btnKirim, btnPilihFoto;
    private EditText Nama, Notelp, Pertanyaan;
    private String nama, notelp, pertanyaan;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ubah_qna);

        btnKirim = findViewById(R.id.btnkirimpertanyaan);
        Nama = findViewById(R.id.namapenanya);
        Notelp = findViewById(R.id.notelppenanya);
        Pertanyaan = findViewById(R.id.pertanyaanpenanya);
        btnPilihFoto = findViewById(R.id.btnPilihFoto);
        fotox = findViewById(R.id.fotox);

        varId = getIntent().getIntExtra("xId", -1);
        varNama = getIntent().getStringExtra("xNama");
        varNotelp = getIntent().getStringExtra("xNoTelp");
        varPertanyaan = getIntent().getStringExtra("xPertanyaan");

        Nama.setText(varNama);
        Notelp.setText(varNotelp);
        Pertanyaan.setText(varPertanyaan);

        btnKirim.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                nama = Nama.getText().toString().trim();
                notelp = Notelp.getText().toString().trim();
                pertanyaan = Pertanyaan.getText().toString().trim();

                if (!nama.isEmpty() && !notelp.isEmpty() &&
!pertanyaan.isEmpty()) {
                    uploadData();
                }
            }
        });
    }
}

```

```

        } else {
            Toast.makeText(UbahQNA.this, "Semua field harus
diisi", Toast.LENGTH_SHORT).show();
        }

    }

});

}

private void uploadData() {
    InterfaceQna interfaceQna =
RetroServer.KonesiAPI().create(InterfaceQna.class);
    Call<ModelQna> ubah = interfaceQna.updateQna(varId, nama,
notelp, pertanyaan);
    ubah.enqueue(new Callback<ModelQna>() {
        @Override
        public void onResponse(Call<ModelQna> call,
Response<ModelQna> response) {
            Toast.makeText(UbahQNA.this, "Berhasil Edit Data",
Toast.LENGTH_SHORT).show();
            finish();
        }
        @Override
        public void onFailure(Call<ModelQna> call, Throwable t)
{
            Toast.makeText(UbahQNA.this, "Gagak Edit Data",
Toast.LENGTH_SHORT).show();
        }
    });
}
}
}

```

SourceCode 1. 50 Ubah QnA

h) Produk

1) Kelola Produk

```

package com.example.cakraagroindonesia.Activity.Produk;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;

```



```

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Adapter.DataKelolaFungisida;
import com.example.cakraagroindonesia.Adapter.DataKelolaHerbisida;
import com.example.cakraagroindonesia.Adapter.DataKelolaInsektisida;
import com.example.cakraagroindonesia.Adapter.DataKelolaPgr;
import com.example.cakraagroindonesia.Interface.InterfaceFungisida;
import com.example.cakraagroindonesia.Interface.InterfaceHerbisida;
import
com.example.cakraagroindonesia.Interface.InterfaceInsektisida;
import com.example.cakraagroindonesia.Interface.InterfacePgr;
import com.example.cakraagroindonesia.Model.ModelFungisida;
import com.example.cakraagroindonesia.Model.ModelHerbisida;
import com.example.cakraagroindonesia.Model.ModelInsektisida;
import com.example.cakraagroindonesia.Model.ModelPgr;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class KelolaProduk extends AppCompatActivity{
    TextView
filterInsetisida,filterHerbisida,filterFungisida,filterPgr,btnTambah
Produk;
    LinearLayout detail;

    RelativeLayout hide;

    private TextView lastSelectedTextView;
    private float currentRotation = 0;
    private ImageView btndetail;

    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_kelola_produk);

```

```

        filterFungisida = findViewById(R.id.filterfungisida);
        filterHerbisida = findViewById(R.id.filterherbisida);
        filterInsetisida = findViewById(R.id.filterinsektisida);
        filterPgr = findViewById(R.id.filterpgr);
        btnTambahProduk = findViewById(R.id.btntambah);
        hide = findViewById(R.id.hide);
        detail = findViewById(R.id.detail);
        btndetail = findViewById(R.id.btndetail);

        btnTambahProduk.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaProduk.this,
TambahProduk.class));
            }
        });
        hide.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                rotateImage();
                int invisible = detail.getVisibility();
                if(invisible == View.VISIBLE){
                    detail.setVisibility(View.GONE);
                }else {
                    detail.setVisibility(View.VISIBLE);
                }
            }
        });

    }
    private void rotateImage() {
        currentRotation += 180;
        btndetail.setRotation(currentRotation);
    }

    private void setBoldText(TextView textView) {
        if (lastSelectedTextView != null) {
            lastSelectedTextView.setTypeface(null,
android.graphics.Typeface.NORMAL);
        }

        textView.setTypeface(null, android.graphics.Typeface.BOLD);
        lastSelectedTextView = textView;
    }
}

```

```

@Override
protected void onResume() {
    super.onResume();
    filterInsetisida.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        setBoldText((TextView) view);
        recyclerView = findViewById(R.id.rv_kelolaproduk);
        layoutManager = new
LinearLayoutManager(KelolaProduk.this, LinearLayoutManager.VERTICAL,
false);
        recyclerView.setLayoutManager(layoutManager);

        InterfaceInsektisida interfaceInsektisida =
RetroServer.KonesiAPI().create(InterfaceInsektisida.class);
        Call<ModelInsektisida> tampil =
interfaceInsektisida.getDataInsektisida();
        tampil.enqueue(new Callback<ModelInsektisida>() {
            @Override
            public void onResponse(Call<ModelInsektisida>
call, Response<ModelInsektisida> response) {
                ArrayList<ModelInsektisida.data_insektisida>
listData = response.body().getData_insektisida();
                adapter = new
DataKelolaInsektisida(KelolaProduk.this, listData);
                recyclerView.setAdapter(adapter);
                adapter.notifyDataSetChanged();
            }

            @Override
            public void onFailure(Call<ModelInsektisida>
call, Throwable t) {

            }
        });
    }
});

    filterHerbisida.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        setBoldText((TextView) view);
        recyclerView = findViewById(R.id.rv_kelolaproduk);

```

```

        layoutManager = new
LinearLayoutManager(KelolaProduk.this, LinearLayoutManager.VERTICAL,
false);
        recyclerView.setLayoutManager(layoutManager);

        InterfaceHerbisida interfaceHerbisida =
RetroServer.KonesiAPI().create(InterfaceHerbisida.class);
        Call<ModelHerbisida> tampil =
interfaceHerbisida.getDataHerbisida();
        tampil.enqueue(new Callback<ModelHerbisida>() {
            @Override
            public void onResponse(Call<ModelHerbisida>
call, Response<ModelHerbisida> response) {
                ArrayList<ModelHerbisida.data_herbisida>
listData = response.body().getData_herbisida();
                adapter = new
DataKelolaHerbisida(KelolaProduk.this, listData);
                recyclerView.setAdapter(adapter);
                adapter.notifyDataSetChanged();
            }

            @Override
            public void onFailure(Call<ModelHerbisida> call,
Throwable t) {

            }
        });
    });
}

filterFungisida.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        setBoldText((TextView) view);
        recyclerView = findViewById(R.id.rv_kelolaproduk);
        layoutManager = new
LinearLayoutManager(KelolaProduk.this, LinearLayoutManager.VERTICAL,
false);
        recyclerView.setLayoutManager(layoutManager);

        InterfaceFungisida interfaceFungisida =
RetroServer.KonesiAPI().create(InterfaceFungisida.class);
        Call<ModelFungisida> tampil =
interfaceFungisida.getDataFungisida();
        tampil.enqueue(new Callback<ModelFungisida>() {
            @Override

```

```

        public void onResponse(Call<ModelFungsida>
call, Response<ModelFungsida> response) {
            ArrayList<ModelFungsida.data_fungsida>
listData = response.body().getData_fungsida();
            adapter = new
DataKelolaFungsida(KelolaProduk.this, listData);
            recyclerView.setAdapter(adapter);
            adapter.notifyDataSetChanged();
        }

        @Override
        public void onFailure(Call<ModelFungsida> call,
Throwable t) {
        }
    });
}
});

filterPgr.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        setBoldText((TextView) view);
        recyclerView = findViewById(R.id.rv_kelolaproduk);
        layoutManager = new
LinearLayoutManager(KelolaProduk.this, LinearLayoutManager.VERTICAL,
false);
        recyclerView.setLayoutManager(layoutManager);

        InterfacePgr interfacePgr =
RetroServer.KonesiAPI().create(InterfacePgr.class);
        Call<ModelPgr> tampil = interfacePgr.getDataPgr();
        tampil.enqueue(new Callback<ModelPgr>() {
            @Override
            public void onResponse(Call<ModelPgr> call,
Response<ModelPgr> response) {
                ArrayList<ModelPgr.data_pgr> listData =
response.body().getData_pgr();
                adapter = new
DataKelolaPgr(KelolaProduk.this, listData);
                recyclerView.setAdapter(adapter);
                adapter.notifyDataSetChanged();
            }

            @Override
            public void onFailure(Call<ModelPgr> call,
Throwable t) {

```



```

import com.example.cakraagroindonesia.Model.ModelPgr;
import com.example.cakraagroindonesia.R;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.IOException;
import java.io.InputStream;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class TambahProduk extends AppCompatActivity implements
AdapterView.OnItemClickListener {

    private Spinner spinner;
    private EditText namaproduk, deskripsiproduk;
    private TextView btntambah, uploadfoto;
    private ImageView tampilfoto;
    private String JenisProduk, NamaProduk, DeskripsiProduk,
selectedItem;

    private Uri selectedImageUri;

    private static final int REQUEST_CODE_PICK_IMAGE = 101;
    private static final int PERMISSION_REQUEST_CODE = 102;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tambah_produk);

        spinner = findViewById(R.id.spinner);
        namaproduk = findViewById(R.id.namaproduk);
        deskripsiproduk = findViewById(R.id.deskripsiproduk);
        tampilfoto = findViewById(R.id.tampilfoto);
        btntambah = findViewById(R.id.btntambah);
        uploadfoto = findViewById(R.id.uploadfoto);

        // Buat adapter untuk spinner menggunakan string array
        ArrayAdapter<CharSequence> adapter =
ArrayAdapter.createFromResource(
            this,
            R.array.spinner_items,

```

```

        android.R.layout.simple_spinner_item
    );
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    spinner.setAdapter(adapter);

    spinner.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parentView,
View selectedItemView, int position, long id) {
            selectedItem =
parentView.getItemAtPosition(position).toString();

            Toast.makeText(TambahProduk.this, selectedItem,
Toast.LENGTH_SHORT).show();
        }

        @Override
        public void onNothingSelected(AdapterView<?> parentView)
{
            // Dilakukan ketika tidak ada item yang dipilih
        }
    });

    uploadfoto.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            checkAndRequestPermission();
        }
    });

    btntambah.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            NamaProduk = namaproduk.getText().toString().trim();
            DeskripsiProduk =
deskripsiproduk.getText().toString().trim();

            if (!NamaProduk.isEmpty() &&
!DeskripsiProduk.isEmpty()){
                File imagePath = new
File(getRealPathFromURI(selectedImageUri));

                tambahData(imagePath, NamaProduk,
DeskripsiProduk);
            }else{

```



```

        Toast.makeText(TambahProduk.this, "Semua field
harus diisi", Toast.LENGTH_SHORT).show();
    }
}
});

}

@Override
public void onItemClick(AdapterView<?> adapterView, View
view, int i, long l) {
    JenisProduk = adapterView.getItemAtPosition(i).toString();
}

@Override
public void onNothingSelected(AdapterView<?> adapterView) {
}

private void checkAndRequestPermission() {
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
        openGallery();
    } else {
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
PERMISSION_REQUEST_CODE);
    }
}

private void openGallery() {
    Intent intent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
    intent.setType("image/png");
    startActivityForResult(intent, REQUEST_CODE_PICK_IMAGE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_CODE_PICK_IMAGE && resultCode ==
RESULT_OK && data != null) {
        selectedImageUri = data.getData();
        Glide.with(this).load(selectedImageUri).into(tampilfoto)
;
    }
}

```

```

//      tampilfoto.setImageURI(selectedImageUri);
      if (selectedImageUri != null){
          Toast.makeText(this, "Berhasil Memilih Foto",
Toast.LENGTH_SHORT).show();
      }else{
          Toast.makeText(this, "Gagal Memilih Foto",
Toast.LENGTH_SHORT).show();
      }
    }
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
    if (requestCode == PERMISSION_REQUEST_CODE) {
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            openGallery();
        } else {
            Toast.makeText(this, "Izin akses galeri dibutuhkan
untuk memilih foto", Toast.LENGTH_SHORT).show();
        }
    }
}

private String getRealPathFromURI(Uri contentUri) {
    String[] projection = {MediaStore.Images.Media.DATA};
    Cursor cursor = getContentResolver().query(contentUri,
projection, null, null, null);
    if (cursor == null) {
        return contentUri.getPath();
    } else {
        int columnIndex =
cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
        cursor.moveToFirst();
        String filePath = cursor.getString(columnIndex);
        cursor.close();
        return filePath;
    }
}

private byte[] getBytesFromInputStream(InputStream inputStream)
throws IOException {
    ByteArrayOutputStream byteBuffer = new
ByteArrayOutputStream();
}

```

```

        int bufferSize = 1024;
        byte[] buffer = new byte[bufferSize];
        int len;
        while ((len = inputStream.read(buffer)) != -1) {
            byteBuffer.write(buffer, 0, len);
        }
        return byteBuffer.toByteArray();
    }

    private void tambahData(File imgFile, String NamaProduk, String
DeskripsiProduk){
        switch (JenisProduk){
            case "INSEKTISIDA" :
                try {
                    InputStream inputStream =
getContentResolver().openInputStream(selectedImageUri);

                    byte[] imageByte =
getBytesFromInputStream(inputStream);

                    RequestBody imageRequestBodyInsektisida =
RequestBody.create(MediaType.parse("image/png"), imageByte);

                    MultipartBody.Part FotoInsektisida =
MultipartBody.Part.createFormData("browsersure",imgFile.getName(),
imageRequestBodyInsektisida);
                    RequestBody MerkInsektisida =
RequestBody.create(MediaType.parse("text/plain"), NamaProduk);
                    RequestBody DeksripsiInsektisida =
RequestBody.create(MediaType.parse("text/plain"), DeskripsiProduk);

                    Log.d("MyTag", "tambahData INSEKTISIDA:
"+imageRequestBodyInsektisida>NamaProduk+DeskripsiProduk);

                    InterfaceInsektisida interfaceInsektisida =
RetroServer.KonesiAPI().create(InterfaceInsektisida.class);
                    Call<ModelInsektisida> tambahInsektisida =
interfaceInsektisida.setDataInsektisida(FotoInsektisida,MerkInsektis
ida,DeksripsiInsektisida);
                    tambahInsektisida.enqueue(new
Callback<ModelInsektisida>() {
                        @Override
                        public void
onResponse(Call<ModelInsektisida> call, Response<ModelInsektisida>
response) {
                            Toast.makeText(TambahProduk.this,
"Berhasil Tambah Data", Toast.LENGTH_SHORT).show();

```

```

        finish();
    }

    @Override
    public void onFailure(Call<ModelInsektisida>
call, Throwable t) {
        Toast.makeText(TambahProduk.this,
"Berhasil Tambah Data", Toast.LENGTH_SHORT).show();
        finish();
    }
});
} catch (IOException e){
    e.printStackTrace();
    Toast.makeText(this, "Gagal Membaca Gambar",
Toast.LENGTH_SHORT).show();
}

break;
case "FUNGISIDA" :
try {
    InputStream inputStream =
getContentResolver().openInputStream(selectedImageUri);

    byte[] imageByte =
getBytesFromInputStream(inputStream);

    RequestBody imageRequestBodyFungisida =
RequestBody.create(MediaType.parse("image/png"), imageByte);

    MultipartBody.Part FotoFungisida =
MultipartBody.Part.createFormData("browsersure",imgFile.getName(),
imageRequestBodyFungisida);
    RequestBody MerkFungisida =
RequestBody.create(MediaType.parse("text/plain"), NamaProduk);
    RequestBody DeksripsiFungisida =
RequestBody.create(MediaType.parse("text/plain"), DeskripsiProduk);

    Log.d("MyTag", "tambahData FUNGISIDA :
"+imageRequestBodyFungisida>NamaProduk+DeskripsiProduk);

    InterfaceFungisida interfaceFungisida =
RetroServer.KonesiAPI().create(InterfaceFungisida.class);
    Call<ModelFungisida> tambahFungisida =
interfaceFungisida.setDataFungisida(FotoFungisida,MerkFungisida,Deks
ripsiFungisida);
    tambahFungisida.enqueue(new
Callback<ModelFungisida>() {

```

```

        @Override
        public void onResponse(Call<ModelFungisida>
call, Response<ModelFungisida> response) {
            Toast.makeText(TambahProduk.this,
"Berhasil Tambah Data", Toast.LENGTH_SHORT).show();
            finish();
        }

        @Override
        public void onFailure(Call<ModelFungisida>
call, Throwable t) {
            Toast.makeText(TambahProduk.this,
"Berhasil Tambah Data", Toast.LENGTH_SHORT).show();
            finish();
        }
    });
} catch (IOException e){
    e.printStackTrace();
    Toast.makeText(this, "Gagal Membaca Gambar",
Toast.LENGTH_SHORT).show();
}

break;
case "HERBISIDA" :
    try {
        InputStream inputStream =
getContentResolver().openInputStream(selectedImageUri);

        byte[] imageByte =
getBytesFromInputStream(inputStream);

        RequestBody imageRequestBodyHerbisida =
RequestBody.create(MediaType.parse("image/png"), imageByte);

        MultipartBody.Part FotoHerbisida =
MultipartBody.Part.createFormData("browsersure",imgFile.getName(),
imageRequestBodyHerbisida);
        RequestBody MerkHerbisida =
RequestBody.create(MediaType.parse("text/plain"), NamaProduk);
        RequestBody DeksripsiHerbisida =
RequestBody.create(MediaType.parse("text/plain"), DeskripsiProduk);

        Log.d("MyTag", "tambahData HERBISIDA:
"+imageRequestBodyHerbisida>NamaProduk+DeskripsiProduk);

        InterfaceHerbisida interfaceHerbisida =
RetroServer.KonesiAPI().create(InterfaceHerbisida.class);

```

```

        Call<ModelHerbisida> tambahHerbisida =
interfaceHerbisida.setDataHerbisida(FotoHerbisida, MerkHerbisida, DeskripsiHerbisida);

        tambahHerbisida.enqueue(new
Callback<ModelHerbisida>() {
            @Override
            public void onResponse(Call<ModelHerbisida>
call, Response<ModelHerbisida> response) {
                Toast.makeText(TambahProduk.this,
"Berhasil Tambah Data", Toast.LENGTH_SHORT).show();
                finish();
            }

            @Override
            public void onFailure(Call<ModelHerbisida>
call, Throwable t) {
                Toast.makeText(TambahProduk.this,
"Berhasil Tambah Data", Toast.LENGTH_SHORT).show();
                finish();
            }
        });
    } catch (IOException e){
        e.printStackTrace();
        Toast.makeText(this, "Gagal Membaca Gambar",
Toast.LENGTH_SHORT).show();
    }
    break;
    case "PGR" :
        try {
            InputStream inputStream =
getContentResolver().openInputStream(selectedImageUri);

            byte[] imageByte =
getBytesFromInputStream(inputStream);

            RequestBody imageRequestBodyPgr =
RequestBody.create(MediaType.parse("image/png"), imageByte);

            MultipartBody.Part FotoPgr =
MultipartBody.Part.createFormData("browsure", imgFile.getName(),
imageRequestBodyPgr);

            RequestBody MerkPgr =
RequestBody.create(MediaType.parse("text/plain"), NamaProduk);

            RequestBody DeskripsiPgr =
RequestBody.create(MediaType.parse("text/plain"), DeskripsiProduk);

```



```

import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Interface.InterfaceFungisida;
import com.example.cakraagroindonesia.Interface.InterfaceHerbisida;
import
com.example.cakraagroindonesia.Interface.InterfaceInsektisida;
import com.example.cakraagroindonesia.Interface.InterfacePgr;
import com.example.cakraagroindonesia.Model.ModelFungisida;
import com.example.cakraagroindonesia.Model.ModelHerbisida;
import com.example.cakraagroindonesia.Model.ModelInsektisida;
import com.example.cakraagroindonesia.Model.ModelPgr;
import com.example.cakraagroindonesia.R;

import java.io.File;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class UbahProduk extends AppCompatActivity implements
AdapterView.OnItemSelectedListener {

    private int varId;
    private String varNamaProduk, varDeskripsi, varBrowsure,
varJenisProduk;

    private Spinner jenisproduk;

```



```

private EditText namaproduk, kandungan, deskripsiproduk;
private TextView btntambah, uploadfoto;
private ImageView tampilfoto;
private String JenisProduk, NamaProduk, Kandungan,
DeskripsiProduk;

private Uri selectedImageUri;

private static final int REQUEST_CODE_PICK_IMAGE = 101;
private static final int PERMISSION_REQUEST_CODE = 102;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_tambah_produk);

    jenisproduk = findViewById(R.id.jenisproduk);
    namaproduk = findViewById(R.id.namaproduk);
    kandungan = findViewById(R.id.kandungan);
    deskripsiproduk = findViewById(R.id.deskripsiproduk);
    tampilfoto = findViewById(R.id.tampilfoto);
    btntambah = findViewById(R.id.btntambah);
    uploadfoto = findViewById(R.id.uploadfoto);

    varId = getIntent().getIntExtra("xId", -1);
    varNamaProduk = getIntent().getStringExtra("xNamaProduk");
    varDeskripsi = getIntent().getStringExtra("xDeskripsi");
    varBrowsure = getIntent().getStringExtra("xBrowsure");
    varJenisProduk = getIntent().getStringExtra("xJenisProduk");

    namaproduk.setText(varNamaProduk);
    deskripsiproduk.setText(varDeskripsi);
    // Glide.with(this).load(varBrowsure).into(tampilfoto);

    uploadfoto.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            checkAndRequestPermission();
        }
    });

    btntambah.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            NamaProduk = namaproduk.getText().toString().trim();
            Kandungan = kandungan.getText().toString().trim();

```

```

        DeskripsiProduk =
deskripsiproduk.getText().toString().trim();

        if (!NamaProduk.isEmpty() &&
!DeskripsiProduk.isEmpty()){
            String imagePath =
getRealPathFromURI(selectedImageUri);
            tambahData(varJenisProduk, imagePath,
NamaProduk, DeskripsiProduk);
        }else{
            Toast.makeText(UbahProduk.this, "Semua field
harus diisi", Toast.LENGTH_SHORT).show();
        }
    }
});

}

@Override
public void onItemClick(AdapterView<?> adapterView, View
view, int i, long l) {
    JenisProduk = adapterView.getItemAtPosition(i).toString();
}

@Override
public void onNothingSelected(AdapterView<?> adapterView) {

}

private void checkAndRequestPermission() {
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
        openGallery();
    } else {
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
PERMISSION_REQUEST_CODE);
    }
}

private void openGallery() {
    Intent intent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
    intent.setType("image/png");
    startActivityForResult(intent, REQUEST_CODE_PICK_IMAGE);
}

```

```

    @Override
    protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == REQUEST_CODE_PICK_IMAGE && resultCode ==
RESULT_OK && data != null) {
            selectedImageUri = data.getData();
            tampilfoto.setImageURI(selectedImageUri);
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        if (requestCode == PERMISSION_REQUEST_CODE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                openGallery();
            } else {
                Toast.makeText(this, "Izin akses galeri dibutuhkan
untuk memilih foto", Toast.LENGTH_SHORT).show();
            }
        }
    }

    private String getRealPathFromURI(Uri contentUri) {
        String[] projection = {MediaStore.Images.Media.DATA};
        Cursor cursor = getContentResolver().query(contentUri,
projection, null, null, null);
        if (cursor == null) {
            return contentUri.getPath();
        } else {
            int columnIndex =
cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
            cursor.moveToFirst();
            String filePath = cursor.getString(columnIndex);
            cursor.close();
            return filePath;
        }
    }

    private void tambahData(String JenisProduk,String imagePath,
String NamaProduk, String DeskripsiProduk){
        switch (JenisProduk){

```

```

        case "INSEKTISIDA" :
            Log.d("MyTag", "tambahData: "+JenisProduk+varId);
            File imageFileInsektisida = new File(imagePath);
            RequestBody imageRequestBodyInsektisida =
RequestBody.create(MediaType.parse("image/*"),
imageFileInsektisida);

            MultipartBody.Part FotoInsektisida =
MultipartBody.Part.createFormData("browsure",imageFileInsektisida.ge
tName(), imageRequestBodyInsektisida);
            RequestBody MerkInsektisida =
RequestBody.create(MediaType.parse("text/plain"), NamaProduk);
            RequestBody DeksripsiInsektisida =
RequestBody.create(MediaType.parse("text/plain"), DeskripsiProduk);

            Log.d("MyTag", "tambahData INSEKTISIDA:
"+imageRequestBodyInsektisida>NamaProduk+DeskripsiProduk);

            InterfaceInsektisida interfaceInsektisida =
RetroServer.KonesiAPI().create(InterfaceInsektisida.class);
            Call<ModelInsektisida> tambahInsektisida =
interfaceInsektisida.updateInsektisida(varId,FotoInsektisida,MerkIns
ektisida,DeksripsiInsektisida);
            tambahInsektisida.enqueue(new
Callback<ModelInsektisida>() {
                @Override
                public void onResponse(Call<ModelInsektisida>
call, Response<ModelInsektisida> response) {
                    Toast.makeText(UbahProduk.this, "Berhasil
Tambah Data", Toast.LENGTH_SHORT).show();
                    finish();
                }

                @Override
                public void onFailure(Call<ModelInsektisida>
call, Throwable t) {
                    Toast.makeText(UbahProduk.this, "Berhasil
Tambah Data", Toast.LENGTH_SHORT).show();
                    finish();
                }
            });
            break;
        case "FUNGISIDA" :
            Log.d("MyTag", "tambahData: "+JenisProduk+varId);
            File imageFileFungisida = new File(imagePath);

```

```

        RequestBody imageRequestBodyFungisida =
RequestBody.create(MediaType.parse("image/png"),
imageFileFungisida);

        MultipartBody.Part FotoFungisida =
MultipartBody.Part.createFormData("browsersure",imageFileFungisida.getN
ame(), imageRequestBodyFungisida);
        RequestBody MerkFungisida =
RequestBody.create(MediaType.parse("text/plain"), NamaProduk);
        RequestBody DeksripsiFungisida =
RequestBody.create(MediaType.parse("text/plain"), DeskripsiProduk);

        Log.d("MyTag", "tambahData FUNGISIDA :
"+imagePath+NamaProduk+DeskripsiProduk);

        InterfaceFungisida interfaceFungisida =
RetroServer.KonesiAPI().create(InterfaceFungisida.class);
        Call<ModelFungisida> tambahFungisida =
interfaceFungisida.updateFungisida(varId,FotoFungisida,MerkFungisida
,DeksripsiFungisida);
        tambahFungisida.enqueue(new
Callback<ModelFungisida>() {
            @Override
            public void onResponse(Call<ModelFungisida>
call, Response<ModelFungisida> response) {
                Toast.makeText(UbahProduk.this, "Berhasil
Tambah Data", Toast.LENGTH_SHORT).show();
                finish();
            }

            @Override
            public void onFailure(Call<ModelFungisida> call,
Throwable t) {
                Toast.makeText(UbahProduk.this, "Berhasil
Tambah Data", Toast.LENGTH_SHORT).show();
                finish();}
        });

        break;
        case "HERBISIDA" :
            Log.d("MyTag", "tambahData: "+JenisProduk+varId);
            File imageFileHerbisida = new File(imagePath);
            RequestBody imageRequestBodyHerbisida =
RequestBody.create(MediaType.parse("image/png"),
imageFileHerbisida);

```

```

        MultipartBody.Part FotoHerbisida =
MultipartBody.Part.createFormData("browsure",imageFileHerbisida.getN
ame(), imageRequestBodyHerbisida);
        RequestBody MerkHerbisida =
RequestBody.create(MediaType.parse("text/plain"), NamaProduk);
        RequestBody DeksripsiHerbisida =
RequestBody.create(MediaType.parse("text/plain"), DeskripsiProduk);

        Log.d("MyTag", "tambahData HERBISIDA :
"+imagePath+NamaProduk+DeskripsiProduk);

        InterfaceHerbisida interfaceHerbisida =
RetroServer.KonesiAPI().create(InterfaceHerbisida.class);
        Call<ModelHerbisida> tambahHerbisida =
interfaceHerbisida.setDataHerbisida(FotoHerbisida,MerkHerbisida,Deks
ripsiHerbisida);
        tambahHerbisida.enqueue(new
Callback<ModelHerbisida>() {
            @Override
            public void onResponse(Call<ModelHerbisida>
call, Response<ModelHerbisida> response) {
                Toast.makeText(UbahProduk.this, "Berhasil
Tambah Data", Toast.LENGTH_SHORT).show();
                finish();
            }

            @Override
            public void onFailure(Call<ModelHerbisida> call,
Throwable t) {
                Toast.makeText(UbahProduk.this, "Berhasil
Tambah Data", Toast.LENGTH_SHORT).show();
                finish();
            }
        });
        break;
        case "PGR" :
            Log.d("MyTag", "tambahData: "+JenisProduk+varId);
            File imageFilePgr = new File(imagePath);
            RequestBody imageRequestBodyPgr =
RequestBody.create(MediaType.parse("image/png"), imageFilePgr);

            MultipartBody.Part FotoPgr =
MultipartBody.Part.createFormData("browsure",imageFilePgr.getName(),
imageRequestBodyPgr);
            RequestBody MerkPgr =
RequestBody.create(MediaType.parse("text/plain"), NamaProduk);

```



```

import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.example.cakraagroindonesia.API.RetroServer;
import
com.example.cakraagroindonesia.Interface.InterfaceDemonstrator;
import com.example.cakraagroindonesia.Interface.InterfaceReport;
import com.example.cakraagroindonesia.Model.ModelDemonstrator;
import com.example.cakraagroindonesia.Model.ModelReport;
import com.example.cakraagroindonesia.R;

import java.io.File;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.List;
import java.util.Locale;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class DaftarReportBaru extends AppCompatActivity {
    private Calendar calendar;

    private EditText notelp, namapetani, produk, dosis, tanaman,
kabupaten, kecamatan, desa, result, status, statusapps;
    private Spinner spinner;

```



```

private ImageView foto;
private TextView tanggalreport, btnPilihFoto,btnsubmit;

private String TanggalReport, Notelp, NamaPetani, Produk, Dosis,
Tanaman, Kabupaten, Kecamatan, Desa, Result, Status, StatusApps;
private String kodeDemonstratorTerpilih, kodeSupervisorTerpilih,
kodeManagerTerpilih, namaDemonstratorTerpilih,
namaSupervisorTerpilih, kabupatenDemonstratorTerpilih,
provinsiDemonstratorTerpilih;

private ArrayAdapter spinnerAdapter;

private Uri selectedImageUri;
private static final int REQUEST_CODE_PICK_IMAGE = 101;
private static final int PERMISSION_REQUEST_CODE = 102;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_daftar_report_baru);

    calendar = Calendar.getInstance();

    tanggalreport = findViewById(R.id.tanggalreport);
    notelp = findViewById(R.id.notelp);
    namapetani = findViewById(R.id.namapetani);
    produk = findViewById(R.id.produk);
    dosis = findViewById(R.id.dosis);
    tanaman = findViewById(R.id.tanaman);
    kabupaten = findViewById(R.id.kabupaten);
    kecamatan = findViewById(R.id.kecamatan);
    desa = findViewById(R.id.desa);
    spinner = findViewById(R.id.spinner);
    foto = findViewById(R.id.foto);
    btnPilihFoto = findViewById(R.id.btnPilihFoto);
    btnsubmit = findViewById(R.id.btnsubmit);
    result = findViewById(R.id.result);
    status = findViewById(R.id.status);
    statusapps = findViewById(R.id.statusapps);

    tanggalreport.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view) {
        showDatePickerDialog();
    }
});
spinnerDemonstrator();

```

```

        btnPilihFoto.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                checkAndRequestPermission();
            }
        });
        btnsubmit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                TanggalReport =
tanggalreport.getText().toString().trim();
                Notelp = notelp.getText().toString().trim();
                NamaPetani = namapetani.getText().toString().trim();
                Produk = produk.getText().toString().trim();
                Dosis = dosis.getText().toString().trim();
                Tanaman = tanaman.getText().toString().trim();
                Kabupaten = kabupaten.getText().toString().trim();
                Kecamatan = kecamatan.getText().toString().trim();
                Desa = desa.getText().toString().trim();
                Result = result.getText().toString().trim();
                Status = status.getText().toString().trim();
                StatusApps = statusapps.getText().toString().trim();

                if (!TanggalReport.isEmpty() && !Notelp.isEmpty() &&
!NamaPetani.isEmpty() && !Produk.isEmpty() && !Kabupaten.isEmpty()
&& !Kecamatan.isEmpty() && !Desa.isEmpty() && !Result.isEmpty() &&
!Status.isEmpty() && !StatusApps.isEmpty()){
                    if (selectedImageUri != null){
                        String imagePath =
getRealPathFromURI(selectedImageUri);
                        tambahData(imagePath, TanggalReport,
Notelp,NamaPetani, Produk, Dosis, Tanaman, Kabupaten, Kecamatan,
Desa, Result, Status, StatusApps);
                    }
                }
            }
        });

    }
    private void checkAndRequestPermission() {
        if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
            openGallery();
        } else {

```

```

        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
PERMISSION_REQUEST_CODE);
    }
}
private void openGallery() {
    Intent intent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
    startActivityForResult(intent, REQUEST_CODE_PICK_IMAGE);
}
@Override
protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_CODE_PICK_IMAGE && resultCode ==
RESULT_OK && data != null) {
        selectedImageUri = data.getData();
        foto.setImageURI(selectedImageUri);
    }
}
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
    if (requestCode == PERMISSION_REQUEST_CODE) {
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            openGallery();
        } else {
            Toast.makeText(this, "Izin akses galeri dibutuhkan
untuk memilih foto", Toast.LENGTH_SHORT).show();
        }
    }
}
private String getRealPathFromURI(Uri contentUri) {
    String[] projection = {MediaStore.Images.Media.DATA};
    Cursor cursor = getContentResolver().query(contentUri,
projection, null, null, null);
    if (cursor == null) {
        return contentUri.getPath();
    } else {
        int columnIndex =
cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
        cursor.moveToFirst();
        String filePath = cursor.getString(columnIndex);
        cursor.close();
    }
}

```

```

        return filePath;
    }
}

private void spinnerDemonstrator(){
    InterfaceDemonstrator interfaceDemonstrator =
RetroServer.KonesiAPI().create(InterfaceDemonstrator.class);
    Call<ModelDemonstrator> Spinner =
interfaceDemonstrator.getDemonstrator();
    Spinner.enqueue(new Callback<ModelDemonstrator>() {
        @Override
        public void onResponse(Call<ModelDemonstrator> call,
Response<ModelDemonstrator> response) {
            ModelDemonstrator modelDemonstrator =
response.body();
            ArrayList<ModelDemonstrator.demonstrator>
demonstratorList = modelDemonstrator.getDemonstrator();

            List<String> namaDemonstrator = new ArrayList<>();
            for (ModelDemonstrator.demonstrator namaDs :
demonstratorList){
                namaDemonstrator.add(namaDs.getNama_demonstrator
());
            }
            List<String> kodeDemonstrator = new ArrayList<>();
            for (ModelDemonstrator.demonstrator kodeDs :
demonstratorList){
                kodeDemonstrator.add(kodeDs.getKode_ds());
            }
            List<String> kodeSupervisor = new ArrayList<>();
            for (ModelDemonstrator.demonstrator kodeSv :
demonstratorList){
                kodeSupervisor.add(kodeSv.getKode_sv());
            }
            List<String> namaSupervisor = new ArrayList<>();
            for (ModelDemonstrator.demonstrator namaSv :
demonstratorList){
                namaSupervisor.add(namaSv.getNama_supervisor());
            }
            List<String> kodeManager = new ArrayList<>();
            for (ModelDemonstrator.demonstrator kodeMg :
demonstratorList){
                kodeManager.add(kodeMg.getKode_mg());
            }
            List<String> kabupatenDesmontrator = new
ArrayList<>();

```

```

        for (ModelDemonstrator.demonstrator kabupatenDs :
demonstratorList){
            kabupatenDesmontrator.add(kabupatenDs.getKabupat
en());
        }
        List<String> provinsiDemonstrator = new
ArrayList<>();
        for (ModelDemonstrator.demonstrator provinsiDs :
demonstratorList){
            provinsiDemonstrator.add(provinsiDs.getProvinsi(
));
        }
        spinnerAdapter = new
ArrayAdapter<>(DaftarReportBaru.this,
android.R.layout.simple_spinner_item, namaDemonstrator);
        spinnerAdapter.setDropDownViewResource(android.R.lay
out.simple_spinner_dropdown_item);
        spinner.setAdapter(spinnerAdapter);

        spinner.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?>
adapterView, View view, int i, long l) {
                kodeDemonstratorTerpilih =
kodeDemonstrator.get(i);
                kodeSupervisorTerpilih =
kodeSupervisor.get(i);
                kodeManagerTerpilih = kodeManager.get(i);
                namaDemonstratorTerpilih =
namaDemonstrator.get(i);
                namaSupervisorTerpilih =
namaSupervisor.get(i);
                kabupatenDemonstratorTerpilih =
kabupatenDesmontrator.get(i);
                provinsiDemonstratorTerpilih =
provinsiDemonstrator.get(i);
            }
            @Override
            public void onNothingSelected(AdapterView<?>
adapterView) {
            }
        });
    }
    @Override
    public void onFailure(Call<ModelDemonstrator> call,
Throwable t) {

```

```

    }
    });
}
private void showDatePickerDialog() {
    DatePickerDialog datePickerDialog = new DatePickerDialog(
        DaftarReportBaru.this,
        (view, year, month, dayOfMonth) -> {
            calendar.set(year, month, dayOfMonth);
            String myFormat = "yyyy-MM-dd"; // Format
tanggal yang diinginkan
            SimpleDateFormat sdf = new
SimpleDateFormat(myFormat, Locale.getDefault());
            tanggalreport.setText(sdf.format(calendar.getTim
e()));
        },
        calendar.get(Calendar.YEAR),
        calendar.get(Calendar.MONTH),
        calendar.get(Calendar.DAY_OF_MONTH)
    );
//
// // Batasi tanggal yang bisa dipilih (opsional)
// datePickerDialog.getDatePicker().setMaxDate(System.current
TimeMillis()); // Hanya tanggal hari ini atau sebelumnya
// //
datePickerDialog.getDatePicker().setMinDate(System.currentTimeMillis
()); // Hanya tanggal hari ini atau setelahnya
//
    datePickerDialog.show();
}
private void tambahData(String imagePath,String
STanggalReport,String SNotelp, String SNamaPetani, String SProduk,
String SDosis, String STanaman, String SKabupaten, String
SKecamatan,String SDesa, String SResult, String SStatus, String
SStatusApps) {
    File imageFile = new File(imagePath);
    RequestBody imageRequestBody =
RequestBody.create(MediaType.parse("image/*"), imageFile);
    MultipartBody.Part Foto =
MultipartBody.Part.createFormData("foto_buku", imageFile.getName(),
imageRequestBody);
    RequestBody Tgl =
RequestBody.create(MediaType.parse("text/plain"), STanggalReport);
    RequestBody Notelp =
RequestBody.create(MediaType.parse("text/plain"), SNotelp);
    RequestBody Petani =
RequestBody.create(MediaType.parse("text/plain"), SNamaPetani);

```

```

        RequestBody Produk =
RequestBody.create(MediaType.parse("text/plain"), SProduk);
        RequestBody Dosis =
RequestBody.create(MediaType.parse("text/plain"), SDosis);
        RequestBody Tanaman =
RequestBody.create(MediaType.parse("text/plain"), STanaman);
        RequestBody Kabupaten =
RequestBody.create(MediaType.parse("text/plain"), SKabupaten);
        RequestBody Kecamatan =
RequestBody.create(MediaType.parse("text/plain"), SKecamatan);
        RequestBody Desa =
RequestBody.create(MediaType.parse("text/plain"), SDesa);
        RequestBody Result =
RequestBody.create(MediaType.parse("text/plain"), SResult);
        RequestBody Status =
RequestBody.create(MediaType.parse("text/plain"), SStatus);
        RequestBody StatusApps =
RequestBody.create(MediaType.parse("text/plain"), SStatusApps);
        RequestBody kodeMG =
RequestBody.create(MediaType.parse("text/plain"),
kodeManagerTerpilih);
        RequestBody kodeDs =
RequestBody.create(MediaType.parse("text/plain"),
kodeDemonstratorTerpilih);
        RequestBody namaDs =
RequestBody.create(MediaType.parse("text/plain"),
namaDemonstratorTerpilih);
        RequestBody kodeSV =
RequestBody.create(MediaType.parse("text/plain"),
kodeSupervisorTerpilih);
        RequestBody namaSV =
RequestBody.create(MediaType.parse("text/plain"),
namaSupervisorTerpilih);
        RequestBody kabupatenDs =
RequestBody.create(MediaType.parse("text/plain"),
kabupatenDemonstratorTerpilih);
        RequestBody provinsiDs =
RequestBody.create(MediaType.parse("text/plain"),
provinsiDemonstratorTerpilih);

        Log.d("MyTag", "onResponse: " + imagePath + STanggalReport +
SNotelp + SNamaPetani + SProduk + SDosis + STanaman + SKabupaten +
SKecamatan + SDesa + SResult + SStatus + SStatusApps +
kodeManagerTerpilih + kodeDemonstratorTerpilih +
namaDemonstratorTerpilih + kodeSupervisorTerpilih +
namaSupervisorTerpilih);

```

```

        InterfaceReport interfaceReport =
RetroServer.KonesiAPI().create(InterfaceReport.class);
        Call<ModelReport> simpan =
interfaceReport.setReport(Foto,kodeDs,kodeSV,kodeMG,namaDs,namaSV,ka
bupatenDs,provinsiDs,Petani,Tgl,Notelp,Desa,Kecamatan,Kabupaten,Tana
man,Produk,Dosis,Result,Status,StatusApps);
        simpan.enqueue(new Callback<ModelReport>() {
            @Override
            public void onResponse(Call<ModelReport> call,
Response<ModelReport> response) {
                Toast.makeText(DaftarReportBaru.this, "Data Berhasil
Disimpan", Toast.LENGTH_SHORT).show();
                finish();
            }
            @Override
            public void onFailure(Call<ModelReport> call, Throwable
t) {
                Toast.makeText(DaftarReportBaru.this, "Data Gagal
Disimpan", Toast.LENGTH_SHORT).show();
            }
        });
    }
}

```

SourceCode 1. 54 Daftar Report Baru

2) Kelola Report

```

package com.example.cakraagroindonesia.Activity.Report;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Adapter.DataReport;
import com.example.cakraagroindonesia.Interface.InterfaceReport;
import com.example.cakraagroindonesia.Model.ModelReport;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

```



```

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class KelolaReport extends AppCompatActivity {
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;

    TextView btnTambah;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_daftar_report);

        btnTambah = findViewById(R.id.btntambahreportdemonstrator);
        btnTambah.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaReport.this,
DaftarReportBaru.class));
            }
        });
    }

    @Override
    protected void onResume() {
        super.onResume();
        recyclerView = findViewById(R.id.rv_kelolareport);
        layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL, false);
        recyclerView.setLayoutManager(layoutManager);

        InterfaceReport interfaceReport =
RetroServer.KonesiAPI().create(InterfaceReport.class);
        Call<ModelReport> tampil = interfaceReport.getReport();
        tampil.enqueue(new Callback<ModelReport>() {
            @Override
            public void onResponse(Call<ModelReport> call,
Response<ModelReport> response) {
                ArrayList<ModelReport.report> listData =
response.body().getReport();
                adapter = new DataReport(KelolaReport.this,
listData);
                recyclerView.setAdapter(adapter);
                adapter.notifyDataSetChanged();
            }
        });
    }
}

```

```

        @Override
        public void onFailure(Call<ModelReport> call, Throwable
t) {

            }
        });
    }
}

```

SourceCode 1. 55 Kelola Report

3) Ubah Report

```

package com.example.cakraagroindonesia.Activity.Report;

import android.Manifest;
import android.app.DatePickerDialog;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.example.cakraagroindonesia.API.RetroServer;
import
com.example.cakraagroindonesia.Interface.InterfaceDemonstrator;
import com.example.cakraagroindonesia.Interface.InterfaceReport;
import com.example.cakraagroindonesia.Model.ModelDemonstrator;
import com.example.cakraagroindonesia.Model.ModelReport;
import com.example.cakraagroindonesia.R;

```

```

import java.io.File;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.List;
import java.util.Locale;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class UbahReport extends AppCompatActivity {

    private String varId, varTanggalReport, varNotelp,
varNamaPetani, varProduk, varDosis, varTanaman, varKabupaten,
varKecamatan, varDesa, varResult, varStatus, varStatusApps;

    private Calendar calendar;

    private EditText notelp, namapetani, produk, dosis, tanaman,
kabupaten, kecamatan, desa, result, status, statusapps;
    private Spinner spinner;
    private ImageView foto;
    private TextView tanggalreport, btnPilihFoto,btnsubmit;

    private String TanggalReport, Notelp, NamaPetani, Produk, Dosis,
Tanaman, Kabupaten, Kecamatan, Desa, Result, Status, StatusApps;
    private String kodeDemonstratorTerpilih, kodeSupervisorTerpilih,
kodeManagerTerpilih, namaDemonstratorTerpilih,
namaSupervisorTerpilih, kabupatenDemonstratorTerpilih,
provinsiDemonstratorTerpilih;

    private ArrayAdapter spinnerAdapter;

    private Uri selectedImageUri;
    private static final int REQUEST_CODE_PICK_IMAGE = 101;
    private static final int PERMISSION_REQUEST_CODE = 102;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ubah_report);
    }

```

```

calendar = Calendar.getInstance();

tanggalreport = findViewById(R.id.tanggalreport);
notelp = findViewById(R.id.notelp);
namapetani = findViewById(R.id.namapetani);
produk = findViewById(R.id.produk);
dosis = findViewById(R.id.dosis);
tanaman = findViewById(R.id.tanaman);
kabupaten = findViewById(R.id.kabupaten);
kecamatan = findViewById(R.id.kecamatan);
desa = findViewById(R.id.desa);
spinner = findViewById(R.id.spinner);
foto = findViewById(R.id.foto);
btnPilihFoto = findViewById(R.id.btnPilihFoto);
btnsubmit = findViewById(R.id.btnsubmit);
result = findViewById(R.id.result);
status = findViewById(R.id.status);
statusapps = findViewById(R.id.statusapps);

varId = getIntent().getStringExtra("xId");
varTanggalReport =
getIntent().getStringExtra("xTanggalReport");
varNotelp = getIntent().getStringExtra("xNotelp");
varNamaPetani = getIntent().getStringExtra("xNamaPetani");
varProduk = getIntent().getStringExtra("xProduk");
varDosis = getIntent().getStringExtra("xDosis");
varTanaman = getIntent().getStringExtra("xTanaman");
varKabupaten = getIntent().getStringExtra("xKabupaten");
varKecamatan = getIntent().getStringExtra("xKecamatan");
varDesa = getIntent().getStringExtra("xDesa");
varResult = getIntent().getStringExtra("xResult");
varStatus = getIntent().getStringExtra("xStatus");
varStatusApps = getIntent().getStringExtra("xStatusApps");

tanggalreport.setText(varTanggalReport);
notelp.setText(varNotelp);
namapetani.setText(varNamaPetani);
produk.setText(varProduk);
dosis.setText(varDosis);
tanaman.setText(varTanaman);
kabupaten.setText(varKabupaten);
kecamatan.setText(varKecamatan);
desa.setText(varDesa);
result.setText(varResult);
status.setText(varStatus);
statusapps.setText(varStatusApps);

```

```

        tanggalreport.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view) {
        showDatePickerDialog();
    }
});
spinnerDemonstrator();
btnPilihFoto.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        checkAndRequestPermission();
    }
});
btnsubmit.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        TanggalReport =
tanggalreport.getText().toString().trim();
        Notelp = notelp.getText().toString().trim();
        NamaPetani = namapetani.getText().toString().trim();
        Produk = produk.getText().toString().trim();
        Dosis = dosis.getText().toString().trim();
        Tanaman = tanaman.getText().toString().trim();
        Kabupaten = kabupaten.getText().toString().trim();
        Kecamatan = kecamatan.getText().toString().trim();
        Desa = desa.getText().toString().trim();
        Result = result.getText().toString().trim();
        Status = status.getText().toString().trim();
        StatusApps = statusapps.getText().toString().trim();

        if (!TanggalReport.isEmpty() && !Notelp.isEmpty() &&
!NamaPetani.isEmpty() && !Produk.isEmpty() && !Kabupaten.isEmpty()
&& !Kecamatan.isEmpty() && !Desa.isEmpty() && !Result.isEmpty() &&
!Status.isEmpty() && !StatusApps.isEmpty()){
            if (selectedImageUri != null){
                String imagePath =
getRealPathFromURI(selectedImageUri);
                tambahData(imagePath, TanggalReport,
Notelp,NamaPetani, Produk, Dosis, Tanaman, Kabupaten, Kecamatan,
Desa, Result, Status, StatusApps);
            }
        }
    }
});
}
}

```

```

        private void checkAndRequestPermission() {
            if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
                openGallery();
            } else {
                ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
PERMISSION_REQUEST_CODE);
            }
        }
        private void openGallery() {
            Intent intent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
            startActivityForResult(intent, REQUEST_CODE_PICK_IMAGE);
        }
        @Override
        protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
            super.onActivityResult(requestCode, resultCode, data);
            if (requestCode == REQUEST_CODE_PICK_IMAGE && resultCode ==
RESULT_OK && data != null) {
                selectedImageUri = data.getData();
                foto.setImageURI(selectedImageUri);
            }
        }
        @Override
        public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
            super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
            if (requestCode == PERMISSION_REQUEST_CODE) {
                if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                    openGallery();
                } else {
                    Toast.makeText(this, "Izin akses galeri dibutuhkan
untuk memilih foto", Toast.LENGTH_SHORT).show();
                }
            }
        }
        private String getRealPathFromURI(Uri contentUri) {
            String[] projection = {MediaStore.Images.Media.DATA};
            Cursor cursor = getContentResolver().query(contentUri,
projection, null, null, null);
            if (cursor == null) {

```

```

        return contentUri.getPath();
    } else {
        int columnIndex =
cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
        cursor.moveToFirst();
        String filePath = cursor.getString(columnIndex);
        cursor.close();
        return filePath;
    }
}

private void spinnerDemonstrator(){
    InterfaceDemonstrator interfaceDemonstrator =
RetroServer.KonesiAPI().create(InterfaceDemonstrator.class);
    Call<ModelDemonstrator> Spinner =
interfaceDemonstrator.getDemonstrator();
    Spinner.enqueue(new Callback<ModelDemonstrator>() {
        @Override
        public void onResponse(Call<ModelDemonstrator> call,
Response<ModelDemonstrator> response) {
            ModelDemonstrator modelDemonstrator =
response.body();
            ArrayList<ModelDemonstrator.demonstrator>
demonstratorList = modelDemonstrator.getDemonstrator();

            List<String> namaDemonstrator = new ArrayList<>();
            for (ModelDemonstrator.demonstrator namaDs :
demonstratorList){
                namaDemonstrator.add(namaDs.getNama_demonstrator
());
            }
            List<String> kodeDemonstrator = new ArrayList<>();
            for (ModelDemonstrator.demonstrator kodeDs :
demonstratorList){
                kodeDemonstrator.add(kodeDs.getKode_ds());
            }
            List<String> kodeSupervisor = new ArrayList<>();
            for (ModelDemonstrator.demonstrator kodeSv :
demonstratorList){
                kodeSupervisor.add(kodeSv.getKode_sv());
            }
            List<String> namaSupervisor = new ArrayList<>();
            for (ModelDemonstrator.demonstrator namaSv :
demonstratorList){
                namaSupervisor.add(namaSv.getNama_supervisor());
            }
            List<String> kodeManager = new ArrayList<>();

```

```

        for (ModelDemonstrator.demonstrator kodeMg :
demonstratorList){
            kodeManager.add(kodeMg.getKode_mg());
        }
        List<String> kabupatenDesmontrator = new
ArrayList<>();
        for (ModelDemonstrator.demonstrator kabupatenDs :
demonstratorList){
            kabupatenDesmontrator.add(kabupatenDs.getKabupat
en());
        }
        List<String> provinsiDemonstrator = new
ArrayList<>();
        for (ModelDemonstrator.demonstrator provinsiDs :
demonstratorList){
            provinsiDemonstrator.add(provinsiDs.getProvinsi(
));
        }
        spinnerAdapter = new ArrayAdapter<>(UbahReport.this,
android.R.layout.simple_spinner_item, namaDemonstrator);
        spinnerAdapter.setDropDownViewResource(android.R.lay
out.simple_spinner_dropdown_item);
        spinner.setAdapter(spinnerAdapter);

        spinner.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?>
adapterView, View view, int i, long l) {
                kodeDemonstratorTerpilih =
kodeDemonstrator.get(i);
                kodeSupervisorTerpilih =
kodeSupervisor.get(i);
                kodeManagerTerpilih = kodeManager.get(i);
                namaDemonstratorTerpilih =
namaDemonstrator.get(i);
                namaSupervisorTerpilih =
namaSupervisor.get(i);
                kabupatenDemonstratorTerpilih =
kabupatenDesmontrator.get(i);
                provinsiDemonstratorTerpilih =
provinsiDemonstrator.get(i);
            }
            @Override
            public void onNothingSelected(AdapterView<?>
adapterView) {
            }
        }

```



```

        });
    }
    @Override
    public void onFailure(Call<ModelDemonstrator> call,
    Throwable t) {
    }
    });
}
private void showDatePickerDialog() {
    DatePickerDialog datePickerDialog = new DatePickerDialog(
        UbahReport.this,
        (view, year, month, dayOfMonth) -> {
            calendar.set(year, month, dayOfMonth);
            String myFormat = "yyyy-MM-dd"; // Format
tanggal yang diinginkan
            SimpleDateFormat sdf = new
SimpleDateFormat(myFormat, Locale.getDefault());
            tanggalreport.setText(sdf.format(calendar.getTim
e()));
        },
        calendar.get(Calendar.YEAR),
        calendar.get(Calendar.MONTH),
        calendar.get(Calendar.DAY_OF_MONTH)
    );
    //
    // // Batasi tanggal yang bisa dipilih (opsional)
    // datePickerDialog.getDatePicker().setMaxDate(System.current
TimeMillis()); // Hanya tanggal hari ini atau sebelumnya
    // //
    datePickerDialog.getDatePicker().setMinDate(System.currentTimeMillis
()); // Hanya tanggal hari ini atau setelahnya
    //
    datePickerDialog.show();
}
private void tambahData(String imagePath, String
STanggalReport, String SNotelp, String SNamaPetani, String SProduk,
String SDosis, String STanaman, String SKabupaten, String
SKecamatan, String SDesa, String SResult, String SStatus, String
SStatusApps) {
    File imageFile = new File(imagePath);
    RequestBody imageRequestBody =
RequestBody.create(MediaType.parse("image/*"), imageFile);
    MultipartBody.Part Foto =
MultipartBody.Part.createFormData("foto_buku", imageFile.getName(),
imageRequestBody);
    RequestBody Tgl =
RequestBody.create(MediaType.parse("text/plain"), STanggalReport);

```

```

        RequestBody Notelp =
RequestBody.create(MediaType.parse("text/plain"), SNotelp);
        RequestBody Petani =
RequestBody.create(MediaType.parse("text/plain"), SNamaPetani);
        RequestBody Produk =
RequestBody.create(MediaType.parse("text/plain"), SProduk);
        RequestBody Dosis =
RequestBody.create(MediaType.parse("text/plain"), SDosis);
        RequestBody Tanaman =
RequestBody.create(MediaType.parse("text/plain"), STanaman);
        RequestBody Kabupaten =
RequestBody.create(MediaType.parse("text/plain"), SKabupaten);
        RequestBody Kecamatan =
RequestBody.create(MediaType.parse("text/plain"), SKecamatan);
        RequestBody Desa =
RequestBody.create(MediaType.parse("text/plain"), SDesa);
        RequestBody Result =
RequestBody.create(MediaType.parse("text/plain"), SResult);
        RequestBody Status =
RequestBody.create(MediaType.parse("text/plain"), SStatus);
        RequestBody StatusApps =
RequestBody.create(MediaType.parse("text/plain"), SStatusApps);
        RequestBody kodeMG =
RequestBody.create(MediaType.parse("text/plain"),
kodeManagerTerpilih);
        RequestBody kodeDs =
RequestBody.create(MediaType.parse("text/plain"),
kodeDemonstratorTerpilih);
        RequestBody namaDs =
RequestBody.create(MediaType.parse("text/plain"),
namaDemonstratorTerpilih);
        RequestBody kodeSV =
RequestBody.create(MediaType.parse("text/plain"),
kodeSupervisorTerpilih);
        RequestBody namaSV =
RequestBody.create(MediaType.parse("text/plain"),
namaSupervisorTerpilih);
        RequestBody kabupatenDs =
RequestBody.create(MediaType.parse("text/plain"),
kabupatenDemonstratorTerpilih);
        RequestBody provinsiDs =
RequestBody.create(MediaType.parse("text/plain"),
provinsiDemonstratorTerpilih);

        Log.d("MyTag", "onResponse: " + imagePath + STanggalReport +
SNotelp + SNamaPetani + SProduk + SDosis + STanaman + SKabupaten +
SKecamatan + SDesa + SResult + SStatus + SStatusApps +

```

```

kodeManagerTerpilih + kodeDemonstratorTerpilih +
namaDemonstratorTerpilih + kodeSupervisorTerpilih +
namaSupervisorTerpilih);

        InterfaceReport interfaceReport =
RetroServer.KonesiAPI().create(InterfaceReport.class);
        Call<ModelReport> simpan =
interfaceReport.updateReport(varId,Foto,kodeDs,kodeSV,kodeMG,namaDs,
namaSV,kabupatenDs,provinsiDs,Petani,Tgl,Notelp,Desa,Kecamatan,Kabup
aten,Tanaman,Produk,Dosis,Result,Status,StatusApps);
        simpan.enqueue(new Callback<ModelReport>() {
            @Override
            public void onResponse(Call<ModelReport> call,
Response<ModelReport> response) {
                Toast.makeText(UbahReport.this, "Data Berhasil
Disimpan", Toast.LENGTH_SHORT).show();
                finish();
            }
            @Override
            public void onFailure(Call<ModelReport> call, Throwable
t) {
                Toast.makeText(UbahReport.this, "Data Gagal
Disimpan", Toast.LENGTH_SHORT).show();
            }
        });
    }
}
}

```

SourceCode 1. 56 Ubah Report

j) Tentang Kami

1) Deskripsi Visi Misi

```

package com.example.cakraagroindonesia.Activity.TentangKami;

import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.API.RetroServer;

```

```

import
com.example.cakraagroindonesia.Interface.InterfaceTentangKami;
import com.example.cakraagroindonesia.Model.ModelTentangKami;
import com.example.cakraagroindonesia.R;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class DeskripsiVisiMisi extends AppCompatActivity {
    private EditText deskripsi, visi, misi;
    private String Deskripsi,Visi,Misi;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_deskripsi_visi_misi);

        TextView tvDeskripsi = findViewById(R.id.deskripsi);
        TextView tvVisi = findViewById(R.id.visi);
        TextView tvMisi = findViewById(R.id.misi);
        TextView btnUbah = findViewById(R.id.bnubah);

        deskripsi = findViewById(R.id.deskripsi);
        visi = findViewById(R.id.visi);
        misi = findViewById(R.id.misi);

        InterfaceTentangKami interfaceTentangKami =
RetroServer.KonesiAPI().create(InterfaceTentangKami.class);

        //TAMPIL DATA
        Call<ModelTentangKami> tampilTentangKami =
interfaceTentangKami.getTentangKami(1);
        tampilTentangKami.enqueue(new Callback<ModelTentangKami>() {
            @Override
            public void onResponse(Call<ModelTentangKami>
tampilTentangKami, Response<ModelTentangKami> response) {
                if (response.isSuccessful()) {
                    ModelTentangKami modelTentangKami =
response.body();

                    String deskripsi =
modelTentangKami.getDeskripsi();
                    String visi = modelTentangKami.getVisi();
                    String misi = modelTentangKami.getMisi();

                    tvDeskripsi.setText(deskripsi);

```


2) Kelola Alamat

```
package com.example.cakraagroindonesia.Activity.TentangKami;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Adapter.DataKelolaAlamat;
import com.example.cakraagroindonesia.Interface.InterfaceAlamat;
import com.example.cakraagroindonesia.Model.ModelAlamat;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;
import java.util.Collections;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class KelolaAlamat extends AppCompatActivity{
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;

    TextView btnTambah;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_alamat);

        btnTambah = findViewById(R.id.btntambah);
        btnTambah.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaAlamat.this,
TambahAlamat.class));
            }
        });

        recyclerView = findViewById(R.id.rv_kelolaalamat);
```

```

        layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL, false);
        recyclerView.setLayoutManager(layoutManager);
    }

    @Override
    protected void onResume() {
        super.onResume();
        InterfaceAlamat interfaceAlamat =
RetroServer.KonesiAPI().create(InterfaceAlamat.class);
        Call<ModelAlamat> tampil = interfaceAlamat.getAlamat();
        tampil.enqueue(new Callback<ModelAlamat>() {
            @Override
            public void onResponse(Call<ModelAlamat> call,
Response<ModelAlamat> response) {
                ArrayList<ModelAlamat.alamat> listData =
response.body().getDataAlamat();
                adapter = new DataKelolaAlamat(KelolaAlamat.this,
listData);

                Collections.reverse(listData);
                recyclerView.setAdapter(adapter);
                adapter.notifyDataSetChanged();
            }

            @Override
            public void onFailure(Call<ModelAlamat> call, Throwable
t) {

            }
        });
    }
}

```

SourceCode 1. 58 Kelola Alamat

3) Kelola Tentang Kami

```

package com.example.cakraagroindonesia.Activity.TentangKami;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.R;

```

```

public class KelolaTentangKami extends AppCompatActivity {
    TextView btnDeskripsiVisiMisi, btnAlamat;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_kelola_tentang_kami);

        btnDeskripsiVisiMisi =
findViewById(R.id.btndeskripsivisimisi);
        btnAlamat = findViewById(R.id.btnalamat);

        btnDeskripsiVisiMisi.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaTentangKami.this,
DeskripsiVisiMisi.class));
            }
        });
        btnAlamat.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaTentangKami.this,
KelolaAlamat.class));
            }
        });
    }
}

```

SourceCode 1. 59 Kelola Tentang Kami

4) Tambah Alamat

```

package com.example.cakraagroindonesia.Activity.TentangKami;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageView;

```



```

import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Interface.InterfaceAlamat;
import com.example.cakraagroindonesia.Model.ModelAlamat;
import com.example.cakraagroindonesia.R;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.IOException;
import java.io.InputStream;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class TambahAlamat extends AppCompatActivity {
    private TextView btnTambah, btnPilihFoto;
    private EditText namakantor, alamat, telepon;
    private ImageView foto;
    private Uri selectedImageUri;

    private static final int REQUEST_CODE_PICK_IMAGE = 101;
    private static final int PERMISSION_REQUEST_CODE = 102;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tambah_alamat);

        btnTambah = findViewById(R.id.btntambah);
        namakantor = findViewById(R.id.namakantor);
        alamat = findViewById(R.id.alamat);
        telepon = findViewById(R.id.telepon);
        foto = findViewById(R.id.foto);
    }

```

```

        btnPilihFoto = findViewById(R.id.btnPilihFoto);

        btnPilihFoto.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                checkAndRequestPermission();
            }
        });

        btnTambah.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String NamaKantor =
namakantor.getText().toString().trim();
                String Alamat = alamat.getText().toString().trim();
                String Telepon =
telepon.getText().toString().trim();

                if (!NamaKantor.isEmpty() && !Alamat.isEmpty() &&
!Telepon.isEmpty()) {
                    if (selectedImageUri != null){
                        File imagePath = new
File(getRealPathFromURI(selectedImageUri));

                        uploadImage(imagePath, NamaKantor, Alamat, Tele
pon);
                    }else {
                        Toast.makeText(TambahAlamat.this, "Pilih
Foto Terlebih dahulu", Toast.LENGTH_SHORT).show();
                    }
                } else {
                    Toast.makeText(TambahAlamat.this, "Semua field
harus diisi", Toast.LENGTH_SHORT).show();
                }
            }
        });
    }

    private void checkAndRequestPermission() {
        if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
            openGallery();
        } else {
            ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
PERMISSION_REQUEST_CODE);
        }
    }
}

```

```

    }
}

private void openGallery() {
    Intent intent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
    intent.setType("image/*");
    startActivityForResult(intent, REQUEST_CODE_PICK_IMAGE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_CODE_PICK_IMAGE && resultCode ==
RESULT_OK && data != null) {
        selectedImageUri = data.getData();
        Glide.with(this).load(selectedImageUri).into(foto);
//        tampilfoto.setImageURI(selectedImageUri);
        if (selectedImageUri != null){
            Toast.makeText(this, "Berhasil Memilih Foto",
Toast.LENGTH_SHORT).show();
        }else{
            Toast.makeText(this, "Gagal Memilih Foto",
Toast.LENGTH_SHORT).show();
        }
    }
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
    if (requestCode == PERMISSION_REQUEST_CODE) {
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            openGallery();
        } else {
            Toast.makeText(this, "Izin akses galeri dibutuhkan
untuk memilih foto", Toast.LENGTH_SHORT).show();
        }
    }
}

private String getRealPathFromURI(Uri contentUri) {
    String[] projection = {MediaStore.Images.Media.DATA};
}

```

```

        Cursor cursor = getContentResolver().query(contentUri,
projection, null, null, null);
        if (cursor == null) {
            return contentUri.getPath();
        } else {
            int columnIndex =
cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
            cursor.moveToFirst();
            String filePath = cursor.getString(columnIndex);
            cursor.close();
            return filePath;
        }
    }

    private byte[] getBytesFromInputStream(InputStream inputStream)
throws IOException {
        ByteArrayOutputStream byteBuffer = new
ByteArrayOutputStream();
        int bufferSize = 1024;
        byte[] buffer = new byte[bufferSize];
        int len;
        while ((len = inputStream.read(buffer)) != -1) {
            byteBuffer.write(buffer, 0, len);
        }
        return byteBuffer.toByteArray();
    }

    private void uploadImage(File imgFile, String NamaKantor, String
Alamat, String Telepon) {
        try {
            InputStream inputStream =
getContentResolver().openInputStream(selectedImageUri);

            byte[] imageByte = getBytesFromInputStream(inputStream);

            RequestBody imageRequestBody =
RequestBody.create(MediaType.parse("image/*"), imageByte);

            MultipartBody.Part Foto =
MultipartBody.Part.createFormData("foto", imgFile.getName(),
imageRequestBody);

            RequestBody Kantor =
RequestBody.create(MediaType.parse("text/plain"), NamaKantor);
            RequestBody Alt =
RequestBody.create(MediaType.parse("text/plain"), Alamat);
            RequestBody Telp =
RequestBody.create(MediaType.parse("text/plain"), Telepon);

```



```

import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Interface.InterfaceAlamat;
import com.example.cakraagroindonesia.Model.ModelAlamat;
import com.example.cakraagroindonesia.R;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.IOException;
import java.io.InputStream;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class UbahAlamat extends AppCompatActivity {
    private TextView btnTambah, btnPilihFoto, idalamat;
    private EditText namakantor, alamat, telepon;
    private ImageView fotox;

    private int varId;
    private String varNamaKantor, varAlamat, varTelepon, varFoto;
    private String NamaKantor,Alamat,Telepon;

    private Uri selectedImageUri;

    private static final int REQUEST_CODE_PICK_IMAGE = 101;
    private static final int PERMISSION_REQUEST_CODE = 102;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ubah_alamat);

        idalamat = findViewById(R.id.idalamat);

```

```

        btnTambah = findViewById(R.id.btntambah);
        namakantor = findViewById(R.id.namakantor);
        alamat = findViewById(R.id.alamat);
        telepon = findViewById(R.id.telepon);
        fotox = findViewById(R.id.fotox);
        btnPilihFoto = findViewById(R.id.btnPilihFoto);

        varId = getIntent().getIntExtra("xId", -1);
        varNamaKantor = getIntent().getStringExtra("xNamaKantor");
        varAlamat = getIntent().getStringExtra("xAlamat");
        varTelepon = getIntent().getStringExtra("xNotelp");
        varFoto = getIntent().getStringExtra("xFoto");

        Log.d("MyTag", "onCreate:
"+varId+varNamaKantor+varAlamat+varTelepon+varFoto);

        namakantor.setText(varNamaKantor);
        alamat.setText(varAlamat);
        telepon.setText(varTelepon);
        Glide.with(UbahAlamat.this).load(varFoto).into(fotox);

        //UBAH
        btnPilihFoto.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                checkAndRequestPermission();
            }
        });

        btnTambah.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String NamaKantor =
namakantor.getText().toString().trim();
                String Alamat = alamat.getText().toString().trim();
                String Telepon =
telepon.getText().toString().trim();

                if (!NamaKantor.isEmpty() && !Alamat.isEmpty() &&
!Telepon.isEmpty()) {
                    if (selectedImageUri != null){
                        File imagePath = new
File(getRealPathFromURI(selectedImageUri));

```

```

        uploadImage(imagePath, NamaKantor, Alamat, Telepon);
    }else {

        InterfaceAlamat interfaceAlamat =
RetroServer.KonesiAPI().create(InterfaceAlamat.class);
        Call<ModelAlamat> update =
interfaceAlamat.update(varId, NamaKantor, Alamat, Telepon);
        update.enqueue(new Callback<ModelAlamat>() {
            @Override
            public void onResponse(Call<ModelAlamat>
call, Response<ModelAlamat> response) {
                Toast.makeText(UbahAlamat.this, "Data
Berhasil Disimpan ", Toast.LENGTH_SHORT).show();
                finish();
            }
            @Override
            public void onFailure(Call<ModelAlamat>
call, Throwable t) {
            }
        });
        Toast.makeText(UbahAlamat.this, "Pilih Foto
Terlebih dahulu", Toast.LENGTH_SHORT).show();
    }
    } else {
        Toast.makeText(UbahAlamat.this, "Semua field
harus diisi", Toast.LENGTH_SHORT).show();
    }
}

private void checkAndRequestPermission() {
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
        openGallery();
    } else {
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
PERMISSION_REQUEST_CODE);
    }
}
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {

```



```

        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        if (requestCode == PERMISSION_REQUEST_CODE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                openGallery();
            } else {
                Toast.makeText(this, "Izin akses galeri dibutuhkan
untuk memilih foto", Toast.LENGTH_SHORT).show();
            }
        }
    }
    private void openGallery() {
        Intent intent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
        intent.setType("image/*");
        startActivityForResult(intent, REQUEST_CODE_PICK_IMAGE);
    }
    @Override
    protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == REQUEST_CODE_PICK_IMAGE && resultCode ==
RESULT_OK && data != null) {
            selectedImageUri = data.getData();
            // varFoto = data.getData();
            Glide.with(this).load(selectedImageUri).into(fotox);
            // tampilfoto.setImageURI(selectedImageUri);
            if (selectedImageUri != null){
                Toast.makeText(this, "Berhasil Memilih Foto",
Toast.LENGTH_SHORT).show();
            }else{
                Toast.makeText(this, "Gagal Memilih Foto",
Toast.LENGTH_SHORT).show();
            }
        }
    }

    private String getRealPathFromURI(Uri contentUri) {
        String[] projection = {MediaStore.Images.Media.DATA};
        Cursor cursor = getContentResolver().query(contentUri,
projection, null, null, null);
        if (cursor == null) {
            return contentUri.getPath();
        } else {
            int columnIndex =
cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);

```

```

        cursor.moveToFirst();
        String filePath = cursor.getString(columnIndex);
        cursor.close();
        return filePath;
    }
}

private byte[] getBytesFromInputStream(InputStream inputStream)
throws IOException {
    ByteArrayOutputStream byteBuffer = new
ByteArrayOutputStream();
    int bufferSize = 1024;
    byte[] buffer = new byte[bufferSize];
    int len;
    while ((len = inputStream.read(buffer)) != -1) {
        byteBuffer.write(buffer, 0, len);
    }
    return byteBuffer.toByteArray();
}

private void uploadImage(File imgFile, String NamaKantor, String
Alamat, String Telepon) {
    try {
        InputStream inputStream =
getContentResolver().openInputStream(selectedImageUri);

        byte[] imageByte = getBytesFromInputStream(inputStream);

        RequestBody imageRequestBody =
RequestBody.create(MediaType.parse("image/*"), imageByte);

        MultipartBody.Part Foto =
MultipartBody.Part.createFormData("foto", imgFile.getName(),
imageRequestBody);
        RequestBody Kantor =
RequestBody.create(MediaType.parse("text/plain"), NamaKantor);
        RequestBody Alt =
RequestBody.create(MediaType.parse("text/plain"), Alamat);
        RequestBody Telp =
RequestBody.create(MediaType.parse("text/plain"), Telepon);

        Log.d("MyTag", "uploadImage:
"+imgFile+NamaKantor+Alamat+Telepon);
        InterfaceAlamat interfaceAlamat =
RetroServer.KonesiAPI().create(InterfaceAlamat.class);
        Call<ModelAlamat> call =
interfaceAlamat.updateAlamat(varId,Foto, Kantor, Alt, Telp);

```

```

        call.enqueue(new Callback<ModelAlamat>() {
            @Override
            public void onResponse(Call<ModelAlamat> call,
Response<ModelAlamat> response) {
                Toast.makeText(UbahAlamat.this, "Data Berhasil
Disimpan ", Toast.LENGTH_SHORT).show();
                finish();
            }
            @Override
            public void onFailure(Call<ModelAlamat> call,
Throwable t) {
                Toast.makeText(UbahAlamat.this, "Data Gagal
Disimpan", Toast.LENGTH_SHORT).show();
            }
        });
    } catch (IOException e){
        e.printStackTrace();
        Toast.makeText(this, "Gagal Membaca Gambar",
Toast.LENGTH_SHORT).show();
    }
}
}
}

```

SourceCode 1. 61 Ubah Alamat

k) Dashboard

```

package com.example.cakraagroindonesia.Activity;

import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.R;

public class Dashboard extends AppCompatActivity {
    TextView idSuperAdmin, namaSuperAdmin, usernameSuperadmin,
passwordSuperadmin;
    ImageView btnEdit;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

```

```

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dashboard);

        deklarasi();

        btnEdit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast.makeText(Dashboard.this, "Belum Ada Aksi",
                    Toast.LENGTH_SHORT).show();
            }
        });
    }

    private void deklarasi() {
        idSuperAdmin = findViewById(R.id.idsuperadmin);
        namaSuperAdmin = findViewById(R.id.namasuperadmin);
        usernameSuperadmin = findViewById(R.id.namasuperadmin);
        passwordSuperadmin = findViewById(R.id.passwordsuperadmin);
        btnEdit = findViewById(R.id.btneditsuperadmin);
    }
}

```

SourceCode 1. 62 Dashboard

l) Panel

```

package com.example.cakraagroindonesia.Activity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.Activity.Aktivitas.KelolaAktivitas;
import com.example.cakraagroindonesia.Activity.Beranda.KelolaBeranda;
import com.example.cakraagroindonesia.Activity.KelolaDataMaster.KelolaDataMaster;
import com.example.cakraagroindonesia.Activity.Inovasi.KelolaInovasi;

```

```

import com.example.cakraagroindonesia.Activity.Log.Log;
import
com.example.cakraagroindonesia.Activity.Pertanyaan.KelolaPertanyaan;
import com.example.cakraagroindonesia.Activity.Produk.KelolaProduk;
import com.example.cakraagroindonesia.Activity.Report.KelolaReport;
import
com.example.cakraagroindonesia.Activity.TentangKami.KelolaTentangKam
i;
import com.example.cakraagroindonesia.R;

public class Panel extends AppCompatActivity {
    TextView btnDashboard, btnNotifikasi, btnLihatLog,
    btnManageReport, btnManageAktivitas, btnManageDataMaster,
    btnManageQna, btnManageBeranda, btnManageProduk, btnManageInovasi,
    btnManageTentangKami;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_panel);

        deklarasi();
        btnDashboard.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(Panel.this,
Dashboard.class));
            }
        });

        btnNotifikasi.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View view) {
                Toast.makeText(Panel.this, "Dalam Pengembangan
Developer...", Toast.LENGTH_SHORT).show();
                // startActivity(new Intent(Panel.this,
Notifikasi.class));
            }
        });

        btnLihatLog.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(Panel.this, Log.class));
            }
        });
    }
}

```

```

        btnManageReport.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(Panel.this,
KelolaReport.class));
            }
        });

        btnManageAktivitas.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(Panel.this,
KelolaAktivitas.class));
            }
        });

        btnManageDataMaster.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(Panel.this,
KelolaDataMaster.class));
            }
        });

        btnManageQna.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(Panel.this,
KelolaPertanyaan.class));
            }
        });

        btnManageBeranda.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(Panel.this,
KelolaBeranda.class));
            }
        });

        btnManageProduk.setOnClickListener(new
View.OnClickListener() {
            @Override

```

```

        public void onClick(View view) {
            startActivity(new Intent(Panel.this,
KelolaProduk.class));
        }
    });

    btnManageInovasi.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {
            startActivity(new Intent(Panel.this,
KelolaInovasi.class));
        }
    });
    btnManageTentangKami.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {
            startActivity(new Intent(Panel.this,
KelolaTentangKami.class));
        }
    });
}

private void deklarasi() {
    btnDashboard = findViewById(R.id.btndashboard);
    btnNotifikasi = findViewById(R.id.btnnotifikasi);
    btnLihatLog = findViewById(R.id.btnlihatlog);
    btnManageReport = findViewById(R.id.btnmanagereport);
    btnManageAktivitas = findViewById(R.id.btnmanageaktivitas);
    btnManageDataMaster =
findViewById(R.id.btnmanagedatamaster);
    btnManageQna = findViewById(R.id.btnkelolapertanyaan);
    btnManageBeranda = findViewById(R.id.btnmanageberanda);
    btnManageProduk = findViewById(R.id.btnmanageproduk);
    btnManageInovasi = findViewById(R.id.btnmanageinovasi);
    btnManageTentangKami
=findViewById(R.id.btnmanagetentangkami);
}
}

```

SourceCode 1. 63 Panel

2. Activity Customer

a) Fragment

1) Fragment Aktivitas

```
package com.example.cakraagroindonesia.ActivityCustomer.Fragment;

import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

import com.example.cakraagroindonesia.ActivityCustomer.Demplot;
import com.example.cakraagroindonesia.ActivityCustomer.FAQ;
import com.example.cakraagroindonesia.ActivityCustomer.PaketProduk;
import com.example.cakraagroindonesia.ActivityCustomer.ProdukBaru;
import com.example.cakraagroindonesia.ActivityCustomer.Aktivitas;
import com.example.cakraagroindonesia.R;

public class FragmentAktivitas extends Fragment {
    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater,
@Nullable ViewGroup container, @Nullable Bundle savedInstanceState)
    {

        View rootView =
inflater.inflate(R.layout.fragment_aktivitas,container,false);

        ImageView aktivitas = (ImageView)
rootView.findViewById(R.id.btnaktivitas);
        aktivitas.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent aktivitas = new Intent(getActivity(),
Aktivitas.class);
                startActivity(aktivitas);
            }
        });

        ImageView demplot = (ImageView)
rootView.findViewById(R.id.btndemplot);
        demplot.setOnClickListener(new View.OnClickListener() {
```



```

        @Override
        public void onClick(View v) {
            Intent demplot = new Intent(getActivity(),
Demplot.class);
            startActivity(demplot);
        }
    });

    ImageView produkbaru = (ImageView)
rootView.findViewById(R.id.btnprodukbaru);
    produkbaru.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent produkbaru = new Intent(getActivity(),
ProdukBaru.class);
            startActivity(produkbaru);
        }
    });

    ImageView paketproduk = (ImageView)
rootView.findViewById(R.id.btnpaketproduk);
    paketproduk.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent paketproduk = new Intent(getActivity(),
PaketProduk.class);
            startActivity(paketproduk);
        }
    });

    TextView btnFaq = rootView.findViewById(R.id.btnfaq);
    btnFaq.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent faq = new Intent(getActivity(), FAQ.class);
            startActivity(faq);
        }
    });

    return rootView;
}
}

```

SourceCode 1. 64 Fragment Aktivitas

2) Fragment Home

```
package com.example.cakraagroindonesia.ActivityCustomer.Fragment;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.denzcoskun.imageslider.ImageSlider;
import com.denzcoskun.imageslider.constants.ScaleTypes;
import com.denzcoskun.imageslider.models.SlideModel;
import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Activity.Panel;
import com.example.cakraagroindonesia.Adapter.DataBerita;
import
com.example.cakraagroindonesia.Adapter.DataTampilDemploHomePage;
import com.example.cakraagroindonesia.Interface.InterfaceBeranda;
import com.example.cakraagroindonesia.Interface.InterfaceBerita;
import com.example.cakraagroindonesia.Interface.InterfaceReport;
import com.example.cakraagroindonesia.Model.ModelBeranda;
import com.example.cakraagroindonesia.Model.ModelBerita;
import com.example.cakraagroindonesia.Model.ModelReport;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class FragmentHome extends Fragment {

    private Context fragmentContext;
    private ImageSlider imageSlider;
```

```

private RecyclerView rvBerita, rvDemplot;
private RecyclerView.Adapter adBerita, adDemplot;
private RecyclerView.LayoutManager lmBerita, lmDemplot;

private TextView btn;

@Nullable
@Override
public View onCreateView(@NonNull LayoutInflater inflater,
@Nullable ViewGroup container, @Nullable Bundle savedInstanceState)
{
    View rootView = inflater.inflate(R.layout.fragment_home,
container, false);
    AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE
_NIGHT_NO);

    btn = rootView.findViewById(R.id.btn);
    btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            startActivity(new Intent(getActivity(),
Panel.class));
        }
    });

    imageSlider = rootView.findViewById(R.id.image_slider);

    ArrayList<SlideModel> slideModels = new ArrayList<>();

    slideModels.add(new SlideModel(R.drawable.displayproduk1,
ScaleTypes.CENTER_INSIDE));
    slideModels.add(new SlideModel(R.drawable.displayproduk2,
ScaleTypes.CENTER_INSIDE));
    slideModels.add(new SlideModel(R.drawable.displayproduk3,
ScaleTypes.CENTER_INSIDE));
    slideModels.add(new SlideModel(R.drawable.displayproduk4,
ScaleTypes.CENTER_INSIDE));
    imageSlider.setImageList(slideModels, ScaleTypes.FIT);

    //DeklarasiBeranda
    TextView tvJudul = rootView.findViewById(R.id.judulberanda);
    TextView tvDeksripsi =
rootView.findViewById(R.id.deksripsiberanda);
    ImageView gambarberanda =
rootView.findViewById(R.id.gambarberanda);

```

```

        InterfaceBeranda interfaceBeranda =
RetroServer.KonesiAPI().create(InterfaceBeranda.class);
        Call<ModelBeranda> tampilBeranda =
interfaceBeranda.getBeranda(3);
        tampilBeranda.enqueue(new Callback<ModelBeranda>() {
            @Override
            public void onResponse(Call<ModelBeranda> call,
Response<ModelBeranda> response) {
                if (response.isSuccessful()) {
                    ModelBeranda modelBeranda = response.body();
                    if (!isDetached() && getActivity() != null){
                        tvJudul.setText(modelBeranda.getJudul());
                        tvDeksripsi.setText(modelBeranda.getDeskrips
i());
                                Glide.with(fragmentContext).load(modelBerand
a.getFoto()).into(gambarberanda);
                    }
                } else {
                }
            }

            @Override
            public void onFailure(Call<ModelBeranda> call, Throwable
t) {
            }
        });

        //TAMPIL DEMPLOT
        rvDemplot = rootView.findViewById(R.id.rv_tampildemplot);
        lmDemplot = new LinearLayoutManager(getActivity(),
LinearLayoutManager.HORIZONTAL, false);
        rvDemplot.setLayoutManager(lmDemplot);
        InterfaceReport interfaceReport =
RetroServer.KonesiAPI().create(InterfaceReport.class);
        Call<ModelReport> tampildemplot =
interfaceReport.getReport();
        tampildemplot.enqueue(new Callback<ModelReport>() {
            @Override
            public void onResponse(Call<ModelReport> call,
Response<ModelReport> response) {
                ArrayList<ModelReport.report> listDemplot =
response.body().getReport();
                adDemplot = new
DataTampilDemplotHomePage(getActivity(), listDemplot);
                rvDemplot.setAdapter(adDemplot);
                adDemplot.notifyDataSetChanged();
            }
        });

```

```

        @Override
        public void onFailure(Call<ModelReport> call, Throwable
t) {}
    });

    //TAMPIL BERITA
    //Deklarasi
    rvBerita = rootView.findViewById(R.id.rv_berita);
    lmBerita = new LinearLayoutManager(getActivity(),
LinearLayoutManager.VERTICAL,true);
    rvBerita.setLayoutManager(lmBerita);

    InterfaceBerita interfaceBerita =
RetroServer.KonesiAPI().create(InterfaceBerita.class);
    Call<ModelBerita> tampilBerita =
interfaceBerita.getBerita();
    tampilBerita.enqueue(new Callback<ModelBerita>() {
        @Override
        public void onResponse(Call<ModelBerita> call,
Response<ModelBerita> response) {
            ArrayList<ModelBerita.data_berita> listBerita=
response.body().getData_berita();
            adBerita = new DataBerita(getActivity(),listBerita);
            rvBerita.setAdapter(adBerita);
            adBerita.notifyDataSetChanged();
        }

        @Override
        public void onFailure(Call<ModelBerita> call, Throwable
t) {}
    });

    return rootView;
}
@Override
public void onAttach(@NonNull Context context) {
    super.onAttach(context);
    fragmentContext = context;
}
}

```

SourceCode 1. 65 Fragment Home

3) Fragment Other

```
package com.example.cakraagroindonesia.ActivityCustomer.Fragment;

import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Login;
import com.example.cakraagroindonesia.Adapter.DataAlamat;
import com.example.cakraagroindonesia.Interface.InterfaceAlamat;
import
com.example.cakraagroindonesia.Interface.InterfaceTentangKami;
import com.example.cakraagroindonesia.Model.ModelAlamat;
import com.example.cakraagroindonesia.Model.ModelTentangKami;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class FragmentOther extends Fragment{
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater,
@Nullable ViewGroup container, @Nullable Bundle savedInstanceState)
    {
        View rootView = inflater.inflate(R.layout.fragment_other,
container, false);
```

```

        //DeklatasiTentangKami
        TextView tvDeskripsi =
rootView.findViewById(R.id.deskripsi);
        TextView tvVisi = rootView.findViewById(R.id.visi);
        TextView tvMisi = rootView.findViewById(R.id.misi);
        //Deklarasi Sosial Media
        ImageView fb = rootView.findViewById(R.id.btnfb);
        ImageView ig = rootView.findViewById(R.id.btnig);
        ImageView yt = rootView.findViewById(R.id.btnyt);

        InterfaceTentangKami interfaceTentangKami =
RetroServer.KonesiAPI().create(InterfaceTentangKami.class);
        InterfaceAlamat interfaceAlamat =
RetroServer.KonesiAPI().create(InterfaceAlamat.class);

        Call<ModelTentangKami> tampilTentangKami =
interfaceTentangKami.getTentangKami(1);
        tampilTentangKami.enqueue(new
Callback<ModelTentangKami>() {
            @Override
            public void onResponse(Call<ModelTentangKami>
tampilTentangKami, Response<ModelTentangKami> response) {
                if (response.isSuccessful()) {
                    ModelTentangKami modelTentangKami =
response.body();

                    String deskripsi =
modelTentangKami.getDeskripsi();
                    String visi = modelTentangKami.getVisi();
                    String misi = modelTentangKami.getMisi();

                    tvDeskripsi.setText(deskripsi);
                    tvVisi.setText(visi);
                    tvMisi.setText(misi);
                } else {}
            }
            @Override
            public void onFailure(Call<ModelTentangKami>
tampilTentangKami, Throwable t) {}
        });

        recyclerView = rootView.findViewById(R.id.rv_alamat);

```

```

        layoutManager = new
LinearLayoutManager(getActivity(),LinearLayoutManager.VERTICAL,false
);
        recyclerView.setLayoutManager(layoutManager);

        Call<ModelAlamat> tampilAlamat =
interfaceAlamat.getAlamat();
        tampilAlamat.enqueue(new Callback<ModelAlamat>() {
            @Override
            public void onResponse(Call<ModelAlamat> call,
Response<ModelAlamat> response) {
                ArrayList<ModelAlamat.alamat> listData =
response.body().getDataAlamat();
                adapter = new DataAlamat(getActivity(),listData);
                recyclerView.setAdapter(adapter);
                adapter.notifyDataSetChanged();
            }

            @Override
            public void onFailure(Call<ModelAlamat> call, Throwable
t) {

            }
        });

        //Aktivitas Login
        TextView btnLogin = rootView.findViewById(R.id.btnlogin);
        btnLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent login = new Intent(getActivity(),
Login.class);
                startActivity(login);
            }
        });

        fb.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast.makeText(getActivity(), "FACEBOOK",
Toast.LENGTH_SHORT).show();
            }
        });
        ig.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

```



```

        Toast.makeText(getActivity(), "INSTAGRAM",
Toast.LENGTH_SHORT).show();
    }
});
yt.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Toast.makeText(getActivity(), "YOUTUBE",
Toast.LENGTH_SHORT).show();
    }
});

return rootView;
}
}

```

SourceCode 1. 66 Fragment Other

4) Fragment Produk

```

package com.example.cakraagroindonesia.ActivityCustomer.Fragment;

import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

import com.example.cakraagroindonesia.ActivityCustomer.Fungisida;
import com.example.cakraagroindonesia.ActivityCustomer.Herbisida;
import com.example.cakraagroindonesia.ActivityCustomer.Insektisida;
import com.example.cakraagroindonesia.ActivityCustomer.Pgr;
import com.example.cakraagroindonesia.R;

public class FragmentProduk extends Fragment {
    @Nullable
    @Override

```

```

    public View onCreateView(@NonNull LayoutInflater inflater,
@Nullable ViewGroup container, @Nullable Bundle savedInstanceState)
{

    View rootView =
inflater.inflate(R.layout.fragment_produk, container, false);

    TextView insektisida = (TextView)
rootView.findViewById(R.id.btninsektisida);
    TextView fungsinya = (TextView)
rootView.findViewById(R.id.btnfungsinya);
    TextView herbisida = (TextView)
rootView.findViewById(R.id.btnherbisida);
    TextView pgr = (TextView)
rootView.findViewById(R.id.btnpgr);

    insektisida.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent insektisida = new Intent(getActivity(),
Insektisida.class);
            startActivity(insektisida);
        }
    });

    fungsinya.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent fungsinya = new Intent(getActivity(),
Fungsinya.class);
            startActivity(fungsinya);
        }
    });

    herbisida.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent herbisida = new Intent(getActivity(),
Herbisida.class);
            startActivity(herbisida);
        }
    });

    pgr.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent pgr = new Intent(getActivity(), Pgr.class);

```

```

        startActivity(pgr);
    }
});

    return rootView;
}
}

```

SourceCode 1. 67 Fragment Produk

b) Aktivitas

```

package com.example.cakraagroindonesia.ActivityCustomer;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Adapter.DataTampilAktivitas;
import com.example.cakraagroindonesia.Interface.InterfaceAktivitas;
import com.example.cakraagroindonesia.Model.ModelAktivitas;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class Aktivitas extends AppCompatActivity {
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_aktivitas);

        recyclerView = findViewById(R.id.rv_aktivitas);
        layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL, false);
        recyclerView.setLayoutManager(layoutManager);
    }
}

```

```

        InterfaceAktivitas interfaceAktivitas =
RetroServer.KonesiAPI().create(InterfaceAktivitas.class);
        Call<ModelAktivitas> tampil =
interfaceAktivitas.getAktivitas();
        tampil.enqueue(new Callback<ModelAktivitas>() {
            @Override
            public void onResponse(Call<ModelAktivitas> call,
Response<ModelAktivitas> response) {
                ArrayList<ModelAktivitas.aktivitas> listData =
response.body().getAktivitas();
                adapter = new DataTampilAktivitas(Aktivitas.this,
listData);
                recyclerView.setAdapter(adapter);
                adapter.notifyDataSetChanged();
            }

            @Override
            public void onFailure(Call<ModelAktivitas> call,
Throwable t) {

            }
        });
    }
}

```

SourceCode 1. 68 Aktivitas

c) Demplot

```

package com.example.cakraagroindonesia.ActivityCustomer;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Adapter.DataTampilAktivitas;
import com.example.cakraagroindonesia.Adapter.DataTampilDemplot;
import com.example.cakraagroindonesia.Interface.InterfaceAktivitas;
import
com.example.cakraagroindonesia.Interface.InterfaceDemonstrator;
import com.example.cakraagroindonesia.Interface.InterfaceReport;
import com.example.cakraagroindonesia.Model.ModelAktivitas;
import com.example.cakraagroindonesia.Model.ModelDemonstrator;
import com.example.cakraagroindonesia.Model.ModelReport;

```

```

import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class Demplot extends AppCompatActivity {
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_demplot);

        recyclerView = findViewById(R.id.rv_demplot);
        layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL, false);
        recyclerView.setLayoutManager(layoutManager);

        InterfaceReport interfaceReport =
RetroServer.KonesiAPI().create(InterfaceReport.class);
        Call<ModelReport> tampil = interfaceReport.getReport();
        tampil.enqueue(new Callback<ModelReport>() {
            @Override
            public void onResponse(Call<ModelReport> call,
Response<ModelReport> response) {
                ArrayList<ModelReport.report> listData =
response.body().getReport();
                adapter = new DataTampilDemplot(Demplot.this,
listData);
                recyclerView.setAdapter(adapter);
                adapter.notifyDataSetChanged();
            }

            @Override
            public void onFailure(Call<ModelReport> call, Throwable
t) {

            }

        });
    }
}

```

SourceCode 1. 69 Demplot

d) Detail Produk

```
package com.example.cakraagroindonesia.ActivityCustomer;

import android.os.Bundle;
import android.text.Html;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.R;

public class DetailProduk extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detail_produk);

        ImageView gambar = findViewById(R.id.gambarproduk);
        TextView merk = findViewById(R.id.merk);
        TextView penjelasanProduk =
findViewById(R.id.penjelasanproduk);

        String Gambar = getIntent().getStringExtra("gambar");
        String Merk = getIntent().getStringExtra("merk");
        String PenjelasanProduk =
getIntent().getStringExtra("penjelasanProduk");

        Glide.with(gambar).load(Gambar).into(gambar);
        merk.setText(Merk);
        penjelasanProduk.setText(Html.fromHtml(PenjelasanProduk));
    }
}
```

SourceCode 1. 70 Detail Produk

e) FAQ

```
package com.example.cakraagroindonesia.ActivityCustomer;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
```

```

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Adapter.DataFaq;
import com.example.cakraagroindonesia.Interface.InterfaceFaq;
import com.example.cakraagroindonesia.Model.ModelFaq;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class FAQ extends AppCompatActivity {
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_faq);

        TextView btn = findViewById(R.id.btnfaq);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(FAQ.this, QNA.class);
                startActivity(intent);
            }
        });

        recyclerView = findViewById(R.id.rv_faq);
        layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL, true);
        recyclerView.setLayoutManager(layoutManager);

        InterfaceFaq interfaceFaq =
RetroServer.KonesiAPI().create(InterfaceFaq.class);
        Call<ModelFaq> tampil = interfaceFaq.getFaq();
        tampil.enqueue(new Callback<ModelFaq>() {
            @Override

```

```

        public void onResponse(Call<ModelFaq> call,
Response<ModelFaq> response) {
            ArrayList<ModelFaq.data_faq> listData =
response.body().getData_faq();
            adapter = new DataFaq(FAQ.this, listData);
            recyclerView.setAdapter(adapter);
            adapter.notifyDataSetChanged();
        }

        @Override
        public void onFailure(Call<ModelFaq> call, Throwable t)
{
            }
        });
    }
}

```

SourceCode 1. 71 FAQ

f) Fungisida

```

package com.example.cakraagroindonesia.ActivityCustomer;

import android.content.Intent;
import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.GridLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Adapter.DataFungisida;
import com.example.cakraagroindonesia.Interface.InterfaceFungisida;
import com.example.cakraagroindonesia.Model.ModelFungisida;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class Fungisida extends AppCompatActivity implements
DataFungisida.OnItemClickListener {
    private RecyclerView recyclerView;

```



```

private RecyclerView.Adapter adapter;
private RecyclerView.LayoutManager layoutManager;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_fungisida);

    recyclerView = findViewById(R.id.rv_fungisida);
    layoutManager = new GridLayoutManager(this, 2);
    recyclerView.setLayoutManager(layoutManager);

    InterfaceFungisida interfaceFungisida =
RetroServer.KonesiAPI().create(InterfaceFungisida.class);
    Call<ModelFungisida> tampil =
interfaceFungisida.getDataFungisida();
    tampil.enqueue(new Callback<ModelFungisida>() {
        @Override
        public void onResponse(Call<ModelFungisida> call,
Response<ModelFungisida> response) {
            ArrayList<ModelFungisida.data_fungisida> listData =
response.body().getData_fungisida();
            adapter = new DataFungisida(Fungisida.this,
listData, Fungisida.this);
            recyclerView.setAdapter(adapter);
            adapter.notifyDataSetChanged();
        }

        @Override
        public void onFailure(Call<ModelFungisida> call,
Throwable t) {

        }
    });
}

@Override
public void onItemClick(ModelFungisida.data_fungisida
modelFungisida) {
    Intent detail = new Intent(Fungisida.this,
DetailProduk.class);
    detail.putExtra("gambar", modelFungisida.getBrowsure_url());
    detail.putExtra("merk", modelFungisida.getMerk());
    detail.putExtra("penjelasanProduk",
modelFungisida.getPenjelasan_produk());
    startActivity(detail);
}
}

```

```
}
```

SourceCode 1. 72 Fungsida

g) Herbisida

```
package com.example.cakraagroindonesia.ActivityCustomer;

import android.content.Intent;
import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.GridLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Adapter.DataHerbisida;
import com.example.cakraagroindonesia.Interface.InterfaceHerbisida;
import com.example.cakraagroindonesia.Model.ModelHerbisida;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class Herbisida extends AppCompatActivity implements
DataHerbisida.OnItemClickListener {
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_herbisida);

        recyclerView = findViewById(R.id.rv_herbisida);
        layoutManager = new GridLayoutManager(this, 2);
        recyclerView.setLayoutManager(layoutManager);

        InterfaceHerbisida interfaceHerbisida =
RetroServer.KonesiAPI().create(InterfaceHerbisida.class);
        Call<ModelHerbisida> tampil =
interfaceHerbisida.getDataHerbisida();
```

```

        tampil.enqueue(new Callback<ModelHerbisida>() {
            @Override
            public void onResponse(Call<ModelHerbisida> call,
Response<ModelHerbisida> response) {
                ArrayList<ModelHerbisida.data_herbisida> listData =
response.body().getData_herbisida();
                adapter = new DataHerbisida(Herbisida.this,
listData, Herbisida.this);
                recyclerView.setAdapter(adapter);
                adapter.notifyDataSetChanged();
            }

            @Override
            public void onFailure(Call<ModelHerbisida> call,
Throwable t) {

            }

        });
    }

    @Override
    public void onItemClick(@NonNull ModelHerbisida.data_herbisida
modelHerbisida) {
        Intent detail = new Intent(Herbisida.this,
DetailProduk.class);
        detail.putExtra("gambar", modelHerbisida.getBrowsure_url());
        detail.putExtra("merk", modelHerbisida.getMerk());
        detail.putExtra("penjelasanProduk",
modelHerbisida.getPenjelasan_produk());
        startActivity(detail);
    }
}

```

SourceCode 1. 73 Herbisida

h) Insektisida

```

package com.example.cakraagroindonesia.ActivityCustomer;

import android.content.Intent;
import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.GridLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;

```

```

import com.example.cakraagroindonesia.Adapter.DataInsektisida;
import
com.example.cakraagroindonesia.Interface.InterfaceInsektisida;
import com.example.cakraagroindonesia.Model.ModelInsektisida;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class Insektisida extends AppCompatActivity implements
DataInsektisida.OnItemClickListener{
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_insektisida);

        recyclerView = findViewById(R.id.rv_insektisida);
        layoutManager = new GridLayoutManager(this,2);
        recyclerView.setLayoutManager(layoutManager);

        InterfaceInsektisida interfaceInsektisida =
RetroServer.KonesiAPI().create(InterfaceInsektisida.class);
        Call<ModelInsektisida> tampil =
interfaceInsektisida.getDataInsektisida();
        tampil.enqueue(new Callback<ModelInsektisida>() {
            @Override
            public void onResponse(Call<ModelInsektisida> call,
Response<ModelInsektisida> response) {
                ArrayList<ModelInsektisida.data_insektisida>
listData = response.body().getData_insektisida();
                adapter = new DataInsektisida(Insektisida.this,
listData, Insektisida.this);
                recyclerView.setAdapter(adapter);
                adapter.notifyDataSetChanged();
            }

            @Override
            public void onFailure(Call<ModelInsektisida> call,
Throwable t) {

```

```

        }
    });

}

@Override
public void onItemClick(ModelInsektisida.data_insektisida
modelinsektisida) {
    Intent detail = new Intent(Insektisida.this,
DetailProduk.class);
    detail.putExtra("gambar",
modelinsektisida.getBrowsure_url());
    detail.putExtra("merk", modelinsektisida.getMerk());
    detail.putExtra("penjelasanProduk",
modelinsektisida.getPenjelasan_produk());
    startActivity(detail);
}
}

```

SourceCode 1. 74 Insektisida

i) Paket Produk

```

package com.example.cakraagroindonesia.ActivityCustomer;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Adapter.DataPaketProduk;
import
com.example.cakraagroindonesia.Interface.InterfacePaketProduk;
import com.example.cakraagroindonesia.Interface.OnClickPaketProduk;
import com.example.cakraagroindonesia.Model.ModelPaketProduk;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

```

```

public class PaketProduk extends AppCompatActivity implements
OnClickPaketProduk {
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_paket_produk);

        recyclerView = findViewById(R.id.rv_paketproduk);
        layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL,false);
        recyclerView.setLayoutManager(layoutManager);

        InterfacePaketProduk interfacePaketProduk =
RetroServer.KonesiAPI().create(InterfacePaketProduk.class);
        Call<ModelPaketProduk> tampil =
interfacePaketProduk.getPaketProduk();
        tampil.enqueue(new Callback<ModelPaketProduk>() {
            @Override
            public void onResponse(Call<ModelPaketProduk> call,
Response<ModelPaketProduk> response) {
                ArrayList<ModelPaketProduk.paket_produk> listData =
response.body().getPaket_produk();
                adapter = new DataPaketProduk(PaketProduk.this,
listData, PaketProduk.this);
                recyclerView.setAdapter(adapter);
                adapter.notifyDataSetChanged();
            }
            @Override
            public void onFailure(Call<ModelPaketProduk> call,
Throwable t) {}
        });

    }
    @Override
    public void onItemClick(ModelPaketProduk.paket_produk
paketProduk) {
    }
}

```

SourceCode 1. 75 Paket Produk

j) Pgr

```
package com.example.cakraagroindonesia.ActivityCustomer;

import android.content.Intent;
import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.GridLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Adapter.DataPgr;
import com.example.cakraagroindonesia.Interface.InterfacePgr;
import com.example.cakraagroindonesia.Model.ModelPgr;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class Pgr extends AppCompatActivity implements
DataPgr.OnItemClickListener {
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_pgr);

        recyclerView = findViewById(R.id.rv_pgr);
        layoutManager = new GridLayoutManager(this, 2);
        recyclerView.setLayoutManager(layoutManager);

        InterfacePgr interfacePgr =
RetroServer.KonesiAPI().create(InterfacePgr.class);
        Call<ModelPgr> tampil = interfacePgr.getDataPgr();
        tampil.enqueue(new Callback<ModelPgr>() {
            @Override
            public void onResponse(Call<ModelPgr> call,
Response<ModelPgr> response) {
```

```

        ArrayList<ModelPgr.data_pgr> listData =
response.body().getData_pgr();
        adapter = new DataPgr(Pgr.this, listData, Pgr.this);
        recyclerView.setAdapter(adapter);
        adapter.notifyDataSetChanged();
    }

    @Override
    public void onFailure(Call<ModelPgr> call, Throwable t)
{
    }
});
}

@Override
public void onItemClick(ModelPgr.data_pgr modelPgr) {
    Intent detail = new Intent(Pgr.this, DetailProduk.class);
    detail.putExtra("gambar", modelPgr.getBrowsure_url());
    detail.putExtra("merk", modelPgr.getMerk());
    detail.putExtra("penjelasanProduk",
modelPgr.getPenjelasan_produk());
    startActivity(detail);
}
}
}

```

SourceCode 1. 76 Pgr

k) Produk Baru

```

package com.example.cakraagroindonesia.ActivityCustomer;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Adapter.DataProdukBaru;
import com.example.cakraagroindonesia.Interface.InterfaceProdukBaru;
import com.example.cakraagroindonesia.Model.ModelProdukBaru;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

```



```

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class ProdukBaru extends AppCompatActivity {
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_produk_baru);

        recyclerView = findViewById(R.id.rv_produkbaru);
        layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL, false);
        recyclerView.setLayoutManager(layoutManager);

        InterfaceProdukBaru interfaceProdukBaru =
RetroServer.KonesiAPI().create(InterfaceProdukBaru.class);
        Call<ModelProdukBaru> tampil =
interfaceProdukBaru.getProdukBaru();
        tampil.enqueue(new Callback<ModelProdukBaru>() {
            @Override
            public void onResponse(Call<ModelProdukBaru> call,
Response<ModelProdukBaru> response) {
                ArrayList<ModelProdukBaru.produk_baru> listData =
response.body().getProduk_baru();
                adapter = new DataProdukBaru(ProdukBaru.this,
listData);
                recyclerView.setAdapter(adapter);
                adapter.notifyDataSetChanged();
            }

            @Override
            public void onFailure(Call<ModelProdukBaru> call,
Throwable t) {}
        });
    }
}

```

SourceCode 1. 77 Produk Baru

l) QnA

```
package com.example.cakraagroindonesia.ActivityCustomer;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Activity.Pertanyaan.TambahQna;
import com.example.cakraagroindonesia.Adapter.DataQna;
import com.example.cakraagroindonesia.Interface.InterfaceQna;
import com.example.cakraagroindonesia.Model.ModelQna;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;
import java.util.Collections;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class QNA extends AppCompatActivity {
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_qna);

        TextView btnTanya = findViewById(R.id.btntambahqna);

        btnTanya.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(QNA.this,
TambahQna.class));
            }
        });
    }
}
```

```

//TAMPIL DATA
recyclerView = findViewById(R.id.rv_qna);
layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL,false);
recyclerView.setLayoutManager(layoutManager);

InterfaceQna interfaceQna =
RetroServer.KonesiAPI().create(InterfaceQna.class);
Call<ModelQna> tampil = interfaceQna.getQna();
tampil.enqueue(new Callback<ModelQna>() {
    @Override
    public void onResponse(Call<ModelQna> call,
Response<ModelQna> response) {
        ArrayList<ModelQna.data_qna> listData =
response.body().getData_qna();
        adapter = new DataQna(QNA.this, listData);
        recyclerView.setAdapter(adapter);
        Collections.reverse(listData);
        adapter.notifyDataSetChanged();
    }

    @Override
    public void onFailure(Call<ModelQna> call, Throwable t)
{

    }
});
}
}

```

SourceCode 1. 78 QnA

3. Activity Manager

a) Dashboard Manager

```

package com.example.cakraagroindonesia.ActivityManager;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

```

```

import
com.example.cakraagroindonesia.DataMaster.ProfileMaster.ProfileManag
er;
import com.example.cakraagroindonesia.DataMaster.Demonstrator;
import com.example.cakraagroindonesia.DataMaster.Distributor;
import com.example.cakraagroindonesia.DataMaster.Secretary;
import com.example.cakraagroindonesia.DataMaster.Supervisor;
import com.example.cakraagroindonesia.R;

public class DashboardManager extends AppCompatActivity {
    TextView
    infomanager,infosupervisor,infosecretary,infodemonstrator,infodistri
    butor;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dashboard_manager);

        infomanager = findViewById(R.id.infomanager);
        infosupervisor = findViewById(R.id.infosupervisor);
        infodemonstrator = findViewById(R.id.infodemonstrator);
        infosecretary = findViewById(R.id.infosecretary);
        infodistributor = findViewById(R.id.infodistributor);

        infomanager.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(DashboardManager.this,
ProfileManager.class));
            }
        });
        infosupervisor.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View view) {
                startActivity(new Intent(DashboardManager.this,
Supervisor.class));
            }
        });
        infosecretary.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View view) {
                startActivity(new Intent(DashboardManager.this,
Secretary.class));
            }
        });
    }
}

```



```

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_kelola_manager);

        aktivitassupervisor =
findViewById(R.id.aktivitassupervisor);
        aktivitassupervisor.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaManager.this,
KelolaAktivitas.class));
            }
        });

        reportdemonstrator = findViewById(R.id.reportdemonstrator);
        reportdemonstrator.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(KelolaManager.this,
KelolaReport.class));
            }
        });

        bonusdistributor = findViewById(R.id.bonusdistributor);
        bonusdistributor.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast.makeText(KelolaManager.this, "Dalam Proses
Pengembangan ... ", Toast.LENGTH_SHORT).show();
            }
        });

        budgetingsupervisor =
findViewById(R.id.budgetingsupervisor);
        budgetingsupervisor.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast.makeText(KelolaManager.this, "Dalam Proses
Pengembangan ... ", Toast.LENGTH_SHORT).show();
            }
        });
    }
}

```

SourceCode 1. 80 Kelola Manager

c) Panel Manager

```
package com.example.cakraagroindonesia.ActivityManager;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.R;

public class PanelManager extends AppCompatActivity {
    TextView btnDashboard, btnKelolaManager;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_panel_manager);

        deklarasi();
        btnDashboard.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(PanelManager.this,
DashboardManager.class));
            }
        });

        btnKelolaManager.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(PanelManager.this,
KelolaManager.class));
            }
        });
    }

    private void deklarasi() {
        btnDashboard = findViewById(R.id.btndasbord);
        btnKelolaManager =findViewById(R.id.btnkelolamanager);
    }
}
```

SourceCode 1. 81 Panel Manager

4. Activity Secretary

a) Dashboard Secretary

```
package com.example.cakraagroindonesia.ActivitySecretary;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.DataMaster.Distributor;
import
com.example.cakraagroindonesia.DataMaster.ProfileMaster.ProfileSecre
tary;
import com.example.cakraagroindonesia.R;

public class DashboardSecretary extends AppCompatActivity {

    private TextView infoalessesecretary, infodistributor;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dashboard_secretary);

        infoalessesecretary = findViewById(R.id.infoalessesecretary);
        infoalessesecretary.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(DashboardSecretary.this,
ProfileSecretary.class));
            }
        });

        infodistributor = findViewById(R.id.infodistributor);
        infodistributor.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(DashboardSecretary.this,
Distributor.class));
            }
        });
    }
}
```



```
}
```

SourceCode 1. 82 Dashboard Secretary

b) Kelola Secretary

```
package com.example.cakraagroindonesia.ActivitySecretary;

import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.R;

public class KelolaSecretary extends AppCompatActivity {

    private TextView inputbonusdistributor,
    inforeviewbonusdistributor;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_kelola_secretary);

        inputbonusdistributor =
        findViewById(R.id.inputbonusdistributor);
        inputbonusdistributor.setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast.makeText(KelolaSecretary.this, "Masih Dalam
                Pengembangan ... ", Toast.LENGTH_SHORT).show();
            }
        });

        inforeviewbonusdistributor =
        findViewById(R.id.inforeviewbonusdistributor);
        inforeviewbonusdistributor.setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast.makeText(KelolaSecretary.this, "Masih Dalam
                Pengembangan ... ", Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

```
}  
}
```

SourceCode 1. 83 Kelola Secretary

c) Panel Secretary

```
package com.example.cakraagroindonesia.ActivitySecretary;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.TextView;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.example.cakraagroindonesia.R;  
  
public class PanelSecretary extends AppCompatActivity {  
  
    private TextView dashboard, kelolasecretary;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_panel_secretary);  
  
        dashboard = findViewById(R.id.dashboard);  
        dashboard.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                startActivity(new Intent(PanelSecretary.this,  
DashboardSecretary.class));  
            }  
        });  
  
        kelolasecretary = findViewById(R.id.kelolasecretarty);  
        kelolasecretary.setOnClickListener(new  
View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                startActivity(new Intent(PanelSecretary.this,  
DashboardSecretary.class));  
            }  
        });  
    }  
}
```

SourceCode 1. 84 Panel Secretary

5. Activity Supervisor

a) Dashboard Supervisor

```
package com.example.cakraagroindonesia.ActivitySupervisor;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.DataMaster.ProfileMaster.ProfileSupervisor;
import com.example.cakraagroindonesia.R;

public class DashboardSupervisor extends AppCompatActivity {

    private TextView infosupervisor, infodemonstrator;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dashboard_supervisor);

        infosupervisor = findViewById(R.id.infosupervisor);
        infodemonstrator.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(DashboardSupervisor.this,
ProfileSupervisor.class));
            }
        });
    }
}
```

SourceCode 1. 85 Dashboard Supervisor

b) Panel Supervisor

```
package com.example.cakraagroindonesia.ActivitySupervisor;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.R;

public class PanelSupervisor extends AppCompatActivity {

    private TextView dashboard, keloladatasupervisor;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_panel_supervisor);

        dashboard = findViewById(R.id.dashboard);
        dashboard.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(PanelSupervisor.this,
DashboardSupervisor.class));
            }
        });
    }
}
```

SourceCode 1. 86 Panel Supervisor

6. Adapter

a) Data Admin

```
package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
```

```

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import
com.example.cakraagroindonesia.Activity.KelolaDataMaster.UbahDataAdmin;
import com.example.cakraagroindonesia.HapusData;
import com.example.cakraagroindonesia.Interface.InterfaceAdmin;
import com.example.cakraagroindonesia.Model.ModelAdmin;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class DataAdmin extends
RecyclerView.Adapter<DataAdmin.HolderData> {
    private Context context;
    private ArrayList<ModelAdmin.data_admin> listData;

    private float currentRotation = 0;

    private static final int VIEW_TYPE_ITEM = 0;
    private static final int VIEW_TYPE_ITEM_LAST = 1;

    public DataAdmin(Context context,
ArrayList<ModelAdmin.data_admin> listData) {
        this.context = context;
        this.listData = listData;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        if (viewType == VIEW_TYPE_ITEM) {
            View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_kelolaadmin,parent,false);
            return new HolderData(layout);
        } else if (viewType == VIEW_TYPE_ITEM_LAST) {

```

```

        View footer =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_foote
r,parent,false);
        return new HolderData(footer);
    }

    throw new IllegalArgumentException("Invalid view type");

}

@Override
public void onBindViewHolder(@NonNull HolderData holder, int
position) {
    if (getItemViewType(position) == VIEW_TYPE_ITEM) {
        // Bind item data
        ModelAdmin.data_admin model = listData.get(position);

        holder>Nama.setText(model.getNama());
        holder.Id.setText(model.getId_admin());
        holder.Username.setText(model.getUsername());
        holder.Password.setText(model.getPassword());

        holder.btnEdit.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                InterfaceAdmin admin =
RetroServer.KonesiAPI().create(InterfaceAdmin.class);
                Call<ModelAdmin> kirim =
admin.getEdit(model.getId_admin());
                kirim.enqueue(new Callback<ModelAdmin>() {
                    @Override
                    public void onResponse(Call<ModelAdmin>
call, Response<ModelAdmin> response) {

                        String varId = model.getId_admin();
                        String varNama = model.getNama();
                        String varUsername =
model.getPassword();
                        String varPassword =
model.getPassword();

                        Intent kirim = new Intent(context,
UbahDataAdmin.class);
                        kirim.putExtra("xId", varId);
                        kirim.putExtra("xNama", varNama);

```

```

        kirim.putExtra("xUsername",
varUsername);
        kirim.putExtra("xPassword",
varPassword);
        context.startActivity(kirim);
    }

    @Override
    public void onFailure(Call<ModelAdmin> call,
Throwable t) {
    }
    });
    });
    holder.btnHapus.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String Data = "Admin";
        Intent hapus = new Intent(context,
HapusData.class);
        hapus.putExtra("IDhapus", model.getId_admin());
        hapus.putExtra("Data", Data);
        context.startActivity(hapus);
    }
    });
    } else if (getItemViewType(position) == VIEW_TYPE_ITEM_LAST)
{
    // Bind data for the last item
}
}

@Override
public int getItemCount() {
    return listData.size() + 1;
}

public class HolderData extends RecyclerView.ViewHolder{
    TextView Nama, Id, Username, Password;
    ImageView btnHapus, btnEdit;
    public HolderData(@NonNull View itemView) {
        super(itemView);

        Nama = itemView.findViewById(R.id.nama);
        Id = itemView.findViewById(R.id.id);
    }
}

```

```

        Username = itemView.findViewById(R.id.username);
        Password = itemView.findViewById(R.id.password);
        btnHapus = itemView.findViewById(R.id.btnhapus);
        btnEdit = itemView.findViewById(R.id.btncedit);
    }
}

public String getLastId(){
    if (listData != null && listData.size() > 0){
        int lastIndex = listData.size()-1;
        ModelAdmin.data_admin lastData =
listData.get(lastIndex);
        return lastData.getId_admin();
    }
    return null;
}

@Override
public int getItemViewType(int position) {
    if (position == listData.size()) {
        return VIEW_TYPE_ITEM_LAST; // Tampilan terakhir
    }
    return VIEW_TYPE_ITEM; // Tampilan item biasa
}
}
}

```

SourceCode 1. 87 Data Admin

b) Data Aktivitas

```

package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;

```



```

import com.example.cakraagroindonesia.API.RetroServer;
import
com.example.cakraagroindonesia.Activity.Aktivitas.UbahAktivitas;
import com.example.cakraagroindonesia.HapusData;
import com.example.cakraagroindonesia.Interface.InterfaceAktivitas;
import com.example.cakraagroindonesia.Model.ModelAktivitas;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class DataAktivitas extends
RecyclerView.Adapter<DataAktivitas.HolderData>{
    private Context context;
    private ArrayList<ModelAktivitas.aktivitas> listData;

    private float currentRotation;

    private static final int VIEW_TYPE_ITEM = 0;
    private static final int VIEW_TYPE_ITEM_LAST = 1;

    public DataAktivitas(Context context,
ArrayList<ModelAktivitas.aktivitas> listData) {
        this.context = context;
        this.listData = listData;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        if (viewType == VIEW_TYPE_ITEM) {
            View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_kelol
aaktivitas,parent,false);
            return new HolderData(layout);
        } else if (viewType == VIEW_TYPE_ITEM_LAST) {
            View footer =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_foote
r,parent,false);
            return new HolderData(footer);
        }

        throw new IllegalArgumentException("Invalid view type");
    }
}

```

```

    }

    @Override
    public void onBindViewHolder(@NonNull HolderData holder, int
position) {
        if (getItemViewType(position) == VIEW_TYPE_ITEM) {
            // Bind item data
            ModelAktivitas.aktivitas model = listData.get(position);

            holder.supervisor.setText(model.getNama_supervisor());
            holder.kegiatan.setText(model.getNama_kegiatan());
            holder.manager.setText(model.getNama_manager());
            holder.namakios.setText(model.getNama_kios());
            holder.produk.setText(model.getProduk());
            holder.partisipan.setText(model.getJumlah_partisipan());
            holder.budget.setText(model.getBudget());
            holder.lokasi.setText(model.getLokasi());
            holder.areasales.setText(model.getArea_sales());
            holder.provinsi.setText(model.getProvinsi());
            Glide.with(context).load(model.getFoto_url()).into(holder.foto);

            holder.btnEdit.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    InterfaceAktivitas kirim =
RetroServer.KonesiAPI().create(InterfaceAktivitas.class);
                    Call<ModelAktivitas> call =
kirim.getEdit(model.getKode_ak());
                    call.enqueue(new Callback<ModelAktivitas>() {
                        @Override
                        public void onResponse(Call<ModelAktivitas>
call, Response<ModelAktivitas> response) {
                            String varId = model.getKode_ak();
                            String varTgl =
model.getTanggal_kegiatan();
                            String varStatus = model.getStatus();
                            String varNamaKegiatan =
model.getNama_kegiatan();
                            String varNamaKios =
model.getNama_kios();
                            String varProduk = model.getProduk();
                            String varJumlahPartisipan =
model.getJumlah_partisipan();
                            String varBudget = model.getBudget();

```

```

        String varLokasi = model.getLokasi();
        String varAreaSales =
model.getArea_sales();
        String varProvinsi =
model.getProvinsi();
        String varStatusApps =
model.getStarus_apps();
        String varTFoto = model.getFoto_url();
        String varFoto =
model.getFoto_kegiatan();

        Intent kirim = new Intent(context,
UbahAktivitas.class);
        kirim.putExtra("xId", varId);
        kirim.putExtra("xTgl", varTgl);
        kirim.putExtra("xStatus", varStatus);
        kirim.putExtra("xNamaKegiatan",
varNamaKegiatan);
        kirim.putExtra("xNamaKios",
varNamaKios);
        kirim.putExtra("xProduk", varProduk);
        kirim.putExtra("xJumlahPartisipan",
varJumlahPartisipan);
        kirim.putExtra("xBudget", varBudget);
        kirim.putExtra("xLokasi", varLokasi);
        kirim.putExtra("xAreaSales",
varAreaSales);
        kirim.putExtra("xProvinsi",
varProvinsi);
        kirim.putExtra("xStatusApps",
varStatusApps);
        kirim.putExtra("xTFoto", varTFoto);
        kirim.putExtra("xFoto", varFoto);
        context.startActivity(kirim);
    }

    @Override
    public void onFailure(Call<ModelAktivitas>
call, Throwable t) {
    }
    });
    }
    });
    holder.btnHapus.setOnClickListener(new
View.OnClickListener() {
        @Override

```

```

        public void onClick(View view) {
            String Data = "Aktivitas";
            Intent hapus = new Intent(context,
HapusData.class);
            hapus.putExtra("IDhapus" , model.getKode_ak());
            hapus.putExtra("Data", Data);
            context.startActivity(hapus);
        }
    });
    holder.btn.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view) {
//        float currentRotation += 180;
        float currentRotation =
holder.btndetail.getRotation();
        int invisible = holder.detail.getVisibility();
        if (invisible == View.VISIBLE) {

            // Jika LinearLayout terlihat, ubah menjadi
GONE

            holder.detail.setVisibility(View.GONE);
            holder.btndetail.setRotation(0);
        } else {
            // Jika LinearLayout tidak terlihat, ubah
menjadi VISIBLE

            holder.detail.setVisibility(View.VISIBLE);
            holder.btndetail.setRotation(+180);
        }
    }
});
    } else if (getItemViewType(position) == VIEW_TYPE_ITEM_LAST)
{
    // Bind data for the last item
    }
}

@Override
public int getItemCount() {
    return listData.size() + 1;
}

public class HolderData extends RecyclerView.ViewHolder{
    TextView supervisor, kegiatan, manager, namakios, produk,
partisipan, budget, lokasi, areasales, provinsi;
    ImageView btnEdit, btnHapus, foto, btndetail;
    LinearLayout show, detail;
}

```

```

RelativeLayout btn;

public HolderData(@NonNull View itemView) {
    super(itemView);
    supervisor = itemView.findViewById(R.id.supervisor);
    kegiatan = itemView.findViewById(R.id.kegiatan);
    manager = itemView.findViewById(R.id.manager);
    namakios = itemView.findViewById(R.id.namakios);
    produk = itemView.findViewById(R.id.produk);
    partisipan = itemView.findViewById(R.id.partisipan);
    budget = itemView.findViewById(R.id.budget);
    lokasi = itemView.findViewById(R.id.lokasi);
    areasales = itemView.findViewById(R.id.areasales);
    provinsi = itemView.findViewById(R.id.provinsi);
    foto = itemView.findViewById(R.id.foto);
    btnEdit = itemView.findViewById(R.id.bnedit);
    btnHapus = itemView.findViewById(R.id.bnhapus);
    detail = itemView.findViewById(R.id.detail);
    btndetail = itemView.findViewById(R.id.btndetail);
    btn = itemView.findViewById(R.id.btn);
}
}
@Override
public int getItemViewType(int position) {
    if (position == listData.size()) {
        return VIEW_TYPE_ITEM_LAST; // Tampilan terakhir
    }
    return VIEW_TYPE_ITEM; // Tampilan item biasa
}
}

```

SourceCode 1. 88 Data Aktivitas

c) Data Alamat

```

package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

```

```

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.Model.ModelAlamat;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

public class DataAlamat extends
RecyclerView.Adapter<DataAlamat.HolderData>{
    private Context context;
    private ArrayList<ModelAlamat.alamat> listData;

    public DataAlamat(Context context, ArrayList<ModelAlamat.alamat>
listData) {
        this.context = context;
        this.listData = listData;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_alama
t,parent,false);
        return new HolderData(layout);
    }

    @Override
    public void onBindViewHolder(@NonNull HolderData holder, int
position) {
        ModelAlamat.alamat model = listData.get(position);

        holder.namakantor.setText(model.getNama_kantor());
        holder.alamat.setText(model.getAlamat());
        holder.telepon.setText(model.getTelepon());
        Glide.with(context).load(model.getFoto_url()).into(holder.fo
to);
    }

    @Override
    public int getItemCount() {
        return listData.size();
    }

    public static class HolderData extends RecyclerView.ViewHolder{
        TextView id, namakantor, alamat, telepon;
    }
}

```

```

        ImageView foto;
        public HolderData(@NonNull View itemView) {
            super(itemView);
            id = itemView.findViewById(R.id.idalamat);
            namakantor = itemView.findViewById(R.id.namakantor);
            alamat = itemView.findViewById(R.id.alamat);
            telepon = itemView.findViewById(R.id.telepon);
            foto = itemView.findViewById(R.id.foto);
        }
    }
}

```

SourceCode 1. 89 Data Alamat

d) Data Berita

```

package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.text.Html;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.Model.ModelBerita;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

public class DataBerita extends
RecyclerView.Adapter<DataBerita.HolderData>{
    private Context context;
    private ArrayList<ModelBerita.data_berita> listData;

    public DataBerita(Context context,
ArrayList<ModelBerita.data_berita> listData) {
        this.context = context;
        this.listData = listData;
    }

    @NonNull
    @Override

```

```

    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
    View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_berita,
parent, false);
    return new HolderData(layout);
    }

    @Override
    public void onBindViewHolder(@NonNull HolderData holder, int
position) {
    ModelBerita.data_berita model = listData.get(position);

    holder.tvIdBerita.setText(String.valueOf(model.getId_berita(
)));
    holder.tvJudulBerita.setText(model.getJudul_berita());
    // holder.tvIsiBerita.setText(model.getIsi_berita());
    holder.tvIsiBerita.setText(Html.fromHtml(model.getIsi_berita
()));
    }

    @Override
    public int getItemCount() {
    return listData.size();
    }

    public class HolderData extends RecyclerView.ViewHolder{
    TextView tvIdBerita, tvJudulBerita, tvIsiBerita;

    public HolderData(@NonNull View itemView) {
    super(itemView);

    tvIdBerita = itemView.findViewById(R.id.idberita);
    tvJudulBerita = itemView.findViewById(R.id.judulberita);
    tvIsiBerita = itemView.findViewById(R.id.isiberita);
    }
    }
}

```

SourceCode 1. 90 Data Berita

e) Data Demonstrator

```
package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.API.RetroServer;
import
com.example.cakraagroindonesia.Activity.KelolaDataMaster.UbahDemonst
rator;
import com.example.cakraagroindonesia.HapusData;
import
com.example.cakraagroindonesia.Interface.InterfaceDemonstrator;
import com.example.cakraagroindonesia.Model.ModelDemonstrator;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class DataDemonstrator extends
RecyclerView.Adapter<DataDemonstrator.HolderData>{
    private Context context;
    private ArrayList<ModelDemonstrator.demonstrator> listData;

    private float currentRotation = 0;

    private static final int VIEW_TYPE_ITEM = 0;
    private static final int VIEW_TYPE_ITEM_LAST = 1;

    public DataDemonstrator(Context context,
ArrayList<ModelDemonstrator.demonstrator> listData) {
```

```

        this.context = context;
        this.listData = listData;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        if (viewType == VIEW_TYPE_ITEM) {
            View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_kelol
ademonstrator,parent,false);
            return new HolderData(layout);
        } else if (viewType == VIEW_TYPE_ITEM_LAST) {
            View footer =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_foote
r,parent,false);
            return new HolderData(footer);
        }

        throw new IllegalArgumentException("Invalid view type");
    }

    @Override
    public void onBindViewHolder(@NonNull HolderData holder, int
position) {
        if (getItemViewType(position) == VIEW_TYPE_ITEM) {
            // Bind item data
            ModelDemonstrator.demonstrator model =
listData.get(position);

            holder.nama.setText(model.getNama_demonstrator());
            Glide.with(context).load(model.getFoto_url()).into(holde
r.foto);

            holder.id.setText(model.getKode_ds());
            holder.demonstrator.setText(model.getNama_demonstrator()
);

            holder.supervisor.setText(model.getNama_supervisor());
            holder.provinsi.setText(model.getProvinsi());
            holder.kabupaten.setText(model.getKabupaten());
            holder.username.setText(model.getUsername());
            holder.password.setText(model.getPassword());

            holder.btnEdit.setOnClickListener(new
View.OnClickListener() {
                @Override

```

```

        public void onClick(View view) {
            InterfaceDemonstrator interfaceDemonstrator =
RetroServer.KonesiAPI().create(InterfaceDemonstrator.class);
            Call<ModelDemonstrator> call =
interfaceDemonstrator.getEdit(model.getKode_ds());
            call.enqueue(new Callback<ModelDemonstrator>() {
                @Override
                public void
onResponse(Call<ModelDemonstrator> call, Response<ModelDemonstrator>
response) {
                    String varId = model.getKode_ds();
                    String varNama =
model.getNama_demonstrator();
                    String varProvinsi =
model.getProvinsi();
                    String varKabupaten =
model.getKabupaten();
                    String varUsername =
model.getUsername();
                    String varPassword =
model.getPassword();
                    String varTFoto = model.getFoto_url();
                    String varFoto =
model.getFotodemonstrator();

                    Intent kirim = new Intent(context,
UbahDemonstrator.class);
                    kirim.putExtra("xId", varId);
                    kirim.putExtra("xNama", varNama);
                    kirim.putExtra("xProvinsi",
varProvinsi);
                    kirim.putExtra("xKabupaten",
varKabupaten);
                    kirim.putExtra("xUsername",
varUsername);
                    kirim.putExtra("xPassword",
varPassword);
                    kirim.putExtra("xTFoto", varTFoto);
                    kirim.putExtra("xFoto", varFoto);
                    context.startActivity(kirim);
                }

                @Override
                public void
onFailure(Call<ModelDemonstrator> call, Throwable t) {

            }
        }
    }
}

```

```

        });
    }
});
holder.btnHapus.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String Data = "Demonstrator";

        Intent hapus = new Intent(context,
HapusData.class);
        hapus.putExtra("IDhapus", model.getKode_ds());
        hapus.putExtra("Data", Data);
        context.startActivity(hapus);
    }
});
holder.btn.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view) {
        int invisible = holder.detail.getVisibility();
        if (invisible == View.VISIBLE) {
            // Jika LinearLayout terlihat, ubah menjadi
GONE

            holder.detail.setVisibility(View.GONE);
            holder.btndetail.setRotation(0);
        } else {
            // Jika LinearLayout tidak terlihat, ubah
menjadi VISIBLE

            holder.detail.setVisibility(View.VISIBLE);
            holder.btndetail.setRotation(+180);
        }
    }
});
} else if (getItemViewType(position) == VIEW_TYPE_ITEM_LAST)
{
    // Bind data for the last item
}

}

@Override
public int getItemCount() {
    return listData.size() + 1;
}

public class HolderData extends RecyclerView.ViewHolder{

```

```

        TextView nama, id, demonstrator, supervisor, provinsi,
kabupaten, username, password;
        ImageView btnHapus, btnEdit, foto, btndetail;
        RelativeLayout btn;
        LinearLayout detail;
        public HolderData(@NonNull View itemView) {
            super(itemView);
            nama = itemView.findViewById(R.id.nama);
            foto = itemView.findViewById(R.id.foto);
            id = itemView.findViewById(R.id.id);
            supervisor = itemView.findViewById(R.id.supervisor);
            demonstrator = itemView.findViewById(R.id.demonstrator);
            kabupaten = itemView.findViewById(R.id.kabupaten);
            provinsi = itemView.findViewById(R.id.provinsi);
            username = itemView.findViewById(R.id.username);
            password = itemView.findViewById(R.id.password);
            btnEdit = itemView.findViewById(R.id.btndedit);
            btnHapus = itemView.findViewById(R.id.btnhapus);
            detail = itemView.findViewById(R.id.detail);
            btndetail = itemView.findViewById(R.id.btndetail);
            btn = itemView.findViewById(R.id.btn);
        }
    }
    @Override
    public int getItemViewType(int position) {
        if (position == listData.size()) {
            return VIEW_TYPE_ITEM_LAST; // Tampilan terakhir
        }
        return VIEW_TYPE_ITEM; // Tampilan item biasa
    }
    public String getLastId(){
        if (listData != null && listData.size() > 0){
            int lastIndex = listData.size()-1;
            ModelDemonstrator.demonstrator lastData =
listData.get(lastIndex);
            return lastData.getKode_ds();
        }
        return null;
    }
}
}

```

SourceCode 1. 91 Data Demonstrator

f) Data Distributor

```
package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.API.RetroServer;
import
com.example.cakraagroindonesia.Activity.KelolaDataMaster.UbahDistrib
utor;
import com.example.cakraagroindonesia.HapusData;
import
com.example.cakraagroindonesia.Interface.InterfaceDistributor;
import com.example.cakraagroindonesia.Model.ModelDistributor;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class DataDistributor extends
RecyclerView.Adapter<DataDistributor.HolderData>{
    private Context context;
    private ArrayList<ModelDistributor.distributor> listData;

    private float currentRotation = 0;

    private static final int VIEW_TYPE_ITEM = 0;
    private static final int VIEW_TYPE_ITEM_LAST = 1;

    public DataDistributor(Context context,
ArrayList<ModelDistributor.distributor> listData) {
        this.context = context;
```

```

        this.listData = listData;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        if (viewType == VIEW_TYPE_ITEM) {
            View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_kelol
adistributor,parent,false);
            return new HolderData(layout);
        } else if (viewType == VIEW_TYPE_ITEM_LAST) {
            View footer =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_foote
r,parent,false);
            return new HolderData(footer);
        }

        throw new IllegalArgumentException("Invalid view type");
    }

    @Override
    public void onBindViewHolder(@NonNull HolderData holder, int
position) {
        if (getItemViewType(position) == VIEW_TYPE_ITEM) {
            // Bind item data
            ModelDistributor.distributor model =
listData.get(position);

            holder.nama.setText(model.getNama_distributor());
            Glide.with(context).load(model.getFoto_url()).into(holde
r.foto);

            holder.id.setText(model.getKode_dt());
            holder.distributor.setText(model.getNama_distributor());
            holder.secretary.setText(model.getNama_secretary());
            holder.perusahaan.setText(model.getPerusahaan());
            holder.username.setText(model.getUsername());
            holder.password.setText(model.getPassword());

            holder.btnEdit.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    InterfaceDistributor interfaceDistributor =
RetroServer.KonesiAPI().create(InterfaceDistributor.class);

```

```

        Call<ModelDistributor> kirim =
interfaceDistributor.getEdit(model.getKode_dt());
        kirim.enqueue(new Callback<ModelDistributor>() {
            @Override
            public void
onResponse(Call<ModelDistributor> call, Response<ModelDistributor>
response) {
                String varId = model.getKode_dt();
                String varNama =
model.getNama_distributor();
                String varUsername =
model.getUsername();
                String varPassword =
model.getPassword();
                String varLevel = model.getLevel();
                String varKodesc = model.getKode_sc();
                String varNamasv =
model.getNama_secretary();
                String varTFoto = model.getFoto_url();
                String varFoto =
model.getFotodistributor();
                String varKodemg = model.getKode_mg();
                String varPerusahaan =
model.getPerusahaan();

                Intent kirim = new Intent(context,
UbahDistributor.class);

                kirim.putExtra("xId", varId);
                kirim.putExtra("xNama", varNama);
                kirim.putExtra("xUsername",
varUsername);
                kirim.putExtra("xPassword",
varPassword);
                kirim.putExtra("xLevel", varLevel);
                kirim.putExtra("xKodesc", varKodesc);
                kirim.putExtra("xNamasv", varNamasv);
                kirim.putExtra("xTFoto", varTFoto);
                kirim.putExtra("xFoto", varFoto);
                kirim.putExtra("xKodemg", varKodemg);
                kirim.putExtra("xPerusahaan",
varPerusahaan);

                context.startActivity(kirim);
            }
            @Override
            public void onFailure(Call<ModelDistributor>
call, Throwable t) {

```



```

        }
    });
}
});
holder.btnHapus.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String Data = "Distributor";
        Intent hapus = new Intent(context,
HapusData.class);
        hapus.putExtra("IDhapus", model.getKode_dt());
        hapus.putExtra("Data", Data);
        context.startActivity(hapus);
    }
});
holder.btn.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view) {
        rotateImage(view);
        int invisible = holder.detail.getVisibility();
        if (invisible == View.VISIBLE) {
            // Jika LinearLayout terlihat, ubah menjadi
GONE
            holder.detail.setVisibility(View.GONE);
        } else {
            // Jika LinearLayout tidak terlihat, ubah
menjadi VISIBLE
            holder.detail.setVisibility(View.VISIBLE);
        }
    }
    public void rotateImage(View view) {
        currentRotation += 180;
        holder.btndetail.setRotation(currentRotation);
    }
});
} else if (getItemViewType(position) == VIEW_TYPE_ITEM_LAST)
{
    // Bind data for the last item
}
}

@Override
public int getItemCount() {
    return listData.size() + 1;
}
}

```

```

    public class HolderData extends RecyclerView.ViewHolder{
        TextView nama, id, distributor, secretary, perusahaan,
username, password;
        ImageView btnHapus, btnEdit,foto, btndetail;
        RelativeLayout btn;
        LinearLayout show,detail;
        public HolderData(@NonNull View itemView) {
            super(itemView);

            nama = itemView.findViewById(R.id.nama);
            foto = itemView.findViewById(R.id.foto);
            id = itemView.findViewById(R.id.id);
            distributor = itemView.findViewById(R.id.distributor);
            secretary = itemView.findViewById(R.id.secretary);
            perusahaan = itemView.findViewById(R.id.perusahaan);
            username = itemView.findViewById(R.id.username);
            password = itemView.findViewById(R.id.password);
            btnEdit = itemView.findViewById(R.id.btncedit);
            btnHapus = itemView.findViewById(R.id.btnhapus);
            detail = itemView.findViewById(R.id.detail);
            btndetail = itemView.findViewById(R.id.btndetail);
            btn = itemView.findViewById(R.id.btn);
        }
    }
    @Override
    public int getItemViewType(int position) {
        if (position == listData.size()) {
            return VIEW_TYPE_ITEM_LAST; // Tampilan terakhir
        }
        return VIEW_TYPE_ITEM; // Tampilan item biasa
    }
    public String getLastId(){
        if (listData != null && listData.size() > 0){
            int lastIndex = listData.size()-1;
            ModelDistributor.distributor lastData =
listData.get(lastIndex);
            return lastData.getKode_dt();
        }
        return null;
    }
}
}

```

SourceCode 1. 92 Data Distributor

g) Data FAQ

```
package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.LinearLayout;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.Model.ModelFaq;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

public class DataFaq extends
RecyclerView.Adapter<DataFaq.HolderData>{
    private Context context;
    private ArrayList<ModelFaq.data_faq> listData;

    public DataFaq(Context context, ArrayList<ModelFaq.data_faq>
listData) {
        this.context = context;
        this.listData = listData;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_faq,p
arent,false);
        return new HolderData(layout);
    }

    @Override
    public void onBindViewHolder(@NonNull HolderData holder, int
position) {
        ModelFaq.data_faq model = listData.get(position);

        holder.pertanyaan.setText(model.getPertanyaan());
        holder.jawaban.setText(model.getJawaban());
    }
}
```

```

        holder.pertanyaan.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        int invisible = holder.detail.getVisibility();
        if (invisible == View.VISIBLE) {
            // Jika LinearLayout terlihat, ubah menjadi GONE
            holder.detail.setVisibility(View.GONE);
        } else {
            // Jika LinearLayout tidak terlihat, ubah
menjadi VISIBLE
            holder.detail.setVisibility(View.VISIBLE);
        }
    }
});
}

@Override
public int getItemCount() {
    return listData.size();
}

public class HolderData extends RecyclerView.ViewHolder{
    TextView pertanyaan, jawaban;
    LinearLayout detail;

    public HolderData(@NonNull View itemView) {
        super(itemView);
        pertanyaan = itemView.findViewById(R.id.pertanyaan);
        jawaban = itemView.findViewById(R.id.jawaban);
        detail = itemView.findViewById(R.id.detail);
    }
}
}
}

```

SourceCode 1. 93 Data FAQ

h) Data Fungisisda

```

package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

```

```

import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.Model.ModelFungsida;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

public class DataFungsida extends
RecyclerView.Adapter<DataFungsida.HolderData> {
    private Context context;
    private ArrayList<ModelFungsida.data_fungsida> listData;
    private OnItemClickListener onItemClicked;

    public interface OnItemClickListener{
        void onItemClick(ModelFungsida.data_fungsida
modelFungsida);
    }

    public DataFungsida(Context context,
ArrayList<ModelFungsida.data_fungsida> listData, OnItemClickListener
onItemClicked) {
        this.context = context;
        this.listData = listData;
        this.onItemClicked = onItemClicked;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_fungi
sida,parent,false);
        return new HolderData(layout);
    }

    @Override
    public void onBindViewHolder(@NonNull HolderData holder, int
position) {
        ModelFungsida.data_fungsida model = listData.get(position);

        holder.id.setText(String.valueOf(model.getId_fungsida()));

```

```

        holder.penjelasanproduk.setText(model.getPenjelasan_produk()
);
        holder.merk.setText(model.getMerk());
        Glide.with(context).load(model.getBrowsure_url()).into(holder.gambarproduk);
        holder.gambarproduk.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                onItemClick.onItemClick(model);
            }
        });
    }

    @Override
    public int getItemCount() {
        return listData.size();
    }

    public class HolderData extends RecyclerView.ViewHolder{
        TextView id, penjelasanproduk, merk;
        ImageView gambarproduk;

        public HolderData(@NonNull View itemView) {
            super(itemView);
            id = itemView.findViewById(R.id.id);
            penjelasanproduk =
itemView.findViewById(R.id.penjelasanproduk);
            merk = itemView.findViewById(R.id.merk);
            gambarproduk = itemView.findViewById(R.id.gambarproduk);
        }
    }
}

```

SourceCode 1. 94 Data Fungsida

i) Data Herbisida

```

package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

```

```

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.Model.ModelHerbisida;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

public class DataHerbisida extends
RecyclerView.Adapter<DataHerbisida.HolderData> {
    private Context context;
    private ArrayList<ModelHerbisida.data_herbisida> listData;
    private OnItemClickListener onItemClicked;

    public interface OnItemClickListener{
        void onItemClick(ModelHerbisida.data_herbisida
modelHerbisida);
    }

    public DataHerbisida(Context context,
ArrayList<ModelHerbisida.data_herbisida> listData,
OnItemClickListener onItemClicked) {
        this.context = context;
        this.listData = listData;
        this.onItemClicked = onItemClicked;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_herbi
sida,parent,false);
        return new HolderData(layout);
    }

    @Override
    public void onBindViewHolder(@NonNull HolderData holder, int
position) {
        ModelHerbisida.data_herbisida model =
listData.get(position);

        holder.id.setText(String.valueOf(model.getId_herbisida()));

```

```

        holder.penjelasanproduk.setText(model.getPenjelasan_produk()
);
        holder.merk.setText(model.getMerk());
        Glide.with(context).load(model.getBrowsure_url()).into(holder.gambarproduk);
        holder.gambarproduk.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                onItemClick.onItemClick(model);
            }
        });
    }

    @Override
    public int getItemCount() {
        return listData.size();
    }

    public class HolderData extends RecyclerView.ViewHolder{
        TextView id, penjelasanproduk, merk;
        ImageView gambarproduk;

        public HolderData(@NonNull View itemView) {
            super(itemView);
            id = itemView.findViewById(R.id.id);
            penjelasanproduk =
itemView.findViewById(R.id.penjelasanproduk);
            merk = itemView.findViewById(R.id.merk);
            gambarproduk = itemView.findViewById(R.id.gambarproduk);
        }
    }
}

```

SourceCode 1. 95 Data Herbisida

j) Data Insektisida

```

package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

```



```

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.Model.ModelInsektisida;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

public class DataInsektisida extends
RecyclerView.Adapter<DataInsektisida.HolderData>{
    private Context context;
    private ArrayList<ModelInsektisida.data_insektisida> listData;
    private OnItemClickListener onItemClicked;

    public interface OnItemClickListener{
        void onItemClick(ModelInsektisida.data_insektisida
modelHerbisida);
    }

    public DataInsektisida(Context context,
ArrayList<ModelInsektisida.data_insektisida> listData,
OnItemClickListener onItemClicked) {
        this.context = context;
        this.listData = listData;
        this.onItemClicked = onItemClicked;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_insek
tisida,parent,false);
        return new HolderData(layout);
    }

    @Override
    public void onBindViewHolder(@NonNull HolderData holder, int
position) {
        ModelInsektisida.data_insektisida model =
listData.get(position);

        holder.idinsektisida.setText(String.valueOf(model.getId_inse
ktisida()));
    }
}

```

```

        holder.penjelasanproduk.setText(model.getPenjelasan_produk()
);
        holder.merk.setText(model.getMerk());
        Glide.with(context).load(model.getBrowsure_url()).into(holder.gambarproduk);
        holder.gambarproduk.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                onItemClick.onItemClick(model);
            }
        });
    }

    @Override
    public int getItemCount() {
        return listData.size();
    }

    public class HolderData extends RecyclerView.ViewHolder{
        TextView idinsektisida,penjelasanproduk,merk;
        ImageView gambarproduk;
        public HolderData(@NonNull View itemView) {
            super(itemView);
            idinsektisida =
itemView.findViewById(R.id.idinsektisida);
            penjelasanproduk =
itemView.findViewById(R.id.penjelasanproduk);
            merk = itemView.findViewById(R.id.merk);
            gambarproduk = itemView.findViewById(R.id.gambarproduk);
        }
    }
}

```

SourceCode 1. 96 Data Insektisida

k) Data Kelola Alamat

```

package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;

```

```

import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import
com.example.cakraagroindonesia.Activity.TentangKami.UbahAlamat;
import com.example.cakraagroindonesia.HapusData;
import com.example.cakraagroindonesia.Interface.InterfaceAlamat;
import com.example.cakraagroindonesia.Model.ModelAlamat;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class DataKelolaAlamat extends
RecyclerView.Adapter<DataKelolaAlamat.HolderData>{
    private static Context context;
    private ArrayList<ModelAlamat.alamat> listData;
    static int Id;

    private static final int VIEW_TYPE_ITEM = 0;
    private static final int VIEW_TYPE_ITEM_LAST = 1;

    public DataKelolaAlamat(Context context,
ArrayList<ModelAlamat.alamat> listData) {
        this.context = context;
        this.listData = listData;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {

        if (viewType == VIEW_TYPE_ITEM) {
            View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_kelol
aalamat,parent,false);
            return new HolderData(layout);
        } else if (viewType == VIEW_TYPE_ITEM_LAST) {

```

```

        View footer =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_foote
r,parent,false);
        return new HolderData(footer);
    }

    throw new IllegalArgumentException("Invalid view type");
}

@Override
public void onBindViewHolder(@NonNull HolderData holder, int
position) {
    if (getItemViewType(position) == VIEW_TYPE_ITEM) {
        // Bind item data
        ModelAlamat.alamat model = listData.get(position);

        holder.idalamat.setText(String.valueOf(model.getId_alama
t()));
        holder.namakantor.setText(model.getNama_kantor());
        holder.alamat.setText(model.getAlamat());
        holder.telepon.setText(model.getTelepon());

        holder.btnEdit.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                InterfaceAlamat interfaceAlamat =
RetroServer.KonesiAPI().create(InterfaceAlamat.class);
                Call<ModelAlamat> ambil =
interfaceAlamat.getEditAlamat(Id);
                ambil.enqueue(new Callback<ModelAlamat>() {
                    @Override
                    public void onResponse(Call<ModelAlamat>
call, Response<ModelAlamat> response) {

                        int varId = model.getId_alamat();
                        String varNamaKantor =
model.getNama_kantor();

                        String varAlamat = model.getAlamat();
                        String varNotelp = model.getTelepon();
                        String varFoto = model.getFoto_url();

                        // Toast.makeText(context,
                        ""+varId+varNamaKantor+varAlamat+varNotelp,
                        Toast.LENGTH_SHORT).show();
                    }
                });
            }
        });
    }
}

```

```

        Intent kirim = new Intent(context,
UbahAlamat.class);

        kirim.putExtra("xId", varId);
        kirim.putExtra("xNamaKantor",
varNamaKantor);

        kirim.putExtra("xAlamat", varAlamat);
        kirim.putExtra("xNotelp", varNotelp);
        kirim.putExtra("xFoto", varFoto);
        context.startActivity(kirim);
    }

    @Override
    public void onFailure(Call<ModelAlamat>
call, Throwable t) {

    }

    });
}

});

        holder.btnHapus.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String Data = "Alamat";

        Intent hapus = new Intent(context,
HapusData.class);

        hapus.putExtra("idHapus", model.getId_alamat());
        hapus.putExtra("Data", Data);
        context.startActivity(hapus);
    }
});
} else if (getItemViewType(position) == VIEW_TYPE_ITEM_LAST)
{
    // Bind data for the last item
}
}

@Override
public int getItemCount() {
    return listData.size() + 1;
}

public static class HolderData extends RecyclerView.ViewHolder{
    TextView namakantor, alamat, telepon, idalamat;
    ImageView btnHapus, btnEdit;
}

```

```

    public HolderData(@NonNull View itemView) {
        super(itemView);
        namakantor = itemView.findViewById(R.id.namakantor);
        alamat = itemView.findViewById(R.id.alamat);
        btnHapus = itemView.findViewById(R.id.btnhapus);
        btnEdit = itemView.findViewById(R.id.btncedit);
        telepon = itemView.findViewById(R.id.telepon);
        idalamat = itemView.findViewById(R.id.idalamat);
    }
}

@Override
public int getItemViewType(int position) {
    if (position == listData.size()) {
        return VIEW_TYPE_ITEM_LAST; // Tampilan terakhir
    }
    return VIEW_TYPE_ITEM; // Tampilan item biasa
}
}

```

SourceCode 1. 97 Data kelola Alamat

l) Data Kelola Berita

```

package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.content.Intent;
import android.text.Html;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Activity.Beranda.UbahBerita;
import com.example.cakraagroindonesia.HapusData;
import com.example.cakraagroindonesia.Interface.InterfaceBerita;
import com.example.cakraagroindonesia.Model.ModelBerita;

```

```

import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class DataKelolaBerita extends
RecyclerView.Adapter<DataKelolaBerita.HolderData>{
    private Context context;
    private ArrayList<ModelBerita.data_berita> listData;
    int Id;

    private float currentRotation = 0;

    private static final int VIEW_TYPE_ITEM = 0;
    private static final int VIEW_TYPE_ITEM_LAST = 1;

    public DataKelolaBerita(Context context,
ArrayList<ModelBerita.data_berita> listData) {
        this.context = context;
        this.listData = listData;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        if (viewType == VIEW_TYPE_ITEM) {
            View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_kelol
aberita,parent,false);
            return new HolderData(layout);
        } else if (viewType == VIEW_TYPE_ITEM_LAST) {
            View footer =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_foote
r,parent,false);
            return new HolderData(footer);
        }

        throw new IllegalArgumentException("Invalid view type");
    }

    @Override
    public void onBindViewHolder(@NonNull HolderData holder, int
position) {

```



```

        holder.btnHapus.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String Data = "Berita";
                Intent hapus = new Intent(context,
HapusData.class);
                hapus.putExtra("idHapus", model.getId_berita());
                hapus.putExtra("Data",Data);
                context.startActivity(hapus);
            }
        });

        holder.btn.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View view) {
                int invisible = holder.detail.getVisibility();
                if (invisible == View.VISIBLE) {
                    // Jika LinearLayout terlihat, ubah menjadi
GONE
                    holder.detail.setVisibility(View.GONE);
                    holder.btndetail.setRotation(0);
                } else {
                    // Jika LinearLayout tidak terlihat, ubah
menjadi VISIBLE
                    holder.detail.setVisibility(View.VISIBLE);
                    holder.btndetail.setRotation(+180);
                }
            }
        });
    } else if (getItemViewType(position) == VIEW_TYPE_ITEM_LAST)
{
        // Bind data for the last item
    }
}

@Override
public int getItemCount() {
    return listData.size() + 1;
}

public class HolderData extends RecyclerView.ViewHolder{
    TextView tvIdBerita, tvJudulBerita, tvIsiBerita;
    LinearLayout detail;
    RelativeLayout btn;
}

```

```

        ImageView btnHapus, btnEdit, btndetail;

        public HolderData(@NonNull View itemView) {
            super(itemView);

            tvIdBerita = itemView.findViewById(R.id.idberita);
            tvJudulBerita = itemView.findViewById(R.id.judulberita);
            tvIsiBerita = itemView.findViewById(R.id.isiberita);
            detail = itemView.findViewById(R.id.detail);
            btnHapus = itemView.findViewById(R.id.btnhapus);
            btnEdit = itemView.findViewById(R.id.btncedit);
            btndetail = itemView.findViewById(R.id.btndetail);
            btn = itemView.findViewById(R.id.btn);
        }
    }
    @Override
    public int getItemViewType(int position) {
        if (position == listData.size()) {
            return VIEW_TYPE_ITEM_LAST; // Tampilan terakhir
        }
        return VIEW_TYPE_ITEM; // Tampilan item biasa
    }
}

```

SourceCode 1. 98 Data Kelola Berita

m) Data Kelola FAQ

```

package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.HapusData;
import com.example.cakraagroindonesia.Activity.Pertanyaan.UbahFAQ;

```

```

import com.example.cakraagroindonesia.Interface.InterfaceFaq;
import com.example.cakraagroindonesia.Model.ModelFaq;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class DataKelolaFaq extends
RecyclerView.Adapter<DataKelolaFaq.HolderData>{
    private Context context;
    private ArrayList<ModelFaq.data_faq> listData;
    int Id;

    private float currentRotation = 0;

    private static final int VIEW_TYPE_ITEM = 0;
    private static final int VIEW_TYPE_ITEM_LAST = 1;

    public DataKelolaFaq(Context context,
ArrayList<ModelFaq.data_faq> listData) {
        this.context = context;
        this.listData = listData;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        if (viewType == VIEW_TYPE_ITEM) {
            View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_kelola
afaq,parent,false);
            return new HolderData(layout);
        } else if (viewType == VIEW_TYPE_ITEM_LAST) {
            View footer =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_foote
r,parent,false);
            return new HolderData(footer);
        }

        throw new IllegalArgumentException("Invalid view type");
    }
}

```

```

    @Override
    public void onBindViewHolder(@NonNull HolderData holder, int
position) {
        if (getItemViewType(position) == VIEW_TYPE_ITEM) {
            // Bind item data
            ModelFaq.data_faq model = listData.get(position);

            holder.pertanyaan.setText(model.getPertanyaan());
            holder.jawaban.setText(model.getJawaban());
            holder.btn.setOnClickListener(new View.OnClickListener()
{
                @Override
                public void onClick(View view) {
                    int invisible = holder.detail.getVisibility();
                    if (invisible == View.VISIBLE) {
                        // Jika LinearLayout terlihat, ubah menjadi
GONE

                        holder.detail.setVisibility(View.GONE);
                        holder.btndetail.setRotation(0);
                    } else {
                        // Jika LinearLayout tidak terlihat, ubah
menjadi VISIBLE

                        holder.detail.setVisibility(View.VISIBLE);
                        holder.btndetail.setRotation(+180);
                    }
                }
            });
            holder.btnEdit.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    InterfaceFaq interfaceFaq =
RetroServer.KonesiAPI().create(InterfaceFaq.class);
                    Call<ModelFaq> modelFaqCall =
interfaceFaq.getDataFaq(Id);
                    modelFaqCall.enqueue(new Callback<ModelFaq>() {
                        @Override
                        public void onResponse(Call<ModelFaq> call,
Response<ModelFaq> response) {
                            int varId = model.getId_faq();
                            String varPertanyaan =
model.getPertanyaan();

                            String varJawaban = model.getJawaban();

                            Intent kirim = new Intent(context,
UbahFAQ.class);

                            kirim.putExtra("xId", varId);

```

```

                kirim.putExtra("xPertanyaan",
varPertanyaan);
                kirim.putExtra("xPertanyaan",
varJawaban);
                context.startActivity(kirim);
            }

            @Override
            public void onFailure(Call<ModelFaq> call,
Throwable t) {
                }
            });
        }
    });

    holder.btnHapus.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String Data = "FAQ";

            Intent hapus = new Intent(context,
HapusData.class);

            hapus.putExtra("idHapus", model.getId_faq());
            hapus.putExtra("Data", Data);
            context.startActivity(hapus);
        }
    });
} else if (getItemViewType(position) == VIEW_TYPE_ITEM_LAST)
{
    // Bind data for the last item
}

}

@Override
public int getItemCount() {
    return listData.size() + 1;
}

public class HolderData extends RecyclerView.ViewHolder{
    TextView pertanyaan, jawaban;
    LinearLayout detail;
    RelativeLayout btn;
    ImageView btnHapus, btnEdit, btndetail;
}

```

```

public HolderData(@NonNull View itemView) {
    super(itemView);
    pertanyaan = itemView.findViewById(R.id.pertanyaan);
    jawaban = itemView.findViewById(R.id.jawaban);
    detail = itemView.findViewById(R.id.detail);
    btnEdit = itemView.findViewById(R.id.btnedit);
    btnHapus = itemView.findViewById(R.id.btnhapus);
    btndetail = itemView.findViewById(R.id.btndetail);
    btn = itemView.findViewById(R.id.btn);
}
}
@Override
public int getItemViewType(int position) {
    if (position == listData.size()) {
        return VIEW_TYPE_ITEM_LAST; // Tampilan terakhir
    }
    return VIEW_TYPE_ITEM; // Tampilan item biasa
}
}

```

SourceCode 1. 99 Data Kelola FAQ

n) Data Kelola Fungsida

```

package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.content.Intent;
import android.text.Html;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Activity.Produk.UbahProduk;
import com.example.cakraagroindonesia.HapusData;
import com.example.cakraagroindonesia.Interface.InterfaceFungsida;

```

```

import com.example.cakraagroindonesia.Model.ModelFungsida;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class DataKelolaFungsida extends
RecyclerView.Adapter<DataKelolaFungsida.HolderData>{
    private Context context;
    private ArrayList<ModelFungsida.data_fungsida> listData;
    int Id;
    private float currentRotation = 0;

    private static final int VIEW_TYPE_ITEM = 0;
    private static final int VIEW_TYPE_ITEM_LAST = 1;

    public DataKelolaFungsida(Context context,
ArrayList<ModelFungsida.data_fungsida> listData) {
        this.context = context;
        this.listData = listData;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        if (viewType == VIEW_TYPE_ITEM) {
            View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_kelol
aproduk,parent,false);
            return new HolderData(layout);
        } else if (viewType == VIEW_TYPE_ITEM_LAST) {
            View footer =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_foote
r,parent,false);
            return new HolderData(footer);
        }

        throw new IllegalArgumentException("Invalid view type");
    }

    @Override

```

```

    public void onBindViewHolder(@NonNull HolderData holder, int
position) {
        if (getItemViewType(position) == VIEW_TYPE_ITEM) {
            // Bind item data
            ModelFungsida.data_fungsida model =
listData.get(position);

            holder.id.setText(String.valueOf(model.getId_fungsida()
));
            holder.penjelasanproduk.setText(Html.fromHtml(model.getP
enjelasan_produk()));
            holder.merk.setText(model.getMerk());
            Glide.with(context).load(model.getBrowsure_url()).into(h
older.gambarProduk);
            holder.btn.setOnClickListener(new View.OnClickListener()
{
                @Override
                public void onClick(View view) {
                    int invisible = holder.detail.getVisibility();
                    if (invisible == View.VISIBLE) {
                        // Jika LinearLayout terlihat, ubah menjadi
GONE

                        holder.detail.setVisibility(View.GONE);
                        holder.btndetail.setRotation(0);
                    } else {
                        // Jika LinearLayout tidak terlihat, ubah
menjadi VISIBLE

                        holder.detail.setVisibility(View.VISIBLE);
                        holder.btndetail.setRotation(+180);
                    }
                }
            });
            holder.hapus.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    String Data = "Fungsida";
                    Intent hapus = new Intent(context,
HapusData.class);
                    hapus.putExtra("idHapus",
model.getId_fungsida());
                    hapus.putExtra("Data", Data);
                    context.startActivity(hapus);
                }
            });
            holder.edit.setOnClickListener(new
View.OnClickListener() {

```



```

        @Override
        public void onClick(View view) {
            InterfaceFungsida interfaceFungsida =
RetroServer.KonesiAPI().create(InterfaceFungsida.class);
            Call<ModelFungsida> ambil =
interfaceFungsida.getEdit(Id);
            ambil.enqueue(new Callback<ModelFungsida>() {
                @Override
                public void onResponse(Call<ModelFungsida>
call, Response<ModelFungsida> response) {
                    int varId = model.getId_fungsida();
                    String varNamaProduk = model.getMerk();
                    String varDeskripsi =
model.getPenjelasan_produk();
                    String varBrowsure =
model.getBrowsure_url();
                    String varJenisProduk = "FUNGISIDA";

                    Intent kirim = new Intent(context,
UbahProduk.class);

                    kirim.putExtra("xId", varId);
                    kirim.putExtra("xNamaProduk",
varNamaProduk);
                    kirim.putExtra("xDeskripsi",
varDeskripsi);
                    kirim.putExtra("xBrowsure",
varBrowsure);
                    kirim.putExtra("xJenisProduk",
varJenisProduk);
                    context.startActivity(kirim);
                }

                @Override
                public void onFailure(Call<ModelFungsida>
call, Throwable t) {

                }
            });
        }
    });
} else if (getItemViewType(position) == VIEW_TYPE_ITEM_LAST)
{
    // Bind data for the last item
}
}

@Override

```

```

public int getItemCount() {
    return listData.size() + 1;
}

public class HolderData extends RecyclerView.ViewHolder{
    TextView id, kandungan, penjelasanproduk, merk;
    LinearLayout detail;
    RelativeLayout btn;
    ImageView hapus,edit,gambarProduk, btndetail;
    public HolderData(@NonNull View itemView) {
        super(itemView);
        id = itemView.findViewById(R.id.id);
        merk = itemView.findViewById(R.id.merk);
        detail = itemView.findViewById(R.id.detail);
        kandungan = itemView.findViewById(R.id.kandungan);
        penjelasanproduk =
itemView.findViewById(R.id.penjelasanproduk);
        gambarProduk = itemView.findViewById(R.id.gambarproduk);
        hapus = itemView.findViewById(R.id.hapus);
        edit = itemView.findViewById(R.id.edit);
        btndetail = itemView.findViewById(R.id.btndetail);
        btn = itemView.findViewById(R.id.btn);
    }
}
@Override
public int getItemViewType(int position) {
    if (position == listData.size()) {
        return VIEW_TYPE_ITEM_LAST; // Tampilan terakhir
    }
    return VIEW_TYPE_ITEM; // Tampilan item biasa
}
}
}

```

SourceCode 1. 100 Data kelola Fungisida

o) Data Kelola Herbisida

```

package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.content.Intent;
import android.text.Html;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;

```

```

import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.HapusData;
import com.example.cakraagroindonesia.Activity.Produk.UbahProduk;
import com.example.cakraagroindonesia.Interface.InterfaceHerbisida;
import com.example.cakraagroindonesia.Model.ModelHerbisida;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class DataKelolaHerbisida extends
RecyclerView.Adapter<DataKelolaHerbisida.HolderData>{
    private Context context;
    private ArrayList<ModelHerbisida.data_herbisida> listData;
    private int Id;
    private float currentRotation = 0;

    private static final int VIEW_TYPE_ITEM = 0;
    private static final int VIEW_TYPE_ITEM_LAST = 1;

    public DataKelolaHerbisida(Context context,
ArrayList<ModelHerbisida.data_herbisida> listData) {
        this.context = context;
        this.listData = listData;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        if (viewType == VIEW_TYPE_ITEM) {
            View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_kelol
aproduk,parent,false);
            return new HolderData(layout);

```

```

        } else if (viewType == VIEW_TYPE_ITEM_LAST) {
            View footer =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_foote
r,parent,false);
            return new HolderData(footer);
        }

        throw new IllegalArgumentException("Invalid view type");
    }

    @Override
    public void onBindViewHolder(@NonNull HolderData holder, int
position) {

        if (getItemViewType(position) == VIEW_TYPE_ITEM) {
            // Bind item data
            ModelHerbisida.data_herbisida model =
listData.get(position);

            holder.id.setText(String.valueOf(model.getId_herbisida()
));
            holder.penjelasanproduk.setText(Html.fromHtml(model.getP
enjelasan_produk()));
            holder.merk.setText(model.getMerk());
            Glide.with(context).load(model.getBrowsure_url()).into(h
older.gambarProduk);
            holder.btn.setOnClickListener(new View.OnClickListener()
{
                @Override
                public void onClick(View view) {
                    int invisible = holder.detail.getVisibility();
                    if (invisible == View.VISIBLE) {
                        // Jika LinearLayout terlihat, ubah menjadi
GONE

                        holder.detail.setVisibility(View.GONE);
                        holder.btndetail.setRotation(0);
                    } else {
                        // Jika LinearLayout tidak terlihat, ubah
menjadi VISIBLE

                        holder.detail.setVisibility(View.VISIBLE);
                        holder.btndetail.setRotation(+180);
                    }
                }
            });
            holder.hapus.setOnClickListener(new
View.OnClickListener() {

```

```

        @Override
        public void onClick(View view) {
            String Data = "Herbisida";

            Intent hapus = new Intent(context,
HapusData.class);
            hapus.putExtra("Data", Data);
            hapus.putExtra("idHapus",
model.getId_herbisida());
            context.startActivity(hapus);
        }
    });
    holder.edit.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {
            InterfaceHerbisida ambidata =
RetroServer.KonesiAPI().create(InterfaceHerbisida.class);
            Call<ModelHerbisida> ambil =
ambidata.getEditDataHerbisida(Id);
            ambil.enqueue(new Callback<ModelHerbisida>() {
                @Override
                public void onResponse(Call<ModelHerbisida>
call, Response<ModelHerbisida> response) {
                    int varId = model.getId_herbisida();
                    String varNamaProduk = model.getMerk();
                    String varDeskripsi =
model.getPenjelasan_produk();
                    String varBrowsure =
model.getBrowsure_url();
                    String varJenisProduk = "HERBISIDA";

                    Intent kirim = new Intent(context,
UbahProduk.class);

                    kirim.putExtra("xId", varId);
                    kirim.putExtra("xNamaProduk",
varNamaProduk);
                    kirim.putExtra("xDeskripsi",
varDeskripsi);
                    kirim.putExtra("xBrowsure",
varBrowsure);
                    kirim.putExtra("xJenisProduk",
varJenisProduk);
                    context.startActivity(kirim);
                }
            })

        @Override

```

```

        public void onFailure(Call<ModelHerbisida>
call, Throwable t) {
            }
        });
    }
});
} else if (getItemViewType(position) == VIEW_TYPE_ITEM_LAST)
{
    // Bind data for the last item
}

}

@Override
public int getItemCount() {
    return listData.size() + 1;
}

public class HolderData extends RecyclerView.ViewHolder{
    TextView id, kandungan, penjelasanproduk, merk;
    LinearLayout detail;
    RelativeLayout btn;
    ImageView hapus,edit,gambarProduk, btndetail;
    public HolderData(@NonNull View itemView) {
        super(itemView);
        id = itemView.findViewById(R.id.id);
        merk = itemView.findViewById(R.id.merk);
        detail = itemView.findViewById(R.id.detail);
        kandungan = itemView.findViewById(R.id.kandungan);
        penjelasanproduk =
itemView.findViewById(R.id.penjelasanproduk);
        gambarProduk = itemView.findViewById(R.id.gambarproduk);
        hapus = itemView.findViewById(R.id.hapus);
        edit = itemView.findViewById(R.id.edit);
        btndetail = itemView.findViewById(R.id.btndetail);
        btn = itemView.findViewById(R.id.btn);
    }
}

@Override
public int getItemViewType(int position) {
    if (position == listData.size()) {
        return VIEW_TYPE_ITEM_LAST; // Tampilan terakhir
    }
    return VIEW_TYPE_ITEM; // Tampilan item biasa
}

```

```
}  
}
```

SourceCode 1. 101 Data Kelola Herbisida

p) Data Kelola Insektisida

```
package com.example.cakraagroindonesia.Adapter;  
  
import android.content.Context;  
import android.content.Intent;  
import android.text.Html;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
import android.widget.LinearLayout;  
import android.widget.RelativeLayout;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.bumptech.glide.Glide;  
import com.example.cakraagroindonesia.API.RetroServer;  
import com.example.cakraagroindonesia.Activity.Produk.UbahProduk;  
import com.example.cakraagroindonesia.HapusData;  
import  
com.example.cakraagroindonesia.Interface.InterfaceInsektisida;  
import com.example.cakraagroindonesia.Model.ModelInsektisida;  
import com.example.cakraagroindonesia.R;  
  
import java.util.ArrayList;  
  
import retrofit2.Call;  
import retrofit2.Callback;  
import retrofit2.Response;  
  
public class DataKelolaInsektisida extends  
RecyclerView.Adapter<DataKelolaInsektisida.HolderData>{  
    private Context context;  
    private ArrayList<ModelInsektisida.data_insektisida> listData;  
    private int Id;  
    private float currentRotation = 0;  
  
    private static final int VIEW_TYPE_ITEM = 0;
```

```

private static final int VIEW_TYPE_ITEM_LAST = 1;

public DataKelolaInsektisida(Context context,
ArrayList<ModelInsektisida.data_insektisida> listData) {
    this.context = context;
    this.listData = listData;
}

@NonNull
@Override
public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {

    if (viewType == VIEW_TYPE_ITEM) {
        View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_kelola
aproduk,parent,false);
        return new HolderData(layout);
    } else if (viewType == VIEW_TYPE_ITEM_LAST) {
        View footer =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_foote
r,parent,false);
        return new HolderData(footer);
    }

    throw new IllegalArgumentException("Invalid view type");

}

@Override
public void onBindViewHolder(@NonNull HolderData holder, int
position) {
    if (getItemViewType(position) == VIEW_TYPE_ITEM) {
        ModelInsektisida.data_insektisida model =
listData.get(position);

        holder.id.setText(String.valueOf(model.getId_insektisida
()));
        holder.penjelasanproduk.setText(Html.fromHtml(model.getP
enjelasan_produk()));
        holder.merk.setText(model.getMerk());
        Glide.with(context).load(model.getBrowsure_url()).into(h
older.gambarProduk);
        holder.btn.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View view) {

```



```

        int invisible = holder.detail.getVisibility();
        if (invisible == View.VISIBLE) {
            // Jika LinearLayout terlihat, ubah menjadi
GONE
            holder.detail.setVisibility(View.GONE);
            holder.btndetail.setRotation(0);
        } else {
            // Jika LinearLayout tidak terlihat, ubah
menjadi VISIBLE
            holder.detail.setVisibility(View.VISIBLE);
            holder.btndetail.setRotation(+180);
        }
    }
});
holder.hapus.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String Data = "Insektisida";

        Intent hapus = new Intent(context,
HapusData.class);
        hapus.putExtra("idHapus",
model.getId_insektisida());
        hapus.putExtra("Data", Data);
        context.startActivity(hapus);
    }
});
holder.edit.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        InterfaceInsektisida ambildata =
RetroServer.KonesiAPI().create(InterfaceInsektisida.class);
        Call<ModelInsektisida> ambil =
ambildata.getEditInsektisida(Id);
        ambil.enqueue(new Callback<ModelInsektisida>() {
            @Override
            public void
onResponse(Call<ModelInsektisida> call, Response<ModelInsektisida>
response) {
                int varId = model.getId_insektisida();
                String varNamaProduk = model.getMerk();
                String varDeskripsi =
model.getPenjelasan_produk();

```

```

        String varBrowsure =
model.getBrowsure_url();
        String varJenisProduk = "INSEKTISIDA";

        Intent kirim = new Intent(context,
UbahProduk.class);
        kirim.putExtra("xId", varId);
        kirim.putExtra("xNamaProduk",
varNamaProduk);
        kirim.putExtra("xDeskripsi",
varDeskripsi);
        kirim.putExtra("xBrowsure",
varBrowsure);
        kirim.putExtra("xJenisProduk",
varJenisProduk);
        context.startActivity(kirim);
    }

    @Override
    public void onFailure(Call<ModelInsektisida>
call, Throwable t) {
        }
    });
}
});
} else if (getItemViewType(position) == VIEW_TYPE_ITEM_LAST)
{
    // Bind data for the last item
}
}

@Override
public int getItemCount() {
    return listData.size() + 1;
}

public class HolderData extends RecyclerView.ViewHolder{
    TextView id, kandungan, penjelasanproduk, merk;
    LinearLayout detail;
    RelativeLayout btn;
    ImageView hapus,edit,gambarProduk, btndetail;
    public HolderData(@NonNull View itemView) {
        super(itemView);
        id = itemView.findViewById(R.id.id);
        merk = itemView.findViewById(R.id.merk);
        detail = itemView.findViewById(R.id.detail);
    }
}

```

```

        kandungan = itemView.findViewById(R.id.kandungan);
        penjelasanproduk =
itemView.findViewById(R.id.penjelasanproduk);
        gambarProduk = itemView.findViewById(R.id.gambarproduk);
        hapus = itemView.findViewById(R.id.hapus);
        edit = itemView.findViewById(R.id.edit);
        btndetail = itemView.findViewById(R.id.btndetail);
        btn = itemView.findViewById(R.id.btn);
    }
}
@Override
public int getItemViewType(int position) {
    if (position == listData.size()) {
        return VIEW_TYPE_ITEM_LAST; // Tampilan terakhir
    }
    return VIEW_TYPE_ITEM; // Tampilan item biasa
}
}
}

```

SourceCode 1. 102 Data Kelola Insektisida

q) Data Kelola Paket Produk

```

package com.example.cakraagroindonesia.Adapter;

import android.app.DownloadManager;
import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.os.Environment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import
com.example.cakraagroindonesia.Activity.Inovasi.UbahPaketProduk;
import com.example.cakraagroindonesia.HapusData;
import
com.example.cakraagroindonesia.Interface.InterfacePaketProduk;
import com.example.cakraagroindonesia.Model.ModelPaketProduk;

```

```

import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class DataKelolaPaketProduk extends
RecyclerView.Adapter<DataKelolaPaketProduk.HolderData> {
    private Context context;
    private ArrayList<ModelPaketProduk.paket_produk> listData;
    int Id;

    private static final int VIEW_TYPE_ITEM = 0;
    private static final int VIEW_TYPE_ITEM_LAST = 1;

    public DataKelolaPaketProduk(Context context,
ArrayList<ModelPaketProduk.paket_produk> listData) {
        this.context = context;
        this.listData = listData;
    }

    public class HolderData extends RecyclerView.ViewHolder{
        TextView tvIdPaket, tvPaketProduk, tvTanaman, tvIterasi,
tvHasil, ibPpt;
        ImageView btnHapus, btnEdit;
        public HolderData(@NonNull View itemView) {
            super(itemView);
            tvIdPaket = itemView.findViewById(R.id.idpaket);
            tvPaketProduk = itemView.findViewById(R.id.paketproduk);
            tvTanaman = itemView.findViewById(R.id.tanaman);
            tvIterasi = itemView.findViewById(R.id.iterasi);
            tvHasil = itemView.findViewById(R.id.hasil);
            ibPpt = itemView.findViewById(R.id.btnlihatppt);
            btnHapus = itemView.findViewById(R.id.hapus);
            btnEdit = itemView.findViewById(R.id.edit);
        }
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        if (viewType == VIEW_TYPE_ITEM) {

```

```

        View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_kelol
apaketproduk,parent,false);
        return new HolderData(layout);
    } else if (viewType == VIEW_TYPE_ITEM_LAST) {
        View footer =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_foote
r,parent,false);
        return new HolderData(footer);
    }

    throw new IllegalArgumentException("Invalid view type");
}

@Override
public void onBindViewHolder(@NonNull HolderData holder, int
position) {
    if (getItemViewType(position) == VIEW_TYPE_ITEM) {
        // Bind item data
        ModelPaketProduk.paket_produk model =
listData.get(position);
        String posisi = String.valueOf(position+1);
        holder.tvIdPaket.setText(posisi);
        holder.tvPaketProduk.setText(model.getPaket_produk());
        holder.tvTanaman.setText(model.getTanaman());
        holder.tvIterasi.setText(model.getIterasi());
        holder.tvHasil.setText(model.getHasil());
        holder.ibPpt.setOnClickListner(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String pptUrl = model.getFoto_url(); // Ambil
URL PPT dari data paket
                DownloadManager.Request request = new
DownloadManager.Request(Uri.parse(pptUrl));
                request.setTitle(model.getPpt());
                request.setDescription("Mengunduh file PPT...");
                request.setNotificationVisibility(DownloadManag
er.Request.VISIBILITY_VISIBLE_NOTIFY_COMPLETED);
                request.setDestinationInExternalPublicDir(Enviro
nment.DIRECTORY_DOWNLOADS, model.getPpt());

                DownloadManager downloadManager =
(DownloadManager)
context.getSystemService(Context.DOWNLOAD_SERVICE);
                downloadManager.enqueue(request);
            }
        });
    }
}

```

```

    });
    holder.btnHapus.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String Data = "PaketProduk";

        Intent hapus = new Intent(context,
HapusData.class);
        hapus.putExtra("idHapus",model.getId_paket());
        hapus.putExtra("Data", Data);
        context.startActivity(hapus);
    }
});
    holder.btnEdit.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        InterfacePaketProduk interfacePaketProduk =
RetroServer.KonesiAPI().create(InterfacePaketProduk.class);
        Call<ModelPaketProduk> edit =
interfacePaketProduk.getEdit(Id);
        edit.enqueue(new Callback<ModelPaketProduk>() {
            @Override
            public void
onResponse(Call<ModelPaketProduk> call, Response<ModelPaketProduk>
response) {
                int varId = model.getId_paket();
                String varNamaPaket =
model.getPaket_produk();
                String varTanaman = model.getTanaman();
                String varDeskripsi =
model.getIterasi();
                String varHasil = model.getHasil();

                Intent kirim = new Intent(context,
UbahPaketProduk.class);
                kirim.putExtra("xId", varId);
                kirim.putExtra("xNamaPaket",
varNamaPaket);
                kirim.putExtra("xDeskripsi",
varDeskripsi);
                kirim.putExtra("xTanaman", varTanaman);
                kirim.putExtra("xHasil", varHasil);
                context.startActivity(kirim);
            }
        });
    }
});
    @Override

```

```

        public void onFailure(Call<ModelPaketProduk>
call, Throwable t) {
            }
        });
    }
});
} else if (getItemViewType(position) == VIEW_TYPE_ITEM_LAST)
{
    // Bind data for the last item
}
}

@Override
public int getItemCount() {
    return listData.size() + 1;
}

@Override
public int getItemViewType(int position) {
    if (position == listData.size()) {
        return VIEW_TYPE_ITEM_LAST; // Tampilan terakhir
    }
    return VIEW_TYPE_ITEM; // Tampilan item biasa
}
}
}

```

SourceCode 1. 103 Data Kelola Paket Produk

r) Data Kelola Pgr

```

package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.content.Intent;
import android.text.Html;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

```

```

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Activity.Produk.UbahProduk;
import com.example.cakraagroindonesia.HapusData;
import com.example.cakraagroindonesia.Interface.InterfacePgr;
import com.example.cakraagroindonesia.Model.ModelPgr;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class DataKelolaPgr extends
RecyclerView.Adapter<DataKelolaPgr.HolderData>{
    private Context context;
    private ArrayList<ModelPgr.data_pgr> listData;
    int Id;
    private float currentRotation = 0;

    private static final int VIEW_TYPE_ITEM = 0;
    private static final int VIEW_TYPE_ITEM_LAST = 1;

    public DataKelolaPgr(Context context,
ArrayList<ModelPgr.data_pgr> listData) {
        this.context = context;
        this.listData = listData;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        if (viewType == VIEW_TYPE_ITEM) {
            View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_kelol
aproduk,parent,false);
            return new HolderData(layout);
        } else if (viewType == VIEW_TYPE_ITEM_LAST) {
            View footer =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_foote
r,parent,false);
            return new HolderData(footer);
        }
    }
}

```



```

        throw new IllegalArgumentException("Invalid view type");
    }

    @Override
    public void onBindViewHolder(@NonNull HolderData holder, int
position) {
        if (getItemViewType(position) == VIEW_TYPE_ITEM) {
            // Bind item data
            ModelPgr.data_pgr model = listData.get(position);

            holder.id.setText(String.valueOf(model.getId_pgr()));
            holder.penjelasanproduk.setText(Html.fromHtml(model.getP
enjelasan_produk()));
            holder.merk.setText(model.getMerk());
            Glide.with(context).load(model.getBrowsure_url()).into(h
older.gambarProduk);
            holder.btn.setOnClickListener(new View.OnClickListener()
{
                @Override
                public void onClick(View view) {
                    int invisible = holder.detail.getVisibility();
                    if (invisible == View.VISIBLE) {
                        // Jika LinearLayout terlihat, ubah menjadi
GONE

                        holder.detail.setVisibility(View.GONE);
                        holder.btndetail.setRotation(0);
                    } else {
                        // Jika LinearLayout tidak terlihat, ubah
menjadi VISIBLE

                        holder.detail.setVisibility(View.VISIBLE);
                        holder.btndetail.setRotation(+180);
                    }
                }
            });
            holder.hapus.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    String Data = "PGR";
                    Intent hapus = new Intent(context,
HapusData.class);

                    hapus.putExtra("idHapus", model.getId_pgr());
                    hapus.putExtra("Data", Data);
                    context.startActivity(hapus);
                }
            });
        }
    }
}

```

```

        }
    });
    holder.edit.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        InterfacePgr interfacePgr =
RetroServer.KonesiAPI().create(InterfacePgr.class);
        Call<ModelPgr> update =
interfacePgr.getEdit(Id);
        update.enqueue(new Callback<ModelPgr>() {
            @Override
            public void onResponse(Call<ModelPgr> call,
Response<ModelPgr> response) {
                int varId = model.getId_pgr();
                String varNamaProduk = model.getMerk();
                String varDeskripsi =
model.getPenjelasan_produk();
                String varBrowsure =
model.getBrowsure_url();
                String varJenisProduk = "PGR";

                Intent kirim = new Intent(context,
UbahProduk.class);

                kirim.putExtra("xId", varId);
                kirim.putExtra("xNamaProduk",
varNamaProduk);

                kirim.putExtra("xDeskripsi",
varDeskripsi);

                kirim.putExtra("xBrowsure",
varBrowsure);

                kirim.putExtra("xJenisProduk",
varJenisProduk);

                context.startActivity(kirim);
            }

            @Override
            public void onFailure(Call<ModelPgr> call,
Throwable t) {

            }
        });
    }
});
} else if (getItemViewType(position) == VIEW_TYPE_ITEM_LAST)
{
    // Bind data for the last item

```

```

    }
}

@Override
public int getItemCount() {
    return listData.size() + 1;
}

public class HolderData extends RecyclerView.ViewHolder{
    TextView id, kandungan, penjelasanproduk, merk;
    LinearLayout detail;
    RelativeLayout btn;
    ImageView hapus,edit,gambarProduk, btndetail;
    public HolderData(@NonNull View itemView) {
        super(itemView);
        id = itemView.findViewById(R.id.id);
        merk = itemView.findViewById(R.id.merk);
        detail = itemView.findViewById(R.id.detail);
        kandungan = itemView.findViewById(R.id.kandungan);
        penjelasanproduk =
itemView.findViewById(R.id.penjelasanproduk);
        gambarProduk = itemView.findViewById(R.id.gambarproduk);
        hapus = itemView.findViewById(R.id.hapus);
        edit = itemView.findViewById(R.id.edit);
        btndetail = itemView.findViewById(R.id.btndetail);
        btn = itemView.findViewById(R.id.btn);
    }
}

@Override
public int getItemViewType(int position) {
    if (position == listData.size()) {
        return VIEW_TYPE_ITEM_LAST; // Tampilan terakhir
    }
    return VIEW_TYPE_ITEM; // Tampilan item biasa
}
}
}

```

SourceCode 1. 104 Data Kelola Pgr

s) Data Kelola Produk Baru

```

package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.content.Intent;

```

```

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import
com.example.cakraagroindonesia.Activity.Inovasi.UbahProdukBaru;
import com.example.cakraagroindonesia.HapusData;
import com.example.cakraagroindonesia.Interface.InterfaceProdukBaru;
import com.example.cakraagroindonesia.Model.ModelProdukBaru;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class DataKelolaProdukBaru extends
RecyclerView.Adapter<DataKelolaProdukBaru.HolderData> {
    private Context context;
    private ArrayList<ModelProdukBaru.produk_baru> listData;
    static int Id;

    private static final int VIEW_TYPE_ITEM = 0;
    private static final int VIEW_TYPE_ITEM_LAST = 1;

    public DataKelolaProdukBaru(Context context,
ArrayList<ModelProdukBaru.produk_baru> listData) {
        this.context = context;
        this.listData = listData;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        if (viewType == VIEW_TYPE_ITEM) {
            View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_kelol
aprodukbaru, parent, false);

```

```

        return new HolderData(layout);
    } else if (viewType == VIEW_TYPE_ITEM_LAST) {
        View footer =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_foote
r,parent,false);
        return new HolderData(footer);
    }

    throw new IllegalArgumentException("Invalid view type");
}

@Override
public void onBindViewHolder(@NonNull HolderData holder, int
position) {
    if (getItemViewType(position) == VIEW_TYPE_ITEM) {
        // Bind item data
        ModelProdukBaru.produk_baru model =
listData.get(position);

        holder.tvIdProduk.setText(String.valueOf(model.getId_pro
duk()));
        holder.tvNamaBahan.setText(model.getNama_bahan());
        holder.tvFormulasi.setText(model.getFormulasi());
        holder.tvTanaman.setText(model.getTanaman());
        holder.btnHapus.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String Data = "ProdukBaru";

                Intent hapus = new Intent(context,
HapusData.class);
                hapus.putExtra("idHapus", model.getId_produk());
                hapus.putExtra("Data", Data);
                context.startActivity(hapus);
            }
        });
        holder.btnEdit.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                InterfaceProdukBaru ambildata =
RetroServer.KonesiAPI().create(InterfaceProdukBaru.class);
                Call<ModelProdukBaru> ambil =
ambildata.getEditProdukbaru(Id);
                ambil.enqueue(new Callback<ModelProdukBaru>() {
                    @Override

```

```

        public void onResponse(Call<ModelProdukBaru>
call, Response<ModelProdukBaru> response) {
            int varId = model.getId_produk();
            String varNamaBahan =
model.getNama_bahan();
            String varFormulasi =
model.getFormulasi();
            String varTanaman = model.getTanaman();

            Intent kirim = new Intent(context,
UbahProdukBaru.class);

            kirim.putExtra("xId",varId);
            kirim.putExtra("xNamaBahan",
varNamaBahan);
            kirim.putExtra("xFormulasi",
varFormulasi);
            kirim.putExtra("xTanaman", varTanaman);
            context.startActivity(kirim);
        }

        @Override
        public void onFailure(Call<ModelProdukBaru>
call, Throwable t) {

        }
    });
}

} else if (getItemViewType(position) == VIEW_TYPE_ITEM_LAST)
{
    // Bind data for the last item
}
}

@Override
public int getItemCount() {
    return listData.size() + 1;
}

public class HolderData extends RecyclerView.ViewHolder{
    TextView tvIdProduk, tvNamaBahan, tvFormulasi, tvTanaman;
    ImageView btnHapus, btnEdit;

    public HolderData(@NonNull View itemView) {
        super(itemView);
        tvIdProduk = itemView.findViewById(R.id.idproduk);
    }
}

```

```

        tvNamaBahan = itemView.findViewById(R.id.namabahan);
        tvFormulasi = itemView.findViewById(R.id.formulasi);
        tvTanaman = itemView.findViewById(R.id.tanaman);
        btnHapus = itemView.findViewById(R.id.hapus);
        btnEdit = itemView.findViewById(R.id.edit);
    }
}

@Override
public int getItemViewType(int position) {
    if (position == listData.size()) {
        return VIEW_TYPE_ITEM_LAST; // Tampilan terakhir
    }
    return VIEW_TYPE_ITEM; // Tampilan item biasa
}
}
}

```

SourceCode 1. 105 Data Kelola Produk Baru

t) Data Kelola Produk Homepage

```

package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.API.RetroServer;
import
com.example.cakraagroindonesia.Activity.Beranda.UbahProdukHomepage;
import com.example.cakraagroindonesia.HapusData;
import
com.example.cakraagroindonesia.Interface.InterfaceProdukHomepage;
import com.example.cakraagroindonesia.Model.ModelProdukHomepage;
import com.example.cakraagroindonesia.R;

```

```

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class DataKelolaProdukHomepage extends
RecyclerView.Adapter<DataKelolaProdukHomepage.HolderData>{
    private Context context;
    private ArrayList<ModelProdukHomepage.produk_beranda> listData;
    int Id;

    private float currentRotation = 0;

    private static final int VIEW_TYPE_ITEM = 0;
    private static final int VIEW_TYPE_ITEM_LAST = 1;

    public DataKelolaProdukHomepage(Context context,
ArrayList<ModelProdukHomepage.produk_beranda> listData) {
        this.context = context;
        this.listData = listData;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        if (viewType == VIEW_TYPE_ITEM) {
            View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_kelol
aprodukhomepage,parent,false);
            return new HolderData(view);
        } else if (viewType == VIEW_TYPE_ITEM_LAST) {
            View footer =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_foote
r,parent,false);
            return new HolderData(footer);
        }

        throw new IllegalArgumentException("Invalid view type");
    }

    @Override
    public void onBindViewHolder(@NonNull HolderData holder, int
position) {
        if (getItemViewType(position) == VIEW_TYPE_ITEM) {
            // Bind item data

```



```

        holder.hapus.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String Data = "ProdukHomepage";
                Intent hapus = new Intent(context,
HapusData.class);
                hapus.putExtra("idHapus",
model.getId_produkberanda());
                hapus.putExtra("Data", Data);
                context.startActivity(hapus);
            }
        });

        holder.btn.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View view) {
                int invisible = holder.detail.getVisibility();
                if (invisible == View.VISIBLE) {
                    // Jika LinearLayout terlihat, ubah menjadi
GONE
                    holder.detail.setVisibility(View.GONE);
                    holder.btndetail.setRotation(0);
                } else {
                    // Jika LinearLayout tidak terlihat, ubah
menjadi VISIBLE
                    holder.detail.setVisibility(View.VISIBLE);
                    holder.btndetail.setRotation(+180);
                }
            }
        });
    } else if (getItemViewType(position) == VIEW_TYPE_ITEM_LAST)
{
        // Bind data for the last item
    }

}

@Override
public int getItemCount() {
    return listData.size() + 1;
}

public static class HolderData extends RecyclerView.ViewHolder{
    TextView merkproduk,merk,deskripsi;
    ImageView foto,edit,hapus, btndetail;
}

```

```

RelativeLayout btn;
LinearLayout detail;
public HolderData(@NonNull View itemView) {
    super(itemView);
    merk = itemView.findViewById(R.id.merk);
    merkproduk = itemView.findViewById(R.id.merkproduk);
    deskripsi = itemView.findViewById(R.id.deskripsi);
    edit = itemView.findViewById(R.id.edit);
    hapus = itemView.findViewById(R.id.hapus);
    foto = itemView.findViewById(R.id.foto);
    detail = itemView.findViewById(R.id.detail);
    btndetail = itemView.findViewById(R.id.btndetail);
    btn = itemView.findViewById(R.id.btn);
}
}
@Override
public int getItemViewType(int position) {
    if (position == listData.size()) {
        return VIEW_TYPE_ITEM_LAST; // Tampilan terakhir
    }
    return VIEW_TYPE_ITEM; // Tampilan item biasa
}
}
}

```

SourceCode 1. 106 Data Kelola Produk Homepage

u) Data Kelola QnA

```

package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Activity.Pertanyaan.UbahQNA;

```

```

import com.example.cakraagroindonesia.HapusData;
import com.example.cakraagroindonesia.Interface.InterfaceQna;
import com.example.cakraagroindonesia.Model.ModelQna;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class DataKelolaQna extends
RecyclerView.Adapter<DataKelolaQna.HolderData>{
    private Context context;
    private ArrayList<ModelQna.data_qna> listData;
    int Id;

    private float currentRotation = 0;

    private static final int VIEW_TYPE_ITEM = 0;
    private static final int VIEW_TYPE_ITEM_LAST = 1;

    public DataKelolaQna(Context context,
ArrayList<ModelQna.data_qna> listData) {
        this.context = context;
        this.listData = listData;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        if (viewType == VIEW_TYPE_ITEM) {
            View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_kelol
aqna,parent,false);
            return new HolderData(layout);
        } else if (viewType == VIEW_TYPE_ITEM_LAST) {
            View footer =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_foote
r,parent,false);
            return new HolderData(footer);
        }

        throw new IllegalArgumentException("Invalid view type");
    }
}

```

```

        @Override
        public void onBindViewHolder(@NonNull HolderData holder, int
position) {
            if (getItemViewType(position) == VIEW_TYPE_ITEM) {
                // Bind item data
                ModelQna.data_qna model = listData.get(position);

//                holder.idQna.setText(model.getId_qna());
                holder.pertanyaanQna.setText(model.getPertanyaan());
                holder.namaQna.setText(model.getNama());
                holder.noTelp.setText(model.getTelepon());
                Glide.with(context).load(model.getFoto_url()).into(holder.fotoQna);
                holder.jawabanQna.setText(model.getJawaban());
                holder.btn.setOnClickListener(new View.OnClickListener()
{
                    @Override
                    public void onClick(View view) {
                        int invisible = holder.detail.getVisibility();
                        if (invisible == View.VISIBLE) {
                            // Jika LinearLayout terlihat, ubah menjadi
GONE

                            holder.detail.setVisibility(View.GONE);
                            holder.btndetail.setRotation(0);
                        } else {
                            // Jika LinearLayout tidak terlihat, ubah
menjadi VISIBLE

                            holder.detail.setVisibility(View.VISIBLE);
                            holder.btndetail.setRotation(+180);
                        }
                    }
                });
                holder.btnEdit.setOnClickListener(new
View.OnClickListener() {
                    @Override
                    public void onClick(View view) {
                        InterfaceQna interfaceQna =
RetroServer.KonesiAPI().create(InterfaceQna.class);
                        Call<ModelQna> edit = interfaceQna.getEdit(Id);
                        edit.enqueue(new Callback<ModelQna>() {
                            @Override
                            public void onResponse(Call<ModelQna> call,
Response<ModelQna> response) {
                                int varId = model.getId_qna();
                                String varNama = model.getNama();
                                String varNotelp = model.getTelepon();

```

```

        String varPertanyaan =
model.getPertanyaan();

        Intent kirim = new Intent(context,
UbahQNA.class);

        kirim.putExtra("xId", varId);
        kirim.putExtra("xNama", varNama);
        kirim.putExtra("xNoTelp", varNotelp);
        kirim.putExtra("xPertanyaan",
varPertanyaan);

        context.startActivity(kirim);
    }

    @Override
    public void onFailure(Call<ModelQna> call,
Throwable t) {

    }

    });
}

holder.btnHapus.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String Data = "QNA";

        Intent hapus = new Intent(context,
HapusData.class);

        hapus.putExtra("idHapus", model.getId_qna());
        hapus.putExtra("Data", Data);
        context.startActivity(hapus);
    }
});

} else if (getItemViewType(position) == VIEW_TYPE_ITEM_LAST)
{
    // Bind data for the last item
}

}

@Override
public int getItemCount() {
    return listData.size() + 1;
}
}

```

```

public class HolderData extends RecyclerView.ViewHolder{
    LinearLayout detail;
    RelativeLayout btn;
    TextView idQna,namaQna, pertanyaanQna, jawabanQna, noTelp;
    ImageView btnHapus, btnEdit, fotoQna, btndetail;
    public HolderData(@NonNull View itemView) {
        super(itemView);

        idQna = itemView.findViewById(R.id.id);
        detail = itemView.findViewById(R.id.detail);
        namaQna = itemView.findViewById(R.id.nama);
        pertanyaanQna = itemView.findViewById(R.id.pertanyaan);
        fotoQna = itemView.findViewById(R.id.fotoqna);
        jawabanQna = itemView.findViewById(R.id.jawaban);
        noTelp = itemView.findViewById(R.id.notelp);
        btnHapus = itemView.findViewById(R.id.btnhapus);
        btnEdit = itemView.findViewById(R.id.btndedit);
        btndetail = itemView.findViewById(R.id.btndetail);
        btn = itemView.findViewById(R.id.btn);

    }
}
@Override
public int getItemViewType(int position) {
    if (position == listData.size()) {
        return VIEW_TYPE_ITEM_LAST; // Tampilan terakhir
    }
    return VIEW_TYPE_ITEM; // Tampilan item biasa
}
}

```

SourceCode 1. 107 Data Kelola QnA

v) Data Manager

```

package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;

```

```

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.API.RetroServer;
import
com.example.cakraagroindonesia.Activity.KelolaDataMaster.UbahManager
;
import com.example.cakraagroindonesia.HapusData;
import com.example.cakraagroindonesia.Interface.InterfaceManager;
import com.example.cakraagroindonesia.Model.ModelManager;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class DataManager extends
RecyclerView.Adapter<DataManager.HolderData>{
    private Context context;
    private ArrayList<ModelManager.manager> listData;

    private float currentRotation = 0;

    private static final int VIEW_TYPE_ITEM = 0;
    private static final int VIEW_TYPE_ITEM_LAST = 1;

    public DataManager(Context context,
ArrayList<ModelManager.manager> listData) {
        this.context = context;
        this.listData = listData;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        if (viewType == VIEW_TYPE_ITEM) {
            View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_kelol
amanager,parent,false);
            return new HolderData(layout);
        } else if (viewType == VIEW_TYPE_ITEM_LAST) {

```



```

        View footer =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_foote
r,parent,false);
        return new HolderData(footer);
    }

    throw new IllegalArgumentException("Invalid view type");

}

@Override
public void onBindViewHolder(@NonNull HolderData holder, int
position) {
    if (getItemViewType(position) == VIEW_TYPE_ITEM) {
        // Bind item data
        ModelManager.manager model = listData.get(position);

        holder.nama.setText(model.getNama_manager());
        Glide.with(context).load(model.getFoto_url()).into(holde
r.foto);

        holder.id.setText(model.getKode_mg());
        holder.manager.setText(model.getNama_manager());
        holder.username.setText(model.getUsername());
        holder.password.setText(model.getPassword());

        holder.btnEdit.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                InterfaceManager interfaceManager =
RetroServer.KonesiAPI().create(InterfaceManager.class);
                Call<ModelManager> call =
interfaceManager.getEdit(model.getKode_mg());
                call.enqueue(new Callback<ModelManager>() {
                    @Override
                    public void onResponse(Call<ModelManager>
call, Response<ModelManager> response) {
                        String varId = model.getKode_mg();
                        String varNama =
model.getNama_manager();
                        String varUsername =
model.getUsername();
                        String varPassword =
model.getPassword();
                        String varTFoto = model.getFoto_url();
                        String varFoto = model.getFotomanager();
                    }
                });
            }
        });
    }
}

```

```

        Intent kirim = new Intent(context,
UbahManager.class);

        kirim.putExtra("xId", varId);
        kirim.putExtra("xNama", varNama);
        kirim.putExtra("xUsername",
varUsername);
        kirim.putExtra("xPassword",
varPassword);

        kirim.putExtra("xTFoto", varTFoto);
        kirim.putExtra("xFoto", varFoto);
        context.startActivity(kirim);

    }

    @Override
    public void onFailure(Call<ModelManager>
call, Throwable t) {

    }

    });
}

holder.btnHapus.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String Data = "Manager";
        Intent hapus = new Intent(context,
HapusData.class);

        hapus.putExtra("IDhapus", model.getKode_mg());
        hapus.putExtra("Data", Data);
        context.startActivity(hapus);

    }

});
holder.btn.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view) {
        rotateImage(view);
        int invisible = holder.detail.getVisibility();
        if (invisible == View.VISIBLE) {
            // Jika LinearLayout terlihat, ubah menjadi
GONE

            holder.detail.setVisibility(View.GONE);
        } else {
            // Jika LinearLayout tidak terlihat, ubah
menjadi VISIBLE

```

```

        holder.detail.setVisibility(View.VISIBLE);
    }
}
public void rotateImage(View view) {
    currentRotation += 180;
    holder.btndetail.setRotation(currentRotation);
}
});
} else if (getItemViewType(position) == VIEW_TYPE_ITEM_LAST)
{
    // Bind data for the last item
}
}

@Override
public int getItemCount() {
    return listData.size() + 1;
}

public class HolderData extends RecyclerView.ViewHolder{
    TextView nama,id,manager,username,password;
    ImageView btnHapus, btnEdit,foto, btndetail;
    LinearLayout show,detail;
    RelativeLayout btn;
    public HolderData(@NonNull View itemView) {
        super(itemView);

        nama = itemView.findViewById(R.id.nama);
        foto = itemView.findViewById(R.id.foto);
        id = itemView.findViewById(R.id.id);
        manager = itemView.findViewById(R.id.manager);
        username = itemView.findViewById(R.id.username);
        password = itemView.findViewById(R.id.password);
        btnEdit = itemView.findViewById(R.id.btndetail);
        btnHapus = itemView.findViewById(R.id.btnhapus);
        detail = itemView.findViewById(R.id.detail);
        btndetail = itemView.findViewById(R.id.btndetail);
        btn = itemView.findViewById(R.id.btn);
    }
}
@Override
public int getItemViewType(int position) {
    if (position == listData.size()) {
        return VIEW_TYPE_ITEM_LAST; // Tampilan terakhir
    }
    return VIEW_TYPE_ITEM; // Tampilan item biasa
}
}

```

```

public String getLastId(){
    if (listData != null && listData.size() > 0){
        int lastIndex = listData.size()-1;
        ModelManager.manager lastData = listData.get(lastIndex);
        return lastData.getKode_mg();
    }
    return null;
}
}

```

SourceCode 1. 108 Data Manager

w) Data Notifikasi

```

package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.RelativeLayout;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.API.RetroServer;
import
com.example.cakraagroindonesia.Activity.Notifikasi.Notifikasi;
import com.example.cakraagroindonesia.Interface.InterfaceNotifikasi;
import com.example.cakraagroindonesia.Model.ModelNotifikasi;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class DataNotifikasi extends
RecyclerView.Adapter<DataNotifikasi.HolderData>{
    private Context context;
    private ArrayList<ModelNotifikasi.notifikasi> listData;
    int Id;
}

```

```

private static final int VIEW_TYPE_ITEM = 0;
private static final int VIEW_TYPE_ITEM_LAST = 1;

public DataNotifikasi(Context context,
ArrayList<ModelNotifikasi.notifikasi> listData) {
    this.context = context;
    this.listData = listData;
}

@NonNull
@Override
public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
    if (viewType == VIEW_TYPE_ITEM) {
        View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_notif
ifikasi,parent,false);
        return new HolderData(layout);
    } else if (viewType == VIEW_TYPE_ITEM_LAST) {
        View footer =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_foote
r,parent,false);
        return new HolderData(footer);
    }

    throw new IllegalArgumentException("Invalid view type");
}

@Override
public void onBindViewHolder(@NonNull HolderData holder, int
position) {
    if (getItemViewType(position) == VIEW_TYPE_ITEM) {
        // Bind item data
        ModelNotifikasi.notifikasi model =
listData.get(position);
        holder.jenisnotifikasi.setText(model.getJenis());
        holder.namanotifikasi.setText(model.getNama_notif());
        holder.komentar.setText(model.getKomentar());
        holder.tanggal.setText(model.getTime());

        InterfaceNotifikasi interfaceNotifikasi =
RetroServer.KonesiAPI().create(InterfaceNotifikasi.class);
        Call<ModelNotifikasi> call =
interfaceNotifikasi.getStatusNotifikasi(Id);
        call.enqueue(new Callback<ModelNotifikasi>() {
            @Override

```

```

        public void onResponse(Call<ModelNotifikasi> call,
Response<ModelNotifikasi> response) {
            ModelNotifikasi.notifikasi status =
listData.get(Id);

            if (status.getStatus_admin().equals("baru")){
                holder.notiflama.setVisibility(View.GONE);
                holder.notifbaru.setVisibility(View.VISIBLE)
;

                holder.btntinjau.setOnClickListener(new
View.OnClickListener() {
                    @Override
                    public void onClick(View view) {
                        String lama = "lama";
                        status.setStatus_admin(lama);

                        context.startActivity(new
Intent(context, Notifikasi.class));
                    }
                });

                }else if
(status.getStatus_admin().equals("lama")){
                    holder.notiflama.setVisibility(View.VISIBLE)
;

                    holder.notifbaru.setVisibility(View.GONE);

                    holder.btntinjau.setOnClickListener(new
View.OnClickListener() {
                        @Override
                        public void onClick(View view) {
                            String baru = "baru";
                            status.setStatus_admin(baru);

                            context.startActivity(new
Intent(context, Notifikasi.class));
                        }
                    });
                }else{
                }
            }
            @Override
            public void onFailure(Call<ModelNotifikasi> call,
Throwable t) {
            }
        });

```

```

        holder.btn.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view) {
        int invisible =
holder.btntinjau.getVisibility();
        if (invisible == View.VISIBLE) {
            // Jika LinearLayout terlihat, ubah menjadi
GONE

            holder.btntinjau.setVisibility(View.GONE);
            holder.btndetail.setRotation(0);
        } else {
            // Jika LinearLayout tidak terlihat, ubah
menjadi VISIBLE

            holder.btntinjau.setVisibility(View.VISIBLE)
;
            holder.btndetail.setRotation(+180);
        }
    }
});
    } else if (getItemViewType(position) == VIEW_TYPE_ITEM_LAST)
{
    // Bind data for the last item
    }
}

@Override
public int getItemCount() {
    return listData.size() + 1;
}

public class HolderData extends RecyclerView.ViewHolder{
    TextView jenisnotifikasi, namanotifikasi, komentar, tanggal;
    ImageView notiflama, notifbaru, btndetail, btntinjau;
    RelativeLayout btn;
    public HolderData(@NonNull View itemView) {
        super(itemView);
        jenisnotifikasi =
itemView.findViewById(R.id.jenisnotifikasi);
        namanotifikasi =
itemView.findViewById(R.id.namanotifikasi);
        komentar = itemView.findViewById(R.id.komentar);
        tanggal = itemView.findViewById(R.id.tanggal);
        notiflama = itemView.findViewById(R.id.notiflama);
        notifbaru = itemView.findViewById(R.id.notifbaru);
        btndetail = itemView.findViewById(R.id.btndetail);
    }
}

```

```

        btntinjau = itemView.findViewById(R.id.btntinjau);
        btn = itemView.findViewById(R.id.btn);
    }
}
@Override
public int getItemViewType(int position) {
    if (position == listData.size()) {
        return VIEW_TYPE_ITEM_LAST; // Tampilan terakhir
    }
    return VIEW_TYPE_ITEM; // Tampilan item biasa
}
}
}

```

SourceCode 1. 109 Data Notifikasi

x) Data Paket Produk

```

package com.example.cakraagroindonesia.Adapter;

import android.app.DownloadManager;
import android.content.Context;
import android.net.Uri;
import android.os.Environment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.Interface.OnClickPaketProduk;
import com.example.cakraagroindonesia.Model.ModelPaketProduk;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

public class DataPaketProduk extends
RecyclerView.Adapter<DataPaketProduk.HolderData> {
    private Context context;
    private ArrayList<ModelPaketProduk.paket_produk> listData;
    private OnClickPaketProduk onClickPaketProduk;
}

```



```

    public DataPaketProduk(Context context,
ArrayList<ModelPaketProduk.paket_produk> listData,
OnClickPaketProduk onClickPaketProduk) {
    this.context = context;
    this.listData = listData;
    this.onClickPaketProduk = onClickPaketProduk;
}

@NonNull
@Override
public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
    View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_paket
produk,parent,false);
    return new HolderData(layout);
}

@Override
public void onBindViewHolder(@NonNull HolderData holder, int
position) {
    ModelPaketProduk.paket_produk model =
listData.get(position);

    holder.tvIdPaket.setText(String.valueOf(model.getId_paket()
));
    holder.tvPaketProduk.setText(model.getPaket_produk());
    holder.tvTanaman.setText(model.getTanaman());
    holder.tvIterasi.setText(model.getIterasi());
    holder.tvHasil.setText(model.getHasil());
    holder.ibPpt.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String pptUrl = model.getFoto_url(); // Ambil URL
PPT dari data paket
            DownloadManager.Request request = new
DownloadManager.Request(Uri.parse(pptUrl));
            request.setTitle(model.getPpt());
            request.setDescription("Mengunduh file PPT...");
            request.setNotificationVisibility(DownloadManager.Re
quest.VISIBILITY_VISIBLE_NOTIFY_COMPLETED);
            request.setDestinationInExternalPublicDir(Environmen
t.DIRECTORY_DOWNLOADS, model.getPpt());

            DownloadManager downloadManager = (DownloadManager)
context.getSystemService(Context.DOWNLOAD_SERVICE);
            downloadManager.enqueue(request);

```

```

        }
    });
}

@Override
public int getItemCount() {
    return listData.size();
}

public class HolderData extends RecyclerView.ViewHolder{
    TextView tvIdPaket, tvPaketProduk, tvTanaman, tvIterasi,
tvHasil, ibPpt;
    public HolderData(@NonNull View itemView) {
        super(itemView);
        tvIdPaket = itemView.findViewById(R.id.idpaket);
        tvPaketProduk = itemView.findViewById(R.id.paketproduk);
        tvTanaman = itemView.findViewById(R.id.tanaman);
        tvIterasi = itemView.findViewById(R.id.iterasi);
        tvHasil = itemView.findViewById(R.id.hasil);
        ibPpt = itemView.findViewById(R.id.btnlihatppt);
    }
}
}
}

```

SourceCode 1. 110 Data Paket Produk

y) Data Pgr

```

package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.Model.ModelPgr;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

```

```

public class DataPgr extends
RecyclerView.Adapter<DataPgr.HolderData> {
    private Context context;
    private ArrayList<ModelPgr.data_pgr> listData;
    private OnItemClickListener onItemClick;

    public interface OnItemClickListener {
        void onItemClick(ModelPgr.data_pgr modelPgr);
    }

    public DataPgr(Context context, ArrayList<ModelPgr.data_pgr>
listData, OnItemClickListener onItemClick) {
        this.context = context;
        this.listData = listData;
        this.onItemClick = onItemClick;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_pgr,p
arent,false);
        return new HolderData(layout);
    }

    @Override
    public void onBindViewHolder(@NonNull HolderData holder, int
position) {
        ModelPgr.data_pgr model = listData.get(position);

        holder.id.setText(String.valueOf(model.getId_pgr()));
        holder.penjelasanproduk.setText(model.getPenjelasan_produk()
);
        holder.merk.setText(model.getMerk());
        Glide.with(context).load(model.getBrowsure_url()).into(holder.gambarproduk);
        holder.gambarproduk.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                onItemClick.onItemClick(model);
            }
        });
    }
}

```

```

@Override
public int getItemCount() {
    return listData.size();
}

public class HolderData extends RecyclerView.ViewHolder{
    TextView id, penjelasanproduk, merk;
    ImageView gambarproduk;

    public HolderData(@NonNull View itemView) {
        super(itemView);
        id = itemView.findViewById(R.id.id);
        penjelasanproduk =
itemView.findViewById(R.id.penjelasanproduk);
        merk = itemView.findViewById(R.id.merk);
        gambarproduk = itemView.findViewById(R.id.gambarproduk);
    }
}
}

```

SourceCode 1. 111 Data Pgr

z) Data Produk Baru

```

package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.example.cakraagroindonesia.Model.ModelProdukBaru;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

public class DataProdukBaru extends
RecyclerView.Adapter<DataProdukBaru.HolderData> {
    private Context context;
    private ArrayList<ModelProdukBaru.produk_baru> listData;

```

```

    public DataProdukBaru(Context context,
ArrayList<ModelProdukBaru.produk_baru> listData) {
        this.context = context;
        this.listData = listData;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_produ
kbaru,parent,false);
        return new HolderData(layout);
    }

    @Override
    public void onBindViewHolder(@NonNull HolderData holder, int
position) {
        ModelProdukBaru.produk_baru model = listData.get(position);

        holder.tvIdProduk.setText(String.valueOf(model.getId_produk(
)));
        holder.tvNamaBahan.setText(model.getNama_bahan());
        holder.tvFormulasi.setText(model.getFormulasi());
        holder.tvTanaman.setText(model.getTanaman());
    }

    @Override
    public int getItemCount() {
        return listData.size();
    }

    public class HolderData extends RecyclerView.ViewHolder{
        TextView tvIdProduk, tvNamaBahan, tvFormulasi, tvTanaman;

        public HolderData(@NonNull View itemView) {
            super(itemView);
            tvIdProduk = itemView.findViewById(R.id.idproduk);
            tvNamaBahan = itemView.findViewById(R.id.namabahan);
            tvFormulasi = itemView.findViewById(R.id.formulasi);
            tvTanaman = itemView.findViewById(R.id.tanaman);
        }
    }
}

```

SourceCode 1. 112 Data Produk Baru

aa) Data QnA

```
package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.Model.ModelQna;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

public class DataQna extends
RecyclerView.Adapter<DataQna.HolderData>{
    private Context context;
    private ArrayList<ModelQna.data_qna> listData;

    public DataQna(Context context, ArrayList<ModelQna.data_qna>
listData) {
        this.context = context;
        this.listData = listData;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_qna,p
arent,false);
        return new HolderData(layout);
    }

    @Override
    public void onBindViewHolder(@NonNull HolderData holder, int
position) {
        ModelQna.data_qna model = listData.get(position);
```

```

//      holder.idQna.setText(model.getId_qna());
holder.pertanyaanQna.setText(model.getPertanyaan());
holder.namaQna.setText(model.getNama());
Glide.with(context).load(model.getFoto_url()).into(holder.fotoQna);
holder.jawabanQna.setText(model.getJawaban());
holder.tampilDetail.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        int invisible = holder.detail.getVisibility();
        if (invisible == View.VISIBLE) {
            // Jika LinearLayout terlihat, ubah menjadi GONE
            holder.detail.setVisibility(View.GONE);
        } else {
            // Jika LinearLayout tidak terlihat, ubah
menjadi VISIBLE
            holder.detail.setVisibility(View.VISIBLE);
        }
    }
});
}

@Override
public int getItemCount() {
    return listData.size();
}

public class HolderData extends RecyclerView.ViewHolder{
    RelativeLayout tampilDetail;
    LinearLayout detail;
    TextView idQna,namaQna, pertanyaanQna, jawabanQna;
    ImageView fotoQna;
    public HolderData(@NonNull View itemView) {
        super(itemView);

        idQna = itemView.findViewById(R.id.id);
        tampilDetail = itemView.findViewById(R.id.tampildetail);
        detail = itemView.findViewById(R.id.detail);
        namaQna = itemView.findViewById(R.id.nama);
        pertanyaanQna = itemView.findViewById(R.id.pertanyaan);
        fotoQna = itemView.findViewById(R.id.fotoqna);
        jawabanQna = itemView.findViewById(R.id.jawaban);

    }
}
}

```

```
}
```

SourceCode 1. 113 Data QnA

bb) Data Report

```
package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.HapusData;
import com.example.cakraagroindonesia.Activity.Report.UbahReport;
import com.example.cakraagroindonesia.Model.ModelReport;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

public class DataReport extends
RecyclerView.Adapter<DataReport.HolderData>{
    private Context context;
    private ArrayList<ModelReport.report> listData;

    private static final int VIEW_TYPE_ITEM = 0;
    private static final int VIEW_TYPE_ITEM_LAST = 1;

    public DataReport(Context context, ArrayList<ModelReport.report>
listData) {
        this.context = context;
        this.listData = listData;
    }

    @NonNull
    @Override
```



```

    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        if (viewType == VIEW_TYPE_ITEM) {
            View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_kelol
areport,parent,false);
            return new HolderData(layout);
        } else if (viewType == VIEW_TYPE_ITEM_LAST) {
            View footer =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_foote
r,parent,false);
            return new HolderData(footer);
        }

        throw new IllegalArgumentException("Invalid view type");
    }

    @Override
    public void onBindViewHolder(@NonNull HolderData holder, int
position) {
        if (getItemViewType(position) == VIEW_TYPE_ITEM) {
            // Bind item data
            ModelReport.model = listData.get(position);

            holder.supervisor.setText(model.getNama_supervisor());
            holder.demonstrator.setText(model.getNama_demonstrator(
));

            holder.provinsi.setText(model.getProvinsi_ds());
            holder.kabupatenDs.setText(model.getKabupaten_ds());
            holder.telepon.setText(model.getNomor_telpon());
            holder.petani.setText(model.getNama_petani());
            holder.produk.setText(model.getProduk());
            holder.dosis.setText(model.getDosis());
            holder.tanaman.setText(model.getTanaman());
            holder.kabupaten.setText(model.getKabupaten());
            holder.kecamatan.setText(model.getKecamatan());
            holder.desa.setText(model.getDesa());
            Glide.with(context).load(model.getFoto_url()).into(holde
r.foto);

            holder.btnEdit.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    String varId = model.getKode_report();

```

```

        String varTanggalReport =
model.getTanggal_demplot();
        String varNotelp = model.getNomor_telpon();
        String varNamaPetani = model.getNama_petani();
        String varProduk = model.getProduk();
        String varDosis = model.getDosis();
        String varTanaman = model.getTanaman();
        String varKabupaten = model.getKabupaten();
        String varKecamatan = model.getKecamatan();
        String varDesa = model.getDesa();
        String varResult = model.getResult();
        String varStatus = model.getStatus();
        String varStatusApps = model.getStatus_apps();

        Intent kirim = new Intent(context,
UbahReport.class);
        kirim.putExtra("xId", varId);
        kirim.putExtra("xTanggalReport",
varTanggalReport);
        kirim.putExtra("xNotelp", varNotelp);
        kirim.putExtra("xNamaPetani", varNamaPetani);
        kirim.putExtra("xProduk", varProduk);
        kirim.putExtra("xDosis", varDosis);
        kirim.putExtra("xTanaman", varTanaman);
        kirim.putExtra("xKabupaten", varKabupaten);
        kirim.putExtra("xKecamatan", varKecamatan);
        kirim.putExtra("xDesa", varDesa);
        kirim.putExtra("xResult", varResult);
        kirim.putExtra("xStatus", varStatus);
        kirim.putExtra("xStatusApps", varStatusApps);
        context.startActivity(kirim);

    }
});

holder.btnHapus.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String Data = "Report";
        Intent hapus = new Intent(context,
HapusData.class);
        hapus.putExtra("IDhapus",
model.getKode_report());
        hapus.putExtra("Data", Data);
        context.startActivity(hapus);
    }
}

```

```

    });

    holder.btn.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view) {
        int invisible = holder.detail.getVisibility();
        if (invisible == View.VISIBLE) {
            // Jika LinearLayout terlihat, ubah menjadi
GONE

            holder.detail.setVisibility(View.GONE);
            holder.btndetail.setRotation(0);
        } else {
            // Jika LinearLayout tidak terlihat, ubah
menjadi VISIBLE

            holder.detail.setVisibility(View.VISIBLE);
            holder.btndetail.setRotation(+180);
        }
    }
});
} else if (getItemViewType(position) == VIEW_TYPE_ITEM_LAST)
{
    // Bind data for the last item
}

}

@Override
public int getItemCount() {
    return listData.size() + 1;
}

public class HolderData extends RecyclerView.ViewHolder{
    TextView
supervisor,demonstrator,provinsi,kabupatenDs,telepon,petani,produk,d
osis,tanaman,kabupaten,kecamatan,desa;
    ImageView btnHapus,btnEdit,foto, btndetail;
    LinearLayout detail;
    RelativeLayout btn;

    public HolderData(@NonNull View itemView) {
        super(itemView);
        supervisor = itemView.findViewById(R.id.supervisor);
        demonstrator = itemView.findViewById(R.id.demonstrator);
        provinsi = itemView.findViewById(R.id.provinsi);
        kabupatenDs = itemView.findViewById(R.id.kabupatenDs);
        telepon = itemView.findViewById(R.id.notelp);

```

```

        petani = itemView.findViewById(R.id.petani);
        produk = itemView.findViewById(R.id.produk);
        dosis = itemView.findViewById(R.id.dosis);
        tanaman = itemView.findViewById(R.id.tanaman);
        kabupaten = itemView.findViewById(R.id.kabupaten);
        kecamatan = itemView.findViewById(R.id.kecamatan);
        desa = itemView.findViewById(R.id.desa);
        foto = itemView.findViewById(R.id.foto);
        btnEdit = itemView.findViewById(R.id.btnedit);
        btnHapus = itemView.findViewById(R.id.btnhapus);
        detail = itemView.findViewById(R.id.detail);
        btndetail = itemView.findViewById(R.id.btndetail);
        btn = itemView.findViewById(R.id.btn);
    }
}
@Override
public int getItemViewType(int position) {
    if (position == listData.size()) {
        return VIEW_TYPE_ITEM_LAST; // Tampilan terakhir
    }
    return VIEW_TYPE_ITEM; // Tampilan item biasa
}
}

```

SourceCode 1. 114 Data Report

cc)Data Secretary

```

package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.API.RetroServer;

```

```

import
com.example.cakraagroindonesia.Activity.KelolaDataMaster.UbahDataSecretary;
import com.example.cakraagroindonesia.HapusData;
import com.example.cakraagroindonesia.Interface.InterfaceSecretary;
import com.example.cakraagroindonesia.Model.ModelSecretary;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class DataSecretary extends
RecyclerView.Adapter<DataSecretary.HolderData>{
    private Context context;
    private ArrayList<ModelSecretary.secretary> listData;

    private float currentRotation = 0;

    private static final int VIEW_TYPE_ITEM = 0;
    private static final int VIEW_TYPE_ITEM_LAST = 1;

    public DataSecretary(Context context,
ArrayList<ModelSecretary.secretary> listData) {
        this.context = context;
        this.listData = listData;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        if (viewType == VIEW_TYPE_ITEM) {
            View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_kelolasecretary,parent,false);
            return new HolderData(layout);
        } else if (viewType == VIEW_TYPE_ITEM_LAST) {
            View footer =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_footer,parent,false);
            return new HolderData(footer);
        }

        throw new IllegalArgumentException("Invalid view type");
    }
}

```

```

    }

    @Override
    public void onBindViewHolder(@NonNull HolderData holder, int
position) {
        if (getItemViewType(position) == VIEW_TYPE_ITEM) {
            // Bind item data
            ModelSecretary.secretary model = listData.get(position);

            holder.nama.setText(model.getNama_secretary());
            holder.id.setText(model.getKode_sc());
            holder.secretary.setText(model.getNama_secretary());
            holder.manager.setText(model.getNama_manager());
            holder.username.setText(model.getUsername());
            holder.password.setText(model.getPassword());
            Glide.with(context).load(model.getFoto_url()).into(holder.foto);

            holder.btnEdit.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    InterfaceSecretary interfaceSecretary =
RetroServer.KonesiAPI().create(InterfaceSecretary.class);
                    Call<ModelSecretary> kirim =
interfaceSecretary.getEdit(model.getKode_sc());
                    kirim.enqueue(new Callback<ModelSecretary>() {
                        @Override
                        public void onResponse(Call<ModelSecretary>
call, Response<ModelSecretary> response) {
                            String varId = model.getKode_sc();
                            String varNama =
model.getNama_secretary();
                            String varNamaManager =
model.getNama_manager();
                            String varUsername =
model.getUsername();
                            String varPassword =
model.getPassword();
                            String varTFoto = model.getFoto_url();
                            String varFoto =
model.getFotosecretary();

                            Intent kirim = new Intent(context,
UbahDataSecretary.class);
                            kirim.putExtra("xId", varId);

```

```

        kirim.putExtra("xNama", varNama);
        kirim.putExtra("xNamaManager",
varNamaManager);
        kirim.putExtra("xUsername",
varUsername);
        kirim.putExtra("xPassword",
varPassword);
        kirim.putExtra("xTFoto", varTFoto);
        kirim.putExtra("xFoto", varFoto);
        context.startActivity(kirim);
    }

    @Override
    public void onFailure(Call<ModelSecretary>
call, Throwable t) {
    }
    });
    }
    });
    holder.btnHapus.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String Data = "Secretary";
            Intent hapus = new Intent(context,
HapusData.class);
            hapus.putExtra("IDhapus", model.getKode_sc());
            hapus.putExtra("Data", Data);
            context.startActivity(hapus);
        }
    });
    holder.btn.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view) {
        int invisible = holder.detail.getVisibility();
        if (invisible == View.VISIBLE) {
            // Jika LinearLayout terlihat, ubah menjadi
GONE

            holder.detail.setVisibility(View.GONE);
            holder.btndetail.setRotation(0);
        } else {
            // Jika LinearLayout tidak terlihat, ubah
menjadi VISIBLE

            holder.detail.setVisibility(View.VISIBLE);
            holder.btndetail.setRotation(+180);

```

```

        }
    }
    });
    } else if (getItemViewType(position) == VIEW_TYPE_ITEM_LAST)
{
    // Bind data for the last item
    }
}

@Override
public int getItemCount() {
    return listData.size() + 1;
}

public class HolderData extends RecyclerView.ViewHolder{
    TextView nama,id,secretary,manager,username,password;
    ImageView btnEdit, btnHapus, foto, btndetail;
    RelativeLayout btn;
    LinearLayout show,detail;

    public HolderData(@NonNull View itemView) {
        super(itemView);
        nama = itemView.findViewById(R.id.nama);
        foto = itemView.findViewById(R.id.foto);
        id = itemView.findViewById(R.id.id);
        secretary = itemView.findViewById(R.id.secretary);
        manager = itemView.findViewById(R.id.manager);
        username = itemView.findViewById(R.id.username);
        password = itemView.findViewById(R.id.password);
        btnEdit = itemView.findViewById(R.id.btncedit);
        btnHapus = itemView.findViewById(R.id.btnhapus);
        detail = itemView.findViewById(R.id.detail);
        btndetail = itemView.findViewById(R.id.btndetail);
        btn = itemView.findViewById(R.id.btn);
    }
}

@Override
public int getItemViewType(int position) {
    if (position == listData.size()) {
        return VIEW_TYPE_ITEM_LAST; // Tampilan terakhir
    }
    return VIEW_TYPE_ITEM; // Tampilan item biasa
}

public String getLastId(){
    if (listData != null && listData.size() > 0){
        int lastIndex = listData.size()-1;

```



```

        ModelSecretary.secretary lastData =
listData.get(lastIndex);
        return lastData.getKode_sc();
    }
    return null;
}
}

```

SourceCode 1. 115 Data Secretary

dd) Data Supervisor

```

package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.API.RetroServer;
import
com.example.cakraagroindonesia.Activity.KelolaDataMaster.UbahDataSup
ervisor;
import com.example.cakraagroindonesia.HapusData;
import com.example.cakraagroindonesia.Interface.InterfaceSupervisor;
import com.example.cakraagroindonesia.Model.ModelSupervisor;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class DataSupervisor extends
RecyclerView.Adapter<DataSupervisor.HolderData>{
    private Context context;

```

```

private ArrayList<ModelSupervisor.supervisor> listData;

private float currentRotation = 0;

private static final int VIEW_TYPE_ITEM = 0;
private static final int VIEW_TYPE_ITEM_LAST = 1;

public DataSupervisor(Context context,
ArrayList<ModelSupervisor.supervisor> listData) {
    this.context = context;
    this.listData = listData;
}

@NonNull
@Override
public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
    if (viewType == VIEW_TYPE_ITEM) {
        View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_kelola
supervisor,parent,false);
        return new HolderData(layout);
    } else if (viewType == VIEW_TYPE_ITEM_LAST) {
        View footer =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_foote
r,parent,false);
        return new HolderData(footer);
    }

    throw new IllegalArgumentException("Invalid view type");
}

@Override
public void onBindViewHolder(@NonNull HolderData holder, int
position) {
    if (getItemViewType(position) == VIEW_TYPE_ITEM) {
        // Bind item data
        ModelSupervisor.supervisor model =
listData.get(position);

        holder.nama.setText(model.getNama_supervisor());
        holder.supervisor.setText(model.getNama_supervisor());
        holder.id.setText(model.getKode_sv());
        holder.budget.setText(model.getBudget_tersedia());
        holder.areaSales.setText(model.getArea_sales());
        holder.provinsi.setText(model.getProvinsi());
    }
}

```

```

        holder.manager.setText(model.getNama_manager());
        holder.username.setText(model.getUsername());
        holder.password.setText(model.getPassword());
        Glide.with(context).load(model.getFoto_url()).into(holder.foto);

        holder.btnEdit.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                InterfaceSupervisor interfaceSupervisor =
RetroServer.KonesiAPI().create(InterfaceSupervisor.class);
                Call<ModelSupervisor> kirim =
interfaceSupervisor.getEdit(model.getKode_sv());
                kirim.enqueue(new Callback<ModelSupervisor>() {
                    @Override
                    public void onResponse(Call<ModelSupervisor>
call, Response<ModelSupervisor> response) {
                        String varId = model.getKode_sv();
                        String varNama =
model.getNama_supervisor();
                        String varNamaManager =
model.getNama_manager();
                        String varBudget =
model.getBudget_tersedia();
                        String varAreaSales =
model.getArea_sales();
                        String varProvinsi =
model.getProvinsi();
                        String varUsername =
model.getUsername();
                        String varPassword =
model.getPassword();
                        String varTFoto = model.getFoto_url();
                        String varFoto =
model.getFotosupervisor();

                        Intent kirim = new Intent(context,
UbahDataSupervisor.class);
                        kirim.putExtra("xId", varId);
                        kirim.putExtra("xNama", varNama);
                        kirim.putExtra("xNamaManager", varNamaMan
ager);
                        kirim.putExtra("xBudget", varBudget);
                        kirim.putExtra("xAreaSales", varAreaSales
);
                        kirim.putExtra("xProvinsi", varProvinsi);

```

```

        kirim.putExtra("xUsername", varUsername);
        kirim.putExtra("xPassword", varPassword);
        kirim.putExtra("xTFoto", varTFoto);
        kirim.putExtra("xFoto", varFoto);
        context.startActivity(kirim);
    }

    @Override
    public void onFailure(Call<ModelSupervisor>
call, Throwable t) {
        }
    });
}
});
holder.btnHapus.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String Data = "Supervisor";

        Intent hapus = new Intent(context,
HapusData.class);
        hapus.putExtra("IDhapus", model.getKode_sv());
        hapus.putExtra("Data", Data);
        context.startActivity(hapus);
    }
});
holder.btn.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view) {
        int invisible = holder.detail.getVisibility();
        if (invisible == View.VISIBLE) {
            // Jika LinearLayout terlihat, ubah menjadi
GONE

            holder.detail.setVisibility(View.GONE);
            holder.btndetail.setRotation(0);
        } else {
            // Jika LinearLayout tidak terlihat, ubah
menjadi VISIBLE

            holder.detail.setVisibility(View.VISIBLE);
            holder.btndetail.setRotation(+180);
        }
    }
});
});

```

```

        } else if (getItemViewType(position) == VIEW_TYPE_ITEM_LAST)
    {
        // Bind data for the last item
    }

}

@Override
public int getItemCount() {
    return listData.size() + 1;
}

public class HolderData extends RecyclerView.ViewHolder{
    TextView nama, id, supervisor, manager, budget, areaSales,
provinsi, username, password;
    ImageView btnEdit, btnHapus, foto, btndetail;
    RelativeLayout btn;
    LinearLayout show, detail;
    public HolderData(@NonNull View itemView) {
        super(itemView);
        nama = itemView.findViewById(R.id.nama);
        foto = itemView.findViewById(R.id.foto);
        id = itemView.findViewById(R.id.id);
        supervisor = itemView.findViewById(R.id.supervisor);
        manager = itemView.findViewById(R.id.manager);
        budget = itemView.findViewById(R.id.budget);
        areaSales = itemView.findViewById(R.id.areasales);
        provinsi = itemView.findViewById(R.id.provinsi);
        username = itemView.findViewById(R.id.username);
        password = itemView.findViewById(R.id.password);
        btnEdit = itemView.findViewById(R.id.btncedit);
        btnHapus = itemView.findViewById(R.id.btnhapus);
        detail = itemView.findViewById(R.id.detail);
        btndetail = itemView.findViewById(R.id.btndetail);
        btn = itemView.findViewById(R.id.btn);
    }

}

@Override
public int getItemViewType(int position) {
    if (position == listData.size()) {
        return VIEW_TYPE_ITEM_LAST; // Tampilan terakhir
    }
    return VIEW_TYPE_ITEM; // Tampilan item biasa
}

public String getLastId(){

```

```

        if (listData != null && listData.size() > 0){
            int lastIndex = listData.size()-1;
            ModelSupervisor.supervisor lastData =
listData.get(lastIndex);
            return lastData.getKode_sv();
        }
        return null;
    }
}

```

SourceCode 1. 116 Data Supervisor

ee) Data Tampil Aktivitas

```

package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.ColorUtil;
import com.example.cakraagroindonesia.Model.ModelAktivitas;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

public class DataTampilAktivitas extends
RecyclerView.Adapter<DataTampilAktivitas.HolderData>{
    private Context context;
    private ArrayList<ModelAktivitas.aktivitas> listData;

    public DataTampilAktivitas(Context context,
ArrayList<ModelAktivitas.aktivitas> listData) {
        this.context = context;
        this.listData = listData;
    }

    @NonNull

```

```

    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_aktiv
itas,parent,false);
        return new HolderData(layout);
    }

    @Override
    public void onBindViewHolder(@NonNull HolderData holder, int
position) {
        ModelAktivitas.aktivitas model = listData.get(position);

        int randomColor = ColorUtil.getRandomColor();

        Glide.with(context).load(model.getFoto_url()).into(holder.fo
to);
        holder.kegiatan.setText(model.getNama_kegiatan());
        holder.lokasi.setText(model.getLokasi());
        holder.tanggalaktivitas.setText(model.getTanggal_kegiatan())
;
//        holder.bottom.setBackgroundColor(randomColor);
    }

    @Override
    public int getItemCount() {
        return listData.size();
    }

    public class HolderData extends RecyclerView.ViewHolder{
        TextView kegiatan,lokasi,tanggalaktivitas;
        LinearLayout bottom;
        ImageView foto;
        public HolderData(@NonNull View itemView) {
            super(itemView);

            kegiatan = itemView.findViewById(R.id.kegiatan);
            lokasi = itemView.findViewById(R.id.lokasi);
            tanggalaktivitas =
itemView.findViewById(R.id.tanggalaktivitas);
            bottom = itemView.findViewById(R.id.bottom);
            foto = itemView.findViewById(R.id.foto);
        }
    }
}

```

ff) Data Tampil Demplot

```

package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.ColorUtil;
import com.example.cakraagroindonesia.Model.ModelReport;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

public class DataTampilDemplot extends
RecyclerView.Adapter<DataTampilDemplot.HolderData>{
    private Context context;
    private ArrayList<ModelReport.report> listData;

    public DataTampilDemplot(Context context,
ArrayList<ModelReport.report> listData) {
        this.context = context;
        this.listData = listData;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_dempl
ot,parent,false);
        return new HolderData(layout);
    }

    @Override

```



```

    public void onBindViewHolder(@NonNull HolderData holder, int
position) {
        ModelReport.report model = listData.get(position);

        int randomColor = ColorUtil.getRandomColor();

        Glide.with(context).load(model.getFoto_url()).into(holder.fo
to);
        holder.namademonstrator.setText(model.getNama_demonstrator()
);
        holder.kabupaten.setText(model.getKabupaten_ds());
        holder.tanggaldemplot.setText(model.getTanggal_demplot());
        holder.bottom.setBackgroundColor(randomColor);
    }

    @Override
    public int getItemCount() {
        return listData.size();
    }

    public class HolderData extends RecyclerView.ViewHolder{
        TextView namademonstrator, kabupaten, tanggaldemplot;
        LinearLayout bottom;
        ImageView foto;
        public HolderData(@NonNull View itemView) {
            super(itemView);

            namademonstrator =
itemView.findViewById(R.id.namademonstrator);
            kabupaten = itemView.findViewById(R.id.kabupaten);
            tanggaldemplot =
itemView.findViewById(R.id.tanggaldemplot);
            bottom = itemView.findViewById(R.id.bottom);
            foto = itemView.findViewById(R.id.foto);
        }
    }
}

```

SourceCode 1. 118 Data Tampil Demplot

gg) Data Tampil Demplot Homepage

```

package com.example.cakraagroindonesia.Adapter;

import android.content.Context;
import android.view.LayoutInflater;

```

```

import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.example.cakraagroindonesia.ColorUtil;
import com.example.cakraagroindonesia.Model.ModelReport;
import com.example.cakraagroindonesia.R;

import java.util.ArrayList;

public class DataTampilDemplotHomePage extends
RecyclerView.Adapter<DataTampilDemplotHomePage.HolderData>{
    private Context context;
    private ArrayList<ModelReport.report> listData;
    private int limit = 5;

    public DataTampilDemplotHomePage(Context context,
ArrayList<ModelReport.report> listData) {
        this.context = context;
        this.listData = listData;
    }

    @NonNull
    @Override
    public HolderData onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_dempl
othomepage,parent,false);
        return new HolderData(layout);
    }

    @Override
    public void onBindViewHolder(@NonNull HolderData holder, int
position) {
        ModelReport.report model = listData.get(position);

        int randomColor = ColorUtil.getRandomColor();

        Glide.with(context).load(model.getFoto_url()).into(holder.fo
to);

```

```

        holder.namademonstrator.setText(model.getNama_demonstrator()
);
        holder.kabupaten.setText(model.getKabupaten_ds());
        holder.tanggaldemplot.setText(model.getTanggal_demplot());
        holder.bottom.setBackgroundColor(randomColor);
    }

    @Override
    public int getItemCount() {
        return Math.min(listData.size(), limit);
    }

    public class HolderData extends RecyclerView.ViewHolder{
        TextView namademonstrator, kabupaten, tanggaldemplot;
        LinearLayout bottom;
        ImageView foto;
        public HolderData(@NonNull View itemView) {
            super(itemView);

            namademonstrator =
itemView.findViewById(R.id.namademonstrator);
            kabupaten = itemView.findViewById(R.id.kabupaten);
            tanggaldemplot =
itemView.findViewById(R.id.tanggaldemplot);
            bottom = itemView.findViewById(R.id.bottom);
            foto = itemView.findViewById(R.id.foto);
        }
    }
}

```

SourceCode 1. 119 Data Tampil Demplot Homepage

7. API

```

package com.example.cakraagroindonesia.API;

import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

public class RetroServer {
    public final static String baseUrl =
"https://dbcaihost.000webhostapp.com/";
    private static Retrofit retro;

    public static Retrofit KonesiAPI(){
        if (retro == null){

```

```

        retro = new Retrofit.Builder()
            .baseUrl(baseUrl)
            .addConverterFactory(GsonConverterFactory.create())
            .build();
    }
    return retro;
}
}

```

SourceCode 1. 120 API Retro Server

8. Data Master

9. Interface

10. Model

11. Color Util

```

package com.example.cakraagroindonesia;

import android.graphics.Color;

import java.util.Random;

public class ColorUtil {

    private static final String[] COLORS = {
        "#F37C66", "#F9CB8F", "#18BB7C", "#4ABCD7", "#B0B0B0",
        "#5F5F5F"
    };

    public static int getRandomColor() {
        Random random = new Random();
        int randomIndex = random.nextInt(COLORS.length);
        String randomColorString = COLORS[randomIndex];
        return Color.parseColor(randomColorString);
    }
}

```

SourceCode 1. 121 Color Util

12. Hapus Data

```
package com.example.cakraagroindonesia;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;

import com.example.cakraagroindonesia.API.RetroServer;
import com.example.cakraagroindonesia.Interface.InterfaceAdmin;
import com.example.cakraagroindonesia.Interface.InterfaceAktivitas;
import com.example.cakraagroindonesia.Interface.InterfaceAlamat;
import com.example.cakraagroindonesia.Interface.InterfaceBerita;
import
com.example.cakraagroindonesia.Interface.InterfaceDemonstrator;
import
com.example.cakraagroindonesia.Interface.InterfaceDistributor;
import com.example.cakraagroindonesia.Interface.InterfaceFaq;
import com.example.cakraagroindonesia.Interface.InterfaceFungsida;
import com.example.cakraagroindonesia.Interface.InterfaceHerbisida;
import
com.example.cakraagroindonesia.Interface.InterfaceInsektisida;
import com.example.cakraagroindonesia.Interface.InterfaceManager;
import
com.example.cakraagroindonesia.Interface.InterfacePaketProduk;
import com.example.cakraagroindonesia.Interface.InterfacePgr;
import com.example.cakraagroindonesia.Interface.InterfaceProdukBaru;
import
com.example.cakraagroindonesia.Interface.InterfaceProdukHomepage;
import com.example.cakraagroindonesia.Interface.InterfaceQna;
import com.example.cakraagroindonesia.Interface.InterfaceReport;
import com.example.cakraagroindonesia.Interface.InterfaceSecretary;
import com.example.cakraagroindonesia.Interface.InterfaceSupervisor;
import com.example.cakraagroindonesia.Model.ModelAdmin;
import com.example.cakraagroindonesia.Model.ModelAktivitas;
import com.example.cakraagroindonesia.Model.ModelAlamat;
import com.example.cakraagroindonesia.Model.ModelBerita;
import com.example.cakraagroindonesia.Model.ModelDemonstrator;
import com.example.cakraagroindonesia.Model.ModelDistributor;
import com.example.cakraagroindonesia.Model.ModelFaq;
import com.example.cakraagroindonesia.Model.ModelFungsida;
import com.example.cakraagroindonesia.Model.ModelHerbisida;
import com.example.cakraagroindonesia.Model.ModelInsektisida;
import com.example.cakraagroindonesia.Model.ModelManager;
```

```

import com.example.cakraagroindonesia.Model.ModelPaketProduk;
import com.example.cakraagroindonesia.Model.ModelPgr;
import com.example.cakraagroindonesia.Model.ModelProdukBaru;
import com.example.cakraagroindonesia.Model.ModelProdukHomepage;
import com.example.cakraagroindonesia.Model.ModelQna;
import com.example.cakraagroindonesia.Model.ModelReport;
import com.example.cakraagroindonesia.Model.ModelSecretary;
import com.example.cakraagroindonesia.Model.ModelSupervisor;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class HapusData extends Activity {
    private int Id;
    private String stringId;
    private String Data;
    private TextView btnhapus;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hapus_data);
        btnhapus = findViewById(R.id.btnhapus);

        Id = getIntent().getIntExtra("idHapus", -1);
        stringId = getIntent().getStringExtra("IDhapus");
        Data = getIntent().getStringExtra("Data");

        if(Data.equals("Alamat")){
            Log.d("MyTag", "Alamat: "+Id+Data);
            btnhapus.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    InterfaceAlamat alamat =
RetroServer.KonesiAPI().create(InterfaceAlamat.class);
                    Call<ModelAlamat> hapusalamat =
alamat.deleteAlamat(Id);
                    hapusalamat.enqueue(new Callback<ModelAlamat>()
{
                        @Override
                        public void onResponse(Call<ModelAlamat>
call, Response<ModelAlamat> response) {
                            Toast.makeText(HapusData.this, "Data
Berhasil Dihapus", Toast.LENGTH_SHORT).show();
                            finish();
                        }
                    }
                }
            });
        }
    }
}

```



```

    }

    @Override
    public void onFailure(Call<ModelFungisida>
call, Throwable t) {
        Toast.makeText(HapusData.this, "Data
Gagal Dihapus", Toast.LENGTH_SHORT).show();
    }
});
}
});
}else if (Data.equals("PGR")){
    Log.d("MyTag", "PGR: "+Id+Data);
    btnhapus.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            InterfacePgr pgr =
RetroServer.KonesiAPI().create(InterfacePgr.class);
            Call<ModelPgr> hapusPgr = pgr.deletePgr(Id);
            hapusPgr.enqueue(new Callback<ModelPgr>() {
                @Override
                public void onResponse(Call<ModelPgr> call,
Response<ModelPgr> response) {
                    Toast.makeText(HapusData.this, "Data
Berhasil Dihapus", Toast.LENGTH_SHORT).show();
                    finish();
                }

                @Override
                public void onFailure(Call<ModelPgr> call,
Throwable t) {
                    Toast.makeText(HapusData.this, "Data
Gagal Dihapus", Toast.LENGTH_SHORT).show();
                }
            });
        }
    });
}else if (Data.equals("FAQ")){
    Log.d("MyTag", "FAQ: "+Id+Data);
    btnhapus.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            InterfaceFaq faq =
RetroServer.KonesiAPI().create(InterfaceFaq.class);
            Call<ModelFaq> hapusFaq = faq.deleteFaq(Id);
            hapusFaq.enqueue(new Callback<ModelFaq>() {
                @Override

```



```

    } else if (Data.equals("Demonstrator")){
        Log.d("MyTag", "Demonstrator: "+stringId+Data);
        btnhapus.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                InterfaceDemonstrator demonstrator =
RetroServer.KonesiAPI().create(InterfaceDemonstrator.class);
                Call<ModelDemonstrator> hapus =
demonstrator.deleteDemonstrator(stringId);
                hapus.enqueue(new Callback<ModelDemonstrator>()
{
                    @Override
                    public void
onResponse(Call<ModelDemonstrator> call, Response<ModelDemonstrator>
response) {
                        Toast.makeText(HapusData.this, "Data
Berhasil Dihapus", Toast.LENGTH_SHORT).show();
                        finish();
                    }
                    @Override
                    public void
onFailure(Call<ModelDemonstrator> call, Throwable t) {
                        Toast.makeText(HapusData.this, "Data
Gagal Dihapus", Toast.LENGTH_SHORT).show();
                    }
                });
            }
        });
    } else if (Data.equals("Supervisor")){
        Log.d("MyTag", "Supervisor: "+stringId+Data);
        btnhapus.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                InterfaceSupervisor supervisor =
RetroServer.KonesiAPI().create(InterfaceSupervisor.class);
                Call<ModelSupervisor> hapus =
supervisor.deleteSupervisor(stringId);
                hapus.enqueue(new Callback<ModelSupervisor>() {
                    @Override
                    public void onResponse(Call<ModelSupervisor>
call, Response<ModelSupervisor> response) {
                        Toast.makeText(HapusData.this, "Data
Berhasil Dihapus", Toast.LENGTH_SHORT).show();
                        finish();
                    }
                    @Override

```

```

        public void onFailure(Call<ModelSupervisor>
call, Throwable t) {
            Toast.makeText(HapusData.this, "Data
Gagal Dihapus", Toast.LENGTH_SHORT).show();
        }
    });
}
});
} else if (Data.equals("Secretary")){
    Log.d("MyTag", "Secretary: "+stringId+Data);
    btnhapus.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            InterfaceSecretary secretary =
RetroServer.KonesiAPI().create(InterfaceSecretary.class);
            Call<ModelSecretary> hapus =
secretary.deleteSecretary(stringId);
            hapus.enqueue(new Callback<ModelSecretary>() {
                @Override
                public void onResponse(Call<ModelSecretary>
call, Response<ModelSecretary> response) {
                    Toast.makeText(HapusData.this, "Data
Berhasil Dihapus", Toast.LENGTH_SHORT).show();
                    finish();
                }
                @Override
                public void onFailure(Call<ModelSecretary>
call, Throwable t) {
                    Toast.makeText(HapusData.this, "Data
Gagal Dihapus", Toast.LENGTH_SHORT).show();
                }
            });
        }
    });
} else if (Data.equals("Admin")){
    Log.d("MyTag", "Admin: "+stringId+Data);
    btnhapus.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            InterfaceAdmin admin =
RetroServer.KonesiAPI().create(InterfaceAdmin.class);
            Call<ModelAdmin> hapus =
admin.deleteAdmin(stringId);
            hapus.enqueue(new Callback<ModelAdmin>() {
                @Override
                public void onResponse(Call<ModelAdmin>
call, Response<ModelAdmin> response) {

```

```

                Toast.makeText(HapusData.this, "Data
Berhasil Dihapus", Toast.LENGTH_SHORT).show();
                finish();
            }
            @Override
            public void onFailure(Call<ModelAdmin> call,
Throwable t) {
                Toast.makeText(HapusData.this, "Data
Gagal Dihapus", Toast.LENGTH_SHORT).show();
            }
        });
    }
});
}
});
}
else if (Data.equals("Berita")){
    Log.d("MyTag", "Berita: "+Id+Data);
    btnhapus.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            InterfaceBerita berita =
RetroServer.KonesiAPI().create(InterfaceBerita.class);
            Call<ModelBerita> hapus =
berita.deleteBerita(Id);
            hapus.enqueue(new Callback<ModelBerita>() {
                @Override
                public void onResponse(Call<ModelBerita>
call, Response<ModelBerita> response) {
                    Toast.makeText(HapusData.this, "Data
Berhasil Dihapus", Toast.LENGTH_SHORT).show();
                    finish();
                }
            }
            @Override
            public void onFailure(Call<ModelBerita>
call, Throwable t) {
                Toast.makeText(HapusData.this, "Data
Gagal Dihapus", Toast.LENGTH_SHORT).show();
            }
        });
    }
});
}
else if (Data.equals("Aktivitas")){
    Log.d("MyTag", ": Aktivitas"+stringId+Data);
    btnhapus.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            InterfaceAktivitas aktivitas =
RetroServer.KonesiAPI().create(InterfaceAktivitas.class);

```

```

        Call<ModelAktivitas> hapus =
    aktivitas.deleteAktivitas(stringId);
        hapus.enqueue(new Callback<ModelAktivitas>() {
            @Override
            public void onResponse(Call<ModelAktivitas>
call, Response<ModelAktivitas> response) {
                Toast.makeText(HapusData.this, "Data
Berhasil Dihapus", Toast.LENGTH_SHORT).show();
                finish();
            }
            @Override
            public void onFailure(Call<ModelAktivitas>
call, Throwable t) {
                Toast.makeText(HapusData.this, "Data
Gagal Dihapus", Toast.LENGTH_SHORT).show();
            }
        });
    });
    }else if (Data.equals("Report")){
        Log.d("MyTag", ": Report"+stringId+Data);
        btnhapus.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                InterfaceReport report =
RetroServer.KonesiAPI().create(InterfaceReport.class);
                Call<ModelReport> hapus =
report.deleteReport(stringId);
                hapus.enqueue(new Callback<ModelReport>() {
                    @Override
                    public void onResponse(Call<ModelReport>
call, Response<ModelReport> response) {
                        Toast.makeText(HapusData.this, "Data
Berhasil Dihapus", Toast.LENGTH_SHORT).show();
                        finish();
                    }
                    @Override
                    public void onFailure(Call<ModelReport>
call, Throwable t) {
                        Toast.makeText(HapusData.this, "Data
Gagal Dihapus", Toast.LENGTH_SHORT).show();
                    }
                });
            }
        });
    }else if (Data.equals("ProdukHomepage")){
        Log.d("MyTag", "ProdukHomepage: "+Id+Data);

```



```

        btnhapus.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                InterfaceProdukHomepage produkHomepage =
RetroServer.KonesiAPI().create(InterfaceProdukHomepage.class);
                Call<ModelProdukHomepage> hapus =
produkHomepage.deleteProdukHomepage(Id);
                hapus.enqueue(new
Callback<ModelProdukHomepage>() {
                    @Override
                    public void
onResponse(Call<ModelProdukHomepage> call,
Response<ModelProdukHomepage> response) {
                        Toast.makeText(HapusData.this, "Data
Berhasil Dihapus", Toast.LENGTH_SHORT).show();
                        finish();
                    }
                    @Override
                    public void
onFailure(Call<ModelProdukHomepage> call, Throwable t) {
                        Toast.makeText(HapusData.this, "Data
Gagal Dihapus", Toast.LENGTH_SHORT).show();
                    }
                });
            }
        });
    }else {
        Toast.makeText(this, "GAGAL HAPUS DATA",
Toast.LENGTH_SHORT).show();
        finish();
    }
}
}
}

```

SourceCode 1. 122 Hapus Data

13. Login

```

package com.example.cakraagroindonesia;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

```

```

import androidx.appcompat.app.AppCompatActivity;

import com.example.cakraagroindonesia.Activity.Panel;
import com.example.cakraagroindonesia.ActivityManager.PanelManager;
import
com.example.cakraagroindonesia.ActivitySecretary.PanelSecretary;

public class Login extends AppCompatActivity {

    private String admin = "admin";
    private String manager = "manager";
    private String secretary = "secretary";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        TextView btnLogin = findViewById(R.id.btnmasuk);
        EditText username = findViewById(R.id.username);
        EditText password = findViewById(R.id.password);

        btnLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String Username =
username.getText().toString().trim();
                String Password =
password.getText().toString().trim();

                if (Username.isEmpty() || Password.isEmpty()) {

                } else if (Username.equals(admin) &&
Password.equals(admin)) {

                    startActivity(new Intent(Login.this,
Panel.class));

                    finish();

                } else if (Username.equals(manager) &&
Password.equals(manager)) {

                    startActivity(new Intent(Login.this,
PanelManager.class));

                    finish();
                }
            }
        });
    }
}

```

```

        }else if (Username.equals(secretary) &&
Password.equals(secretary)) {

            startActivity(new Intent(Login.this,
PanelSecretary.class));
            finish();

        }else {
            // Login gagal
            Toast.makeText(Login.this, "BLOK",
Toast.LENGTH_SHORT).show();
        }
    }
});
}
}

```

SourceCode 1. 123 Login

14. Main Activity

```

package com.example.cakraagroindonesia;

import android.os.Build;
import android.os.Bundle;
import android.view.MenuItem;
import android.view.Window;
import android.view.WindowManager;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;

import
com.example.cakraagroindonesia.ActivityCustomer.Fragment.FragmentAkt
ivitas;
import
com.example.cakraagroindonesia.ActivityCustomer.Fragment.FragmentHom
e;
import
com.example.cakraagroindonesia.ActivityCustomer.Fragment.FragmentOth
er;
import
com.example.cakraagroindonesia.ActivityCustomer.Fragment.FragmentPro
duk;

```

```

import
com.google.android.material.bottomnavigation.BottomNavigationView;

public class MainActivity extends AppCompatActivity {
    BottomNavigationView bottomNavigationView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        switchFragment(new FragmentHome());

        bottomNavigationView = findViewById(R.id.navigation);
        bottomNavigationView.setOnNavigationItemSelectedListener(new
BottomNavigationView.OnNavigationItemSelectedListener() {
            @Override
            public boolean onNavigationItemSelected(@NonNull
MenuItem item) {
                switch (item.getItemId()) {
                    case R.id.nav_home:
                        switchFragment(new FragmentHome());
                        return true;
                    case R.id.nav_produk:
                        switchFragment(new FragmentProduk());
                        return true;
                    case R.id.nav_aktivitas:
                        switchFragment(new FragmentAktivitas());
                        return true;
                    case R.id.nav_other:
                        switchFragment(new FragmentOther());
                        return true;
                }
                return false;
            }
        });

        if (Build.VERSION.SDK_INT >= 21) {
            Window window = this.getWindow();
            window.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SY
STEM_BAR_BACKGROUNDS);
            window.clearFlags(WindowManager.LayoutParams.FLAG_TRANSL
UCENT_STATUS);
            window.setStatusBarColor(this.getResources().getColor(R.
color.white));
        }
    }
}

```

```

    }

    private void switchFragment(Fragment fragment) {
        FragmentManager fragmentManager =
getSupportFragmentManager();
        FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();
        fragmentTransaction.replace(R.id.container_fragment,
fragment);
        fragmentTransaction.commit();
    }
}

```

SourceCode 1. 124 Main Activity

15. Splash Screen

```

package com.example.cakraagroindonesia;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;

import androidx.appcompat.app.AppCompatActivity;

public class SplashScreen extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash_screen);

        View decorView = getWindow().getDecorView();
        // Hide App bar
        if (getSupportActionBar() != null) {
            getSupportActionBar().hide();
        }

        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                startActivity(new Intent(SplashScreen.this,
MainActivity.class));
                finish();
            }
        });
    }
}

```

```
    }, 1000);  
  }  
}
```

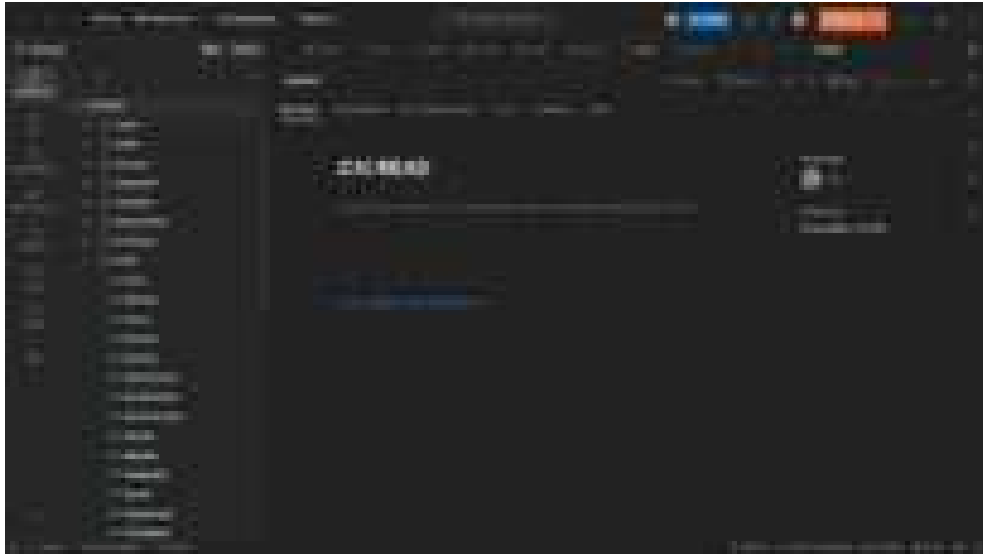
SourceCode 1. 125 Splash Screen

viii. Testing

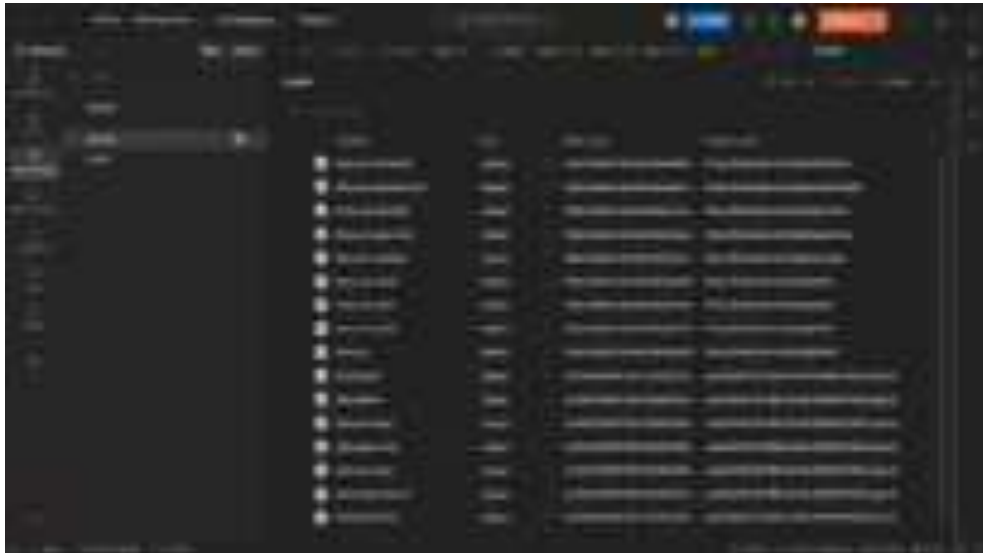
Laporan Pengujian RestAPI Menggunakan Postman

Pengujian dilakukan pada postman dengan membuat *collection* yang berisi CRUD dari 7 jenis user yang berbeda yaitu Superadmin, admin, manager, supervisor, secretary, demonstrator, dan distributor

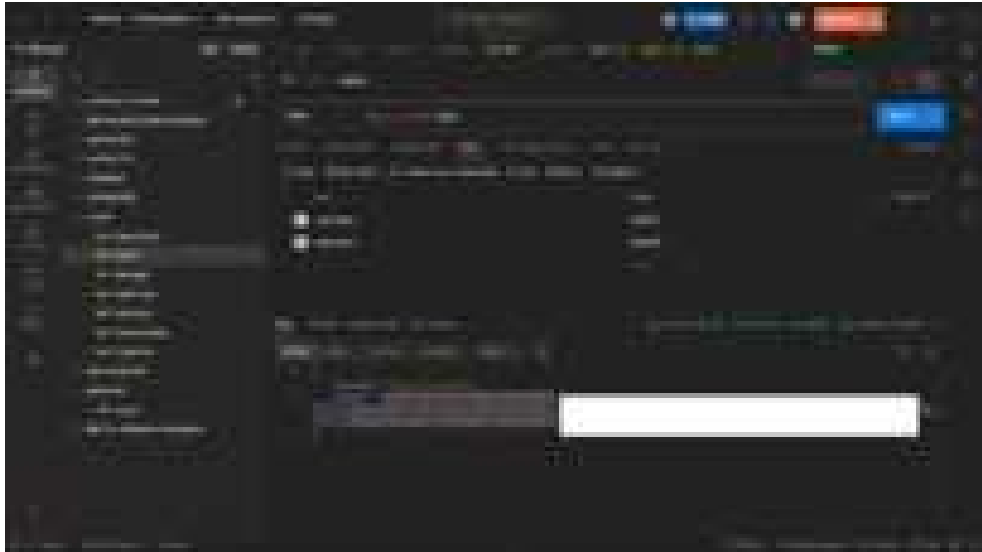
Dan juga akses yang bersifat open/public dengan konfigurasi collection sebagai berikut :



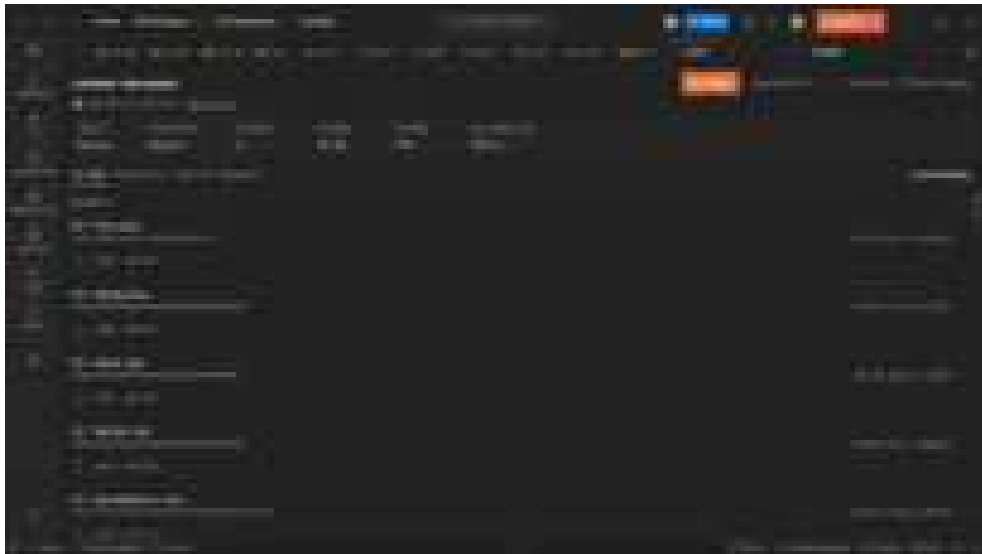
Dari collection tersebut dibuat *environment* atau lingkungan pengujian yaitu sebagai berikut:



environment tersebut berisikan `base_url` dari masing-masing jenis atau level user, dan juga token yang digenerate dari login user tersebut yang diambil dari collection dibawah ini



berikutnya dilakukan `run test` dengan hasil seperti berikut:



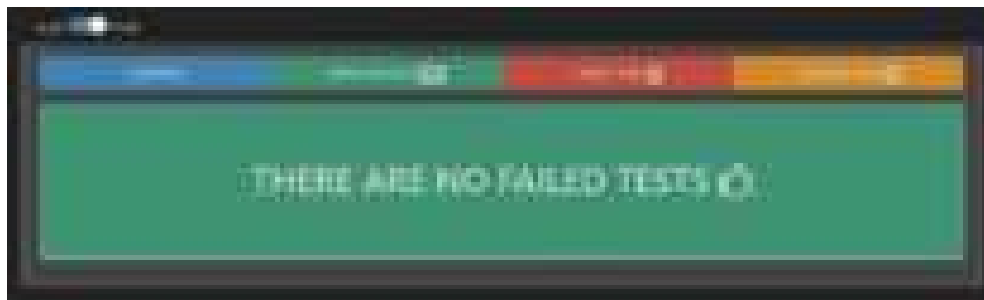
berikutnya dari pengujian tersebut saya ekspor menjadi JSON dan berikutnya saya gunakan newman untuk membuat dokumentasi pengujian dengan hasil seperti dibawah ini



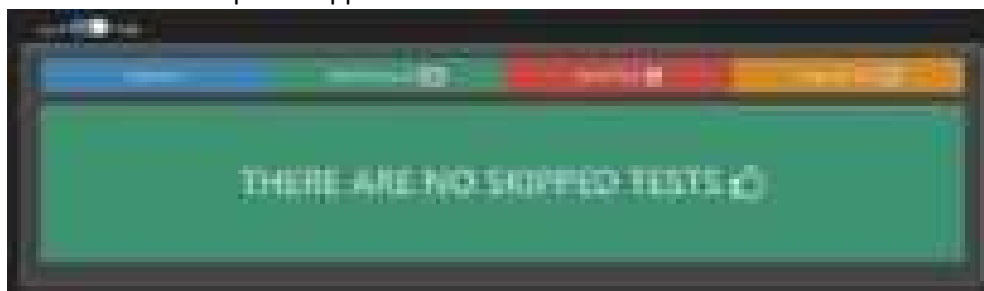
Berikut detail total request yang diba



Berikut adalah detail failed tests



Berikut adalah tampilan skipped test



**Skenario Pengujian “Pembuatan Aplikasi Komersial PT. Cakra Agro
Indonesia Berbasis Mobile”**

a. Super Admin

Test Scenario	Test Case	Test Step	Result	Status
Halaman utama	Super Admin membuka aplikasi mobile	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman utama aplikasi mobile 2. Super Admin memeriksa tampilan foto slide produk dan fungsinya 3. Super Admin memeriksa tampilan penjelasan singkat produk 4. Super Admin memeriksa tampilan demplot terbaru 5. Super Admin memeriksa tampilan berita terbaru 	Super Admin dapat melihat foto slide produk dan fungsinya, penjelasan singkat produk, demplot terbaru, dan berita terbaru	Sesuai
Halaman produk kami	Super Admin membuka halaman produk kami	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman produk kami 2. Super Admin memeriksa tampilan produk kami 	Super Admin dapat melihat tampilan produk kami	Sesuai
	Super Admin masuk pada halaman Insektisida	<ol style="list-style-type: none"> 1. Super Admin menekan Insektisida 2. Super Admin berada di halaman Insektisida 3. Super Admin memeriksa produk Insektisida 4. Super Admin melakukan search untuk menemukan produk Insektisida 	Super Admin dapat melihat keseluruhan produk insektisida beserta deskripsinya dan dapat melakukan search untuk mencari produk insektisida	Sesuai
	Super Admin masuk pada halaman Fungisida	<ol style="list-style-type: none"> 1. Super Admin menekan button Fungisida 	Super Admin dapat melihat keseluruhan produk	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		<ol style="list-style-type: none"> 2. Super Admin berada di halaman Fungisida 3. Super Admin memeriksa produk Fungisida 4. Super Admin melakukan search untuk menemukan produk Fungisida 	<p>fungisida beserta deskripsinya dan dapat melakukan search untuk mencari produk Fungisida</p>	
	Super Admin masuk pada halaman Herbisida	<ol style="list-style-type: none"> 1. Super Admin menekan button Herbisida 2. Super Admin berada di halaman Herbisida 3. Super Admin memeriksa produk Herbisida 4. Super Admin melakukan search untuk menemukan produk Herbisida 	<p>Super Admin dapat melihat keseluruhan produk Herbisida beserta deskripsinya dan dapat melakukan search untuk mencari produk Herbisida</p>	Sesuai
	Super Admin masuk pada halaman PGR	<ol style="list-style-type: none"> 1. Super Admin menekan button PGR 2. Super Admin berada di halaman PGR 3. Super Admin memeriksa produk PGR 4. Super Admin melakukan search untuk menemukan produk PGR 	<p>Super Admin dapat melihat keseluruhan produk PGR beserta deskripsinya dan dapat melakukan search untuk mencari produk PGR</p>	Sesuai
Halaman Event, Halaman Inovasi, dan FAQ	Super Admin masuk pada halaman Event	<ol style="list-style-type: none"> 1. Super Admin menekan image button aktivitas 2. Super Admin masuk pada halaman aktivitas 3. Super Admin menekan image button demplot 4. Super Admin masuk pada halaman aktivitas 	<p>Super Admin dapat melihat tampilan aktivitas dan tampilan pada demplot</p>	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
	Super Admin masuk pada halaman Inovasi	<ol style="list-style-type: none"> 1. Super Admin menekan image button produk baru 2. Super Admin masuk pada halaman produk baru 3. Super Admin menekan image button paket produk 4. Super Admin masuk pada halaman paket produk 5. Super Admin klik Lihat PPT untuk mengunduh paket produk 	Super Admin dapat melihat tampilan produk baru, Super Admin dapat melihat tampilan pada paket produk, dan user dapat melihat isi paket produk dengan mengunduh PPT	Sesuai
	Super Admin masuk pada halaman FAQ	<ol style="list-style-type: none"> 1. Super Admin menekan button FAQ 2. Super Admin menekan pertanyaan dan sudah tersedia jawabannya pada FAQ 	Super Admin dapat melihat pertanyaan yang sudah tersedia dan terdapat jawabannya pada FAQ	Sesuai
	Super Admin masuk pada halaman Pertanyaan Lain (QnA)	<ol style="list-style-type: none"> 1. Super Admin mereview pertanyaan-pertanyaan yang diajukan oleh customer lain 	Super Admin dapat melihat review-an yang sudah diajukan oleh customer lain	Sesuai
	Super Admin menambahkan pertanyaan	<ol style="list-style-type: none"> 1. Super Admin klik Tambahkan Pertanyaan untuk mengajukan pertanyaan 2. Super Admin menambahkan pertanyaan dengan mengisi semua field pada kolom pertanyaan dan disertai upload foto 3. Super Admin menambahkan pertanyaan dengan mengisi semua field pada kolom 	Super Admin dapat mengajukan pertanyaan disertai upload foto ataupun tidak disertai upload foto	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		pertanyaan dengan tidak disertai upload foto		
Halaman Tentang Kami	Super Admin masuk pada halaman Tentang kami	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman Tentang Kami 2. Super Admin memeriksa tampilan profil, Visi dan Misi, serta Alamat 3. Super Admin klik image button pada sosial media untuk facebook 4. Super Admin klik image button pada sosial media untuk Instagram 5. Super Admin klik image button pada sosial media untuk Instagram 6. Super Admin klik image button pada sosial media untuk youtube 	Super Admin dapat melihat tampilan profil, visi dan misi, Alamat, dan pada sosial media langsung bisa terhubung pada facebook, Instagram, dan youtube	Sesuai
Super Admin Login dari halaman Login	Super Admin Login dengan mengisi data yang valid di halaman Login	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman login aplikasi komersial PT. Cakra Agro Indonesia 2. Super Admin memasukan username yang valid pada field username 3. Super Admin memasukan password yang valid pada field password 4. Super Admin klik tombol login 	Super Admin masuk ke halaman panel atau Homepage SuperAdmin	Sesuai
	Super Admin Login dengan mengisi data yang tidak valid di halaman Login	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman login aplikasi komersial PT. Cakra Agro Indonesia 2. Super Admin memasukan username yang tidak valid pada field username 	Super Admin tidak dapat login karena data tidak valid dan muncul notifikasi "username	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		<ol style="list-style-type: none"> 3. Super Admin memasukkan password yang tidak valid pada field password 4. Super Admin klik tombol login 	atau password salah”	
	Super Admin Login dengan mengosongkan field mandatory di halaman login	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman login aplikasi komersial PT. Cakra Agro Indonesia 2. Super Admin mengosongkan field username dan password 3. Super Admin klik tombol login 	Super Admin tidak dapat login dan terdapat notifikasi “masukan username dan password”	Sesuai
Halaman Dashboard	Super Admin masuk pada halaman Dashboard	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman Dashboard 2. Super Admin memeriksa tampilan info data Super Admin username dan password 	Super Admin dapat melihat tampilan info data Super Admin	Sesuai
	Super Admin Update data pada halaman Dashboard	<ol style="list-style-type: none"> 1. Super Admin melakukan update info Super Admin untuk username dan password 2. Super Admin klik image button update untuk mengubah info 3. Super Admin mengubah field yang akan diupdate 4. Super Admin klik buton submit 	Super Admin berhasil melakukan update	Sesuai
Logout Aplikasi	Super Admin logout aplikasi pada halaman dashboard	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman Dashboard Super Admin 2. Super Admin melakukan logout dengan klik button Logout 	Super Admin berhasil melakukan logout	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
Halaman Lihat LOG	Super Admin masuk pada halaman LOG dan masuk pada halaman LOG Super Admin	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman LOG 2. Super Admin klik button LOG Super Admin 3. Super Admin memeriksa riwayat aktivitas Super Admin 	Super Admin dapat melihat tampilan riwayat aktivitas yang telah dilakukan oleh Super Admin	Sesuai
	Super Admin masuk pada halaman LOG Admin	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman LOG Admin 2. Super Admin klik button LOG Admin 3. Super Admin memeriksa riwayat aktivitas Admin 	Super Admin dapat melihat tampilan riwayat aktivitas yang telah dilakukan oleh Admin	Sesuai
	Super Admin masuk pada halaman LOG Master	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman LOG Master 2. Super Admin klik button LOG Master 3. Super Admin memeriksa riwayat aktivitas Master 	Super Admin dapat melihat tampilan riwayat aktivitas yang telah dilakukan oleh Master	Sesuai
Halaman Kelola Data Master	Super Admin masuk pada halaman Data Admin	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman Data Admin 2. Super Admin membuat akun baru untuk admin dengan klik button tambahkan 3. Super Admin mengisi semua field untuk menambahkan akun baru 4. Super Admin klik button submit 	Super Admin berhasil menambahkan akun baru	Sesuai
	Super Admin tambah data pada halaman Data Admin	<ol style="list-style-type: none"> 1. Super Admin menambahkan Data Admin 2. Super Admin klik button Tambah untuk menambahkan data 3. Super Admin mengisi semua field yang ada 	Super Admin berhasil menambahkan Data Admin	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		<p>pada form tambah data</p> <p>4. Super Admin klik submit</p>		
	Super Admin Update data pada halaman Data Admin	<p>1. Super Admin melakukan update data admin</p> <p>2. Super Admin klik image button update untuk mengubah data</p>	Super Admin berhasil melakukan update data	Sesuai
	Super Admin hapus data pada halaman Data Admin	<p>1. Super Admin melakukan hapus akun</p> <p>2. Super Admin klik image button hapus dan muncul notifikasi “apakah anda yakin ingin menghapus? “, lalu klik hapus</p>	Super Admin berhasil melakukan hapus akun	Sesuai
	Super Admin masuk pada halaman Data Secretary dengan membuat akun disertai foto	<p>1. Super Admin berada pada halaman Data Secretary</p> <p>2. Super Admin membuat akun baru untuk secretary dengan klik button tambahkan</p> <p>3. Super Admin mengisi semua field untuk menambahkan akun baru</p> <p>4. Super Admin melakukan upload foto untuk identitas sekretaris</p> <p>5. Super Admin klik button submit dan muncul notifikasi “apakah anda yakin ingin menghapus? “, lalu klik hapus</p>	Super Admin berhasil menambahkan akun baru	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
	Super Admin masuk pada halaman Data Secretary dengan membuat akun yang tidak disertai foto	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman Data Secretary 2. Super Admin membuat akun baru untuk secretary dengan klik button tambahkan 3. Super Admin mengisi semua field data secretary untuk menambahkan akun baru 4. Super Admin tidak melakukan upload foto 5. Super Admin klik button submit 	Super Admin tidak berhasil menambahkan akun baru karena tidak disertai upload foto dan muncul notifikasi “pilih foto terlebih dahulu”	Sesuai
	Super Admin Update data pada halaman Data Secretary	<ol style="list-style-type: none"> 1. Super Admin melakukan update data secretary 2. Super Admin klik image button update untuk mengubah data 3. Super Admin mengganti field data secretary yang akan diubah 4. Super Admin klik Submit 	Super Admin berhasil melakukan update data	Sesuai
	Super Admin hapus data pada halaman Data Secretary	<ol style="list-style-type: none"> 1. Super Admin melakukan hapus akun 2. Super Admin klik image button hapus dan muncul notifikasi “apakah anda yakin ingin menghapus? “, lalu klik hapus 	Super Admin berhasil melakukan hapus akun	Sesuai
	Super Admin masuk pada halaman Data Supervisor dengan membuat akun yang disertai foto	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman Data Supervisor 2. Super Admin membuat akun baru untuk supervisor dengan klik button tambahkan 	Super Admin berhasil menambahkan akun baru	Sesuai s

Test Scenario	Test Case	Test Step	Result	Status
		<ol style="list-style-type: none"> 3. Super Admin mengisi semua field data supervisor untuk menambahkan akun baru 4. Super Admin tidak melakukan upload foto 5. Super Admin klik button submit 		
	Super Admin masuk pada halaman Data Supervisor dengan membuat akun yang tidak disertai foto	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman Data Supervisor 2. Super Admin membuat akun baru untuk supervisor dengan klik button tambahkan 3. Super Admin mengisi semua field data supervisor untuk menambahkan akun baru 4. Super Admin tidak melakukan upload foto 5. Super Admin klik button submit 	Tidak berhasil menambahkan akun baru karena tidak disertai upload foto dan muncul notifikasi “pilih foto terlebih dahulu”	Sesuai
	Super Admin Update data pada halaman Data Supervisor	<ol style="list-style-type: none"> 1. Super Admin melakukan update data supervisor 2. Super Admin klik image button update untuk mengubah data 3. Super Admin mengganti field data Supervisor yang akan diubah 4. Super Admin klik Submit 	Super Admin berhasil melakukan update data	Sesuai
	Super Admin hapus data pada halaman Data Supervisor	<ol style="list-style-type: none"> 1. Super Admin melakukan hapus akun 2. Super Admin klik image button hapus dan muncul notifikasi “apakah anda yakin 	Super Admin berhasil melakukan hapus akun	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		ingin menghapus? “, lalu klik hapus		
	Super Admin masuk pada halaman Data Demonstrator dengan membuat akun yang disertai foto	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman Data Supervisor 2. Super Admin membuat akun baru untuk Demonstrator dengan klik button tambahkan 3. Super Admin mengisi semua field data Demonstrator untuk menambahkan akun baru 4. Super Admin melakukan upload foto 5. Super Admin klik button submit 	Super Admin berhasil menambahkan akun baru	Sesuai
	Super Admin masuk pada halaman Data Demonstrator dengan membuat akun yang tidak disertai foto	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman Data Demonstrator 2. Super Admin membuat akun baru untuk Demonstrator dengan klik button tambahkan 3. Super Admin mengisi semua field data Demonstrator untuk menambahkan akun baru 4. Super Admin tidak melakukan upload foto 5. Super Admin klik button submit 	Tidak berhasil menambahkan akun baru karena tidak disertai upload foto dan muncul notifikasi “pilih foto terlebih dahulu”	Sesuai
	Super Admin Update data pada halaman Data Demonstrator	<ol style="list-style-type: none"> 1. Super Admin melakukan update data Demonstrator 	Super Admin berhasil melakukan update data	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		<ol style="list-style-type: none"> 2. Super Admin klik image button update untuk mengubah data 3. Super Admin mengganti field data Demonstrator yang akan diubah 4. Super Admin klik Submit 		
	Super Admin hapus data pada halaman Data Demonstrator	<ol style="list-style-type: none"> 1. Super Admin melakukan hapus akun 2. Super Admin klik image button hapus dan muncul notifikasi “apakah anda yakin ingin menghapus? “, lalu klik hapus 	Super Admin berhasil melakukan hapus akun	Sesuai s
	Super Admin masuk pada halaman Data Distributor dengan membuat akun yang disertai foto	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman Data Distributor 2. Super Admin membuat akun baru untuk Distributor dengan klik button tambahkan 3. Super Admin mengisi semua field data Distributor untuk menambahkan akun baru 4. Super Admin melakukan upload foto 5. Super Admin klik button submit 	Super Admin berhasil menambahkan akun baru	Sesuai
	Super Admin masuk pada halaman Data Distributor dengan membuat akun yang tidak disertai foto	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman Data Distributor 2. Super Admin membuat akun baru untuk Distributor dengan klik button tambahkan 3. Super Admin mengisi semua field data Distributor untuk 	Tidak berhasil menambahkan akun baru karena tidak disertai upload foto dan muncul notifikasi “pilih foto terlebih dahulu”	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		menambahkan akun baru 4. Super Admin tidak melakukan upload foto 5. Super Admin klik button submit		
	Super Admin Update data pada halaman Data Distributor	1. Super Admin melakukan update data Distributor 2. Super Admin klik image button update untuk mengubah data 3. Super Admin mengganti field data Distributor yang akan diubah 4. Super Admin klik Submit	Super Admin berhasil melakukan update data	Sesuai
	Super Admin hapus data pada halaman Data Distributor	1. Super Admin melakukan hapus akun 2. Super Admin klik image button hapus dan muncul notifikasi “apakah anda yakin ingin menghapus? “, lalu klik hapus	Super Admin berhasil menambahkan akun baru	Sesuai
	Super Admin masuk pada halaman Data Manager dengan membuat akun yang disertai foto	1. Super Admin berada pada halaman Data Manager 2. Super Admin membuat akun baru untuk Manager dengan klik button tambahkan 3. Super Admin mengisi semua field data Manager untuk menambahkan akun baru 4. Super Admin melakukan upload foto 5. Super Admin klik button submit	Super Admin berhasil menambahkan akun baru	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
	Super Admin masuk pada halaman Data Manager dengan membuat akun yang tidak disertai foto	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman Data Manager 2. Super Admin membuat akun baru untuk Manager dengan klik button tambahkan 3. Super Admin mengisi semua field data Manager untuk menambahkan akun baru 4. Super Admin tidak melakukan upload foto 5. Super Admin klik button submit 	Tidak berhasil menambahkan akun baru karena tidak disertai upload foto dan muncul notifikasi “pilih foto terlebih dahulu”	Sesuai
	Super Admin Update data pada halaman Data Manager	<ol style="list-style-type: none"> 1. Super Admin melakukan update data Manager 2. Super Admin klik image button update untuk mengubah data 3. Super Admin mengganti field data Manager yang akan diubah 4. Super Admin klik Submit 	Super Admin berhasil melakukan update data	Sesuai
	Super Admin hapus data pada halaman Data Manager	<ol style="list-style-type: none"> 1. Super Admin melakukan hapus akun 2. Super Admin klik image button hapus dan muncul notifikasi “apakah anda yakin ingin menghapus? “, lalu klik hapus 	Super Admin berhasil menambahkan akun baru	Sesuai
Halaman Kelola Report Demonstrator	Super Admin masuk pada halaman Kelola Report Demonstrator	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman Kelola Report Demonstrator 2. Super Admin memeriksa tampilan pada Daftar Report 	Super Admin berhasil masuk dan dapat melihat Daftar Report	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
	Super Admin masuk pada halaman Daftar Report Demonstrator dengan menambahkan produk yang disertai foto	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman Daftar Report Demonstrator 2. Super Admin menambahkan produk dengan klik button Tambahkan 3. Super Admin mengisi semua field data produk 4. Super Admin melakukan upload foto 5. Super Admin klik button submit 	Super Admin berhasil menambahkan Daftar Report Baru	Sesuai
	Super Admin masuk pada halaman Daftar Report Demonstrator dengan menambahkan produk yang tidak disertai foto	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman Daftar Report Demonstrator 2. Super Admin menambahkan produk dengan klik button Tambahkan 3. Super Admin mengisi semua field data produk 4. Super Admin tidak melakukan upload foto 5. Super Admin klik button submit 	Tidak berhasil menambahkan akun baru karena tidak disertai upload foto dan muncul notifikasi “pilih foto terlebih dahulu”	Sesuai
	Super Admin Update data pada halaman Daftar Report Demonstrator	<ol style="list-style-type: none"> 1. Super Admin melakukan update Daftar Report Demonstrator 2. Super Admin klik image button update untuk mengubah data 3. Super Admin mengganti field pada data Daftar Report Demonstrator yang akan diubah 4. Super Admin memilih opsi “lama” untuk menyatakan bahwa 	Super Admin berhasil melakukan update data dengan status berwarna hijau “disetujui” dan status berwarna merah “belum disetujui”	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		<p>laporan demonstrator tersebut sudah setuju</p> <p>5. Super Admin memilih opsi “baru” untuk menyatakan bahwa laporan demonstrator tersebut belum setuju</p> <p>6. Super Admin klik Submit</p>		
	Super Admin hapus data pada halaman Daftar Report Demonstrator	<p>1. Super Admin melakukan hapus akun</p> <p>2. Super Admin klik image button hapus dan muncul notifikasi “apakah anda yakin ingin menghapus? “, lalu klik hapus</p>	Super Admin berhasil menambahkan akun baru	Sesuai
Halaman Kelola Beranda	Super Admin masuk pada halaman Kelola Beranda	<p>1. Super Admin berada dalam halaman Kelola Beranda</p> <p>2. Super Admin memeriksa tampilan Kelola Beranda</p>	Super Admin berhasil masuk ke halaman Kelola Beranda dan dapat melihat tampilannya	Sesuai
	Super Admin masuk pada halaman Homepage di Kelola Beranda	<p>1. Super Admin berada dalam halaman Homepage</p> <p>2. Super Admin mengubah judul dan deskripsi pada Homepage</p> <p>3. Super Admin melakukan upload foto tentang perusahaan</p>	Super Admin berhasil mengubah judul dan deskripsi, serta dapat mengupload foto tentang perusahaan	Sesuai
	Super Admin masuk pada halaman Produk Homepage di Kelola Beranda	<p>1. Super Admin berada dalam halaman Produk Homepage</p> <p>2. Super Admin memeriksa tampilan Produk Homepage</p>	Super Admin berhasil masuk ke halaman Produk Homepage dan dapat melihat tampilannya	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
	Super Admin masuk pada Produk Homepage dengan menambahkan produk yang disertai foto	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman Produk Homepage 2. Super Admin menambahkan produk dengan klik button Tambahkan 3. Super Admin mengisi semua field data produk 4. Super Admin melakukan upload foto 5. Super Admin klik button submit 	Super Admin berhasil menambahkan Produk Homepage	
	Super Admin masuk pada Produk Homepage dengan menambahkan produk yang tidak disertai foto	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman Produk Homepage 2. Super Admin menambahkan produk dengan klik button Tambahkan 3. Super Admin mengisi semua field data produk 4. Super Admin tidak melakukan upload foto 5. Super Admin klik button submit 	Tidak berhasil menambahkan Produk Homepage karena tidak disertai upload foto dan muncul notifikasi “pilih foto terlebih dahulu”	
	Super Admin Update data pada halaman Produk Homepage	<ol style="list-style-type: none"> 1. Super Admin melakukan update Produk Homepage 2. Super Admin klik image button update untuk mengubah data 3. Super Admin mengganti field pada data Produk Homepage yang akan diubah 4. Super Admin klik button submit 	Super Admin berhasil melakukan update data	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
	Super Admin hapus data pada halaman Produk Homepage	<ol style="list-style-type: none"> 1. Super Admin melakukan hapus data 2. Super Admin klik image button hapus dan muncul notifikasi “apakah anda yakin ingin menghapus? “, lalu klik hapus 	Super Admin berhasil menghapus data	Sesuai
	Super Admin masuk pada halaman Berita di Kelola Beranda	<ol style="list-style-type: none"> 1. Super Admin berada dalam halaman Berita 2. Super Admin memeriksa tampilan Berita 	Super Admin berhasil masuk ke halaman Berita dan dapat melihat tampilannya	Sesuai
	Super Admin Tambah data pada halaman Berita	<ol style="list-style-type: none"> 1. Super Admin menambahkan data pada Berita 2. Super Admin klik button Tambahkan untuk menambahkan data 3. Super Admin mengisi field data Berita baru yang akan ditambahkan 4. Super Admin klik button submit 	Super Admin berhasil menambahkan data pada Berita	Sesuai
	Super Admin Update data pada halaman Berita	<ol style="list-style-type: none"> 1. Super Admin melakukan update Berita 2. Super Admin klik image button update untuk mengubah data 3. Super Admin mengganti field data Berita yang akan diubah 4. Super Admin klik Submit 	Super Admin berhasil melakukan update data	Sesuai
	Super Admin hapus data pada halaman Berita	<ol style="list-style-type: none"> 1. Super Admin melakukan hapus data 2. Super Admin klik image button hapus dan muncul notifikasi “apakah anda yakin ingin menghapus? “, lalu klik hapus 	Super Admin berhasil menghapus Berita	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
Halaman Kelola Produk	Super Admin masuk pada halaman Kelola Produk	<ol style="list-style-type: none"> 1. Super Admin berada dalam halaman Kelola Produk 2. Super Admin dapat memilih jenis produk 	Super Admin berhasil masuk dan dapat memilih jenis produk pada halaman Kelola Produk	Sesuai
	Super Admin masuk pada Kelola Produk dengan menambahkan produk yang disertai foto	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman Kelola Produk 2. Super Admin menambahkan produk dengan klik button Tambahkan 3. Super Admin mengisi semua field data produk 4. Super Admin melakukan upload foto 5. Super Admin klik button submit 	Super Admin berhasil menambahkan Kelola Produk	
	Super Admin masuk pada Kelola Produk dengan menambahkan produk yang tidak disertai foto	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman Kelola Produk 2. Super Admin menambahkan produk dengan klik button Tambahkan 3. Super Admin mengisi semua field data produk 4. Super Admin tidak melakukan upload foto 5. Super Admin klik button submit 	Tidak berhasil menambahkan Kelola Produk karena tidak disertai upload foto dan muncul notifikasi “Isi Gambar terlebih dahulu”	
	Super Admin Update data pada halaman Kelola Produk	<ol style="list-style-type: none"> 1. Super Admin melakukan update Kelola Produk 2. Super Admin klik image button update untuk mengubah data 3. Super Admin mengganti field data pada Kelola produk yang akan diubah 	Super Admin berhasil melakukan update data	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		4. Super Admin klik Submit		
	Super Admin hapus data pada halaman Kelola Produk	<ol style="list-style-type: none"> 1. Super Admin melakukan hapus data 2. Super Admin klik image button hapus dan muncul notifikasi “apakah anda yakin ingin menghapus? “, lalu klik hapus 	Super Admin berhasil menghapus data	Sesuai
Halaman Kelola Inovasi	Super Admin masuk pada halaman Kelola Inovasi	<ol style="list-style-type: none"> 1. Super Admin berada dalam halaman Kelola Inovasi 2. Super Admin memeriksa tampilan Kelola Inovasi 	Super Admin dapat melihat tampilan pada halaman Kelola Inovasi	Sesuai
	Super Admin masuk pada halaman Produk Baru di Kelola Inovasi	<ol style="list-style-type: none"> 1. Super Admin berada dalam halaman Produk Baru 2. Super Admin memeriksa tampilan Produk Baru 	Super Admin dapat melihat tampilan pada halaman Produk Baru	Sesuai
	Super Admin Tambah data pada halaman Produk Baru	<ol style="list-style-type: none"> 1. Super Admin menambahkan data baru 2. Super Admin klik button Tambah untuk menambahkan data 3. Super Admin mengisi semua field data Produk Baru untuk menambahkan data baru 4. Super Admin klik button submit 	Super Admin berhasil menambahkan data pada halaman Produk Baru	Sesuai
	Super Admin Update data pada halaman Produk Baru	<ol style="list-style-type: none"> 1. Super Admin melakukan update Produk Baru 2. Super Admin klik image button update untuk mengubah data 3. Super Admin mengganti field data 	Super Admin berhasil melakukan update data	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		<p>Produk Baru yang akan diubah</p> <ol style="list-style-type: none"> Super Admin klik Submit 		
	Super Admin hapus data pada halaman Produk Baru	<ol style="list-style-type: none"> Super Admin melakukan hapus data pada halaman Produk Baru Super Admin klik image button hapus dan muncul notifikasi “apakah anda yakin ingin menghapus? “, lalu klik hapus 	Super Admin berhasil menghapus data	Sesuai
	Super Admin Tambah data dengan mengupload PPT pada halaman Paket Produk	<ol style="list-style-type: none"> Super Admin menambahkan paket baru Super Admin klik button Tambah untuk menambahkan data Super Admin mengisi semua field data Paket Produk untuk menambahkan data baru Super Admin melakukan upload PPT Klik button submit 	Super Admin berhasil menambahkan data paket produk baru pada halaman Paket Produk	Sesuai
Halaman Kelola Pertanyaan	Super Admin masuk pada halaman Kelola Pertanyaan	<ol style="list-style-type: none"> Super Admin berada pada halaman Kelola Pertanyaan Super Admin memeriksa tampilan halaman kelolaan Pertanyaan 	Super Admin berhasil melihat tampilan Kelola Pertanyaan	Sesuai
	Super Admin masuk pada halaman QNA di Kelola Pertanyaan	<ol style="list-style-type: none"> Super Admin berada pada halaman QNA Super Admin melihat status pada data QNA, untuk warna hijau “jawaban sudah direspon oleh admin” dan warna merah 	Super Admin dapat menjawab pertanyaan dan status berubah menjadi hijau	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		<p>“jawaban belum direspon oleh admin</p> <p>3. Super Admin menjawab pertanyaan yang diajukan oleh customer dengan klik image button tulis dan klik kirim</p>		
	Super Admin hapus pertanyaan pada QNA	1. Super Admin menghapus pertanyaan yang telah diajukan dengan klik image button hapus	Super Admin dapat menghapus data penanya beserta pertanyaan dan dapat menghapus respon jawabannya	Sesuai
	Super Admin masuk pada halaman FAQ di Kelola Pertanyaan	<p>1. Super Admin berada pada halaman FAQ dan klik button tambahkan pertanyaan</p> <p>2. Super Admin mengisi semua field data FAQ untuk menambahkan pertanyaan beserta jawaban pada data baru</p> <p>3. Super Admin melakukan upload foto</p> <p>4. Klik button submit</p>	Super Admin berhasil menambahkan pertanyaan beserta jawaban pada halaman FAQ dengan melakukan upload foto	Sesuai
	Super Admin menambahkan data FAQ tanpa upload foto	<p>1. Super Admin mengisi semua field data FAQ untuk menambahkan pertanyaan beserta jawaban pada data baru</p> <p>2. Super Admin mengisi data FAQ tanpa melakukan upload foto</p> <p>3. Klik button submit</p>	Super Admin berhasil menambahkan pertanyaan beserta jawaban pada halaman FAQ tanpa melakukan upload foto	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
	Super Admin hapus data pada halaman FAQ	<ol style="list-style-type: none"> 1. Super Admin melakukan hapus data pada halaman FAQ 2. Super Admin klik image button hapus dan muncul notifikasi “apakah anda yakin ingin menghapus? “, lalu klik hapus 	Super Admin berhasil menghapus data	Sesuai
	Super Admin Update data pada halaman FAQ	<ol style="list-style-type: none"> 1. Super Admin melakukan update FAQ 2. Super Admin klik image button update untuk mengubah data FAQ 3. Super Admin mengganti field data FAQ yang akan diubah 4. Super Admin klik Submit 	Super Admin berhasil melakukan update data	Sesuai
Halaman Kelola Tentang Kami	Super Admin masuk pada halaman Kelola Tentang Kami	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman Kelola Tentang Kami 2. Super Admin memeriksa tampilan pada halaman tentang kami 	Super Admin berhasil melihat tampilan Tentang Kami	Sesuai
	Super Admin masuk pada halaman Deskripsi, visi dan misi di Kelola Tentang Kami	<ol style="list-style-type: none"> 1. Super Admin memeriksa Deskripsi, visi dan misi Perusahaan 2. Super Admin Mengubah data Deskripsi, Visi dan Misi dengan mengganti teks nya 3. Super Admin klik button ubah untuk mengubah dan menyimpan data 	Super Admin berhasil mengubah data	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
	Super Admin masuk pada halaman Alamat di Kelola Tentang Kami	<ol style="list-style-type: none"> 1. Super Admin dapat menambahkan Alamat baru pada button Tambahkan 2. Super Admin menambahkan Alamat dengan mengisi semua field pada tambah Alamat dan upload foto untuk detail alamatnya 3. Klik submit 4. Super Admin melakukan update alamat pada button image update 5. Super Admin melakukan hapus Alamat dengan klik button image hapus 	Super Admin berhasil menambahkan, mengupdate, dan menghapus alamat	Sesuai
Notifikasi	Super Admin masuk pada halaman notifikasi	<ol style="list-style-type: none"> 1. Super Admin berada pada halaman Notifikasi 2. Super Admin memeriksa tampilan Notifikasi 3. Super Admin memeriksa semua Notifikasi Aktivitas, Report, dan Bonus 	Super Admin dapat melihat tampilan Notifikasi dan Semua Notifikasi Aktivitas, Report, dan Bonus	Sesuai
	Super Admin Tinjau aktivitas	<ol style="list-style-type: none"> 1. Super Admin klik button Tinjau pada salah satu Aktivitas yang masuk pada Notifikasi 2. Super Admin setelah klik button Tinjau, akan muncul notif "APAKAH aktivitas DI SETUJUI? " dan klik button "Tidak" atau "Setuju" 	Super Admin berhasil menyetujui atau tidak menyetujui Aktivitas	Sesuai

b. Admin

Test Scenario	Test Case	Test Stecp	Result	Status
Halaman utama	Admin membuka aplikasi mobile	<ol style="list-style-type: none"> 1. Admin berada pada halaman utama aplikasi mobile 2. Admin memeriksa tampilam foto slide produk dan fungsinya 3. Admin memeriksa tampilan penjelasan singkat produk 4. Admin memeriksa tampilan demplot terbaru 5. Admin memeriksa tampilan berita terbaru 	Admin dapat melihat foto slide produk dan fungsinya, penjelasa singkat produk, demplot terbaru, dan berita terbaru	Sesuai
Halaman produk kami	Admin membuka halaman produk kami	<ol style="list-style-type: none"> 1. Admin berada pada halaman produk kami 2. Admin memeriksa tampilan produk kami 	Admin dapat melihat tampilan produk kami	Sesuai
	Admin masuk pada halaman Insektisida	<ol style="list-style-type: none"> 1. Admin menekan Insektisida 2. Admin berada di halaman Insektisida 3. Admin memeriksa produk Insektisida 4. Admin melakukan search untuk menemukan produk Insektisida 	Admin dapat melihat keseluruhan produk Insektisida beserta deskripsinya dan dapat melakukan search untuk mencari produk Insektisida	Sesuai
	Admin masuk pada halaman Fungisida	<ol style="list-style-type: none"> 1. Admin menekan button Fungisida 2. Admin berada di halaman Fungisida 3. Admin memeriksa produk Fungisida 4. Admin melakukan search untuk menemukan produk Fungisida 	Admin dapat melihat keseluruhan produk fungisida beserta deskripsinya dan dapat melakukan search untuk mencari produk Fungisida	Sesuai

Test Scenario	Test Case	Test Stecp	Result	Status
	Admin masuk pada halaman Herbisida	<ol style="list-style-type: none"> 1. Admin menekan button Herbisida 2. Admin berada di halaman Herbisida 3. Admin memeriksa produk Herbisida 4. Admin melakukan search untuk menemukan produk Herbisida 	Admin dapat melihat keseluruhan produk Herbisida beserta deskripsinya dan dapat melakukan search untuk mencari produk Herbisida	Sesuai
	Admin masuk pada halaman PGR	<ol style="list-style-type: none"> 1. Admin menekan button PGR 2. Admin berada di halaman PGR 3. Admin memeriksa produk PGR 4. Admin melakukan search untuk menemukan produk PGR 	Admin dapat melihat keseluruhan produk PGR beserta deskripsinya dan dapat melakukan search untuk mencari produk PGR	Sesuai
Halaman Event, Halaman Inovasi, dan FAQ	Admin masuk pada halaman Event	<ol style="list-style-type: none"> 1. Admin menekan image button aktivitas 2. Admin masuk pada halaman aktivitas 3. Admin menekan image button demplot 4. Admin masuk pada halaman aktivitas 	Admin dapat melihat tampilan aktivitas dan tampilan pada demplot	Sesuai
	Admin masuk pada halaman Inovasi	<ol style="list-style-type: none"> 1. Admin menekan image button produk baru 2. Admin masuk pada halaman produk baru 3. Admin menekan image button paket produk 4. Admin masuk pada halaman paket produk 5. Admin klik Lihat PPT untuk mengunduh paket produk 	Admin dapat melihat tampilan produk baru, Admin dapat melihat tampilan pada paket produk, dan user dapat melihat isi paket produk dengan mengunduh PPT	Sesuai

Test Scenario	Test Case	Test Stecp	Result	Status
	Admin masuk pada halaman FAQ	<ol style="list-style-type: none"> 1. Admin menekan button FAQ 2. Admin menekan pertanyaan dan sudah tersedia jawabannya pada FAQ 	Admin dapat melihat pertanyaan yang sudah tersedia dan terdapat jawabannya pada FAQ	Sesuai
	Admin masuk pada halaman Pertanyaan Lain (QnA)	<ol style="list-style-type: none"> 1. Admin mereview pertanyaan-pertanyaan yang diajukan oleh customer lain 	Admin dapat melihat review-an yang sudah diajukan oleh customer lain	Sesuai
	Admin menambahkan pertanyaan	<ol style="list-style-type: none"> 1. Admin klik Tambahkan Pertanyaan untuk mengajukan pertanyaan 2. Admin menambahkan pertanyaan dengan mengisi semua field pada kolom pertanyaan dan disertai upload foto 3. Admin menambahkan pertanyaan dengan mengisi semua field pada kolom pertanyaan dengan tidak disertai upload foto 	Admin dapat mengajukan pertanyaan disertai upload foto ataupun tidak disertai upload foto	Sesuai
Halaman Tentang Kami	Admin masuk pada halaman Tentang kami	<ol style="list-style-type: none"> 1. Admin berada pada halaman Tentang Kami 2. Admin memeriksa tampilan profil, Visi dan Misi, serta Alamat 3. Admin klik image button pada sosial media untuk facebook 4. Admin klik image button pada sosial media untuk Instagram 	Admin dapat melihat tampilan profil, visi dan misi, Alamat, dan pada sosial media langsung bisa terhubung pada facebook, Instagram, dan youtube	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		<ol style="list-style-type: none"> Admin klik image button pada sosial media untuk Instagram Admin klik image button pada sosial media untuk youtube 		
Admin Login dari halaman Login	Admin Login dengan mengisi data yang valid di halaman Login	<ol style="list-style-type: none"> Admin berada pada halaman login aplikasi komersial PT. Cakra Agro Indonesia Admin memasukan username yang valid pada field username Admin memasukan password yang valid pada field password Admin klik tombol login 	Admin masuk ke halaman panel atau Homepage SuperAdmin	Sesuai
	Admin Login dengan mengisi data yang tidak valid di halaman Login	<ol style="list-style-type: none"> Admin berada pada halaman login aplikasi komersial PT. Cakra Agro Indonesia Admin memasukan username yang tidak valid pada field username Admin memasukan password yang tidak valid pada field password Admin klik tombol login 	Admin tidak dapat login karena data tidak valid dan muncul notifikasi "username atau password salah"	Sesuai
	Admin Login dengan mengosongkan field mandatory di halaman login	<ol style="list-style-type: none"> Admin berada pada halaman login aplikasi komersial PT. Cakra Agro Indonesia Admin mengosongkan field username dan password Admin klik tombol login 	Admin tidak dapat login dan terdapat notifikasi "masukan username dan password"	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
Halaman Dashboard	Admin masuk pada halaman Dashboard	<ol style="list-style-type: none"> Admin berada pada halaman Dashboard Admin memeriksa tampilan info data Admin username dan password 	Admin dapat melihat tampilan info data Admin	Sesuai
	Admin Update data pada halaman Dashboard	<ol style="list-style-type: none"> Admin melakukan update info Admin untuk username dan password Admin klik image button update untuk mengubah info Admin mengubah field yang akan diupdate r Admin klik buton submit 	Admin berhasil melakukan update	Sesuai
Logout Aplikasi	Admin logout aplikasi pada halaman dashboard	<ol style="list-style-type: none"> Admin berada pada halaman Dashboard Admin Admin melakukan logout dengan klik button Logout 	Admin berhasil melakukan logout	Sesuai
Halaman Lihat LOG	Admin masuk pada halaman LOG dan masuk pada halaman LOG Admin	<ol style="list-style-type: none"> Admin berada pada halaman LOG Admin klik button LOG Admin Admin memeriksa riwayat aktivitas Admin 	Admin dapat melihat tampilan riwayat aktivitas yang telah dilakukan oleh Admin	Sesuai
	Admin masuk pada halaman LOG Master	<ol style="list-style-type: none"> Admin berada pada halaman LOG Master Admin klik button LOG Master Admin memeriksa riwayat aktivitas Master 	Admin dapat melihat tampilan riwayat aktivitas yang telah dilakukan oleh Master	Sesuai
Halaman Kelola Data Master	Admin masuk pada halaman Data Admin	<ol style="list-style-type: none"> Admin berada pada halaman Data Admin Admin membuat akun baru untuk admin dengan klik button tambahkan Admin mengisi semua field untuk 	Admin berhasil menambahkan akun baru	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		menambahkan akun baru 4. Admin klik button submit		
	Admin tambah data pada halaman Data Admin	1. Admin menambahkan Data Admin 2. Admin klik button Tambah untuk menambahkan data 3. Admin mengisi semua field yang ada pada form tambah data 4. Admin klik submit	Admin berhasil menambahkan Data Admin	Sesuai
	Admin Update data pada halaman Data Admin	1. Admin melakukan update data admin 2. Admin klik image button update untuk mengubah data	Admin berhasil melakukan update data	Sesuai
	Admin hapus data pada halaman Data Admin	1. Admin melakukan hapus akun 2. Admin klik image button hapus dan muncul notifikasi “apakah anda yakin ingin menghapus? “, lalu klik hapus	Admin berhasil melakukan hapus akun	Sesuai
	Admin masuk pada halaman Data Secretary dengan membuat akun disertai foto	1. Admin berada pada halaman Data Secretary 2. Admin membuat akun baru untuk secretary dengan klik button tambahkan 3. Admin mengisi semua field untuk menambahkan akun baru 4. Admin melakukan upload foto untuk identitas sekretaris 5. Admin klik button submit	Admin berhasil menambahkan akun baru	Sesuai

Test Scenario	Test Case	Test Stecp	Result	Status
	Admin masuk pada halaman Data Secretary dengan membuat akun yang tidak disertai foto	<ol style="list-style-type: none"> 1. Admin berada pada halaman Data Secretary 2. Admin membuat akun baru untuk secretary dengan klik button tambahkan 3. Admin mengisi semua field data secretary untuk menambahkan akun baru 4. Admin tidak melakukan upload foto 5. Admin klik button submit 	Admin tidak berhasil menambahkan akun baru karena tidak disertai upload foto dan muncul notifikasi “pilih foto terlebih dahulu”	Sesuai
	Admin Update data pada halaman Data Secretary	<ol style="list-style-type: none"> 1. Admin melakukan update data secretary 2. Admin klik image button update untuk mengubah data 3. Admin mengganti field data secretary yang akan diubah 4. Admin klik Submit 	Admin berhasil melakukan update data	Sesuai
	Admin hapus data pada halaman Data Secretary	<ol style="list-style-type: none"> 1. Admin melakukan hapus akun 2. Admin klik image button hapus dan muncul notifikasi “apakah anda yakin ingin menghapus? “, lalu klik hapus 	Admin berhasil melakukan hapus akun	Sesuai
	Admin masuk pada halaman Data Supervisor dengan membuat akun yang disertai foto	<ol style="list-style-type: none"> 1. Admin berada pada halaman Data Supervisor 2. Admin membuat akun baru untuk supervisor dengan klik button tambahkan 3. Admin mengisi semua field data supervisor untuk menambahkan akun baru 	Admin berhasil menambahkan akun baru	Sesuai

Test Scenario	Test Case	Test Stecp	Result	Status
		<ol style="list-style-type: none"> 4. Admin tidak melakukan upload foto 5. Admin klik button submit 		
	Admin masuk pada halaman Data Supervisor dengan membuat akun yang tidak disertai foto	<ol style="list-style-type: none"> 1. Admin berada pada halaman Data Supervisor 2. Admin membuat akun baru untuk supervisor dengan klik button tambahkan 3. Admin mengisi semua field data supervisor untuk menambahkan akun baru 4. Admin tidak melakukan upload foto 5. Admin klik button submit 	Tidak berhasil menambahkan akun baru karena tidak disertai upload foto dan muncul notifikasi “pilih foto terlebih dahulu”	Sesuai
	Admin Update data pada halaman Data Supervisor	<ol style="list-style-type: none"> 1. Admin melakukan update data supervisor 2. Admin klik image button update untuk mengubah data 3. Admin mengganti field data Supervisor yang akan diubah 4. Admin klik Submit 	Admin berhasil melakukan update data	Sesuai
	Admin hapus data pada halaman Data Supervisor	<ol style="list-style-type: none"> 1. Admin melakukan hapus akun 2. Admin klik image button hapus dan muncul notifikasi “apakah anda yakin ingin menghapus? “, lalu klik hapus 	Admin berhasil melakukan hapus akun	Sesuai
	Admin masuk pada halaman Data Demonstrator dengan membuat akun	<ol style="list-style-type: none"> 1. Admin berada pada halaman Data Supervisor 2. Admin membuat akun baru untuk Demonstrator dengan 	Admin berhasil menambahkan akun baru	Sesuai

Test Scenario	Test Case	Test Stecp	Result	Status
	yang disertai foto	<p>klik button tambahkan</p> <ol style="list-style-type: none"> Admin mengisi semua field data Demonstrator untuk menambahkan akun baru Admin melakukan upload foto Admin klik button submit 		
	Admin masuk pada halaman Data Demonstrator dengan membuat akun yang tidak disertai foto	<ol style="list-style-type: none"> Admin berada pada halaman Data Demonstrator Admin membuat akun baru untuk Demonstrator dengan klik button tambahkan Admin mengisi semua field data Demonstrator untuk menambahkan akun baru Admin tidak melakukan upload foto Admin klik button submit 	Tidak berhasil menambahkan akun baru karena tidak disertai upload foto dan muncul notifikasi “pilih foto terlebih dahulu”	Sesuai
	Admin Update data pada halaman Data Demonstrator	<ol style="list-style-type: none"> Admin melakukan update data Demonstrator Admin klik image button update untuk mengubah data Admin mengganti field data Demonstrator yang akan diubah Admin klik Submit 	Admin berhasil melakukan update data	Sesuai
	Admin hapus data pada halaman Data Demonstrator	<ol style="list-style-type: none"> Admin melakukan hapus akun Admin klik image button hapus dan muncul notifikasi “apakah anda yakin 	Admin berhasil melakukan hapus akun	Sesuai

Test Scenario	Test Case	Test Stecp	Result	Status
		ingin menghapus? “, lalu klik hapus		
	Admin masuk pada halaman Data Distributor dengan membuat akun yang disertai foto	<ol style="list-style-type: none"> 1. Admin berada pada halaman Data Distributor 2. Admin membuat akun baru untuk Distributor dengan klik button tambahkan 3. Admin mengisi semua field data Distributor untuk menambahkan akun baru 4. Admin melakukan upload foto 5. Admin klik button submit 	Admin berhasil menambahkan akun baru	Sesuai
	Admin masuk pada halaman Data Distributor dengan membuat akun yang tidak disertai foto	<ol style="list-style-type: none"> 1. Admin berada pada halaman Data Distributor 2. Admin membuat akun baru untuk Distributor dengan klik button tambahkan 3. Admin mengisi semua field data Distributor untuk menambahkan akun baru 4. Admin tidak melakukan upload foto 5. Admin klik button submit 	Tidak berhasil menambahkan akun baru karena tidak disertai upload foto dan muncul notifikasi “pilih foto terlebih dahulu”	Sesuai
	Admin Update data pada halaman Data Distributor	<ol style="list-style-type: none"> 1. Admin melakukan update data Distributor 2. Admin klik image button update untuk mengubah data 	Admin berhasil melakukan update data	Sesuai

Test Scenario	Test Case	Test Stecp	Result	Status
		<ol style="list-style-type: none"> 3. Admin mengganti field data Distributor yang akan diubah 4. Admin klik Submit 		
	Admin hapus data pada halaman Data Distributor	<ol style="list-style-type: none"> 1. Admin melakukan hapus akun 2. Admin klik image button hapus dan muncul notifikasi “apakah anda yakin ingin menghapus? “, lalu klik hapus 	Admin berhasil menambahkan akun baru	Sesuai
	Admin masuk pada halaman Data Manager dengan membuat akun yang disertai foto	<ol style="list-style-type: none"> 1. Admin berada pada halaman Data Manager 2. Admin membuat akun baru untuk Manager dengan klik button tambahkan 3. Admin mengisi semua field data Manager untuk menambahkan akun baru 4. Admin melakukan upload foto 5. Admin klik button submit 	Admin berhasil menambahkan akun baru	Sesuai
	Admin masuk pada halaman Data Manager dengan membuat akun yang tidak disertai foto	<ol style="list-style-type: none"> 1. Admin berada pada halaman Data Manager 2. Admin membuat akun baru untuk Manager dengan klik button tambahkan 3. Admin mengisi semua field data Manager untuk menambahkan akun baru 4. Admin tidak melakukan upload foto 5. Admin klik button submit 	Tidak berhasil menambahkan akun baru karena tidak disertai upload foto dan muncul notifikasi “pilih foto terlebih dahulu”	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
	Admin Update data pada halaman Data Manager	<ol style="list-style-type: none"> 1. Admin melakukan update data Manager 2. Admin klik image button update untuk mengubah data 3. Admin mengganti field data Manager yang akan diubah 4. Admin klik Submit 	Admin berhasil melakukan update data	Sesuai
	Admin hapus data pada halaman Data Manager	<ol style="list-style-type: none"> 1. Admin melakukan hapus akun 2. Admin klik image button hapus dan muncul notifikasi “apakah anda yakin ingin menghapus? “, lalu klik hapus 	Admin berhasil menambahkan akun baru	Sesuai
Halaman Kelola Report Demonstrator	Admin masuk pada halaman Kelola Report Demonstrator	<ol style="list-style-type: none"> 1. Admin berada pada halaman Kelola Report Demonstrator 2. Admin memeriksa tampilan pada Daftar Report 	Admin berhasil masuk dan dapat melihat Daftar Report	Sesuai
	Admin masuk pada halaman Daftar Report Demonstrator dengan menambahkan produk yang disertai foto	<ol style="list-style-type: none"> 1. Admin berada pada halaman halaman Daftar Report Demonstrator 2. Admin menambahkan produk dengan klik button Tambahkan 3. Admin mengisi semua field data produk 4. Admin melakukan upload foto 5. Admin klik button submit 	Admin berhasil menambahkan Daftar Report Baru	Sesuai
	Admin masuk pada halaman Daftar Report Demonstrator dengan menambahkan produk yang tidak disertai foto	<ol style="list-style-type: none"> 1. Admin berada pada halaman halaman Daftar Report Demonstrator 2. Admin menambahkan produk dengan klik button Tambahkan 3. Admin mengisi semua field data produk 	Tidak berhasil menambahkan akun baru karena tidak disertai upload foto dan muncul notifikasi “pilih foto	Sesuai

Test Scenario	Test Case	Test Stecp	Result	Status
		<ol style="list-style-type: none"> 4. Admin tidak melakukan upload foto 5. Admin klik button submit 	terlebih dahulu”	
	Admin Update data pada halaman Daftar Report Demonstrator	<ol style="list-style-type: none"> 1. Admin melakukan update Daftar Report Demonstrator 2. Admin klik image button update untuk mengubah data 3. Admin mengganti field pada data Daftar Report Demonstrator yang akan diubah 4. Admin memilih opsi “lama” untuk menyatakan bahwa laporan demonstrator tersebut sudah setuju 5. Admin memilih opsi “baru” untuk menyatakan bahwa laporan demonstrator tersebut belum setuju 6. Admin klik Submit 	Admin berhasil melakukan update data dengan status berwarna hijau “disetujui” dan status berwarna merah “belum disetujui”	Sesuai
	Admin hapus data pada halaman Daftar Report Demonstrator	<ol style="list-style-type: none"> 1. Admin melakukan hapus akun 2. Admin klik image button hapus dan muncul notifikasi “apakah anda yakin ingin menghapus? “, lalu klik hapus 	Admin berhasil menambahkan akun baru	Sesuai
Halaman Kelola Beranda	Admin masuk pada halaman Kelola Beranda	<ol style="list-style-type: none"> 1. Admin berada dalam halaman Kelola Beranda 2. Admin memeriksa tampilan Kelola Beranda 	Admin berhasil masuk ke halaman Kelola Beranda dan dapat melihat tampilannya	Sesuai
	Admin masuk pada halaman Homepage di Kelola Beranda	<ol style="list-style-type: none"> 1. Admin berada dalam halaman Homepage 2. Admin mengubah judul dan deskripsi pada Homepage 	Admin berhasil mengubah judul dan deskripsi, serta	Sesuai

Test Scenario	Test Case	Test Stecp	Result	Status
		3. Admin melakukan upload foto tentang perusahaan	dapat mengupload foto tentang perusahaan	
	Admin masuk pada halaman Produk Homepage di Kelola Beranda	<ol style="list-style-type: none"> Admin berada dalam halaman Produk Homepage Admin memeriksa tampilan Produk Homepage 	Admin berhasil masuk ke halaman Produk Homepage dan dapat melihat tampilannya	Sesuai
	Admin masuk pada Produk Homepage dengan menambahkan produk yang disertai foto	<ol style="list-style-type: none"> Admin berada pada halaman Produk Homepage Admin menambahkan produk dengan klik button Tambahkan Admin mengisi semua field data produk Admin melakukan upload foto Admin klik button submit 	Admin berhasil menambahkan Produk Homepage	
	Admin masuk pada Produk Homepage dengan menambahkan produk yang tidak disertai foto	<ol style="list-style-type: none"> Admin berada pada halaman Produk Homepage Admin menambahkan produk dengan klik button Tambahkan Admin mengisi semua field data produk Admin tidak melakukan upload foto Admin klik button submit 	Tidak berhasil menambahkan Produk Homepage karena tidak disertai upload foto dan muncul notifikasi “pilih foto terlebih dahulu”	
	Admin Update data pada halaman Produk Homepage	<ol style="list-style-type: none"> Admin melakukan update Produk Homepage Admin klik image button update untuk mengubah data 	Admin berhasil melakukan update data	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		<ol style="list-style-type: none"> 3. Admin mengganti field pada data Produk Homepage yang akan diubah 4. Admin klik Submit 		
	Admin hapus data pada halaman Produk Homepage	<ol style="list-style-type: none"> 1. Admin melakukan hapus data 2. Admin klik image button hapus dan muncul notifikasi “apakah anda yakin ingin menghapus? “, lalu klik hapus 	Admin berhasil menghapus data	Sesuai
	Admin masuk pada halaman Berita di Kelola Beranda	<ol style="list-style-type: none"> 1. Admin berada dalam halaman Berita 2. Admin memeriksa tampilan Berita 	Admin berhasil masuk ke halaman Berita dan dapat melihat tampilannya	Sesuai
	Admin Tambah data pada halaman Berita	<ol style="list-style-type: none"> 1. Admin menambahkan data pada Berita 2. Admin klik button Tambah untuk menambahkan data 3. Super Admin mengisi field data Berita baru yang akan ditambahkan 4. Super Admin klik button submit 	Admin berhasil menambahkan data pada Berita	Sesuai
	Admin Update data pada halaman Berita	<ol style="list-style-type: none"> 1. Admin melakukan update Berita 2. Admin klik image button update untuk mengubah data 3. Admin mengganti field data Berita yang akan diubah 4. Admin klik Submit 	Admin berhasil melakukan update data	Sesuai
	Admin hapus data pada halaman Berita	<ol style="list-style-type: none"> 1. Admin melakukan hapus data 2. Admin klik image button hapus dan muncul notifikasi “apakah anda yakin ingin menghapus? “, lalu klik hapus 	Admin berhasil menghapus Berita	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
Halaman Kelola Produk	Admin masuk pada halaman Kelola Produk	<ol style="list-style-type: none"> 1. Admin berada dalam halaman Kelola Produk 2. Admin dapat memilih jenis produk 	Admin berhasil masuk dan dapat memilih jenis produk pada halaman Kelola Produk	Sesuai
	Admin masuk pada Kelola Produk dengan menambahkan produk yang disertai foto	<ol style="list-style-type: none"> 1. Admin berada pada halaman Kelola Produk 2. Admin menambahkan produk dengan klik button Tambahkan 3. Admin mengisi semua field data produk 4. Admin melakukan upload foto 5. Admin klik button submit 	Admin berhasil menambahkan Kelola Produk	
	Admin masuk pada Kelola Produk dengan menambahkan produk yang tidak disertai foto	<ol style="list-style-type: none"> 1. Admin berada pada halaman Kelola Produk 2. Admin menambahkan produk dengan klik button Tambahkan 3. Admin mengisi semua field data produk 4. Admin tidak melakukan upload foto 5. Admin klik button submit 	Tidak berhasil menambahkan Kelola Produk karena tidak disertai upload foto dan muncul notifikasi “pilih foto terlebih dahulu”	
	Admin Update data pada halaman Kelola Produk	<ol style="list-style-type: none"> 1. Admin melakukan update Kelola Produk 2. Admin klik image button update untuk mengubah data 3. Admin mengganti field data pada Kelola produk yang akan diubah 4. Admin klik Submit 	Admin berhasil melakukan update data	Sesuai
	Admin hapus data pada	<ol style="list-style-type: none"> 1. Admin melakukan hapus data 	Admin berhasil	Sesuai

Test Scenario	Test Case	Test Stecp	Result	Status
	halaman Kelola Produk	2. Admin klik image button hapus dan muncul notifikasi “apakah anda yakin ingin menghapus? “, lalu klik hapus	menghapus data	
Halaman Kelola Inovasi	Admin masuk pada halaman Kelola Inovasi	1. Admin berada dalam halaman Kelola Inovasi 2. Admin memeriksa tampilan Kelola Inovasi	Admin dapat melihat tampilan pada halaman Kelola Inovasi	Sesuai
	Admin masuk pada halaman Produk Baru di Kelola Inovasi	1. Admin berada dalam halaman Produk Baru 2. Admin memeriksa tampilan Produk Baru	Admin dapat melihat tampilan pada halaman Produk Baru	Sesuai
	Admin Tambah data pada halaman Produk Baru	1. Admin menambahkan data baru 2. Admin klik button Tambah untuk menambahkan data 3. Admin mengisi semua field data Produk Baru untuk menambahkan data baru 4. Admin klik button submit	Admin berhasil menambahkan data pada halaman Produk Baru	Sesuai
	Admin Update data pada halaman Produk Baru	1. Admin melakukan update Produk Baru 2. Admin klik image button update untuk mengubah data 3. Admin mengganti field data Produk Baru yang akan diubah 4. Admin klik Submit	Admin berhasil melakukan update data	Sesuai
	Admin hapus data pada halaman Produk Baru	1. Admin melakukan hapus data pada halaman Produk Baru 2. Admin klik image button hapus dan muncul notifikasi “apakah anda yakin	Admin berhasil menghapus data	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		ingin menghapus? “, lalu klik hapus		
	Admin Tambah data dengan mengupload PPT pada halaman Paket Produk	<ol style="list-style-type: none"> 1. Admin menambahkan paket baru 2. Admin klik button Tambah untuk menambahkan data 3. Admin mengisi semua field data Paket Produk untuk menambahkan data baru 4. Admin melakukan upload PPT 5. Klik button submit 	Admin berhasil menambahkan data paket produk baru pada halaman Paket Produk	Sesuai
Halaman Kelola Pertanyaan	Admin masuk pada halaman Kelola Pertanyaan	<ol style="list-style-type: none"> 1. Admin berada pada halaman Kelola Pertanyaan 2. Admin memeriksa tampilan halaman kelolaan Pertanyaan 	Admin berhasil melihat tampilan Kelola Pertanyaan	Sesuai
	Admin masuk pada halaman QNA di Kelola Pertanyaan	<ol style="list-style-type: none"> 1. Admin berada pada halaman QNA dan menjawab pertanyaan yang diajukan oleh customer dengan klik image button tulis dan klik kirim 2. Admin mennghapus pertanyaan yang telah diajukan dengan klik image button hapus 	Admin dapat menjawab pertanyaan dan menghapus pertanyaan	Sesuai
	Admin masuk pada halaman QNA di Kelola Pertanyaan	<ol style="list-style-type: none"> 1. Admin berada pada halaman QNA 2. Admin melihat status pada data QNA, untuk warna hijau “jawaban sudah direspon oleh admin” dan warna merah “jawaban belum direspon oleh admin 	Admin dapat menjawab pertanyaan dan status berubah menjadi hijau	Sesuai

Test Scenario	Test Case	Test Stecp	Result	Status
		3. Admin menjawab pertanyaan yang diajukan oleh customer dengan klik image button tulis dan klik kirim		
	Admin hapus pertanyaan pada QNA	1. Admin menghapus pertanyaan yang telah diajukan dengan klik image button hapus	Admin dapat menghapus data penanya beserta pertanyaan dan dapat menghapus respon jawabannya	
	Admin masuk pada halaman FAQ di Kelola Pertanyaan	<ol style="list-style-type: none"> 1. Admin berada pada halaman FAQ dan klik button tambahkan pertanyaan 2. Admin mengisi semua field data FAQ untuk menambahkan pertanyaan beserta jawaban pada data baru 3. Admin melakukan upload foto 4. Admin klik button submit 	Admin berhasil menambahkan pertanyaan beserta jawaban pada halaman FAQ dengan melakukan upload foto	Sesuai
	Admin menambahkan data FAQ tanpa upload foto	<ol style="list-style-type: none"> 1. Admin mengisi semua field data FAQ untuk menambahkan pertanyaan beserta jawaban pada data baru 2. Admin mengisi data FAQ tanpa melakukan upload foto 3. Klik button submit 	Admin berhasil menambahkan pertanyaan beserta jawaban pada halaman FAQ tanpa melakukan upload foto	Sesuai
	Admin hapus data pada halaman FAQ	<ol style="list-style-type: none"> 1. Admin melakukan hapus data pada halaman FAQ 2. Admin klik image button hapus dan muncul notifikasi "apakah anda yakin 	Admin berhasil menghapus data	Sesuai

Test Scenario	Test Case	Test Stecp	Result	Status
		ingin menghapus? “, lalu klik hapus		
	Admin Update data pada halaman FAQ	<ol style="list-style-type: none"> 1. Admin melakukan update FAQ 2. Admin klik image button update untuk mengubah data FAQ 	Admin berhasil melakukan update data	Sesuai
Halaman Kelola Tentang Kami	Admin masuk pada halaman Kelola Tentang Kami	<ol style="list-style-type: none"> 1. Admin berada pada halaman Kelola Tentang Kami 2. Admin memeriksa tampilan pada halaman tentang kami 	Admin berhasil melihat tampilan Tentang Kami	Sesuai
	Admin masuk pada halaman Deskripsi, visi dan misi di Kelola Tentang Kami	<ol style="list-style-type: none"> 1. Admin memeriksa Deskripsi, visi dan misi Perusahaan 2. Admin Mengubah data Deskripsi, Visi dan Misi dengan mengganti teks nya 3. Admin klik button ubah untuk mengubah dan menyimpan data 	Admin berhasil mengubah data	Sesuai
	Admin masuk pada halaman Alamat di Kelola Tentang Kami	<ol style="list-style-type: none"> 1. Admin dapat menambahkan Alamat baru pada button Tambahkan 2. Menambahkan Alamat dengan mengisi semua field pada tambah Alamat dan upload foto untuk detail alamatnya 3. Klik submit 4. Melakukan update alamat pada button image update 5. Melakukan hapus Alamat dengan klik button image hapus 	Admin berhasil menambahkan, mengupdate, dan menghapus alamat	Sesuai
Notifikasi	Admin masuk pada halaman notifikasi	<ol style="list-style-type: none"> 1. Admin berada pada halaman Notifikasi 2. Admin memeriksa tampilan Notifikasi 3. Admin memeriksa semua Notifikasi 	Admin dapat melihat tampilan Notifikasi dan Semua Notifikasi	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		Aktivitas. Report, dan Bonus	Aktivitas, Report, dan Bonus	
	Admin Tinjau aktivitas	<ol style="list-style-type: none"> Admin klik button Tinjau pada salah satu Aktivitas yang masuk pada Notifikasi Admin setelah klik button Tinjau, akan muncul notif “APAKAH aktivitas DI SETUJUI? “ dan klik button “Tidak” atau “Setuju” 	Admin berhasil menyetujui atau tidak menyetujui Aktivitas	Sesuai

c. Manager

Test Scenario	Test Case	Test Step	Result	Status
Halaman utama	Manager membuka aplikasi mobile	<ol style="list-style-type: none"> Manager berada pada halaman utama aplikasi mobile Manager memeriksa tampilam foto slide produk dan fungsinya Manager memeriksa tampilan penjelasan singkat produk Manager memeriksa tampilan demplot terbaru Manager memeriksa tampilan berita terbaru 	Manager dapat melihat foto slide produk dan fungsinya, penjelasa singkat produk, demplot terbaru, dan berita terbaru	Sesuai
Halaman produk kami	Manager membuka halaman produk kami	<ol style="list-style-type: none"> Manager berada pada halaman produk kami Manager memeriksa tampilan produk kami 	Manager dapat melihat tampilan produk kami	Sesuai
	Manager masuk pada halaman Insektisida	<ol style="list-style-type: none"> Manager menekan Insektisida Manager berada di halaman Insektisida Manager memeriksa produk Insektisida 	Manager dapat melihat keseluruhan produk Insektisida beserta	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		4. Manager melakukan search untuk menemukan produk Insektisida	deskripsinya dan dapat melakukan search untuk mencari produk Insektisida	
	Manager masuk pada halaman Fungisida	<ol style="list-style-type: none"> 1. Manager menekan button Fungisida 2. Manager berada di halaman Fungisida 3. Manager memeriksa produk Fungisida 4. Manager melakukan search untuk menemukan produk Fungisida 	Manager dapat melihat keseluruhan produk fungisida beserta deskripsinya dan dapat melakukan search untuk mencari produk Fungisida	Sesuai
	Manager masuk pada halaman Herbisida	<ol style="list-style-type: none"> 1. Manager menekan button Herbisida 2. Manager berada di halaman Herbisida 3. Manager memeriksa produk Herbisida 4. Manager melakukan search untuk menemukan produk Herbisida 	Manager dapat melihat keseluruhan produk Herbisida beserta deskripsinya dan dapat melakukan search untuk mencari produk Herbisida	Sesuai
	Manager masuk pada halaman PGR	<ol style="list-style-type: none"> 1. Manager menekan button PGR 2. Manager berada di halaman PGR 3. Manager memeriksa produk PGR 4. Manager melakukan search untuk menemukan produk PGR 	Manager dapat melihat keseluruhan produk PGR beserta deskripsinya dan dapat melakukan search untuk mencari produk PGR	Sesuai
Halaman Event, Halaman	Manager masuk pada halaman Event	1. Manager menekan image button aktivitas	Manager dapat melihat tampilan	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
Inovasi, dan FAQ		<ol style="list-style-type: none"> 2. Manager masuk pada halaman aktivitas 3. Manager menekan image button demplot 4. Manager masuk pada halaman aktivitas 	aktivitas dan tampilan pada demplot	
	Manager masuk pada halaman Inovasi	<ol style="list-style-type: none"> 1. Manager menekan image button produk baru 2. Manager masuk pada halaman produk baru 3. Manager menekan image button paket produk 4. Manager masuk pada halaman paket produk 5. Manager klik Lihat PPT untuk mengunduh paket produk 	Manager dapat melihat tampilan produk baru, Manager dapat melihat tampilan pada paket produk, dan user dapat melihat isi paket produk dengan mengunduh PPT	Sesuai
	Manager masuk pada halaman FAQ	<ol style="list-style-type: none"> 1. Manager menekan button FAQ 2. Manager menekan pertanyaan dan sudah tersedia jawabannya pada FAQ 	Manager dapat melihat pertanyaan yang sudah tersedia dan terdapat jawabannya pada FAQ	Sesuai
	Manager masuk pada halaman Pertanyaan Lain (QnA)	<ol style="list-style-type: none"> 1. Manager mereview pertanyaan-pertanyaan yang diajukan oleh customer lain 	Manager dapat melihat review-an yang sudah diajukan oleh customer lain	Sesuai
	Manager menambahkan pertanyaan	<ol style="list-style-type: none"> 1. Manager klik Tambahkan Pertanyaan untuk mengajukan pertanyaan 2. Manager menambahkan pertanyaan dengan mengisi semua field pada kolom pertanyaan dan disertai upload foto 	Manager dapat mengajukan pertanyaan disertai upload foto ataupun tidak disertai upload foto	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		3. Manager menambahkan pertanyaan dengan mengisi semua field pada kolom pertanyaan dengan tidak disertai upload foto		
Halaman Tentang Kami	Manager masuk pada halaman Tentang kami	<ol style="list-style-type: none"> 1. Manager berada pada halaman Tentang Kami 2. Manager memeriksa tampilan profil, Visi dan Misi, serta Alamat 3. Manager klik image button pada sosial media untuk facebook 4. Manager klik image button pada sosial media untuk Instagram 3. Manager klik image button pada sosial media untuk Instagram 4. Manager klik image button pada sosial media untuk youtube 	Manager dapat melihat tampilan profil, visi dan misi, Alamat, dan pada sosial media langsung bisa terhubung pada facebook, Instagram, dan youtube	Sesuai
Manager Login dari halaman Login	Manager Login dengan mengisi data yang valid di halaman Login	<ol style="list-style-type: none"> 1. Manager berada pada halaman login aplikasi komersial PT. Cakra Agro Indonesia 2. Manager memasukan username yang valid pada field username 3. Manager memasukan password yang valid pada field password 4. Manager klik tombol login 	Manager masuk ke halaman panel atau Homepage Manager	Sesuai
	Manager Login dengan mengisi data yang tidak valid di halaman Login	<ol style="list-style-type: none"> 1. Manager berada pada halaman login aplikasi komersial PT. Cakra Agro Indonesia 2. Manager memasukan username yang tidak 	Manager tidak dapat login karena data tidak valid dan muncul notifikasi	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		<p>valid pada field username</p> <ol style="list-style-type: none"> 3. Manager memasukan password yang tidak valid pada field password 4. Manager klik tombol login 	“username atau password salah”	
	Manager Login dengan mengosongkan field mandatory di halaman login	<ol style="list-style-type: none"> 1. Manager berada pada halaman login aplikasi komersial PT. Cakra Agro Indonesia 2. Manager mengosongkan field username dan password 3. Manager klik tombol login 	Manager tidak dapat login dan terdapat notifikasi “masukan username dan password”	Sesuai
Halaman Dashboard	Manager masuk pada halaman Dashboard	<ol style="list-style-type: none"> 1. Manager berada pada halaman Dashboard 2. Manager memeriksa tampilan Dashboard 	Manager dapat melihat tampilan Dashboard Manager	Sesuai
	Manager masuk pada halaman Info Manager di halaman Dashboard Manager	<ol style="list-style-type: none"> 1. Manager berada pada halaman Info Manager 2. Manager memeriksa tampilan pada halaman Info Manager 	Manager dapat melihat tampilan Info profil Manager	Sesuai
	Manager Edit data pada halaman Info Manager	<ol style="list-style-type: none"> 1. Manager melakukan update info Manager 2. Manager klik button Edit untuk mengubah info data Manager 3. Manager ganti foto manager dan mengubah field nama manager, username, password, dan konfirmasi password 4. Manager klik button simpan 	Manager berhasil melakukan edit data Manager	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
	Manager logout aplikasi	<ol style="list-style-type: none"> 1. Manager klik button logout pada info manager 2. Manager melakukan logout 	Manager berhasil melakukan logout aplikasis	Sesuai
	Manager masuk pada halaman Info Supervisor di halaman Dashboard Manager	<ol style="list-style-type: none"> 1. Manager berada pada halaman Info Supervisor 2. Manager memeriksa tampilan info profil data Supervisor 	Manager dapat melihat tampilan Info profil pada data Supervisor	Sesuai
	Manager masuk pada halaman Info Secretary di halaman Dashboard Manager	<ol style="list-style-type: none"> 1. Manager berada pada halaman Info Secretary 2. Manager memeriksa tampilan info profil data Secretary 	Manager dapat melihat tampilan Info profil pada data Secretary	Sesuai
	Manager masuk pada halaman Info Demonstrator di halaman Dashboard Manager	<ol style="list-style-type: none"> 1. Manager berada pada halaman Info Demonstrator 2. Manager memeriksa tampilan info profil data Demonstrator 	Manager dapat melihat tampilan Info profil pada data Demonstrator	Sesuai
	Manager masuk pada halaman Info Distributor di halaman Dashboard Manager	<ol style="list-style-type: none"> 1. Manager berada pada halaman Info Distributor 2. Manager memeriksa tampilan info profil data Distributor 	Manager dapat melihat tampilan Info profil pada data Distributor	Sesuai
Halaman Kelola Manager	Manager masuk pada halaman Kelola Manager	<ol style="list-style-type: none"> 1. Manager berada pada halaman Kelola Manager 2. Manager memeriksa tampilan Kelola Manager 	Manager dapat melihat tampilan Kelola Manager	Sesuai
	Manager masuk pada halaman Aktivitas Supervisor di halaman	<ol style="list-style-type: none"> 1. Manager berada pada halaman Aktivitas Supervisor 2. Manager memeriksa tampilan Aktivitas Supervisor 	Manager dapat melihat tampilan Aktivitas Supervisor	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
	Kelola Manager			
	Manager masuk pada halaman Report Demonstrator di halaman Kelola Manager	<ol style="list-style-type: none"> 1. Manager berada pada halaman Daftar Report Demonstrator 2. Manager dapat melihat Daftar Report Demonstrator dengan status hijau “disetujui” dan merah “belum disetujui” 	Manager dapat melihat tampilan pada Daftar Report Demonstrator	Sesuai
	Manager masuk pada halaman Bonus Distributor di halaman Kelola Manager	<ol style="list-style-type: none"> 1. Manager berada pada halaman Bonus Distributor 2. Manager memeriksa Bonus yang diperoleh Distributor 	Manager dapat melihat tampilan pada Bonus yang diperoleh Distributor	Sesuai

d. Secretary

Test Scenario	Test Case	Test Step	Result	Status
Halaman utama	Secretary membuka aplikasi mobile	<ol style="list-style-type: none"> 1. Secretary berada pada halaman utama aplikasi mobile 2. Secretary memeriksa tampilan foto slide produk dan fungsinya 3. Secretary memeriksa tampilan penjelasan singkat produk 4. Secretary memeriksa tampilan demplot terbaru 5. Secretary memeriksa tampilan berita terbaru 	Secretary dapat melihat foto slide produk dan fungsinya, penjelasan singkat produk, demplot terbaru, dan berita terbaru	Sesuai
Halaman produk kami	Secretary membuka halaman produk kami	<ol style="list-style-type: none"> 1. Secretary berada pada halaman produk kami 2. Secretary memeriksa tampilan produk kami 	Secretary dapat melihat tampilan produk kami	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
	Secretary masuk pada halaman Insektisida	<ol style="list-style-type: none"> 1. Secretary menekan Insektisida 2. Secretary berada di halaman Insektisida 3. Secretary mengecek produk Insektisida 4. Secretary melakukan search untuk menemukan produk Insektisida 	Secretary dapat melihat keseluruhan produk Insektisida beserta deskripsinya dan dapat melakukan search untuk mencari produk Insektisida	Sesuai
	Secretary masuk pada halaman Fungisida	<ol style="list-style-type: none"> 1. Secretary menekan button Fungisida 2. Secretary berada di halaman Fungisida 3. Secretary mengecek produk Fungisida 4. Secretary melakukan search untuk menemukan produk Fungisida 	Secretary dapat melihat keseluruhan produk fungisida beserta deskripsinya dan dapat melakukan search untuk mencari produk Fungisida	Sesuai
	Secretary masuk pada halaman Herbisida	<ol style="list-style-type: none"> 1. Secretary menekan button Herbisida 2. Secretary berada di halaman Herbisida 3. Secretary mengecek produk Herbisida 4. Secretary melakukan search untuk menemukan produk Herbisida 	Secretary dapat melihat keseluruhan produk Herbisida beserta deskripsinya dan dapat melakukan search untuk mencari produk Herbisida	Sesuai
	Secretary masuk pada halaman PGR	<ol style="list-style-type: none"> 1. Secretary menekan button PGR 2. Secretary berada di halaman PGR 3. Secretary mengecek produk PGR 4. Secretary melakukan search untuk 	Secretary dapat melihat keseluruhan produk PGR beserta deskripsinya dan dapat melakukan	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		menemukan produk PGR	search untuk mencari produk PGR	
Halaman Event, Halaman Inovasi, dan FAQ	Secretary masuk pada halaman Event	<ol style="list-style-type: none"> 1. Secretary menekan image button aktivitas 2. Secretary masuk pada halaman aktivitas 3. Secretary menekan image button demplot 4. Secretary masuk pada halaman aktivitas 	Secretary dapat melihat tampilan aktivitas dan tampilan pada demplot	Sesuai
	Secretary masuk pada halaman Inovasi	<ol style="list-style-type: none"> 1. Secretary menekan image button produk baru 2. Secretary masuk pada halaman produk baru 3. Secretary menekan image button paket produk 4. Secretary masuk pada halaman paket produk 5. Secretary klik Lihat PPT untuk mengunduh paket produk 	Secretary dapat melihat tampilan produk baru, Secretary dapat melihat tampilan pada paket produk, dan user dapat melihat isi paket produk dengan mengunduh PPT	Sesuai
	Secretary masuk pada halaman FAQ	<ol style="list-style-type: none"> 1. Secretary menekan button FAQ 2. Secretary menekan pertanyaan dan sudah tersedia jawabannya pada FAQ 	Secretary dapat melihat pertanyaan yang sudah tersedia dan terdapat jawabannya pada FAQ	Sesuai
	Secretary masuk pada halaman Pertanyaan Lain (QnA)	<ol style="list-style-type: none"> 1. Secretary mereview pertanyaan-pertanyaan yang diajukan oleh customer lain 	Secretary dapat melihat review-an yang sudah diajukan oleh customer lain	Sesuai
	Secretary menambahkan pertanyaan	<ol style="list-style-type: none"> 1. Secretary klik Tambahkan Pertanyaan untuk mengajukan pertanyaan 2. Secretary menambahkan pertanyaan dengan 	Secretary dapat mengajukan pertanyaan disertai upload foto ataupun tidak disertai upload foto	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		<p>mengisi semua field pada kolom pertanyaan dan disertai upload foto</p> <p>3. Secretary menambahkan pertanyaan dengan mengisi semua field pada kolom pertanyaan dengan tidak disertai upload foto</p>		
Halaman Tentang Kami	Secretary masuk pada halaman Tentang kami	<ol style="list-style-type: none"> 1. Secretary berada pada halaman Tentang Kami 2. Secretary memeriksa tampilan profil, Visi dan Misi, serta Alamat 3. Secretary klik image button pada sosial media untuk facebook 4. Secretary klik image button pada sosial media untuk Instagram 5. Secretary klik image button pada sosial media untuk Instagram 6. Secretary klik image button pada sosial media untuk youtube 	Secretary dapat melihat tampilan profil, visi dan misi, Alamat, dan pada sosial media langsung bisa terhubung pada facebook, Instagram, dan youtube	Sesuai
Secretary Login dari halaman Login	Secretary Login dengan mengisi data yang valid di halaman Login	<ol style="list-style-type: none"> 1. Secretary berada pada halaman login aplikasi komersial PT. Cakra Agro Indonesia 2. Secretary memasukan username yang valid pada field username 3. Secretary memasukan password yang valid pada field password 4. Secretary klik tombol login 	Secretary masuk ke halaman panel atau Homepage SuperSecretary	Sesuai
	Secretary Login dengan	<ol style="list-style-type: none"> 1. Secretary berada pada halaman login 	Secretary tidak dapat login	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
	mengisi data yang tidak valid di halaman Login	<p>aplikasi komersial PT. Cakra Agro Indonesia</p> <ol style="list-style-type: none"> Secretary memasukan username yang tidak valid pada field username Secretary memasukan password yang tidak valid pada field password Secretary klik tombol login 	karena data tidak valid dan muncul notifikasi “username atau password salah”	
	Secretary Login dengan mengosongkan field mandatory di halaman login	<ol style="list-style-type: none"> Secretary berada pada halaman login aplikasi komersial PT. Cakra Agro Indonesia Secretary mengosongkan field username dan password Secretary klik tombol login 	Secretary tidak dapat login dan terdapat notifikasi “masukan username dan password”	Sesuai
Halaman Dashboard	Secretary masuk pada halaman Dashboard	<ol style="list-style-type: none"> Secretary berada pada halaman Dashboard Secretary memeriksa tampilan Dashboard 	Secretary dapat melihat tampilan Dashboard Secretary	Sesuai
	Secretary masuk pada halaman Info Secretary di halaman Dashboard Secretary	<ol style="list-style-type: none"> Secretary berada pada halaman Info Secretary Secretary memeriksa tampilan pada halaman Info Secretary 	Secretary dapat melihat tampilan Info profil Secretary	Sesuai
	Secretary Edit data pada halaman Info Secretary	<ol style="list-style-type: none"> Secretary melakukan Edit info Secretary Secretary klik button Edit untuk mengubah info data Secretary Secretary ganti foto Secretary dan mengubah field nama Secretary, username, password, dan konfirmasi password Secretary klik button simpan 	Secretary berhasil melakukan edit data Secretary	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
	Secretary logout aplikasi	<ol style="list-style-type: none"> Secretary klik button logout pada info Secretary Secretary melakukan logout 	Secretary berhasil melakukan logout aplikasis	Sesuai
	Secretary masuk pada halaman Info Distributor di halaman Dashboard Secretary	<ol style="list-style-type: none"> Secretary berada pada halaman Info Distributor Secretary memeriksa tampilan info profil data Supervisor 	Secretary dapat melihat tampilan Info profil pada data Supervisor	Sesuai
Halaman Kelola Bonus Distributor	Secretary masuk pada halaman Kelola Bonus Distributor	<ol style="list-style-type: none"> Secretary berada pada halaman Kelola Bonus Distributor Secretary memeriksa tampilan Kelola Bonus Distributor 	Secretary dapat melihat tampilan Kelola Bonus Distributor	Sesuai
	Secretary masuk pada halaman Tambah Bonus Distributor dengan menambahkan data yang disertai foto	<ol style="list-style-type: none"> Secretary berada pada halaman Tambah Bonus Distributor Secretary menambahkan data baru untuk Bonus Distributor dengan klik button tambahkan Secretary mengisi semua field data Tambah Bonus Distributor untuk menambahkan Bonus Distributor baru Secretary melakukan upload foto Admin klik button submit 	Secretary berhasil menambahkan data Bonus Distributor baru yang disertai upload foto dan muncul notifikasi “pilih foto terlebih dahulu”	Sesuai
	Secretary masuk pada halaman Tambah Bonus Distributor dengan menambahkan data yang tidak disertai foto	<ol style="list-style-type: none"> Secretary berada pada halaman Tambah Bonus Distributor Secretary menambahkan data baru untuk Bonus Distributor dengan klik button tambahkan 	Secretary tidak berhasil menambahkan data Bonus Distributor baru karena tidak disertai upload foto dan muncul notifikasi	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		<ol style="list-style-type: none"> 3. Secretary mengisi semua field data Tambah Bonus Distributor untuk menambahkan Bonus Distributor baru 4. Secretary tidak melakukan upload foto 5. Secretary klik button submit 	“pilih foto terlebih dahulu”	
	Secretary update data pada halaman Kelola Bonus Distributor	<ol style="list-style-type: none"> 1. Secretary melakukan update data pada Kelola Bonus Distributor 2. Secretary klik image button update untuk mengubah data 3. Secretary mengganti field data Tambah Bonus Distributor yang akan diubah 4. Secretary klik Submit 	Secretary berhasil melakukan update data	Sesuai
	Secretary hapus data pada halaman Kelola Bonus Distributor	<ol style="list-style-type: none"> 1. Secretary melakukan hapus data 2. Secretary klik image button hapus dan muncul notifikasi “apakah anda yakin ingin menghapus? “, lalu klik hapus 	Secretary berhasil melakukan hapus data	Sesuai

e. Supervisor

Test Scenario	Test Case	Test Step	Result	Status
Halaman utama	Supervisor membuka aplikasi mobile	<ol style="list-style-type: none"> 1. Supervisor berada pada halaman utama aplikasi mobile 2. Supervisor memeriksa tampilan foto slide produk dan fungsinya 3. Supervisor memeriksa tampilan penjelasan singkat produk 	Supervisor dapat melihat foto slide produk dan fungsinya, penjelasa singkat produk, demplot terbaru, dan berita terbaru	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		<ol style="list-style-type: none"> 4. Supervisor memeriksa tampilan demplot terbaru 5. Supervisor memeriksa tampilan berita terbaru 		
Halaman produk kami	Supervisor membuka halaman produk kami	<ol style="list-style-type: none"> 3. Supervisor berada pada halaman produk kami 4. Supervisor memeriksa tampilan produk kami 	Supervisor dapat melihat tampilan produk kami	Sesuai
	Supervisor masuk pada halaman Insektisida	<ol style="list-style-type: none"> 1. Supervisor menekan Insektisida 2. Supervisor berada di halaman Insektisida 3. Supervisor memeriksa produk Insektisida 4. Supervisor melakukan search untuk menemukan produk Insektisida 	Supervisor dapat melihat keseluruhan produk Insektisida beserta deskripsinya dan dapat melakukan search untuk mencari produk Insektisida	Sesuai
	Supervisor masuk pada halaman Fungisida	<ol style="list-style-type: none"> 1. Supervisor menekan button Fungisida 2. Supervisor berada di halaman Fungisida 3. Supervisor memeriksa produk Fungisida 4. Supervisor melakukan search untuk menemukan produk Fungisida 	Supervisor dapat melihat keseluruhan produk fungisida beserta deskripsinya dan dapat melakukan search untuk mencari produk Fungisida	Sesuai
	Supervisor masuk pada halaman Herbisida	<ol style="list-style-type: none"> 1. Supervisor menekan button Herbisida 2. Supervisor berada di halaman Herbisida 3. Supervisor memeriksa produk Herbisida 4. Supervisor melakukan search 	Supervisor dapat melihat keseluruhan produk Herbisida beserta deskripsinya dan dapat melakukan	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		untuk menemukan produk Herbisida	search untuk mencari produk Herbisida	
	Supervisor masuk pada halaman PGR	<ol style="list-style-type: none"> Supervisor menekan button PGR Supervisor berada di halaman PGR Supervisor memeriksa produk PGR Supervisor melakukan search untuk menemukan produk PGR 	Supervisor dapat melihat keseluruhan produk PGR beserta deskripsinya dan dapat melakukan search untuk mencari produk PGR	Sesuai
Halaman Event, Halaman Inovasi, dan FAQ	Supervisor masuk pada halaman Event	<ol style="list-style-type: none"> Supervisor menekan image button aktivitas Supervisor masuk pada halaman aktivitas Supervisor menekan image button demplot Supervisor masuk pada halaman aktivitas 	Supervisor dapat melihat tampilan aktivitas dan tampilan pada demplot	Sesuai
	Supervisor masuk pada halaman Inovasi	<ol style="list-style-type: none"> Supervisor menekan image button produk baru Supervisor masuk pada halaman produk baru Supervisor menekan image button paket produk Supervisor masuk pada halaman paket produk Supervisor klik Lihat PPT untuk mengunduh paket produk 	Supervisor dapat melihat tampilan produk baru, Supervisor dapat melihat tampilan pada paket produk, dan user dapat melihat isi paket produk dengan mengunduh PPT	Sesuai
	Supervisor masuk pada halaman FAQ	<ol style="list-style-type: none"> Supervisor menekan button FAQ Supervisor menekan pertanyaan dan sudah 	Supervisor dapat melihat pertanyaan yang sudah tersedia dan terdapat	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		tersedia jawabannya pada FAQ	jawabannya pada FAQ	
	Supervisor masuk pada halaman Pertanyaan Lain (QnA)	1. Supervisor mereview pertanyaan-pertanyaan yang diajukan oleh customer lain	Supervisor dapat melihat review-an yang sudah diajukan oleh customer lain	Sesuai
	Supervisor menambahkan pertanyaan	<ol style="list-style-type: none"> Supervisor klik Tambahkan Pertanyaan untuk mengajukan pertanyaan Supervisor menambahkan pertanyaan dengan mengisi semua field pada kolom pertanyaan dan disertai upload foto Supervisor menambahkan pertanyaan dengan mengisi semua field pada kolom pertanyaan dengan tidak disertai upload foto 	Supervisor dapat mengajukan pertanyaan disertai upload foto ataupun tidak disertai upload foto	Sesuai
Halaman Tentang Kami	Supervisor masuk pada halaman Tentang kami	<ol style="list-style-type: none"> Supervisor berada pada halaman Tentang Kami Supervisor memeriksa tampilan profil, Visi dan Misi, serta Alamat Supervisor klik image button pada sosial media untuk facebook Supervisor klik image button pada sosial media untuk Instagram Supervisor klik image button pada sosial media untuk Instagram 	Supervisor dapat melihat tampilan profil, visi dan misi, Alamat, dan pada sosial media langsung bisa terhubung pada facebook, Instagram, dan youtube	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		6. Supervisor klik image button pada sosial media untuk youtube		
Supervisor Login dari halaman Login	Supervisor Login dengan mengisi data yang valid di halaman Login	<ol style="list-style-type: none"> Supervisor berada pada halaman login aplikasi komersial PT. Cakra Agro Indonesia Supervisor memasukan username yang valid pada field username Supervisor memasukan password yang valid pada field password Supervisor klik tombol login 	Supervisor masuk ke halaman panel atau Homepage SuperSupervisor	Sesuai
	Supervisor Login dengan mengisi data yang tidak valid di halaman Login	<ol style="list-style-type: none"> Supervisor berada pada halaman login aplikasi komersial PT. Cakra Agro Indonesia Supervisor memasukan username yang tidak valid pada field username Supervisor memasukan password yang tidak valid pada field password Supervisor klik tombol login 	Supervisor tidak dapat login karena data tidak valid dan muncul notifikasi “username atau password salah”	Sesuai
	Supervisor Login dengan mengosongkan field mandatory di halaman login	<ol style="list-style-type: none"> Supervisor berada pada halaman login aplikasi komersial PT. Cakra Agro Indonesia Supervisor mengosongkan field username dan password Supervisor klik tombol login 	Supervisor tidak dapat login dan terdapat notifikasi “masukan username dan password”	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
Halaman Dashboard	Supervisor masuk pada halaman Dashboard	<ol style="list-style-type: none"> Supervisor berada pada halaman Dashboard Supervisor memeriksa tampilan Dashboard 	Supervisor dapat melihat tampilan Dashboard Supervisor	Sesuai
	Supervisor masuk pada halaman Info Supervisor di halaman Dashboard Supervisor	<ol style="list-style-type: none"> Supervisor berada pada halaman Info Supervisor Supervisor memeriksa tampilan pada halaman Info Supervisor 	Supervisor dapat melihat tampilan Info profil Supervisor	Sesuai
	Supervisor Edit data pada halaman Info Supervisor	<ol style="list-style-type: none"> Supervisor melakukan update info Supervisor Supervisor klik button Edit untuk mengubah info data Supervisor Supervisor ganti foto Supervisor dan mengubah field data Supervisor, username, password, dan konfirmasi password Supervisor klik button simpan 	Supervisor berhasil melakukan edit data Supervisor	Sesuai
	Supervisor logout aplikasi	<ol style="list-style-type: none"> Supervisor klik button logout pada info Supervisor Supervisor melakukan logout 	Supervisor berhasil melakukan logout aplikasi	Sesuai
	Supervisor masuk pada halaman Info Demonstrator di halaman Dashboard Supervisor	<ol style="list-style-type: none"> Supervisor berada pada halaman Info Demonstrator Supervisor memeriksa tampilan info profil data Demonstrator 	Supervisor dapat melihat tampilan Info profil pada data Demonstrator	Sesuai
Halaman Kelola Supervisor	Supervisor masuk pada halaman Kelola Supervisor	<ol style="list-style-type: none"> Supervisor berada pada halaman Kelola Supervisor Supervisor memeriksa tampilan 	Supervisor dapat melihat tampilan Kelola Supervisor	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		pada Kelola Supervisor		
	Supervisor masuk pada halaman Review Report Demonstrator di Kelola Supervisor	<ol style="list-style-type: none"> Supervisor berada pada halaman Daftar Report Demonstrator Supervisor dapat melihat Daftar Report Demonstrator dengan status hijau “disetujui” dan merah “belum disetujui” 	Supervisor dapat melihat tampilan pada Daftar Report Demonstrator	Sesuai
	Supervisor masuk pada halaman Kelola Aktivitas Supervisor	<ol style="list-style-type: none"> Supervisor berada pada halaman Kelola Supervisor Supervisor memeriksa tampilan Daftar Aktivitas 	Supervisor dapat melihat tampilan Daftar Supervisor	Sesuai
	Supervisor masuk pada halaman Daftar Aktivitas Baru dengan menambahkan data yang disertai foto	<ol style="list-style-type: none"> Supervisor berada pada halaman Daftar Aktivitas Baru Supervisor menambahkan data baru untuk Aktivitas Baru dengan klik button tambahkan Supervisor mengisi semua field data Daftar Aktivitas Baru untuk menambahkan data Aktivitas Baru Supervisor melakukan upload foto Supervisor klik button submit 	Supervisor berhasil menambahkan Daftar Aktivitas Baru baru yang disertai upload foto dan muncul notifikasi “pilih foto terlebih dahulu”	Sesuai
	Supervisor masuk pada halaman Daftar Aktivitas Baru dengan menambahkan	<ol style="list-style-type: none"> Supervisor berada pada halaman Daftar Aktivitas Baru Supervisor menambahkan data baru untuk Bonus Distributor dengan 	Supervisor tidak berhasil menambahkan data Bonus Distributor baru karena tidak disertai upload	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
	data yang tidak disertai foto	klik button tambahkan 3. Supervisor mengisi semua field data Daftar Aktivitas Baru untuk menambahkan data Aktivitas Baru 4. Supervisor tidak melakukan upload foto 5. Supervisor klik button submit	foto dan muncul notifikasi “pilih foto terlebih dahulu”	
	Supervisor update data pada halaman Kelola Bonus Distributor	1. Supervisor melakukan update data pada Kelola Bonus Distributor 2. Supervisor klik image button update untuk mengubah data 3. Supervisor mengganti field data Tambah Bonus Distributor yang akan diubah 4. Supervisor klik Submit	Supervisor berhasil melakukan update data	Sesuai
	Supervisor hapus data pada halaman Kelola Bonus Distributor	1. Supervisor melakukan hapus data 2. Supervisor klik image button hapus dan muncul notifikasi “apakah anda yakin ingin menghapus? “, lalu klik hapus	Supervisor berhasil melakukan hapus data	Sesuai

f. Demonstrator

Test Scenario	Test Case	Test Step	Result	Status
Halaman utama	Demonstrator membuka aplikasi mobile	1. Demonstrator berada pada halaman utama aplikasi mobile 2. Demonstrator memeriksa tampilam foto slide	Demonstrator dapat melihat foto slide produk dan fungsinya, penjelasa singkat produk, demplot	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		<ul style="list-style-type: none"> produk dan fungsinya 3. Demonstrator memeriksa tampilan penjelasan singkat produk 4. Demonstrator memeriksa tampilan demplot terbaru 5. Demonstrator memeriksa tampilan berita terbaru 	terbaru, dan berita terbaru	
Halaman produk kami	Demonstrator membuka halaman produk kami	<ul style="list-style-type: none"> 1. Demonstrator berada pada halaman produk kami 2. Demonstrator memeriksa tampilan produk kami 	Demonstrator dapat melihat tampilan produk kami	Sesuai
	Demonstrator masuk pada halaman Insektisida	<ul style="list-style-type: none"> 1. Demonstrator menekan Insektisida 2. Demonstrator berada di halaman Insektisida 3. Demonstrator memeriksa produk Insektisida 4. Demonstrator melakukan search untuk menemukan produk Insektisida 	Demonstrator dapat melihat keseluruhan produk Insektisida beserta deskripsinya dan dapat melakukan search untuk mencari produk Insektisida	Sesuai
	Demonstrator masuk pada halaman Fungisida	<ul style="list-style-type: none"> 1. Demonstrator menekan button Fungisida 2. Demonstrator berada di halaman Fungisida 3. Demonstrator memeriksa produk Fungisida 	Demonstrator dapat melihat keseluruhan produk fungisida beserta deskripsinya dan dapat melakukan search untuk	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		4. Demonstrator melakukan search untuk menemukan produk Fungisida	mencari produk Fungisida	
	Demonstrator masuk pada halaman Herbisida	<ol style="list-style-type: none"> 1. Demonstrator menekan button Herbisida 2. Demonstrator berada di halaman Herbisida 3. Demonstrator memeriksa produk Herbisida 4. Demonstrator melakukan search untuk menemukan produk Herbisida 	Demonstrator dapat melihat keseluruhan produk Herbisida beserta deskripsinya dan dapat melakukan search untuk mencari produk Herbisida	Sesuai
	Demonstrator masuk pada halaman PGR	<ol style="list-style-type: none"> 1. Demonstrator menekan button PGR 2. Demonstrator berada di halaman PGR 3. Demonstrator memeriksa produk PGR 4. Demonstrator melakukan search untuk menemukan produk PGR 	Demonstrator dapat melihat keseluruhan produk PGR beserta deskripsinya dan dapat melakukan search untuk mencari produk PGR	Sesuai
Halaman Event, Halaman Inovasi, dan FAQ	Demonstrator masuk pada halaman Event	<ol style="list-style-type: none"> 1. Demonstrator menekan image button aktivitas 2. Demonstrator masuk pada halaman aktivitas 3. Demonstrator menekan image button demplot 4. Demonstrator masuk pada halaman aktivitas 	Demonstrator dapat melihat tampilan aktivitas dan tampilan pada demplot	Sesuai
	Demonstrator masuk pada halaman Inovasi	<ol style="list-style-type: none"> 1. Demonstrator menekan image button produk baru 2. Demonstrator masuk pada 	Demonstrator dapat melihat tampilan produk baru, Demonstrator dapat melihat	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		<p>halaman produk baru</p> <ol style="list-style-type: none"> 3. Demonstrator menekan image button paket produk 4. Demonstrator masuk pada halaman paket produk 5. Demonstrator klik Lihat PPT untuk mengunduh paket produk 	tampilan pada paket produk, dan user dapat melihat isi paket produk dengan mengunduh PPT	
	Demonstrator masuk pada halaman FAQ	<ol style="list-style-type: none"> 1. Demonstrator menekan button FAQ 2. Demonstrator menekan pertanyaan dan sudah tersedia jawabannya pada FAQ 	Demonstrator dapat melihat pertanyaan yang sudah tersedia dan terdapat jawabannya pada FAQ	Sesuai
	Demonstrator masuk pada halaman Pertanyaan Lain (QnA)	<ol style="list-style-type: none"> 1. Demonstrator mereview pertanyaan-pertanyaan yang diajukan oleh customer lain 	Demonstrator dapat melihat review-an yang sudah diajukan oleh customer lain	Sesuai
	Demonstrator menambahkan pertanyaan	<ol style="list-style-type: none"> 1. Demonstrator klik Tambahkan Pertanyaan untuk mengajukan pertanyaan 2. Demonstrator menambahkan pertanyaan dengan mengisi semua field pada kolom pertanyaan dan disertai upload foto 3. Demonstrator menambahkan pertanyaan dengan mengisi semua field pada kolom pertanyaan dengan 	Demonstrator dapat mengajukan pertanyaan disertai upload foto ataupun tidak disertai upload foto	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		tidak disertai upload foto		
Halaman Tentang Kami	Demonstrator masuk pada halaman Tentang kami	<ol style="list-style-type: none"> 1. Demonstrator berada pada halaman Tentang Kami 2. Demonstrator memeriksa tampilan profil, Visi dan Misi, serta Alamat 3. Demonstrator klik image button pada sosial media untuk facebook 4. Demonstrator klik image button pada sosial media untuk Instagram 5. Demonstrator klik image button pada sosial media untuk Instagram 6. Demonstrator klik image button pada sosial media untuk youtube 	Demonstrator dapat melihat tampilan profil, visi dan misi, Alamat, dan pada sosial media langsung bisa terhubung pada facebook, Instagram, dan youtube	Sesuai
Demonstrator Login dari halaman Login	Demonstrator Login dengan mengisi data yang valid di halaman Login	<ol style="list-style-type: none"> 1. Demonstrator berada pada halaman login aplikasi komersial PT. Cakra Agro Indonesia 2. Demonstrator memasukan username yang valid pada field username 3. Demonstrator memasukan password yang valid pada field password 4. Demonstrator klik tombol login 	Demonstrator masuk ke halaman panel atau Homepage SuperDemonstrator	Sesuai
	Demonstrator Login dengan	1. Demonstrator berada pada	Demonstrator tidak dapat login karena	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
	mengisi data yang tidak valid di halaman Login	<p>halaman login aplikasi komersial PT. Cakra Agro Indonesia</p> <p>2. Demonstrator memasukan username yang tidak valid pada field username</p> <p>3. Demonstrator memasukan password yang tidak valid pada field password</p> <p>4. Demonstrator klik tombol login</p>	data tidak valid dan muncul notifikasi “username atau password salah”	
	Demonstrator Login dengan mengosongkan field mandatory di halaman login	<p>1. Demonstrator berada pada halaman login aplikasi komersial PT. Cakra Agro Indonesia</p> <p>2. Demonstrator mengosongkan field username dan password</p> <p>3. Demonstrator klik tombol login</p>	Demonstrator tidak dapat login dan terdapat notifikasi “masukan username dan password”	Sesuai
Halaman Dashboard	Demonstrator masuk pada halaman Dashboard	<p>3. Demonstrator berada pada halaman Dashboard</p> <p>4. Demonstrator memeriksa tampilan Dashboard</p>	Demonstrator dapat melihat tampilan Dashboard Demonstrator	Sesuai
	Demonstrator masuk pada halaman Info Demonstrator di halaman Dashboard Demonstrator	<p>1. Demonstrator berada pada halaman Info Demonstrator</p> <p>2. Demonstrator memeriksa tampilan pada halaman Info Demonstrator</p>	Demonstrator dapat melihat tampilan Info profil Demonstrator	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
	Demonstrator Edit data pada halaman Info Demonstrator	<ol style="list-style-type: none"> 1. Demonstrator melakukan edit info Demonstrator 2. Demonstrator klik button Edit untuk mengubah info data Demonstrator 3. Demonstrator ganti foto Demonstrator dan mengubah field data Demonstrator, username, password, dan konfirmasi password 4. Demonstrator klik button simpan 	Demonstrator berhasil melakukan edit data Demonstrator	Sesuai
	Demonstrator logout aplikasi	<ol style="list-style-type: none"> 1. Demonstrator klik button logout pada info Demonstrator 2. Demonstrator melakukan logout 	Demonstrator berhasil melakukan logout aplikasi	Sesuai
Halaman Kelola Report Demonstrator	Demonstrator masuk pada halaman Review Report Demonstrator di Kelola Demonstrator	<ol style="list-style-type: none"> 1. Demonstrator berada pada halaman Daftar Report Demonstrator 2. Demonstrator dapat melihat Daftar Report Demonstrator dengan status hijau “disetujui” dan merah “belum disetujui” 	Demonstrator dapat melihat tampilan pada Daftar Report Demonstrator	Sesuai
	Demonstrator masuk pada halaman Daftar Report Baru dengan menambahkan data yang disertai foto	<ol style="list-style-type: none"> 1. Demonstrator berada pada halaman Daftar Report Baru 2. Demonstrator menambahkan data baru untuk Report Baru dengan klik button tambahkan 3. Demonstrator mengisi semua field 	Demonstrator berhasil menambahkan Daftar Report Baru baru yang disertai upload foto dan muncul notifikasi “pilih foto terlebih dahulu”	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		<p>data Daftar Report Baru untuk menambahkan data Report Baru</p> <ol style="list-style-type: none"> 4. Demonstrator melakukan upload foto 5. Demonstrator klik button submit 		
	Demonstrator masuk pada halaman Daftar Report Baru dengan menambahkan data yang tidak disertai foto	<ol style="list-style-type: none"> 1. Demonstrator berada pada halaman Daftar Report Baru 2. Demonstrator menambahkan data baru untuk Report Baru dengan klik button tambahkan 3. Demonstrator mengisi semua field data Daftar Report Baru untuk menambahkan data Report Baru 4. Demonstrator tidak melakukan upload foto 5. Demonstrator klik button submit 	Demonstrator tidak berhasil menambahkan Daftar Report Baru baru karena tidak disertai upload foto dan muncul notifikasi “pilih foto terlebih dahulu”	Sesuai
	Demonstrator update data pada halaman Daftar Report	<ol style="list-style-type: none"> 1. Demonstrator melakukan update data pada Daftar Report 2. Demonstrator klik image button update untuk mengubah data 3. Demonstrator mengganti field data Daftar Report Baru yang akan diubah 4. Demonstrator klik Submit 	Demonstrator berhasil melakukan update data	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
	Demonstrator hapus data pada halaman Daftar Report	<ol style="list-style-type: none"> 1. Demonstrator melakukan hapus data 2. Demonstrator klik image button hapus dan muncul notifikasi “apakah anda yakin ingin menghapus? “, lalu klik hapus 	Demonstrator berhasil melakukan hapus data	Sesuai

g. Distributor

Test Scenario	Test Case	Test Step	Result	Status
Halaman utama	Distributor membuka aplikasi mobile	<ol style="list-style-type: none"> 1. Distributor berada pada halaman utama aplikasi mobile 2. Distributor memeriksa tampilam foto slide produk dan fungsinya 3. Distributor memeriksa tampilan penjelasan singkat produk 4. Distributor memeriksa tampilan demplot terbaru 5. Distributor memeriksa tampilan berita terbaru 	Distributor dapat melihat foto slide produk dan fungsinya, penjelasa singkat produk, demplot terbaru, dan berita terbaru	Sesuai
Halaman produk kami	Distributor membuka halaman produk kami	<ol style="list-style-type: none"> 1. Distributor berada pada halaman produk kami 2. Distributor memeriksa tampilan produk kami 	Distributor dapat melihat tampilan produk kami	Sesuai
	Distributor masuk pada	<ol style="list-style-type: none"> 1. Distributor menekan Insektisida 	Distributor dapat melihat keseluruhan	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
	halaman Insektisida	<ol style="list-style-type: none"> 2. Distributor berada di halaman Insektisida 3. Distributor memeriksa produk Insektisida 4. Distributor melakukan search untuk menemukan produk Insektisida 	produk Insektisida beserta deskripsinya dan dapat melakukan search untuk mencari produk Insektisida	
	Distributor masuk pada halaman Fungisida	<ol style="list-style-type: none"> 1. Distributor menekan button Fungisida 2. Distributor berada di halaman Fungisida 3. Distributor memeriksa produk Fungisida 4. Distributor melakukan search untuk menemukan produk Fungisida 	Distributor dapat melihat keseluruhan produk fungisida beserta deskripsinya dan dapat melakukan search untuk mencari produk Fungisida	Sesuai
	Distributor masuk pada halaman Herbisida	<ol style="list-style-type: none"> 1. Distributor menekan button Herbisida 2. Distributor berada di halaman Herbisida 3. Distributor memeriksa produk Herbisida 4. Distributor melakukan search untuk menemukan produk Herbisida 	Distributor dapat melihat keseluruhan produk Herbisida beserta deskripsinya dan dapat melakukan search untuk mencari produk Herbisida	Sesuai
	Distributor masuk pada halaman PGR	<ol style="list-style-type: none"> 1. Distributor menekan button PGR 2. Distributor berada di halaman PGR 3. Distributor memeriksa produk PGR 4. Distributor melakukan search 	Distributor dapat melihat keseluruhan produk PGR beserta deskripsinya dan dapat melakukan search untuk mencari produk PGR	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		untuk menemukan produk PGR		
Halaman Event, Halaman Inovasi, dan FAQ	Distributor masuk pada halaman Event	<ol style="list-style-type: none"> 1. Distributor menekan image button aktivitas 2. Distributor masuk pada halaman aktivitas 3. Distributor menekan image button demplot 4. Distributor masuk pada halaman aktivitas 	Distributor dapat melihat tampilan aktivitas dan tampilan pada demplot	Sesuai
	Distributor masuk pada halaman Inovasi	<ol style="list-style-type: none"> 1. Distributor menekan image button produk baru 2. Distributor masuk pada halaman produk baru 3. Distributor menekan image button paket produk 4. Distributor masuk pada halaman paket produk 5. Distributor klik Lihat PPT untuk mengunduh paket produk 	Distributor dapat melihat tampilan produk baru, Distributor dapat melihat tampilan pada paket produk, dan user dapat melihat isi paket produk dengan mengunduh PPT	Sesuai
	Distributor masuk pada halaman FAQ	<ol style="list-style-type: none"> 1. Distributor menekan button FAQ 2. Distributor menekan pertanyaan dan sudah tersedia jawabannya pada FAQ 	Distributor dapat melihat pertanyaan yang sudah tersedia dan terdapat jawabannya pada FAQ	Sesuai
	Distributor masuk pada halaman Pertanyaan Lain (QnA)	<ol style="list-style-type: none"> 1. Distributor mereview pertanyaan-pertanyaan yang diajukan oleh customer lain 	Distributor dapat melihat review-an yang sudah diajukan oleh customer lain	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
	Distributor menambahkan pertanyaan	<ol style="list-style-type: none"> 1. Distributor klik Tambahkan Pertanyaan untuk mengajukan pertanyaan 2. Distributor menambahkan pertanyaan dengan mengisi semua field pada kolom pertanyaan dan disertai upload foto 3. Distributor menambahkan pertanyaan dengan mengisi semua field pada kolom pertanyaan dengan tidak disertai upload foto 	Distributor dapat mengajukan pertanyaan disertai upload foto ataupun tidak disertai upload foto	Sesuai
Halaman Tentang Kami	Distributor masuk pada halaman Tentang kami	<ol style="list-style-type: none"> 1. Distributor berada pada halaman Tentang Kami 2. Distributor memeriksa tampilan profil, Visi dan Misi, serta Alamat 3. Distributor klik image button pada sosial media untuk facebook 4. Distributor klik image button pada sosial media untuk Instagram 5. Distributor klik image button pada sosial media untuk Instagram 6. Distributor klik image button pada sosial media untuk youtube 	Distributor dapat melihat tampilan profil, visi dan misi, Alamat, dan pada sosial media langsung bisa terhubung pada facebook, Instagram, dan youtube	Sesuai
Distributor Login dari halaman Login	Distributor Login dengan mengisi data	<ol style="list-style-type: none"> 1. Distributor berada pada halaman login aplikasi komersial 	Distributor masuk ke halaman panel	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
	yang valid di halaman Login	PT. Cakra Agro Indonesia 2. Distributor memasukan username yang valid pada field username 3. Distributor memasukan password yang valid pada field password 4. Distributor klik tombol login	atau Homepage SuperDistributor	
	Distributor Login dengan mengisi data yang tidak valid di halaman Login	1. Distributor berada pada halaman login aplikasi komersial PT. Cakra Agro Indonesia 2. Distributor memasukan username yang tidak valid pada field username 3. Distributor memasukan password yang tidak valid pada field password 4. Distributor klik tombol login	Distributor tidak dapat login karena data tidak valid dan muncul notifikasi "username atau password salah"	Sesuai
	Distributor Login dengan mengosongkan field mandatory di halaman login	1. Distributor berada pada halaman login aplikasi komersial PT. Cakra Agro Indonesia 2. Distributor mengosongkan field username dan password 3. Distributor klik tombol login	Distributor tidak dapat login dan terdapat notifikasi "masukan username dan password"	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
Halaman Dashboard	Distributor masuk pada halaman Dashboard	<ol style="list-style-type: none"> 1. Distributor berada pada halaman Dashboard 2. Distributor memeriksa tampilan Dashboard 	Distributor dapat melihat tampilan Dashboard Distributor	Sesuai
	Distributor masuk pada halaman Info Distributor di halaman Dashboard Distributor	<ol style="list-style-type: none"> 3. Distributor berada pada halaman Info Distributor 4. Distributor memeriksa tampilan pada halaman Info Distributor 	Distributor dapat melihat tampilan Info profil Distributor	Sesuai
	Distributor Edit data pada halaman Info Distributor	<ol style="list-style-type: none"> 1. Distributor melakukan edit info Distributor 2. Distributor klik button Edit untuk mengubah info data Distributor 3. Distributor ganti foto Distributor dan mengubah field data Distributor, username, password, dan konfirmasi password 4. Distributor klik button simpan 	Distributor berhasil melakukan edit data Distributor	Sesuai
	Distributor logout aplikasi	<ol style="list-style-type: none"> 1. Distributor klik button logout pada info Distributor 2. Distributor melakukan logout 	Distributor berhasil melakukan logout aplikasi	Sesuai
Halaman Kelola Bonus Distributor	Distributor masuk pada halaman Kelola Bonus Distributor	<ol style="list-style-type: none"> 1. Distributor berada pada halaman Kelola Bonus Distributor 2. Distributor memeriksa Bonus yang diperoleh Distributor 	Distributor dapat melihat Bonus yang diperoleh Distributor	Sesuai

h. Customer (user)

Test Scenario	Test Case	Test Step	Result	Status
Halaman utama	User membuka aplikasi mobile	<ol style="list-style-type: none"> 1. User berada pada halaman utama aplikasi mobile 2. User memeriksa tampilam foto slide produk dan fungsinya 3. User memeriksa tampilan penjelasan singkat produk 4. User memeriksa tampilan demplot terbaru 5. User memeriksa tampilan berita terbaru 	User dapat melihat foto slide produk dan fungsinya, penjelasa singkat produk, demplot terbaru, dan berita terbaru	Sesuai
Halaman produk kami	User membuka halaman produk kami	<ol style="list-style-type: none"> 1. User berada pada halaman produk kami 2. User memeriksa tampilan produk kami 	User dapat melihat tampilan produk kami	Sesuai
	User masuk pada halaman Insektisida	<ol style="list-style-type: none"> 1. User menekan Insektisida 2. User berada di halaman Insektisida 3. User memeriksa produk Insektisida 4. User melakukan search untuk menemukan produk Insektisida 	User dapat melihat keseluruhan produk Insektisida beserta deskripsinya dan dapat melakukan search untuk mencari produk Insektisida	Sesuai
	User masuk pada halaman Fungisida	<ol style="list-style-type: none"> 1. User menekan button Fungisida 2. User berada di halaman Fungisida 3. User memeriksa produk Fungisida 4. User melakukan search untuk menemukan produk Fungisida 	User dapat melihat keseluruhan produk fungisida beserta deskripsinya dan dapat melakukan search untuk	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
			mencari produk Fungisida	
	User masuk pada halaman Herbisida	<ol style="list-style-type: none"> 1. User menekan button Herbisida 2. User berada di halaman Herbisida 3. User memeriksa produk Herbisida 4. User melakukan search untuk menemukan produk Herbisida 	User dapat melihat keseluruhan produk Herbisida beserta deskripsinya dan dapat melakukan search untuk mencari produk Herbisida	Sesuai
	User masuk pada halaman PGR	<ol style="list-style-type: none"> 1. User menekan button PGR 2. User berada di halaman PGR 3. User memeriksa produk PGR 4. User melakukan search untuk menemukan produk PGR 	User dapat melihat keseluruhan produk PGR beserta deskripsinya dan dapat melakukan search untuk mencari produk PGR	Sesuai
Halaman Event, Halaman Inovasi, dan FAQ	User masuk pada halaman Event	<ol style="list-style-type: none"> 1. User menekan image button aktivitas 2. User masuk pada halaman aktivitas 3. User menekan image button demplot 4. User masuk pada halaman aktivitas 	User dapat melihat tampilan aktivitas dan tampilan pada demplot	Sesuai
	User masuk pada halaman Inovasi	<ol style="list-style-type: none"> 1. User menekan image button produk baru 2. User masuk pada halaman produk baru 3. User menekan image button paket produk 4. User masuk pada halaman paket produk 5. User klik Lihat PPT untuk mengunduh paket produk 	User dapat melihat tampilan pada paket produk, dan dapat melihat isi paket produk dengan mengunduh PPT	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
	User masuk pada halaman FAQ	<ol style="list-style-type: none"> 1. User menekan button FAQ 2. User menekan pertanyaan dan sudah tersedia jawabannya pada FAQ 	User dapat melihat pertanyaan yang sudah tersedia dan terdapat jawabannya pada FAQ	Sesuai
	User masuk pada halaman Pertanyaan Lain (QnA)	<ol style="list-style-type: none"> 1. User mereview pertanyaan-pertanyaan yang diajukan oleh customer lain 	User dapat melihat review-an yang sudah diajukan oleh customer lain	Sesuai
	User menambahkan pertanyaan	<ol style="list-style-type: none"> 1. User klik Tambahkan Pertanyaan untuk mengajukan pertanyaan 2. User menambahkan pertanyaan dengan mengisi semua field pada kolom pertanyaan dan disertai upload foto 3. User menambahkan pertanyaan dengan mengisi semua field pada kolom pertanyaan dengan tidak disertai upload foto 	User dapat mengajukan pertanyaan disertai upload foto ataupun tidak disertai upload foto	Sesuai
Halaman Tentang Kami	User masuk pada halaman Tentang kami	<ol style="list-style-type: none"> 1. User berada pada halaman Tentang Kami 2. User memeriksa tampilan profil, Visi dan Misi, serta Alamat 3. User klik image button pada sosial media untuk facebook 4. User klik image button pada sosial media untuk Instagram 5. User klik image button pada sosial 	User dapat melihat tampilan profil, visi dan misi, Alamat, dan pada sosial media langsung bisa terhubung pada facebook, Instagram, dan youtube	Sesuai

Test Scenario	Test Case	Test Step	Result	Status
		<p>media untuk Instagram</p> <p>6. User klik image button pada sosial media untuk youtube</p>		

Pengujian Kelayakan Aplikasi Komersial PT. Cakra Agro Indonesia



a. Customer









b. Staff

The image shows a dashboard with a table. The table has multiple rows and columns, with a light blue header and light blue body. The dashboard has a light green background and a dark blue header.







Pada kuesioner pengujian kelayakan aplikasi untuk customer dan staff dapat diambil kesimpulannya yaitu berhasil atau sudah layak untuk digunakan, karena rata-rata pada jawaban kuesioner tersebut terletak pada angka 3 dan 4 yaitu setuju dan sangat setuju.

ix. **Panduan Penggunaan Aplikasi**

Petunjuk Penggunaan Aplikasi (User Manual)



Pengembangan Aplikasi

PT. Cakra Agro Indonesia

Versi 1.0

Daftar isi

Daftar isi	621
1. Pendahuluan	622
1.1 Tentang Aplikasi	622
1.2 Pengguna Aplikasi	622
1.3 Perangkat yang dibutuhkan	622
2. Petunjuk Penggunaan	623
2.0 Customer (umum)	623
2.1 Superadmin	636
2.2 Admin	Error! Bookmark not defined.
2.3 Manager	738
2.4 Supervisor	750
2.5 Secretary	760
2.6 Demonstrator	762
2.7 Distributor	775

1. Pendahuluan

1.1 Tentang Aplikasi

Aplikasi PT. Cakra Agro Indonesia adalah aplikasi komersial yang berisi katalog dan sistem manajemen perusahaan yang dimiliki oleh PT. Cakra Agro Indonesia, Aplikasi ini memanfaatkan saluran API Database cakraagroin.com, Fitur yang dimiliki aplikasi ini antara lain :

- Fitur Katalog Produk
Menampilkan produk-produk yang dimiliki perusahaan berdasarkan jenis dan merknya.
- Fitur Berita
Menampilkan berita terbaru dari perusahaan.
- Fitur FAQ
Menampilkan pertanyaan dan jawaban yang umum ditanyakan pada perusahaan baik terkait perusahaan itu sendiri ataupun produknya .
- Fitur QNA
Mengizinkan pengguna untuk bertanya secara langsung kepada perusahaan.
- Fitur Profil Perusahaan
Menampilkan Profil yang dimiliki perusahaan yaitu deskripsi, visi, misi, alamat, sosial media, dan juga informasi lainnya.

1.2 Pengguna Aplikasi

Pengguna aplikasi ini secara garis besar dibagi menjadi 3 jenis pengguna berdasarkan yaitu Admin, Master (Staff), dan juga public (Customer).

Admin dibedakan menjadi dua level yaitu Superadmin, dan Admin berdasarkan login yang dilakukan.

Master (Staff) dibedakan menjadi 5 level yaitu Manager, Supervisor, Secretary, Demonstrator, dan Distributor berdasarkan login yang dilakukan.

Public (Customer) adalah jenis pengguna yang tidak perlu login atau akses umum dari aplikasi PT. Cakra Agro Indonesia.

1.3 Perangkat yang dibutuhkan

Perangkat Lunak yang dibutuhkan :

1. Aplikasi PT. Cakra Agro Indonesia

Perangkat Keras yang dibutuhkan :

1. Smartphone

2. Petunjuk Penggunaan

2.0 Customer (umum)

Customer yang ingin mengakses aplikasi PT. Cakra Agro Indonesia bisa mengunduh aplikasi tersebut di playstore ataupun toko aplikasi lainnya seperti Aptoide, setelah diunduh dan dijalankan maka tampilan dari aplikasinya adalah seperti berikut



Gambar diatas menampilkan halaman beranda, terdapat produk beranda yang bisa di slide untuk tiap produknya, dan juga demplot terbaru yang bisa dislide untuk demplot terbaru lainnya. Pada panel paling bawah/panel dasar merupakan panel navigasi yang menunjukan halaman yang sedang diakses saat ini, mulai dari panel paling kiri ke kanan yaitu Beranda, Produk, Inovasi, dan lainnya.

Kemudian jika pengguna scroll kebawah maka tampilan berikutnya/dibawahnya adalah tampilan berita



geser kebawah
untuk melihat
berita terbaru
lainnya

Tampilan berita diatas merupakan tampilan *scroll* atau gulir, pengguna bisa gulir kebawah untuk menampilkan berita-berita terbaru lainnya.

Berikutnya jika pengguna memilih tombol produk (tombol bergambar kubus), maka akan diarahkan ke halaman produk dengan tampilan seperti dibawah



Dari tampilan diatas, pengguna bisa memilih salah satu dari 4 jenis produk diatas, jika sudah dipilih maka tampilan akan seperti berikut



dari halaman tersebut akan ditampilkan produk-produk dari jenis yang dipilih, untuk menemukan produk yang dicari pengguna bisa menggulir kebawah atau melakukan pencarian menggunakan kolom pencarian atau kolom *search*.



Untuk melakukan pencarian, pengguna cukup memasukkan nama produk yang dicari maka secara otomatis akan ditampilkan.

Berikutnya adalah halaman event dan inovasi yang bisa diakses jika pengguna memilih tombol bergambar buku dibagian bawah



Pada halaman event dan inovasi, terdapat 5 tombol yang masing-masing mengarahkan ke halaman yang berbeda, tombol aktivitas akan mengarahkan pengguna ke halaman aktivitas,

tombol demplot akan mengarahkan pengguna ke halaman demplot seperti tampilan dibawah



Pengguna bisa gulir kebawah untuk melihat demplot lainnya.

Berikutnya tombol produk baru akan menampilkan produk terbaru yang dimiliki perusahaan seperti tampilan dibawah



Kemudian tombol paket produk akan menampilkan halaman paket produk seperti tampilan dibawah

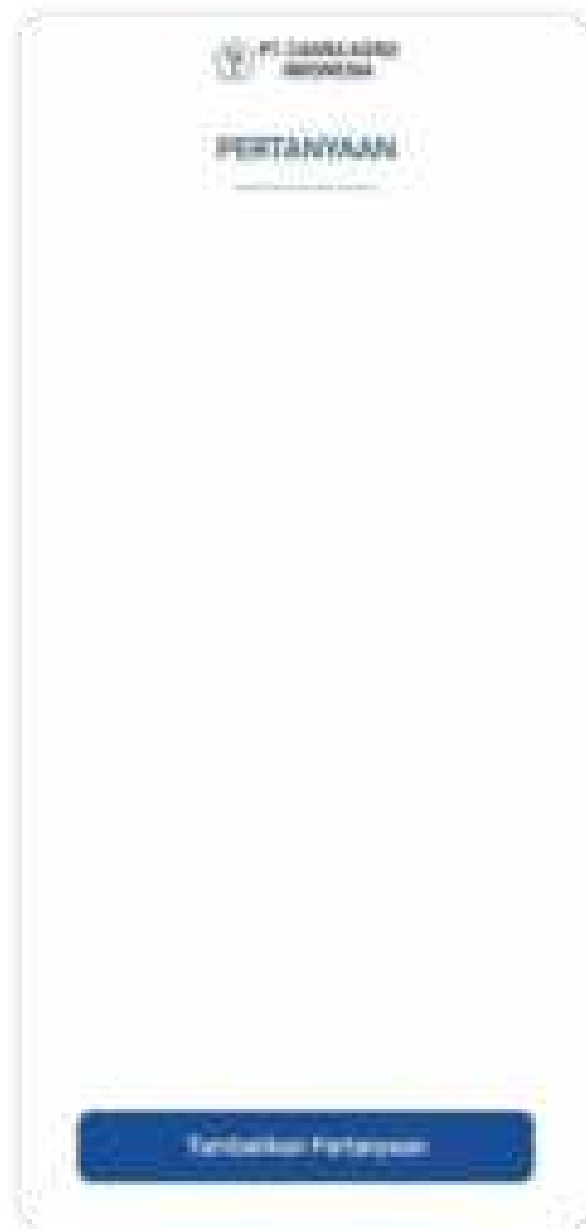


Pada halaman paket produk, pengguna bisa mengunduh PPT atau Presentasi *PowerPoint* dengan menekan tombol Lihat PPT.

Berikutnya dari halaman event dan inovasi terdapat tombol kelima yaitu FAQ, yang akan menampilkan halaman FAQ seperti dibawah ini



Pada tampilan faq berisi sejumlah pertanyaan-pertanyaan yang umum ditanyakan ke Cakra Agro Indonesia, berikutnya jika pengguna memiliki pertanyaan lainnya yang ingin diajukan maka bisa memilih tombol pertanyaan lain



Halaman diatas akan menampilkan pertanyaan-pertanyaan yang diajukan oleh pelanggan-pelanggan Cakra Agro Indonesia.

berikutnya jika ingin mengajukan pertanyaan maka bisa melalui tambahkan pertanyaan, maka berikutnya akan muncul jendela seperti dibawah

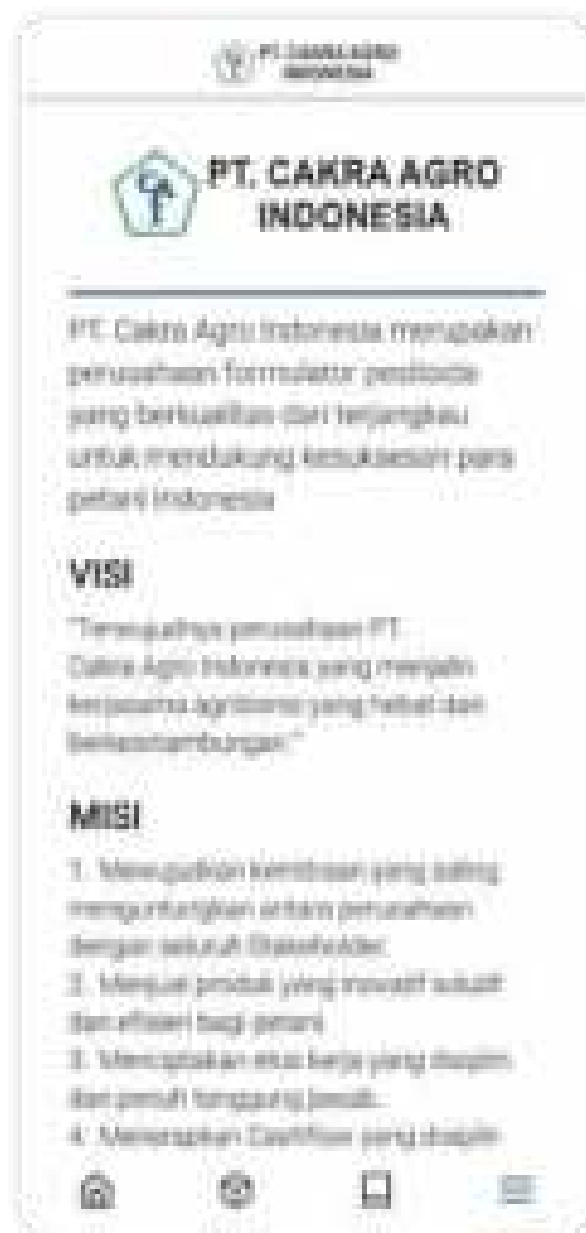


The image shows a mobile application interface for adding a question. The form is titled "Tambah Pertanyaan" and contains the following elements:

- Input field for "Tempat" (Location)
- Input field for "Jenis" (Type)
- Input field for "Waktu" (Time)
- Input field for "Durasi Waktu" (Duration)
- Text area for "Pertanyaan" (Question)
- "Unggah Foto" (Upload Photo) button
- Green "KIRIM" (SEND) button
- Dark blue "Lihat dan Berikan Jawaban" (View and Give Answer) button at the bottom

Untuk mengisi pertanyaan, pengguna bisa mengisi kolom-kolom diatas lalu klik tombol kirim.

Berikutnya adalah halaman lainnya yang bisa diakses dengan klik tombol bergambar tiga garis pada panel navigasi



Halaman lainnya akan menampilkan informasi mengenai Cakra Agro Indonesia seperti deskripsi, visi, dan misi. Selain itu juga pengguna bisa gulir/geser kebawah untuk melihat tampilan seperti dibawah



Pada tampilan diatas akan menampilkan Alamat PT. Cakra Agro Indonesia, dan media sosial yang bisa diakses dengan mengklik salah satu dari ketiga tombol media sosial tersebut. Tombol login hanya bisa diakses oleh karyawan PT. Cakra Agro Indonesia

2.1 Superadmin

Superadmin yang ingin mengakses aplikasi PT. Cakra Agro Indonesia bisa mengunduh aplikasi tersebut di playstore ataupun toko aplikasi lainnya seperti Aptoide, setelah diunduh dan dijalankan maka tampilan awal dari aplikasinya adalah menu Login.



Setelah login maka akan masuk pada tampilan menu yang berisi dashboard, notifikasi, lihat LOG, kelola report demonstrator, kelola aktivitas supervisor, kelola data master, kelola QnA, kelola branda, kelola produk, kelola inovasi, kelola tentang kami, dan kelola kontak.



Ketika memilih tombol Dashboard pada panel maka akan menampilkan info Superadmin dimana didalamnya terdapat ID, Nama Superadmin, username, dan password. Pada tampilan ini juga superadmin dapat mengedit informasi tersebut.



Ketika memilih tombol notifikasi pada panel maka akan menampilkan riwayat notifikasi yang telah masuk sebelumnya.



Ketika memilih tombol notifikasi pada panel maka akan menampilkan menu log yang didalamnya terdapat pilihan menu berupa log Super admin, log Admin, log Master, dan log User.



Ketika memilih tombol notifikasi pada log maka akan menampilkan riwayat aktivitas super admin dan dapat menampilkan riwayat tersebut dengan urutan nama atau tanggal.



Ketika memilih tombol notifikasi pada log maka akan menampilkan riwayat aktivitas admin dan dapat menampilkan riwayat tersebut dengan urutan nama atau tanggal.



Ketika memilih tombol notifikasi pada log maka akan menampilkan riwayat aktivitas staf-staf dan dapat menampilkan riwayat tersebut dengan urutan nama atau tanggal.



Ketika memilih tombol notifikasi pada logl maka akan menampilkan riwayat aktivitas costumer dan dapat menampilkan riwayat tersebut dengan urutan nama atau tanggal.



Ketika memilih tombol Kelola Report Demonstrator pada panel maka akan menampilkan daftar report demonstrator.



Kemudian ketika pada edit memilih kolom yang bergambar tempat sampah maka akan menampilkan tulisan yang memastikan “apakah anda yakin ingin menghapus?” dan tombol untuk menghapus report tersebut.



Ketika memilih tombol Kelola Report Demonstrator pada panel maka akan menambahkan report demonstrator.



Setelah menambahkan report demonstrator maka terdapat tombol tinjau dan ketika tombol tinjau dipilih maka akan menampilkan data report yang sudah ditambahkan tadi.



PT. CAKRA ASINDO
INDONESIA

< TINJAU REPORT BARU



Ganti Foto

Kode Report

Nama Demonstrator

Nama Supervisor

Wibukatan Demonstrator

Provinsi Demonstrator

Tanggal

Nama Petani

Nomor Telpun

Ketika memilih tombol Kelola Aktivitas Baru pada panel maka akan melakukan tambah aktivitas supervisor.



Ketika superadmin memilih untuk mengedit maka semua data pada aktivitas tambah data dapat diedit.



The screenshot shows a web form titled "AKTIVITAS" for PT. LINDA AGRI. The form contains the following fields: a date field with "10 Apr 2024", a dropdown menu, and several empty text input fields. Below the form is a photo of people working in a field. At the bottom of the form is a green "Simpan" button.

Jika pada tambah aktivitas supervisor dan superadmin memilih untuk menghapus data maka akan menampilkan pertanyaan untuk menanyakan apakah yakin untuk menghapus data tersebut.



setelah menambahkan daftar aktivitas supervisor maka akan menampilkan daftar aktivitas baru yang telah dilakukan oleh Supervisor.



Setelah menambahkan data makan akan menampilkan data yang sudah ditambahkan kemudian pilih kolom tinjau untuk melakukan tinjauan aktivitas baru.

The screenshot shows a mobile application interface for a 'TINJAU AKTIVITAS BARU' (New Activity Review) form. At the top, there is a header with the logo of 'PT. JUMALAHIRI BERSEKUTU' and a blue button labeled 'TINJAU AKTIVITAS BARU'. Below the header is a photo of a person in a field, with a 'Galeri Foto' button underneath. The form consists of several input fields, each with a label and a dropdown arrow:

- Nama Kegiatan:** [Empty text field]
- Pemberi Laporan:** [Dropdown menu with 'Pilih Nama User' selected]
- Nama Kegiatan:** [Text field with 'Kegiatan Tinjauan Lapangan' selected]
- Kategori:** [Text field with 'Kategori Tinjauan' selected]
- Tempat:** [Text field with 'Kategori Tinjauan' selected]
- Jenis Kegiatan:** [Text field with 'Kategori Tinjauan' selected]
- Kategori:** [Text field with 'Kategori Tinjauan' selected]
- Tanggal Kegiatan:** [Text field with 'Kategori Tinjauan' selected]
- Tanggal Laporan:** [Text field with 'Kategori Tinjauan' selected]
- Status:** [Text field with 'Kategori Tinjauan' selected]
- Status:** [Text field with 'Kategori Tinjauan' selected]
- Merkal App:** [Text field with 'Kategori Tinjauan' selected]

At the bottom of the form is a green button labeled 'Kembali'.

Ketika memilih tombol Kelola data master pada panel maka akan menampilkan pilihan berupa data admin, data master, data manager, data supervisor, data demonstrator, dan data distributor.



Ketika memilih tombol Data Admin pada Data master maka akan menampilkan daftar data admin yang didalamnya terdapat ID,nama admin, username, password dan edit.



Jika memilih tambah pada data admin maka akan menampilkan form admin yang didalamnya terdapat id, nama, username, dan password.



The image shows a screenshot of an Android application interface. At the top, there is a logo for 'PT. GARBA AGRI INDONESIA' and a back arrow icon. Below the logo, the word 'ADMIN' is displayed in a large, bold, black font. Underneath, there are four input fields stacked vertically, each with a label: 'ID', 'Name', 'Username', and 'Password'. At the bottom of the screen, there is a prominent green button with the white text 'Tambah' (Add).

Ketika memilih tombol Data Manager pada Data master maka akan menampilkan daftar data manager yang didalamnya terdapat ID, nama admin, username, password dan edit. Pilih tambah jika ingin menambah data manager.



Jika memilih tambah pada data manager maka akan menampilkan form manager yang didalamnya terdapat nama manager, username, dan password.

The image shows a web form for adding a manager. At the top, there is a logo for PT. CARRA AGRU INDONESIA. Below the logo, the word "MANAGER" is displayed in a large, bold font. The form consists of three input fields: "Nama Manager", "Username", and "Password". Below these fields is a profile picture of a man in a dark suit and glasses, holding a document. Underneath the profile picture is a button labeled "Simpan". At the bottom of the form, there is a green button labeled "Submit".

Ketika memilih tombol Data Secretary pada Data master maka akan menampilkan daftar data secretary yang didalamnya terdapat ID, secretary, manager, username, password dan edit. Pilih tambah jika ingin menambah data secretary.



Jika memilih tambah pada data secretary maka akan menampilkan form secretary yang didalamnya terdapat nama secretary, nama manager, username, dan password.

The image shows a mobile application interface for adding a secretary. At the top, there is a logo for 'PT CERIA AGRI INDONESIA' and a back arrow. The title 'SECRETARY' is centered. Below the title are four input fields: 'Nama Secretary', 'Manager' (with a dropdown arrow), 'Username', and 'Password'. Underneath the form is a placeholder for a profile picture, showing a woman in a business suit. At the bottom, there is a green button labeled 'Simpan'.

Ketika memilih tombol Data Supervisor pada Data master maka akan menampilkan daftar data supervisor yang didalamnya terdapat ID, supervisor, budget, area sales, provinsi, username, password dan edit. Pilih tambah jika ingin menambah data supervisor.



Jika memilih tambah pada data supervisor maka akan menampilkan form supervisor yang didalamnya terdapat nama supervisor, nama manager, budget, area sales, provinsi, username, dan password.

The image shows a mobile application interface for adding a supervisor. At the top, there is a logo for 'PT. CAKRA BUDI INDONESIA' and the title 'SUPERVISOR'. Below the title, there are seven input fields: 'Nama Supervisor', 'Nama Manager' (with a dropdown arrow), 'Budget', 'Area Sales', 'Provinsi', 'Username', and 'Password'. Below the form is a placeholder image of a man in a suit. At the bottom, there is a green button labeled 'Tambah' and a small icon with a plus sign.

Ketika memilih tombol Data demonstrator pada Data master maka akan menampilkan daftar data demonstrator yang didalamnya terdapat ID, demonstrator, supervisor, provinsi, kabupaten, username, password dan edit. Pilih tambah jika ingin menambah data demonstrator.



Jika memilih tambah pada data demonstrator maka akan menampilkan form demonstrator yang didalamnya terdapat nama demonstrator, nama supervisor, provinsi, kabupaten, username, dan password.



The image shows a mobile application interface for adding a demonstrator. At the top, there is a logo for 'PT CARABANG NEONIDA' and the title 'DEMONSTRATOR'. Below the title, there are six input fields: 'Nama Demonstrator', 'Nama Supervisor', 'Provinsi', 'Kabupaten', 'Username', and 'Password'. Below the form, there is a profile picture of a person wearing a hat and holding a laptop. At the bottom, there is a green 'Save' button.

Ketika memilih tombol Data distributor pada Data master maka akan menampilkan daftar data demonstrator yang didalamnya terdapat ID, distributor, perusahaan, username, password, dan edit. Pilih tambah jika ingin menambah data distributor.



Jika memilih tambah pada data distributor maka akan menampilkan form distributor yang didalamnya terdapat nama distributor, nama secretary, perusahaan, username, dan password.



PT. SARANA KIRI
INDONESIA

DISTRIBUTOR

Nama Distributor:

Nama Sekretaris:

Perusahaan:

Username:

Password:



Kembali

Simpan

Kemudian jika ingin menghapus data yang ditambahkan maka akan muncul pertanyaan “apakah anda yakin ingin menghapus”.



Ketika memilih tombol Kelola pertanyaan pada panel maka akan menampilkan kelola pertanyaan yang didalamnya terdapat pilihan data QnA dan Data FAQ.



Kemudian jika ingin menghapus data yang ditambahkan maka akan muncul pertanyaan “apakah anda yakin ingin menghapus”.



Ketika memilih data FAQ pada kelola pertanyaan maka akan menampilkan data FAQ yang didalamnya terdapat pertanyaan yang diajukan untuk PT. Cakra Argo Indonesia.



Jika ingin menambahkan data FAQ maka akan menampilkan form FAQ yang berisi judul pertanyaan, pertanyaan, jawaban, dan foto.



The screenshot shows a mobile application interface for adding a new FAQ entry. At the top, there is a logo for 'PT LAMBA AGRI MURNI LIA' and the title 'FAQ'. Below the title, there are three text input fields for 'Judul pertanyaan', 'Pertanyaan', and 'Jawaban'. A photo selection area is visible, showing a preview of a green plant with a small orange insect. Below the photo area is a 'Pilih foto' button. At the bottom of the form is a large green 'Simpan' (Save) button.

Kemudian jika ingin menghapus data yang ditambahkan maka akan muncul pertanyaan "apakah anda yakin ingin menghapus".



Ketika memilih tombol Kelola beranda pada panel maka akan menampilkan kelola beranda yang didalamnya terdapat pilihan homepage, produk homepage, dan berita.



Ketika memilih homepage pada kelola produk maka akan menampilkan data homepage yang akan diubah.



Ketika memilih produk homepage pada kelola produk maka akan menampilkan data produk homepage.



Ketika memilih berita pada kelola produk maka akan menampilkan data berita yang sudah ada pada halaman berita sebelumnya.



Jika memilih tambah maka akan diarahkan ke form berita yang berisi judul dan deskripsi.



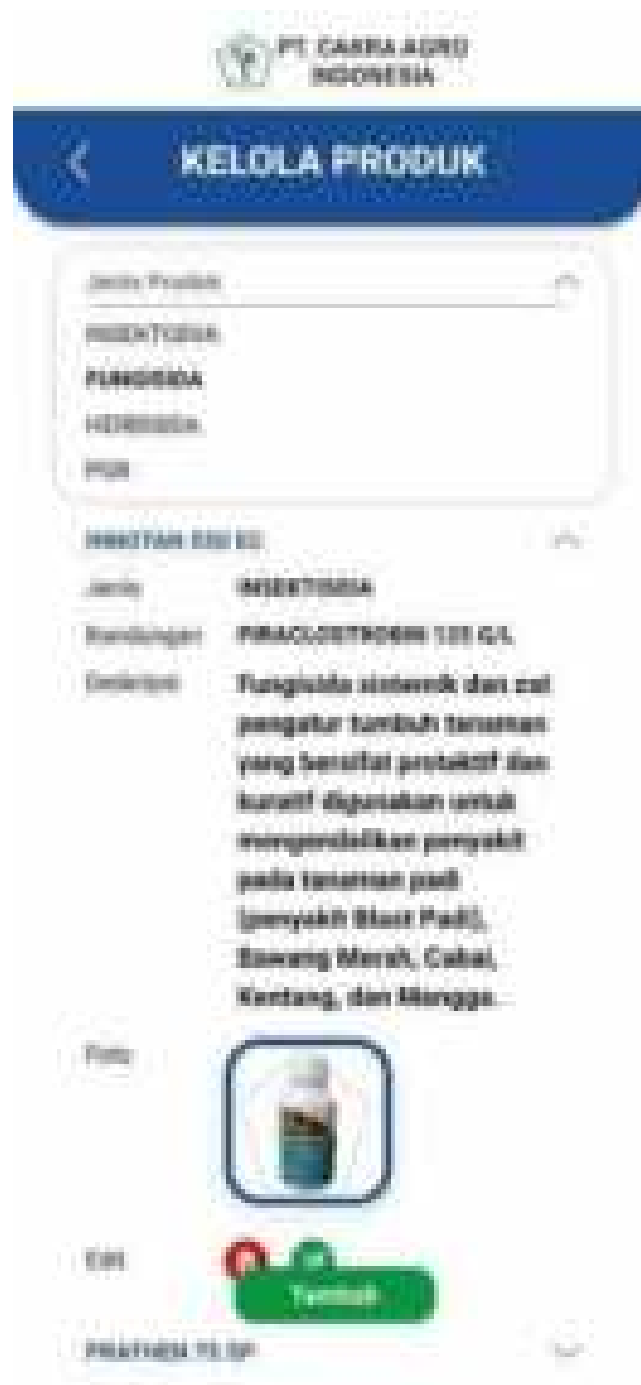
The screenshot shows a web form titled "BERITA" from PT. SARANA BUKU INDONESIA. The form contains two input fields: a smaller one for the title and a larger one for the description. A green "Tambah" button is located at the bottom of the form.

Kemudian jika ingin menghapus data yang ditambahkan maka akan muncul pertanyaan "apakah anda yakin ingin menghapus".



The screenshot shows a confirmation dialog box with the text "Apakah anda yakin ingin menghapus?". A red "Ya" button is visible at the bottom of the dialog.

Ketika memilih tombol Kelola produk pada panel maka akan menampilkan kelola produk yang didalamnya terdapat jenis produk, dan nama produk yang didalam nama produk terdapat jenis, kandungan, deskripsi, foto , dan edit.



Jika ingin menambahkan produk maka akan menampilkan form tambah produk yang berisi jenis, nama produk, kandungan, deskripsi, dan foto.



The screenshot shows a mobile application interface for adding a product. At the top, there is a logo for PT. SARANA AGRO INDONESIA. Below the logo is a back arrow and the title 'TAMBAH PRODUK'. The form consists of several input fields: a dropdown menu for 'Jenis', a text field for 'Nama Produk', a text field for 'Kandungan', and a larger text area for 'Deskripsi'. Below these fields is a photo upload section with a placeholder image of a blue and white bottle. At the bottom of the form is a green button labeled 'Tambah'.

Kemudian jika ingin menghapus data yang ditambahkan maka akan muncul pertanyaan “apakah anda yakin ingin menghapus”.



Ketika memilih tombol Kelola inovasi pada panel maka akan menampilkan kelola inovasi yang didalamnya terdapat pilihan produk baru, dan paket produk.



Ketika memilih produk baru pada kelola inovasi maka akan menampilkan data kelola produk baru.



Jika ingin menambahkan produk baru maka akan menampilkan form tambah produk baru yang berisi nama bahan aktif, formulasi, deskripsi, dan hasil.



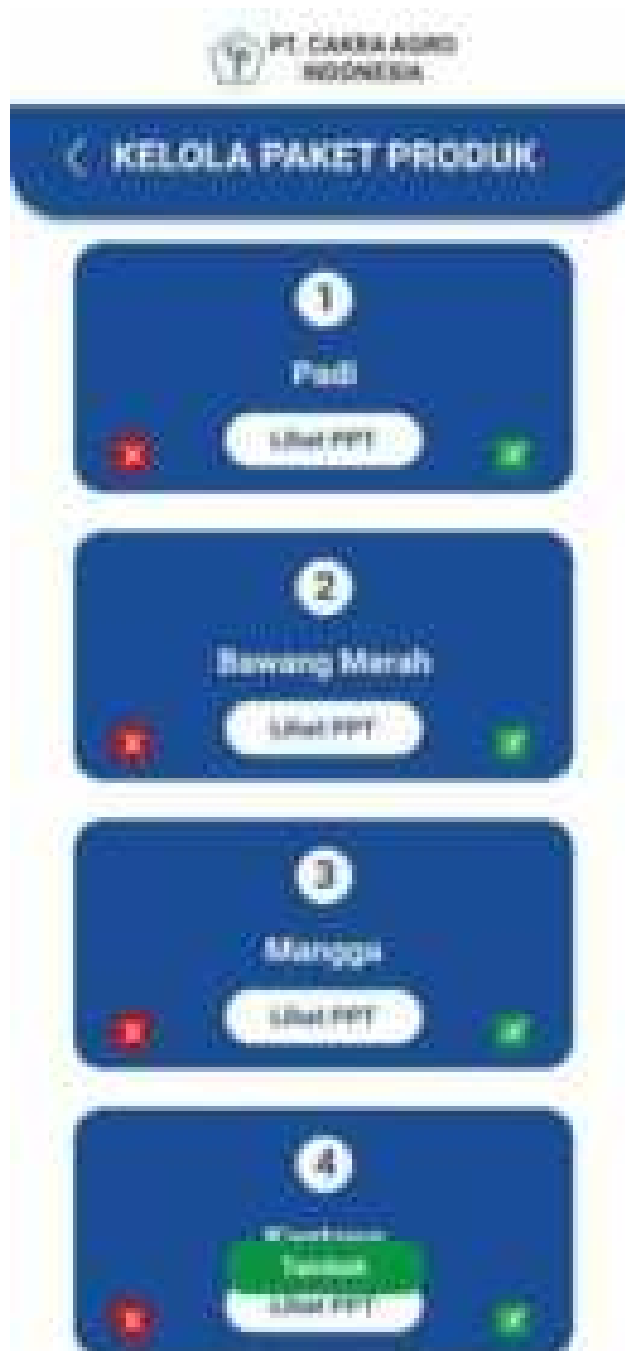
The screenshot shows a mobile application interface for adding a new product. At the top, there is a logo for 'PE-SARANA-UMUM-INDONESIA'. Below the logo, the title 'TAMBAH PRODUK BARU' is displayed. The form consists of four input fields: 'Nama Bahan Aktif', 'Formulasi', 'Deskripsi', and 'Hasil'. A green 'Simpan' button is located at the bottom of the form.

Kemudian jika ingin menghapus data yang ditambahkan maka akan muncul pertanyaan "apakah anda yakin ingin menghapus".



The screenshot shows a confirmation dialog box with the text 'Apakah anda yakin ingin menghapus?'. Below the text is a red 'Hapus' button.

Ketika memilih paket produk pada kelola inovasi maka akan menampilkan data kelola paket produk.



Jika ingin menambahkan paket produk maka akan menampilkan form paket produk baru yang berisi nama paket, tanaman, deskripsi, hasil, dan kolom untuk mengupload PPT.



The screenshot shows a mobile application interface for adding a product package. At the top, there is a logo for PT. TAMBAKANI MOBILE. Below the logo, the title 'TAMBAH PAKET PRODUK' is displayed. The form consists of several input fields: a text field for 'Nama Paket', a text field for 'Tanaman', a larger text area for 'Deskripsi', and a text field for 'Hasil'. Below these fields is a blue button labeled 'Simpan Data'. At the bottom of the form, there is a green button labeled 'Tambah'.

Kemudian jika ingin menghapus data yang ditambahkan maka akan muncul pertanyaan "apakah anda yakin ingin menghapus".

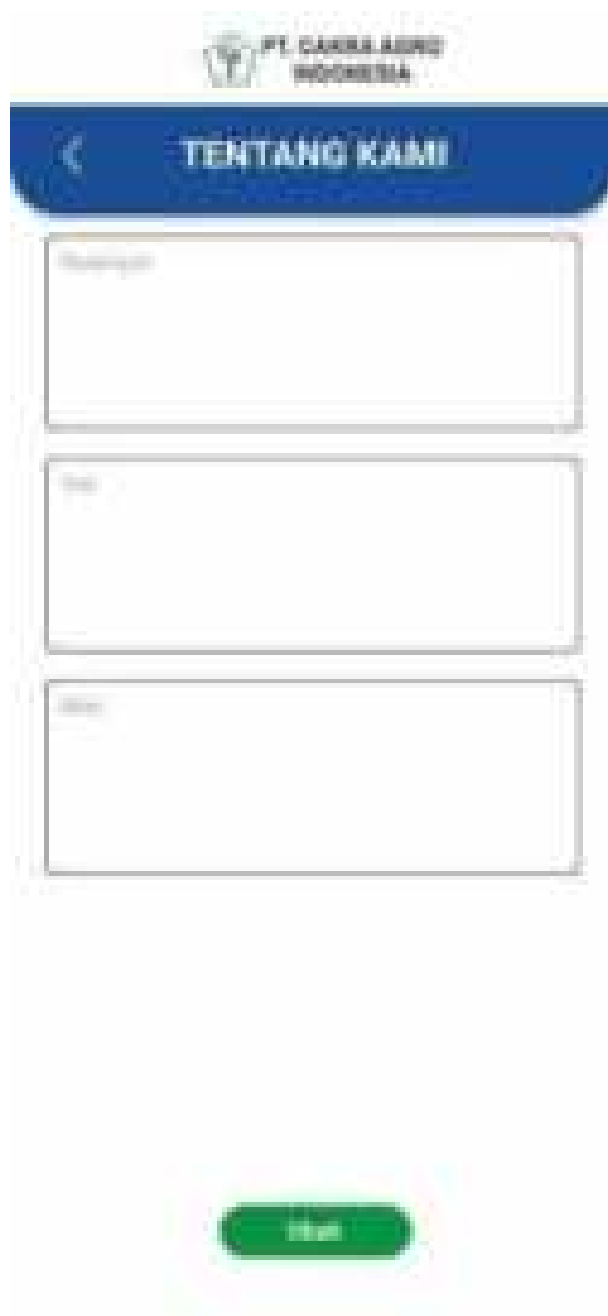


The screenshot shows a confirmation dialog box with a white background and rounded corners. The text inside the dialog reads 'Apakah anda yakin ingin menghapus?'. Below the text is a red button labeled 'Ya'.

Ketika memilih tombol Kelola tentang kami pada panel maka akan menampilkan kelola tentang kami yang didalamnya terdapat pilihan deskripsi, visi, dan misi, alamat, dan sosial media.



jika memilih deskripsi, visi, dan misi maka pada kelola tentang kami akan diarahkan untuk mengubah isi pada halaman tersebut yang berisi deskripsi, visi, dan misi.



jika ingin menambah alamat baru maka akan diarah pada form alamat yang berisi nama kantor, alamat, no telpon.

The screenshot shows a mobile application interface for PT CENARA KORD INDONESIA. The title is 'ALAMAT'. There are three input fields: 'Nama Kantor', 'Alamat', and 'No. Telepon'. Below these is a photo upload area with a camera icon. At the bottom is a green 'Simpan' button.

Kemudian jika ingin menghapus data yang ditambahkan maka akan muncul pertanyaan "apakah anda yakin ingin menghapus".

The screenshot shows a confirmation dialog box with the text 'Apakah anda yakin ingin menghapus?' and a red 'Hapus' button.

jika memilih sosial media maka pada kelola tentang kami akan diarahkan untuk mengubah isi pada halaman tentang kami yang berisi link facebook, link Instagram, dan link youtube.

PT CARRA AGRO INDONESIA

SOSIAL MEDIA

Link Facebook

Link Instagram

Link YouTube

Simpan

2.2 Admin

Admin yang ingin mengakses aplikasi PT. Cakra Agro Indonesia bisa mengunduh aplikasi tersebut di playstore ataupun toko aplikasi lainnya seperti Aptoide, setelah diunduh dan dijalankan maka tampilan awal dari aplikasinya adalah menu Login.



Setelah login maka akan masuk pada tampilan menu yang berisi dashboard, notifikasi, lihat LOG, kelola report demonstrator, kelola aktivitas supervisor, kelola data master, kelola QnA, kelola branda, kelola produk, kelola inovasi, kelola tentang kami, dan kelola kontak.



Ketika memilih tombol Dashboard pada panel maka akan menampilkan info Admin dimana didalamnya terdapat ID, Nama Admin, username, dan password. Pada tampilan ini juga Admin dapat mengedit informasi tersebut.



Ketika memilih tombol notifikasi pada panel maka akan menampilkan riwayat notifikasi yang telah masuk sebelumnya.



Ketika memilih tombol notifikasi pada panel maka akan menampilkan menu log yang didalamnya terdapat pilihan menu berupa log Super admin, log Admin, log Master, dan log User.



Ketika memilih tombol notifikasi pada log maka akan menampilkan riwayat aktivitas admin dan dapat menampilkan riwayat tersebut dengan urutan nama atau tanggal.



Ketika memilih tombol notifikasi pada log master maka akan menampilkan riwayat aktivitas staf-staf dan dapat menampilkan riwayat tersebut dengan urutan nama atau tanggal.



Ketika memilih tombol notifikasi pada log user maka akan menampilkan riwayat aktivitas user dan dapat menampilkan riwayat tersebut dengan urutan nama atau tanggal.



Ketika memilih tombol Kelola Report Demonstrator pada panel maka akan menampilkan daftar report demonstrator.



Kemudian ketika pada edit memilih kolom yang bergambar tempat sampah maka akan menampilkan tulisan yang memastikan “apakah anda yakin ingin menghapus?” dan tombol untuk menghapus report tersebut.



Ketika memilih tombol Kelola Report Demonstrator pada panel maka akan menambahkan report demonstrator.



Setelah menambahkan report demonstrator maka terdapat tombol tinjau dan ketika tombol tinjau dipilih maka akan menampilkan data report yang sudah ditambahkan tadi.



PT. CAKRA ASINDO
INDONESIA

< TINJAU REPORT BARU



Ganti Foto

Kode Report

Nama Demonstrator

Nama Supervisor

Wibukatan Demonstrator

Pisawal Demonstrator

Tanggal

Nama Petani

Nomor Telpun

Uraian

Kategori

Kategori

Periode

Profil

Uraian

Detail (opsional)

Detail

Detail (opsional)

[Kembali ke Atas](#)

Ketika memilih tombol Kelola Aktivitas Baru pada panel maka akan melakukan tambah aktivitas supervisor.



Ketika Admin memilih untuk mengedit maka semua data pada aktivitas tambah data dapat diedit.



The screenshot shows a mobile application interface for managing activities. At the top, there is a logo for 'PT. LINDA AGRI' and the title 'AKTIVITAS'. Below the title is a date picker set to '10 - Mei 2024'. The form contains several input fields: 'Kategori' (a dropdown menu), 'Nama Kegiatan', 'Membuat', 'Tempat', 'Jumlah Peserta', 'Galeri', and 'Status'. Below the form is a photo gallery with one image showing a group of people working in a field. At the bottom of the form is a green 'Simpan' button.

Jika pada hapus aktivitas supervisor, admin memilih untuk menghapus data maka akan menampilkan pertanyaan untuk menanyakan apakah yakin untuk menghapus data tersebut.



setelah menambahkan daftar aktivitas supervisor maka akan menampilkan daftar aktivitas baru yang telah dilakukan oleh Supervisor.



Setelah menambahkan data makan akan menampilkan data yang sudah ditambahkan kemudian pilih kolom tinjau untuk melakukan tinjauan aktivitas baru.



The screenshot shows a mobile application interface for a new activity review. At the top, there is a header with the text 'TINJAU AKTIVITAS BARU' and a back arrow. Below the header is a photograph of a person working in a field. Underneath the photo is a button labeled 'Buka Foto'. The main part of the screen contains a series of input fields, each with a label above it: 'Nama Kegiatan', 'Periode Pelaksanaan', 'Nama Kegiatan', 'Kategori', 'Tempat', 'Jenis Kegiatan', 'Jenis Kegiatan', 'Kategori', 'Status', 'Status', and 'Merkal App'. Each field contains some text, and some have dropdown arrows. At the bottom of the form is a green button labeled 'Simpan Data'.

Ketika memilih tombol Kelola data master pada panel maka akan menampilkan pilihan berupa data admin, data master, data manager, data supervisor, data demonstrator, dan data distributor.



Ketika memilih tombol Data Admin pada Data master maka akan menampilkan daftar data admin yang didalamnya terdapat ID,nama admin, username, password dan edit.



Jika memilih tambah pada data admin maka akan menampilkan form admin yang didalamnya terdapat id, nama, username, dan password.

The image shows a screenshot of an Android application interface. At the top, there is a logo for 'PT. GARBA AGRI INDONESIA' and a back arrow icon. Below the logo, the word 'ADMIN' is displayed in a large, bold, black font. Underneath, there are four input fields stacked vertically, each with a label: 'ID', 'Name', 'Username', and 'Password'. At the bottom of the screen, there is a prominent green button with the white text 'Tambah' (Add).

Ketika memilih tombol Data Manager pada Data master maka akan menampilkan daftar data manager yang didalamnya terdapat ID, nama admin, username, password dan edit. Pilih tambah jika ingin menambah data manager.



Jika memilih tambah pada data manager maka akan menampilkan form manager yang didalamnya terdapat nama manager, username, dan password.



PT. CARRA AGRU
INDONESIA

MANAGER

Nama Manager

Username

Password



Submit

Ketika memilih tombol Data Secretary pada Data master maka akan menampilkan daftar data secretary yang didalamnya terdapat ID, secretary, manager, username, password dan edit. Pilih tambah jika ingin menambah data secretary.



Jika memilih tambah pada data secretary maka akan menampilkan form secretary yang didalamnya terdapat nama secretary, nama manager, username, dan password.

The image shows a mobile application interface for adding a secretary. At the top, there is a logo for 'PT CERIA AGRI INDONESIA' and a back arrow. The title 'SECRETARY' is centered. Below the title are four input fields: 'Name Secretary', 'Manager' (with a dropdown arrow), 'Username', and 'Password'. Underneath the form is a placeholder image of a woman in a business suit. At the bottom, there is a green 'Submit' button.

Ketika memilih tombol Data Supervisor pada Data master maka akan menampilkan daftar data supervisor yang didalamnya terdapat ID, supervisor, budget, area sales, provinsi, username, password dan edit. Pilih tambah jika ingin menambah data supervisor.



Jika memilih tambah pada data supervisor maka akan menampilkan form supervisor yang didalamnya terdapat nama supervisor, nama manager, budget, area sales, provinsi, username, dan password.

The image shows a mobile application interface for adding a supervisor. At the top, there is a logo for 'PT. CAKRA BUDI INDONESIA' and the title 'SUPERVISOR'. Below the title, there are seven input fields: 'Nama Supervisor', 'Nama Manager' (with a dropdown arrow), 'Budget', 'Area Sales', 'Provinsi', 'Username', and 'Password'. Below the form is a placeholder image of a man in a suit. At the bottom, there is a green button labeled 'Tambah' and a small icon with a plus sign.

Ketika memilih tombol Data demonstrator pada Data master maka akan menampilkan daftar data demonstrator yang didalamnya terdapat ID, demonstrator, supervisor, provinsi, kabupaten, username, password dan edit. Pilih tambah jika ingin menambah data demonstrator.



Jika memilih tambah pada data demonstrator maka akan menampilkan form demonstrator yang didalamnya terdapat nama demonstrator, nama supervisor, provinsi, kabupaten, username, dan password.



PT CARRASO
INDONESIA

DEMONSTRATOR

Nama Demonstrator

Nama Supervisor

Provinsi

Kabupaten

Username

Password



Kembali

Simpan

Ketika memilih tombol Data distributor pada Data master maka akan menampilkan daftar data demonstrator yang didalamnya terdapat ID, distributor, perusahaan, username, password, dan edit. Pilih tambah jika ingin menambah data distributor.



Jika memilih tambah pada data distributor maka akan menampilkan form distributor yang didalamnya terdapat nama distributor, nama secretary, perusahaan, username, dan password.



PT. SARANA KUNCI
INDONESIA

DISTRIBUTOR

Nama Distributor:

Nama Sekretaris:

Perusahaan:

Username:

Password:



Kemudian jika ingin menghapus data yang ditambahkan maka akan muncul pertanyaan “apakah anda yakin ingin menghapus”.



Ketika memilih tombol Kelola pertanyaan pada panel maka akan menampilkan kelola pertanyaan yang didalamnya terdapat pilihan data QnA dan Data FAQ.



Ketika memilih data Qna pada kelola pertanyaan maka akan menampilkan data Qna yang didalamnya terdapat pertanyaan, foto, jawaban, nama orang yang bertanya, alamat, no telp, dan tanggal pertanyaan diunggah.



Jika ingin menjawab pertanyaan akan menampilkan popup untuk menjawab pertanyaan.



Kemudian jika ingin menghapus data yang ditambahkan maka akan muncul pertanyaan “apakah anda yakin ingin menghapus”.



Ketika memilih data FAQ pada kelola pertanyaan maka akan menampilkan data FAQ yang didalamnya terdapat pertanyaan yang diajukan untuk PT. Cakra Argo Indonesia.



Jika ingin menambahkan data FAQ maka akan menampilkan form FAQ yang berisi judul pertanyaan, pertanyaan, jawaban, dan foto.



The screenshot shows a mobile application interface for adding a new FAQ entry. At the top, there is a logo for 'PT LAMBA AGRI MANDIRI LIA' and the title 'FAQ'. Below the title, there are three text input fields for entering the question, the question text, and the answer. A fourth field is a photo selection area, which currently displays a photograph of a person in a field. At the bottom of the form, there is a green button labeled 'Simpan' (Save).

Kemudian jika ingin menghapus data yang ditambahkan maka akan muncul pertanyaan “apakah anda yakin ingin menghapus”.



Ketika memilih tombol Kelola beranda pada panel maka akan menampilkan kelola beranda yang didalamnya terdapat pilihan homepage, produk homepage, dan berita.



Ketika memilih homepage pada kelola produk maka akan menampilkan data homepage yang akan diubah.



Ketika memilih produk homepage pada kelola produk maka akan menampilkan data produk homepage.



Kemudian jika memilih ubah pada halam sebelumnya maka akan diarahkan ke halaman edit.



Kemudian jika ingin menghapus data yang ditambahkan maka akan muncul pertanyaan “apakah anda yakin ingin menghapus”.



Ketika memilih berita pada kelola produk maka akan menampilkan data berita yang sudah ada pada halaman berita sebelumnya.



Jika memilih tambah maka akan diarahkan ke form berita yang berisi judul dan deskripsi.



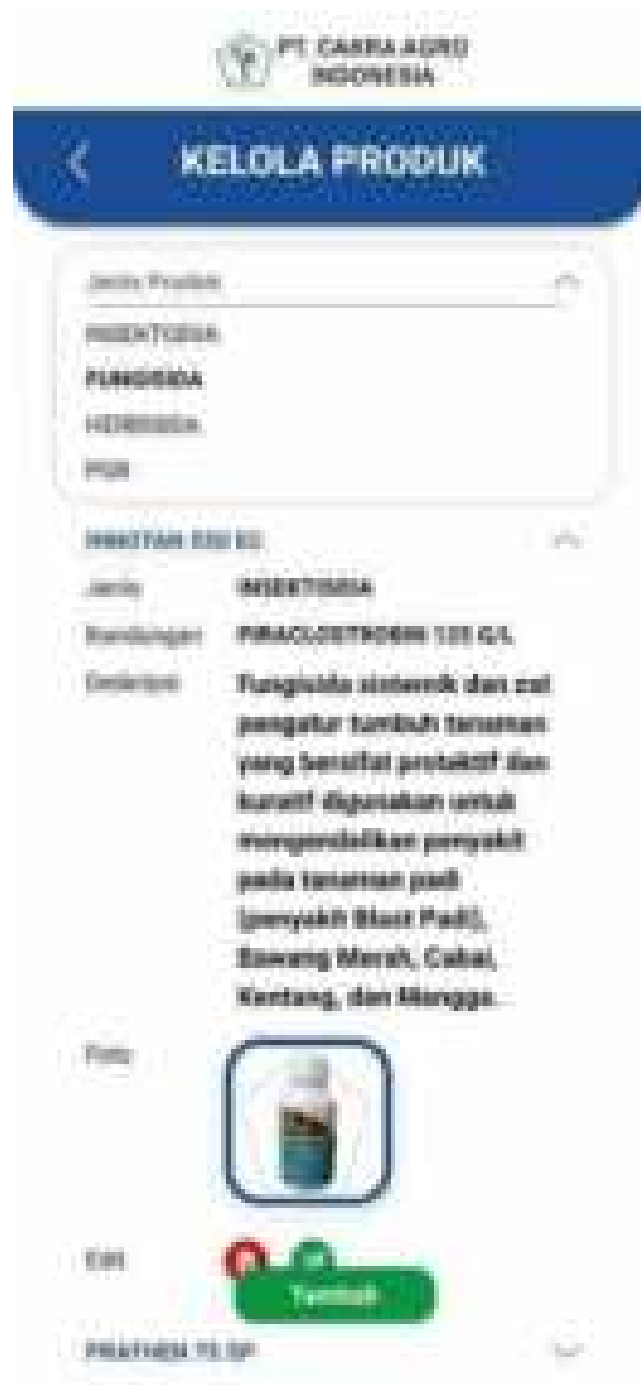
The screenshot shows a web form titled "BERITA" from PT. SARANA BUKU INDONESIA. The form contains two input fields: a smaller one for the title and a larger one for the description. A green "Tambah" button is located at the bottom of the form.

Kemudian jika ingin menghapus data yang ditambahkan maka akan muncul pertanyaan "apakah anda yakin ingin menghapus".



The screenshot shows a confirmation dialog box with the text "Apakah anda yakin ingin menghapus?". A red "Ya" button is visible at the bottom of the dialog.

Ketika memilih tombol Kelola produk pada panel maka akan menampilkan kelola produk yang didalamnya terdapat jenis produk, dan nama produk yang didalam nama produk terdapat jenis, kandungan, deskripsi, foto , dan edit.



Jika ingin menambahkan produk maka akan menampilkan form tambah produk yang berisi jenis, nama produk, kandungan, deskripsi, dan foto.



The screenshot shows a mobile application interface for adding a product. At the top, there is a logo for PT. SARANA AGRO INDONESIA. Below the logo is a back arrow and the title 'TAMBAH PRODUK'. The form consists of several input fields: a dropdown menu for 'Jenis', a text field for 'Nama Produk', a text field for 'Kandungan', and a larger text area for 'Deskripsi'. Below these fields is a photo upload area containing a small image of a white bottle with a blue label. At the bottom of the form is a green button labeled 'Tambah'.

Kemudian jika ingin menghapus data yang ditambahkan maka akan muncul pertanyaan “apakah anda yakin ingin menghapus”.



Ketika memilih tombol Kelola inovasi pada panel maka akan menampilkan kelola inovasi yang didalamnya terdapat pilihan produk baru, dan paket produk.



Ketika memilih produk baru pada kelola inovasi maka akan menampilkan data kelola produk baru.



Jika ingin menambahkan produk baru maka akan menampilkan form tambah produk baru yang berisi nama bahan aktif, formulasi, deskripsi, dan hasil.



The screenshot shows a mobile application interface for adding a new product. At the top, there is a logo for 'PE-SARANA-UMUM-INDONESIA' and a title 'TAMBAH PRODUK BARU'. Below the title, there is a form with four input fields: 'Nama Bahan Aktif', 'Formulasi', 'Deskripsi', and 'Hasil'. A green 'Tambah' button is located at the bottom of the form.

Kemudian jika ingin menghapus data yang ditambahkan maka akan muncul pertanyaan “apakah anda yakin ingin menghapus”.



The screenshot shows a confirmation dialog box with the text 'Apakah anda yakin ingin menghapus?' and a red 'Hapus' button at the bottom.

Ketika memilih paket produk pada kelola inovasi maka akan menampilkan data kelola paket produk.



Jika ingin menambahkan paket produk maka akan menampilkan form paket produk baru yang berisi nama paket, tanaman, deskripsi, hasil, dan kolom untuk mengupload PPT.

The screenshot shows a mobile application interface for adding a product package. At the top, there is a logo for PT TAMBORA AGRI MOBILE. Below the logo, the title 'TAMBAH PAKET PRODUK' is displayed. The form consists of several input fields: a text field for 'Nama Paket', a text field for 'Tanaman', a larger text area for 'Deskripsi', and a text field for 'Hasil'. Below these fields is a blue button labeled 'Simpan Data'. At the bottom of the screen, there is a green button labeled 'Tambah'.

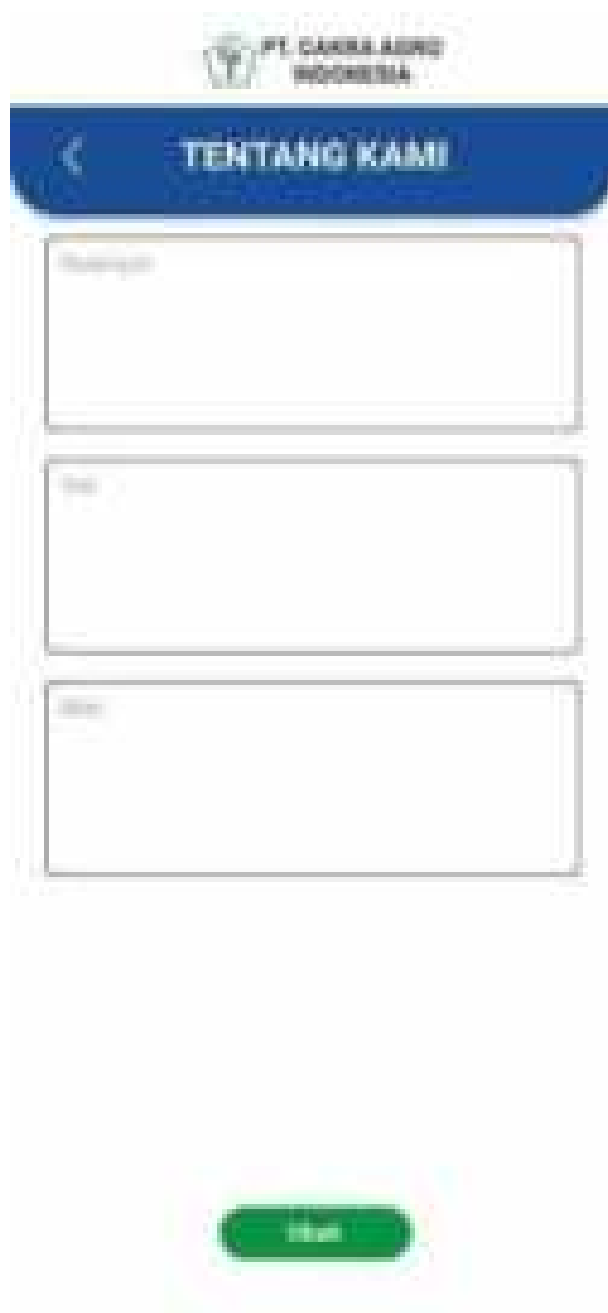
Kemudian jika ingin menghapus data yang ditambahkan maka akan muncul pertanyaan "apakah anda yakin ingin menghapus".



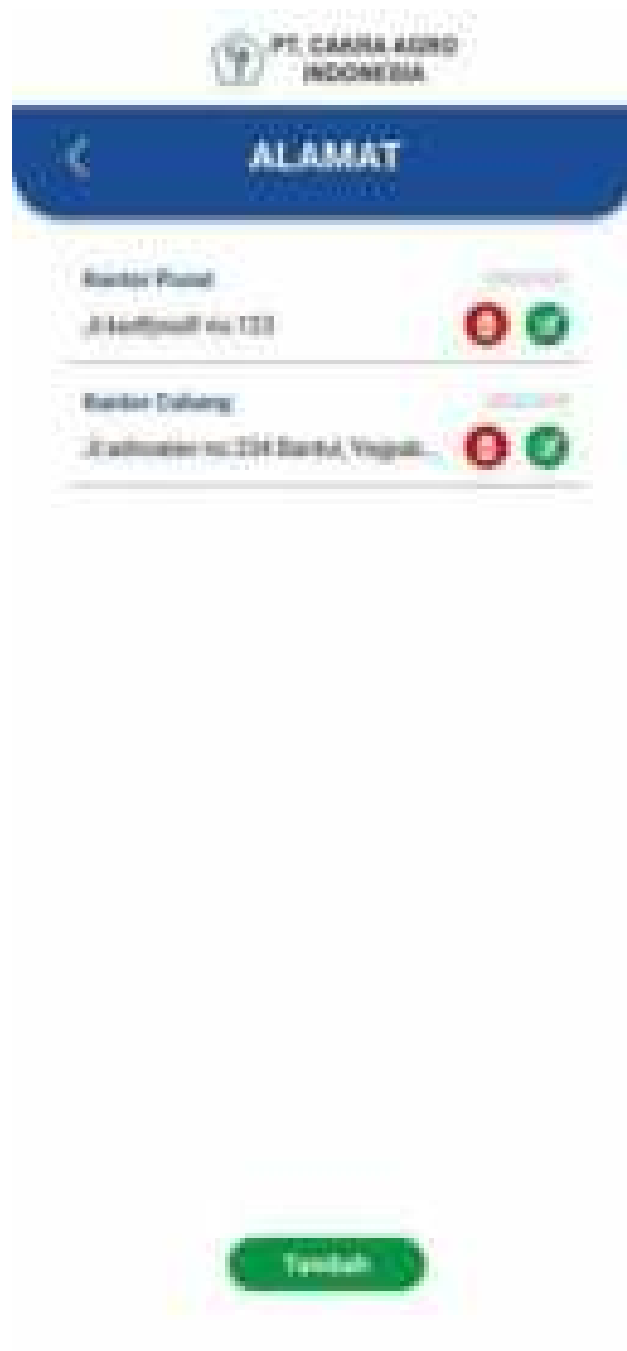
Ketika memilih tombol Kelola tentang kami pada panel maka akan menampilkan kelola tentang kami yang didalamnya terdapat pilihan deskripsi, visi, dan misi, alamat, dan sosial media.



jika memilih deskripsi, visi, dan misi maka pada kelola tentang kami akan diarahkan untuk mengubah isi pada halaman tersebut yang berisi deskripsi, visi, dan misi.



jika memilih alamat maka pada kelola tentang kami maka akan menampilkan alamat yang berada pada halaman tentang kami. Pilih tambah jika ingin menambahkan alamat baru.



jika ingin menambah alamat baru maka akan diarah pada form alamat yang berisi nama kantor, alamat, no telpon.

The screenshot shows a mobile application interface for adding a new address. At the top, there is a logo for PT. CENARA KORD INDONESIA and a back arrow. The title 'ALAMAT' is centered. Below the title are three input fields: 'Nama Kantor', 'Alamat', and 'No. Telepon'. Underneath these fields is a photo upload section with a camera icon and a 'Pilih Foto' button. At the bottom of the form is a large green button labeled 'Simpan'.

Kemudian jika ingin menghapus data yang ditambahkan maka akan muncul pertanyaan "apakah anda yakin ingin menghapus".

The screenshot shows a confirmation dialog box with a white background and rounded corners. The text inside reads 'Apakah anda yakin ingin menghapus?'. At the bottom of the dialog is a red button labeled 'Hapus'.

jika memilih sosial media maka pada kelola tentang kami akan diarahkan untuk mengubah isi pada halaman tentang kami yang berisi link facebook, link Instagram, dan link youtube

The image shows a mobile application interface for managing social media links. At the top, there is a logo for PT CARMA AGRO INDONESIA. Below the logo is a blue header with a back arrow and the text 'SOSIAL MEDIA'. The main content area contains three sections, each with a title and a text input field:

- Link Facebook**: The input field contains the text 'Link Management'.
- Link Instagram**: The input field contains the text 'Link Management'.
- Link YouTube**: The input field contains the text 'Link Management'.

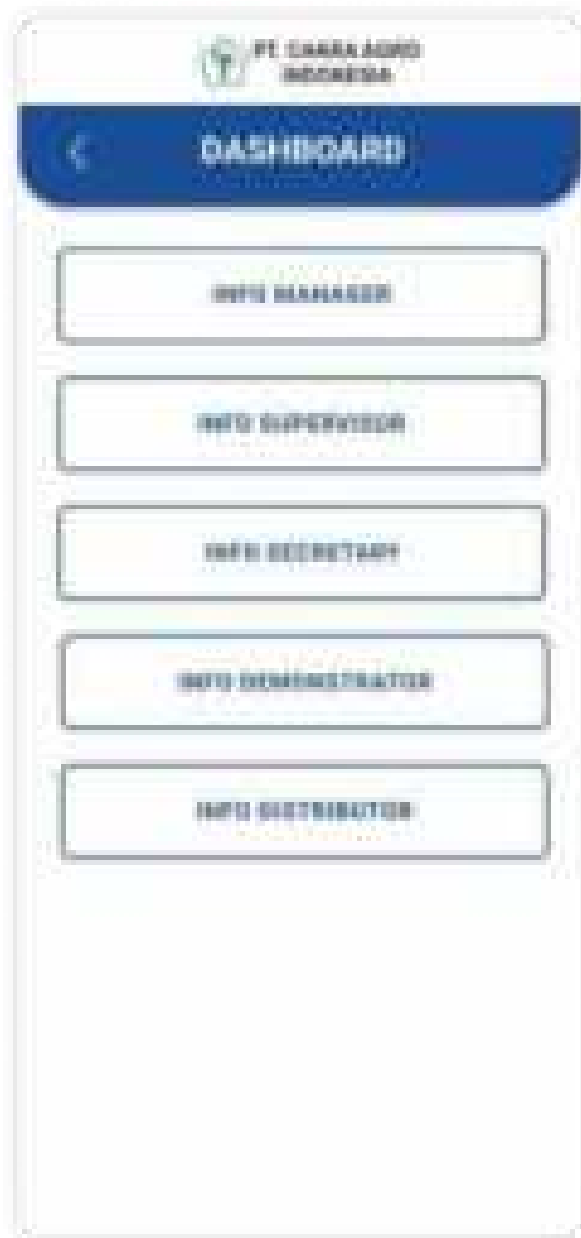
At the bottom of the screen, there is a green button with the text 'Save'.

2.3 Manager

Setelah login dan masuk sebagai manager, maka tampilan awal pada dashboard akan seperti berikut



Gambar diatas merupakan tampilan beranda manager, terdapat tiga tombol yang mengarah ke tiga halaman yang berbeda, halaman pertama yaitu dashboard dengan tampilan seperti dibawah ini



Tampilan dashboard diatas menampilkan 5 jenis tombol untuk info dari 5 jenis pengguna yaitu Manager itu sendiri, supervisor, secretary, demonstrator, dan distributor. Jika pengguna memilih info manager, maka halaman yang akan ditampilkan adalah halaman informasi pengguna seperti dibawah



Halaman info manager diatas merupakan manager yang saat ini sedang login. Terdapat dua tombol yaitu tombol edit dan tombol logout, tombol edit untuk melakukan edit atau ubah info manager saat ini, dan tombol logout untuk logout atau keluar akun, berikut tampilan edit

PT. SARANA AGRO
INDONESIA

EDIT INFO MANAGER



Ganti Foto

Nama Manager

Username

Password

Konfirmasi Password

Simpan

Tampilan diatas adalah tampilan untuk ubah info manager, tekan tombol simpan untuk simpan perubahan.

Setelah halaman info manager berikutnya yaitu halaman supervisor yang dapat diakses jika pengguna(manager) memilih tombol info supervisor, maka tampilannya akan seperti dibawah

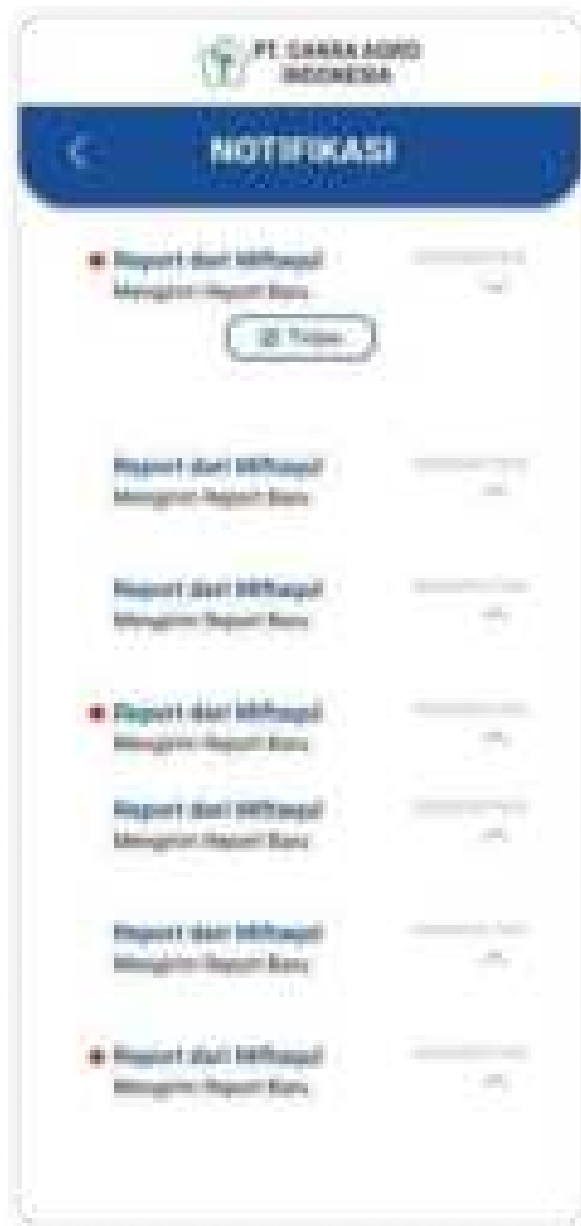


Tampilan diatas menampilkan daftar supervisor dari pengguna (manager), berikutnya pengguna bisa memilih gambar dari supervisor tersebut untuk melihat detailnya, seperti dibawah ini



Setelah itu untuk info dari 3 pengguna lainnya (Secretary, demonstrator, dan distributor) memiliki mekanisme yang sama seperti info supervisor.

Berikutnya adalah halaman Notifikasi yang dapat diakses jika pengguna memilih tombol Notifikasi, maka akan muncul halaman dibawah



Halaman Notifikasi diatas menampilkan sub dari masing-masing pengguna dibawahnya yaitu report, aktivitas, dan bonus distributor. Notifikasi dengan titik merah (●) merupakan notifikasi yang baru masuk ke pengguna dan belum dilakukan tinjau, pengguna (manager) bisa melakukan peninjauan untuk jenis kegiatan yang masuk agar ditentukan apakah kegiatan tersebut ditampilkan di beranda aplikasi atau tidak, peninjauan dapat dilakukan dengan mengklik tombol Tinjau dari notifikasi.

Berikutnya adalah halaman Kelola manager yang bisa diakses jika pengguna (manager) memilih tombol kelola manager pada beranda, sehingga muncul tampilan dibawah



Pada halaman kelola manager, terdapat 3 tombol berbeda yaitu Aktivitas Supervisor, Report Demonstrator dan Bonus Distributor.

Pengguna (manager) bisa mengakses salah satu ketiga tombol tersebut untuk masuk ke halaman review dengan tampilan seperti dibawah:



Tampilan diatas merupakan daftar aktivitas yang menampilkan aktivitas-aktivitas yang sudah dirilis oleh supervisor, pengguna(manager) dapat mereviewnya



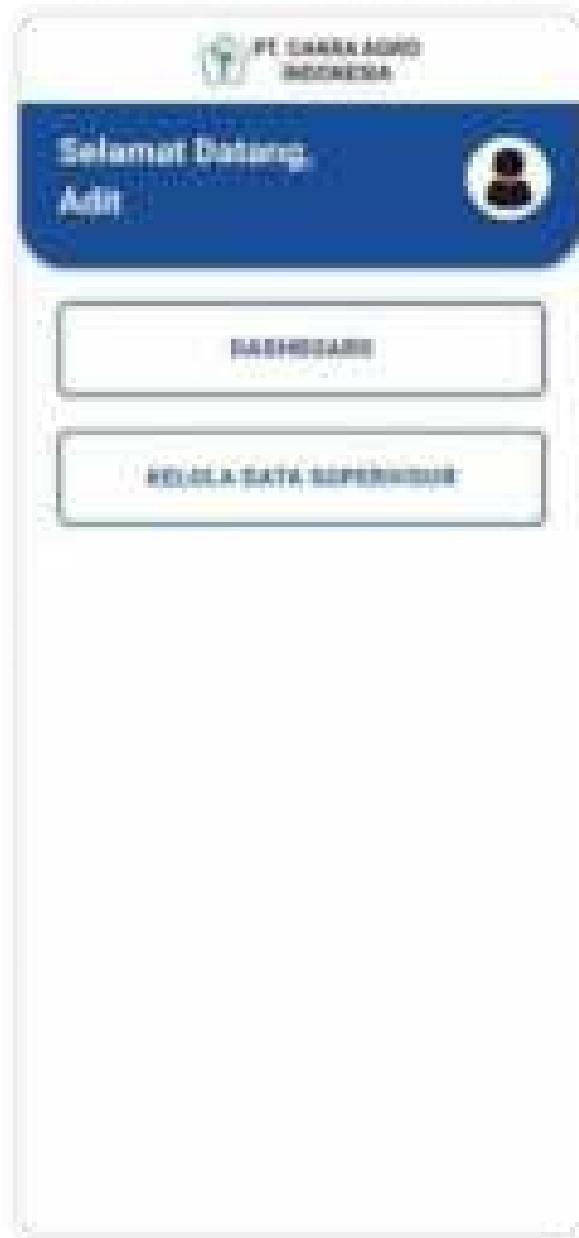
Tampilan diatas merupakan daftar report yang menampilkan report-report yang sudah dirilis oleh Demonstrator , pengguna(manager) dapat mereviewnya



Tampilan diatas merupakan daftar bonus yang menampilkan bonus distributor yang sudah dirilis oleh secretary , pengguna(manager) dapat mereviewnya

2.4 Supervisor

Setelah login dan masuk sebagai supervisor, maka tampilan awal pada dashboard akan seperti berikut



Gambar diatas merupakan tampilan beranda supervisor, terdapat dua tombol yang mengarah ke halaman berbeda, tombol pertama adalah dashboard yang akan mengarah ke halaman dashboard seperti dibawah :



Tampilan diatas merupakan halaman dashboard supervisor, terdapat dua tombol yang mengarah ke halaman berbeda yaitu info supervisor dan info demonstrator.

Info supervisor mengarahkan ke halaman supervisor (pengguna) seperti dibawah ini :



Tombol edit mengarah ke halaman untuk ubah info supervisor (pengguna), dengan tampilan seperti dibawah :



The screenshot displays the 'EDIT INFO SUPERVISOR' interface. At the top, there is a logo for 'PT. SARANA AGRO INDONESIA' and a back arrow. Below the title is a profile picture of a man in a suit and a 'SIMPAN' button. The form contains the following fields:

- Nama Supervisor:** [Input field]
- Alamat:** [Input field]
- No. Telp:** [Input field]
- Phone:** [Input field]
- Alamat:** [Input field]
- Kontak Personel:** [Input field]

At the bottom of the screen, there is a large green 'SIMPAN' button.

Pilih simpan untuk menyimpan perubahan pada info supervisor.

Berikutnya adalah info demonstrator yang akan menampilkan daftar demonstrator dari supervisor(pengguna), dengan tampilan seperti dibawah:

DEMONSTRATOR



00001
Kiki



00002
Yuni



00003
Alex



00004
Eysa



00005
Kiki



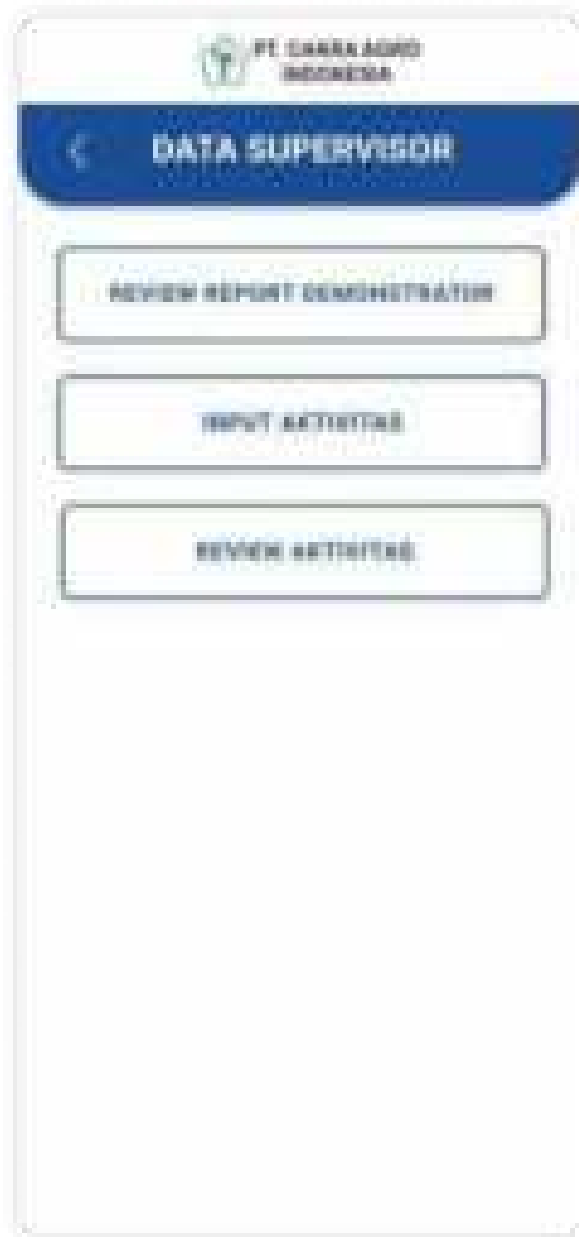
00006
Anni

Pengguna(supervisor) bisa melihat informasi detail dari demonstrator dengan memilih fotonya, maka akan masuk ke halaman info demonstrator seperti dibawah



Tampilan diatas merupakan detail dari info demonstrator yang dipilih.

Berikutnya adalah tampilan Kelola data supervisor yang bisa diakses dengan memilih tombol kelola data supervisor pada halaman yang sama dengan tombol dashboard, berikut tampilan kelola data supervisor



Tampilan diatas merupakan halaman data supervisor, terdapat tiga tombol pada halaman tersebut, yaitu tombol Review Report Demonstrator, Input Aktivitas, dan tombol Review Aktivitas.

Review Report Demonstrator menampilkan halaman dibawah

< DAFTAR REPORT

Departemen: **Wali** 000000

Departemen: **Sambung** 10

Departemen: **Wali** 000000

Departemen: **Wali** 10

Provinsi: **Jagabaya**

Kabupaten: **Saman**

Alamat: **997788888888**

Provinsi: **Saba**

Provinsi: **Wawa**

Provinsi: **Wawa**

Provinsi: **Wawa**

Kabupaten: **Saman**

Kabupaten: **Pantapan**

Provinsi: **Sabulaja**

Provinsi:



Input Aktivitas menampilkan halaman dibawah

The screenshot shows a mobile application interface for PT DARMA AGRI INDONESIA. The title is 'AKTIVITAS'. Below the title is a table with the following data:

Tgl. Mulai	2024
Tgl. Selesai	2024
Tgl. Mulai	2024
Tgl. Selesai	2024

Below the table are several input fields for text entry, including 'Kategori', 'Lokasi Kegiatan', 'Detail', 'Lokasi', 'Atribusi', and 'Status'. Below the form is a photo showing several people working in a field. At the bottom of the screen is a green button labeled 'Simpan'.

Pada halaman tersebut pengguna(supervisor) bisa melakukan input/menambahkan aktivitas

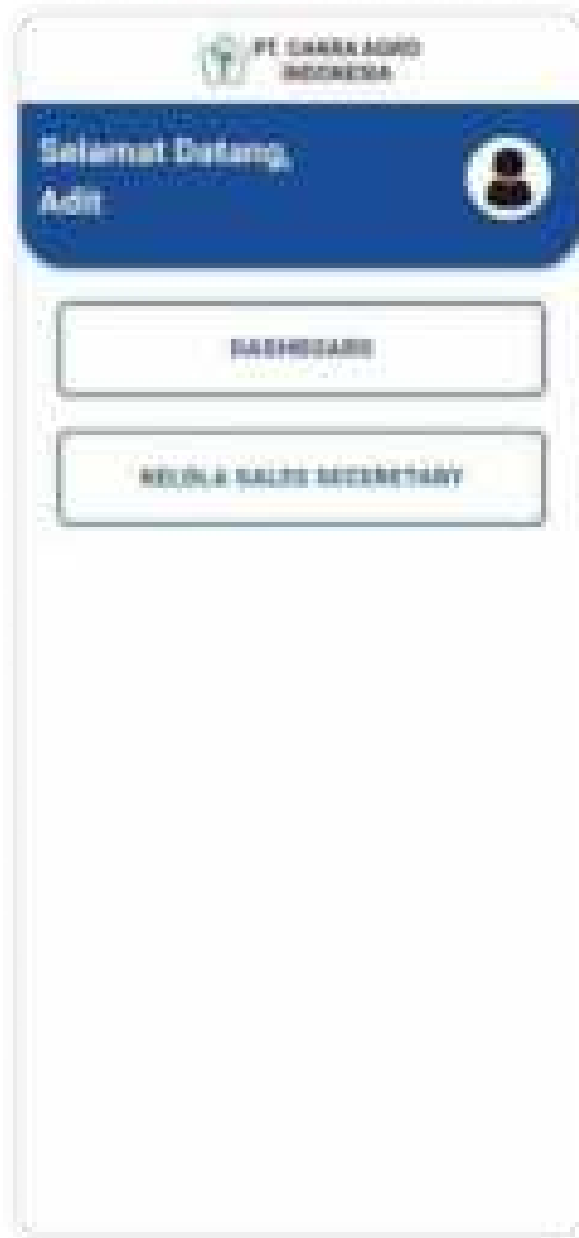
Review aktivitas akan menampilkan halaman dibawah



Pada halaman review aktivitas, pengguna(supervisor) bisa melakukan edit atau hapus dari Aktivitas yang sudah diinput.

2.5 Secretary

Setelah login dan masuk sebagai secretary, maka tampilan awal pada dashboard akan seperti berikut



Pada halaman beranda diatas terdapat dua tombol yaitu tombol dashboard dan Kelola sales secretary, yang masing-masing tombol mengarahkan ke halaman yang berbeda.

Tombol dashboard akan mengarahkan pengguna (Secretary) ke halaman dashboard, dengan tampilan seperti dibawah



Pada halaman dashboard terdapat dua tombol lainnya yaitu info sales secretary dan info distributor, info sales secretary akan mengarahkan ke halaman pengguna (Secretary) dengan tampilan seperti dibawah



Pengguna (Secretary) bisa melakukan perubahan terhadap profilnya dengan masuk ke halaman edit

PT. CARRA AGRO INDONESIA

EDIT INFO SECRETARY

Simpan

Nama Secretary

Alamat

Pegawai

No/Tempat Pendaftaran Pegawai

Simpan

Pengguna bisa menyimpan perubahan pada profil dengan memilih tombol Simpan

Berikutnya adalah tombol info distributor yang akan menampilkan daftar distributor dari Pengguna (Secretary), dengan tampilan seperti dibawah



Pada tampilan diatas pengguna (Secretary) bisa melihat detail info distributor dengan memilih foto distributor



Tampilan diatas adalah informasi detail dari distributor yang dipilih.

Berikutnya adalah tombol Kelola sales secretary yang akan mengarahkan pengguna ke halaman Kelola secretary seperti dibawah



Pada halaman data sales secretary terdapat dua tombol lainnya yaitu input bonus distributor, dan info review bonus distributor.

Tombol input bonus distributor akan mengarahkan pengguna(Secretary) ke halaman input bonus distributor

The screenshot shows a mobile application interface for 'PT CEMARA AGRO INDONESIA'. The title is 'BONUS DITRIBUTOR'. The form contains several input fields: a date field for 'Tanggal Bonus' (2022), a dropdown for 'Bulan Bonus' (2022), a dropdown for 'Tahun Bonus' (2022), a text field for 'Nomor Distributor', a text field for 'Nomor Pembelian', and a text field for 'Nomor Invoice'. Below the form is a photograph of two men standing next to a green tractor. At the bottom, there is a green button labeled 'Simpan'.

Tombol review bonus distributor akan mengarahkan pengguna(Secretary) ke halaman review bonus distributor



Pada halaman review bonus distributor, pengguna(secretary) bisa melakukan delete atau edit terhadap bonus distributor yang sudah diinput.

2.6 Demonstrator

Setelah login dan masuk sebagai Demonstrator, maka tampilan awal pada dashboard akan seperti berikut



Pada tampilan beranda diatas terdapat dua tombol yaitu tombol dashboard dan Kelola data demonstrator.

Tombol dashboard akan mengarahkan pengguna(Demonstrator) ke halaman dashboard info pengguna (Demonstrator) seperti dibawah



Pengguna (Demonstrator) bisa melakukan edit atau logout dengan dua tombol dibawah info demonstrator

Berikut contoh tampilan jika pengguna ingin melakukan perubahan pada profil pengguna

EDIT INFO DEMONSTRATOR



0000000000

Nama Demonstrator

Alamat

Area Tani

Phone

Password

Konfirmasi Password



Berikutnya adalah halaman Kelola data demonstrator



Terdapat dua tombol berbeda yaitu input data demo report (demplot) dan review demo report (demplot).

jika pengguna ingin menambahkan demplot baru maka bisa memilih input data demo report, sehingga akan muncul tampilan dibawah

The screenshot displays the 'INPUT REPORT' screen of a mobile application. At the top, there is a logo for 'ST. JEROME AGRI INDONESIA' and a back arrow. The title 'INPUT REPORT' is centered. Below the title is a form with the following fields: a date field (13- Mei 2022), a time field (08:00), a location field (Kampus ST. JEROME AGRI INDONESIA), a field for 'Nama Petani', a field for 'No. Telp', a field for 'No. HP', a field for 'No. Email', a field for 'No. WhatsApp', a field for 'No. Facebook', a field for 'No. Instagram', a field for 'No. Twitter', a field for 'No. LinkedIn', a field for 'No. YouTube', a field for 'No. Website', and a field for 'No. Blog'. Below the form is a photo of hands holding a tablet in a field. At the bottom, there is a green button labeled 'Simpan'.

jika pengguna ingin melakukan review demplot yang sudah diinput maka bisa memilih review data demo report, sehingga akan muncul tampilan dibawah

DAFTAR REPORT

Departemen: **Wahana** 000000

Divisi: **Sambung** ✓

Departemen: **Wahana** 000000

Divisi: **Wahana** ✓

Provinsi: **Jakarta**

Kabupaten: **Serang**

Alamat: **0000000000**

Provinsi: **Serang**

Provinsi: **000000**

Provinsi: **000000**

Provinsi: **000000**

Kabupaten: **Serang**

Kabupaten: **Prabumulia**

Provinsi: **Serang**



Provinsi: **Serang**



2.7 Distributor

Setelah login dan masuk sebagai Demonstrator, maka tampilan awal pada dashboard akan seperti berikut



Tampilan diatas adalah halaman beranda pengguna distributor.

Terdapat dua tombol yang masing-masing mengarahkan ke halaman yang berbeda, yaitu dashboard untuk melihat info pengguna(distributor) dan tombol Kelola data distributor untuk melihat bonus distributor.

Berikut adalah tampilan dashboard untuk melihat info pengguna(distributor)



Pengguna (Distributor) bisa melakukan perubahan terhadap profil yang dimiliki dengan edit dan logout untuk keluar

Berikut adalah tampilan bonus distributor jika pengguna memilih Kelola data distributor pada beranda



