

Revisi
4/11/25
Ardiansyah

LAPORAN PROYEK
SISTEM INFORMASI LAUNDRY BERBASIS WEB



Disusun Oleh :

Farel Cahyaning Bagaskara 2200018188

Syafiq Azhami Yusman 2200018161

M Reinza Chaniago 2200018390

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS AHMAD DAHLAN
2025

HALAMAN PENGESAHAN

SISTEM INFORMASI LAUNDRY BERBASIS WEB

Farell Cahyaning Bagaskara	2200018180
Syafiq Azhami Yusman	2200018161
M Reinza Chaniago	2200018390
Athallah Fadhil Ahmad	2200018175

PEMBIMBING : Dr. Ardiansyah, S.T., M.Cs.
NIPM : 19790723 200309 111 0932301



15/9 2025
<tgl>

PENGUJI: Drs. Wahyu Pujiyono M.Kom.
NIPM : 196611041991071110677540

W.  <tgl> 17/9/2025

Yogyakarta, 14 September 2025
Kaprodin S1 Informatika



Dr. Murinto, S.Si., M.Kom.
NIPM. 19730710 200409 111 0951298

KATA PENGANTAR

Puji dan syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga kami dapat menyelesaikan laporan proyek ini dengan baik dan lancar. Laporan ini disusun sebagai bagian dari pemenuhan tugas akhir pada mata kuliah Manajemen Proyek Teknologi Informasi, dengan judul “Sistem Informasi Laundry Berbasis Web”.

Laporan ini berisi proses perencanaan, pelaksanaan, dan evaluasi dari proyek pengembangan sistem informasi laundry yang bertujuan untuk membantu pengelolaan layanan laundry secara digital, khususnya dalam pemesanan layanan, pelacakan status pesanan, serta komunikasi antara pelanggan dan pihak pengelola laundry.

Kami menyadari bahwa dalam proses penyusunan laporan ini masih terdapat kekurangan dan keterbatasan. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang membangun demi kesempurnaan laporan ini di masa mendatang.

Ucapan terima kasih kami sampaikan kepada semua pihak yang telah membantu dalam proses pengerjaan proyek ini, khususnya kepada Dr. Ardiansyah, S.T., M.Cs, selaku dosen pembimbing yang telah memberikan arahan, bimbingan, serta motivasi yang sangat berarti selama proses pelaksanaan proyek ini.

Tak lupa kami juga mengucapkan terima kasih kepada seluruh anggota tim pengembang yang telah bekerja sama dengan baik, serta kepada semua pihak yang mendukung dan memberikan kontribusi dalam keberhasilan proyek ini.

Akhir kata, semoga laporan ini dapat memberikan manfaat dan menjadi referensi yang berguna bagi siapa saja yang berkecimpung dalam bidang pengembangan sistem informasi, khususnya dalam konteks layanan jasa laundry.

Akhir kata, kami berharap laporan ini dapat memberikan manfaat bagi semua pihak yang membacanya dan dapat menjadi kontribusi positif dalam bidang manajemen proyek teknologi informasi, khususnya dalam pemantauan data prasarana sekolah dasar.

Demikian kata pengantar ini kami sampaikan. Terima kasih.

Yogyakarta, 24 Juli 2025

Hormat Kami

SentryTEAM

DAFTAR ISI

COVER	1
HALAMAN PENGESAHAN.....	1
KATA PENGANTAR	2
DAFTAR ISI	4
DAFTAR TABEL	6
DAFTAR GAMBAR	7
BAB I Pendahuluan.....	8
A. Latar Belakang	8
B. Project Charter	9
C. Tujuan	11
D. Ruang Lingkup	11
E. Stakeholder	12
BAB II Perencanaan Proyek.....	14
A. Analisis Kelayakan	14
B. Work Breakdown Structure.....	15
C. Kebutuhan Sumber Daya.....	17
D. Sumber Daya Fisik.....	19
E. Rencana Jadwal Pelaksanaan Proyek	20
F. Rencana Nilai Proyek.....	21
BAB III Pelaksanaan Proyek	24
A. Realisasi Jadwal Pelaksanaan	24
B. Realisasi Hasil Pekerjaan	25
C. Penjaminan Kualitas Proyek	46
D. Keberlanjutan Proyek	47
BAB IV PENUTUP	48
A. Kesimpulan	48
B. Saran	48
LAMPIRAN	50
1. Proposal Proyek	50
2. Kontrak Kerja dengan Mitra	55
3. Logbook Kelompok.....	63

4. Logbook Individu	69
5. Foto Dokumentasi Kegiatan Proyek	90
6. Berita Acara	94
7. Bukti Pembiayaan	96
8. Tools.....	97
9. Video Produk Luaran.....	1897
10. Poster	190
11. Link Slide Presentasi	191

DAFTAR TABEL

Tabel 1.1 Project Charter	9
Tabel 2. 1Tabel SWOT	14
Tabel 2. 2 Rencana Jadwal Pelaksanaan Proyek.....	20
Tabel 2. 3 Pemasukan	22
Tabel 2. 4 Pengeluaran	22
Tabel 2. 5 Gaji	23
Tabel 2. 6 Rekapitulasi Dana	23
Tabel 3. 1 Realisasi Jadwal Pelaksanaan	24
Tabel 3. 2 Realisasi Pembagian Tugas Project Manager, Quality Assurance	25
Tabel 3. 3 Gantt Chart.....	26
Tabel 3. 4 Realisasi Tugas Sekretaris & System Analyst.....	27
Tabel 3. 5 Persyaratan Fungsional	28
Tabel 3. 6 Persyaratan Non-Fungsional	29
Tabel 3. 7 Realisasi Tugas UI/UX Designer	33
Tabel 3. 8 Realisasi Tugas Frontend Developer	38
Tabel 3. 9 Realisasi Tugas Backend Developer	42
Tabel 4. 1 Logbook Kelompok.....	63
Tabel 4. 2 Logbook Project Manager & Quality Assurance	69
Tabel 4. 3 Sekretaris & Backend.....	74
Tabel 4. 4 Logbook UI/UX Designer	78
Tabel 4. 5 Logbook Frontend Developer	87

DAFTAR GAMBAR

Gambar 3. 1 Use Case Diagram	30
Gambar 3. 2 ERD	31
Gambar 3. 3 Activity Diagram	31
Gambar 3. 4 Design Halaman Login	34
Gambar 3. 5 Design Halaman Landing Page.....	34
Gambar 3. 6 Design Halaman Buat Pesanan.....	36
Gambar 3. 7 Design Riwayat Pesanan	36
Gambar 3. 8 Design Halaman Detail Pesanan.....	37
Gambar 3. 9 Design Halaman Bantuan FAQ.....	37
Gambar 3. 10 Hasil Implementasi Halaman Login	39
Gambar 3. 11 Hasil Implementasi Halaman Admin	39
Gambar 3. 12 Hasil Implementasi Halaman Daftar Pesanan	39
Gambar 3. 13 Hasil Implementasi Halaman Edit Pesanan	40
Gambar 3. 14 Hasil Implementasi Halaman Laporan Pesanan	41
Gambar 3. 15 Hasil Implementasi Halaman Service	41
Gambar 3. 16 Hasil Implementasi Fitur Cetak Laporan	42
Gambar 3. 17 Struktur CRUD	43
Gambar 3. 18 Fungsi Kategori.....	43
Gambar 3. 19 Kelola Data User.....	44
Gambar 3. 20 Routing Laundry	45
Gambar 3. 21 Rute Pengguna	45
Gambar 4. 1 Bimbingan Bersama Dosen Pembimbing	90
Gambar 4. 2 Dokumentasi Rapat /Pengerjaan Online/Offline.....	91
Gambar 4. 3 Presentasi ke Mitra.....	92
Gambar 4. 4 Serah Terima Website	93
Gambar 4. 5 Pembayaran Mitra	96
Gambar 4. 6 Pengeluaran Mitra	97
Gambar 4. 7Poster.....	190

BAB I

Pendahuluan

A. Latar Belakang

Dalam era digital yang terus mengalami perkembangan pesat, teknologi informasi telah menjadi bagian tak terpisahkan dari kehidupan manusia, termasuk dalam dunia bisnis. Salah satu inovasi teknologi yang paling berperan penting adalah website. Website kini tidak hanya menjadi media promosi, tetapi juga menjadi sarana interaksi, komunikasi, dan manajemen informasi antara pelaku usaha dan konsumennya. Dalam konteks ini, pemanfaatan website telah merambah ke berbagai sektor, termasuk sektor jasa seperti usaha laundry.

Website memberikan kemudahan akses informasi bagi pelanggan, memungkinkan mereka untuk melihat layanan, harga, hingga melakukan pemesanan tanpa harus datang langsung ke lokasi. Hal ini tentu saja sejalan dengan perubahan perilaku konsumen modern yang lebih mengandalkan internet untuk mencari informasi, melakukan pemesanan layanan, dan mengatur aktivitas sehari-hari secara lebih praktis dan efisien.

Dengan adanya Sistem Informasi Manajemen berbasis website, sebuah usaha laundry tidak hanya dapat meningkatkan efisiensi operasional, tetapi juga memberikan pengalaman yang lebih baik bagi pelanggannya. Sistem ini dapat mencakup fitur pemesanan layanan secara online, pelacakan status cucian, pengaturan jadwal antar-jemput, hingga laporan keuangan usaha secara digital. Fitur-fitur tersebut mampu mengurangi kesalahan pencatatan manual, mempercepat proses pelayanan, serta meningkatkan transparansi dan kepercayaan pelanggan terhadap usaha tersebut.

Di tengah meningkatnya persaingan antar usaha kecil dan menengah, inovasi menjadi kunci untuk tetap unggul dan relevan. Usaha laundry, meskipun tergolong usaha mikro atau kecil, tetap dituntut untuk beradaptasi dengan perkembangan teknologi demi meningkatkan kualitas layanan dan daya saing. Dengan memanfaatkan sistem informasi berbasis website, usaha laundry dapat memperluas jangkauan pasar, mengatur manajemen internal dengan lebih rapi, dan memberikan layanan yang cepat serta akurat.

Penelitian ini akan membahas secara mendalam tentang bagaimana Sistem Informasi Manajemen Usaha Laundry Berbasis Website dapat menjadi solusi yang efektif bagi pengelolaan usaha laundry. Fokus penelitian ini adalah merancang sistem yang mampu membantu proses bisnis laundry secara menyeluruh, mulai dari manajemen pelanggan, pengelolaan transaksi, hingga pelaporan keuangan yang terintegrasi dalam satu platform berbasis web.

Selain itu, penelitian ini juga akan membahas tantangan yang mungkin dihadapi dalam penerapan sistem ini, seperti keterbatasan sumber daya manusia, biaya implementasi awal, serta bagaimana cara membangun kepercayaan pelanggan dalam penggunaan sistem digital. Melalui pemahaman yang lebih baik terhadap kebutuhan dan permasalahan di lapangan, diharapkan solusi yang dirancang dapat menjawab kebutuhan pelaku usaha dan memberikan manfaat yang nyata dalam pertumbuhan bisnis mereka.

Dengan demikian, proyek ini tidak hanya menjadi media pembelajaran penerapan ilmu yang diperoleh selama perkuliahan, tetapi juga berpotensi menjadi kontribusi nyata dalam membantu UMKM, khususnya di bidang jasa laundry, untuk bertransformasi secara digital dan menghadapi tantangan industri di era modern ini.

B. Project Charter

Tabel 1.1 Project Charter

Project name: Mamak Laundry	
Project Description:	• Aplikasi mobile untuk memesan layanan laundry pada home laundry. Aplikasi ini memungkinkan pengguna untuk memesan, melacak status cucian mereka, dan melakukan pembayaran melalui aplikasi.
Project Objective:	• Memfasilitasi pemesanan dan pelacakan layanan laundry melalui platform mobile. • Menyediakan antarmuka yang ramah pengguna bagi pelanggan untuk berinteraksi dengan penyedia laundry. • Pastikan layanan laundry cepat dan efisien dengan pembaruan real-time untuk pengguna

Project is considered successful when:	- Aplikasi ini selesai seperti yang kami harapkan. - Aplikasi bermanfaat bagi green home laundry. - Jumlah pesanan laundry yang dilakukan melalui aplikasi meningkat sebesar 7,5% setelah dua bulan		
Project Participants:	Title:	Name:	
	Investor Project Manager Financial Manager Team Members: Database Analysis Design Programming	Dyah Irma Gendraningtyas Farell Cahyaning Bagaskara M Reinza Chaniago Farell Cahyaning Bagaskara Syafiq azhami yusman Syafiq Azhami Yusman	
Available resources:	Rp. 1.500.000 Testing environment: samsung a54,iphone 11,iphone xr,		
Milestones:	Status :	Due:	Deadline:
	Build 1.0 interview klient	12/3/2025	15/7/2025
	Build 2.0 membuat tampilan menu	11/5/2025	12/6/2025
Potential risks:	Melewatkan tenggat waktu karena penambahan fitur yang tidak direncanakan.		
Approval:	Title and name:	Date:	
	M Reinza Chaniago Signature_____	09/07/2025	
Signature_____			
Syafiq Azhami Yusman Signature_____			
Farell Cahyaning Bagaskara Signature_____			

C. Tujuan

Proyek ini bertujuan untuk meningkatkan efisiensi layanan dan aksesibilitas informasi di Mamak Laundry melalui pengembangan sistem informasi laundry berbasis web. Sistem ini dirancang untuk memberikan kemudahan kepada pelanggan dalam melakukan pemesanan layanan, memantau status cucian, serta memperoleh informasi terkait layanan yang tersedia tanpa harus datang langsung ke lokasi. Dengan adanya sistem ini, pelanggan dapat melakukan pemesanan secara online, menerima notifikasi status laundry secara otomatis melalui WhatsApp, serta melihat riwayat transaksi mereka dengan mudah. Selain itu, sistem ini juga membantu pihak pengelola dalam mengelola data pelanggan, pesanan, transaksi pembayaran, serta memantau progres pekerjaan seperti penjemputan, pencucian, penjemuran, penyetrikaan, hingga pengantaran.

Fokus utama dari proyek ini adalah meningkatkan kualitas layanan, mempercepat proses operasional, dan menghadirkan kemudahan dalam interaksi antara pelanggan dan pengelola laundry. Sistem juga dirancang dengan antarmuka yang ramah pengguna dan responsif, sehingga memudahkan pengguna untuk menavigasi website melalui perangkat apa pun, baik itu komputer maupun ponsel.

D. Ruang Lingkup

Ruang lingkup proyek ini dimulai dengan analisis kebutuhan, di mana tim melakukan identifikasi dan dokumentasi terhadap kebutuhan fungsional maupun non-fungsional sistem berdasarkan tujuan proyek dan harapan dari pemilik usaha Mamak Laundry. Tahap ini penting untuk memastikan bahwa sistem informasi yang dibangun benar-benar sesuai dengan kebutuhan operasional usaha dan kenyamanan pelanggan.

Setelah analisis kebutuhan selesai, proyek dilanjutkan dengan tahap perancangan dan pengembangan sistem informasi laundry berbasis web. Pada tahap ini, arsitektur sistem, alur proses, dan antarmuka pengguna akan dirancang dengan mempertimbangkan kemudahan penggunaan, kecepatan akses, serta fitur-fitur utama seperti pemesanan layanan, pelacakan status laundry, dan

notifikasi otomatis via WhatsApp.

Proyek kemudian memasuki fase pengujian dan penjaminan kualitas, di mana seluruh fungsi sistem akan diuji secara menyeluruh untuk memastikan berjalan sebagaimana mestinya. Pengujian mencakup uji fungsionalitas (seperti pemesanan dan update status cucian), uji keamanan data pelanggan, dan uji performa website agar sistem siap digunakan dalam lingkungan operasional sesungguhnya.

Setelah sistem dinyatakan siap, proyek masuk ke tahap deployment, yaitu pengaktifan website ke server produksi dan dapat diakses oleh pelanggan Mamak Laundry secara online. Pelanggan dapat mulai menggunakan layanan secara langsung melalui platform tersebut.

Terakhir, ruang lingkup proyek juga mencakup sosialisasi dan pelatihan penggunaan sistem kepada pemilik dan karyawan laundry agar mereka dapat menjalankan sistem dengan efektif. Tujuannya agar semua pihak yang terlibat memahami cara mengoperasikan sistem secara optimal, mulai dari mengelola data pesanan hingga menanggapi notifikasi pelanggan.

E. Stakeholder

Berikut adalah daftar stakeholder yang terlibat dalam proyek Sistem Informasi Laundry berbasis web ini:

- Sponsor proyek adalah Dyah Irma Gendraningtyas selaku pemilik Mamak Laundry. Beliau memberikan pendanaan, persetujuan akhir sistem, dan menjadi pengambil keputusan utama dalam proyek ini.
- Fresh and Clean Laundry sebagai klien ke 2
- Bintan Express Laundry sebagai klien ke 3
- Bungku Wash & Dry sebagai klien ke 4
- Sinar Bersih Laundry sebagai klien ke 5
- Cepat Berkah Laundry sebagai klien ke 6
- Manajer proyek bertanggung jawab atas pengelolaan dan pengawasan keseluruhan proses pengembangan sistem, mulai dari perencanaan, koordinasi antar anggota tim, hingga pelaporan progres kepada sponsor.

- Tim proyek terdiri dari:
 - Analisis sistem yang bertugas menyusun kebutuhan fungsional dan non-fungsional sistem berdasarkan hasil wawancara

dengan pemilik laundry dan mengubahnya menjadi spesifikasi teknis.

- Desainer UI/UX, yang merancang antarmuka website agar mudah digunakan oleh pelanggan dari berbagai latar belakang, dengan fokus pada kemudahan navigasi dan tampilan yang responsif.
- Pengembang frontend, yang membangun tampilan dan interaksi pengguna di bagian depan sistem, seperti halaman pemesanan, status cucian, dan formulir kontak.
- Pengembang backend, yang mengembangkan sistem di balik layar, seperti pengelolaan data transaksi laundry, user admin, serta integrasi dengan layanan notifikasi WhatsApp.
- Quality Assurance (QA), yang menguji sistem secara menyeluruh untuk memastikan semua fungsi berjalan sesuai kebutuhan, tanpa bug, dan siap digunakan oleh pelanggan.
- Staf Mamak Laundry akan menjadi pengguna internal yang mengelola pemesanan masuk, memperbarui status cucian, dan melakukan verifikasi pembayaran melalui sistem.
- Pelanggan Mamak Laundry adalah pengguna akhir sistem yang akan melakukan pemesanan, memantau status cucian mereka, dan menerima notifikasi otomatis melalui sistem informasi berbasis web ini.

BAB II

Perencanaan Proyek

A. Analisis Kelayakan

Tabel 2. 1Tabel SWOT

STRENGTHS	WEAKNESS
● Kemudahan Penggunaan: Aplikasi menyediakan pemesanan, pelacakan status, dan pembayaran dalam satu platform, memudahkan pengguna dalam mengakses layanan laundry. ● Layanan Antar-Jemput: Memberikan kenyamanan bagi pelanggan yang sibuk, sehingga meningkatkan loyalitas dan kepuasan pelanggan. ● Penghematan Waktu: Dengan pelacakan status cucian dan notifikasi otomatis, pelanggan tahu kapan cucian mereka siap, yang meningkatkan pengalaman pengguna. ● Pembayaran Terintegrasi: Aplikasi mendukung metode pembayaran digital yang ● memudahkan proses transaksi dan mengurangi risiko pembayaran tunai.	● Biaya Pengembangan dan Pemeliharaan: Pembuatan dan pembaruan aplikasi memerlukan biaya, baik dalam pengembangan fitur baru maupun keamanan data. ● Keterbatasan Pengguna Baru: Tidak semua pelanggan laundry terbiasa dengan aplikasi mobile, yang dapat menjadi hambatan dalam adopsi awal ● Ketergantungan pada Koneksi Internet: Aplikasi memerlukan internet untuk pelacakan dan pembayaran, yang bisa jadi kendala.

OPPORTUNITIES	TREATS
● Peningkatan Permintaan Layanan Digital: Tren digitalisasi di kalangan masyarakat yang sibuk menciptakan kebutuhan untuk layanan laundry yang mudah diakses secara online. ● Kolaborasi dengan E-Wallet : Dapat bekerja sama dengan platform pembayaran seperti Gopay, OVO, atau Dana untuk memudahkan transaksi ● Ekspansi Layanan di Area Lain: Kemungkinan memperluas jangkauan layanan ke wilayah baru dengan layanan antar-jemput yang sudah terintegrasi.	● Persaingan yang Tinggi: Banyak aplikasi dan platform lain yang menawarkan layanan serupa, sehingga Green Home Laundry harus selalu berinovasi untuk menarik perhatian pelanggan. ● Ketergantungan pada Teknologi: Jika ada masalah teknis atau server, maka layanan dapat terganggu, yang mempengaruhi pengalaman pelanggan. ● Ancaman Keamanan Data: Risiko kebocoran data pribadi dan data pembayaran pelanggan bisa mengurangi kepercayaan pelanggan pada aplikasi.

B. Work Breakdown Structure

1. Konsep dan inisiasi proyek
 - 1.1. Menentukan proposal, WBS dan Gantt Chart
 - 1.2. Membuat Project Charters
 - 1.2.1. Kebutuhan analisis
 - 1.2.2. Tujuan Proyek
 - 1.2.3. Faktor/kesuksesan kritis
 - 1.2.4. Asumsi proyek
 - 1.2.5. Kendala proyek
 - 1.2.6. Hasil proyek
 - 1.3. Perencanaan Proyek
 - 1.3.1. Mengidentifikasi tim proyek (client)
 - 1.3.2. Menyusun jadwal proyek

- 1.3.3. Menentukan anggaran dan MOU
- 2. Analisis Proyek
 - 2.1. Analisis Kebutuhan
 - 2.1.1. Identifikasi Kebutuhan Pengguna
 - 2.1.2. Penyusunan Spesifikasi Fungsional dan Non-Fungsional
 - 2.2. Rancangan Sistem
 - 2.2.1. Membuat Use Case Sistem
 - 2.2.2. Membuat Entity Relationship Diagram (ERD)
 - 2.2.3. Membuat Activity Diagram
- 3. Desain Sistem
 - 3.1. Merancang Sistem
 - 3.1.1. Melakukan Research Plan
 - 3.1.2. Interview
 - 3.1.3. User Persona dan Empathy Map
 - 3.1.4. Brainstorming
 - 3.2. Merancang Antarmuka Pengguna
 - 3.2.1. Desain sistem
 - 3.2.2. Brand guideline
 - 3.2.3. Sitemap
 - 3.2.4. Membuat wireframe
 - 3.2.5. UI Design
 - 3.2.6. Usability Testing
- 4. Pengembangan Aplikasi
 - 4.1. Desain Sistem
 - 4.1.1. Penerapan Antarmuka Pengguna (UI)
 - 4.1.2. Penerapan Arsitektur Website
 - 4.2. Pengembangan Frontend
 - 4.2.1. Pembuatan Halaman Utama
 - 4.2.2. Pembuatan Halaman Admin
 - 4.2.3. Integrasi Responsif
 - 4.3. Pengembangan Backend

- 4.3.1. Pembuatan Database
 - 4.3.2. Pembuatan fungsi-fungsi Backend
 - 4.3.3. Implementasi koneksi database
- 4.4. Integrasi Frontend dan Backend
 - 4.4.1. Debugging dan koreksi
- 5. Pengujian
 - 5.1. Persiapan Pengujian
 - 5.1.1. Penyusunan rencana pengujian
 - 5.2. Pengujian Fungsional
 - 5.2.1. Pengujian I/O pada Website
 - 5.2.2. Pengujian fungsionalitas setiap fitur
 - 5.3. Pengujian keseluruhan sistem
 - 5.3.1. Pengujian fungsionalitas sistem secara keseluruhan
 - 5.4. Pengujian Keamanan
 - 5.5. Dokumentasi dan Pelaporan
 - 5.5.1. Penyusunan dokumentasi hasil pengujian
 - 5.5.2. Penyusunan laporan Kesimpulan dan rekomendasi
- 6. Implementasi Website
 - 6.1. Pelatihan pengguna akhir
 - 6.2. Persiapan rencana peluncuran
 - 6.3. Peluncuran website
- 7. Evaluasi dan penyesuaian
 - 7.1. Evaluasi kinerja website
 - 7.2. Pengumpulan masukan pengguna
 - 7.3. Perbaikan dan Pembaharuan
- 8. Penyelesaian dan Penutupan

C. Kebutuhan Sumber Daya

- 1. Sumber Daya Manusia
 - Nama: Farel Cahyaning Bagaskara
 - Peran: Ketua Tim (Project Manager & Marketing)

NIM: 2100018188

Tugas:

1. Membentuk tim dan mendistribusikan tugas sesuai keahlian anggota.
2. Memimpin rapat rutin dan mengawasi pelaksanaan proyek.
3. Menandatangani MoU dan menjadi penghubung komunikasi dengan pemilik laundry (klien).
4. Memastikan semua modul dikembangkan sesuai target dan standar kualitas.
5. Menyusun strategi pemasaran dan presentasi akhir ke klien.

Nama: Syafiq Azhami Yusman

Peran: Sekretaris & Backend Developer

NIM: 2100018161

Tugas:

1. Mengembangkan fitur backend termasuk pengelolaan pesanan, pengguna, layanan laundry, dan status transaksi.
2. Menghubungkan sistem dengan layanan pihak ketiga seperti API WhatsApp (Fonnte) untuk notifikasi otomatis.
3. Mengelola database: perancangan skema tabel, relasi antar data, dan optimasi query.
4. Mengintegrasikan sistem konfirmasi pembayaran manual.
5. Menyiapkan dokumentasi teknis dan laporan pengembangan backend.
6. Mencatat log aktivitas anggota dan membuat notulen rapat.

Nama: M. Reinza Chaniago

Peran: System Analyst & Bendahara

NIM: 2100018390

Tugas:

1. Menganalisis kebutuhan pelanggan terhadap sistem laundry.
2. Membuat Use Case Diagram, Activity Diagram, dan DFD berdasarkan hasil wawancara dengan pemilik laundry.
3. Merancang kebutuhan fungsional dan non-fungsional sistem.
4. Menyusun dan mengatur time line kerja proyek.
5. Menyusun dan mencatat rencana anggaran dan pengeluaran tim selama proyek berlangsung.

Nama: Syafiq Azhami Yusman

Peran: UI/UX Designer & Frontend Developer

NIM: 2100018375

Tugas:

1. Mendesain antarmuka website yang user-friendly dan responsif.
2. Mengimplementasikan halaman frontend seperti dashboard pelanggan, halaman pemesanan, dan riwayat pesanan.
3. Menyesuaikan tampilan berdasarkan prinsip UI/UX untuk mempermudah pelanggan memesan layanan.
4. Berkolaborasi dengan backend untuk menyinkronkan tampilan dan data.
5. Menambahkan fitur interaktif seperti kalkulasi estimasi biaya otomatis dan form order.

D. Sumber Daya Fisik

1. Sistem IT dan Komunikasi
 - Laptop untuk setiap anggota tim
 - Koneksi Internet/Wi-Fi stabil untuk koordinasi dan pengembangan

- Akun WhatsApp aktif untuk pengujian integrasi notifikasi
 - Perangkat seluler (smartphone) untuk simulasi pesanan dari pelanggan
2. Kebutuhan Perangkat Lunak
- Visual Studio Code
 - Figma
 - Xampp
 - Draw.io dan Astah
 - Microsoft Word
 - Browser

E. Rencana Jadwal Pelaksanaan Proyek

Tabel 2. 2 Rencana Jadwal Pelaksanaan Proyek

No.	Nama Kegiatan	Maret				April				Mei				Juni				Juli				Agustus			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	Menentukan Jobdesc																								
2.	Menghubungi client (Kajian Awal)																								
3.	Pembuatan Proposal.																								
4.	Melakukan penyerahan																								
	Mou dengan client																								
5.	Identifikasi kebutuhan pengguna																								
6.	Perancangan UI/UX																								

c. Biaya proyek

Berikut adalah rincian biaya yang diperlukan untuk pembuatan proyek website “Sistem Informasi Laundry Berbasis Website” :

Tabel 2. 3 Pemasukan

No.	Keterangan	Harga	Jumlah
1.	Pembayaran sistem dari client	1.500.000	1.500.000
Total Pemasukan (Rp)			1.500.000

Tabel 2.3 merupakan pemasukan yang didapat oleh tim Sentry, terdapat perbedaan pemasukan dari biaya yang diajukan diawal dikarenakan belum terjadi kesepakatan dengan mitra. Ini bisa menjadi evaluasi untuk kedepannya.

Tabel 2. 4 Pengeluaran

No.	Uraian	Detail Rincian	Harga (Rp)
1.	Hosting Web di Rumah Web Host	Kuota Hosting	30.000
2.	Pembelian domain di Rumah Web Host	.my.id	20.000
3.	Pemrograman		25.000
4.	Pemeliharaan		25.000
Total pengeluaran (Rp)			Rp. 100.000

Tabel 2.4 merupakan rincian biaya pengeluaran tim Sentry yang didapatkan dari pemasukan. Terlihat bahwa seluruh biaya yang diterima telah dialokasikan sesuai kebutuhan untuk pengembangan website

Tabel 2. 5 Gaji

No.	Bagian	Total jam Kerja	Gaji (Rp)	Total Gaji (Rp)
1.	Project Manager /Ketua Tim		-	-
2.	Sekretaris & Backend Developer		-	-
3.	UI/UX Designer/ Frontend Developer		-	-
4.	Bendahara /System Analis		-	-
Total pengeluaran (Rp)			-	-

Tabel 2.5 merupakan tabel gaji dan total jam kerja tim Sentry

Tabel 2. 6 Rekapitulasi Dana

Dana	Jumlah (Rp)
1. Pemasukan	100.000
a. Pembayaran sistem dari klien	100.000
Total Pemasukan (Rp) :	
2. Dana yang digunakan	
a. Total pengeluaran	100.000
b. Total gaji anggota	
Total Anggaran (Rp) :	
3. Sisa Dana	
a. Total pemasukan	
b. Total dana yang digunakan	
Anggaran Sisa Pengeluaran (Rp) :	

Tabel 2.6 merupakan rekapitulasi dana dari tim Sentry

BAB III Pelaksanaan Proyek

A. Realisasi Jadwal Pelaksanaan

Tabel 3. 1 Realisasi Jadwal Pelaksanaan

No.	Nama Kegiatan	Maret				April				Mei				Juni				Juli				Agustus			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	Menentukan Jobdesc																								
2.	Menghubungi client (Kajian Awal)																								
3.	Pembuatan Proposal.																								
4.	Melakukan penyerahan Mou dengan client																								
5.	Identifikasi kebutuhan pengguna																								
6.	Perancangan UI/UX																								
7.	Proses pembuatan sistem website																								
8.	Finishing (testing) Aplikasi																								
9.	Uji Coba																								
10.	Pemasangan Hosting																								

Tabel 3.2 merupakan hasil realisasi pekerjaan dari Project Manager
& Quality Assurance

Hasil Pembuatan Jadwal Proyek (Gantt Chart)

Tabel 3. 3 Gantt Chart

No	Nama	Tugas/Activity	Tanggal Mulai	Tanggal Selesai
1	Syafiq Azhami Yusman	Desain UI/UX	2025-03-18	2025-4-22
2	Syafiq Azhami Yusman	Riset Pengalaman Pengguna	2025-03-23	2025-5-27
3	Syafiq Azhami	Front-End Development	2025-03-28	2025-5-07
4	M Reinza Chaniago	Mendesain Endpoint RESTful	2025-04-08	2025-5-17
5	M Reinza Chaniago	Implementasi Skema Database	2025-04-18	2025-6-22
6	Syafiq Azhami	Pengujian Otomatis	2025-04-23	2025-6-27
7	Syafiq Azhami	Penyusunan Kasus Uji	2025-04-28	2025-6-01
8	Syafiq Azhami	Pelaporan Bug	2025-05-05	2025-07-08
9	Farel Cahyaning	Perencanaan Proyek	2025-05-09	2025-07-13
10	Farel Cahyaning	Pembagian Tugas Delegasi	2025-05-14	2025-07-18
11	Farel Cahyaning	Penyusunan Timeline Proyek	2025-06-19	2025-07-23

Tabel 3.3 adalah jadwal proyek (Gantt Chart) Sistem Informasi Laundry Berbasis Website Mamak Laundry PP. Berdasarkan ganttchart waktu realisasi pelaksanaan terdapat kendala waktu selama proses pengembangan web dikarenakan Kelompok Kurang Kompak.

2. System Analyst & Bendahara

Nama : M Reinza Chaniago

NIM 2200018390

Tabel 3. 4 Realisasi Tugas Sekretaris & System Analyst

No	Agenda Kerja	Hari,Tanggal	Waktu	Hasil
1.	Membuat persyaratan fungsional dan non-fungsional yang harus dipenuhi oleh sistem yang akan dikembangkan.	25 April 2025	120 Menit	Membuat semua kebutuhan fungsional dan non-fungsional sistem
2.	Mencatat hasil wawancara tim dengan client MPTI	29 April 2025	120 Menit	Hasil wawancara client kami mendapatkan pengetahuan tentang permintaan client, data yang digunakan, serta fitur yang diinginkan mitra.
3.	Mencatat hasil pertemuan awal dengan dosen pembimbing.	30 April 2025	60 Menit	Hasil pertemuan awal dengan dosen pembimbing meminta masukan terkait MOU yang sudah dibuat, serta diberikan arahan terhadap fitur-

				fitur akan tim kami buat.
4	Membuat notulensi rapat sebagai dokumentasi hasil dari setiap pertemuan yang dilakukan.	Fleksibel	120 Menit	Hasil rapat mingguan progress sentry
5.	Membuat Use Case Diagram	26 Mei 2025	120 Menit	Membuat dokumen use case
6.	Pembuatan ERD	27 Mei 2025	120 Menit	Membuat dokumen ERD
7.	Pembuatan Activity Diagram	27 April 2025	120 Menit	Membuat dokumen Activity Diagram

Membuat persyaratan fungsional dan non-fungsional yang harus dipenuhi oleh sistem yang akan dikembangkan

Tabel 3. 5 Persyaratan Fungsional

No.	Kode	Deskripsi
1.	AL-F1	Registration – Pengguna dapat melakukan registrasi akun di aplikasi dengan memasukkan data pribadi seperti email, username dan password.
2.	AL-F2	Login – Pengguna dapat melakukan login akun pada website setelah melakukan registrasi dengan memasukkan email dan password
3.	AL-F3	User dapat melihat koleksi buku perpustakaan secara online
4.	AL-F4	User dapat mencari data kategori buku dan Buku

5.	AL-F5	User admin dapat Mengelola akun staff yaitu membuat, melihat, menambah, menghapus, serta mencetak laporan buku dan kategori buku.
6.	AL-F6	User dapat melakukan Logout akun pada website
7.	AL=F7	User dapat menginput profile akun staff

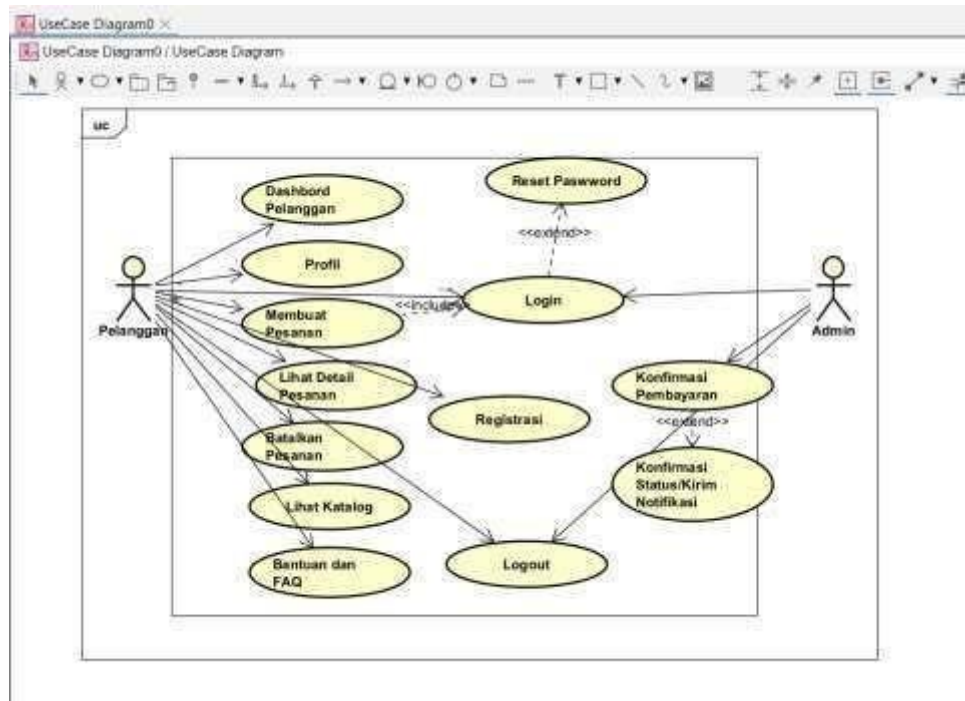
Tabel 3. 6 Persyaratan Non-Fungsional

No.	ID	Parameter	Kebutuhan
1.	AL-NF-01	Availability	Ketersediaan aplikasi untuk dapat diakses oleh pengguna - 24 jam
2.	AL-NF-02	Reliability	kehandalan aplikasi, termasuk aspek teknis seperti koneksi, kebutuhan perangkat keras.
3.	AL-NF-03	Ergonomy	Minimalis, menarik dan desain aplikasi disesuaikan dengan kenyamanan pengguna.
4.	AL-NF-04	Portability	N/A
5.	AL-NF-05	Memory	Ukuran sedang
6.	AL-NF-06	Response time	Aplikasi harus mampu merespon request dari user dalam waktu yang cepat dan tidak menimbulkan rasa frustrasi.
7.	AL-NF-07	Safety	Keamanan data dari aplikasi, serta penggunaan aplikasi dan melindungi data pengguna dari ancaman luar
8.	AL-NF-08	Security	Jaringan dan data
9.	AL-NF-09	Komunikasi	Bahasa Indonesia

Pada tabel 3.5 dan 3.6 merupakan tabel dokumen Persyaratan Fungsional dan Non-Fungsional pada sistem

Membuat Use Case Diagram

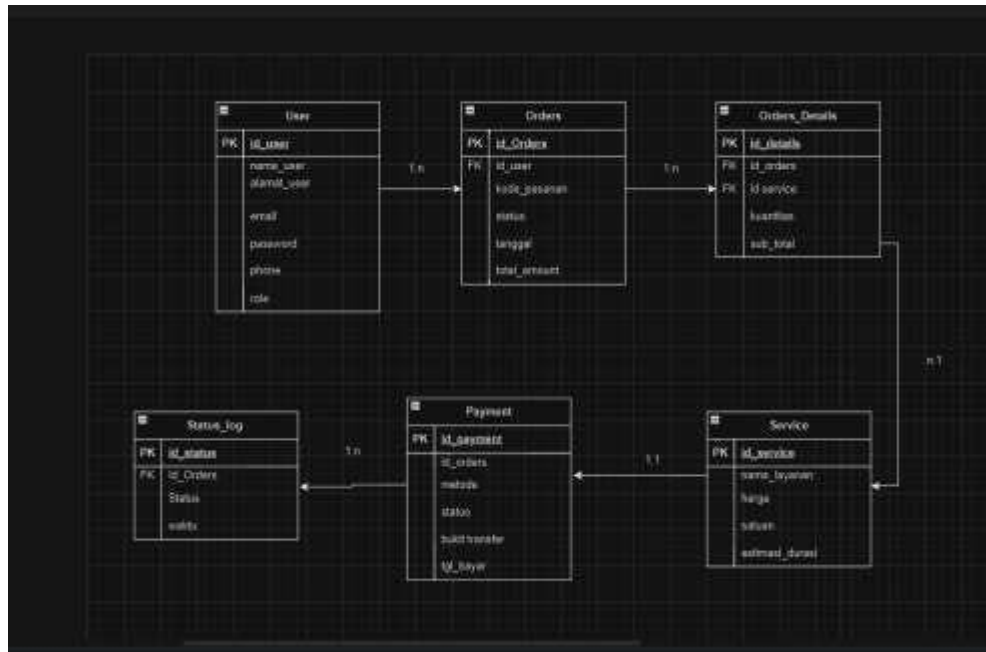
Gambar 3. 1 Use Case Diagram



Gambar 3.1 merupakan use case diagram dari website Sistem Informasi Perpustakaan

Membuat Entity Relationship Diagram (ERD)

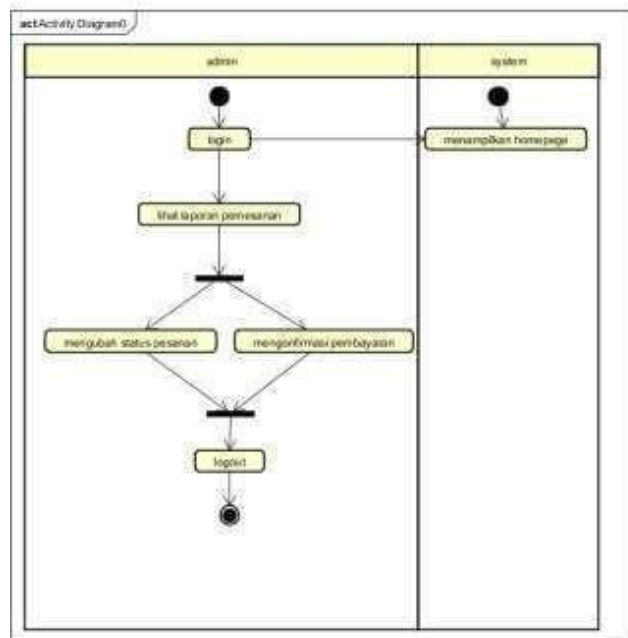
Gambar 3. 2 ERD

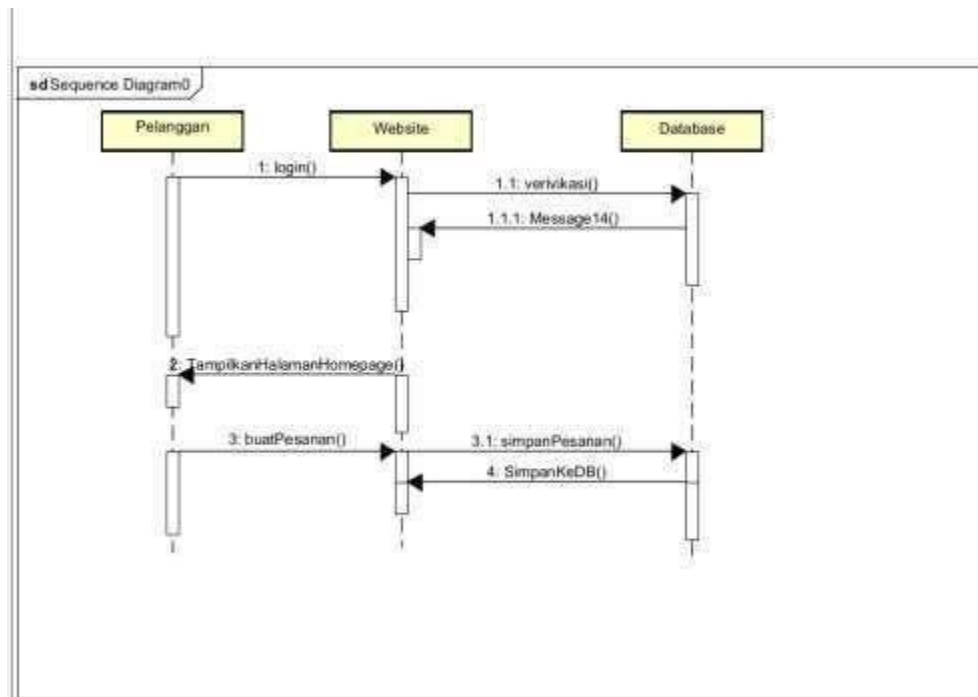
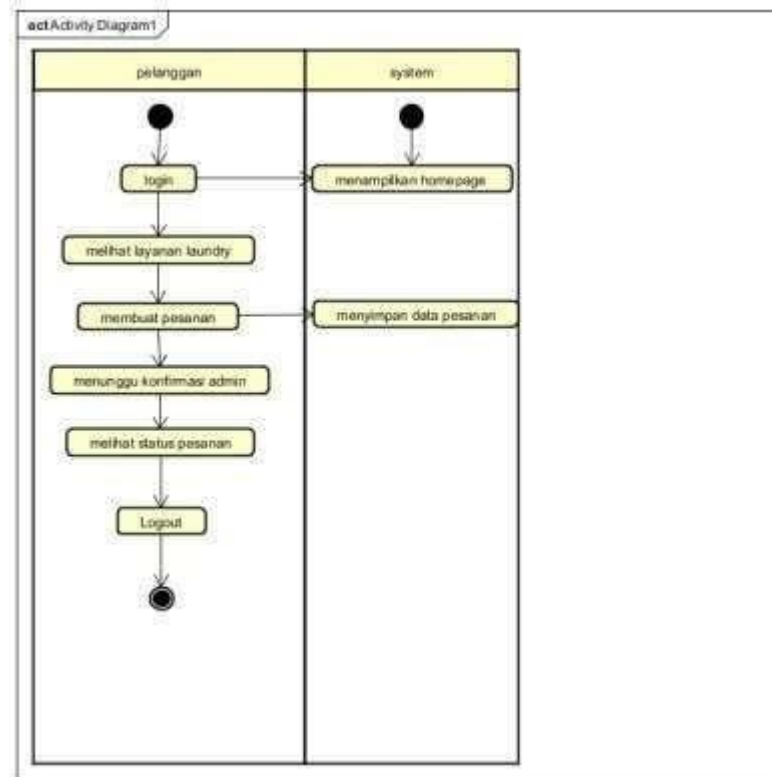


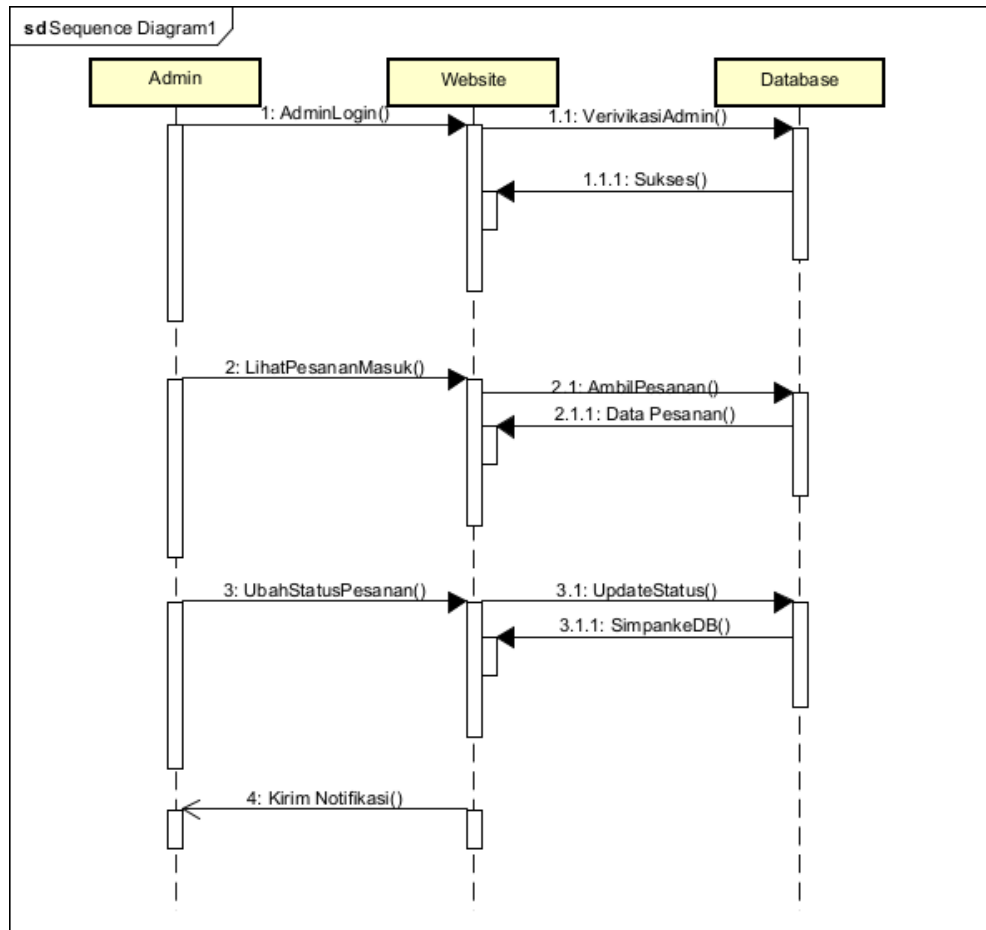
Gambar 3.2 merupakan hasil dari pembuatan Entity Relationship Diagram (ERD)

Activity Diagram

Gambar 3. 3 Activity Diagram







Gambar 3.3 merupakan hasil dari pembuatan Activity Diagram website Sistem Informasi Laundry Berbasis web

3. UI/UX Designer/Frontend

Nama : Syafiq Azhami Yusman

NIM 2200018375

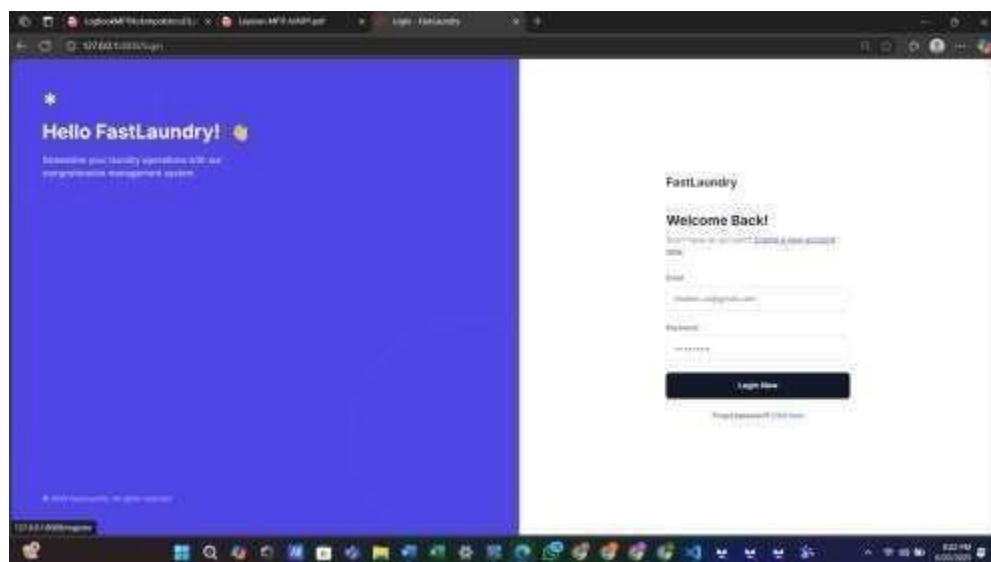
Tabel 3. 7 Realisasi Tugas UI/UX Designer

No	Agenda Kerja	Hari,Tanggal	Waktu	Hasil
1.	Research Plan	15 April 2025	3 jam	Research Plan Document
2.	Interview	19 April 2025	3 jam	Interview Result

3	User Persona dan Empathy Map	21 April 2025	2 hari	Persona and Empathy Map
4	Brainstorming dengan Tim	26 April 2025	1 jam	Discussion
5	Design System	30 April 2025	3 hari	Design System
6	Brand Guideline	4 Mei 2025	1 hari	Brand Guideline
7	Sitemap	5 Mei 2025	1 hari	Sitemap
8	Wireframe	6 Mei 2025	2 hari	Wireframe
9	UI Design	8 Mei 2025	6 hari	UI Design
10	Usability Testing	6 Juni 2025	3 hari	Pengujian SUS

Design Halaman Login

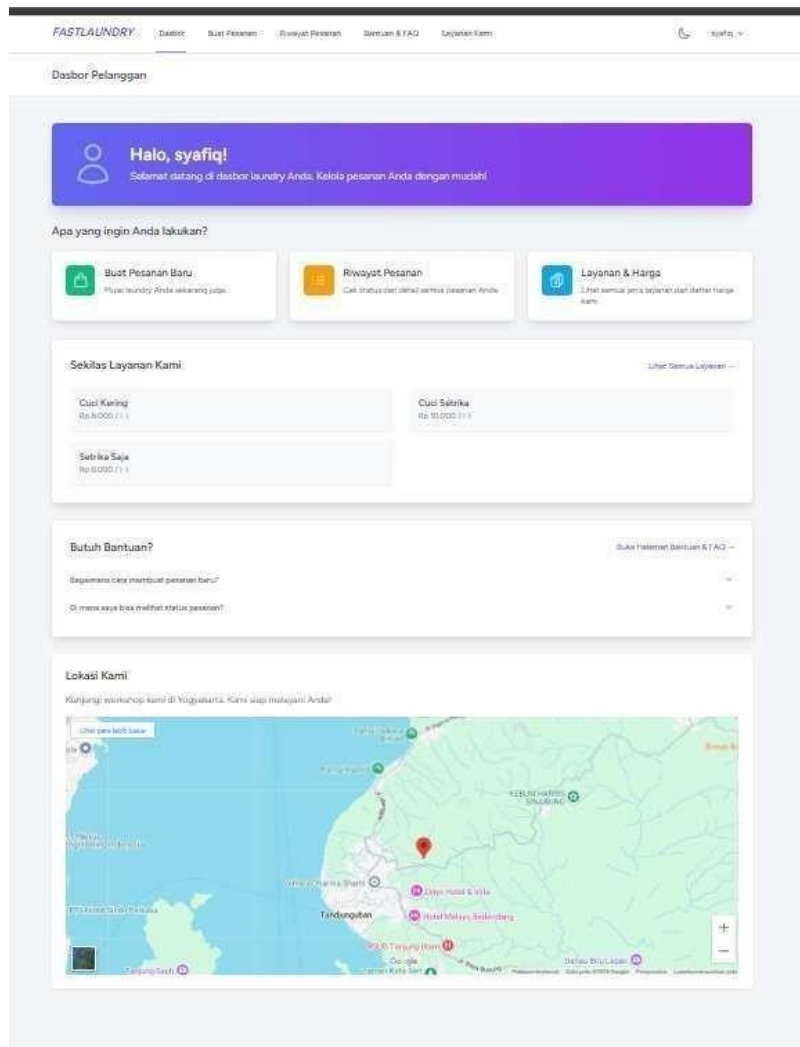
Gambar 3. 4 Design Halaman Login



Gambar 3.4 adalah hasil design halaman login oleh UI/UX Designer

Design Halaman Landing Page

Gambar 3. 5 Design Halaman Landing Page



Gambar 3.5 adalah design halaman Landing Page

Design Halaman Tambah Kategori

Gambar 3. 6 Design Halaman Buat Pesanan

The screenshot shows a web browser window with the URL <http://127.0.0.1:5555/your-website>. The application is titled "FASTLAUNDRY" and has navigation tabs for "Buat Pesanan", "Riwayat Pesanan", "Beranda & TMS", and "Logout User". The main heading is "Buat Pesanan Laundry Baru". The form contains the following fields and elements:

- Form Header:** Includes "No. Laundry" (with a dropdown menu), "Tanggal" (with a date picker), and "Subtotal" (with a text input field).
- Form Body:**
 - A "Pilih Layanan" (Select Service) dropdown menu.
 - A "Catatan Tambahan (Opsional)" (Additional Notes (Optional)) text area.
 - A "Alamat Tujuan" (Destination Address) text area.
 - A "Catat Pengiriman" (Note Delivery) section with a radio button for "Diantar/diambil oleh pelanggan (gratis)" (Delivered/picked up by customer (free)) and a radio button for "Dikirim/diambil sendiri (pembayaran)" (Self-delivered/picked up (payment)).
- Form Footer:** Displays "Total Estimasi Harga: Rp 0" and a "Buat Pesanan" (Create Order) button.

Gambar 3.6 merupakan hasil design dari halaman buat pesanan

Design Halaman Riwayat Pesanan

Gambar 3. 7 Design Riwayat Pesanan

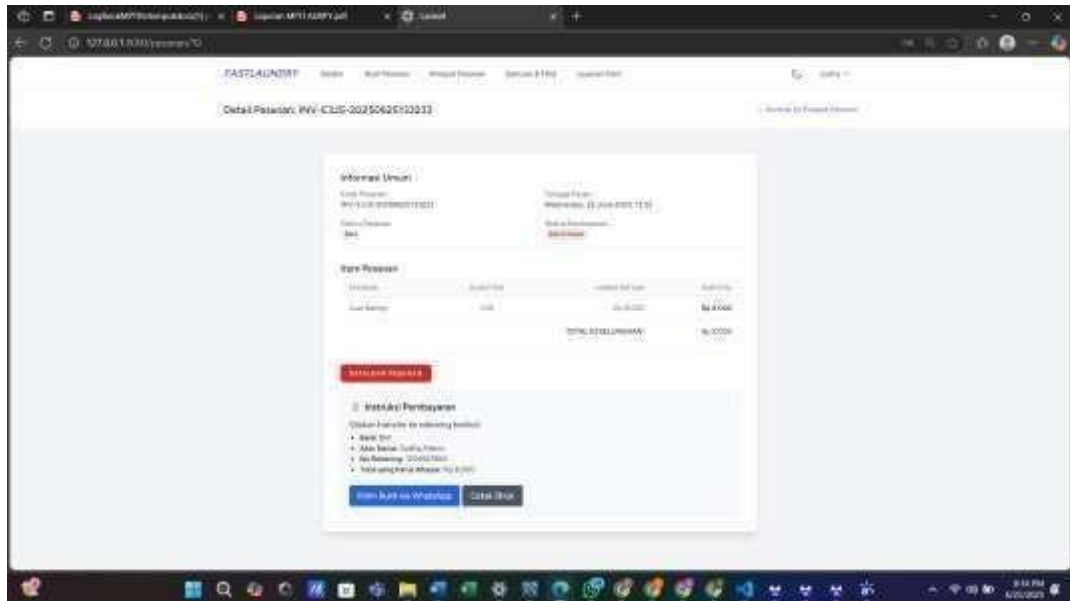
The screenshot shows the "Riwayat Pesanan Saya" (My Order History) page. It features a table with the following columns: "No. Pesanan", "Tanggal", "Status Pesanan", "Alamat Tujuan", "Catat Pengiriman", and "Aksi". The table contains one row of data with the following values:

No. Pesanan	Tanggal	Status Pesanan	Alamat Tujuan	Catat Pengiriman	Aksi
FASTLAUNDRY001	12/01/2023	Belum Bayar			Detail Pesanan

Gambar 3.7 merupakan hasil design dari halaman Riwayat Pesanan

Design Halaman Detail Pesanan

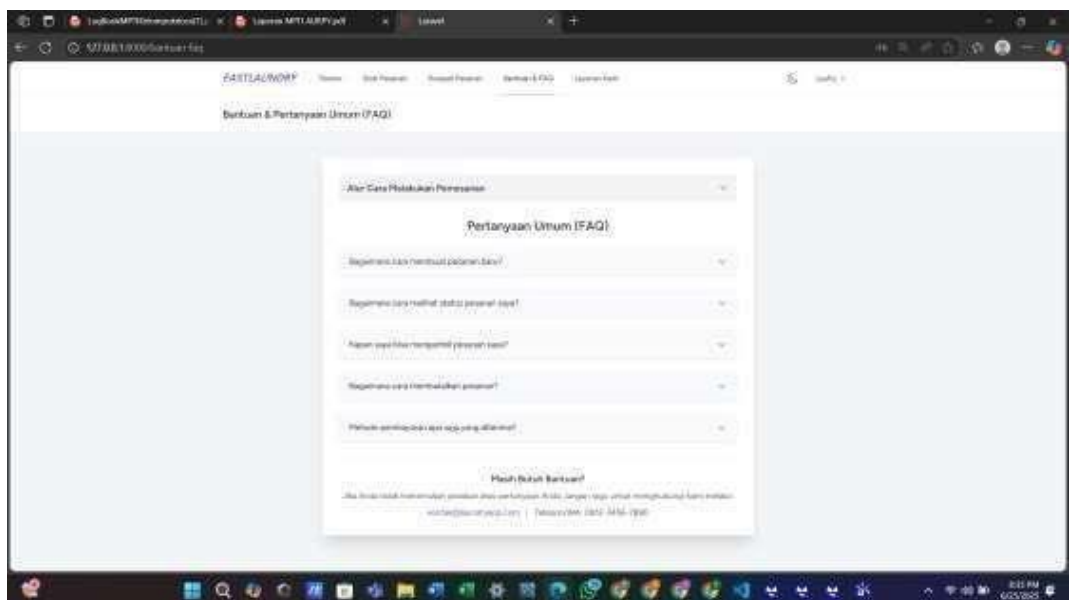
Gambar 3. 8 Design Halaman Detail Pesanan



Gambar 3.8 merupakan hasil design dari halaman detail pesanan

Design Halaman Bantuan FAQ

Gambar 3. 9 Design Halaman Bantuan FAQ



Gambar 3.9 merupakan hasil design dari halaman Bantuan FAQ

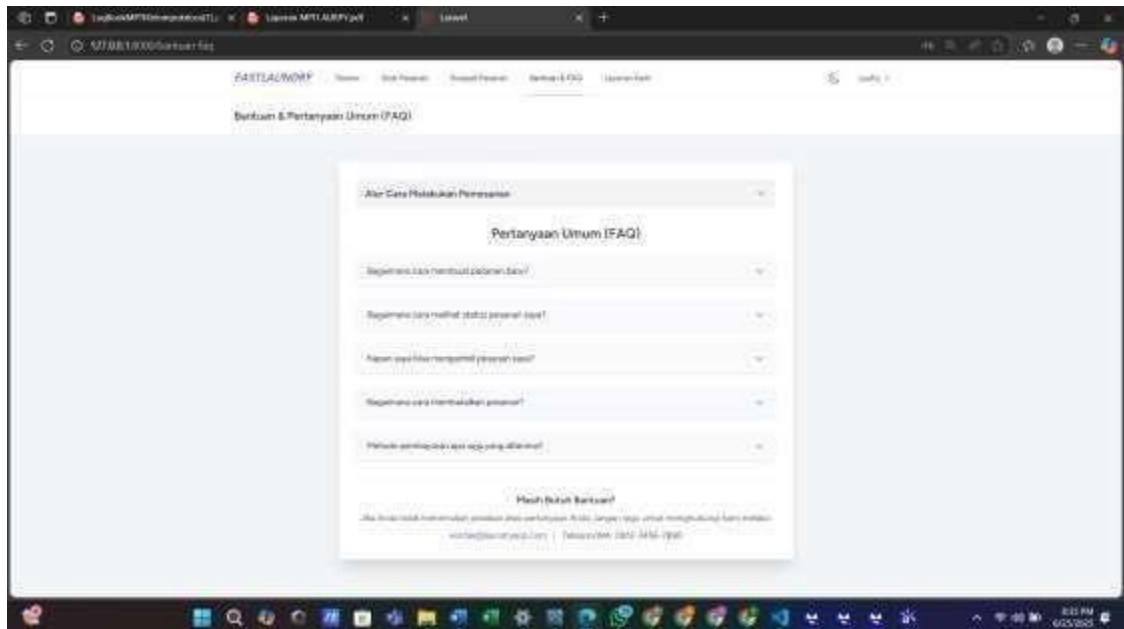
Hasil Usabilty Testing

Tabel 3. 8 Realisasi Tugas Frontend Developer

N o	Agenda Kerja	Hari,Tanggal	Waktu	Hasil
1.	Merancang sistem sesuai dengan alur desain yang telah ditentukan, dengan melakukan analisis sistem terlebih dahulu.	22 Mei 2025	120 Menit	Menetapkan tools dan bahasa pemrograman yang akan digunakan
2.	Membuat Halaman Login	23 Mei 2025	60 Menit	Implementasi halaman login
3.	Implementasi Halaman Admin	24 Mei 2025	120 Menit	Berhasil membuat halaman admin
4.	Implementasi Halaman Tambah Buku	25 Mei 2025	60 Menit	Membuat fitur tambah buku
5.	Implementasi Halaman Koleksi	25 Mei 2025	60 Menit	Membuat halaman koleksi
6.	Implementasi Halaman Forum	26 Mei 2025	3 Hari	Halaman forum
7.	Implementasi Fitur Cetak Laporan	12 Juni 2025	60 Menit	Membuat fitur cetak laporan

Hasil Implementasi Halaman Login

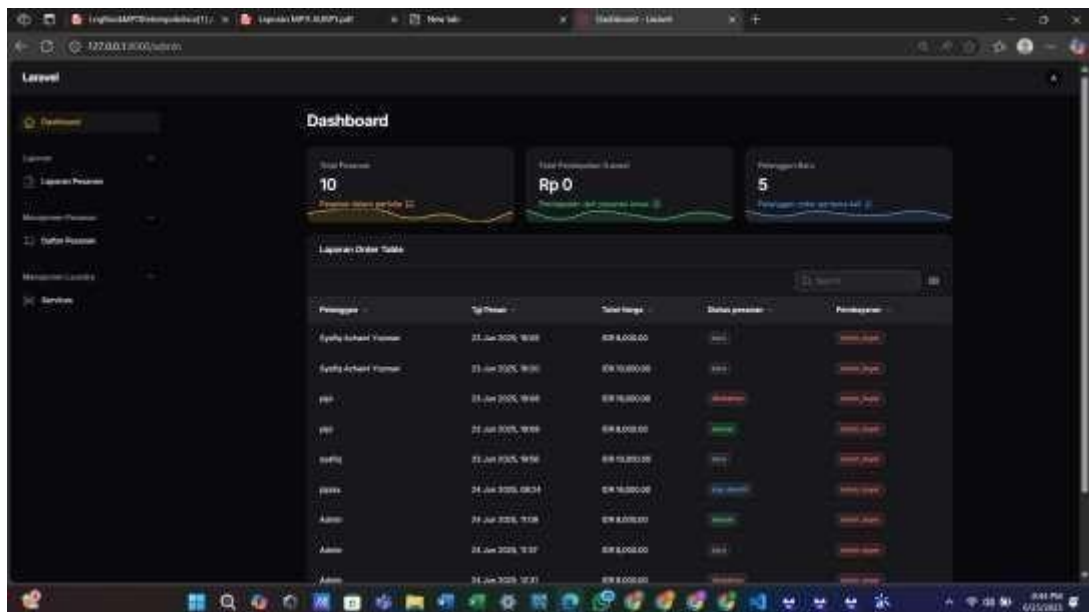
Gambar 3. 10 Hasil Implementasi Halaman Login



Gambar 3.10 merupakan hasil implementasi halaman login

Hasil Implementasi Halaman Admin

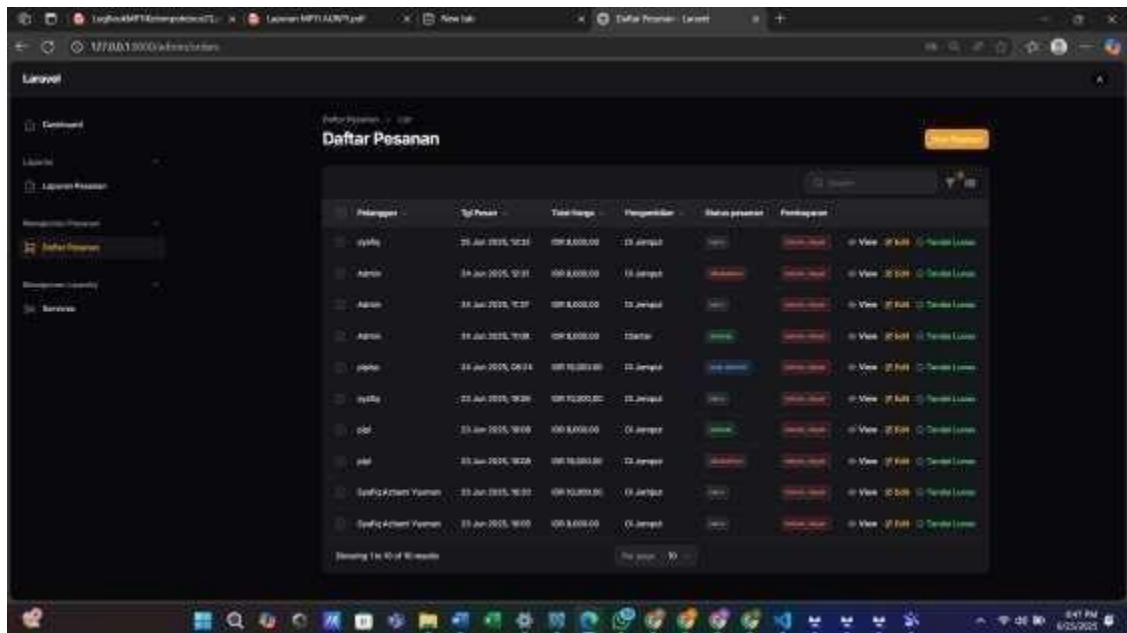
Gambar 3. 11 Hasil Implementasi Halaman Admin



Gambar 3.11 merupakan hasil implementasi halaman admin

Hasil Implementasi Halaman Tambah Kategori

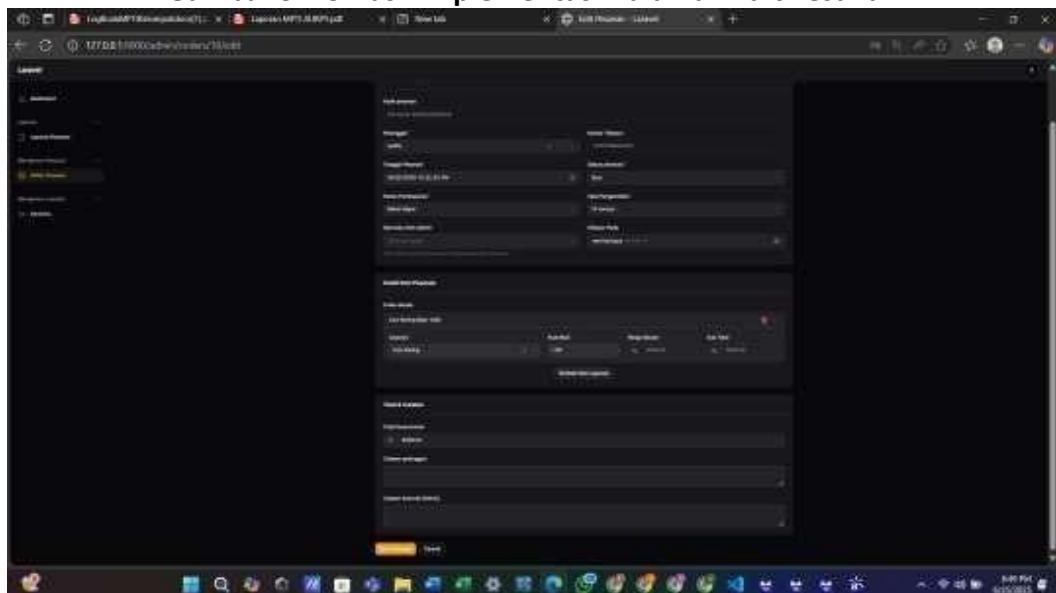
Gambar 3. 12 Hasil Implementasi Halaman Daftar Pesanan



Gambar 3.12 merupakan hasil implementasi halaman Daftar Pesanan

Hasil Implementasi Halaman Edit Pesanan

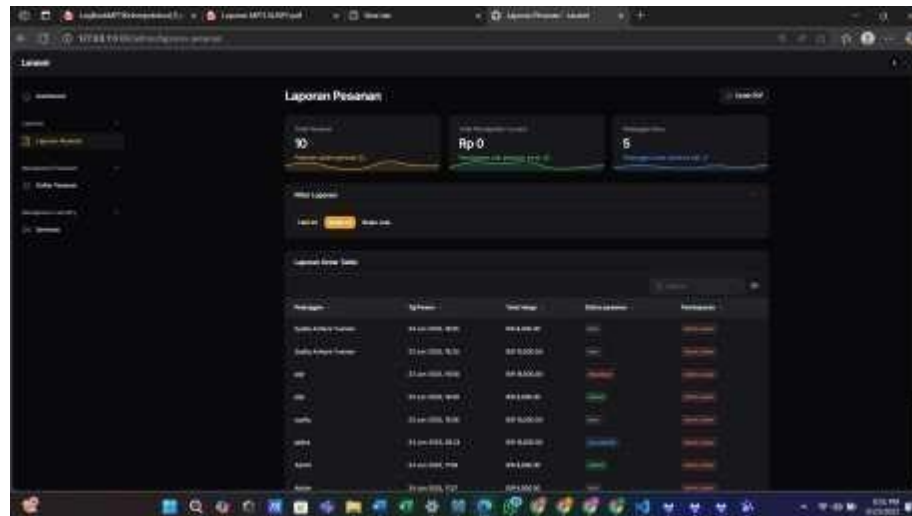
Gambar 3. 13 Hasil Implementasi Halaman Edit Pesanan



Gambar 3.13 merupakan hasil implementasi dari halaman Edit Pesanan

Hasil Implementasi Halaman Laporan Pesanan

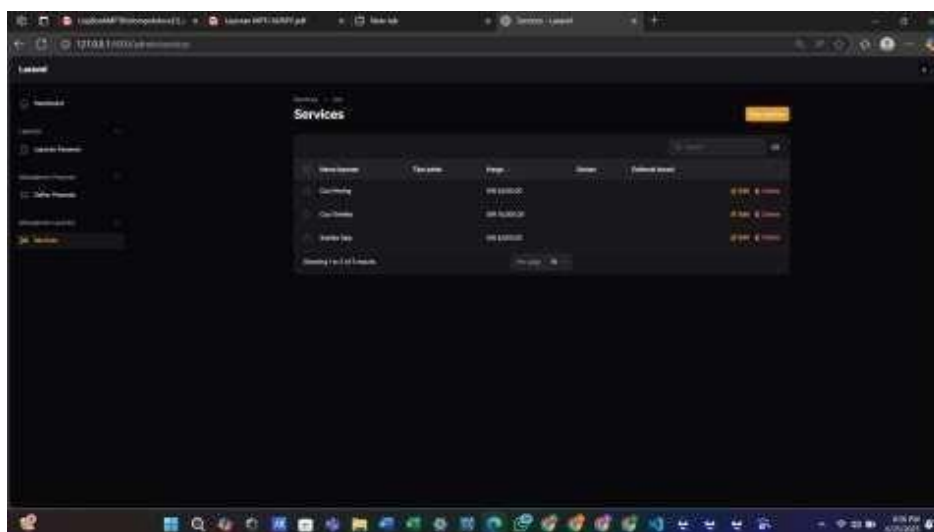
Gambar 3. 14 Hasil Implementasi Halaman Laporan Pesanan



Gambar 3.14 merupakan hasil implementasi halaman koleksi

Hasil Implementasi Halaman Service

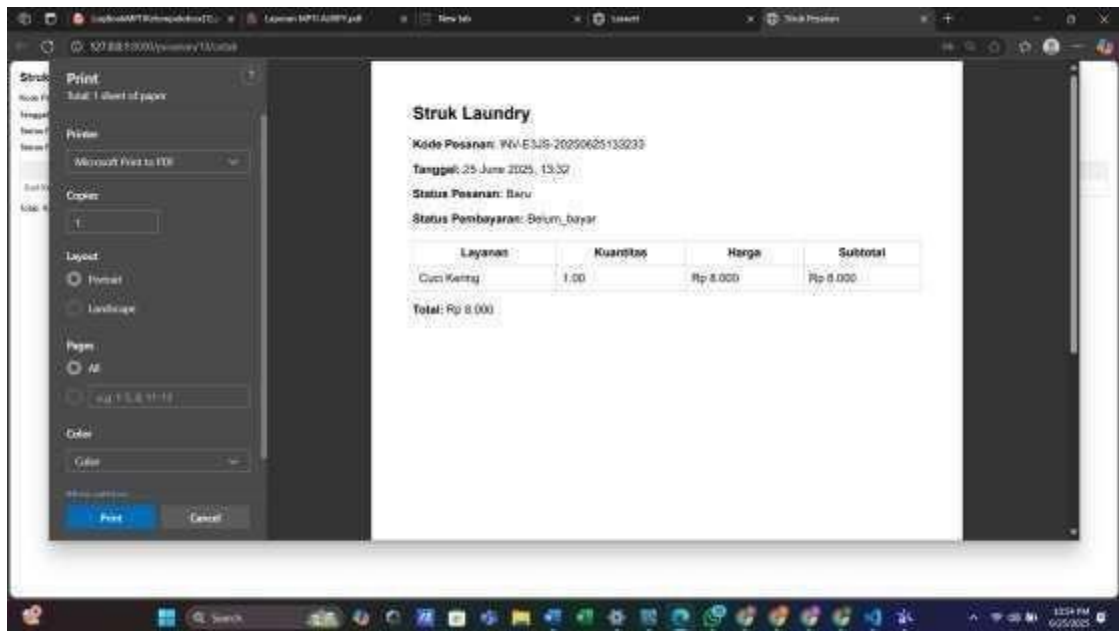
Gambar 3. 15 Hasil Implementasi Halaman Service



Gambar 3.15 merupakan hasil implementasi halaman Service

Hasil Implementasi Fitur Cetak Struk

Gambar 3. 16 Hasil Implementasi Fitur Cetak Laporan



Gambar 3.16 merupakan hasil implementasi dari fitur cetak sturct

Link Figma: <https://www.figma.com/design/gX4MePtbJhUYox0zFIh94O/Sketsa-System?node-id=0-1&p=f&t=XTC9qR31jIVJOXFV-0>

4. Backend Developer

Nama : Syafiq Azhami Yusman

NIM 2200018162

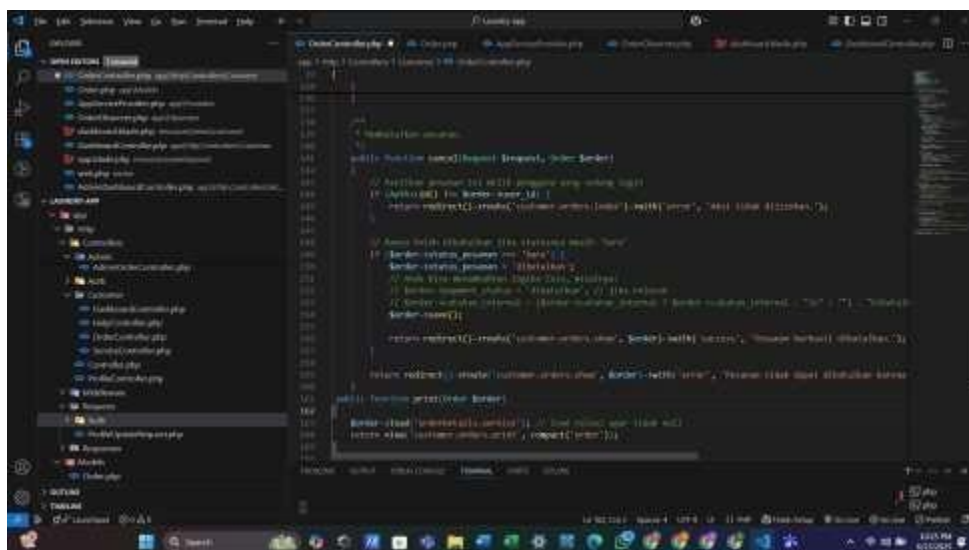
Tabel 3. 9 Realisasi Tugas Backend Developer

No	Agenda Kerja	Hari,Tanggal	Waktu	Hasil
----	--------------	--------------	-------	-------

1.	Terlibat dalam langkah analisis kebutuhan untuk mengidentifikasi informasi yang harus disimpan dalam basis data.	22 Mei 2025	120 Menit	Menetapkan tools yang akan digunakan
2.	Implementasi CRUD pada code.	24 Mei 2025	5 Hari	Membuat fitur CRUD
3.	Mengembangkan struktur data yang lebih rinci dan spesifik	6 Juni 2025	7 hari	Membuat struktur data backend dan logika program pada fitur

Membuat struktur CRUD

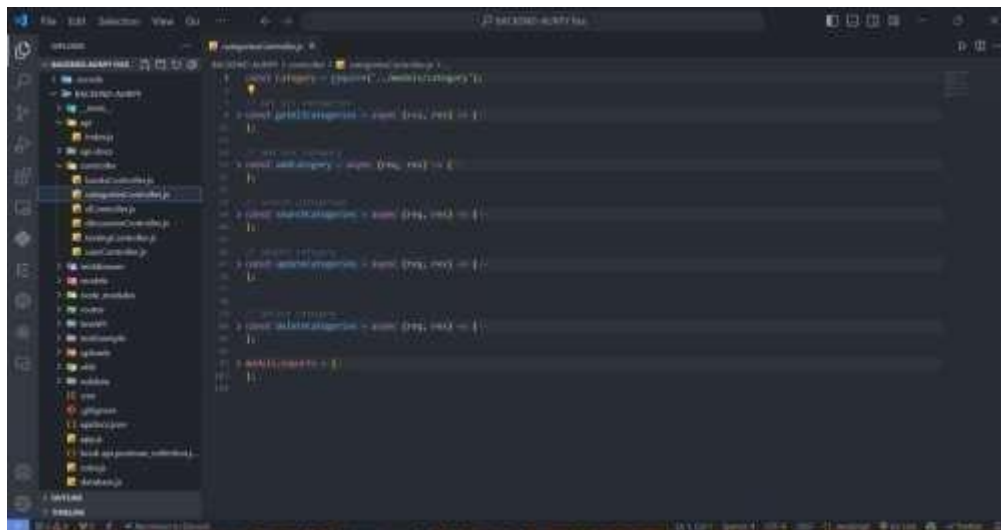
Gambar 3. 17 Struktur CRUD



Gambar 3.17 merupakan fungsi yang mendefinisikan struktur CRUD

Mengelola data kategori sistem

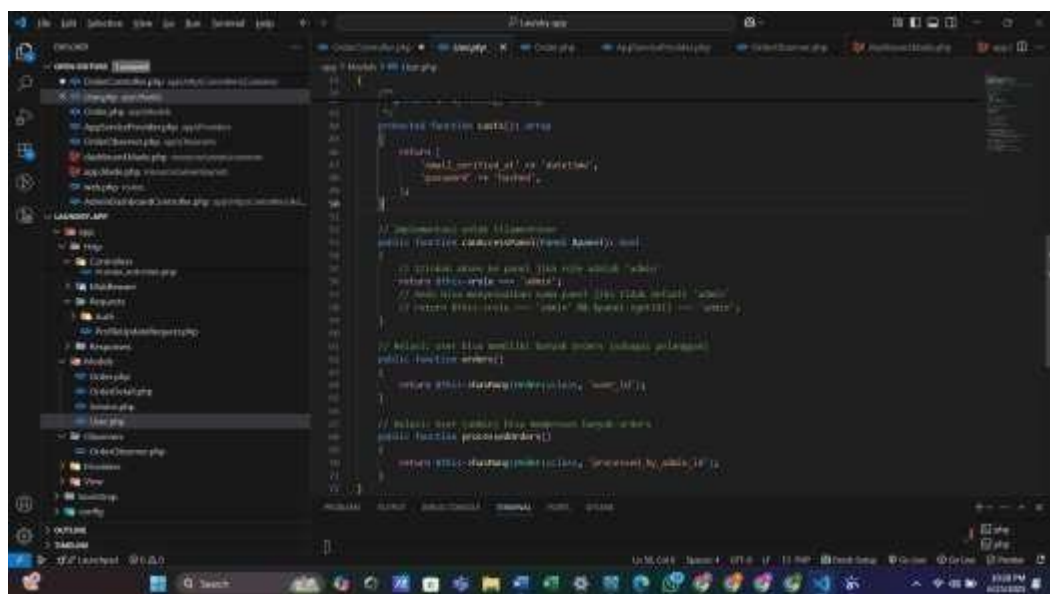
Gambar 3. 18 Fungsi Kategori



Gambar 3.18 merupakan fungsi yang digunakan untuk mengelola kategori Sistem Informasi perpustakaan

Mengelola data User

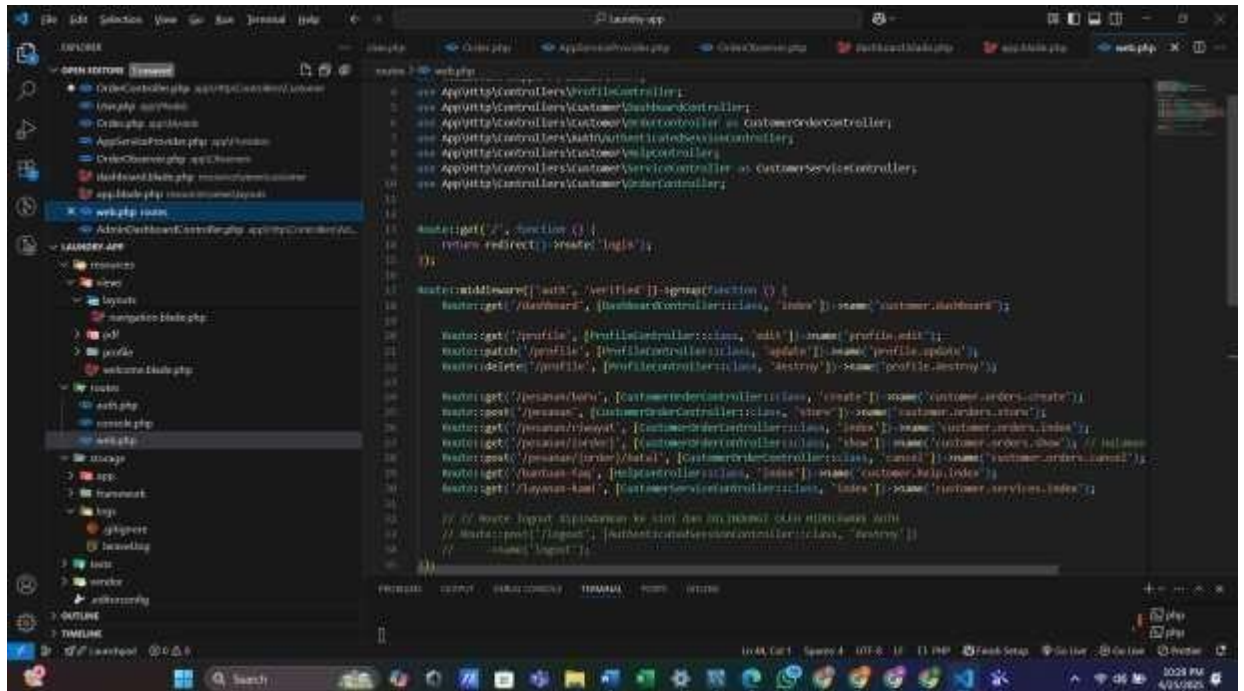
Gambar 3. 19 Kelola Data User



Gambar 3.19 merupakan fungsi yang digunakan untuk CRUD dan mengelola data user

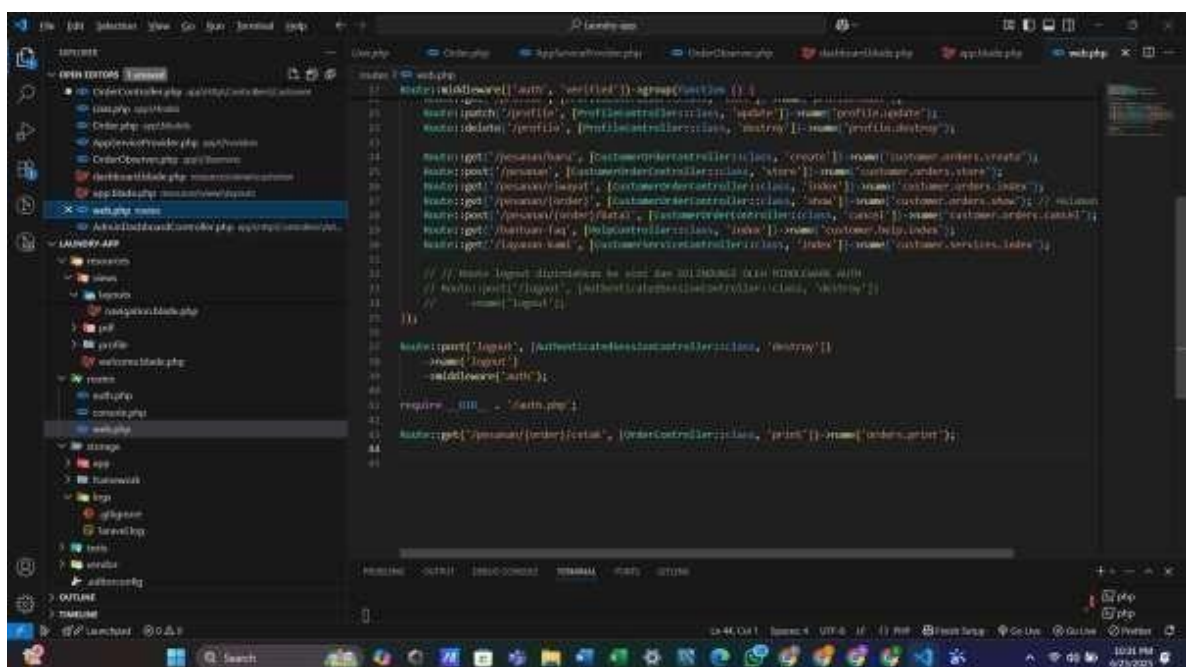
Routing Laundry

Gambar 3. 20 Routing Laundry



Gambar 3.20 merupakan implementasi dari routing yang digunakan untuk menangani berbagai request yang terkait dengan Laundry dan kategori dalam sistem.

Gambar 3. 21 Rute Pengguna



Gambar 3.21 merupakan fungsi yang mengatur operasi terkait autentikasi pengguna

C. Penjaminan Kualitas Proyek

Dalam menjamin kualitas proyek “Sistem Informasi Laundry Berbasis Web Mamak Laundry” agar memiliki performa yang optimal dan layak digunakan oleh pengguna maupun admin, tim pengembang melakukan beberapa pengujian sistem sebagai berikut:

1) Pengujian Fungsi Pemesanan dan Status Pesanan

Pengujian dilakukan pada alur utama aplikasi, yaitu proses pemesanan layanan laundry oleh pelanggan, pengecekan status pesanan, hingga notifikasi perubahan status yang dikirim melalui WhatsApp. Tujuannya untuk memastikan bahwa seluruh proses berjalan sesuai fungsinya, tanpa error, dan data tersimpan dengan benar.

2) Pengujian Antarmuka Web (Frontend)

Pengujian dilakukan pada tampilan antarmuka pengguna (UI/UX), mencakup halaman login, beranda, halaman layanan, formulir pemesanan, riwayat pesanan, dan halaman bantuan. Pengujian ini memastikan bahwa seluruh elemen tampilan dapat diakses dan digunakan dengan baik di berbagai perangkat (responsif).

3) Pengujian Otentikasi dan Hak Akses

Pengujian dilakukan terhadap proses login untuk memastikan keamanan akun pengguna dan admin. Selain itu, diuji juga apakah setiap pengguna hanya bisa mengakses fitur yang sesuai dengan perannya (admin atau pelanggan).

4) Pengujian Integrasi WhatsApp API (Fonnte)

Fitur notifikasi otomatis melalui WhatsApp diuji untuk memastikan bahwa status pesanan yang diubah oleh admin benar-benar terkirim ke pelanggan. Hal ini penting agar pelanggan selalu mendapatkan update secara real-time.

5) Pengujian Kinerja dan Keamanan Dasar

Dilakukan simulasi beberapa pengguna untuk mengakses website secara bersamaan guna memastikan website tetap stabil. Selain itu, dilakukan validasi

input agar sistem tidak mudah disusupi oleh data tidak valid atau potensi serangan sederhana (misalnya: SQL Injection atau XSS).

D. Keberlanjutan Proyek

Proyek ini telah diserahkan kepada Ibu Dyah Irma Gendraningtyas selaku pemilik Mamak Laundry, sebagai bukti penyelesaian pengembangan Website Sistem Informasi Laundry Berbasis Web. Setelah penyerahan, tim pengembang akan tetap menjalin komunikasi aktif dan menyediakan dukungan teknis agar sistem dapat terus digunakan secara optimal dan tepat sasaran dalam kegiatan operasional laundry.

- 6) Kolaborasi Tim : Tim pengembang akan terus menjalin komunikasi dan kolaborasi internal untuk menangani perbaikan, pembaruan, dan pengembangan fitur lanjutan jika dibutuhkan oleh pihak Mamak Laundry.
- 7) Stakeholder Engagement : Pemilik laundry dan pihak terkait akan tetap dilibatkan dalam proses pemantauan dan evaluasi sistem. Hal ini bertujuan untuk memastikan bahwa fitur dan layanan yang tersedia pada website tetap relevan dengan kebutuhan operasional harian laundry serta kenyamanan pelanggan.
- 8) Pemeliharaan Sistem (Maintenance): Tim pengembang akan memberikan panduan teknis dan bersedia melakukan pemeliharaan berkala, termasuk perbaikan bug dan peningkatan performa sistem jika dibutuhkan, agar sistem tetap stabil dan aman

BAB IV PENUTUP

A. Kesimpulan

Kesimpulan dari pelaksanaan Manajemen Proyek Teknologi Informasi dengan judul “Sistem Informasi Laundry Berbasis Web – Mamak Laundry” adalah sebagai berikut:

1. Proyek pembuatan aplikasi Website Sistem Informasi Laundry Berbasis Web untuk Mamak Laundry telah berhasil direalisasikan dengan baik. Sistem ini memberikan kemudahan bagi pelanggan dalam melakukan pemesanan, memantau status cucian, serta mengakses informasi layanan secara cepat dan efisien.
2. Keberhasilan proyek ini sangat dipengaruhi oleh perencanaan yang matang, kerja sama tim yang solid, dan komunikasi aktif antara tim pengembang dan pemilik usaha sebagai mitra. Setiap anggota tim menjalankan perannya dengan tanggung jawab dan profesionalisme.
3. Tahapan proyek mencakup analisis kebutuhan mitra, perancangan sistem, implementasi fitur layanan laundry, hingga pengujian fungsionalitas. Sistem ini diharapkan menjadi solusi jangka panjang yang mendukung operasional Mamak Laundry dan meningkatkan kepuasan pelanggan.

B. Saran

Sistem website yang telah diterapkan dalam pengelolaan tugas proyek ini masih memiliki ruang yang signifikan untuk pengembangan lebih lanjut. Harapannya adalah untuk terus meningkatkan fungsionalitasnya agar bisa memberikan layanan yang lebih baik dan juga menghadirkan beragam jasa yang lebih beragam kedepannya.

Sistem informasi laundry berbasis website yang telah diterapkan dalam proyek ini masih memiliki potensi pengembangan yang luas untuk ke depannya. Harapannya, sistem ini dapat terus ditingkatkan agar mampu memberikan pelayanan yang lebih optimal, baik bagi pelanggan maupun pihak pengelola

laundry.

Untuk itu, beberapa hal yang dapat dijadikan bahan pertimbangan pengembangan lebih lanjut antara lain:

1. Penambahan Fitur Tambahan

Beberapa fitur yang dapat ditambahkan di masa mendatang mencakup:

- a. Sistem notifikasi otomatis berbasis WhatsApp yang lebih kompleks (misalnya notifikasi pembayaran berhasil, penjemputan, pengeringan, hingga pengantaran kembali).
- b. Fitur pelacakan pesanan secara real-time.
- c. Integrasi sistem pembayaran otomatis melalui e-wallet atau QRIS.

2. Pengembangan Layanan Pelanggan

Menyediakan fitur ulasan pelanggan dan sistem rating, sehingga dapat meningkatkan kepercayaan pelanggan terhadap layanan Mamak Laundry.

3. Optimasi Kinerja Sistem

Pengoptimalan dari sisi performa aplikasi juga penting dilakukan secara berkala, baik dalam bentuk:

- a. Pemeliharaan perangkat lunak (software) seperti update framework dan dependency.
- b. Pemeriksaan data dan log sistem untuk memastikan tidak ada error pada proses transaksi.
- c. Pemantauan keamanan sistem, guna menjaga data pelanggan tetap aman dan tidak mudah disalahgunakan.

4. Dukungan Mobile Friendly

Sistem juga dapat disesuaikan lebih lanjut untuk akses optimal melalui perangkat mobile, mengingat mayoritas pelanggan akan mengakses layanan melalui smartphone.

Dengan adanya pengembangan dan pemeliharaan yang berkelanjutan, diharapkan sistem informasi laundry ini dapat memberikan pengalaman pengguna yang lebih baik, serta membantu meningkatkan efisiensi operasional di Mamak Laundry.

LAMPIRAN

1. Proposal Proyek

PROYEK SISTEM INFORMASI LAUNDRY BERBASIS WEBSITE



TIM PELAKSANA:

Farel Cahyaning Bagaskara (2200018188) (D)
Syafiq Azhami Yusman 2100018118) (C)
M Reinza Chaniago (2200018390) (B)

**PROGRAM STUDI INFORMATIKA FAKULTAS
TEKNOLOGI INDUSTRI UNIVERSITAS AHMAD
DAHLAN 2025**

BAB I

Pendahuluan

A. Latar Belakang

Dalam era digital saat ini, perkembangan teknologi informasi telah memberikan dampak signifikan terhadap berbagai bidang kehidupan, termasuk pada sektor jasa layanan rumah tangga seperti laundry. Mamak Laundry, yang berlokasi di Tanjung Uban Utara, Kabupaten Bintan, Kepulauan Riau, merupakan salah satu usaha jasa laundry yang terus berkembang dan memiliki komitmen untuk memberikan pelayanan terbaik kepada pelanggannya.

Sebelumnya, proses pemesanan jasa laundry masih dilakukan secara manual, baik melalui pesan WhatsApp maupun kunjungan langsung ke lokasi. Hal ini seringkali menimbulkan berbagai kendala, seperti keterlambatan informasi, kesulitan dalam pelacakan status pesanan, serta ketidakefisienan dalam pengelolaan data pelanggan dan transaksi.

Melihat permasalahan tersebut, maka dibutuhkan sebuah sistem yang mampu mendukung operasional Mamak Laundry secara lebih efisien dan modern. Oleh karena itu, dibentuklah proyek pengembangan Sistem Informasi Laundry Berbasis Website. Sistem ini dirancang untuk membantu pelanggan dalam melakukan pemesanan, melihat status cucian, serta menerima notifikasi secara otomatis melalui WhatsApp. Selain itu, sistem ini juga akan mempermudah admin dalam mengelola data pelanggan, layanan, transaksi, dan pelaporan.

Dengan mengimplementasikan teknologi informasi dalam operasional bisnis, diharapkan Mamak Laundry dapat meningkatkan kualitas layanan, memperluas jangkauan pelanggan, serta memperkuat daya saing usaha di tengah persaingan industri laundry yang semakin kompetitif.

Tujuan utama dari proyek pengembangan Sistem Informasi Laundry Berbasis Website ini adalah untuk meningkatkan efisiensi operasional, kemudahan akses bagi pelanggan, dan kesinambungan pengelolaan data layanan laundry secara digital. Dengan memanfaatkan teknologi informasi, diharapkan sistem ini dapat mendukung Mamak Laundry menjadi lebih modern, responsif terhadap kebutuhan pelanggan, serta mampu memberikan layanan yang lebih profesional dan terorganisir.

B. Tujuan Proyek

1. Meningkatkan Aksesibilitas Layanan
Memberikan kemudahan bagi pelanggan untuk memesan layanan laundry kapan pun dan di mana pun tanpa harus datang langsung ke lokasi, cukup melalui website yang responsif dan mudah digunakan.
2. Efisiensi dalam Pengelolaan Pesanan dan Data Pelanggan

Menyediakan sistem backend untuk admin agar dapat mengelola data pelanggan, data pesanan, dan status laundry secara lebih cepat, rapi, dan otomatis, tanpa perlu mencatat secara manual.

3. Peningkatan Pengalaman Pengguna

Menyediakan antarmuka website yang ramah pengguna serta dilengkapi dengan fitur notifikasi otomatis (melalui WhatsApp) yang memberikan informasi real-time kepada pelanggan mengenai status pesanan mereka.

4. Digitalisasi dan Otomatisasi Proses

Meningkatkan efisiensi proses bisnis melalui otomatisasi pengelolaan data layanan, status proses laundry (dijemput, dicuci, disetrika, dikirim), dan laporan transaksi secara digital.

BAB II

Pelaksanaan Proyek

A. Jobdesk

Nama: Farel Cahayning Bagaskara

Peran: Project Manager & Marketing

Tugas:

1. Membentuk dan mengoordinasikan tim pengembang proyek website laundry.
2. Menyusun dan membagi jobdesk sesuai kemampuan anggota tim.
3. Memimpin rapat-rapat internal serta bertanggung jawab terhadap hubungan dengan klien.
4. Menyusun dan menandatangani perjanjian kerja sama (MoU) dengan pihak Mamak Laundry.
5. Mengelola jalannya proyek agar sesuai jadwal dan target.
6. Menyusun skenario pengujian dan melakukan verifikasi hasil pengembangan.
7. Mengawasi proses implementasi sistem dan mengkomunikasikan setiap progres dengan pihak klien.
8. Bertugas melakukan promosi dan presentasi hasil proyek kepada klien.

Nama: Syafiq Azhami Yusman

Peran: Sekretaris & Backend Developer

Tugas:

1. Menyusun dokumen administrasi seperti proposal, notulensi rapat, dan laporan akhir proyek.

2. Menganalisis kebutuhan fungsional dan non-fungsional dari sistem berdasarkan masukan klien.
3. Merancang dan mengimplementasikan arsitektur backend sistem (logika, database, dan integrasi).
4. Membuat dan mengelola koneksi API, serta menjaga performa dan keamanan server.
5. Mencatat kebutuhan perubahan dan pembaruan sistem berdasarkan hasil pengujian.
6. Bekerja sama dengan frontend developer untuk memastikan integrasi frontend–backend berjalan lancar.

Nama: M. Reinza Chaniago

Peran: System Analyst & Bendahara

Tugas:

1. Menganalisis proses bisnis dari Mamak Laundry dan mentranslasikan ke dalam alur sistem.
2. Membuat dokumen spesifikasi teknis seperti use case diagram, ERD, dan flow sistem.
3. Melakukan survei kebutuhan sistem langsung kepada klien dan pengguna akhir.
4. Menyusun estimasi biaya proyek, membuat laporan keuangan, dan mengelola dana operasional proyek.
5. Bertanggung jawab atas pencatatan dan distribusi dana untuk keperluan tim.

Nama: Syafiq Azhami Yusman

Peran: UI/UX Designer & Frontend Developer

Tugas:

1. Melakukan riset dan observasi terhadap preferensi pengguna website laundry.
2. Mendesain antarmuka website laundry yang responsif, menarik, dan mudah digunakan.
3. Membuat wireframe dan prototipe UI dari seluruh halaman seperti beranda, form pesanan, status, dan layanan.
4. Mengimplementasikan desain frontend menggunakan HTML, CSS, JavaScript, dan framework Laravel Blade.
5. Mengintegrasikan antarmuka dengan backend melalui API, serta memastikan tampilannya responsif di berbagai perangkat.
6. Mengoptimalkan tampilan UI agar memberikan pengalaman pengguna

2. Kontrak Kerja dengan Mitra

SURAT PERJANJIAN KERJASAMA PENGEMBANGAN APLIKASI WEBSITE LAUNDRY UNTUK MAMAK LAUNDRY

Pada hari ini, Selasa tanggal 26 Bulan Maret tahun 2025 bertempat di Kota Yogyakarta, telah ditandatangani perjanjian kerjasama antara:

Nama : Farel Cahyaning Bagaskara

Jabatan : Project Manager

Bertempat di : Yogyakarta

Dalam hal ini bertindak sebagai pembuat Sistem Informasi Laundry untuk PP Mamak Laundry yang selanjutnya disebut sebagai PIHAK PERTAMA

Nama : DyahIrmaGendraningtyas

Jabatan : Owner Mamak Laundry

Bertempat di : Kep.Riau

Dalam hal ini bertindak sebagai klien pembuatan Website Sistem Informasi Laundry MAMAK Laundry yang selanjutnya disebut PIHAK KEDUA

KEDUA BELAH PIHAK sepakat untuk mengikat diri dalam perjanjian kontrak dengan ketentuan - ketentuan sebagai berikut:

Pasal 1 Jangka Waktu Kerja

1. PIHAK PERTAMA bersedia menerima dan mengerjakan pekerjaan yang diberikan oleh PIHAK KEDUA selama 11 minggu untuk periode dari 25 Maret 2025 sampai dengan 10 Juni 2025.
2. PIHAK KEDUA selaku pemberi pekerjaan bersedia memberikan data dan informasi yang dibutuhkan guna pengembangan Website selama 11 minggu periode dari tanggal 25 Maret 2025 sampai dengan 10 Juni 2025.

Pasal 2 Deskripsi Pekerjaan

1. Nama Website adalah "Sistem Informasi Laundry Berbasis Wwebsite " yang selanjutnya disebut "Website tersebut"
2. PIHAK PERTAMA yang memiliki tim beranggotakan 4 (empat) orang, yaitu 1 (satu) orang Project Manager, 1 (satu) orang sekretaris, 1 (satu) orang Bendahara, 1 (satu) orang Anggota, bertugas membuat Aplikasi tersebut dengan rincian pekerjaan sebagai berikut :
 - a. Menyediakan desain antarmuka website tersebut, yang sesuai dengan kebutuhan dan kesepakatan bersama antara KEDUA BELAH PIHAK.
 - b. PIHAK PERTAMA akan memberikan masukan-masukan terkait proses pengembangan dan penerbitan website kepada PIHAK KEDUA, agar proses tersebut dapat berjalan dengan baik dan sesuai dengan standar yang ditetapkan.
3. PIHAK KEDUA bersedia memberikan data yang dibutuhkan PIHAK PERTAMA di waktu yang telah disepakati bersama, dan

diberikan sebelum pembuatan aplikasi dimulai, yaitu data-data berikut :

- a. Data-data yang dibutuhkan, semisal Logo Website, foto visual, dan lain- lain yang dibutuhkan di waktu pembuatan Website tersebut.
4. PIHAK PERTAMA memiliki fase-fase pembuatan Website sebagai berikut:
- a. Analisis Kebutuhan, yaitu pengumpulan data dan informasi terkait pembuatan web kemudian diubah menjadi poin-poin tugas-tugas yang wajib dilaksanakan oleh PIHAK PERTAMA. Pada tahap ini PIHAK PERTAMA akan menanyakan dan meminta PIHAK KEDUA mengenai segala informasi terkait pembaruan website tersebut , baik itu informasi secara verbal dan tertulis.
 - b. Desain Basis Data, pemodelan sistem (flowchart) dan Desain Antarmuka Website. PIHAK PERTAMA akan menggunakan segala kebutuhan yang telah dirumuskan pada tahap Analisis Kebutuhan (poin a) menjadi rancangan desain basis data dan desain antarmuka Website. PIHAK KEDUA wajib memberikan masukan terkait setiap proses desain yang dikerjakan oleh PIHAK PERTAMA. Maksimal 7 hari setelah PIHAK PERTAMA telah melaporkan kepada PIHAK KEDUA. Hal ini dimaksud agar PIHAK PERTAMA tidak menunggu terlalu lama masukan-masukan yang diberikan oleh PIHAK KEDUA, karena hal tersebut dapat mengganggu jadwal pelaksanaan pembuatan Website tersebut.
 - c. Implementasi setiap desain desain di point b menjadi sebuah kode sumber (source code). PIHAK PERTAMA menulis kode-kode Website sehingga Website tersebut dapat digunakan di desktop. PIHAK PERTAMA wajib menguji coba setiap install website sebelum menyerahkan ke PIHAK KEDUA. dan PIHAK KEDUA wajib memberikan masukan setiap selesai menguji coba website tersebut, untuk kemudian ditindaklanjuti PIHAK PERTAMA sehingga menghasilkan website yang diharapkan dan sesuai dengan kebutuhan yang telah disepakati kedua belah pihak. PIHAK PERTAMA wajib mencatat setiap masukan-masukan PIHAK KEDUA agar nantinya dapat dilaksanakan perbaikan perbaikan website setelah tahap implementasi telah diselesaikan.

- d. Pengujian Website. Menguji setiap komponen website, baik komponen desain, fungsionalitas, validasi input dan output, dan lain lain agar website sesuai dengan apa yang dibutuhkan. PIHAK PERTAMA wajib menguji website yang telah dibuat bersama-sama dengan PIHAK KEDUA. Setiap hasil pengujian berupa laporan masalah (bugs ataupun error) website, wajib dicatat dan di dokumen tertulis sehingga dapat dipantau bersama antara kedua belah pihak Deployment Website. Website yang telah teruji dengan baik akan diterbitkan sesuai keinginan klien. PIHAK KEDUA berkewajiban memberikan informasi yang dibutuhkan untuk keperluan pembuatan Website, seperti: Nama Website, Deskripsi Website, Kategori Website, dokumen Kebijakan privasi, dan lain hal. Apabila PIHAK KEDUA tidak berkenan memberikan informasi-informasi di atas kepada PIHAK PERTAMA, maka penerbitan Website dilakukan oleh PIHAK KEDUA. Akan tetapi, PIHAK PERTAMA wajib membantu proses penerbitan agar PIHAK KEDUA tidak mengalami kesulitan.

Pasal 3 Pelaksanaan Tugas

1. PIHAK PERTAMA wajib menjalankan tugas dengan baik dan memenuhi target pembuatan website e-commerce tersebut yang telah ditentukan oleh PIHAK KEDUA yang tertera di pasal 2 Poin Nomor 1.
2. PIHAK KEDUA berhak memberikan pengarahan mengenai Pengembangan Website yang sesuai dengan kebutuhan PIHAK KEDUA.
3. PIHAK PERTAMA berhak memberi masukan-masukan mengenai pembuatan Website.
4. PIHAK PERTAMA wajib menjaga kerahasiaan PIHAK KEDUA dan menyetujui perjanjian Kerahasiaan (non-disclosure agreement) yang telah disediakan oleh PIHAK KEDUA.
5. PIHAK KEDUA wajib menyediakan data-data dengan format yang tepat, sesuai dengan yang tercantum di Pasal 2 Point 2. Dan disediakan pada waktu tertentu sebelum fase pembuatan Website dimulai.

Pasal 4 Pengupahan

1. PIHAK KEDUA akan memberikan upah kepada PIHAK PERTAMA dengan komponen sebagai berikut :
 - a. Upah pembuatan Website : Rp. 1.500.000,00
 - b. Sudah termasuk hosting dan dibayarkan di akhir pekerjaan melalui bank ke nomor rekening PIHAK PERTAMA.
Penagih Upah akan dilakukan PIHAK PERTAMA melalui dokumen Invoice yang diberikan ke PIHAK KEDUA pada waktu komponen pekerjaan tersebut telah ditunaikan.
 - c. Pembayaran Inisial (Uang Muka) : 10% dari total Upah PIHAK PERTAMA.
 - d. Pembayaran termin final : 90% dari total Upah PIHAK PERTAMA.
2. Apabila PIHAK PERTAMA tidak melaksanakan tugas-tugasnya atau tidak memberi kabar kepada PIHAK KEDUA terkait pengembangan Aplikasi, PIHAK KEDUA berhak memberikan sanksi ataupun pemberhentian kontrak kepada PIHAK PERTAMA.

Pasal 5 Waktu Kerja

1. Waktu kerja PIHAK PERTAMA adalah 3 (tiga) bulan, dan sesuai dengan hari dan jam kerja yang berlaku di Indonesia.
2. Dalam kondisi tertentu PIHAK KEDUA dan PIHAK PERTAMA dapat menambahkan kesepakatan di waktu tertentu untuk pertemuan atau rapat guna membahas masalah-masalah yang terjadi di dalam pembuatan Website tersebut tanpa keluar dari skop pekerjaan yang telah disepakati bersama dan sesuai dengan Pasal 2 dan Pasal 3.
3. PIHAK KEDUA bersedia ditemui apabila PIHAK PERTAMA membutuhkan untuk meminta data terkait pembuatan Website yang sudah disepakati bersama sesuai dengan Pasal 3 dan Pasal 4

Pasal 6 Batasan Pekerjaan

1. Pekerjaan yang dilakukan oleh PIHAK PERTAMA hanya terbatas pada jasa pembuatan Website tersebut.
2. Adapun input data yang dibutuhkan dalam Website tersebut dilaksanakan oleh PIHAK PERTAMA

3. Pembayaran hosting dan biaya lain-lain yang dibutuhkan dalam pembuatan Website tersebut ditanggung oleh PIHAK KEDUA.
4. PIHAK KEDUA tidak memberikan informasi ataupun data apapun diluar pembuatan Website tersebut. Dan berkewajiban memberikan alasan kepada PIHAK PERTAMA agar dipahami bersama.
5. PIHAK PERTAMA akan memberikan source code dan aset desain antarmuka kepada PIHAK KEDUA ketika kontrak kerja berakhir ke dalam bentuk file digital yang dikirimkan melalui Email yang telah disepakati bersama
6. PIHAK PERTAMA bertanggung jawab kepada satu orang perwakilan dari PIHAK KEDUA, agar proses pelaporan pembuatan dapat fokus melalui 1 pintu informasi dari PIHAK KEDUA.
7. PIHAK PERTAMA berhak menolak suatu pekerjaan di luar skop pekerjaan yang telah disepakati bersama.
8. Apabila PIHAK KEDUA menyatakan tambahan Pekerjaan diluar kesepakatan Kontrak Kerja ini, wajib dibahas setelah kesepakatan yang telah berjalan berakhir.
9. PIHAK PERTAMA berkewajiban memperbaiki masalah (bug ataupun error) di website yang dilaporkan oleh PIHAK KEDUA.
10. PIHAK KEDUA berkewajiban melaporkan masalah (bug dan error) di website dengan informasi yang jelas dan waktu terjadinya.
11. Adapun apabila PIHAK KEDUA melaporkan masalah dan meminta mengubah (menambah ataupun mengurangi) fitur tersebut, maka dinyatakan sebagai tambahan Pekerjaan.
12. Perbaikan masalah oleh PIHAK PERTAMA yang dimaksud adalah agar fitur tersebut dapat berjalan dengan baik oleh pengguna Website, tanpa hambatan ataupun gangguan maupun crash.
13. PIHAK PERTAMA bersedia me-mentoring pekerjaan, melakukan perawatan (maintenance) Website yang telah dikembangkan sebagai masa garansi selama masa kontrak belum selesai dalam Pembuatan Website.

Pasal 7 Berakhirnya Perjanjian Kerja

1. PIHAK PERTAMA dan PIHAK KEDUA sepakat untuk mengakhiri Perjanjian Kerja dengan alasan dan atau keadaan sebagai berikut:
 - a. PIHAK PERTAMA telah melaksanakan tugasnya sesuai dengan skup pekerjaan yang telah disepakati.
 - b. PIHAK PERTAMA tidak melaksanakan tugasnya sampai dengan waktu pekerjaan berakhir
 - c. PIHAK PERTAMA dan PIHAK KEDUA melanggar aturan yang telah tercantum di Surat Perjanjian Kontrak ini.

Pasal 8 Perselisihan

1. Semua bentuk perselisihan antara PIHAK PERTAMA dan PIHAK KEDUA akan diselesaikan secara kekeluargaan dan dengan cara musyawarah muafakat.
2. Jika tidak terdapat kesepakatan dalam penyelesaian perselisihan antara PIHAK PERTAMA dan PIHAK KEDUA, maka kedua belah pihak sepakat untuk meminta bantuan kepada LPPHI (Lembaga Penyelesaian Perselisihan Hubungan Industrial) untuk mencari penyelesaiannya.

Pasal 9 Penandatanganan

1. Surat Perjanjian ini ditandatangani dalam keadaan sadar oleh kedua belah pihak, tanpa paksaan dari pihak manapun.
2. Surat Perjanjian ini dibuat rangkap 2 (dua), yaitu 1 (satu) untuk PIHAK PERTAMA, dan 1 (satu) PIHAK KEDUA, yang masing masing memiliki kekuatan hukum yang sama.
3. Dengan ditandatangani Surat Perjanjian Kontrak ini, maka kedua belah pihak sepakat untuk melaksanakan ketentuan-

ketentuan dalam perjanjian kontrak ini dengan demikian kedua belah pihak terkait demi hukum dalam suatu hubungan Kerjasama.

Yogyakarta, 30 Maret 2025

PIHAK PERTAMA PIHAK KEDUA

A handwritten signature in black ink, appearing to read 'Farel', with a large, dark, oval-shaped mark above the first letter.

Farel Cahyaning Bagaskara

SURAT PERJANJIAN KERJASAMA PENGEMBANGAN APLIKASI WEBSITE LAUNDRY

Pada hari ini, Senin tanggal 31 Maret 2025 bertempat di Yogyakarta, telah ditandatangani perjanjian kerja sama antara:

- **PIHAK PERTAMA:** Tim Sentry (Project Manager: Farel Cahyaning Bagaskara)
- **PIHAK KEDUA:** Fresh & Clean Laundry (Owner: Bapak Andi Prasetyo)

Pasal 1 – Ruang Lingkup Kerja

PIHAK PERTAMA mengembangkan Sistem Informasi Laundry berbasis website dengan fitur pemesanan online, pelacakan status cucian, dan notifikasi WhatsApp.

Pasal 2 – Jangka Waktu

Kerja sama berlaku selama 3 bulan sejak 1 September 2025 sampai 1 November 2025.

Pasal 3 – Pembiayaan

Nilai proyek sebesar Rp 1.500.000 yang mencakup domain, hosting, dan pengembangan website.

Pasal 4 – Penutup

Perjanjian ini disetujui oleh kedua belah pihak untuk dilaksanakan sebagaimana mestinya.

SURAT PERJANJIAN KERJASAMA PENGEMBANGAN APLIKASI WEBSITE LAUNDRY

Pada hari ini, Selasa tanggal **1 April 2025** bertempat di Tanjung Uban, Kepulauan Riau, telah ditandatangani perjanjian kerja sama antara:

- **PIHAK PERTAMA:** Tim Sentry (Project Manager: Farel Cahyaning Bagaskara)
- **PIHAK KEDUA:** Bintang Express Laundry (Owner: Ibu Siti Marlia)

Pasal 1 – Ruang Lingkup Kerja

Pengembangan website laundry dengan fitur pemesanan online, antar jemput, dan laporan transaksi digital.

Pasal 2 – Jangka Waktu

Kerja sama berlaku sejak **5 September 2025 sampai 5 November 2025**.

Pasal 3 – Pembiayaan

Nilai proyek sebesar **Rp 1.400.000** termasuk domain, hosting, dan pengembangan website.

Pasal 4 – Penutup

Disetujui kedua belah pihak untuk dilaksanakan sebagaimana mestinya.

SURAT PERJANJIAN KERJASAMA PENGEMBANGAN APLIKASI WEBSITE LAUNDRY

Pada hari ini, Rabu tanggal **2 April 2025** bertempat di Yogyakarta, telah ditandatangani perjanjian kerja sama antara:

- **PIHAK PERTAMA:** Tim Sentry (Project Manager: Farel Cahyaning Bagaskara)
- **PIHAK KEDUA:** Bungku Wash & Dry (Owner: Bapak Rudi Santoso)

Pasal 1 – Ruang Lingkup Kerja

Pengembangan sistem laundry berbasis web khusus untuk pelanggan mahasiswa/kost dengan fitur pemesanan kiloan, status laundry, dan riwayat transaksi.

Pasal 2 – Jangka Waktu

Kerja sama berlaku sejak **10 September 2025** sampai **10 November 2025**.

Pasal 3 – Pembiayaan

Nilai proyek sebesar **Rp 1.000.000** mencakup pengembangan website dan hosting.

Pasal 4 – Penutup

Disetujui kedua belah pihak untuk dilaksanakan sebagaimana mestinya.

SURAT PERJANJIAN KERJASAMA PENGEMBANGAN APLIKASI WEBSITE LAUNDRY

Pada hari ini, Kamis tanggal **3 April 2025** bertempat di Sleman, Yogyakarta, telah ditandatangani perjanjian kerja sama antara:

- **PIHAK PERTAMA:** Tim Sentry (Project Manager: Farel Cahyaning Bagaskara)
- **PIHAK KEDUA:** Sinar Bersih Laundry (Owner: Ibu Dewi Lestari)

Pasal 1 – Ruang Lingkup Kerja

Pengembangan website laundry dengan fitur paket bulanan, pemesanan online, dan antar jemput.

Pasal 2 – Jangka Waktu

Kerja sama berlaku sejak **15 September 2025 sampai 15 November 2025**.

Pasal 3 – Pembiayaan

Nilai proyek sebesar **Rp 1.300.000** termasuk domain dan biaya pengembangan.

Pasal 4 – Penutup

Disetujui kedua belah pihak untuk dilaksanakan sebagaimana mestinya.

SURAT PERJANJIAN KERJASAMA PENGEMBANGAN APLIKASI WEBSITE LAUNDRY

Pada hari ini, Jumat tanggal **4 April 2025** bertempat di Bantul, Yogyakarta, telah ditandatangani perjanjian kerja sama antara:

- **PIHAK PERTAMA:** Tim Sentry (Project Manager: Farel Cahyaning Bagaskara)
- **PIHAK KEDUA:** Cepat Berkah Laundry (Owner: Bapak Hendra Wijaya)

Pasal 1 – Ruang Lingkup Kerja

Pengembangan sistem informasi laundry berbasis web dengan fokus integrasi WhatsApp untuk layanan delivery.

Pasal 2 – Jangka Waktu

Kerja sama berlaku sejak **20 September 2025** sampai **20 November 2025**.

Pasal 3 – Pembiayaan

Nilai proyek sebesar **Rp 1.250.000** mencakup domain, hosting, dan implementasi sistem.

Pasal 4 – Penutup

Disetujui kedua belah pihak untuk dilaksanakan sebagaimana mestinya.

3. Logbook Kelompok

**LOG BOOK MANAJEMEN PROYEK TEKNOLOGI INFORMASI MAHASISWA
PROGRAM STUDI S1 INFORMATIKA UNIVERSITAS AHMAD DAHLAN T.A
2021/2022 (WAJIB DIISI DAN MASUK DALAM PENILAIAN)**

Tabel 4. 1 Logbook Kelompok

Nama Kelompok MPTI :SENTRY TEAM			
	N o	NIM	NAMA
	1	2200018390	M Reinza Chaniago
	2	2200018161	Syafiq Azhami Yusman
	3	2200018188	Farel Cahyaning Bagaskara
Judul Manajemen Tugas Proyek : Pembuatan Laundry			
Dosen Pembimbing : Dr. Ardiansyah ST, M.Cs			

Petunjuk Pengisian Log Book

1. Log book dibuat per minggu
2. Log book ditulis tangan
3. Setiap kegiatan di paraf oleh Mitra atau Client / dosen pembimbing MPTI
4. Log book per minggu diparaf oleh dosen pengampu kelas MPTI
5. Jumlah bimbingan minimal 4x Sebelum UTS

Logbook Minggu 4 sd 7 (sebelum UTS)

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan (Jika ada)	Paraf MITRA/CLIE NT	Paraf Dosen Pembimbing MPTI
		Hari/T G L	Du r asi (Ja m)				
1	Diskusi Susunan Kelompok &Penacarian Mitar	Rabu ,30, April,202 5	-	Struktur kelompok dan pencarian mitra	Tidak ada kendala		
2	Meeting dengan Klien	Kamis, 1, Mei, 2025	-	Metting awal dengan klien	Tidak ada kendala		
3	Pengumpulan data	Jumat, 2, Mei, 2025	-	Data awal dari mitra dikumpulkan	Tidak ada kendala		

4	Proses Pembuatan MOU	Minggu, 9, Mei, 2025	-	Draft MoU mulai dibuat	Tidak ada kendala		
5	Negosiasi dan Penawaran dengan Mitra	Sabtu, 6 Mei, 2025	-	Mitra Menyetujui ruang lingkup kerja sama dan rencana awal proyek	Tidak ada kendala		
6	Diskusi dengan Pembimbing (Jobdesk & Web)	Jumat, 14 Mei, 2025	-	Arahan pembimbing didapatkan terkait pembagian tugas dan rancangan web	Tidak ada kendala		
7	Pembuatan dan Penandatanganan MoU	Minggu, 18 Mei, 2025	-	MoU disepakati dan ditandatangani kedua pihak	Tidak ada kendala		

Logbook Minggu 9 sd 11 (setelah UTS)

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan (Jika ada)	Paraf MITRA/CLIENT	Paraf Dosen Pembimbing MPTI
		Hari/TGL	Jam				
1	Bimbingan bersama dosen pembimbing	Minggu, 10 Juni 2025	1 Jam	Laporan Progress			
2	Berkunjung Mitra	Sabtu, 6 Juli 2025	1 jam	Laporan Progress dengan mitra			
3	Rapat Internal ke-5	Minggu, 7 Juli 2025	2 Jam	Proses Integrasi dan Deployment			

4	Rapat Internal ke-6	Sabtu, 13 Juli 2025	1 Jam	Diskusi Hasil dari laporan dan deployment			
5	Bimbingan bersama dosen Pembimbing	Senin, 15 Juli 2025	1 Jam	Bimbingan terkait seminar MPTI			
6	Berkunjung Mitra	Rabu, 18 Juli 2025	1 Jam	Laporan Akhir dengan mitra dan meminta tanda tangan			
7	Rapat Internal ke-8	Kamis, 19 Juli 2025	1 Jam	Diskusi dan pendaftaran seminar MPTI			

Catatan Pembimbing Lapangan/Dosen Pembimbing MPTI/Dosen Pengampu Kelas MPTI :

.....

.....

.....

.....

.....

.....

Yogyakarta, 25 Juli 2025

Dosen Pengampu Kelas MPTI Ketua MPTI

4. Logbook Individu

**LOG BOOK INDIVIDU MANAJEMEN PROYEK TEKNOLOGI INFORMASI MAHASISWA
PROGRAM STUDI S1 INFORMATIKA UNIVERSITAS AHMAD DAHLAN T.A 2022/2023**
(WAJIB DIISI DAN TERLAMPIR DALAM LAPORAN)

Tabel 4. 2 Logbook Project Manager & Quality Assurance

Nama Kelompok MPTI : SentryTeam
Nim : 2200018188 Nama : Farel Cahyaning Bagaskara Jobdesk : Project Manager/Quality Assurance
Judul MPTI : Sistem Informasi Laundry Berbasis Web
Dosen Pembimbing MPTI : Dr. Ardiansyah ST, M.Cs

Petunjuk Pengisian Log Book

1. Log book dibuat per kegiatan
2. Log book per kegiatan individu dan di paraf oleh dosen pengampu kelas MPTI saat UTS

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan (Jika ada)
		Hari/TGL	Durasi (Jam)		
1	Rapat Internal ke-1	Rabu, 28 Maret 2025	1 Jam	Penentuan Dosen Pembimbing dan Calon Mitra	
2	Rapat Bersama Calon Mitra	Rabu, 29 Maret 2025	1 Jam	Penjelasan Jobdesk yang akan dilakukan selama bekerjasama dengan mitra	Terdapat jobdesk diluar proyek
3	Mendatangi Mitra	Selasa, 5	2 Jam	perkenalan antar anggota tim dan mitra sesuai	

		April 2025		jobdesk masing-masing dan mengidentifikasi masalah yang akan dijadikan proyek	
4	Identifikasi Tujuan dan lingkup proyek	Selasa, 5 April 2025	1 Jam	mengidentifikasi tujuan dan ruang lingkup proyek	
4	Rapat Internal ke-2	Senin, 6 April 2025	1 Jam	Brainstorming proyek yang sudah dibahas dengan calon mitra	
	Penyusunan Rencana Proyek, Penjadwalan dan Anggaran	Sabtu, 9 April 2025	1 Jam	Penyusunan Rencana Proyek	
5	Mendatangi calon client/mitra	Rabu, 13 April 2025	2 Jam	Wawancara dengan calon client pertama dan mengetahui spesifikasi web yang dibutuhkan oleh calon client pertama	
6	Rapat Internal ke-3	Senin, 18 April 2025	1 Jam	Membahas Jobdesk masing-masing team dan merancang proposal MOU	
7	Bimbingan Bersama dosen pembimbing	Jum'at, 26 April 2025	1 Jam	Meminta saran terkait efisiensi waktu dan pengerjaan proyek kedepannya	

8	Rapat Internal ke-4	Sabtu, 20 April 2025	1 Jam	Pembagian tugas masing masing jobdesk dan timeline pengerjaan tugas	
9					

Catatan Pembimbing Lapangan/Dosen Pembimbing MPTI/Dosen Pengampu Kelas MPTI :

Bimbingan bersama dosen pembimbing	Minggu, 10 Juni 2025	1 Jam	Laporan Progress
Berkunjung Mitra	Sabtu, 6 Juli 2025	1 jam	Laporan Progress dengan mitra

Rapat Internal ke-5	Minggu, 7 Juli 2025	2 Jam	Proses Integrasi dan Deployment
Rapat Internal ke-6	Sabtu, 13 Juli 2025	1 Jam	Diskusi Hasil dari laporan dan deployment
Bimbingan bersama dosen Pembimbing	Senin, 15 Juli 2025	1 Jam	Bimbingan terkait seminar MPTI
Mmebuat Rencana Pengujian	Rabu, 18 Juli 2025	1 Jam	Dokumen pengujian
Membuat Testcase Scenario	Kamis, 19 Juli 2025	1 Jam	Dokumen testcase

Catatan Pembimbing Lapangan/Dosen Pembimbing MPTI/Dosen Pengampu Kelas MPTI :

Yogyakarta, 17 Mei 2025

Ketua MPTI

LOG BOOK INDIVIDU MANAJEMEN PROYEK TEKNOLOGI INFORMASI MAHASISWA
PROGRAM STUDI S1 INFORMATIKA UNIVERSITAS AHMAD DAHLAN T.A 2022/2023
(WAJIB DIISI DAN TERLAMPIR DALAM LAPORAN)

Tabel 4. 3 Sekretaris & Backend

Nama Kelompok MPTI : Sentry team
Nim : 220018161 Nama : Syafiq Azhami Yusman Jobdesk : Sekretaris & Backend
Judul MPTI : Sistem Informasi Laundry Berbasis Web
Dosen Pembimbing MPTI : Dr. Ardiansyah ST, M.Cs

Petunjuk Pengisian Log Book

1. Log book dibuat per kegiatan
2. Log book per kegiatan individu dan di paraf oleh dosen pengampu kelas MPTI saat UTS

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan (Jika ada)
		Hari/TGL	Durasi (Jam)		

1	Analisis kebutuhan sistem bersama klien	Rabu, 28 Februari 2025	1 Jam	Penentuan Dosen Pembimbing dan Calon Mitra	
2	Pembuatan spesifikasi fungsional dan non-fungsional	Rabu, 29 Februari 2025	1 Jam	Penjelasan Jobdesk yang akan dilakukan selama bekerjasama dengan mitra	Terdapat jobdesk diluar proyek
3	Merancang struktur database awal (ERD)	Selasa, 5	2 Jam	perkenalan antar anggota tim dan mitra sesuai	

		Maret 2025		jobdesk masing-masing dan mengidentifikasi masalah yang akan dijadikan proyek	
4	Setup backend Laravel & konfigurasi koneksi database	Selasa, 5 Maret 2025	1 Jam	mengidentifikasi tujuan dan ruang lingkup proyek	

5	Implementasi CRUD Layanan & User	Senin, 6 Maret 2025	1 Jam	Brainstorming proyek yang sudah dibahas dengan calon mitra	
---	----------------------------------	---------------------	-------	--	--

6	Membuat API untuk pemesanan & status pesanan	Sabtu, 9 Maret 2025	1 Jam	Penyusunan Rencana Proyek	
7	Rancangan <i>Entity Relationship</i> Diagram	Rabu, 13 Maret 2025	2 Jam	Wawancara dengan calon client pertama dan mengetahui spesifikasi web yang dibutuhkan oleh calon client pertama	
8	Membuat <i>Use Case</i> Diagram Kebutuhan Sistem	Senin, 18 Maret 2025	1 Jam	Membahas Jobdesk masing-masing team dan merancang proposal MOU	
9	Integrasi notifikasi WhatsApp ke backend	Jum'at, 26 Maret 2025	1 Jam	Meminta saran terkait efisiensi waktu dan pengerjaan proyek kedepannya	

10	Bug fixing dan debugging backend	Sabtu, 20 April 2025	1 Jam	Pembagian tugas masing masing jobdesk dan timeline pengerjaan tugas	
----	----------------------------------	----------------------	-------	---	--

Catatan Pembimbing Lapangan/Dosen Pembimbing MPTI/Dosen Pengampu Kelas MPTI :

Yogyakarta, 17 Mei 2025

Ketua MPTI

(Farel Cahyaning Bagaskara)

LOG BOOK MANAJEMEN PROYEK TEKNOLOGI INFORMASI MAHASISWA PROGRAM STUDI S1 INFORMATIKA UNIVERSITAS AHMAD DAHLAN T.A 2021/2022

(WAJIB DIISI DAN MASUK DALAM PENILAIAN)

Nama Kelompok MPTI : SENTRY TEAM			
	No	NIM	NAMA
	1	2200018188	Farel Cahyaning Bagaskara
	2	2200018161	Syafiq Azhami Yusman
	3	2200018390	M Reinza Chaniago
Judul Manajemen Tugas Proyek : Sistem Informasi Laundry Berbasis Website			
Dosen Pembimbing : Dr. Ardiansyah ST, M.Cs			

Petunjuk Pengisian Log Book

1. Log book dibuat per minggu
2. Log book ditulis tangan
3. Setiap kegiatan di paraf oleh Mitra Atau Client / dosen pembimbing MPTI
4. Log book per minggu di paraf oleh dosen pengampu kelas MPTI
5. Jumlah bimbingan minimal 7x Sebelum Daftar Seminar MPTI

Logbook Minggu 9 sd 11 (setelah UTS)

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan (Jika ada)	Paraf MITRA/CLIENT	Paraf Dosen Pembimbing MPTI
		Hari/TGL	Jam				
1	Bimbingan bersama dosen pembimbing	Minggu, 10 Juni 2025	1 Jam	Laporan Progress			
2	Berkunjung Mitra	Sabtu, 6 Juli 2025	1 jam	Laporan Progress dengan mitra			
3	Rapat Internal ke-5	Minggu, 7 Juli 2025	2 Jam	Proses Integrasi dan Deployment			

4	Rapat Internal ke-6	Sabtu, 13 Juli 2025	1 Jam	Diskusi Hasil dari laporan dan deployment			
5	Bimbingan bersama dosen Pembimbing	Senin, 15 Juli 2025	1 Jam	Bimbingan terkait seminar MPTI			
6	Berkunjung Mitra	Rabu, 18 Juli 2025	1 Jam	Laporan Akhir dengan mitra dan meminta tanda tangan			
7	Rapat Internal ke-8	Kamis, 19 Juli 2025	1 Jam	Diskusi dan pendaftaran seminar MPTI			

Catatan Pembimbing Lapangan/Dosen Pembimbing MPTI/Dosen Pengampu Kelas MPTI :

Yogyakarta, 1 Agustus 2025

Ketua MPTI

**LOG BOOK INDIVIDU MANAJEMEN PROYEK TEKNOLOGI
INFORMASI MAHASISWAPROGRAM STUDI S1
INFORMATIKA UNIVERSITAS AHMAD DAHLAN T.A
2022/2023(WAJIB DIISI DAN TERLAMPIR DALAM LAPORAN)**

Tabel 4. 5 Logbook Frontend Developer

Nama Kelompok MPTI : Sentry Team
Nim : 22000390 Nama : M Reinza Chaniago Jobdesk : Frontend Developer
Judul MPTI : Sistem Informasi Laundry Berbasis Website
Dosen Pembimbing MPTI : Dr. Ardiansyah ST, M.Cs

Petunjuk Pengisian Log Book

5. Log book dibuat per kegiatan
6. Log book per kegiatan individu dan di paraf oleh dosen pengampu kelas MPTI saat UTS

Logbook Minggu 4 sd 7 (sebelum UTS)

		Waktu Pelaksanaan		
--	--	--------------------------	--	--

No	Kegiatan	Hari/TGL	Durasi (Jam)	Hasil	Kendala, Rencana Perubahan (Jika ada)
1	Menghasilkan desain dari menu login dan registrasi	Rabu, 28 Februari 2025	1 Jam	Penentuan Dosen Pembimbing dan Calon Mitra	
2	Menghasilkan desain yang baru dari fitur login dan register	Rabu, 29 Februari 2025	1 Jam	Penjelasan Jobdesk yang akan dilakukan selama bekerjasama dengan mitra	Terdapat jobdesk diluar proyek
3	Menghasilkan desain yang baru dari fitur login dan register	Selasa, 5	2 Jam	perkenalan antar anggota tim dan mitra sesuai	

		Maret 2025		jobdesk masing-masing dan mengidentifikasi masalah yang akan dijadikan proyek	
4	Membuat Fitur Cetak Laporan	Selasa, 5 Maret 2025	1 Jam	mengidentifikasi tujuan dan ruang lingkup proyek	

Catatan Pembimbing Lapangan/Dosen Pembimbing MPTI/Dosen Pengampu Kelas MPTI :

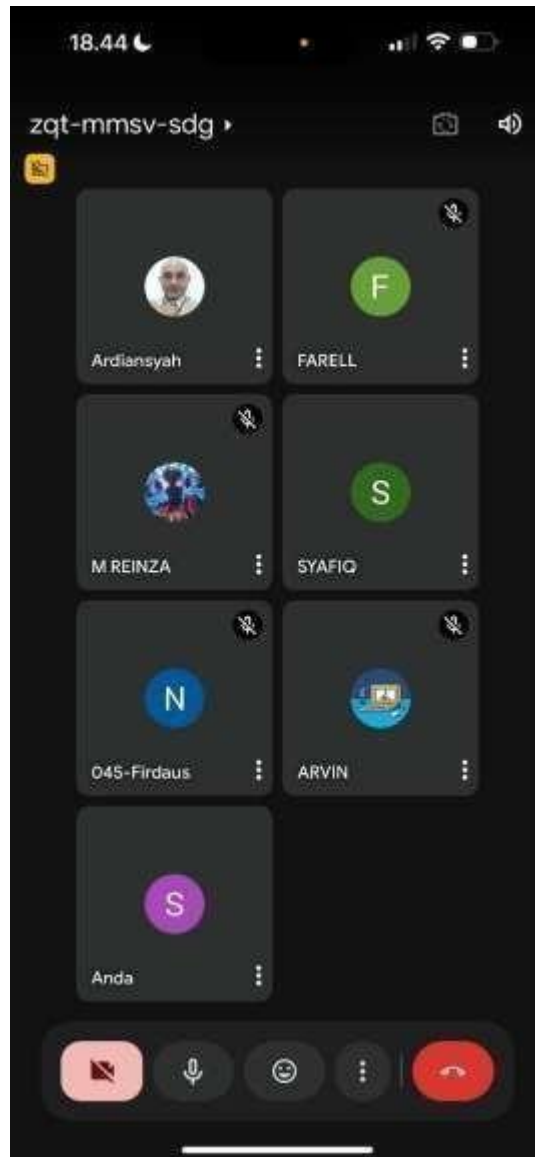
Yogyakarta, 17 Mei 2025

Ketua MPTI

(Farel Cahyaning Bagaskara)

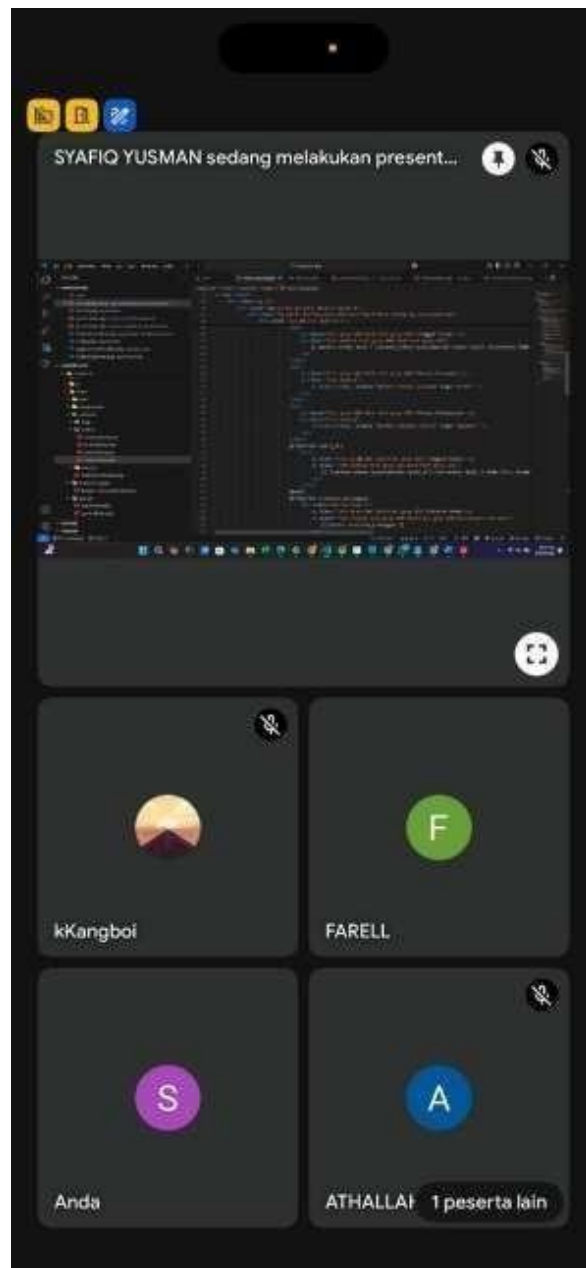
5. Foto Dokumentasi Kegiatan Proyek

Gambar 4. 1 Bimbingan Bersama Dosen Pembimbing



Gambar 4.1 merupakan dokumentasi ketika bimbingan bersama dosen pembimbing

Gambar 4. 2 Dokumentasi Rapat /Pengerjaan Online/Offline





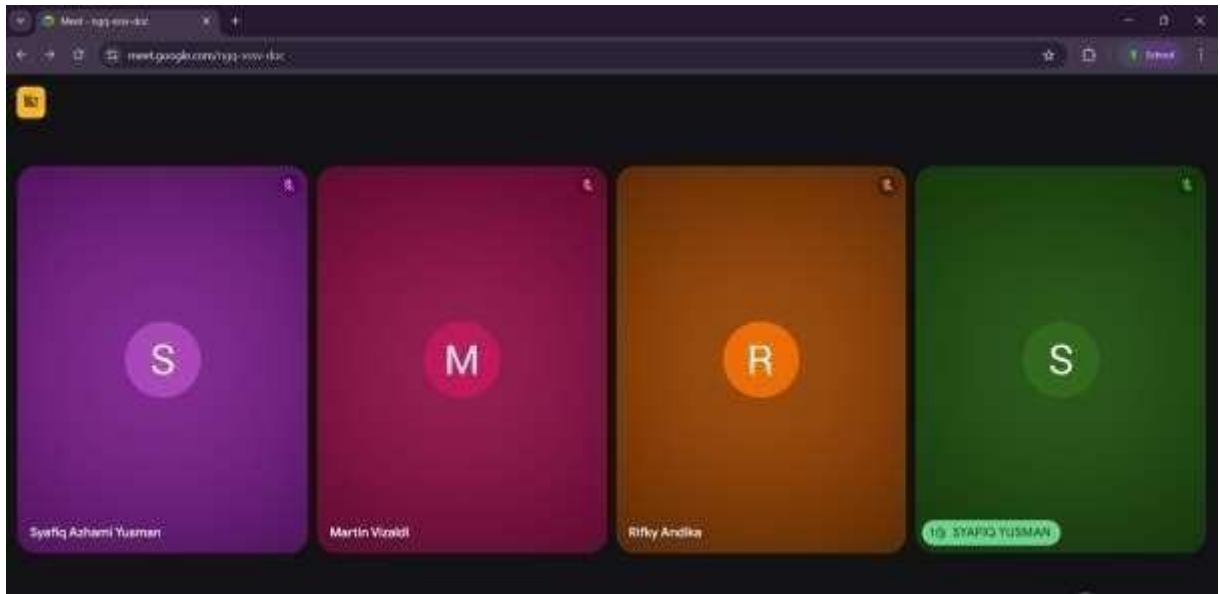
Gambar 4.2 merupakan kegiatan ketika melakukan rapat secara online

Gambar 4. 3 Presentasi ke Mitra



Gambar 4.3 Sekretaeis melakukan presentasi ke pihak mitra client

Gambar 4. 4 Serah Terima Website



Gambar 4.4 serah terima website selanjutnya tertera di berita acara

6. Berita Acara

BERITA ACARA SERAH TERIMA PROYEK SISTEM INFORMASI LAUNDRY “MAMAK LAUNDRY”

Pada hari ini, **Senin, tanggal 29 Juli 2025**, bertempat di **Tanjung Uban Utara, Kabupaten Bintan, Kepulauan Riau**, telah dilaksanakan Serah Terima Proyek Sistem Informasi Laundry berbasis website antara:

PIHAK PERTAMA

Nama : Farel Cahyaning Bagaskara
Jabatan : Project Manager Tim Pengembang
Alamat : Universitas Ahmad Dahlan, Yogyakarta

PIHAK KEDUA

Nama : Dyah Irma Gendraningtyas
Jabatan : Pemilik Usaha Mamak Laundry
Alamat : Tanjung Uban Utara, Kabupaten Bintan, Kepulauan Riau

Kedua belah pihak menyatakan bahwa:

1. PIHAK PERTAMA telah menyelesaikan pengembangan **Sistem Informasi Laundry berbasis Website** yang meliputi fitur pemesanan layanan, status pesanan, notifikasi WhatsApp, serta pengelolaan data pelanggan dan layanan.
2. PIHAK KEDUA telah melakukan pengecekan dan menerima hasil pekerjaan tersebut sesuai dengan kesepakatan yang telah dibuat dalam MoU.
3. PIHAK PERTAMA akan memberikan masa pendampingan dan perawatan sistem hingga **2 minggu setelah serah terima**, sebagai masa garansi.

Demikian Berita Acara ini dibuat dan ditandatangani oleh kedua belah pihak dengan penuh kesadaran dan tanpa paksaan dari pihak mana pun, untuk dipergunakan sebagaimana mestinya.

Tanjung Uban Utara, 29 Juli 2025

PIHAK PERTAMA
Project Manager

PIHAK KEDUA
Pemilik Mamak Laundry

A handwritten signature in black ink, featuring a stylized 'F' and 'C' followed by 'Bagaskara'.

Farel Cahyaning Bagaskara

A handwritten signature in black ink, featuring a stylized 'D' and 'I' followed by 'Gendraningtyas'.

Dyah Irma Gendraningtyas

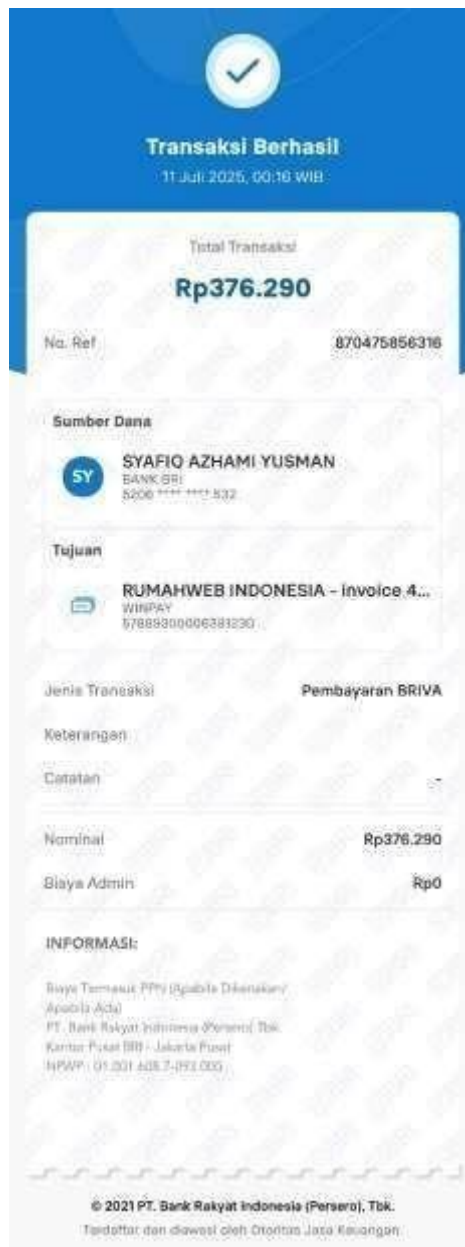
7. Bukti Pembiayaan

1. Pembayaran Mitra ke Tim Sentry

Gambar 4. 5 Pembayaran Mitra



2. Pengeluaran pembayaran/pembelian domain



Gambar 4. 6 Pengeluaran Mitra

8. Tools

Source Code Tools

APP

LaporanPesanan.php

<?php

```
namespace App\Filament\Pages;
```

```
use Filament\Pages\Page;
```

```
use Carbon\Carbon;
```

```
use Filament\Forms\Components\DatePicker;
```

```
use Filament\Forms\Concerns\InteractsWithForms;
```

```
use Filament\Forms\Contracts\HasForms;
```

```

use Filament\Forms\Form;
use App\Filament\Widgets\LaporanStatsOverviewWidget; // Import widget
use App\Filament\Widgets\LaporanOrderTableWidget; // Import widget tabel juga
use App\Models\Order;
use App\Models\User;
use Filament\Actions\Action; // Import Action
use Barryvdh\DomPDF\Facade\Pdf; // Import Facade PDF
use Dompdf\Options;
use Illuminate\Support\Facades\Blade;

class LaporanPesanan extends Page implements HasForms
{
    use InteractsWithForms;

    protected static ?string $navigationIcon = 'heroicon-o-document-chart-bar';
    protected static ?string $navigationLabel = 'Laporan Pesanan';
    protected static ?string $navigationGroup = 'Laporan';
    protected static ?string $title = 'Laporan Pesanan';
    protected static string $view = 'filament.pages.laporan-pesanan'; // Blade view kita

    public ?string $startDate = null;
    public ?string $endDate = null;

    // Tidak perlu properti statistik di sini lagi ($totalOrders, dll.)

    public function mount(): void
    {
        $this->startDate = Carbon::now()->startOfMonth()->toDateString();
        $this->endDate = Carbon::now()->endOfMonth()->toDateString();
        $this->dispatchReportUpdate();
    }

    public function form(Form $form): Form
    {
        return $form
            ->schema([
                DatePicker::make('startDate')
                    ->label('Tanggal Mulai')
                    ->live()
                    ->afterStateUpdated(function () {
                        $this->dispatchReportUpdate();
                    }),
                DatePicker::make('endDate')
                    ->label('Tanggal Selesai')
                    ->live()
                    ->afterStateUpdated(function () {
                        $this->dispatchReportUpdate();
                    }),
            ])
            ->columns(2)
            ->fill([
                'startDate' => $this->startDate,
            ]);
    }
}

```

```

        'endDate' => $this->endDate,
    ]);
}

protected function dispatchReportUpdate(): void
{
    // Kirim event ke kedua widget (Stats dan Tabel)
    $filters = ['startDate' => $this->startDate, 'endDate' => $this->endDate];
    $this->dispatch('updateLaporanStats', filters: $filters); // Untuk
StatsOverviewWidget
    $this->dispatch('dateRangeUpdatedForTable', filters: $filters); // Untuk
LaporanOrderTableWidget
}

public function filterToday(): void
{
    $this->startDate = Carbon::now()->toDateString();
    $this->endDate = Carbon::now()->toDateString();
    $this->form->fill(['startDate' => $this->startDate, 'endDate' => $this->endDate]);
    $this->dispatchReportUpdate();
}

public function filterThisMonth(): void
{
    $this->startDate = Carbon::now()->startOfMonth()->toDateString();
    $this->endDate = Carbon::now()->endOfMonth()->toDateString();
    $this->form->fill(['startDate' => $this->startDate, 'endDate' => $this->endDate]);
    $this->dispatchReportUpdate();
}

public function filterLastMonth(): void
{
    $this->startDate = Carbon::now()->subMonthNoOverflow()->startOfMonth()-
>toDateString();
    $this->endDate = Carbon::now()->subMonthNoOverflow()->endOfMonth()-
>toDateString();
    $this->form->fill(['startDate' => $this->startDate, 'endDate' => $this->endDate]);
    $this->dispatchReportUpdate();
}

// Mendaftarkan widget yang akan ditampilkan di halaman ini
protected function getHeaderWidgets(): array
{
    return [
        LaporanStatsOverviewWidget::class, // Widget statistik kita
    ];
}

protected function getFooterWidgets(): array
{
    return [
        LaporanOrderTableWidget::class, // Widget tabel kita
    ];
}

```

```

    ];
}
protected function getHeaderActions(): array
{
    return [
        Action::make('printPdf')
            ->label('Cetak PDF')
            ->icon('heroicon-o-printer')
            ->color('gray')
            ->action(function () {
                $start = Carbon::parse($this->startDate)->startOfDay();
                $end = Carbon::parse($this->endDate)->endOfDay();

                $baseQueryPdf = Order::whereBetween('created_at', [$start, $end]);
                $pdfTotalOrders = (clone $baseQueryPdf)->count();
                $pdfTotalRevenue = (clone $baseQueryPdf)->where('payment_status',
'lunas')->sum('total_amount');
                $pdfNewCustomers = User::where('role', 'pelanggan')
                    ->whereHas('orders', function ($q) use ($start, $end) {
                        $q->whereBetween('created_at', [$start, $end]);
                    })
                    ->whereDoesntHave('orders', function ($q) use ($start) {
                        $q->where('created_at', '<', $start);
                    })
                    ->count();

                $pdfOrdersData = (clone $baseQueryPdf)
                    ->with('customer:id,name')
                    ->select('kode_pesanan', 'user_id', 'order_date', 'total_amount',
'status_pesanan', 'payment_status')
                    ->orderBy('order_date', 'desc')
                    ->get()
                    ->map(function ($order) {
                        return $order->toArray();
                    })
                    ->all();

                $dataForPdf = [
                    'startDate' => $this->startDate,
                    'endDate' => $this->endDate,
                    'totalOrders' => $pdfTotalOrders,
                    'totalRevenue' => $pdfTotalRevenue,
                    'newCustomers' => $pdfNewCustomers,
                    'ordersData' => $pdfOrdersData,
                ];

                $fileName = 'laporan-pesanan-' . Carbon::parse($this->startDate)-
>format('Ymd') . '-' . Carbon::parse($this->endDate)->format('Ymd') . '.pdf';

                // Menggunakan array untuk setOptions
                $pdf = Pdf::setOptions([
                    'isHtml5ParserEnabled' => true,

```

```

        'isRemoteEnabled' => true, // Untuk gambar/CSS eksternal jika ada
        'defaultCharset' => 'UTF-8', // Opsi untuk charset
    ])->loadView('pdf.laporan-pesanan', $dataForPdf)
        ->setPaper('a4', 'portrait');

    return response()->streamDownload(function () use ($pdf) {
        echo $pdf->output();
    }, $fileName);
    }},
    ];
}
}

```

APP

LaporanOrderTableWidget

<?php

```
namespace App\Filament\Widgets;
```

```
use Filament\Tables;
```

```
use Filament\Tables\Table;
```

```
use Filament\Widgets\TableWidget as BaseWidget;
```

```
use App\Models\Order;
```

```
use Carbon\Carbon;
```

```
use Illuminate\Database\Eloquent\Builder;
```

```
class LaporanOrderTableWidget extends BaseWidget
```

```
{
```

```
    protected static ?int $sort = 3;
```

```
    protected int|string|array $columnSpan = 'full';
```

```
    public ?string $startDate = null;
```

```
    public ?string $endDate = null;
```

```
    protected $listeners = ['dateRangeUpdatedForTable' => 'applyDateRangeFilter'];
```

```
    public function mount(): void
```

```
    {
```

```
        $this->startDate = $this->startDate ?? Carbon::now()->startOfMonth()->toDateString();
```

```
        $this->endDate = $this->endDate ?? Carbon::now()->endOfMonth()->toDateString();
```

```
    }
```

```
    public function applyDateRangeFilter(array $filters): void
```

```
    {
```

```
        $this->startDate = $filters['startDate'] ?? $this->startDate;
```

```
        $this->endDate = $filters['endDate'] ?? $this->endDate;
```

```
        $this->resetTable();
```

}

```

protected function getTableQuery(): Builder
{
    $start = Carbon::parse($this->startDate)->startOfDay();
    $end = Carbon::parse($this->endDate)->endOfDay();

    return Order::query()
        ->with('customer:id,name', 'processedByAdmin:id,name') // Eager load dengan
        kolom spesifik
        ->whereBetween('orders.created_at', [$start, $end])
        // Default order by bisa dihapus atau dibiarkan, Filament akan menyimpannya
        // ->orderBy('orders.order_date', 'desc');
        ; // Biarkan Filament yang menangani default sorting awal jika tidak ada preferensi
    kuat
}

protected function getTableColumns(): array
{
    return [
        Tables\Columns\TextColumn::make('kode_pesanan')
            ->searchable()
            ->sortable()
            ->toggleable(isToggledHiddenByDefault: true),

        // OPSI 1 untuk customer.name (coba ini dulu)
        Tables\Columns\TextColumn::make('customer.name')
            ->label('Pelanggan')
            ->searchable()
            ->sortable(),

        /*
        // OPSI 2 untuk customer.name (jika OPSI 1 tidak berfungsi untuk sorting)
        Tables\Columns\TextColumn::make('customer.name')
            ->label('Pelanggan')
            ->searchable()
            ->sortable(query: function (Builder $query, string $direction): Builder {
                // Pastikan alias 'customers' tidak bentrok jika ada join lain
                return $query
                    ->leftJoin('users as customers', 'orders.user_id', '=', 'customers.id')
                    ->orderBy('customers.name', $direction);
            }),
        */

        Tables\Columns\TextColumn::make('order_date')
            ->label('Tgl Pesan')
            ->dateTime('d M Y, H:i')
            ->sortable(),

        Tables\Columns\TextColumn::make('total_amount')
            ->label('Total Harga')
            ->money('IDR')
            ->sortable(),
    ];
}

```

```

Tables\Columns\TextColumn::make('status_pesanan')
    ->badge()
    ->color(fn(string $state): string => match ($state) {
        'baru' => 'gray',
        'diproses' => 'warning',
        'siap diambil' => 'info',
        'selesai' => 'success',
        'dibatalkan' => 'danger',
        default => 'primary',
    })
    ->searchable()
    ->sortable(),

Tables\Columns\TextColumn::make('payment_status')
    ->label('Pembayaran')
    ->badge()
    ->color(fn(string $state): string => match ($state) {
        'belum_bayar' => 'danger',
        'lunas' => 'success',
        default => 'gray',
    })
    ->searchable()
    ->sortable(),

// OPSI 1 untuk processedByAdmin.name (coba ini dulu)
Tables\Columns\TextColumn::make('processedByAdmin.name')
    ->label('Diproses Oleh')
    ->searchable()
    ->sortable()
    ->toggleable(isToggledHiddenByDefault: true),

/*
// OPSI 2 untuk processedByAdmin.name (jika OPSI 1 tidak berfungsi untuk
sorting)
Tables\Columns\TextColumn::make('processedByAdmin.name')
    ->label('Diproses Oleh')
    ->searchable()
    ->sortable(query: function (Builder $query, string $direction): Builder {
        // Pastikan alias 'admins' tidak bentrok
        return $query
            ->leftJoin('users as admins', 'orders.processed_by_admin_id', '=',
'admins.id')
            ->orderBy('admins.name', $direction);
    })
    ->toggleable(isToggledHiddenByDefault: true),
*/

Tables\Columns\TextColumn::make('paid_at')
    ->label('Tgl Bayar')
    ->dateTime('d M Y, H:i')
    ->sortable()

```

```

        ->toggableable(isToggledHiddenByDefault: true),
    ];
}

protected function isTablePaginationEnabled(): bool
{
    return true;
}
protected function getTableFilters(): array
{
    return [];
}
protected function getTableHeaderActions(): array
{
    return [];
}
protected function getTableActions(): array
{
    return [];
}
protected function getTableBulkActions(): array
{
    return [];
}
}

```

APP

LaporanStatsOverviewWidget

<?php

namespace App\Filament\Widgets;

use Filament\Widgets\StatsOverviewWidget as BaseWidget;

use Filament\Widgets\StatsOverviewWidget\Stat; // Di v3, ini adalah Stat

use App\Models\Order;

use App\Models\User;

use Carbon\Carbon;

class LaporanStatsOverviewWidget extends BaseWidget

{

protected static ?int \$sort = -3; // Agar muncul di atas widget lain jika di dashboard
protected int|string|array \$columnSpan = 'full';

// Properti untuk menerima filter tanggal

public ?string \$startDate = null;

public ?string \$endDate = null;

// Listener untuk event dari halaman kustom

// Nama listener harus sesuai dengan nama metode tanpa 'on' dan huruf pertama

kecil

```
protected $listeners = ['updateLaporanStats' => 'updateFilters'];

public function mount(): void
{
    $this->startDate = $this->startDate ?? Carbon::now()->startOfMonth()->toDateString();
    $this->endDate = $this->endDate ?? Carbon::now()->endOfMonth()->toDateString();
}

public function updateFilters(array $filters): void
{
    $this->startDate = $filters['startDate'] ?? $this->startDate;
    $this->endDate = $filters['endDate'] ?? $this->endDate;
    // Widget akan otomatis me-refresh datanya karena perubahan properti yang di-track Livewire
}

protected function getStats(): array // Di v3 tetap getStats()
{
    // Pastikan startDate dan endDate tidak null sebelum di-parse
    $start = $this->startDate ? Carbon::parse($this->startDate)->startOfDay() : Carbon::now()->startOfMonth()->startOfDay();
    $end = $this->endDate ? Carbon::parse($this->endDate)->endOfDay() : Carbon::now()->endOfMonth()->endOfDay();

    $query = Order::whereBetween('created_at', [$start, $end]);

    $totalOrders = (clone $query)->count();
    $totalRevenue = (clone $query)->where('payment_status', 'lunas')->sum('total_amount');

    $newCustomers = User::where('role', 'pelanggan')
        ->whereHas('orders', function ($q) use ($start, $end) {
            $q->whereBetween('created_at', [$start, $end]);
        })
        ->whereDoesntHave('orders', function ($q) use ($start) {
            $q->where('created_at', '<', $start);
        })
        ->count();

    return [
        Stat::make('Total Pesanan', $this->totalOrders ?? $totalOrders) // Ambil dari properti jika sudah di-set
            ->description('Pesanan dalam periode')
            ->descriptionIcon('heroicon-o-shopping-cart')
            ->color('primary')
            ->chart($this->generateDummyChartData()) // Contoh chart kecil
            ->extraAttributes([
                'class' => 'cursor-pointer', // Contoh atribut tambahan
            ]),
        Stat::make('Total Pendapatan (Lunas)', 'Rp ' . number_format($this->
```

```

>totalRevenue ?? $totalRevenue, 0, ',', '.'))
    ->description('Pendapatan dari pesanan lunas')
    ->descriptionIcon('heroicon-o-currency-dollar')
    ->color('success')
    ->chart($this->generateDummyChartData(true)),
Stat::make('Pelanggan Baru', $this->newCustomers ?? $newCustomers)
    ->description('Pelanggan order pertama kali')
    ->descriptionIcon('heroicon-o-user-plus')
    ->color('info')
    ->chart($this->generateDummyChartData()),
];
}

// Fungsi dummy untuk data chart kecil di kartu stat
protected function generateDummyChartData(bool $reverse = false): array
{
    $data = collect(range(1, 7))->map(fn() => random_int(5, 20))->toArray();
    return $reverse ? array_reverse($data) : $data;
}
}

```

APP

AdminDashboardController

```

<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;

class AdminDashboardController extends Controller
{
    public function index()
    {
        return view('admin.dashboard');
    }
}

```

APP

AdminOrderController

```

<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\Order;

```

```

class AdminOrderController extends Controller
{
    public function update(Request $request, $id)
    {
        $order = Order::findOrFail($id);

        if ($request->status_pesanan !== $order->status_pesanan) {
            $order->status_pesanan = $request->status_pesanan;
            $order->save(); // Akan memicu observer
        }

        return back()->with('success', 'Status berhasil diperbarui!');
    }
}

```

APP

AuthenticatedSessionController

<?php

namespace App\Http\Controllers\Auth;

```

use App\Http\Controllers\Controller;
use App\Http\Requests\Auth\LoginRequest;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Route;
use Illuminate\View\View;

```

class AuthenticatedSessionController extends Controller

```

{
    public function create(): View
    {
        return view('auth.login');
    }

    public function store(LoginRequest $request): RedirectResponse
    {
        $request->authenticate();

        $request->session()->regenerate();

        if (Auth::user()->role === 'admin') {
            $adminPath = filament()->getDefaultPanel()->getPath();
            return redirect()->intended($adminPath);
        }

        return redirect()->intended(route('customer.dashboard'));
    }
}

```

```

    }

    public function destroy(Request $request): RedirectResponse
    {
        Auth::guard('web')->logout();

        $request->session()->invalidate();

        $request->session()->regenerateToken();

        return redirect()->route('login');
    }
}

```

APP

ConfirmablePasswordController

```
<?php
```

```

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Validation\ValidationException;
use Illuminate\View\View;

class ConfirmablePasswordController extends Controller
{
    /**
     * Show the confirm password view.
     */
    public function show(): View
    {
        return view('auth.confirm-password');
    }

    /**
     * Confirm the user's password.
     */
    public function store(Request $request): RedirectResponse
    {
        if (! Auth::guard('web')->validate([
            'email' => $request->user()->email,
            'password' => $request->password,
        ])) {
            throw ValidationException::withMessages([
                'password' => _('auth.password'),
            ]);
        }
    }
}

```

```

    }

    $request->session()->put('auth.password_confirmed_at', time());

    return redirect()->intended(route('dashboard', absolute: false));
}
}

```

APP

EmailVerificationNotificationController

<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;

class EmailVerificationNotificationController extends Controller

```

{
    /**
     * Send a new email verification notification.
     */
    public function store(Request $request): RedirectResponse
    {
        if ($request->user()->hasVerifiedEmail()) {
            return redirect()->intended(route('dashboard', absolute: false));
        }

        $request->user()->sendEmailVerificationNotification();

        return back()->with('status', 'verification-link-sent');
    }
}

```

APP

EmailVerificationPromptController

<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\View\View;

class EmailVerificationPromptController extends Controller

```

{

```

```

/**
 * Display the email verification prompt.
 */
public function __invoke(Request $request): RedirectResponse|View
{
    return $request->user()->hasVerifiedEmail()
        ? redirect()->intended(route('dashboard', absolute: false))
        : view('auth.verify-email');
}
}

```

APP

NewPasswordController

<?php

namespace App\Http\Controllers\Auth;

```

use App\Http\Controllers\Controller;
use App\Models\User;
use Illuminate\Auth\Events>PasswordReset;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades>Password;
use Illuminate\Support\Str;
use Illuminate\Validation\Rules;
use Illuminate\View\View;

```

class NewPasswordController extends Controller

```

{
    /**
     * Display the password reset view.
     */
    public function create(Request $request): View
    {
        return view('auth.reset-password', ['request' => $request]);
    }

    /**
     * Handle an incoming new password request.
     *
     * @throws \Illuminate\Validation\ValidationException
     */
    public function store(Request $request): RedirectResponse
    {
        $request->validate([
            'token' => ['required'],
            'email' => ['required', 'email'],
            'password' => ['required', 'confirmed', Rules\Password::defaults()],
        ]);
    }
}

```

```

// Here we will attempt to reset the user's password. If it is successful we
// will update the password on an actual user model and persist it to the
// database. Otherwise we will parse the error and return the response.
$status = Password::reset(
    $request->only('email', 'password', 'password_confirmation', 'token'),
    function (User $user) use ($request) {
        $user->forceFill([
            'password' => Hash::make($request->password),
            'remember_token' => Str::random(60),
        ]->save();

        event(new PasswordReset($user));
    }
);

// If the password was successfully reset, we will redirect the user back to
// the application's home authenticated view. If there is an error we can
// redirect them back to where they came from with their error message.
return $status == Password::PASSWORD_RESET
    ? redirect()->route('login')->with('status', __($status))
    : back()->withInput($request->only('email'))
    ->withErrors(['email' => __($status)]);
}
}

```

APP

PasswordController

<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;

use Illuminate\Http\RedirectResponse;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Hash;

use Illuminate\Validation\Rules\Password;

class PasswordController extends Controller

```

{
    /**
     * Update the user's password.
     */
    public function update(Request $request): RedirectResponse
    {
        $validated = $request->validateWithBag('updatePassword', [
            'current_password' => ['required', 'current_password'],
            'password' => ['required', Password::defaults(), 'confirmed'],
        ]);
    }
}

```

```

        $request->user()->update([
            'password' => Hash::make($validated['password']),
        ]);

        return back()->with('status', 'password-updated');
    }
}

```

APP

PasswordResetLinkController

<?php

namespace App\Http\Controllers\Auth;

```

use App\Http\Controllers\Controller;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Password;
use Illuminate\View\View;

```

class PasswordResetLinkController extends Controller

```

{
    /**
     * Display the password reset link request view.
     */
    public function create(): View
    {
        return view('auth.forgot-password');
    }

    /**
     * Handle an incoming password reset link request.
     *
     * @throws \Illuminate\Validation\ValidationException
     */
    public function store(Request $request): RedirectResponse
    {
        $request->validate([
            'email' => ['required', 'email'],
        ]);

        // We will send the password reset link to this user. Once we have attempted
        // to send the link, we will examine the response then see the message we
        // need to show to the user. Finally, we'll send out a proper response.
        $status = Password::sendResetLink(
            $request->only('email')
        );

        return $status == Password::RESET_LINK_SENT

```

```

        ? back()->with('status', __($status))
        : back()->withInput($request->only('email'))
        ->withErrors(['email' => __($status)]);
    }
}

```

APP

RegisteredUserController

<?php

```

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use App\Models\User;
use Illuminate\Auth\Events\Registered;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Hash;
use Illuminate\Validation\Rules;
use Illuminate\View\View;

class RegisteredUserController extends Controller
{
    public function create(): View
    {
        return view('auth.register');
    }

    public function store(Request $request): RedirectResponse
    {
        $request->validate([
            'name' => ['required', 'string', 'max:255'],
            'email' => ['required', 'string', 'lowercase', 'email', 'max:255', 'unique:' .
User::class],
            'password' => ['required', 'confirmed', Rules\Password::defaults()],
        ]);

        $user = User::create([
            'name' => $request->name,
            'email' => $request->email,
            'password' => Hash::make($request->password),
            'role' => 'pelanggan',
        ]);

        event(new Registered($user));

        Auth::login($user);

        return redirect(route('customer.dashboard'));
    }
}

```

```
}  
}
```

APP

VerifyEmailController

```
<?php
```

```
namespace App\Http\Controllers\Auth;
```

```
use App\Http\Controllers\Controller;  
use Illuminate\Auth\Events\Verified;  
use Illuminate\Foundation\Auth\EmailVerificationRequest;  
use Illuminate\Contracts\Auth\MustVerifyEmail;  
use Illuminate\Http\RedirectResponse;  
use App\Models\User;
```

```
class VerifyEmailController extends Controller
```

```
{  
    /**  
     * Mark the authenticated user's email address as verified.  
     */  
    public function __invoke(EmailVerificationRequest $request): RedirectResponse  
    {  
        if ($request->user()->hasVerifiedEmail()) {  
            return redirect()->intended(route('dashboard', absolute: false) . '?verified=1');  
        }  
  
        if ($request->user()->markEmailAsVerified()) {  
            event(new Verified($request->user()));  
        }  
  
        return redirect()->intended(route('dashboard', absolute: false) . '?verified=1');  
    }  
}
```

APP

```
<?php
```

```
namespace App\Http\Controllers;
```

```
abstract class Controller
```

```
{  
    //  
}
```

<?php

```
namespace App\Http\Controllers;
```

```
use App\Http\Requests\ProfileUpdateRequest;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Redirect;
use Illuminate\View\View;
```

```
class ProfileController extends Controller
```

```
{
    /**
     * Display the user's profile form.
     */
    public function edit(Request $request): View
    {
        return view('profile.edit', [
            'user' => $request->user(),
        ]);
    }

    /**
     * Update the user's profile information.
     */
    public function update(ProfileUpdateRequest $request): RedirectResponse
    {
        $request->user()->fill($request->validated());

        if ($request->user()->isDirty('email')) {
            $request->user()->email_verified_at = null;
        }

        $request->user()->save();

        return Redirect::route('profile.edit')->with('status', 'profile-updated');
    }

    /**
     * Delete the user's account.
     */
    public function destroy(Request $request): RedirectResponse
    {
        $request->validateWithBag('userDeletion', [
            'password' => ['required', 'current_password'],
        ]);

        $user = $request->user();

        Auth::logout();
```

```

        $user->delete();

        $request->session()->invalidate();
        $request->session()->regenerateToken();

        return Redirect::to('/');
    }
}

```

APP

IsAdmin

<?php

```

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;
use Symfony\Component\HttpFoundation\Response;

class IsAdmin
{
    public function handle(Request $request, Closure $next): Response
    {
        if (auth()->check() && auth()->user()->role === 'admin') {
            return $next($request);
        }

        abort(403, 'Akses ditolak. Anda bukan admin.');
```

APP

LoginRequest

<?php

```

namespace App\Http\Requests\Auth;

use Illuminate\Auth\Events\Lockout;
use Illuminate\Foundation\Http\FormRequest;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\RateLimiter;
use Illuminate\Support\Str;
use Illuminate\Validation\ValidationException;

class LoginRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     */

```

```

public function authorize(): bool
{
    return true;
}

/**
 * Get the validation rules that apply to the request.
 *
 * @return array<string,
\Illuminate\Contracts\Validation\ValidationRule|array<mixed>|string>
 */
public function rules(): array
{
    return [
        'email' => ['required', 'string', 'email'],
        'password' => ['required', 'string'],
    ];
}

/**
 * Attempt to authenticate the request's credentials.
 *
 * @throws \Illuminate\Validation\ValidationException
 */
public function authenticate(): void
{
    $this->ensureIsNotRateLimited();

    if (! Auth::attempt($this->only('email', 'password'), $this->boolean('remember'))) {
        RateLimiter::hit($this->throttleKey());

        throw ValidationException::withMessages([
            'email' => trans('auth.failed'),
        ]);
    }

    RateLimiter::clear($this->throttleKey());
}

/**
 * Ensure the login request is not rate limited.
 *
 * @throws \Illuminate\Validation\ValidationException
 */
public function ensureIsNotRateLimited(): void
{
    if (! RateLimiter::tooManyAttempts($this->throttleKey(), 5)) {
        return;
    }

    event(new Lockout($this));
}

```

```

        $seconds = RateLimiter::availableIn($this->throttleKey());

        throw ValidationException::withMessages([
            'email' => trans('auth.throttle', [
                'seconds' => $seconds,
                'minutes' => ceil($seconds / 60),
            ]),
        ]);
    }

    /**
     * Get the rate limiting throttle key for the request.
     */
    public function throttleKey(): string
    {
        return Str::transliterate(Str::lower($this->string('email')).'|'.$this->ip());
    }
}

```

APP

ProfileUpdateRequest

<?php

```
namespace App\Http\Requests;
```

```
use App\Models\User;
```

```
use Illuminate\Foundation\Http\FormRequest;
```

```
use Illuminate\Validation\Rule;
```

```
class ProfileUpdateRequest extends FormRequest
```

```

{
    /**
     * Get the validation rules that apply to the request.
     *
     * @return array<string,
    \Illuminate\Contracts\Validation\ValidationRule|array<mixed>|string>
     */
    public function rules(): array
    {
        return [
            'name' => ['required', 'string', 'max:255'],
            'email' => [
                'required',
                'string',
                'lowercase',
                'email',
                'max:255',
                Rule::unique(User::class)->ignore($this->user()->id),
            ],
        ],
    }
}

```

```

    ];
}
}

```

APP

FilamentLogoutResponse

<?php

namespace App\Http\Responses; // Pastikan namespace benar

use Filament\Http\Responses\Auth\Contracts\LogoutResponse as
LogoutResponseContract;

use Illuminate\Http\RedirectResponse;

class FilamentLogoutResponse implements LogoutResponseContract

```

{
    public function toResponse($request): RedirectResponse
    {
        // Arahkan ke halaman login Breeze setelah logout dari Filament
        return redirect()->route('login');
    }
}

```

APP

Order

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class Order extends Model

```

{
    use HasFactory;

    protected $fillable = [
        'user_id',
        'kode_pesanan',
        'order_date',
        'status_pesanan',
        'total_amount',
        'catatan_pelanggan',
        'payment_status',
        'delivery_option',
        'paid_at',
        'processed_by_admin_id',
        'catatan_internal',
    ];
}

```

```

];

protected $casts = [
    'order_date' => 'datetime',
    'paid_at' => 'datetime',
];

// Relasi: Satu pesanan dimiliki oleh satu user (pelanggan)
public function customer() // Menggunakan 'customer' agar lebih deskriptif
{
    return $this->belongsTo(User::class, 'user_id');
}

// Relasi: Satu pesanan diproses oleh satu user (admin)
public function processedByAdmin()
{
    return $this->belongsTo(User::class, 'processed_by_admin_id');
}

// Relasi: Satu pesanan memiliki banyak detail pesanan
public function orderDetails()
{
    return $this->hasMany(OrderDetail::class);
}

// Di App\Models\Order.php

public function user()
{
    return $this->belongsTo(User::class);
}

}

```

APP

OrderDetail

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
```

```
class OrderDetail extends Model
```

```
{
```

```
    use HasFactory;
```

```
    protected $fillable = [
        'order_id',

```

```

        'service_id',
        'quantity',
        'harga_saat_pesan',
        'sub_total',
    ];

    // Relasi: Satu detail pesanan milik satu pesanan
    public function order()
    {
        return $this->belongsTo(Order::class);
    }

    // Relasi: Satu detail pesanan merujuk ke satu layanan
    public function service()
    {
        return $this->belongsTo(Service::class);
    }
}

```

APP

Service

<?php

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
```

```
class Service extends Model
{
```

```
    use HasFactory;
```

```
    protected $fillable = [
        'tipe_paket',
        'nama_layanan',
        'harga',
        'satuan',
        'estimasi_durasi',
    ];
```

```
    // Definisikan harga per paket per kg
    public const HARGA_PAKET_KG = [
        'reguler' => 5000,
        'kilat' => 8000,
    ];
```

```
    // Definisikan harga per paket per pcs (CONTOH, sesuaikan jika perlu)
    // Jika harga pcs tidak tergantung paket, Anda bisa buat struktur lain
    public const HARGA_PAKET_PCS = [
        'reguler' => 1500, // Contoh harga pcs reguler
    ];
}
```

```

        'kilat' => 2500, // Contoh harga pcs kilat
    ];

    public const ESTIMASI_DURASI_PAKET = [
        'reguler' => '2-3 Hari',
        'kilat' => '1 Hari',
    ];

    protected static function booted(): void
    {
        static::saving(function ($service) {
            if (!$service->tipe_paket || !$service->satuan) {
                // Jika tipe paket atau satuan belum diisi, jangan proses lebih lanjut
                // untuk menghindari error, biarkan validasi Filament yang menangani
                return;
            }

            $namaLayananBase = "Paket " . ucfirst($service->tipe_paket);
            $estimasi = self::ESTIMASI_DURASI_PAKET[$service->tipe_paket] ?? 'N/A';
            $harga = 0;

            if ($service->satuan === 'kg') {
                $harga = self::HARGA_PAKET_KG[$service->tipe_paket] ?? 0;
                $namaLayananBase .= " (Kg)";
            } elseif ($service->satuan === 'pcs') {
                $harga = self::HARGA_PAKET_PCS[$service->tipe_paket] ?? 0; // Menggunakan
HARGA_PAKET_PCS
                $namaLayananBase .= " (Pcs)";
            }
            // Anda bisa menambahkan logika lain jika ada satuan lain

            $service->nama_layanan = $service->nama_layanan ?: $namaLayananBase; // Isi jika
nama layanan kosong
            $service->harga = $harga;
            $service->estimasi_durasi = $estimasi;
        });
    }

    public function orderDetails()
    {
        return $this->hasMany(OrderDetail::class);
    }
}

```

APP

User
<?php

namespace App\Models;

```

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Filament\Models\Contracts\FilamentUser; // Tambahkan ini
use Filament\Panel; // Tambahkan ini

```

```

class User extends Authenticatable implements FilamentUser // Implementasikan
FilamentUser

```

```

{
    use HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'name',
        'email',
        'password',
        'role', // Tambahkan role
        'phone', // Tambahkan phone
        'address', // Tambahkan address
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * Get the attributes that should be cast.
     *
     * @return array<string, string>
     */
    protected function casts(): array
    {
        return [
            'email_verified_at' => 'datetime',
            'password' => 'hashed',
        ];
    }

    // Implementasi untuk FilamentUser
    public function canAccessPanel(Panels $panels): bool
    {
        // Izinkan akses ke panel jika role adalah 'admin'
    }
}

```

```

        return $this->role === 'admin';
        // Anda bisa menyesuaikan nama panel jika tidak default 'admin'
        // return $this->role === 'admin' && $panel->getId() === 'admin';
    }

    // Relasi: User bisa memiliki banyak orders (sebagai pelanggan)
    public function orders()
    {
        return $this->hasMany(Order::class, 'user_id');
    }

    // Relasi: User (admin) bisa memproses banyak orders
    public function processedOrders()
    {
        return $this->hasMany(Order::class, 'processed_by_admin_id');
    }
}

```

APP

OrderObserver

<?php

namespace App\Observers;

use App\Models\Order;

use Illuminate\Support\Facades\Log;

use Illuminate\Support\Facades\Http;

class OrderObserver

```

{
    /**
     * Handle the Order "created" event.
     */
    public function created(Order $order): void
    {
        //
    }

    /**
     * Handle the Order "updated" event.
     */

```

public function updated(Order \$order)

```

{
    // Muat relasi customer agar tidak null
    $order->load('customer');

    // Debug: Cek apakah customer-nya null atau tidak
    Log::info('Observer dijalankan untuk Order ID: ' . $order->id);
}

```

```

        Log::info('Customer: ', ['customer' => $order->customer]);

        if ($order->isDirty('status_pesanan')) {
            Log::info('Status pesanan berubah menjadi: ' . $order->status_pesanan);

            Http::withHeaders([
                'Authorization' => 'KAcGftNz7notiSjPSAT1',
            ])->post('https://api.fonnte.com/send', [
                'target' => $order->customer->phone,
                'message' => 'Halo, status pesanan Anda sekarang: ' . $order->status_pesanan,
            ]);
        }
    }

    /**
     * Handle the Order "deleted" event.
     */
    public function deleted(Order $order): void
    {
        //
    }

    /**
     * Handle the Order "restored" event.
     */
    public function restored(Order $order): void
    {
        //
    }

    /**
     * Handle the Order "force deleted" event.
     */
    public function forceDeleted(Order $order): void
    {
        //
    }
}

```

APP

AdminPanelProvider

```
<?php
```

```
namespace App\Providers\Filament;
```

```

use Filament\Http\Middleware\Authenticate;
use Filament\Http\Middleware\DisableBladeIconComponents;
use Filament\Http\Middleware\DispatchServingFilamentEvent;
use Filament\Pages;

```

```

use Filament\Panel;
use Filament\PanelProvider;
use Filament\Support\Colors\Color;
use Filament\Widgets;
use Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse;
use Illuminate\Cookie\Middleware\EncryptCookies;
use Illuminate\Foundation\Http\Middleware\VerifyCsrfToken;
use Illuminate\Routing\Middleware\SubstituteBindings;
use Illuminate\Session\Middleware\AuthenticateSession;
use Illuminate\Session\Middleware\StartSession;
use Illuminate\View\Middleware\ShareErrorsFromSession;
use App\Filament\Pages\LaporanPesanan;
use Filament\Navigation\MenuItem; // Pastikan MenuItem di-import

class AdminPanelProvider extends PanelProvider
{
    public function panel(Panels $panels): Panels
    {
        return $panels
            ->default()
            ->id('admin')
            ->path('admin')
            ->login() // Mengaktifkan halaman login default Filament
            // ->logoutUrl(fn(): string => route('logout')) // Ini tidak lagi diperlukan jika kita
            override MenuItem atau menggunakan LogoutResponse binding
            ->colors([
                'primary' => Color::Amber,
            ])
            ->authGuard('web')
            ->discoverResources(in: app_path('Filament/Resources'), for:
'App\Filament\Resources')
            ->discoverPages(in: app_path('Filament/Pages'), for: 'App\Filament\Pages')
            ->pages([
                Pages\Dashboard::class,
                LaporanPesanan::class,
            ])
            ->discoverWidgets(in: app_path('Filament/Widgets'), for: 'App\Filament\Widgets')
            ->widgets([
                // Widgets\AccountWidget::class,
                // Widgets\FilamentInfoWidget::class,
            ])
            ->middleware([
                EncryptCookies::class,
                AddQueuedCookiesToResponse::class,
                StartSession::class,
                AuthenticateSession::class,
                ShareErrorsFromSession::class,
                VerifyCsrfToken::class,
                SubstituteBindings::class,
                DisableBladeIconComponents::class,
                DispatchServingFilamentEvent::class,
            ])
    }
}

```

```

->authMiddleware([
    Authenticate::class,
])
->userMenuItems([
    // Anda bisa menambahkan item menu lain di sini jika perlu, contoh:
    // MenuItem::make()
    //   ->label('Profil Saya (Breeze)')
    //   ->url(fn (): string => route('profile.edit'))
    //   ->icon('heroicon-o-user-circle'),
    MenuItem::make()
        ->label('Profil Saya') // Contoh item lain
        ->url(fn(): string => route('profile.edit')) // Mengarah ke profile Breeze
        ->icon('heroicon-o-user-circle'),

    // Mengganti item logout default untuk mengubah labelnya
    'logout' => MenuItem::make()
        ->label('Sign Out') // Ganti label di sini
        ->icon('heroicon-o-arrow-left-on-rectangle')
    // Biarkan Filament menangani action logout-nya.
    // Action default Filament akan melakukan logout dan kemudian
    // response-nya akan di-handle oleh FilamentLogoutResponse yang sudah kita
    buat,
    // yang akan mengarahkannya ke route('login') Breeze.
]);
}
}

```

APP

AppServiceProvider

<?php

namespace App\Providers;

```

use Illuminate\Support\ServiceProvider;
use Filament\Http\Responses\Auth\Contracts\LogoutResponse as
LogoutResponseContract; // Import Kontrak
use App\Http\Responses\FilamentLogoutResponse; // Import Implementasi Anda
use App\Models\Order;
use App\Observers\OrderObserver;
use Illuminate\Support\Facades\Http; // ✅ Benar

```

```

class AppServiceProvider extends ServiceProvider
{
    /**
     * Register any application services.
     */
    public function register(): void
    {

```

// Binding implementasi LogoutResponse kustom kita ke kontrak Filament

```

        $this->app->singleton(
            LogoutResponseContract::class,
            FilamentLogoutResponse::class
        );
    }

    /**
     * Bootstrap any application services.
     */

    public function boot(): void
    {
        Order::observe(OrderObserver::class);
    }
}

```

APP

AppLayout

```

<?php

namespace App\View\Components;

use Illuminate\View\Component;
use Illuminate\View\View;

class AppLayout extends Component
{
    /**
     * Get the view / contents that represents the component.
     */
    public function render(): View
    {
        return view('layouts.app');
    }
}

```

APP

GuestLayout

```

<?php

namespace App\View\Components;

use Illuminate\View\Component;
use Illuminate\View\View;

class GuestLayout extends Component

```

```

{
    /**
     * Get the view / contents that represents the component.
     */
    public function render(): View
    {
        return view('layouts.guest');
    }
}

```

BOOSTRAP

```

packages
<?php return array (
    'anourvalar/eloquent-serialize' =>
        array (
            'aliases' =>
                array (
                    'EloquentSerialize' =>
                        'AnourValar\\EloquentSerialize\\Facades\\EloquentSerializeFacade',
                ),
            ),
        'barryvdh/laravel-dompdf' =>
            array (
                'aliases' =>
                    array (
                        'PDF' => 'Barryvdh\\DomPDF\\Facade\\Pdf',
                        'Pdf' => 'Barryvdh\\DomPDF\\Facade\\Pdf',
                    ),
                'providers' =>
                    array (
                        0 => 'Barryvdh\\DomPDF\\ServiceProvider',
                    ),
            ),
        'blade-ui-kit/blade-heroicons' =>
            array (
                'providers' =>
                    array (
                        0 => 'BladeUI\\Heroicons\\BladeHeroiconsServiceProvider',
                    ),
            ),
        'blade-ui-kit/blade-icons' =>
            array (
                'providers' =>
                    array (
                        0 => 'BladeUI\\Icons\\BladeIconsServiceProvider',
                    ),
            ),
        'filament/actions' =>
            array (
                'providers' =>

```

```

    array (
        0 => 'Filament\\Actions\\ActionsServiceProvider',
    ),
),
'filament/filament' =>
array (
    'providers' =>
    array (
        0 => 'Filament\\FilamentServiceProvider',
    ),
),
'filament/forms' =>
array (
    'providers' =>
    array (
        0 => 'Filament\\Forms\\FormsServiceProvider',
    ),
),
'filament/infolists' =>
array (
    'providers' =>
    array (
        0 => 'Filament\\Infolists\\InfolistsServiceProvider',
    ),
),
'filament/notifications' =>
array (
    'providers' =>
    array (
        0 => 'Filament\\Notifications\\NotificationsServiceProvider',
    ),
),
'filament/support' =>
array (
    'providers' =>
    array (
        0 => 'Filament\\Support\\SupportServiceProvider',
    ),
),
'filament/tables' =>
array (
    'providers' =>
    array (
        0 => 'Filament\\Tables\\TablesServiceProvider',
    ),
),
'filament/widgets' =>
array (
    'providers' =>
    array (
        0 => 'Filament\\Widgets\\WidgetsServiceProvider',
    ),
),

```

```

),
'kirschbaum-development/eloquent-power-joins' =>
array (
    'providers' =>
        array (
            0 => 'Kirschbaum\\PowerJoins\\PowerJoinsServiceProvider',
        ),
    ),
'laravel/breeze' =>
array (
    'providers' =>
        array (
            0 => 'Laravel\\Breeze\\BreezeServiceProvider',
        ),
    ),
'laravel/pail' =>
array (
    'providers' =>
        array (
            0 => 'Laravel\\Pail\\PailServiceProvider',
        ),
    ),
'laravel/sail' =>
array (
    'providers' =>
        array (
            0 => 'Laravel\\Sail\\SailServiceProvider',
        ),
    ),
'laravel/tinker' =>
array (
    'providers' =>
        array (
            0 => 'Laravel\\Tinker\\TinkerServiceProvider',
        ),
    ),
'livewire/livewire' =>
array (
    'aliases' =>
        array (
            'Livewire' => 'Livewire\\Livewire',
        ),
    'providers' =>
        array (
            0 => 'Livewire\\LivewireServiceProvider',
        ),
    ),
'nesbot/carbon' =>
array (
    'providers' =>
        array (
            0 => 'Carbon\\Laravel\\ServiceProvider',

```

```

    ),
    ),
    'nunomaduro/collision' =>
    array (
        'providers' =>
        array (
            0 => 'NunoMaduro\\Collision\\Adapters\\Laravel\\CollisionServiceProvider',
        ),
    ),
    'nunomaduro/termwind' =>
    array (
        'providers' =>
        array (
            0 => 'Termwind\\Laravel\\TermwindServiceProvider',
        ),
    ),
    'ryangjchandler/blade-capture-directive' =>
    array (
        'aliases' =>
        array (
            'BladeCaptureDirective' =>
            'RyanChandler\\BladeCaptureDirective\\Facades\\BladeCaptureDirective',
        ),
        'providers' =>
        array (
            0 => 'RyanChandler\\BladeCaptureDirective\\BladeCaptureDirectiveServiceProvider',
        ),
    ),
);

```

BOOSTRAP

services

```

<?php return array (
    'providers' =>
    array (
        0 => 'Illuminate\\Auth\\AuthServiceProvider',
        1 => 'Illuminate\\Broadcasting\\BroadcastServiceProvider',
        2 => 'Illuminate\\Bus\\BusServiceProvider',
        3 => 'Illuminate\\Cache\\CacheServiceProvider',
        4 => 'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
        5 => 'Illuminate\\Concurrency\\ConcurrencyServiceProvider',
        6 => 'Illuminate\\Cookie\\CookieServiceProvider',
        7 => 'Illuminate\\Database\\DatabaseServiceProvider',
        8 => 'Illuminate\\Encryption\\EncryptionServiceProvider',
        9 => 'Illuminate\\Filesystem\\FilesystemServiceProvider',
        10 => 'Illuminate\\Foundation\\Providers\\FoundationServiceProvider',
        11 => 'Illuminate\\Hashing\\HashServiceProvider',
        12 => 'Illuminate\\Mail\\MailServiceProvider',
        13 => 'Illuminate\\Notifications\\NotificationServiceProvider',
        14 => 'Illuminate\\Pagination\\PaginationServiceProvider',
    ),
);

```

```

15 => 'Illuminate\Auth\Passwords\PasswordResetServiceProvider',
16 => 'Illuminate\Pipeline\PipelineServiceProvider',
17 => 'Illuminate\Queue\QueueServiceProvider',
18 => 'Illuminate\Redis\RedisServiceProvider',
19 => 'Illuminate\Session\SessionServiceProvider',
20 => 'Illuminate\Translation\TranslationServiceProvider',
21 => 'Illuminate\Validation\ValidationServiceProvider',
22 => 'Illuminate\View\ViewServiceProvider',
23 => 'Barryvdh\DomPDF\ServiceProvider',
24 => 'BladeUI\Heroicons\BladeHeroiconsServiceProvider',
25 => 'BladeUI\Icons\BladeIconsServiceProvider',
26 => 'Filament\Actions\ActionsServiceProvider',
27 => 'Filament\FilamentServiceProvider',
28 => 'Filament\Forms\FormsServiceProvider',
29 => 'Filament\Infolists\InfolistsServiceProvider',
30 => 'Filament\Notifications\NotificationsServiceProvider',
31 => 'Filament\Support\SupportServiceProvider',
32 => 'Filament\Tables\TablesServiceProvider',
33 => 'Filament\Widgets\WidgetsServiceProvider',
34 => 'Kirschbaum\PowerJoins\PowerJoinsServiceProvider',
35 => 'Laravel\Breeze\BreezeServiceProvider',
36 => 'Laravel\Pail\PailServiceProvider',
37 => 'Laravel\Sail\SailServiceProvider',
38 => 'Laravel\Tinker\TinkerServiceProvider',
39 => 'Livewire\LivewireServiceProvider',
40 => 'Carbon\Laravel\ServiceProvider',
41 => 'NunoMaduro\Collision\Adapters\Laravel\CollisionServiceProvider',
42 => 'Termwind\Laravel\TermwindServiceProvider',
43 => 'RyanChandler\BladeCaptureDirective\BladeCaptureDirectiveServiceProvider',
44 => 'App\Providers\AppServiceProvider',
45 => 'App\Providers\Filament\AdminPanelProvider',
),
'eager' =>
array (
    0 => 'Illuminate\Auth\AuthServiceProvider',
    1 => 'Illuminate\Cookie\CookieServiceProvider',
    2 => 'Illuminate\Database\DatabaseServiceProvider',
    3 => 'Illuminate\Encryption\EncryptionServiceProvider',
    4 => 'Illuminate\Filesystem\FilesystemServiceProvider',
    5 => 'Illuminate\Foundation\Providers\FoundationServiceProvider',
    6 => 'Illuminate\Notifications\NotificationServiceProvider',
    7 => 'Illuminate\Pagination\PaginationServiceProvider',
    8 => 'Illuminate\Session\SessionServiceProvider',
    9 => 'Illuminate\View\ViewServiceProvider',
    10 => 'Barryvdh\DomPDF\ServiceProvider',
    11 => 'BladeUI\Heroicons\BladeHeroiconsServiceProvider',
    12 => 'BladeUI\Icons\BladeIconsServiceProvider',
    13 => 'Filament\Actions\ActionsServiceProvider',
    14 => 'Filament\FilamentServiceProvider',
    15 => 'Filament\Forms\FormsServiceProvider',
    16 => 'Filament\Infolists\InfolistsServiceProvider',
    17 => 'Filament\Notifications\NotificationsServiceProvider',

```

```

18 => 'Filament\\Support\\SupportServiceProvider',
19 => 'Filament\\Tables\\TablesServiceProvider',
20 => 'Filament\\Widgets\\WidgetsServiceProvider',
21 => 'Kirschbaum\\PowerJoins\\PowerJoinsServiceProvider',
22 => 'Laravel\\Pail\\PailServiceProvider',
23 => 'Livewire\\LivewireServiceProvider',
24 => 'Carbon\\Laravel\\ServiceProvider',
25 => 'NunoMaduro\\Collision\\Adapters\\Laravel\\CollisionServiceProvider',
26 => 'Termwind\\Laravel\\TermwindServiceProvider',
27 => 'RyanChandler\\BladeCaptureDirective\\BladeCaptureDirectiveServiceProvider',
28 => 'App\\Providers\\AppServiceProvider',
29 => 'App\\Providers\\Filament\\AdminPanelProvider',
),
'deferred' =>
array (
    'Illuminate\\Broadcasting\\BroadcastManager' =>
    'Illuminate\\Broadcasting\\BroadcastServiceProvider',
    'Illuminate\\Contracts\\Broadcasting\\Factory' =>
    'Illuminate\\Broadcasting\\BroadcastServiceProvider',
    'Illuminate\\Contracts\\Broadcasting\\Broadcaster' =>
    'Illuminate\\Broadcasting\\BroadcastServiceProvider',
    'Illuminate\\Bus\\Dispatcher' => 'Illuminate\\Bus\\BusServiceProvider',
    'Illuminate\\Contracts\\Bus\\Dispatcher' => 'Illuminate\\Bus\\BusServiceProvider',
    'Illuminate\\Contracts\\Bus\\QueueingDispatcher' =>
    'Illuminate\\Bus\\BusServiceProvider',
    'Illuminate\\Bus\\BatchRepository' => 'Illuminate\\Bus\\BusServiceProvider',
    'Illuminate\\Bus\\DatabaseBatchRepository' => 'Illuminate\\Bus\\BusServiceProvider',
    'cache' => 'Illuminate\\Cache\\CacheServiceProvider',
    'cache.store' => 'Illuminate\\Cache\\CacheServiceProvider',
    'cache.psr6' => 'Illuminate\\Cache\\CacheServiceProvider',
    'memcached.connector' => 'Illuminate\\Cache\\CacheServiceProvider',
    'Illuminate\\Cache\\RateLimiter' => 'Illuminate\\Cache\\CacheServiceProvider',
    'Illuminate\\Foundation\\Console\\AboutCommand' =>
    'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Cache\\Console\\ClearCommand' =>
    'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Cache\\Console\\ForgetCommand' =>
    'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Foundation\\Console\\ClearCompiledCommand' =>
    'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Auth\\Console\\ClearResetsCommand' =>
    'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Foundation\\Console\\ConfigCacheCommand' =>
    'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Foundation\\Console\\ConfigClearCommand' =>
    'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Foundation\\Console\\ConfigShowCommand' =>
    'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Database\\Console\\DbCommand' =>
    'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Database\\Console\\MonitorCommand' =>
    'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',

```

```

'Illuminate\\Database\\Console\\PruneCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Database\\Console\\ShowCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Database\\Console\\TableCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Database\\Console\\WipeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\DownCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\EnvironmentCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\EnvironmentDecryptCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\EnvironmentEncryptCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\EventCacheCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\EventClearCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\EventListCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Concurrency\\Console\\InvokeSerializedClosureCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\KeyGenerateCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\OptimizeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\OptimizeClearCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\PackageDiscoverCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Cache\\Console\\PruneStaleTagsCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Queue\\Console\\ClearCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Queue\\Console\\ListFailedCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Queue\\Console\\FlushFailedCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Queue\\Console\\ForgetFailedCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Queue\\Console\\ListenCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Queue\\Console\\MonitorCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Queue\\Console\\PruneBatchesCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Queue\\Console\\PruneFailedJobsCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Queue\\Console\\RestartCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',

```

```

'Illuminate\\Queue\\Console\\RetryCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Queue\\Console\\RetryBatchCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Queue\\Console\\WorkCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\RouteCacheCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\RouteClearCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\RouteListCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Database\\Console\\DumpCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Database\\Console\\Seeds\\SeedCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Console\\Scheduling\\ScheduleFinishCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Console\\Scheduling\\ScheduleListCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Console\\Scheduling\\ScheduleRunCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Console\\Scheduling\\ScheduleClearCacheCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Console\\Scheduling\\ScheduleTestCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Console\\Scheduling\\ScheduleWorkCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Console\\Scheduling\\ScheduleInterruptCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Database\\Console\\ShowModelCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\StorageLinkCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\StorageUnlinkCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\UpCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\ViewCacheCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\ViewClearCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\ApiInstallCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\BroadcastingInstallCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Cache\\Console\\CacheTableCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\CastMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\ChannelListCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',

```

```

'Illuminate\\Foundation\\Console\\ChannelMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\ClassMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\ComponentMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\ConfigPublishCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\ConsoleMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Routing\\Console\\ControllerMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\DocsCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\EnumMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\EventGenerateCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\EventMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\ExceptionMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Database\\Console\\Factories\\FactoryMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\InterfaceMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\JobMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\JobMiddlewareMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\LangPublishCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\ListenerMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\MailMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Routing\\Console\\MiddlewareMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\ModelMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\NotificationMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Notifications\\Console\\NotificationTableCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\ObserverMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\PolicyMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\ProviderMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Queue\\Console\\FailedTableCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',

```

```

'Illuminate\\Queue\\Console\\TableCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Queue\\Console\\BatchesTableCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\RequestMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\ResourceMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\RuleMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\ScopeMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Database\\Console\\Seeds\\SeederMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Session\\Console\\SessionTableCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\ServeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\StubPublishCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\TestMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\TraitMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\VendorPublishCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Foundation\\Console\\ViewMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'migrator' => 'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'migration.repository' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'migration.creator' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Database\\Migrations\\Migrator' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Database\\Console\\Migrations\\MigrateCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Database\\Console\\Migrations\\FreshCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Database\\Console\\Migrations\\InstallCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Database\\Console\\Migrations\\RefreshCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Database\\Console\\Migrations\\ResetCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Database\\Console\\Migrations\\RollbackCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Database\\Console\\Migrations\\StatusCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Database\\Console\\Migrations\\MigrateMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'composer' => 'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',

```

```

'Illuminate\\Concurrency\\ConcurrencyManager' =>
'Illuminate\\Concurrency\\ConcurrencyServiceProvider',
'hash' => 'Illuminate\\Hashing\\HashServiceProvider',
'hash.driver' => 'Illuminate\\Hashing\\HashServiceProvider',
'mail.manager' => 'Illuminate\\Mail\\MailServiceProvider',
'mailer' => 'Illuminate\\Mail\\MailServiceProvider',
'Illuminate\\Mail\\Markdown' => 'Illuminate\\Mail\\MailServiceProvider',
'auth.password' => 'Illuminate\\Auth\\Passwords\\PasswordResetServiceProvider',
'auth.password.broker' =>
'Illuminate\\Auth\\Passwords\\PasswordResetServiceProvider',
'Illuminate\\Contracts\\Pipeline\\Hub' =>
'Illuminate\\Pipeline\\PipelineServiceProvider',
'pipeline' => 'Illuminate\\Pipeline\\PipelineServiceProvider',
'queue' => 'Illuminate\\Queue\\QueueServiceProvider',
'queue.connection' => 'Illuminate\\Queue\\QueueServiceProvider',
'queue.failer' => 'Illuminate\\Queue\\QueueServiceProvider',
'queue.listener' => 'Illuminate\\Queue\\QueueServiceProvider',
'queue.worker' => 'Illuminate\\Queue\\QueueServiceProvider',
'redis' => 'Illuminate\\Redis\\RedisServiceProvider',
'redis.connection' => 'Illuminate\\Redis\\RedisServiceProvider',
'translator' => 'Illuminate\\Translation\\TranslationServiceProvider',
'translation.loader' => 'Illuminate\\Translation\\TranslationServiceProvider',
'validator' => 'Illuminate\\Validation\\ValidationServiceProvider',
'validation.presence' => 'Illuminate\\Validation\\ValidationServiceProvider',
'Illuminate\\Contracts\\Validation\\UncompromisedVerifier' =>
'Illuminate\\Validation\\ValidationServiceProvider',
'Laravel\\Breeze\\Console\\InstallCommand' =>
'Laravel\\Breeze\\BreezeServiceProvider',
'Laravel\\Sail\\Console\\InstallCommand' => 'Laravel\\Sail\\SailServiceProvider',
'Laravel\\Sail\\Console\\PublishCommand' => 'Laravel\\Sail\\SailServiceProvider',
'command.tinker' => 'Laravel\\Tinker\\TinkerServiceProvider',
),
'when' =>
array (
'Illuminate\\Broadcasting\\BroadcastServiceProvider' =>
array (
),
'Illuminate\\Bus\\BusServiceProvider' =>
array (
),
'Illuminate\\Cache\\CacheServiceProvider' =>
array (
),
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider' =>
array (
),
'Illuminate\\Concurrency\\ConcurrencyServiceProvider' =>
array (
),
'Illuminate\\Hashing\\HashServiceProvider' =>
array (
),

```

```

'Illuminate\\Mail\\MailServiceProvider' =>
array (
),
'Illuminate\\Auth\\Passwords\\PasswordResetServiceProvider' =>
array (
),
'Illuminate\\Pipeline\\PipelineServiceProvider' =>
array (
),
'Illuminate\\Queue\\QueueServiceProvider' =>
array (
),
'Illuminate\\Redis\\RedisServiceProvider' =>
array (
),
'Illuminate\\Translation\\TranslationServiceProvider' =>
array (
),
'Illuminate\\Validation\\ValidationServiceProvider' =>
array (
),
'Laravel\\Breeze\\BreezeServiceProvider' =>
array (
),
'Laravel\\Sail\\SailServiceProvider' =>
array (
),
'Laravel\\Tinker\\TinkerServiceProvider' =>
array (
),
),
);

```

BOOSTRAP

App

```
<?php
```

```

use Illuminate\Foundation\Application;
use Illuminate\Foundation\Configuration\Exceptions;
use Illuminate\Foundation\Configuration\Middleware;

return Application::configure(basePath: dirname(__DIR__))
    ->withRouting(
        web: __DIR__.'/../routes/web.php',
        commands: __DIR__.'/../routes/console.php',
        health: '/up',
    )
    ->withMiddleware(function (Middleware $middleware) {
        //
    });

```

```

    })
    ->withExceptions(function (Exceptions $exceptions) {
        //
    })->create();

```

BOOSTRAP

providers

```
<?php
```

```

return [
    App\Providers\AppServiceProvider::class,
    App\Providers\Filament\AdminPanelProvider::class,
];

```

BOOSTRAP

app

```
<?php
```

```
return [
```

```

    /*
    |-----
    | Application Name
    |-----
    |
    | This value is the name of your application, which will be used when the
    | framework needs to place the application's name in a notification or
    | other UI elements where an application name needs to be displayed.
    |
    */

```

```
'name' => env('APP_NAME', 'Laravel'),
```

```

    /*
    |-----
    | Application Environment
    |-----
    |
    | This value determines the "environment" your application is currently
    | running in. This may determine how you prefer to configure various
    | services the application utilizes. Set this in your ".env" file.
    |
    */

```

```
'env' => env('APP_ENV', 'production'),
```

```

    /*
    |-----

```

```

| Application Debug Mode
|-----
|
| When your application is in debug mode, detailed error messages with
| stack traces will be shown on every error that occurs within your
| application. If disabled, a simple generic error page is shown.
|
| */
'debug' => (bool) env('APP_DEBUG', false),

/*
|-----
| Application URL
|-----
|
| This URL is used by the console to properly generate URLs when using
| the Artisan command line tool. You should set this to the root of
| the application so that it's available within Artisan commands.
|
| */
'url' => env('APP_URL', 'http://localhost'),

/*
|-----
| Application Timezone
|-----
|
| Here you may specify the default timezone for your application, which
| will be used by the PHP date and date-time functions. The timezone
| is set to "UTC" by default as it is suitable for most use cases.
|
| */
'timezone' => 'UTC',

/*
|-----
| Application Locale Configuration
|-----
|
| The application locale determines the default locale that will be used
| by Laravel's translation / localization methods. This option can be
| set to any locale for which you plan to have translation strings.
|
| */
'locale' => env('APP_LOCALE', 'en'),

'fallback_locale' => env('APP_FALLBACK_LOCALE', 'en'),

```

```

'faker_locale' => env('APP_FAKER_LOCALE', 'en_US'),

/*
|-----
| Encryption Key
|-----
|
| This key is utilized by Laravel's encryption services and should be set
| to a random, 32 character string to ensure that all encrypted values
| are secure. You should do this prior to deploying the application.
|
*/

'cipher' => 'AES-256-CBC',

'key' => env('APP_KEY'),

'previous_keys' => [
    ...array_filter(
        explode(',', env('APP_PREVIOUS_KEYS', ''))
    ),
],

/*
|-----
| Maintenance Mode Driver
|-----
|
| These configuration options determine the driver used to determine and
| manage Laravel's "maintenance mode" status. The "cache" driver will
| allow maintenance mode to be controlled across multiple machines.
|
| Supported drivers: "file", "cache"
|
*/

'maintenance' => [
    'driver' => env('APP_MAINTENANCE_DRIVER', 'file'),
    'store' => env('APP_MAINTENANCE_STORE', 'database'),
],
];

```

BOOSTRAP

Auth

<?php

```

return [

    /*
    |-----
    | Authentication Defaults
    |-----
    |
    | This option defines the default authentication "guard" and password
    | reset "broker" for your application. You may change these values
    | as required, but they're a perfect start for most applications.
    |
    */

    'defaults' => [
        'guard' => env('AUTH_GUARD', 'web'),
        'passwords' => env('AUTH_PASSWORD_BROKER', 'users'),
    ],

    /*
    |-----
    | Authentication Guards
    |-----
    |
    | Next, you may define every authentication guard for your application.
    | Of course, a great default configuration has been defined for you
    | which utilizes session storage plus the Eloquent user provider.
    |
    | All authentication guards have a user provider, which defines how the
    | users are actually retrieved out of your database or other storage
    | system used by the application. Typically, Eloquent is utilized.
    |
    | Supported: "session"
    |
    */

    'guards' => [
        'web' => [
            'driver' => 'session',
            'provider' => 'users',
        ],
    ],

    /*
    |-----
    | User Providers
    |-----
    |
    | All authentication guards have a user provider, which defines how the
    | users are actually retrieved out of your database or other storage
    | system used by the application. Typically, Eloquent is utilized.
    |

```

```
| If you have multiple user tables or models you may configure multiple  
| providers to represent the model / table. These providers may then  
| be assigned to any extra authentication guards you have defined.
```

```
| Supported: "database", "eloquent"
```

```
|  
*/
```

```
'providers' => [  
    'users' => [  
        'driver' => 'eloquent',  
        'model' => env('AUTH_MODEL', App\Models\User::class),  
    ],
```

```
    // 'users' => [  
    //     'driver' => 'database',  
    //     'table' => 'users',  
    // ],  
],
```

```
/*
```

```
|-----  
| Resetting Passwords  
|-----
```

```
| These configuration options specify the behavior of Laravel's password  
| reset functionality, including the table utilized for token storage  
| and the user provider that is invoked to actually retrieve users.
```

```
| The expiry time is the number of minutes that each reset token will be  
| considered valid. This security feature keeps tokens short-lived so  
| they have less time to be guessed. You may change this as needed.
```

```
| The throttle setting is the number of seconds a user must wait before  
| generating more password reset tokens. This prevents the user from  
| quickly generating a very large amount of password reset tokens.
```

```
|  
*/
```

```
'passwords' => [  
    'users' => [  
        'provider' => 'users',  
        'table' => env('AUTH_PASSWORD_RESET_TOKEN_TABLE', 'password_reset_tokens'),  
        'expire' => 60,  
        'throttle' => 60,  
    ],  
],
```

```
/*
```

```
|-----  
| Password Confirmation Timeout  
|-----
```

```
|
| Here you may define the amount of seconds before a password confirmation
| window expires and users are asked to re-enter their password via the
| confirmation screen. By default, the timeout lasts for three hours.
|
| */
|
| 'password_timeout' => env('AUTH_PASSWORD_TIMEOUT', 10800),
|
|];
```

BOOSTRAP

Cache

```
<?php
```

```
use Illuminate\Support\Str;
```

```
return [
```

```
/*
```

```
|-----
| Default Cache Store
|-----
```

```
|
| This option controls the default cache store that will be used by the
| framework. This connection is utilized if another isn't explicitly
| specified when running a cache operation inside the application.
|
```

```
*/
```

```
'default' => env('CACHE_STORE', 'database'),
```

```
/*
```

```
|-----
| Cache Stores
|-----
```

```
|
| Here you may define all of the cache "stores" for your application as
| well as their drivers. You may even define multiple stores for the
| same cache driver to group types of items stored in your caches.
|
```

```
| Supported drivers: "array", "database", "file", "memcached",
|                   "redis", "dynamodb", "octane", "null"
|
```

```
*/
```

```
'stores' => [
```

```
    'array' => [
```

```

    'driver' => 'array',
    'serialize' => false,
],

'database' => [
    'driver' => 'database',
    'connection' => env('DB_CACHE_CONNECTION'),
    'table' => env('DB_CACHE_TABLE', 'cache'),
    'lock_connection' => env('DB_CACHE_LOCK_CONNECTION'),
    'lock_table' => env('DB_CACHE_LOCK_TABLE'),
],

'file' => [
    'driver' => 'file',
    'path' => storage_path('framework/cache/data'),
    'lock_path' => storage_path('framework/cache/data'),
],

'memcached' => [
    'driver' => 'memcached',
    'persistent_id' => env('MEMCACHED_PERSISTENT_ID'),
    'sas!' => [
        env('MEMCACHED_USERNAME'),
        env('MEMCACHED_PASSWORD'),
    ],
    'options' => [
        // Memcached::OPT_CONNECT_TIMEOUT => 2000,
    ],
    'servers' => [
        [
            'host' => env('MEMCACHED_HOST', '127.0.0.1'),
            'port' => env('MEMCACHED_PORT', 11211),
            'weight' => 100,
        ],
    ],
],

'redis' => [
    'driver' => 'redis',
    'connection' => env('REDIS_CACHE_CONNECTION', 'cache'),
    'lock_connection' => env('REDIS_CACHE_LOCK_CONNECTION', 'default'),
],

'dynamodb' => [
    'driver' => 'dynamodb',
    'key' => env('AWS_ACCESS_KEY_ID'),
    'secret' => env('AWS_SECRET_ACCESS_KEY'),
    'region' => env('AWS_DEFAULT_REGION', 'us-east-1'),
    'table' => env('DYNAMODB_CACHE_TABLE', 'cache'),
    'endpoint' => env('DYNAMODB_ENDPOINT'),
],

```

```

        'octane' => [
            'driver' => 'octane',
        ],
    ],

    /*
    |-----
    | Cache Key Prefix
    |-----
    |
    | When utilizing the APC, database, memcached, Redis, and DynamoDB cache
    | stores, there might be other applications using the same cache. For
    | that reason, you may prefix every cache key to avoid collisions.
    |
    */

    'prefix' => env('CACHE_PREFIX', Str::slug(env('APP_NAME', 'laravel'), '_').'_cache_'),
];

```

database

<?php

use Illuminate\Support\Str;

return [

```

    /*
    |-----
    | Default Database Connection Name
    |-----
    |
    | Here you may specify which of the database connections below you wish
    | to use as your default connection for database operations. This is
    | the connection which will be utilized unless another connection
    | is explicitly specified when you execute a query / statement.
    |
    */

```

```

    'default' => env('DB_CONNECTION', 'sqlite'),

```

```

    /*
    |-----
    | Database Connections
    |-----
    |
    | Below are all of the database connections defined for your application.
    | An example configuration is provided for each database system which
    | is supported by Laravel. You're free to add / remove connections.

```

|
*/

```
'connections' => [  
  
    'sqlite' => [  
        'driver' => 'sqlite',  
        'url' => env('DB_URL'),  
        'database' => env('DB_DATABASE', database_path('database.sqlite')),  
        'prefix' => '',  
        'foreign_key_constraints' => env('DB_FOREIGN_KEYS', true),  
        'busy_timeout' => null,  
        'journal_mode' => null,  
        'synchronous' => null,  
    ],  
  
    'mysql' => [  
        'driver' => 'mysql',  
        'url' => env('DB_URL'),  
        'host' => env('DB_HOST', '127.0.0.1'),  
        'port' => env('DB_PORT', '3306'),  
        'database' => env('DB_DATABASE', 'laravel'),  
        'username' => env('DB_USERNAME', 'root'),  
        'password' => env('DB_PASSWORD', ''),  
        'unix_socket' => env('DB_SOCKET', ''),  
        'charset' => env('DB_CHARSET', 'utf8mb4'),  
        'collation' => env('DB_COLLATION', 'utf8mb4_unicode_ci'),  
        'prefix' => '',  
        'prefix_indexes' => true,  
        'strict' => false,  
        'engine' => null,  
        'options' => extension_loaded('pdo_mysql') ? array_filter([  
            PDO::MYSQL_ATTR_SSL_CA => env('MYSQL_ATTR_SSL_CA'),  
        ]) : [],  
    ],  
  
    'mariadb' => [  
        'driver' => 'mariadb',  
        'url' => env('DB_URL'),  
        'host' => env('DB_HOST', '127.0.0.1'),  
        'port' => env('DB_PORT', '3306'),  
        'database' => env('DB_DATABASE', 'laravel'),  
        'username' => env('DB_USERNAME', 'root'),  
        'password' => env('DB_PASSWORD', ''),  
        'unix_socket' => env('DB_SOCKET', ''),  
        'charset' => env('DB_CHARSET', 'utf8mb4'),  
        'collation' => env('DB_COLLATION', 'utf8mb4_unicode_ci'),  
        'prefix' => '',  
        'prefix_indexes' => true,  
        'strict' => true,  
        'engine' => null,  
        'options' => extension_loaded('pdo_mysql') ? array_filter([
```

```

        PDO::MYSQL_ATTR_SSL_CA => env('MYSQL_ATTR_SSL_CA'),
    ]) : [],
],

'pgsql' => [
    'driver' => 'pgsql',
    'url' => env('DB_URL'),
    'host' => env('DB_HOST', '127.0.0.1'),
    'port' => env('DB_PORT', '5432'),
    'database' => env('DB_DATABASE', 'laravel'),
    'username' => env('DB_USERNAME', 'root'),
    'password' => env('DB_PASSWORD', ''),
    'charset' => env('DB_CHARSET', 'utf8'),
    'prefix' => '',
    'prefix_indexes' => true,
    'search_path' => 'public',
    'sslmode' => 'prefer',
],

'sqlsrv' => [
    'driver' => 'sqlsrv',
    'url' => env('DB_URL'),
    'host' => env('DB_HOST', 'localhost'),
    'port' => env('DB_PORT', '1433'),
    'database' => env('DB_DATABASE', 'laravel'),
    'username' => env('DB_USERNAME', 'root'),
    'password' => env('DB_PASSWORD', ''),
    'charset' => env('DB_CHARSET', 'utf8'),
    'prefix' => '',
    'prefix_indexes' => true,
    // 'encrypt' => env('DB_ENCRYPT', 'yes'),
    // 'trust_server_certificate' => env('DB_TRUST_SERVER_CERTIFICATE', 'false'),
],

],

/*
|-----
| Migration Repository Table
|-----
|
| This table keeps track of all the migrations that have already run for
| your application. Using this information, we can determine which of
| the migrations on disk haven't actually been run on the database.
|
*/

'migrations' => [
    'table' => 'migrations',
    'update_date_on_publish' => true,
],

```

```

/*
|-----
| Redis Databases
|-----
|
| Redis is an open source, fast, and advanced key-value store that also
| provides a richer body of commands than a typical key-value system
| such as Memcached. You may define your connection settings here.
|
*/

'redis' => [

    'client' => env('REDIS_CLIENT', 'phpredis'),

    'options' => [
        'cluster' => env('REDIS_CLUSTER', 'redis'),
        'prefix' => env('REDIS_PREFIX', Str::slug(env('APP_NAME', 'laravel'), '_').
'_database_'),
        'persistent' => env('REDIS_PERSISTENT', false),
    ],

    'default' => [
        'url' => env('REDIS_URL'),
        'host' => env('REDIS_HOST', '127.0.0.1'),
        'username' => env('REDIS_USERNAME'),
        'password' => env('REDIS_PASSWORD'),
        'port' => env('REDIS_PORT', '6379'),
        'database' => env('REDIS_DB', '0'),
    ],

    'cache' => [
        'url' => env('REDIS_URL'),
        'host' => env('REDIS_HOST', '127.0.0.1'),
        'username' => env('REDIS_USERNAME'),
        'password' => env('REDIS_PASSWORD'),
        'port' => env('REDIS_PORT', '6379'),
        'database' => env('REDIS_CACHE_DB', '1'),
    ],

],

];

```

BOOSTRAP

Dompdf

<?php

return [

```

/*
|-----
| Settings
|-----
|
| Set some default values. It is possible to add all defines that can be set
| in dompdf_config.inc.php. You can also override the entire config file.
|
*/
'show_warnings' => false, // Throw an Exception on warnings from dompdf

'public_path' => null, // Override the public path if needed

/*
 * Dejavu Sans font is missing glyphs for converted entities, turn it off if you need to
show € and £.
 */
'convert_entities' => true,

'options' => [
  /**
   * The location of the DOMPDP font directory
   *
   * The location of the directory where DOMPDP will store fonts and font metrics
   * Note: This directory must exist and be writable by the webserver process.
   * *Please note the trailing slash.*
   *
   * Notes regarding fonts:
   * Additional .afm font metrics can be added by executing load_font.php from
command line.
   *
   * Only the original "Base 14 fonts" are present on all pdf viewers. Additional fonts
must
   * be embedded in the pdf file or the PDF may not display correctly. This can
significantly
   * increase file size unless font subsetting is enabled. Before embedding a font please
   * review your rights under the font license.
   *
   * Any font specification in the source HTML is translated to the closest font available
   * in the font directory.
   *
   * The pdf standard "Base 14 fonts" are:
   * Courier, Courier-Bold, Courier-BoldOblique, Courier-Oblique,
   * Helvetica, Helvetica-Bold, Helvetica-BoldOblique, Helvetica-Oblique,
   * Times-Roman, Times-Bold, Times-BoldItalic, Times-Italic,
   * Symbol, ZapfDingbats.
   */
  'font_dir' => storage_path('fonts'), // advised by dompdf
(https://github.com/dompdf/dompdf/pull/782)

  /**

```

```

* The location of the DOMPdf font cache directory
*
* This directory contains the cached font metrics for the fonts used by DOMPdf.
* This directory can be the same as DOMPdf_FONT_DIR
*
* Note: This directory must exist and be writable by the webserver process.
*/
'font_cache' => storage_path('fonts'),

/**
* The location of a temporary directory.
*
* The directory specified must be writeable by the webserver process.
* The temporary directory is required to download remote images and when
* using the PDFlib back end.
*/
'temp_dir' => sys_get_temp_dir(),

/**
* ===== IMPORTANT =====
*
* dompdf's "chroot": Prevents dompdf from accessing system files or other
* files on the webserver. All local files opened by dompdf must be in a
* subdirectory of this directory. DO NOT set it to '/' since this could
* allow an attacker to use dompdf to read any files on the server. This
* should be an absolute path.
* This is only checked on command line call by dompdf.php, but not by
* direct class use like:
* $dompdf = new DOMPdf(); $dompdf->load_html($htmldata); $dompdf->render();
$pdffdata = $dompdf->output();
*/
'chroot' => realpath(base_path()),

/**
* Protocol whitelist
*
* Protocols and PHP wrappers allowed in URIs, and the validation rules
* that determine if a resource may be loaded. Full support is not guaranteed
* for the protocols/wrappers specified
* by this array.
*
* @var array
*/
'allowed_protocols' => [
    'data://' => ['rules' => []],
    'file://' => ['rules' => []],
    'http://' => ['rules' => []],
    'https://' => ['rules' => []],
],

/**
* Operational artifact (log files, temporary files) path validation

```

```

*/
'artifactPathValidation' => null,

/**
 * @var string
 */
'log_output_file' => null,

/**
 * Whether to enable font subsetting or not.
 */
'enable_font_subsetting' => false,

/**
 * The PDF rendering backend to use
 *
 * Valid settings are 'PDFLib', 'CPDF' (the bundled R&OS PDF class), 'GD' and
 * 'auto'. 'auto' will look for PDFLib and use it if found, or if not it will
 * fall back on CPDF. 'GD' renders PDFs to graphic files.
 * {@link * Canvas_Factory} ultimately determines which rendering class to
 * instantiate based on this setting.
 *
 * Both PDFLib & CPDF rendering backends provide sufficient rendering
 * capabilities for dompdf, however additional features (e.g. object,
 * image and font support, etc.) differ between backends. Please see
 * {@link PDFLib_Adapter} for more information on the PDFLib backend
 * and {@link CPDF_Adapter} and lib/class.pdf.php for more information
 * on CPDF. Also see the documentation for each backend at the links
 * below.
 *
 * The GD rendering backend is a little different than PDFLib and
 * CPDF. Several features of CPDF and PDFLib are not supported or do
 * not make any sense when creating image files. For example,
 * multiple pages are not supported, nor are PDF 'objects'. Have a
 * look at {@link GD_Adapter} for more information. GD support is
 * experimental, so use it at your own risk.
 *
 * @link http://www.pdflib.com
 * @link http://www.ros.co.nz/pdf
 * @link http://www.php.net/image
 */
'pdf_backend' => 'CPDF',

/**
 * html target media view which should be rendered into pdf.
 * List of types and parsing rules for future extensions:
 * http://www.w3.org/TR/REC-html40/types.html
 * screen, tty, tv, projection, handheld, print, braille, aural, all
 * Note: aural is deprecated in CSS 2.1 because it is replaced by speech in CSS 3.
 * Note, even though the generated pdf file is intended for print output,
 * the desired content might be different (e.g. screen or projection view of html file).
 * Therefore allow specification of content here.

```

```

*/
'default_media_type' => 'screen',

/**
 * The default paper size.
 *
 * North America standard is "letter"; other countries generally "a4"
 *
 * @see CPDF_Adapter::PAPER_SIZES for valid sizes ('letter', 'legal', 'A4', etc.)
 */
'default_paper_size' => 'a4',

/**
 * The default paper orientation.
 *
 * The orientation of the page (portrait or landscape).
 *
 * @var string
 */
'default_paper_orientation' => 'portrait',

/**
 * The default font family
 *
 * Used if no suitable fonts can be found. This must exist in the font folder.
 *
 * @var string
 */
'default_font' => 'serif',

/**
 * Image DPI setting
 *
 * This setting determines the default DPI setting for images and fonts. The
 * DPI may be overridden for inline images by explicitly setting the
 * image's width & height style attributes (i.e. if the image's native
 * width is 600 pixels and you specify the image's width as 72 points,
 * the image will have a DPI of 600 in the rendered PDF. The DPI of
 * background images can not be overridden and is controlled entirely
 * via this parameter.
 *
 * For the purposes of DOMPDF, pixels per inch (PPI) = dots per inch (DPI).
 * If a size in html is given as px (or without unit as image size),
 * this tells the corresponding size in pt.
 * This adjusts the relative sizes to be similar to the rendering of the
 * html page in a reference browser.
 *
 * In pdf, always 1 pt = 1/72 inch
 *
 * Rendering resolution of various browsers in px per inch:
 * Windows Firefox and Internet Explorer:
 * SystemControl->Display properties->FontResolution: Default:96, largefonts:120,

```

custom:?

```
* Linux Firefox:
* about:config *resolution: Default:96
* (xorg screen dimension in mm and Desktop font dpi settings are ignored)
*
* Take care about extra font/image zoom factor of browser.
*
* In images, <img> size in pixel attribute, img css style, are overriding
* the real image dimension in px for rendering.
*
* @var int
*/
'dpi' => 96,

/**
* Enable embedded PHP
*
* If this setting is set to true then DOMPDF will automatically evaluate embedded PHP
contained
* within <script type="text/php"> ... </script> tags.
*
* ===== IMPORTANT ===== Enabling this for documents you do not trust (e.g. arbitrary
remote html pages)
* is a security risk.
* Embedded scripts are run with the same level of system access available to dompdf.
* Set this option to false (recommended) if you wish to process untrusted documents.
* This setting may increase the risk of system exploit.
* Do not change this settings without understanding the consequences.
* Additional documentation is available on the dompdf wiki at:
* https://github.com/dompdf/dompdf/wiki
*
* @var bool
*/
'enable_php' => false,

/**
* Enable inline JavaScript
*
* If this setting is set to true then DOMPDF will automatically insert JavaScript code
contained
* within <script type="text/javascript"> ... </script> tags as written into the PDF.
* NOTE: This is PDF-based JavaScript to be executed by the PDF viewer,
* not browser-based JavaScript executed by Dompdf.
*
* @var bool
*/
'enable_javascript' => true,

/**
* Enable remote file access
*
* If this setting is set to true, DOMPDF will access remote sites for
```

```

    * images and CSS files as required.
    *
    * ==== IMPORTANT ====
    * This can be a security risk, in particular in combination with isPhpEnabled and
    * allowing remote html code to be passed to $dompdf = new DOMPDF(); $dompdf->load_html(...);
    * This allows anonymous users to download legally doubtful internet content which
on
    * tracing back appears to being downloaded by your server, or allows malicious php
code
    * in remote html pages to be executed by your server with your account privileges.
    *
    * This setting may increase the risk of system exploit. Do not change
    * this settings without understanding the consequences. Additional
    * documentation is available on the dompdf wiki at:
    * https://github.com/dompdf/dompdf/wiki
    *
    * @var bool
    */
    'enable_remote' => false,

    /**
    * List of allowed remote hosts
    *
    * Each value of the array must be a valid hostname.
    *
    * This will be used to filter which resources can be loaded in combination with
    * isRemoteEnabled. If enable_remote is FALSE, then this will have no effect.
    *
    * Leave to NULL to allow any remote host.
    *
    * @var array|null
    */
    'allowed_remote_hosts' => null,

    /**
    * A ratio applied to the fonts height to be more like browsers' line height
    */
    'font_height_ratio' => 1.1,

    /**
    * Use the HTML5 Lib parser
    *
    * @deprecated This feature is now always on in dompdf 2.x
    *
    * @var bool
    */
    'enable_html5_parser' => true,
],
];

```

filesystems

<?php

return [

```

/*
|-----|
| Default Filesystem Disk
|-----|
|
| Here you may specify the default filesystem disk that should be used
| by the framework. The "local" disk, as well as a variety of cloud
| based disks are available to your application for file storage.
|
*/

'default' => env('FILESYSTEM_DISK', 'local'),

/*
|-----|
| Filesystem Disks
|-----|
|
| Below you may configure as many filesystem disks as necessary, and you
| may even configure multiple disks for the same driver. Examples for
| most supported storage drivers are configured here for reference.
|
| Supported drivers: "local", "ftp", "sftp", "s3"
|
*/

'disks' => [

    'local' => [
        'driver' => 'local',
        'root' => storage_path('app/private'),
        'serve' => true,
        'throw' => false,
        'report' => false,
    ],

    'public' => [
        'driver' => 'local',
        'root' => storage_path('app/public'),
        'url' => env('APP_URL').'/storage',
        'visibility' => 'public',
        'throw' => false,
        'report' => false,
    ],

```

```

    ],

    's3' => [
        'driver' => 's3',
        'key' => env('AWS_ACCESS_KEY_ID'),
        'secret' => env('AWS_SECRET_ACCESS_KEY'),
        'region' => env('AWS_DEFAULT_REGION'),
        'bucket' => env('AWS_BUCKET'),
        'url' => env('AWS_URL'),
        'endpoint' => env('AWS_ENDPOINT'),
        'use_path_style_endpoint' => env('AWS_USE_PATH_STYLE_ENDPOINT', false),
        'throw' => false,
        'report' => false,
    ],

    ],

    /*
    |-----
    | Symbolic Links
    |-----
    |
    | Here you may configure the symbolic links that will be created when the
    | `storage:link` Artisan command is executed. The array keys should be
    | the locations of the links and the values should be their targets.
    |
    */

    'links' => [
        public_path('storage') => storage_path('app/public'),
    ],

];

```

BOOSTRAP

Logging

```
<?php
```

```

use Monolog\Handler\NullHandler;
use Monolog\Handler\StreamHandler;
use Monolog\Handler\SyslogUdpHandler;
use Monolog\Processor\PsrLogMessageProcessor;

```

```
return [
```

```

    /*
    |-----
    | Default Log Channel
    |-----
    |

```

```

| This option defines the default log channel that is utilized to write
| messages to your logs. The value provided here should match one of
| the channels present in the list of "channels" configured below.
|
*/

'default' => env('LOG_CHANNEL', 'stack'),

/*
|-----
| Deprecations Log Channel
|-----
|
| This option controls the log channel that should be used to log warnings
| regarding deprecated PHP and library features. This allows you to get
| your application ready for upcoming major versions of dependencies.
|
*/

'deprecations' => [
    'channel' => env('LOG_DEPRECATED_CHANNEL', 'null'),
    'trace' => env('LOG_DEPRECATED_TRACE', false),
],

/*
|-----
| Log Channels
|-----
|
| Here you may configure the log channels for your application. Laravel
| utilizes the Monolog PHP logging library, which includes a variety
| of powerful log handlers and formatters that you're free to use.
|
| Available drivers: "single", "daily", "slack", "syslog",
|                   "errorlog", "monolog", "custom", "stack"
|
*/

'channels' => [

    'stack' => [
        'driver' => 'stack',
        'channels' => explode(',', env('LOG_STACK', 'single')),
        'ignore_exceptions' => false,
    ],

    'single' => [
        'driver' => 'single',
        'path' => storage_path('logs/laravel.log'),
        'level' => env('LOG_LEVEL', 'debug'),
        'replace_placeholders' => true,
    ],
],

```

```

'daily' => [
    'driver' => 'daily',
    'path' => storage_path('logs/laravel.log'),
    'level' => env('LOG_LEVEL', 'debug'),
    'days' => env('LOG_DAILY_DAYS', 14),
    'replace_placeholders' => true,
],

'slack' => [
    'driver' => 'slack',
    'url' => env('LOG_SLACK_WEBHOOK_URL'),
    'username' => env('LOG_SLACK_USERNAME', 'Laravel Log'),
    'emoji' => env('LOG_SLACK_EMOJI', ':boom:'),
    'level' => env('LOG_LEVEL', 'critical'),
    'replace_placeholders' => true,
],

'papertrail' => [
    'driver' => 'monolog',
    'level' => env('LOG_LEVEL', 'debug'),
    'handler' => env('LOG_PAPERTRAIL_HANDLER', SyslogUdpHandler::class),
    'handler_with' => [
        'host' => env('PAPERTRAIL_URL'),
        'port' => env('PAPERTRAIL_PORT'),
        'connectionString' => 'tls://'.env('PAPERTRAIL_URL').':'.env('PAPERTRAIL_PORT'),
    ],
    'processors' => [PsrLogMessageProcessor::class],
],

'stderr' => [
    'driver' => 'monolog',
    'level' => env('LOG_LEVEL', 'debug'),
    'handler' => StreamHandler::class,
    'handler_with' => [
        'stream' => 'php://stderr',
    ],
    'formatter' => env('LOG_STDERR_FORMATTER'),
    'processors' => [PsrLogMessageProcessor::class],
],

'syslog' => [
    'driver' => 'syslog',
    'level' => env('LOG_LEVEL', 'debug'),
    'facility' => env('LOG_SYSLOG_FACILITY', LOG_USER),
    'replace_placeholders' => true,
],

'errorlog' => [
    'driver' => 'errorlog',
    'level' => env('LOG_LEVEL', 'debug'),
    'replace_placeholders' => true,

```

```

    ],

    'null' => [
        'driver' => 'monolog',
        'handler' => NullHandler::class,
    ],

    'emergency' => [
        'path' => storage_path('logs/laravel.log'),
    ],
],

];

```

BOOSTRAP

mail

<?php

return [

```

/*
|-----
| Default Mailer
|-----
|
| This option controls the default mailer that is used to send all email
| messages unless another mailer is explicitly specified when sending
| the message. All additional mailers can be configured within the
| "mailers" array. Examples of each type of mailer are provided.
|
*/

'default' => env('MAIL_MAILER', 'log'),

/*
|-----
| Mailer Configurations
|-----
|
| Here you may configure all of the mailers used by your application plus
| their respective settings. Several examples have been configured for
| you and you are free to add your own as your application requires.
|
| Laravel supports a variety of mail "transport" drivers that can be used
| when delivering an email. You may specify which one you're using for
| your mailers below. You may also add additional mailers if needed.
|
| Supported: "smtp", "sendmail", "mailgun", "ses", "ses-v2",

```

```

|      "postmark", "resend", "log", "array",
|      "failover", "roundrobin"
|
*/

'mailers' => [

  'smtp' => [
    'transport' => 'smtp',
    'scheme' => env('MAIL_SCHEME'),
    'url' => env('MAIL_URL'),
    'host' => env('MAIL_HOST', '127.0.0.1'),
    'port' => env('MAIL_PORT', 2525),
    'username' => env('MAIL_USERNAME'),
    'password' => env('MAIL_PASSWORD'),
    'timeout' => null,
    'local_domain' => env('MAIL_EHLO_DOMAIN', parse_url(env('APP_URL',
'http://localhost'), PHP_URL_HOST)),
  ],

  'ses' => [
    'transport' => 'ses',
  ],

  'postmark' => [
    'transport' => 'postmark',
    // 'message_stream_id' => env('POSTMARK_MESSAGE_STREAM_ID'),
    // 'client' => [
    //   'timeout' => 5,
    // ],
  ],

  'resend' => [
    'transport' => 'resend',
  ],

  'sendmail' => [
    'transport' => 'sendmail',
    'path' => env('MAIL_SENDMAIL_PATH', '/usr/sbin/sendmail -bs -i'),
  ],

  'log' => [
    'transport' => 'log',
    'channel' => env('MAIL_LOG_CHANNEL'),
  ],

  'array' => [
    'transport' => 'array',
  ],

  'failover' => [
    'transport' => 'failover',

```

```

        'mailers' => [
            'smtp',
            'log',
        ],
        'retry_after' => 60,
    ],

    'roundrobin' => [
        'transport' => 'roundrobin',
        'mailers' => [
            'ses',
            'postmark',
        ],
        'retry_after' => 60,
    ],
],

/*
|-----|
| Global "From" Address |
|-----|
|
| You may wish for all emails sent by your application to be sent from
| the same address. Here you may specify a name and address that is
| used globally for all emails that are sent by your application.
|
*/

'from' => [
    'address' => env('MAIL_FROM_ADDRESS', 'hello@example.com'),
    'name' => env('MAIL_FROM_NAME', 'Example'),
],
];

```

BOOSTRAP

Queue

<?php

return [

```

/*
|-----|
| Default Queue Connection Name |
|-----|
|
| Laravel's queue supports a variety of backends via a single, unified
| API, giving you convenient access to each backend using identical
| syntax for each. The default queue connection is defined below.

```

```

|
*/

'default' => env('QUEUE_CONNECTION', 'database'),

/*
|-----
| Queue Connections
|-----
|
| Here you may configure the connection options for every queue backend
| used by your application. An example configuration is provided for
| each backend supported by Laravel. You're also free to add more.
|
| Drivers: "sync", "database", "beanstalkd", "sqs", "redis", "null"
|
*/

'connections' => [

    'sync' => [
        'driver' => 'sync',
    ],

    'database' => [
        'driver' => 'database',
        'connection' => env('DB_QUEUE_CONNECTION'),
        'table' => env('DB_QUEUE_TABLE', 'jobs'),
        'queue' => env('DB_QUEUE', 'default'),
        'retry_after' => (int) env('DB_QUEUE_RETRY_AFTER', 90),
        'after_commit' => false,
    ],

    'beanstalkd' => [
        'driver' => 'beanstalkd',
        'host' => env('BEANSTALKD_QUEUE_HOST', 'localhost'),
        'queue' => env('BEANSTALKD_QUEUE', 'default'),
        'retry_after' => (int) env('BEANSTALKD_QUEUE_RETRY_AFTER', 90),
        'block_for' => 0,
        'after_commit' => false,
    ],

    'sqs' => [
        'driver' => 'sqs',
        'key' => env('AWS_ACCESS_KEY_ID'),
        'secret' => env('AWS_SECRET_ACCESS_KEY'),
        'prefix' => env('SQS_PREFIX', 'https://sqs.us-east-1.amazonaws.com/your-account-
id'),
        'queue' => env('SQS_QUEUE', 'default'),
        'suffix' => env('SQS_SUFFIX'),
        'region' => env('AWS_DEFAULT_REGION', 'us-east-1'),
        'after_commit' => false,

```

```

    ],

    'redis' => [
        'driver' => 'redis',
        'connection' => env('REDIS_QUEUE_CONNECTION', 'default'),
        'queue' => env('REDIS_QUEUE', 'default'),
        'retry_after' => (int) env('REDIS_QUEUE_RETRY_AFTER', 90),
        'block_for' => null,
        'after_commit' => false,
    ],

],

/*
|-----
| Job Batching
|-----
|
| The following options configure the database and table that store job
| batching information. These options can be updated to any database
| connection and table which has been defined by your application.
|
*/

'batching' => [
    'database' => env('DB_CONNECTION', 'sqlite'),
    'table' => 'job_batches',
],

/*
|-----
| Failed Queue Jobs
|-----
|
| These options configure the behavior of failed queue job logging so you
| can control how and where failed jobs are stored. Laravel ships with
| support for storing failed jobs in a simple file or in a database.
|
| Supported drivers: "database-uuids", "dynamodb", "file", "null"
|
*/

'failed' => [
    'driver' => env('QUEUE_FAILED_DRIVER', 'database-uuids'),
    'database' => env('DB_CONNECTION', 'sqlite'),
    'table' => 'failed_jobs',
],

];

```

BOOSTRAP

Services

<?php

return [

```
/*
|-----
| Third Party Services
|-----
|
| This file is for storing the credentials for third party services such
| as Mailgun, Postmark, AWS and more. This file provides the de facto
| location for this type of information, allowing packages to have
| a conventional file to locate the various service credentials.
|
*/

'postmark' => [
    'token' => env('POSTMARK_TOKEN'),
],

'ses' => [
    'key' => env('AWS_ACCESS_KEY_ID'),
    'secret' => env('AWS_SECRET_ACCESS_KEY'),
    'region' => env('AWS_DEFAULT_REGION', 'us-east-1'),
],

'resend' => [
    'key' => env('RESEND_KEY'),
],

'slack' => [
    'notifications' => [
        'bot_user_oauth_token' => env('SLACK_BOT_USER_OAUTH_TOKEN'),
        'channel' => env('SLACK_BOT_USER_DEFAULT_CHANNEL'),
    ],
],
];
```

BOOSTRAP

session

<?php

use Illuminate\Support\Str;

return [

```

/*
|-----|
| Default Session Driver |
|-----|
|
| This option determines the default session driver that is utilized for
| incoming requests. Laravel supports a variety of storage options to
| persist session data. Database storage is a great default choice.
|
| Supported: "file", "cookie", "database", "memcached",
|           "redis", "dynamodb", "array"
|
*/

'driver' => env('SESSION_DRIVER', 'database'),

/*
|-----|
| Session Lifetime |
|-----|
|
| Here you may specify the number of minutes that you wish the session
| to be allowed to remain idle before it expires. If you want them
| to expire immediately when the browser is closed then you may
| indicate that via the expire_on_close configuration option.
|
*/

'lifetime' => (int) env('SESSION_LIFETIME', 120),

'expire_on_close' => env('SESSION_EXPIRE_ON_CLOSE', false),

/*
|-----|
| Session Encryption |
|-----|
|
| This option allows you to easily specify that all of your session data
| should be encrypted before it's stored. All encryption is performed
| automatically by Laravel and you may use the session like normal.
|
*/

'encrypt' => env('SESSION_ENCRYPT', false),

/*
|-----|
| Session File Location |
|-----|
|
| When utilizing the "file" session driver, the session files are placed

```

| on disk. The default storage location is defined here; however, you
| are free to provide another location where they should be stored.

|
*/

'files' => storage_path('framework/sessions'),

/*

|-----
| **Session Database Connection**
|-----

|
| When using the "database" or "redis" session drivers, you may specify a
| connection that should be used to manage these sessions. This should
| correspond to a connection in your database configuration options.

|
*/

'connection' => env('SESSION_CONNECTION'),

/*

|-----
| **Session Database Table**
|-----

|
| When using the "database" session driver, you may specify the table to
| be used to store sessions. Of course, a sensible default is defined
| for you; however, you're welcome to change this to another table.

|
*/

'table' => env('SESSION_TABLE', 'sessions'),

/*

|-----
| **Session Cache Store**
|-----

|
| When using one of the framework's cache driven session backends, you may
| define the cache store which should be used to store the session data
| between requests. This must match one of your defined cache stores.

|
| Affects: "apc", "dynamodb", "memcached", "redis"

|
*/

'store' => env('SESSION_STORE'),

/*

|-----
| **Session Sweeping Lottery**
|-----

```

|
| Some session drivers must manually sweep their storage location to get
| rid of old sessions from storage. Here are the chances that it will
| happen on a given request. By default, the odds are 2 out of 100.
|
| */
|
| 'lottery' => [2, 100],
|
| /*
| -----
| Session Cookie Name
| -----
|
| Here you may change the name of the session cookie that is created by
| the framework. Typically, you should not need to change this value
| since doing so does not grant a meaningful security improvement.
|
| */
|
| 'cookie' => env(
|     'SESSION_COOKIE',
|     Str::slug(env('APP_NAME', 'laravel'), '_').'_session'
| ),
|
| /*
| -----
| Session Cookie Path
| -----
|
| The session cookie path determines the path for which the cookie will
| be regarded as available. Typically, this will be the root path of
| your application, but you're free to change this when necessary.
|
| */
|
| 'path' => env('SESSION_PATH', '/'),
|
| /*
| -----
| Session Cookie Domain
| -----
|
| This value determines the domain and subdomains the session cookie is
| available to. By default, the cookie will be available to the root
| domain and all subdomains. Typically, this shouldn't be changed.
|
| */
|
| 'domain' => env('SESSION_DOMAIN'),
|
| /*

```

```

|-----|
| HTTPS Only Cookies
|-----|
|
| By setting this option to true, session cookies will only be sent back
| to the server if the browser has a HTTPS connection. This will keep
| the cookie from being sent to you when it can't be done securely.
|
| */
|
| 'secure' => env('SESSION_SECURE_COOKIE'),
|
| /*
|-----|
| HTTP Access Only
|-----|
|
| Setting this value to true will prevent JavaScript from accessing the
| value of the cookie and the cookie will only be accessible through
| the HTTP protocol. It's unlikely you should disable this option.
|
| */
|
| 'http_only' => env('SESSION_HTTP_ONLY', true),
|
| /*
|-----|
| Same-Site Cookies
|-----|
|
| This option determines how your cookies behave when cross-site requests
| take place, and can be used to mitigate CSRF attacks. By default, we
| will set this value to "lax" to permit secure cross-site requests.
|
| See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-
Cookie#samesitesamesite-value
|
| Supported: "lax", "strict", "none", null
|
| */
|
| 'same_site' => env('SESSION_SAME_SITE', 'lax'),
|
| /*
|-----|
| Partitioned Cookies
|-----|
|
| Setting this value to true will tie the cookie to the top-level site for
| a cross-site context. Partitioned cookies are accepted by the browser
| when flagged "secure" and the Same-Site attribute is set to "none".
|

```

```

*/

'partitioned' => env('SESSION_PARTITIONED_COOKIE', false),
];

```

DATABASE

UserFactory.php

```

<?php

namespace Database\Factories;

use Illuminate\Database\Eloquent\Factories\Factory;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Str;

/**
 * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\User>
 */
class UserFactory extends Factory
{
    /**
     * The current password being used by the factory.
     */
    protected static ?string $password;

    /**
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */
    public function definition(): array
    {
        return [
            'name' => fake()->name(),
            'email' => fake()->unique()->safeEmail(),
            'email_verified_at' => now(),
            'password' => static::$password ??= Hash::make('password'),
            'remember_token' => Str::random(10),
        ];
    }

    /**
     * Indicate that the model's email address should be unverified.
     */
    public function unverified(): static
    {
        return $this->state(fn (array $attributes) => [
            'email_verified_at' => null,

```

```

    });
}
}

```

DATABASE

0001_01_01_000000_create_users_table

<?php

```

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

```

return new class extends Migration

```

{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });

        Schema::create('password_reset_tokens', function (Blueprint $table) {
            $table->string('email')->primary();
            $table->string('token');
            $table->timestamp('created_at')->nullable();
        });

        Schema::create('sessions', function (Blueprint $table) {
            $table->string('id')->primary();
            $table->foreignId('user_id')->nullable()->index();
            $table->string('ip_address', 45)->nullable();
            $table->text('user_agent')->nullable();
            $table->longText('payload');
            $table->integer('last_activity')->index();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void

```

```

{
    Schema::dropIfExists('users');
    Schema::dropIfExists('password_reset_tokens');
    Schema::dropIfExists('sessions');
}
};

```

DATABASE

0001_01_01_000001_create_cache_table

<?php

```

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

```

return new class extends Migration

```

{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('cache', function (Blueprint $table) {
            $table->string('key')->primary();
            $table->mediumText('value');
            $table->integer('expiration');
        });

        Schema::create('cache_locks', function (Blueprint $table) {
            $table->string('key')->primary();
            $table->string('owner');
            $table->integer('expiration');
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('cache');
        Schema::dropIfExists('cache_locks');
    }
};

```

0001_01_01_000002_create_jobs_table

<?php

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
```

return new class extends Migration

```
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('jobs', function (Blueprint $table) {
            $table->id();
            $table->string('queue')->index();
            $table->longText('payload');
            $table->unsignedTinyInteger('attempts');
            $table->unsignedInteger('reserved_at')->nullable();
            $table->unsignedInteger('available_at');
            $table->unsignedInteger('created_at');
        });

        Schema::create('job_batches', function (Blueprint $table) {
            $table->string('id')->primary();
            $table->string('name');
            $table->integer('total_jobs');
            $table->integer('pending_jobs');
            $table->integer('failed_jobs');
            $table->longText('failed_job_ids');
            $table->mediumText('options')->nullable();
            $table->integer('cancelled_at')->nullable();
            $table->integer('created_at');
            $table->integer('finished_at')->nullable();
        });

        Schema::create('failed_jobs', function (Blueprint $table) {
            $table->id();
            $table->string('uuid')->unique();
            $table->text('connection');
            $table->text('queue');
            $table->longText('payload');
            $table->longText('exception');
            $table->timestamp('failed_at')->useCurrent();
        });
    }
}

/**
```

```

    * Reverse the migrations.
    */
    public function down(): void
    {
        Schema::dropIfExists('jobs');
        Schema::dropIfExists('job_batches');
        Schema::dropIfExists('failed_jobs');
    }
};

```

DATABASE

2025_05_19_171957_add_extra_fields_to_users_table

<?php

```

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration {
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::table('users', function (Blueprint $table) {
            // Pastikan kolom ditambahkan setelah kolom tertentu jika diinginkan,
            // contoh: $table->string('role')->after('email_verified_at')->default('pelanggan');
            $table->string('role')->default('pelanggan'); // Contoh: 'admin', 'pelanggan'
            $table->string('phone')->nullable();
            $table->text('address')->nullable();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::table('users', function (Blueprint $table) {
            $table->dropColumn(['role', 'phone', 'address']);
        });
    }
};

```

DATABASE

2025_05_19_172015_create_services_table

```
<?php
```

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
```

```
return new class extends Migration {
```

```
    /**
```

```
     * Run the migrations.
```

```
     */
```

```
    public function up(): void
```

```
    {
```

```
        Schema::create('services', function (Blueprint $table) {
```

```
            $table->id(); // service_id
```

```
            $table->string('nama_layanan');
```

```
            $table->decimal('harga', 10, 2); // 10 digit total, 2 digit di belakang koma
```

```
            $table->string('satuan'); // misal: 'kg', 'pcs', 'm2'
```

```
            $table->string('estimasi_durasi')->nullable(); // misal: '1 Hari', '3 Jam', dll.
```

```
            $table->timestamps(); // created_at dan updated_at
```

```
        });
```

```
    }
```

```
    /**
```

```
     * Reverse the migrations.
```

```
     */
```

```
    public function down(): void
```

```
    {
```

```
        Schema::dropIfExists('services');
```

```
    }
```

```
};
```

DATABASE

2025_05_19_172242_create_orders_table

```
<?php
```

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
```

```
return new class extends Migration {
```

```
    /**
```

```
     * Run the migrations.
```

```
     */
```

```
    public function up(): void
```

```
    {
```

```
        Schema::create('orders', function (Blueprint $table) {
```

```
            $table->id(); // order_id
```

```
            $table->foreignId('user_id')->constrained('users')->onDelete('cascade');
```

```

$table->string('kode_pesanan')->unique();
$table->timestamp('order_date')->useCurrent();
$table->string('status_pesanan')->default('baru');
// status_pesanan: 'baru', 'diproses', 'siap diambil', 'selesai', 'dibatalkan'

$table->decimal('total_amount', 12, 2)->nullable();
$table->text('catatan_pelanggan')->nullable();

// Kolom terkait pembayaran langsung
$table->string('payment_status')->default('belum_bayar');
// payment_status: 'belum_bayar', 'lunas'
$table->timestamp('paid_at')->nullable(); // Waktu ketika pembayaran dikonfirmasi
$table->foreignId('processed_by_admin_id')->nullable()->constrained('users')-
>onDelete('set null'); // Admin yang memproses pembayaran/penyerahan
$table->text('catatan_internal')->nullable(); // Catatan dari admin terkait
order/pembayaran

    $table->timestamps();
    });
}

/**
 * Reverse the migrations.
 */
public function down(): void
{
    Schema::dropIfExists('orders');
}
};

```

DATABASE

2025_05_19_172243_create_order_details_table

```
// database/migrations/xxxx_xx_xx_XXXXXX_create_order_details_table.php
<?php
```

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
```

```
return new class extends Migration {
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('order_details', function (Blueprint $table) {
            $table->id(); // order_detail_id
            $table->foreignId('order_id')->constrained('orders')->onDelete('cascade');
        });
    }
};

```

```

        $table->foreignId('service_id')->constrained('services')->onDelete('restrict'); // atau
onDelete('cascade') jika lebih sesuai
        $table->decimal('quantity', 8, 2); // Kuantitas, bisa desimal untuk kg
        $table->decimal('harga_saat_pesan', 10, 2); // Harga layanan saat item ini dipesan
        $table->decimal('sub_total', 12, 2);
        $table->timestamps();
    });
}

/**
 * Reverse the migrations.
 */
public function down(): void
{
    Schema::dropIfExists('order_details');
}
};

```

DATABASE

2025_05_19_200047_modify_services_table_for_packages

<?php

```

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration {
    public function up(): void
    {
        Schema::table('services', function (Blueprint $table) {
            // Tambahkan kolom tipe_paket
            $table->string('tipe_paket')->after('id')->comment('Contoh: reguler, kilat');
            // Buat nama_layanan nullable atau beri default jika ingin digenerate otomatis
            $table->string('nama_layanan')->nullable()->change();
            // Kita tetap simpan harga, tapi akan diisi otomatis nanti
            // Jika estimasi durasi juga tergantung paket, bisa disesuaikan di sini
        });
    }

    public function down(): void
    {
        Schema::table('services', function (Blueprint $table) {
            $table->dropColumn('tipe_paket');
            $table->string('nama_layanan')->nullable(false)->change(); // Kembalikan seperti
semula jika perlu
        });
    }
};

```

DATABASE

2025_05_19_211958_add_indexes_to_orders_and_users_tables

<?php

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
```

```
return new class extends Migration {
    public function up(): void
    {
        Schema::table('orders', function (Blueprint $table) {
            $table->index('created_at');
            $table->index('payment_status');
            $table->index('status_pesanan');
            // user_id sudah seharusnya terindeks jika menggunakan foreignId()->constrained()
            // tapi bisa ditambahkan secara eksplisit jika ragu: $table->index('user_id');
            // processed_by_admin_id juga bisa dipertimbangkan jika sering difilter
        });

        Schema::table('users', function (Blueprint $table) {
            $table->index('role'); // Untuk filter pelanggan/admin
        });

        Schema::table('order_details', function (Blueprint $table) {
            // order_id dan service_id sudah seharusnya terindeks karena foreign key
            // $table->index('order_id');
            // $table->index('service_id');
        });
    }

    public function down(): void
    {
        Schema::table('orders', function (Blueprint $table) {
            $table->dropIndex(['created_at']); // Nama indeks default adalah
nama_tabel_nama_kolom_index
            $table->dropIndex(['payment_status']);
            $table->dropIndex(['status_pesanan']);
            // $table->dropIndex(['user_id']);
        });

        Schema::table('users', function (Blueprint $table) {
            $table->dropIndex(['role']);
        });

        // Tidak perlu dropIndex untuk order_details jika hanya foreign key default
    }
};
```

DATABASE

DatabaseSeeder

```
<?php
```

```
namespace Database\Seeders;
```

```
// use Illuminate\Database\Console\Seeds\WithoutModelEvents;
```

```
use Illuminate\Database\Seeder;
```

```
class DatabaseSeeder extends Seeder
```

```
{
```

```
    /**
```

```
     * Seed the application's database.
```

```
     */
```

```
    public function run(): void
```

```
    {
```

```
        // Panggil seeder Anda di sini dalam urutan yang benar
```

```
        $this->call([
```

```
            UserSeeder::class,
```

```
            ServiceSeeder::class,
```

```
            OrderSeeder::class, // OrderSeeder dijalankan terakhir karena butuh User dan
```

```
Service
```

```
        ]);
```

```
    }
```

```
}
```

DATABASE

OrderSeeder

```
<?php
```

```
namespace Database\Seeders;
```

```
use Illuminate\Database\Console\Seeds\WithoutModelEvents;
```

```
use Illuminate\Database\Seeder;
```

```
use App\Models\Order;
```

```
use App\Models\OrderDetail;
```

```
use App\Models\User;
```

```
use App\Models\Service;
```

```
use Illuminate\Support\Str;
```

```
use Carbon\Carbon; // Untuk manipulasi tanggal
```

```
class OrderSeeder extends Seeder
```

```
{
```

```

/**
 * Run the database seeds.
 */
public function run(): void
{
    // Ambil ID pelanggan dan admin yang ada
    $customerIds = User::where('role', 'pelanggan')->pluck('id')->toArray();
    $adminIds = User::where('role', 'admin')->pluck('id')->toArray();
    $serviceIds = Service::pluck('id')->toArray();

    if (empty($customerIds) || empty($serviceIds)) {
        $this->command->info('Tidak ada pelanggan atau layanan yang tersedia untuk
membuat pesanan. Silakan jalankan UserSeeder dan ServiceSeeder terlebih dahulu.');
```

return;

```
    }

    $faker = \Faker\Factory::create('id_ID'); // Menggunakan Faker Indonesia

    for ($i = 0; $i < 50; $i++) { // Buat 50 pesanan sampel
        $customer = User::find($faker->randomElement($customerIds));
        $orderDate = Carbon::instance($faker->dateTimeBetween('-3 months', 'now')); //
Pesanan dalam 3 bulan terakhir

        $orderStatusOptions = ['baru', 'diproses', 'siap diambil', 'selesai', 'dibatalkan'];
        $paymentStatusOptions = ['belum_bayar', 'lunas'];

        $statusPesanan = $faker->randomElement($orderStatusOptions);
        $paymentStatus = ($statusPesanan === 'dibatalkan') ? 'belum_bayar' : $faker-
>randomElement($paymentStatusOptions);

        $paidAt = null;
        $processedByAdminId = null;

        if ($paymentStatus === 'lunas' && $statusPesanan !== 'dibatalkan') {
            $paidAt = $orderDate->copy()->addDays($faker->numberBetween(0, 5))-
>addHours($faker->numberBetween(1, 12));
            if (!empty($adminIds)) {
                $processedByAdminId = $faker->randomElement($adminIds);
            }
        }

        if ($statusPesanan === 'selesai' && $paymentStatus === 'belum_bayar') {
            // Jika selesai tapi belum bayar, pastikan tidak ada paid_at
            $paidAt = null;

```

```

    }

    $order = Order::create([
        'user_id' => $customer->id,
        'kode_pesanan' => 'INV-' . strtoupper(Str::random(4)) . '-' . $orderDate-
>format('YmdHis') . $i,
        'order_date' => $orderDate,
        'status_pesanan' => $statusPesanan,
        'total_amount' => 0, // Akan diupdate nanti
        'catatan_pelanggan' => $faker->optional(0.3)->sentence, // 30% kemungkinan ada
catatan
        'payment_status' => $paymentStatus,
        'paid_at' => $paidAt,
        'processed_by_admin_id' => $processedByAdminId,
        'catatan_internal' => $faker->optional(0.2)->sentence,
        'created_at' => $orderDate,
        'updated_at' => $orderDate->copy()->addHours($faker->numberBetween(1, 24)),
    ]);

    $numberOfItems = $faker->numberBetween(1, 4); // 1 sampai 4 item per pesanan
    $currentTotalAmount = 0;

    for ($j = 0; $j < $numberOfItems; $j++) {
        $service = Service::find($faker->randomElement($serviceIds));
        if (!$service)
            continue;

        $quantity = ($service->satuan === 'kg') ? $faker->randomFloat(1, 0.5, 5) : $faker-
>numberBetween(1, 10); // Kuantitas kg atau pcs
        $hargaSaatPesan = $service->harga; // Ambil harga dari master layanan
        $subTotal = $hargaSaatPesan * $quantity;

        OrderDetail::create([
            'order_id' => $order->id,
            'service_id' => $service->id,
            'quantity' => $quantity,
            'harga_saat_pesan' => $hargaSaatPesan,
            'sub_total' => $subTotal,
        ]);
        $currentTotalAmount += $subTotal;
    }

    // Update total_amount di pesanan
    $order->total_amount = $currentTotalAmount;

```

```

        $order->save();
    }
}
}

```

DATABASE

ServiceSeeder

```
<?php
```

```
namespace Database\Seeders;
```

```
use Illuminate\Database\Console\Seeds\WithoutModelEvents;
```

```
use Illuminate\Database\Seeder;
```

```
use App\Models\Service;
```

```
class ServiceSeeder extends Seeder
```

```
{
```

```
    /**
```

```
     * Run the database seeds.
```

```
     */
```

```
    public function run(): void
```

```
    {
```

```
        // Hapus data lama agar tidak duplikat jika seeder dijalankan berkali-kali
```

```
        // Service::truncate(); // Hati-hati jika ada foreign key constraint
```

```
        $servicesData = [
```

```
            // Paket Kiloan
```

```
            ['tipe_paket' => 'reguler', 'satuan' => 'kg'],
```

```
            ['tipe_paket' => 'kilat', 'satuan' => 'kg'],
```

```
            // Paket Satuan (Pcs) - Contoh
```

```
            // Anda bisa menambahkan nama layanan spesifik jika logika otomatis tidak cukup
```

```
            // atau biarkan model yang men-generate nama_layanan.
```

```
            ['tipe_paket' => 'reguler', 'satuan' => 'pcs', /* 'nama_layanan' => 'Kemeja Reguler (Pcs)' */],
```

```
            ['tipe_paket' => 'kilat', 'satuan' => 'pcs', /* 'nama_layanan' => 'Kemeja Kilat (Pcs)' */],
```

```
            // Contoh layanan lain jika ada
```

```
            ['tipe_paket' => 'reguler', 'satuan' => 'pcs', 'nama_layanan' => 'Boneka Reguler (Pcs)'],
```

```

        // ['tipe_paket' => 'express', 'satuan' => 'kg', 'nama_layanan' => 'Karpét Express
        (Kg)'] // Jika ada tipe paket lain
    ];

    foreach ($servicesData as $data) {
        // Biarkan event 'saving' di model Service yang mengisi 'nama_layanan',
        'harga', dan 'estimasi_durasi'
        Service::create([
            'tipe_paket' => $data['tipe_paket'],
            'satuan' => $data['satuan'],
            // Jika Anda ingin override nama_layanan yang di-generate model,
            // tambahkan di sini:
            // 'nama_layanan' => $data['nama_layanan'] ?? null,
            // Jika Anda ingin override harga atau estimasi, juga bisa ditambahkan,
            // tapi idealnya biarkan model yang mengatur berdasarkan tipe_paket dan
            // satuan.
        ]);
    }
}
}
}

```

DATABASE

UserSeeder

```
<?php
```

```
namespace Database\Seeders;
```

```
use Illuminate\Database\Console\Seeds\WithoutModelEvents;
```

```
use Illuminate\Database\Seeder;
```

```
use App\Models\User;
```

```
use Illuminate\Support\Facades\Hash;
```

```
class UserSeeder extends Seeder
```

```
{
```

```
    /**
```

```
     * Run the database seeds.
```

```
     */
```

```
    public function run(): void
```

```
    {
```

```
        // Buat Admin Utama
```

```
        User::create([
```

```
            'name' => 'Admin Laundry',
```

```
            'email' => 'admin@laundry.test',
```

```
            'password' => Hash::make('password'), // Ganti dengan password yang aman
```

```
            'role' => 'admin',
```

```
            'phone' => '081234567890',
```

```

        'address' => 'Jl. Admin No. 1',
        'email_verified_at' => now(),
    ));

// Buat Admin Tambahan (Opsional)
User::create([
    'name' => 'Staff Admin',
    'email' => 'staff@laundry.test',
    'password' => Hash::make('password'),
    'role' => 'admin',
    'phone' => '089876543210',
    'address' => 'Jl. Staff No. 2',
    'email_verified_at' => now(),
]);

// Buat Beberapa Pelanggan menggunakan Factory (lebih baik) atau create manual
// Jika menggunakan factory, pastikan UserFactory Anda sudah disesuaikan
// User::factory(20)->create(); // Ini jika UserFactory sudah diatur untuk role pelanggan

// Atau buat pelanggan secara manual:
User::create([
    'name' => 'Budi Pelanggan',
    'email' => 'budi@pelanggan.test',
    'password' => Hash::make('password'),
    'role' => 'pelanggan',
    'phone' => '081111111111',
    'address' => 'Jl. Pelanggan Budi No. 10',
    'email_verified_at' => now(),
]);

User::create([
    'name' => 'Ani Pelanggan',
    'email' => 'ani@pelanggan.test',
    'password' => Hash::make('password'),
    'role' => 'pelanggan',
    'phone' => '082222222222',
    'address' => 'Jl. Pelanggan Ani No. 20',
    'email_verified_at' => now(),
]);

// Untuk data yang lebih banyak, gunakan factory.
// Pastikan UserFactory Anda menghasilkan user dengan role 'pelanggan'
// dan mengisi field phone & address jika perlu.
// Contoh modifikasi UserFactory:
// File: database/factories/UserFactory.php
/*
public function definition(): array
{
    return [
        'name' => fake()->name(),
        'email' => fake()->unique()->safeEmail(),
        'email_verified_at' => now(),
    ];
}

```

```

        'password' => static::$password ??= Hash::make('password'),
        'remember_token' => Str::random(10),
        'role' => 'pelanggan', // Default role
        'phone' => fake()->phoneNumber(),
        'address' => fake()->address(),
    ];
}
// Tambahkan ini di UserFactory.php untuk membuat admin jika perlu:
public function admin(): static
{
    return $this->state(fn (array $attributes) => [
        'role' => 'admin',
    ]);
}
*/
// Jika UserFactory sudah siap, Anda bisa buat banyak pelanggan seperti ini:
if (class_exists(Database\Factories\UserFactory::class)) {
    User::factory(10)->create(); // Membuat 10 pelanggan tambahan
}
}
}

```

ENV

```

APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:U+FvvUzzTFSVuHI0IAg6sQ95tXbY5STjzXdtSjSa7dA=
APP_DEBUG=true
APP_URL=http://localhost

APP_LOCALE=en
APP_FALLBACK_LOCALE=en
APP_FAKER_LOCALE=en_US

APP_MAINTENANCE_DRIVER=file
# APP_MAINTENANCE_STORE=database

PHP_CLI_SERVER_WORKERS=4

BCRYPT_ROUNDS=12

LOG_CHANNEL=stack
LOG_STACK=single
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=sqlite
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel

```

DB_USERNAME=root
DB_PASSWORD=

SESSION_DRIVER=database
SESSION_LIFETIME=120
SESSION_ENCRYPT=false
SESSION_PATH=/
SESSION_DOMAIN=null

BROADCAST_CONNECTION=log
FILESYSTEM_DISK=local
QUEUE_CONNECTION=database

CACHE_STORE=database
CACHE_PREFIX=

MEMCACHED_HOST=127.0.0.1

REDIS_CLIENT=phpredis
REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_MAILER=log
MAIL_SCHEME=null
MAIL_HOST=127.0.0.1
MAIL_PORT=2525
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_FROM_ADDRESS="hello@example.com"
MAIL_FROM_NAME="\${APP_NAME}"

AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
AWS_DEFAULT_REGION=us-east-1
AWS_BUCKET=
AWS_USE_PATH_STYLE_ENDPOINT=false

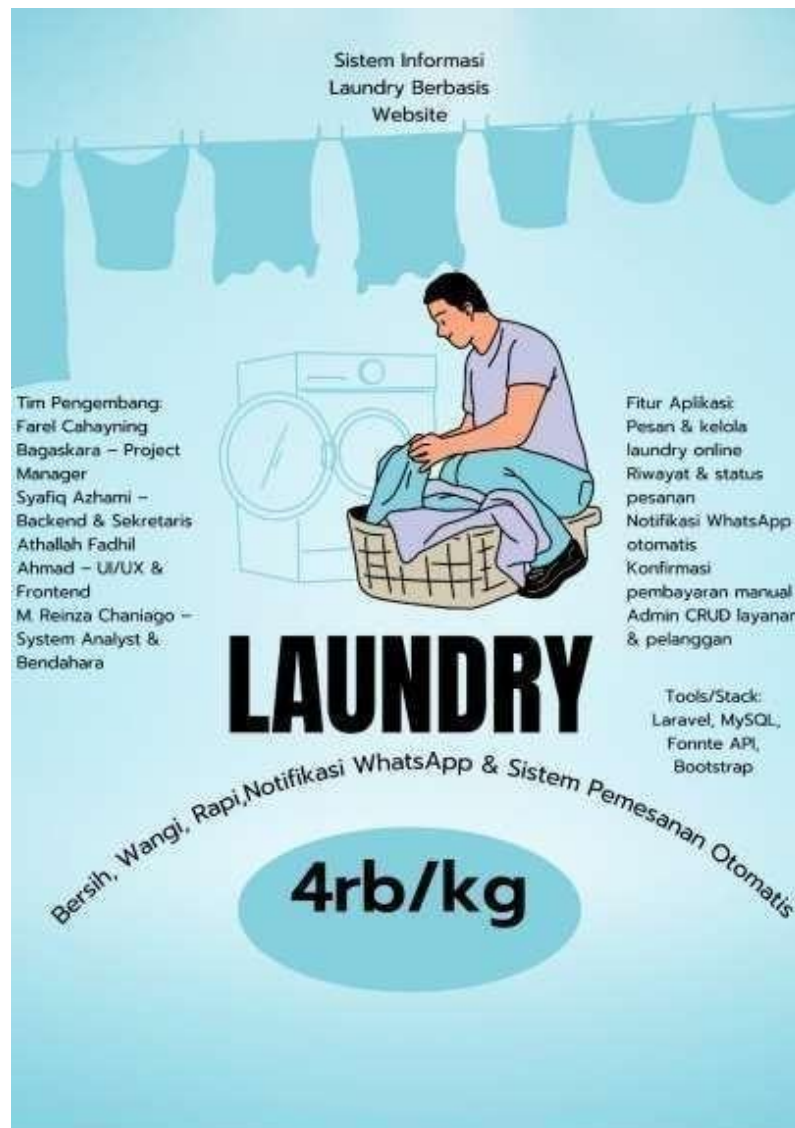
VITE_APP_NAME="\${APP_NAME}"

10. Link Video profil Luaran

https://www.canva.com/design/DAGseu4UXTk/S_IXAf2rN1nw93MulB5Tiw/edit?utm_content=DAGseu4UXTk&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

9. Poster

Gambar 4. 7Poster



10. Link Slide Presentasi

https://www.canva.com/design/DAGsvKpT4Ho/JhSFQL2VQqZ3uYZFO5Oh4Q/edit?utm_content=DAGsvKpT4Ho&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton