

**LAPORAN MANAJEMEN PROYEK TEKNOLOGI INFORMASI PEMBUATAN
SISTEM MANAGEMEN KURIR PAKET ALINDO CARGO**



Disusun Oleh :

M. Izza Mubarak	2200018432 (Project Manager)
Daffa Fachrezy	2200018433 (Back End)
Haura Faesa Syahputri A	2200018434 (System Analyst)
M. Sandy Alfachari	2200018479 (Front End)

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

HALAMAN PENGESAHAN

LAPORAN MANAJEMEN PROYEK TEKNOLOGI INFORMASI PEMBUATAN SISTEM
MANAGEMENT KURIR PAKET ALINDO CARGO



Muhammad Izza Mubarak 2200018432
Daffa Fachrezy Dharmawan 2200018433
Haura Faesa Syahputri Arindi 2200018434
Muhammad Sandy Alfachari 2200018479

PEMBIMBING

: *Dwi Normawati S.T., M.Eng.*

2/9/2025

19860804 201606 011 1236590

PENGUJI

: *Bambang Robiin, S.T., M.T.*

3/9/2025

197907202005011002

Yogyakarta, 02 September 2025

Kaprodi S1 Informatika

Dr. Murinto, S.Si., M.Kom.

NIPM. 19730710 200409 111 0951298

DAFTAR ISI

DAFTAR ISI	III
DAFTAR TABEL	IV
DAFTAR GAMBAR	V
BAB I PENDAHULUAN	1
1. 1. Latar Belakang.....	1
1. 2. Project Charter	2
1. 3. Stakeholder	4
BAB II PERENCANAAN PROYEK	7
A. Analisis Kelayakan (SWOT).....	7
B. Work Breakdown Structure.....	8
C. Kebutuhan Sumber Daya	8
D. Rencana Jadwal Perencanaan Proyek.....	9
E. Kebutuhan Sistem	10
F. Rencana Nilai Proyek	11
BAB III PELAKSANAAN PROYEK	15
A. Realisasi Jadwal Pelaksanaan	15
B. Realisasi hasil pekerjaan	16
C. Penjamin kualitas proyek.....	64
D. Kendala Proyek	70
E. Keberlanjutan Proyek	70
BAB IV PENUTUP	71
A. Kesimpulan	71
B. Saran.....	71
LAMPIRAN	72

DAFTAR TABEL

Tabel 1. 1 Stakeholder	4
Tabel 2. 1 Kebutuhan Sumber Daya Fisik.....	9
Tabel 2. 2 Jadwal Perencanaan Proyek.....	9
Tabel 2. 3 Rencana Nilai Proyek	11
Tabel 3. 1 Realisasi Jadwal Pelaksanaan	15
Tabel 3. 2 Analisis Kebutuhan Fungsional.....	16
Tabel 3. 3 Analisis Kebutuhan Non-Fungsional	17
Tabel 3. 4 Lampiran Kode Program Front-End	42
Tabel 3. 5 Lampiran Kode Program Back-End	54
Tabel 3. 6 Hasil Testing	64
Tabel 4. 1 Dokumentasi.....	84

DAFTAR GAMBAR

Gambar 1.1 Logo Perusahaan Alindo Cargo.....	5
Gambar 1.2 Lokasi Alindo Cargo	6
Gambar 2.1 Work Breakdown Structure	8
Gambar 3. 1 Side Map Super Admin.....	17
Gambar 3. 2 Side Map Admin	18
Gambar 3. 3 Side Map Tim Operasional	18
Gambar 3. 4 Invoice Kertas Alindo Cargo	19
Gambar 3. 5 Proses Bisnis Saat Ini.....	19
Gambar 3. 6 Proses Bisnis Setelah Ada Sistem	20
Gambar 3. 7 Use Case Diagram	21
Gambar 3. 8 Activity Diagram login.....	22
Gambar 3. 9 Activity Diagram Setelah Login Admin.....	23
Gambar 3. 10 Activity Diagram Vendor	24
Gambar 3. 11 Acitivity Diagram Cancel Paket.....	25
Gambar 3. 12 Activity Diagram Ubah Status Paket	26
Gambar 3. 13 Activity Diagram Tim Operasional	27
Gambar 3. 14 Activity Diagram Tambah Invoice Tim Operasional.....	28
Gambar 3. 15 Activity Diagram Super Admin	29
Gambar 3. 16 Activity Diagram Tambah Karyawan Super Admin.....	30
Gambar 3. 17 Wireframe Login	31
Gambar 3. 18 Wireframe Dashboard Super Admin.....	31
Gambar 3. 19 Wireframe Halaman Paket Super Admin	32
Gambar 3. 20 Wireframe Halaman Karyawan "Admin" Super Admin	32
Gambar 3. 21 Wireframe Halaman Karyawan "Tim Operasional" Super Admin	32
Gambar 3. 22 Wireframe Dashboard Admin	33

Gambar 3. 23 Wireframe Halaman Paket Admin	33
Gambar 3. 24 Wireframe Halaman Vendor Admin	33
Gambar 3. 25 Wireframe Halaman Biaya Operasional.....	34
Gambar 3. 26 Wireframe Dashboard Tim Operasional	34
Gambar 3. 27 Wireframe Halaman Paket Tim Operasional.....	34
Gambar 3. 28 UI Login.....	35
Gambar 3. 29 UI Dashboard Super Admin	35
Gambar 3. 30 UI Dashboard Paket	36
Gambar 3. 31 UI Dashboard Karyawan Admin.....	36
Gambar 3. 32 UI Dashboard Tambah Admin	36
Gambar 3. 33 UI Dashboard Karyawan Tim Operasional	37
Gambar 3. 34 UI Dashboard Tambah Tim Operasional	37
Gambar 3. 35 UI Dashboard Biaya Operasional	37
Gambar 3. 36 UI Dashboard Edit Biaya Operasional	38
Gambar 3. 37 UI Dashboard Laporan	38
Gambar 3. 38 UI Dashboard Cetak Laporan	38
Gambar 3. 39 UI Dashboard Admin.....	39
Gambar 3. 40 UI Dashboard Data Paket	39
Gambar 3. 41 UI Dashboard Edit Invoice	39
Gambar 3. 42 UI Dashboard Tambah Vendor	40
Gambar 3. 43 UI Dashboard Biaya Operasional	40
Gambar 3. 44 UI Dashboard Input Biaya Operasional.....	40
Gambar 3. 45 UI Dashboard Edit Biaya Operasional	41
Gambar 3. 46 UI Dashboard Tim Operasional.....	41
Gambar 3. 47 UI Dashboard Input Invoice	41
Gambar 3. 48 Deployment	64
Gambar 3. 49 Uji Normalitas.....	68
Gambar 3. 50 Uji Autokorelasi	68

Gambar 3. 51 Uji Multikolinearitas	68
Gambar 3. 52 Uji Heteroskedastisitas	69
Gambar 3. 53 Uji T	69

BAB I

PENDAHULUAN

1. 1. Latar Belakang

Sektor logistik dan ekspedisi kargo merupakan tulang punggung dari sistem distribusi barang nasional maupun internasional. Dalam operasional hariannya, perusahaan ekspedisi menangani ribuan transaksi pengiriman yang masing-masing memerlukan dokumen pendukung, termasuk *invoice* (faktur pengiriman) sebagai dokumen keuangan dan bukti transaksi. Namun, kompleksitas transaksi yang tinggi, penggunaan sistem manual atau semi-digital, serta lemahnya kontrol internal membuka peluang terjadinya duplikasi *invoice* baik yang bersifat tidak disengaja maupun sebagai bentuk penipuan terencana.

Masalah duplikasi *invoice* ini dapat terjadi karena sistem yang tidak terintegrasi antar departemen, lemahnya protokol validasi data, serta minimnya pengawasan berbasis teknologi terhadap akses dan modifikasi dokumen keuangan. Dalam banyak kasus, *invoice* yang sama dapat diinput lebih dari satu kali ke dalam sistem, atau bahkan sengaja diduplikasi dengan nomor yang sedikit dimodifikasi untuk menciptakan transaksi fiktif.

Dalam jangka panjang, permasalahan ini bukan hanya menyebabkan kerugian finansial, seperti pembayaran ganda atau pengeluaran untuk pengiriman palsu, tetapi juga mengganggu integritas sistem perusahaan, menurunkan kepercayaan pelanggan dan mitra, hingga menimbulkan konsekuensi hukum karena potensi manipulasi laporan keuangan dan pelanggaran pajak.

Kondisi ini menunjukkan bahwa perusahaan ekspedisi perlu melakukan transformasi sistemik dalam hal digitalisasi proses, pengendalian akses, dan penerapan audit berbasis risiko. Tanpa intervensi yang tepat, celah ini bisa dimanfaatkan oleh pihak-pihak tidak bertanggung jawab untuk melakukan *fraud* yang berulang dan sistematis.

Dalam konteks ini, klien sebagai pemilik perusahaan ekspedisi memegang peranan penting dalam menentukan arah transformasi operasional. Salah satu kekhawatiran utama yang muncul adalah ketidakefisienan kerja tim akibat sistem pengelolaan *invoice* yang rumit, lambat, dan rawan kesalahan. Banyak waktu dan

energi tim operasional dan keuangan dihabiskan untuk memverifikasi invoice satu per satu, melakukan konfirmasi silang manual, dan menelusuri dokumen pendukung, yang seharusnya bisa dialihkan untuk pekerjaan bernilai strategis.

Dengan menerapkan sistem yang terintegrasi dan otomatis mulai dari pembuatan invoice, hingga pencatatan keuangan perusahaan tidak hanya mampu mencegah terjadinya duplikasi, tetapi juga memastikan bahwa seluruh proses berjalan dengan transparan, efisien, dan terdokumentasi dengan baik.

1. 2. Project Charter

A. Tujuan

Membangun sistem manajemen *invoice* digital yang terintegrasi untuk mencegah duplikasi faktur, menyederhanakan proses kerja tim operasional dan keuangan, dan memungkinkan manajemen peran pengguna untuk mengatur akses dan tanggung jawab setiap divisi.

B. Ruang Lingkup

Pengembangan *website* sistem manajemen internal perusahaan Kargo Alindo yang memiliki tiga sisi utama yaitu *dashboard* super admin, admin, dan tim operasional yang bersifat informatif dan interaktif. Website ini akan saling terintegrasi untuk mencegah duplikasi faktur. Serta memungkinkan peran pengguna untuk mengatur akses dan tanggung jawab masing – masing. Website ini belum mendukung fitur pelacakan paket, dan fitur lain yang bersangkutan dengan konsumen paket. Berikut ini adalah fitur umum yang bisa di akses oleh pengguna :

1. Super Admin

- a) *Dashboard* : Halaman beranda yang menghasilkan pendapatan, jumlah paket, grafik pesanan bulanan, dan titik lokasi pengantaran.
- b) Data Paket : Berisikan tabel yang terdiri dari data paket, serta nama tim operasional yang mengelola.
- c) Karyawan : Fitur ini memungkinkan super admin dapat menambahkan karyawan seperti admin, dan tim operasional sehingga hanya mereka yang dapat mengakses *website* ini.

d) Biaya Operasional : Berisikan total harga paket, pengiriman, vendor, biaya lainnya seperti bahan bakar dan lain - lain, sehingga dapat ditemukan total untung dan rugi.

e) Laporan : Fitur ini berisikan rekap data pesanan paket perbulan.

2. Admin

a) Data Paket : Berisikan tabel yang terdiri dari data paket, serta nama tim operasional yang mengelola.

b) Vendor : Fitur ini berisi tabel yang terdiri dari nama vendor, lokasi vendor, dan harga vendor.

c) Biaya Operasional : Berisikan total harga paket, pengiriman, vendor, biaya lainnya seperti bahan bakar dan lain - lain, sehingga dapat ditemukan total untung dan rugi.

3. Tim Operasional

a) Invoice Paket : Fitur ini adalah form input untuk menghasilkan data paket yang akan dikelola admin.

Jadi, alur dari website ini ialah dimulai dari tim operasional yang mengisikan data paket kemudian ditambahkan vendor dan biaya operasional terkait pengiriman paket ini. Terakhir, semua data akan masuk ke super admin dan super admin dapat mengetahui semua data dari mulai paket, karyawan yang mengisikan, serta vendor terkait.

1. 3. Stakeholder

Pada tabel 1.1 berisi stakeholder yaitu struktur yang memuat apa saja tugas dan posisi dari kelompok.

Tabel 1. 1 Stakeholder

Nama Stakeholder					
	Instansi	Izza	Sandy	Haura	Daffa
Organsisasi		Project Manager & Tim Proyek (UI/UX)	Tim Proyek (Front End)	Tim Proyek (Data Analyst) & Sekretaris	Tim Proyek (Back End)
Peran dalam proyek	Sponsor proyek dan salah satu pendiri lembaga	Merencanakan proyek dan memastikan pencapaian juga tujuan akhir berjalan dengan efektif dan efisien	Mengembangkan antarmuka pengguna berbasis web untuk dashboard multi-brand, input invoice, serta tampilan data konsumen dan tarif pengiriman. Memastikan kompatibilitas antar perangkat dan responsifitas.	Menyusun laporan, dan Menganalisis data untuk membuat visualisasi dan laporan dalam pengambilan keputusan berbasis data yang efektif	Membangun dan mengelola server, database, dan API untuk otentikasi pengguna (berbasis role), manajemen data invoice, tarif, vendor, serta log aktivitas. Menjamin keamanan dan skalabilitas sistem.

1. 4. Profil Perusahaan

A. Alindo Cargo



PT ALIF LOGISTIK INDONESIA

Gambar 1.1 Logo Perusahaan Alindo Cargo

Alindo Cargo merupakan salah satu jasa layanan antar paket atau barang yang terletak di daerah Godean, Yogyakarta. Usaha ini di dirikan oleh saudara Tengku Iswandi. Alindo Cargo didirikan pada pertengahan tahun 2018 dan masih berdiri hingga tahun ini. Di google maps, Alindo Cargo memiliki rating 4,6 dari 76 ulasan. Alindo Cargo dulu memiliki 3 cabang namun semenjak covid 2020 cabang tersebut ditutup dan sekarang hanya ada 1 cabang yang tersisa, yaitu cabang yang ada di daerah godean.

B. Lokasi dan Kontak Utama

1. Nama Usaha : Alinfo Cargo
2. Kontak Pemilik : +62 812-2848-3338
3. Alamat : Jl. Nogosaren Baru No.60 A, Nogosaren, Nogotirto, Kec. Gamping, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55292



Gambar 1.2 Lokasi Alindo Cargo

BAB II

PERENCANAAN PROYEK

2. 1. Analisis Kelayakan (SWOT)

A. Strengths (Kekuatan)

1. Peningkatan Efisiensi Operasional

Sistem otomatisasi memungkinkan tracking paket real-time, manajemen kurir, dan pelaporan yang lebih efisien.

2. Dukungan Manajemen Proyek yang Terdokumentasi

Laporan manajemen proyek menunjukkan perencanaan terstruktur.

3. Pemanfaatan Teknologi Terkini

Sistem berbasis web/mobile yang dapat diakses kapan saja memungkinkan fleksibilitas tinggi untuk admin, kurir, dan pelanggan.

B. Weaknesses (Kelemahan)

1. Ketergantungan pada Infrastruktur

Kualitas sistem bergantung pada konektivitas internet dan perangkat yang digunakan kurir dan admin.

2. Keterbatasan Sumber Daya

Proyek dijalankan dengan dana terbatas, dan anggota tim sedikit.

C. Opportunities (Peluang)

1. Peningkatan Citra Perusahaan

Sistem digital dapat menjadi nilai jual kompetitif yang menunjukkan profesionalisme dan modernisasi layanan Alindo Cargo.

2. Tren Digitalisasi Industri Logistik

Banyak perusahaan logistik berlomba mendigitalisasi operasional, membuat proyek ini sangat relevan dan memiliki potensi tinggi.

3. Skalabilitas Sistem

Sistem bisa dikembangkan untuk wilayah distribusi yang lebih luas atau kemitraan dengan jasa pengiriman lain.

D. Threats (Ancaman)

1. Persaingan dari Perusahaan Besar

Perusahaan logistik besar seperti JNE, J&T, SiCepat sudah memiliki sistem serupa yang lebih matang.

2. Ancaman Keamanan Data

Jika sistem tidak dirancang dengan keamanan yang kuat, bisa terjadi kebocoran data pelanggan atau kurir.

3. Resistensi terhadap Perubahan

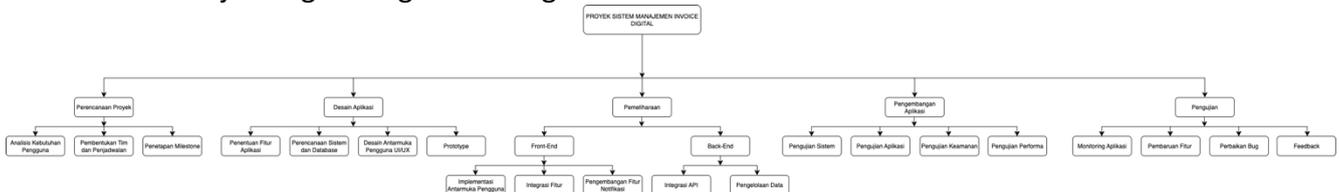
Beberapa karyawan mungkin menolak beralih dari sistem manual ke digital karena merasa kurang nyaman.

4. Kegagalan Integrasi

Risiko teknis dalam integrasi sistem baru dengan sistem operasional lama (misalnya, database, sistem keuangan).

2. 2. Work Breakdown Structure

Work Breakdown Structure berisi uraian pekerjaan proyek yang dipecah menjadibagian-bagian kecil agar lebih mudah dikelola.



Gambar 2.3 Work Breakdown Structure

2. 3. Kebutuhan Sumber Daya

A. Sumber Daya Manusia

1. Izza : Project Manager & Tim Proyek (UI/UX)

Tugas : Merencanakan proyek dan memastikan pencapaian juga tujuan akhir berjalan dengan efektif dan efisien.

2. Sandy : Tim Proyek (Front End)

Tugas : Mengembangkan antarmuka pengguna berbasis web untuk dashboard multi-brand, input invoice, serta tampilan data konsumen dan tarif pengiriman. Memastikan kompatibilitas antar perangkat dan responsifitas .

3. Haura : Tim Proyek (Data Analyst) & Sekretaris

Tugas : Menyusun laporan, dan Menganalisis data untuk membuat visualisasi dan laporan dalam pengambilan keputusan berbasis data yang efektif.

4. Daffa : Tim Proyek (Back End)

Tugas : Membangun dan mengelola server, database, dan API untuk otentikasi pengguna (berbasis role), manajemen data invoice, tarif, vendor, serta log aktivitas. Menjamin keamanan dan skalabilitas sistem.

B. Sumber Daya Fisik

Pada tabel 2.1 berisi sumber daya fisik yang diperlukan dalam pembuatan website ini berupa alat pendukung yang dibutuhkan.

Tabel 2. 1 Kebutuhan Sumber Daya Fisik

no	Nama Barang / Aset	Spesifikasi/ Keterangan	Jumlah	Keterangan Tambahan
1	Laptop	Spesifikasi tinggi untuk pengembangan software	4 Unit	Untuk tim dev
2	Koneksi Internet	Internet stabil untuk menggarap proyek	1 set	Diperlukan untuk akses sistem
3	Akun Tools Kolaborasi	GitHub, Figma, Google Workspace, Visual Studio Code, Laragon	Beberapa akun	Untuk mendukung kolaborasi dan dokumentasi
4	Server	Untuk menyimpan database dan menjalankan sistem	1 Akun	Untuk hosting lokal

2. 4. Rencana Jadwal Perencanaan Proyek

Pada 2.2 berisi jadwal perencanaan proyek pembuatan website ini.

Tabel 2. 2 Jadwal Perencanaan Proyek

Task	Start Date	Due Date	Duration	April				Mei				Juni				Juli			
Perencanaan Proyek				1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Analisis Kebutuhan Pengguna	12-April-2025	26-Juli-2025	106																
Pembentukan Tim dan Penjadwalan	23-April-2025	26-Juli-2025	95																
Penetapan Milestone	23-April-2025	26-Juli-2025	95																
Desain Aplikasi				1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Penentuan Fitur Aplikasi	10-Juni-2025	26-Juli-2025	47																
Desain Antarmuka Pengguna UI/UX	10-Juni-2025	26-Juli-2025	47																
Perencanaan Sistem dan Database	11-Juni-2025	26-Juli-2025	46																
Prototype	10-Juni-2025	26-Juli-2025	47																

Pengembangan Aplikasi				1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Front-End																			
Implementasi Antarmuka Pengguna	15-Juni- 2025	26-Juli-2025	42																
Integrasi Fitur	01-Juni-2025	26-Juli-2025	56																
Pengembangan Fitur Notifikasi	01-Juni-2025	26 -Juli- 2025	56																
Back-End																			
Integrasi API	15-Juni-2025	26-Juli-2025	42																
Pengelolaan Data	15-Juni-2025	26-Juli-2025	42																
Pengujian				1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Pengujian Aplikasi	10-juli-2025	26-Juli-2025	17																
Pengujian Sistem	10-juli-2025	26-Juli-2025	17																
Pengujian Keamanan	10-juli-2025	26-Juli-2025	17																
Pengujian Performa	10-juli-2025	26-Juli-2025	17																
Pemeliharaan				1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Monitoring Aplikasi	15-juli-2025	26-Juli-2025	12																
Pembaharuan Fitur	17-juli-2025	26-Juli-2025	10																
Perbaikan Bug	17-juli-2025	26-Juli-2025	10																
Feedback	17-juli-2025	26-Juli-2025	10																

2. 5. Kebutuhan Sistem

Untuk meningkatkan efisiensi pengembangan sistem, sistem web profil membutuhkan teknologi dan peralatan yang mendukung.

A. Analisis Kebutuhan Perangkat Lunak

1. Laragon

Laragon merupakan sebuah software web server apache yang didalamnya sudah tersedia database server mysql dan support php programming.

2. IDE Visual Studio Code

Visual Studio Code merupakan editor kode sumber yang dikembangkan microsoft untuk Windows, linux, dan macOS. Software ini termasuk dukungan untuk debugging, kontrol Git tertanam, penyorotan sintaksis, penyelesaian kode cerdas, dan refactoring kode.

3. Microsoft Word

Software ini merupakan pengolah kata andalan microsoft yang digunakan sebagai membuat MOU, Proposal, laporan dan lain-lain.

4. Google Chrome

Chrome merupakan peramban web yang cepat, sederhana, dan aman khusus

untuk web modern.

5. Draw.io

Draw.io adalah sebuah alat perangkat lunak yang digunakan untuk membuat diagram secara visual. Ini adalah aplikasi berbasis web yang memungkinkan pengguna untuk membuat berbagai jenis diagram, seperti diagram alur kerja, diagram organisasi, dan diagram aliran data.

6. Figma

Figma adalah alat desain berbasis web yang memungkinkan para desainer untuk bekerja secara kolaboratif dalam membuat desain antarmuka pengguna (UI) dan pengalaman pengguna (UX). Figma memungkinkan pengguna untuk membuat, mengedit, dan berbagi desain secara real-time, baik dengan tim lokal maupun tim yang tersebar di berbagai lokasi geografis.

2. 6. Rencana Nilai Proyek

Tabel 2. 3 berisi rencana nilai atau upah proyek perkegiatan dalam proyek ini.

Tabel 2. 3 Rencana Nilai Proyek

RENCANA ANGGARAN BIAYA					
Tahap Perencanaan Proyek					
Kegiatan	Durasi (Jam)	Upah	Biaya Personal	Biaya Bahan	Total Biaya
Analisis Kebutuhan Pengguna	10	Rp20.000	Rp200.000	Rp50.000	Rp250.000
Pembentukan Tim dan Penjadwalan	6	Rp15.000	Rp90.000	Rp -	Rp90.000
Penetapan Milestone proyek	5	Rp16.000	Rp80.000	Rp -	Rp80.000
Total Tahap Perencanaan Proyek	21		Rp370.000	Rp50.000	Rp420.000

Tahap Desain Aplikasi					
Kegiatan	Durasi (Jam)	Upah	Biaya Personal	Biaya Bahan	Total Biaya
Penentuan Fitur Aplikasi	5	Rp18.000	Rp90.000	Rp -	Rp90.000
Desain Antar Muka Pengguna UI/UX	6	Rp20.000	Rp120.000	Rp20.000	Rp140.000
Perancangan Sistem dan Database	6	Rp15.000	Rp90.000	Rp -	Rp90.000
Prototyping	5	Rp16.000	Rp80.000	Rp -	Rp80.000
Total Tahap Perancangan dan Desain Aplikasi	22		Rp380.000	Rp20.000	Rp400.000
Tahap Pengembangan Aplikasi					
Kegiatan	Durasi (Jam)	Upah	Biaya Personal	Biaya Bahan	Total Biaya
Front-End					
Implementasi Antar Muka UI / UX	3	Rp10.000	Rp30.000	Rp -	Rp30.000
Integrasi Fitur Interaktif	3	Rp10.000	Rp30.000	Rp -	Rp30.000
Pengembangan Fitur Notifikasi	3	Rp10.000	Rp30.000	Rp -	Rp30.000
Back-End					
Integrasi API	42	Rp30.000	Rp30.000	Rp -	Rp30.000

Pengelolaan Data	2	Rp10.000	Rp20.000	Rp -	Rp20.000
Total Tahap Pengembangan Aplikasi	53		Rp140.000	Rp -	Rp140.000
Tahap Pengujian					
Kegiatan	Durasi (Jam)	Upah	Biaya Personal	Biaya Bahan	Total Biaya
Pengujian Aplikasi	2	Rp10.000	Rp20.000	Rp -	Rp20.000
Pengujian Sistem	2	Rp10.000	Rp20.000	Rp -	Rp20.000
Pengujian Keamanan	2	Rp10.000	Rp20.000	Rp -	Rp20.000
Pengujian Performa	2	Rp10.000	Rp20.000	Rp -	Rp20.000
Total Tahap Pengujian	8		Rp80.000	Rp -	Rp80.000
Tahap Pemeliharaan					
Kegiatan	Durasi (Jam)	Upah	Biaya Personal	Biaya Bahan	Total Biaya
Monitoring Aplikasi	2	Rp10.000	Rp20.000	Rp -	Rp20.000
Pembaruan Fitur	2	Rp10.000	Rp20.000	Rp -	Rp20.000
Perbaikan Bug	2	Rp10.000	Rp20.000	Rp -	Rp20.000
Feedback	2	Rp10.000	Rp20.000	Rp	Rp20.000

Total Tahap Perencanaan Proyek	8		Rp80.000	Rp -	Rp80.000
Total Durasi dan Biaya Proyek	112				Rp 1.500.000,00

BAB III

PELAKSANAAN PROYEK

3. 1. Realisasi Jadwal Pelaksanaan

Pada tabel 3. 1 berisi realisasi dari jadwal pelaksanaan dari rencana pada tabel

2. 2

Tabel 3. 1 Realisasi Jadwal Pelaksanaan

Task	Start Date	Due Date	Duration	April				Mei				Juni				Juli			
				1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Perencanaan Proyek				1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Analisis Kebutuhan Pengguna	12-April-2025	26-Juli-2025	106																
Pembentukan Tim dan Penjadwalan	23-April-2025	26-Juli-2025	95																
Penetapan Milestone	23-April-2025	26-Juli-2025	95																
Desain Aplikasi				1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Penentuan Fitur Aplikasi	10-Juni-2025	26-Juli-2025	47																
Desain Antarmuka Pengguna UI/UX	10-Juni-2025	26-Juli-2025	47																
Perencanaan Sistem dan Database	11-Juni-2025	26-Juli-2025	46																
Prototype	10-Juni-2025	26-Juli-2025	47																
Pengembangan Aplikasi				1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Front-End																			
Implementasi Antarmuka Pengguna	15-Juni- 2025	26-Juli-2025	42																
Integrasi Fitur	01-Juni-2025	26-Juli-2025	56																
Pengembangan Fitur Notifikasi	01-Juni-2025	26 -Juli- 2025	56																
Back-End																			
Integrasi API	15-Juni-2025	26-Juli-2025	42																
Pengelolaan Data	15-Juni-2025	26-Juli-2025	42																
Pengujian				1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Pengujian Blackbox	10-juli-2025	26-Juli-2025	17																
Pengujian SUS	10-juli-2025	26-Juli-2025	17																
Pengujian Keamanan	10-juli-2025	26-Juli-2025	17																
Pengujian Performa	10-juli-2025	26-Juli-2025	17																
Pemeliharaan				1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Monitoring Aplikasi	15-juli-2025	26-Juli-2025	12																
Pembaharuan Fitur	17-juli-2025	26-Juli-2025	10																
Perbaiki Bug	17-juli-2025	26-Juli-2025	10																
Feedback	17-juli-2025	26-Juli-2025	10																

3. 2. Realisasi hasil pekerjaan

A. Sekretaris & Tim Proyek Data analyst

1. Membuat Proposal
2. Membuat Laporan
3. Menganalisis permintaan client

a) Analisis Kebutuhan Fungsional

Tabel 3. 2 Analisis Kebutuhan Fungsional

Tabel Kebutuhan Fungsional		
No	Kode	Deskripsi
1	SKPL-F1	Pengguna harus terbagi menjadi 3 role yaitu Super Admin, Admin dan Tim Operasional
2	SKPL-F2	Pengguna dapat melakukan login
3	SKPL-F3	Sistem harus bisa redirect pengguna berdasarkan role masing-masing
4	SKPL-F4	Super Admin dapat mencetak dan mengedit data paket, biaya operasional, dan laporan
5	SKPL-F5	Super Admin dapat menghapus, mengedit dan menambah data karyawan
6	SKPL-F6	Admin dapat mengedit data paket dan biaya operasional
7	SKPL-F7	Tim Operasional harus bisa menginputkan invoice
8	SKPL-F8	Sistem harus bisa membuat resi berdasarkan tanggal pembuatan dan sesuai dengan urutan paket pada hari itu secara otomatis
9	SKPL-F9	Sistem harus bisa kelola vendor <i>CRUD</i> setelah data paket baru di tambahkan
10	SKPL-F10	Sistem harus dapat memudahkan tim operasional dalam mencatat detail paket baru, termasuk informasi pengirim, penerima, berat paket, tujuan pengiriman, dan menambah vendor
11	SKPL-F11	Sistem harus bisa untuk mengubah status paket, misalnya dari "Proses" menjadi "Selesai" atau "Batal".
12	SKPL-F12	Sistem harus menghasilkan laporan tentang jumlah paket yang masuk, yang keluar, dan statusnya.
13	SKPL-F13	Sistem harus bisa melakukan cetak invoice paket, cetak resi paket dalam format pdf dan cetak laporan dalam bentuk excel

b) Analisis Kebutuhan Non-Fungsional

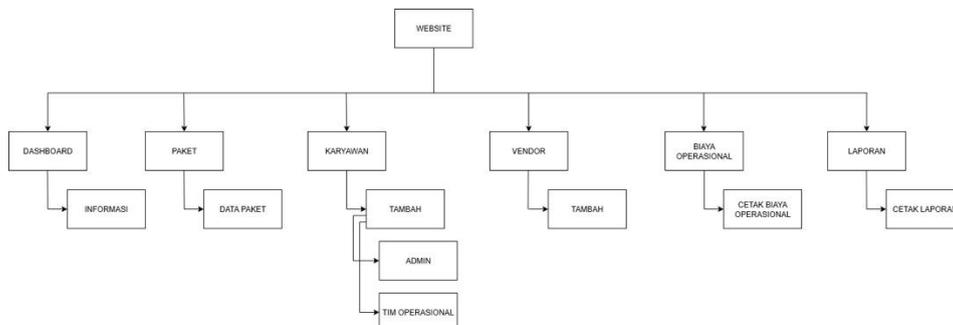
Tabel 3. 3 Analisis Kebutuhan Non-Fungsional

Tabel Kebutuhan Non-Fungsional			
No	Kode	Parameter	Kebutuhan Non Fungsional
1	SKPL-NF1	Keamanan	Sistem harus melindungi data paket yang sensitif, seperti alamat pengirim dan penerima, agar tidak dapat diakses oleh pihak yang tidak berwenang.
2	SKPL-NF2	Kinerja	Sistem harus mampu mengelola sejumlah besar data paket dengan respons yang cepat, termasuk saat pencarian dan pembaruan status.
4	SKPL-NF3	Usability	Antarmuka pengguna sistem harus sederhana dan mudah digunakan, bahkan bagi pengguna yang tidak terbiasa dengan teknologi.
5	SKPL-NF4	Scalability (Skalabilitas)	Sistem harus dapat dengan mudah menangani peningkatan volume paket seiring waktu tanpa mengorbankan kinerja.
5	SKPL-NF5	Ketersediaan	Sistem harus tersedia secara konsisten untuk mencatat dan mengelola paket baru, tanpa waktu tidak aktif yang berlebihan.
6	SKPL-NF6	Integritas Data	Sistem harus menjaga keutuhan data paket, termasuk mencegah duplikasi data atau kehilangan data.

4. Site Map

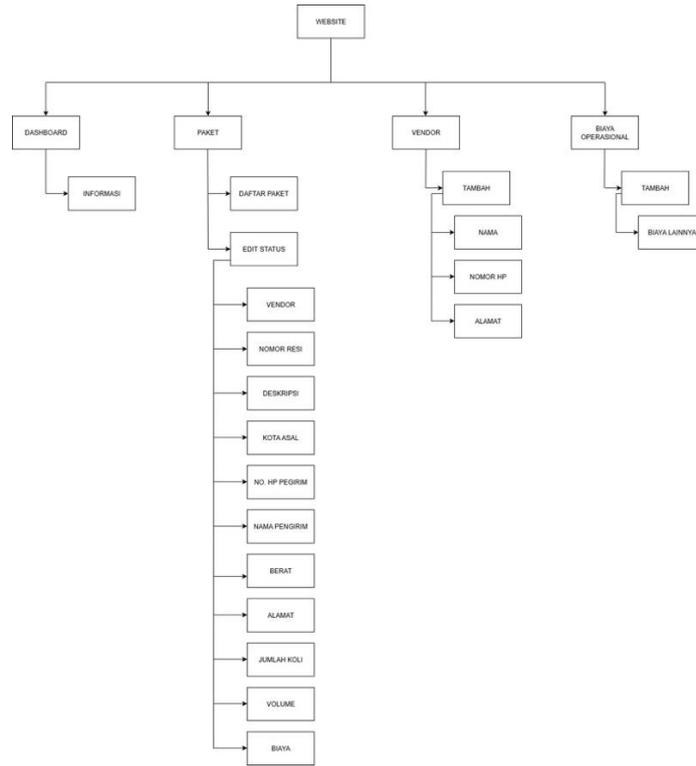
Site map website ini memberikan informasi tentang fitur apa saja yang bisa diakses dalam website tersebut, site map ini terbagi menjadi 3 halaman yaitu Super Admin, Admin dan Tim Operasional.

a) Site Map Super Admin



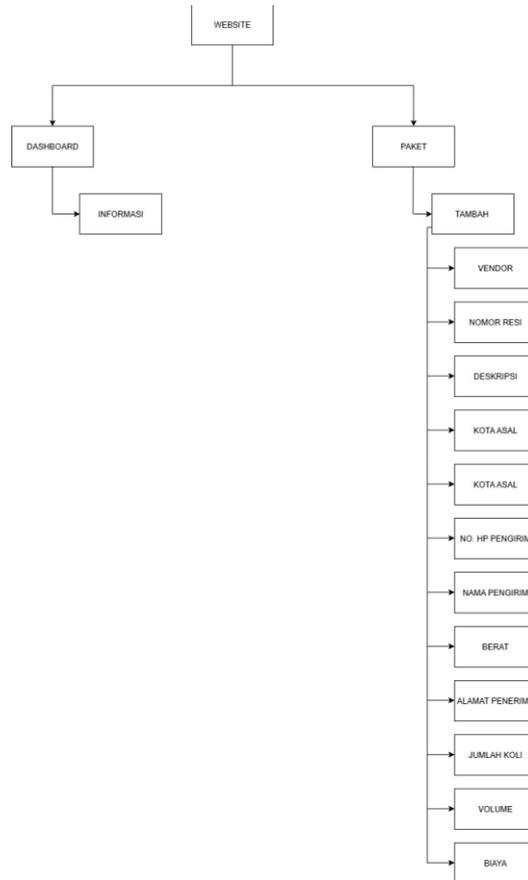
Gambar 3. 1 Side Map Super Admin

b) Side Map Admin



Gambar 3. 2 Side Map Admin

c) Side Map Tim Operasional



Gambar 3. 3 Side Map Tim Operasional

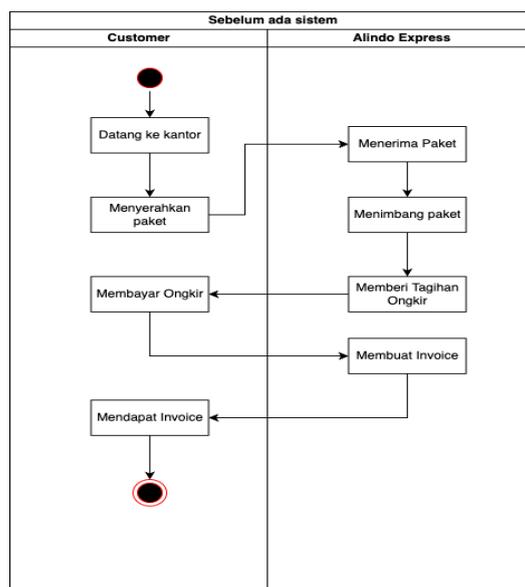
5. Proses Bisnis Sebelum Adanya Sistem

Sistem manajemen paket Aliando Cargo sebelumnya mendata paket dengan cara yang manual, yaitu dengan kertas yang diisi secara manual. Selanjutnya, sistem manajemen yang akan dibuat ini akan memungkinkan Aliando Cargo secara efektif mengelola ribuan paket yang dikirim setiap hari dengan akurat.

Gambar 3. 4 Invoice Kertas Alindo Cargo

a) Proses Bisnis Sebelum Ada Sistem

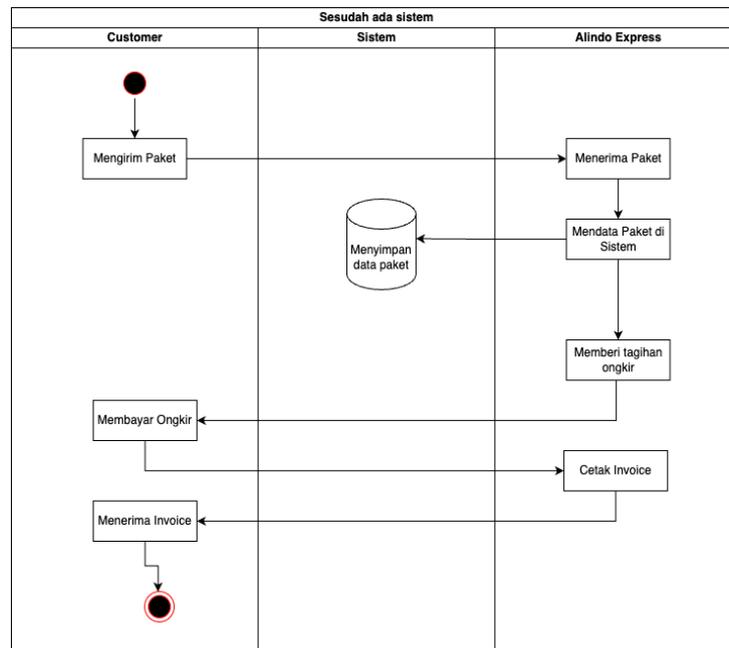
Berdasarkan gambar diatas, proses bisnis yang dilakukan saat ini adalah manual dengan cara customer datang ke kantor, lalu karyawan akan menerima paket tersebut lalu melakukan penimbangan berat paket dan memberikan tagihan ongkir, lalu menuliskan invoice.



Gambar 3. 5 Proses Bisnis Saat Ini

b) Proses Bisnis Setelah Ada Sistem

Setelah ada sistem pendataan paket perusahaan aliando cargo sudah menggunakan input website.



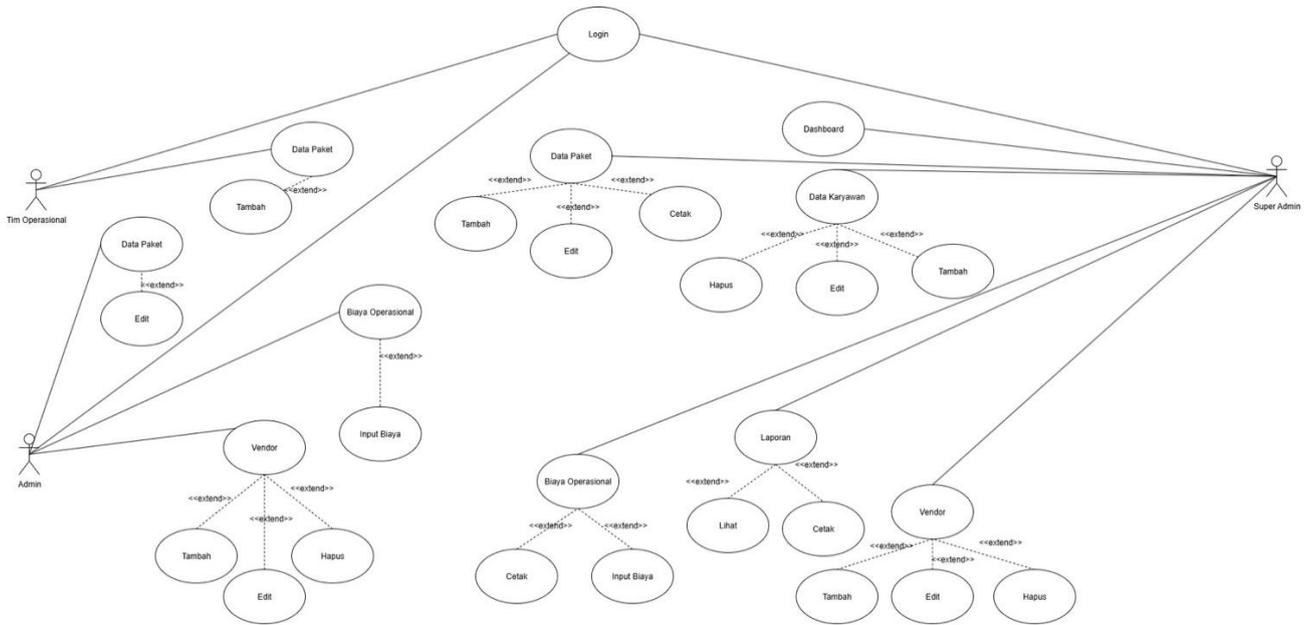
Gambar 3. 6 Proses Bisnis Setelah Ada Sistem

6. Perancangan Sistem

Perancangan sistem adalah sebuah proses merancang dan mendesain sebuah sistem dengan langkah-langkah operasi yang diperlukan untuk mendukung operasi sistem. Tujuannya untuk memenuhi kebutuhan pengguna sistem dan memberikan gambaran yang jelas. Dalam perancangan sistem " sistem managemen kurir paket aliando cargo" menggunakan pendekatan metode waterfall. Metode waterfall adalah metode siklus hidup klasik yang sistematis dan berurutan dalam pengembangan perangkat lunak. Yang dimulai dengan analisis kebutuhan sistem, lalu dilanjutkan dengan tahapan perencanaan, pemodelan, konstruksi, hingga penyerahan sistem kepada pelanggan dengan dukungan perangkat lunak yang telah tersedia.

a) Rancangan Proses dan Kebutuhan data Client

Use case adalah sebuah model yang digunakan untuk menggambarkan alur sistem yang dibuat dan memberikan informasi hubungan antara aktor-aktor yang terlibat. Dalam rancangan UML (Unified Modeling Language) untuk sistem website "sistem managemen kurir paket aliando cargo", terdapat use case yang telah diidentifikasi adalah berikut.



Gambar 3. 7 Use Case Diagram

Pada Gambar 3.7 Use Case Diagram diatas dapat diperoleh kesimpulan bahwa :

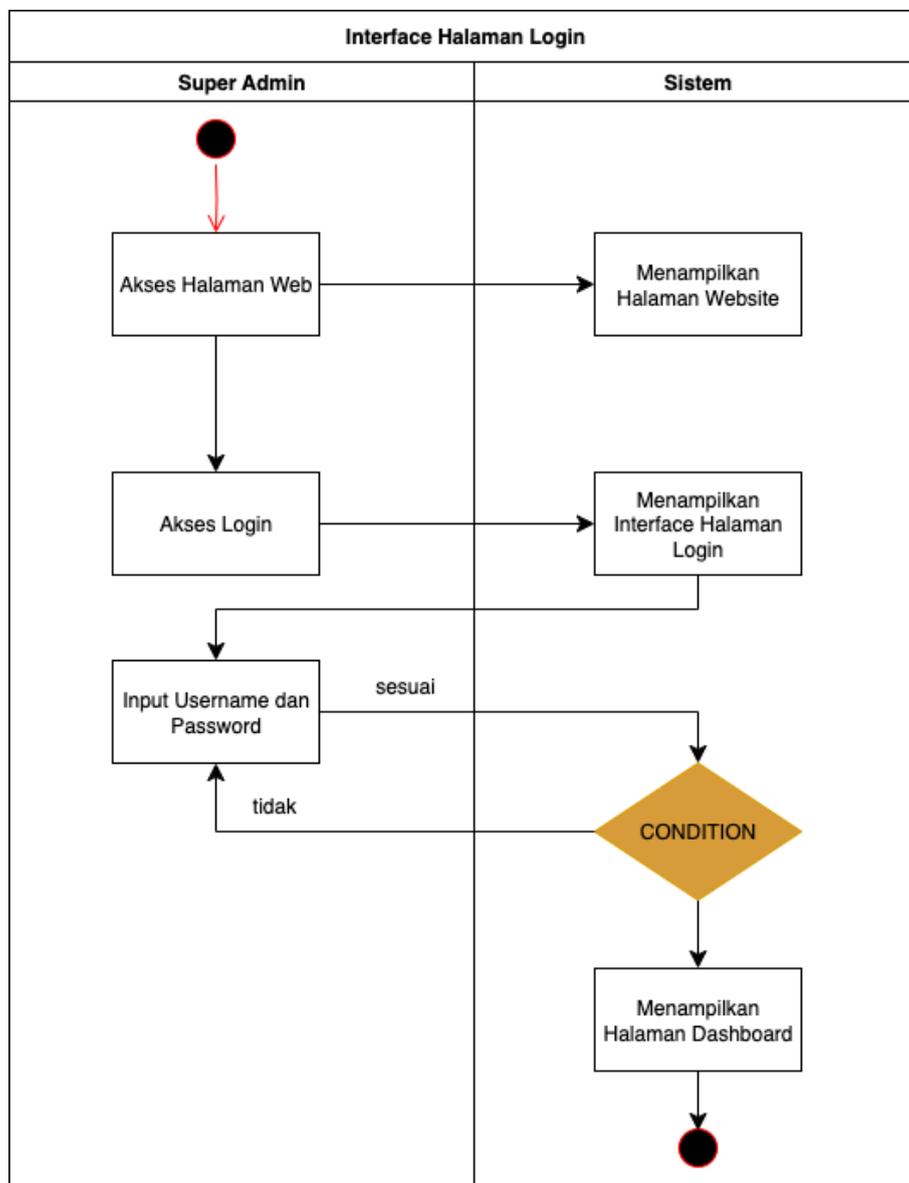
1. Super Admin dapat melakukan Login
2. Super Admin memiliki Dashboard
3. Super Admin mencetak Data Paket
4. Super Admin dapat menghapus, mengedit, dan menambah Data Karyawan
5. Super Admin dapat mengedit dan mencetak Laporan
6. Super Admin dapat mengedit dan mencetak biaya operasional
7. Super Admin dapat menghapus, mengedit, dan menambah Data Vendor
8. Admin dapat melakukan Login
9. Admin memiliki Dashboard
10. Admin dapat mengedit Data Paket
11. Admin menginput Biaya Operasional
12. Admin dapat menghapus, mengedit, dan menambah Data Vendor
13. Tim Operasional memiliki Dashboard
14. Tim Operasional dapat melakukan Login
15. Tim Operasional menambahkan Data Paket

7. Activity Diagram

Diagram aktivitas adalah sebuah gambar atau alur kerja dari use case bisnis dan kejadian dalam use case dalam bentuk grafik. Diagram ini mengilustrasikan langkah-langkah dari alur kerja, orang yang melakukan setiap aktivitas, dan objek-objek yang digunakan dalam proses tersebut.

a) Activity Diagram Login

Proses dibawah merupakan alur activity untuk melakukan login, dimana pengguna akan melakukan akses ke halaman website, lalu menginputkan username dan password , selanjutnya sistem akan memvalidasi jika username dan password sesuai makan akan masuk pada halaman dashboard, jika salah akan menampilkan pesan dan tetap ke halaman login.

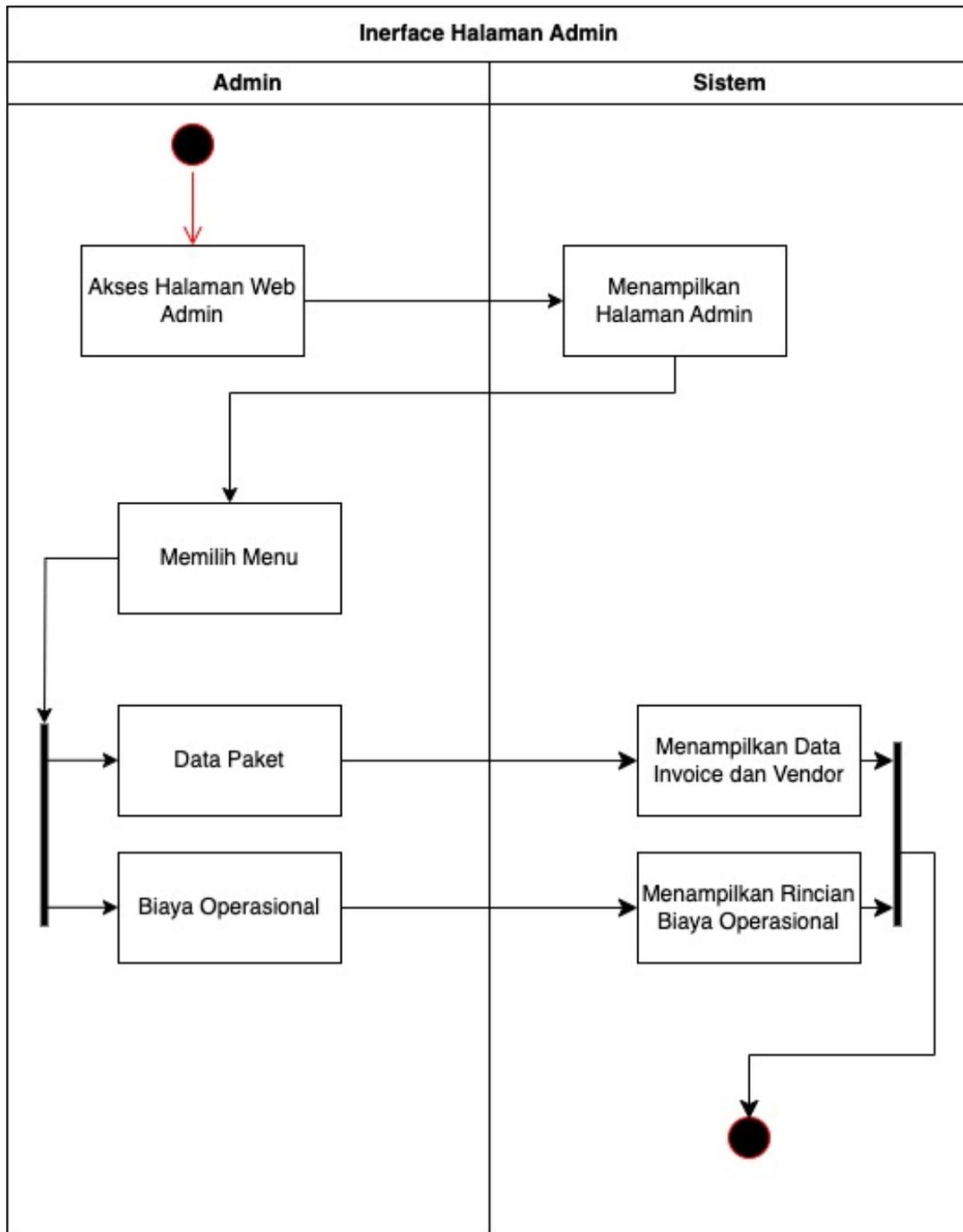


Gambar 3. 8 Activity Diagram login

b) ADMIN

1) Activity Diagram Admin

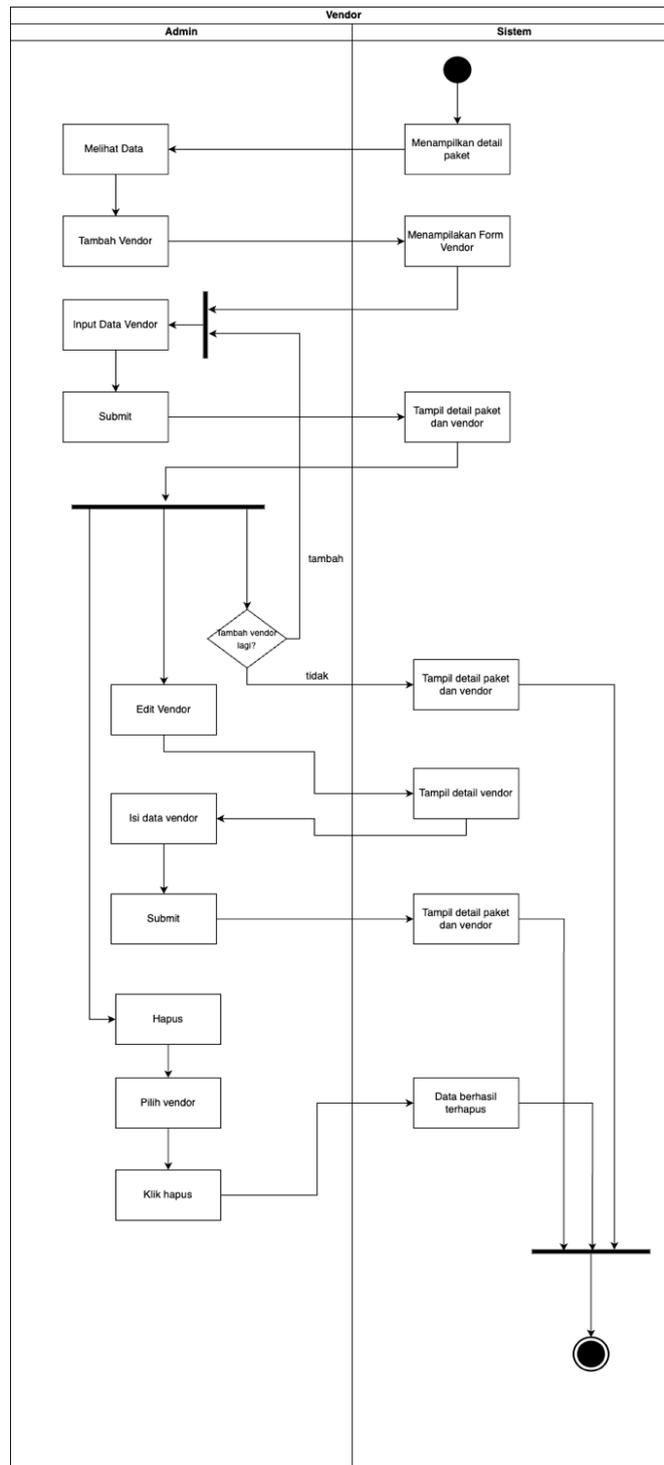
Gambar dibawah merupakan activity ketika setelah login, pada login admin terdapat dua menu yaitu Data Paket dimana admin dapat melihat dan menambahkan data invoice dan vendor. Selanjutnya menu Biaya Operasional di menu tersebut admin dapat menambah dan melihat rincian dari biaya operasional.



Gambar 3. 9 Activity Diagram Setelah Login Admin

2) Activity Diagram Tambah Vendor

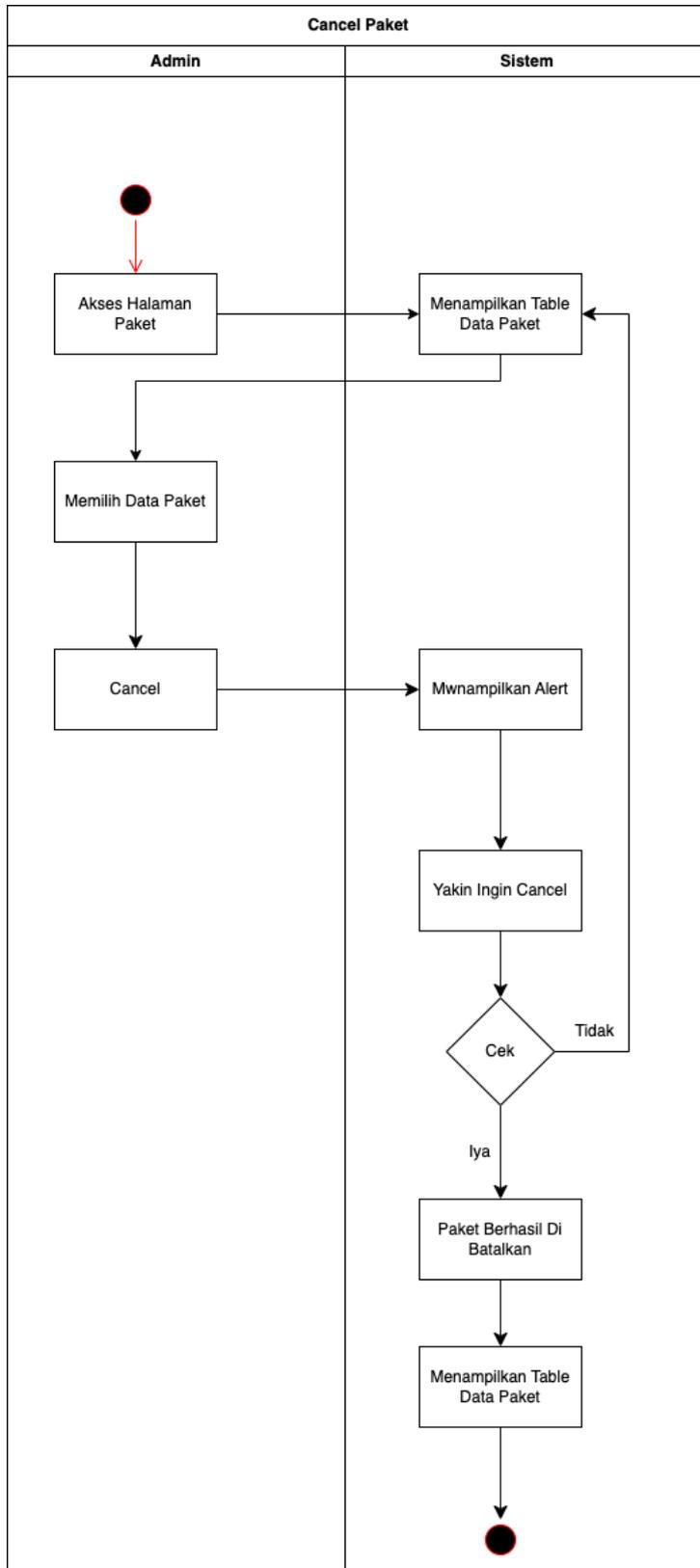
Vendor adalah pihak ke 2 dalam proses pengiriman, setelah barang dikirimkan oleh pihak Alindo Cargo selanjutnya barang akan diterima oleh beberapa vendor tergantung lokasi tujuan paket untuk pengantaran paket selanjutnya. Gambar dibawah adalah proses yang terjadi jika ketika akan menambah, mengedit, dan menghapus data vendor.



Gambar 3. 10 Activity Diagram Vendor

3) Activity Diagram Paket Batal

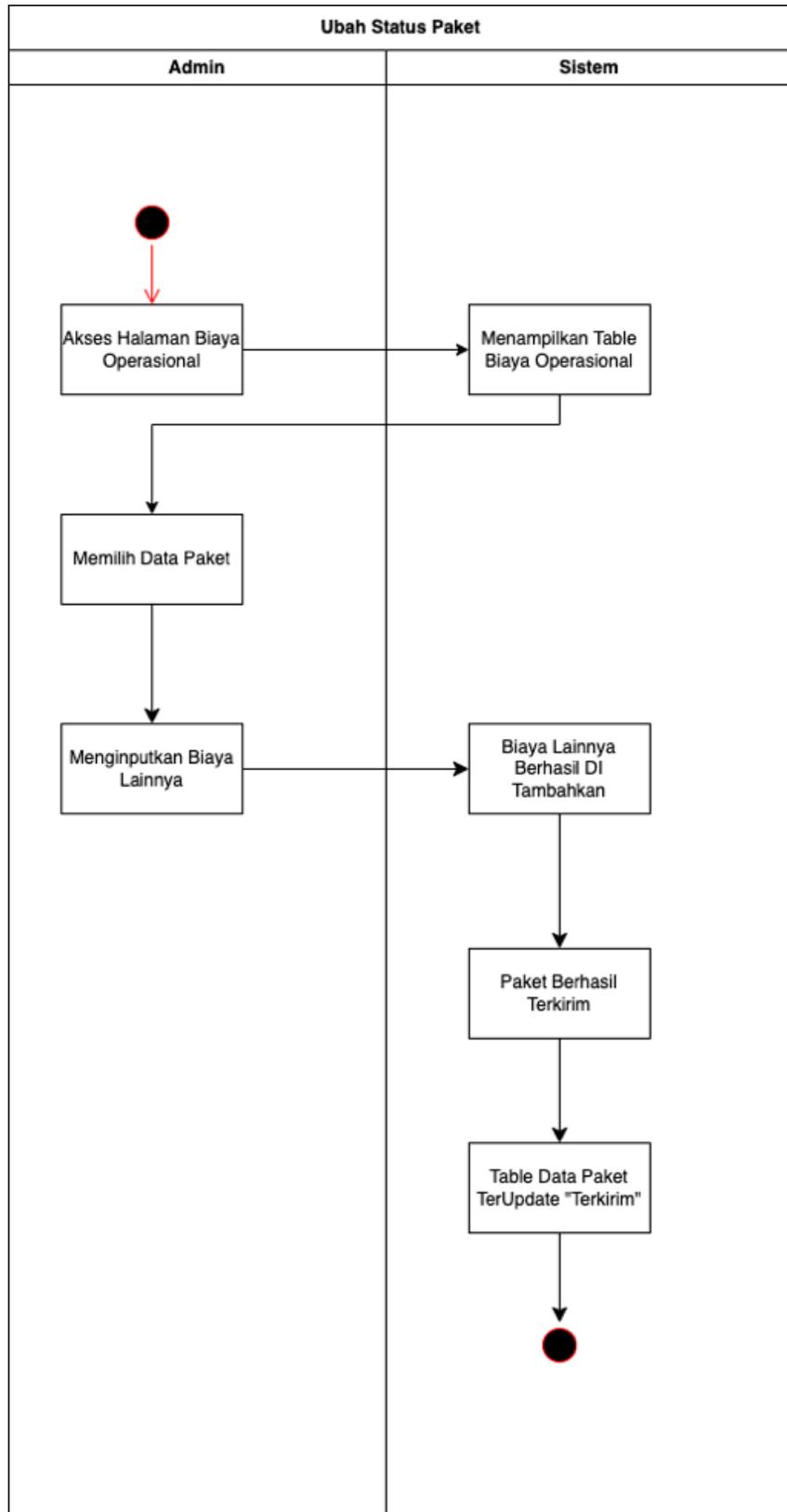
Gambar dibawah merupakan activity diagram jika pelanggan melakukan pembatalan pada order.



Gambar 3. 11 Acitivity Diagram Cancel Paket

4) Activity Diagram Ubah Status Paket

Gambar dibawah merupakan Activity Diagram jika admin akan mengubah status paket.

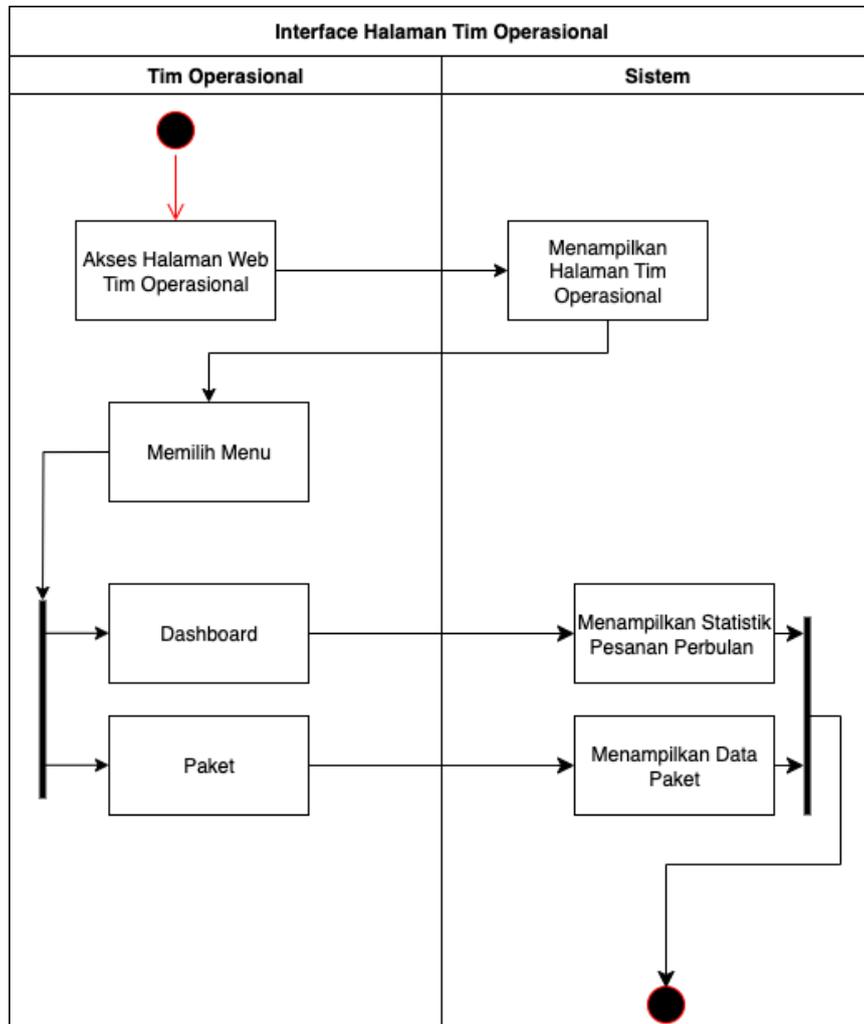


Gambar 3. 12 Activity Diagram Ubah Status Paket

c) TIM OPERASIONAL

1) Activity Diagram Interface Tim Operasional

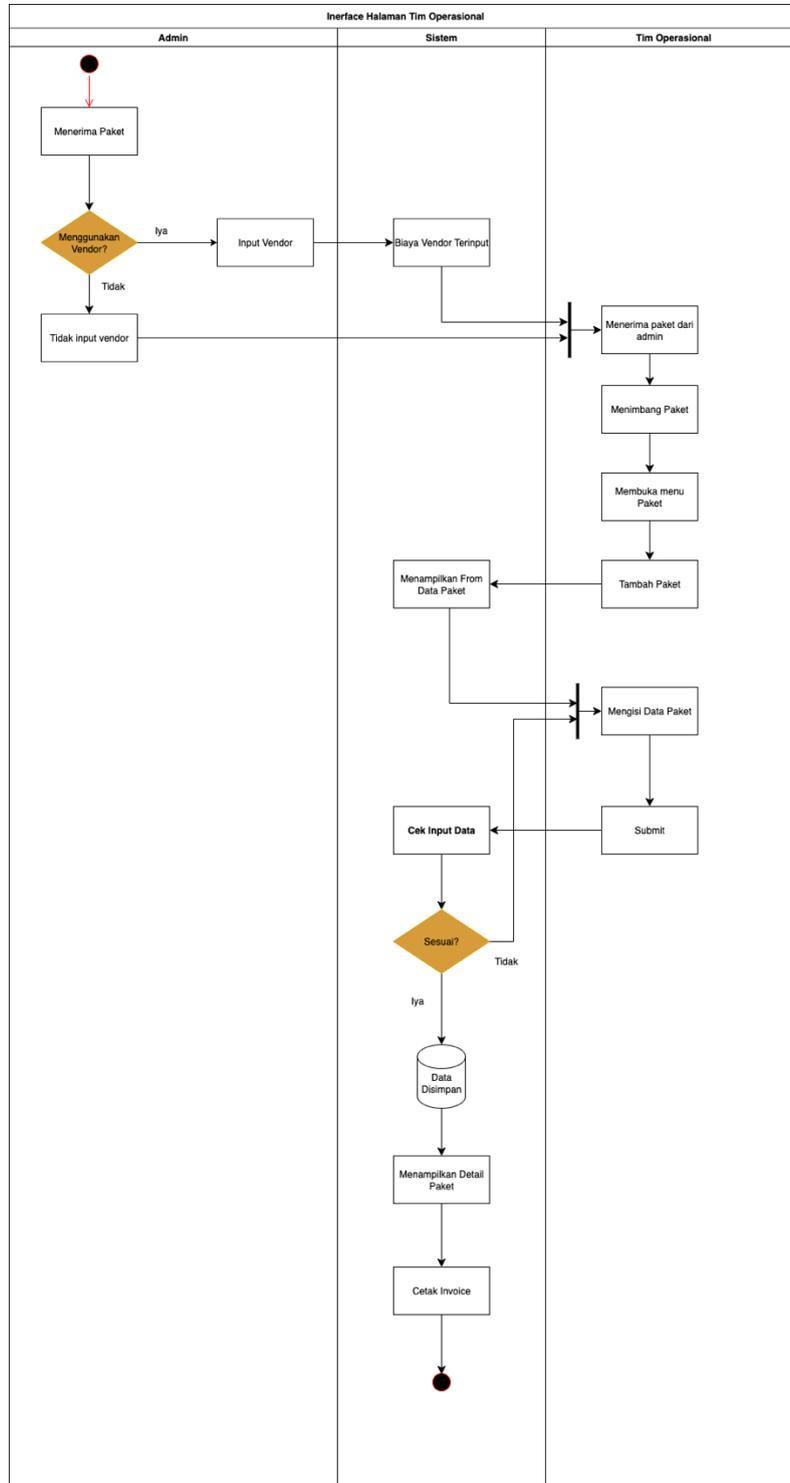
Gambar dibawah merupakan activity ketika setelah login, pada Tim Operasional terdapat dua menu yaitu Dashboar dan paket, pada menu paket Tim Operasional akan menginputkan data paket.



Gambar 3. 13 Activity Diagram Tim Operasional

2) Activity Diagram Input Paket Tim Operasional

Tim Operasional adalah tim yang bertugas untuk menginput invoice data pelanggan dan paket, Gambar dibawah adalah proses yang terjadi saat paket masuk ke dalam sistem dan akan diinputkan oleh tim operasional.

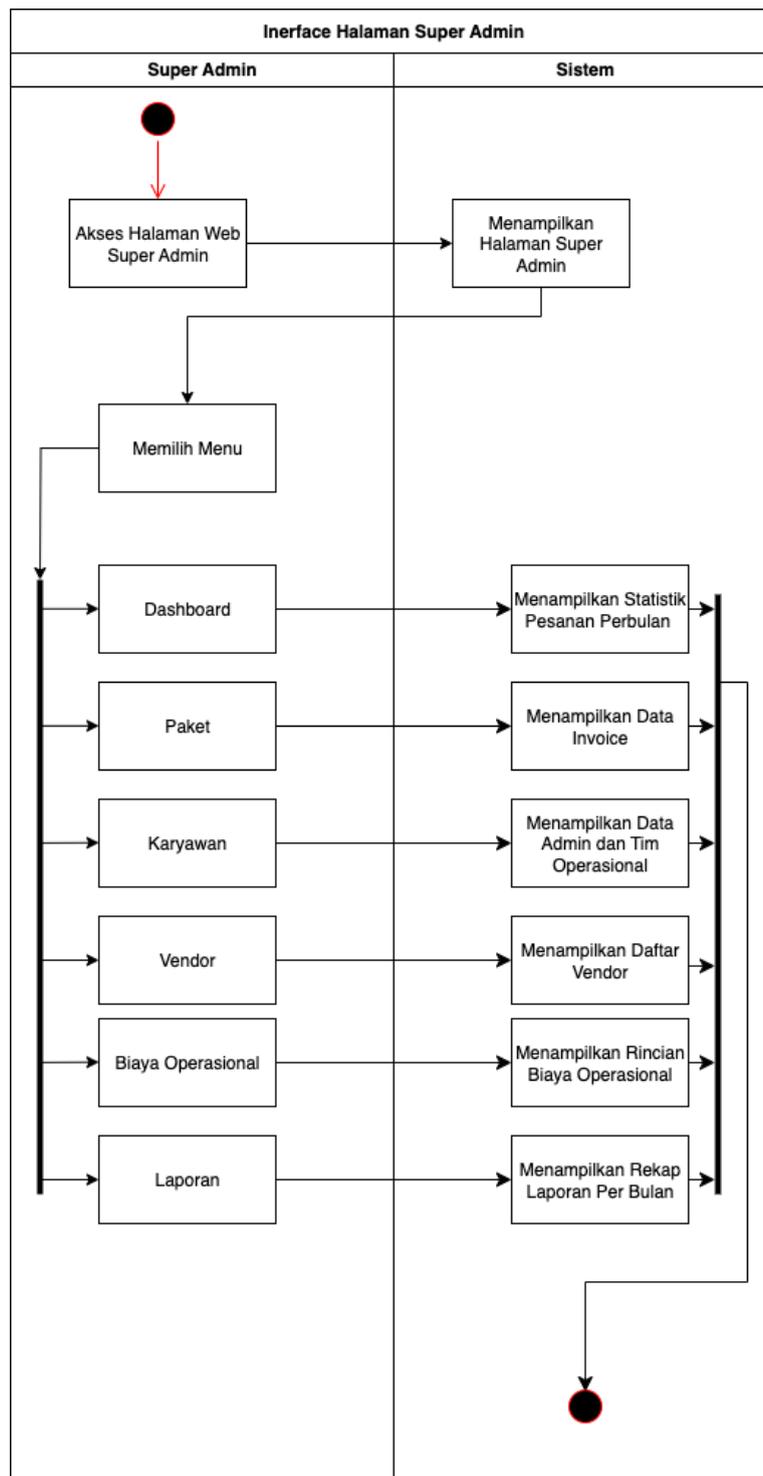


Gambar 3. 14 Activity Diagram Tambah Invoice Tim Operasional

d) SUPER ADMIN

1) Activity Diagram Super Admin

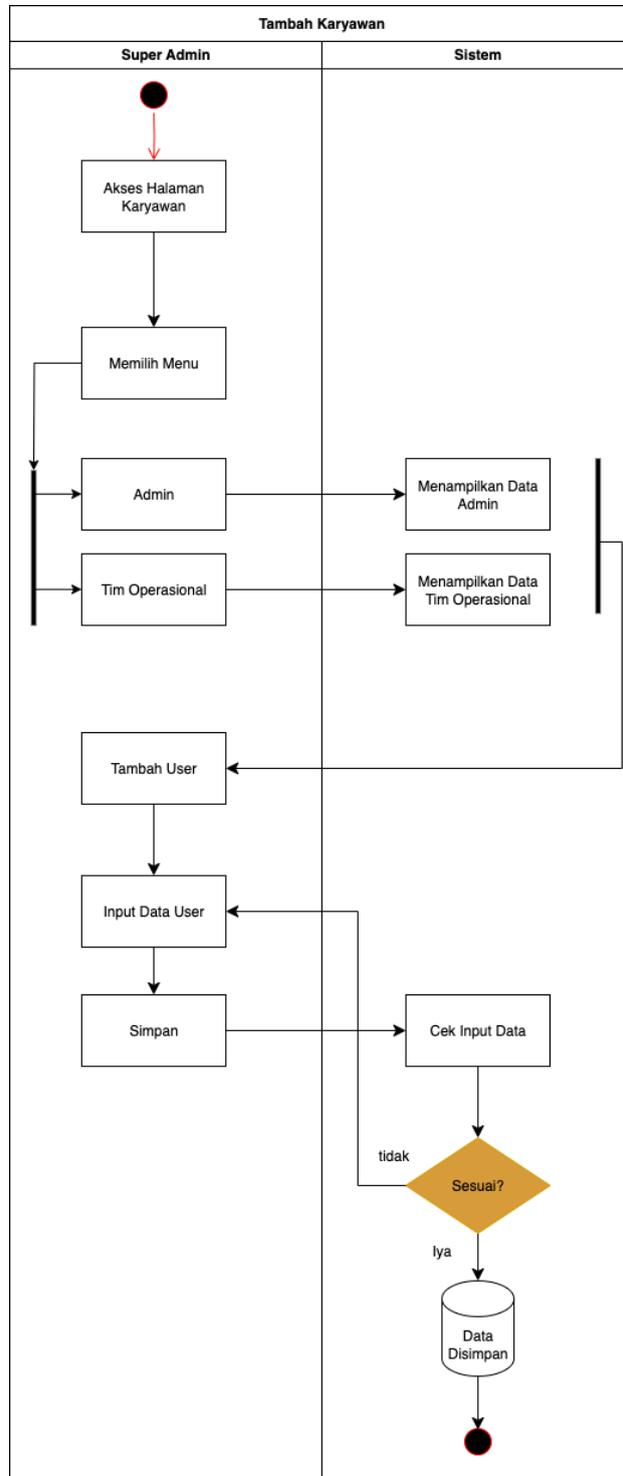
Gambar dibawah merupakan activity ketika setelah login, pada Super Admin terdapat lima menu yaitu Dashboard, Paket, Karyawan, Biaya Operasional, dan Laporan. Super admin hanya bisa menambahkan data Karyawan, dan Vendor.



Gambar 3. 15 Activity Diagram Super Admin

2) Activity Diagram Tambah Karyawan

Gambar dibawah adalah alur jika super admin ingin menambahkan karyawan



Gambar 3. 16 Activity Diagram Tambah Karyawan Super Admin

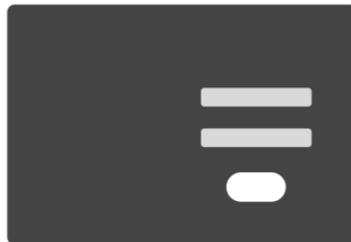
B. Project Manager & Tim Proyek (UI/UX)

1. Membagi jobdesk tiap anggota
2. Membuat jadwal pertemuan dengan client
3. Mendesain struktur navigasi
4. Mendesain komponen UI/UX
5. Membuat Prototipe interaktif
6. Mendesain ikon, warna dll
7. Rancangan Wireframe

Adapun Wireframe sendiri mempunyai berfungsi sebagai blueprint dari sistem, untuk menunjukkan struktur halaman, alur navigasi, dan letak komponen utama tanpa detail desain grafis. Intinya, wireframe lebih menekankan ke *fungsi dan posisi elemen* daripada tampilan visual.

a) Login

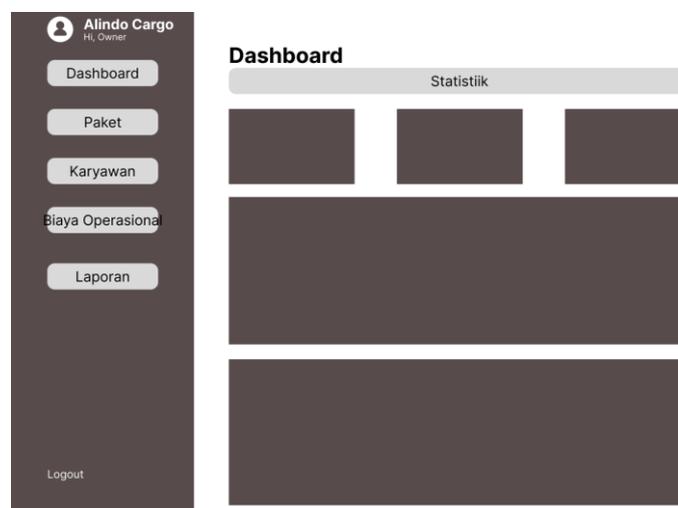
Login adalah halaman awal untuk autentikasi pengguna.



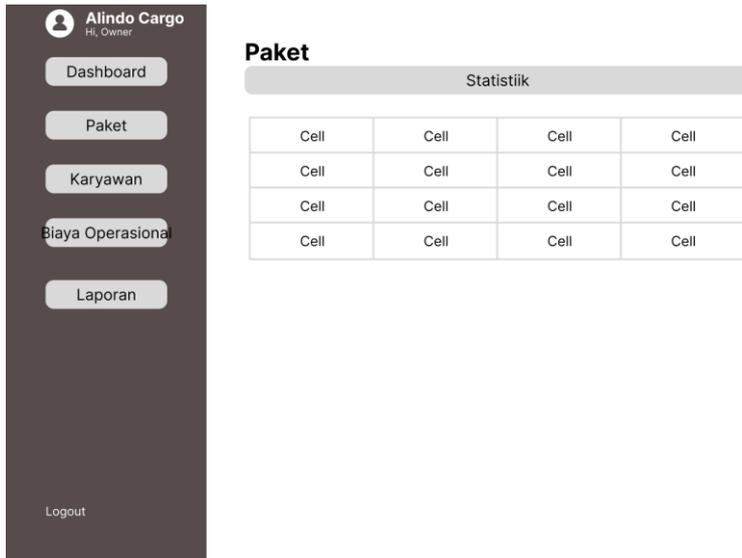
Gambar 3. 17 Wireframe Login

b) Super Admin

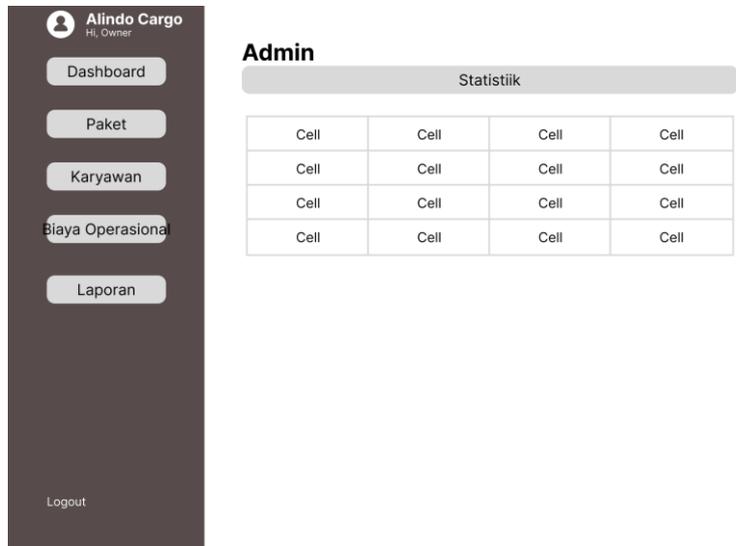
Super Admin memiliki dashboard utama, halaman data paket, manajemen karyawan (admin & tim operasional), biaya operasional, dan laporan.



Gambar 3. 18 Wireframe Dashboard Super Admin



Gambar 3. 19 Wireframe Halaman Paket Super Admin



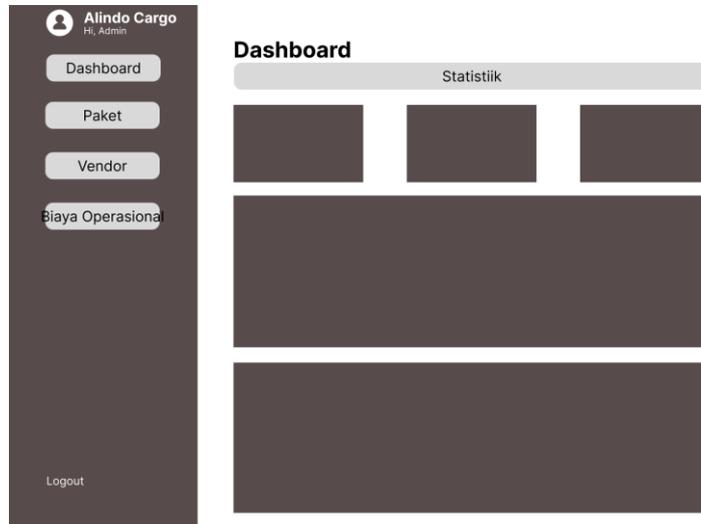
Gambar 3. 20 Wireframe Halaman Karyawan "Admin" Super Admin



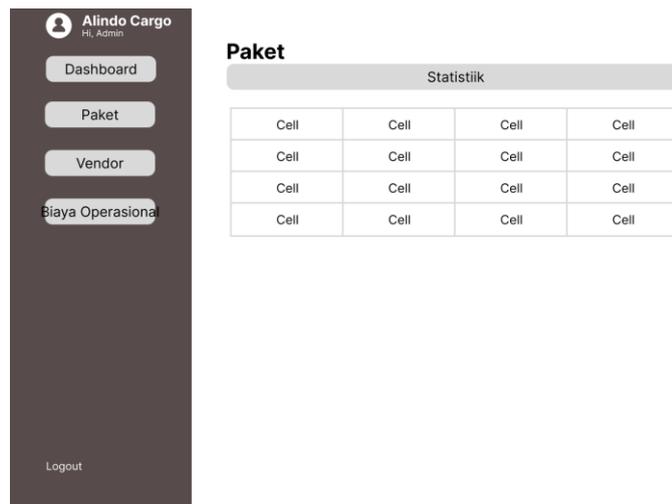
Gambar 3. 21 Wireframe Halaman Karyawan "Tim Operasional" Super Admin

c) Admin

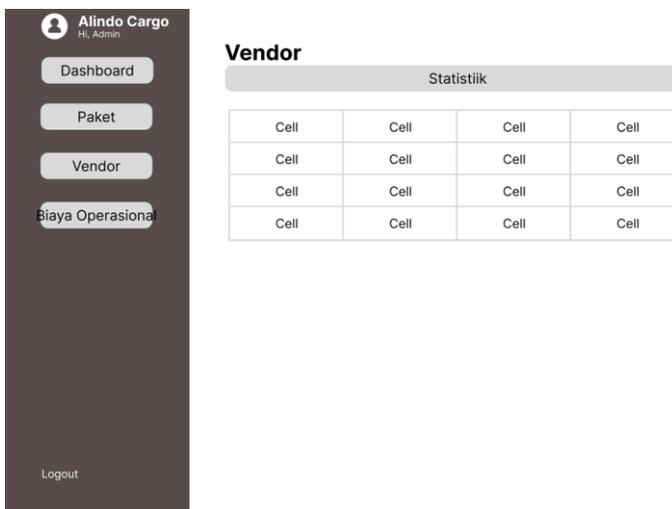
Admin adalah dashboard yang menampilkan data paket, vendor, serta biaya operasional.



Gambar 3. 22 Wireframe Dashboard Admin



Gambar 3. 23 Wireframe Halaman Paket Admin



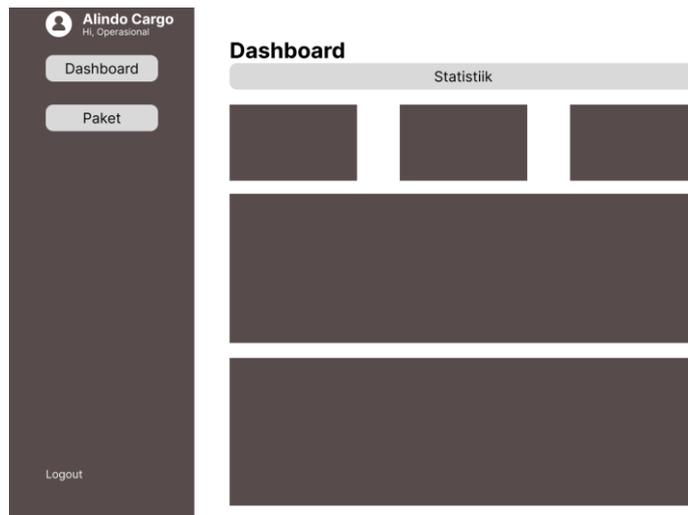
Gambar 3. 24 Wireframe Halaman Vendor Admin



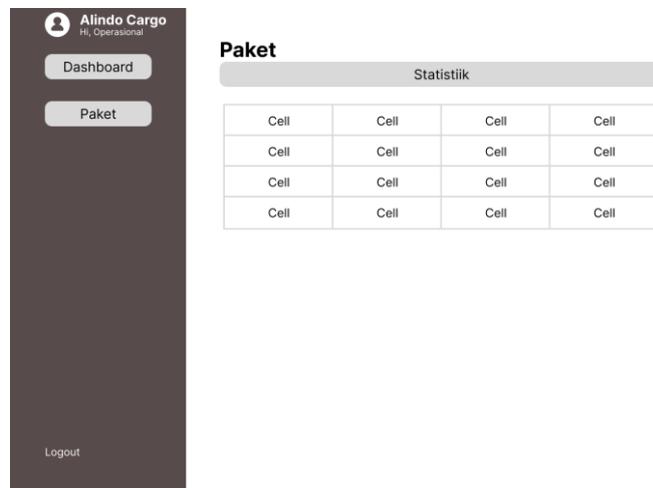
Gambar 3. 25 Wireframe Halaman Biaya Operasional

d) Tim Operasional

Tim Operasional memiliki dashboard dan halaman input paket/invoice.



Gambar 3. 26 Wireframe Dashboard Tim Operasional



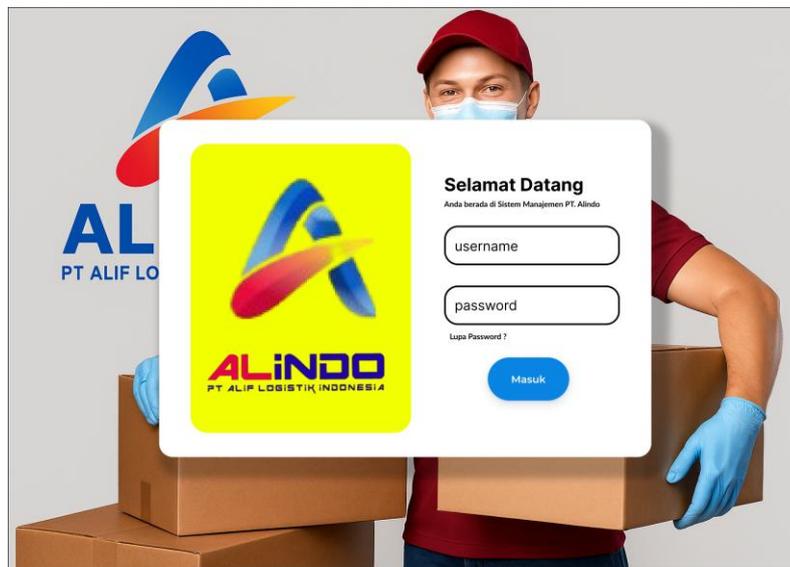
Gambar 3. 27 Wireframe Halaman Paket Tim Operasional

8. Prototype Desain Interface Sistem

Jika wireframe adalah kerangka dasar, maka UI/Interface adalah desain yang sudah memperlihatkan warna, ikon, tombol, tabel data, grafik, hingga alur penggunaan nyata. Interface juga lebih fokus pada *usability* (mudah digunakan), *visual consistency* (konsistensi warna dan layout), serta *responsiveness* (penyesuaian perangkat).

a) Login

Pada UI login memungkinkan pengguna login dengan input email/username dan password.

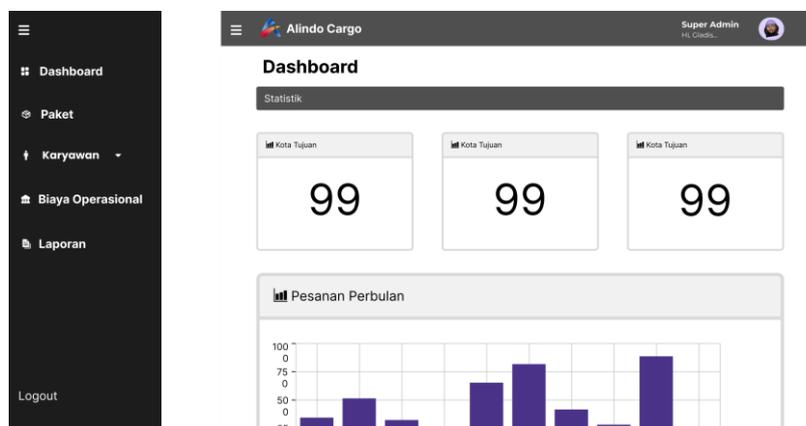


Gambar 3. 28 UI Login

b) Interface Superadmin

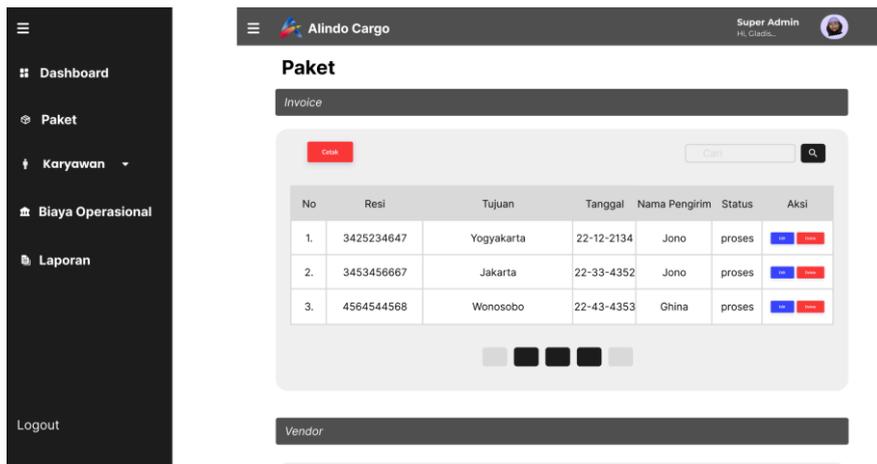
Pada UI Super Admin menampilkan dashboard berisi grafik pendapatan, data paket, manajemen karyawan, biaya operasional, hingga cetak laporan.

1) Dashboard Superadmin



Gambar 3. 29 UI Dashboard Super Admin

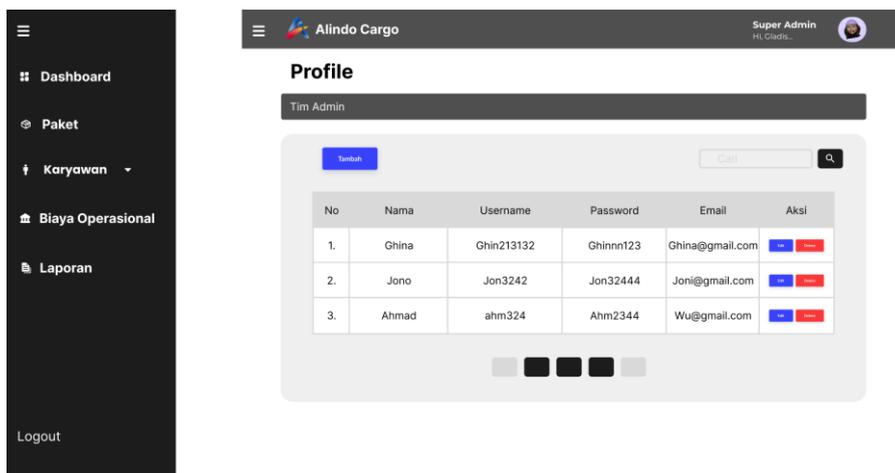
2) Paket



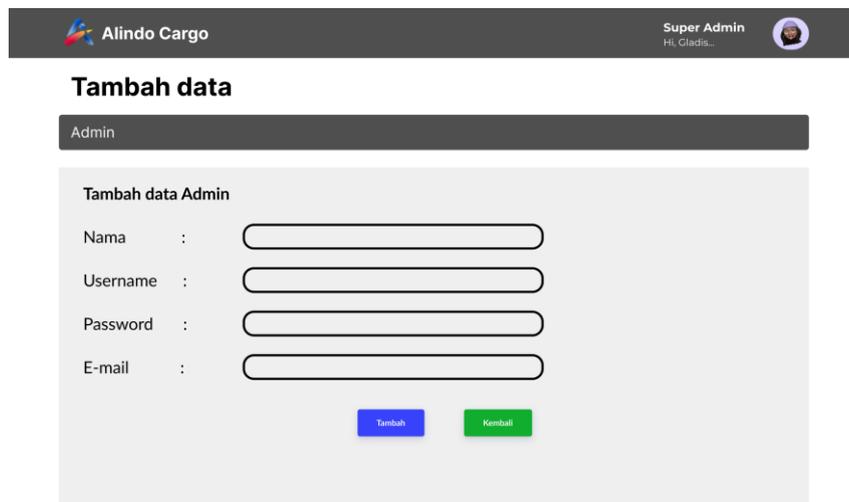
Gambar 3. 30 UI Dashboard Paket

3) Karyawan

a. Admin



Gambar 3. 31 UI Dashboard Karyawan Admin



Gambar 3. 32 UI Dashboard Tambah Admin

b. Tim Operasional

Profile
Tim Operasional

Tambah

No	Nama	Username	Asal Daerah	Email	Aksi
1.	Ghina	Ghin213132	Ghin123	Ghina@gmail.com	Edit Hapus
2.	Jono	Jon3242	Jon32444	Joni@gmail.com	Edit Hapus
3.	Ahmad	ahm324	Ahm2344	Wu@gmail.com	Edit Hapus

Gambar 3. 33 UI Dashboard Karyawan Tim Operasional

Tambah data
Tim Operasional

Tambah data Tim Operasional

Nama :

Username :

Password :

E-mail :

Tambah Kembali

Gambar 3. 34 UI Dashboard Tambah Tim Operasional

4) Biaya Operasional

Biaya Operasional
Biaya Operasional

Tambah

No	Resi	Pengemasan	Bahan Bakar	Vendor	Total	Aksi
1.	3425234647	50000	40000	60000	1500000	Edit Hapus
2.	3453456667	40000	40000	58000	1380000	Edit Hapus
3.	4564544568	20000	50000	87000	1570000	Edit Hapus

Gambar 3. 35 UI Dashboard Biaya Operasional

Alindo Cargo Super Admin
Hi, Gladis...

Edit

Edit Biaya Operasional

Resi

Nama Vendor

Kota Vendor

Biaya Vendor

Total Biaya

Edit Kembali

Gambar 3. 36 UI Dashboard Edit Biaya Operasional

5) Laporan

Alindo Cargo Super Admin
Hi, Gladis...

Laporan

Laporan

Cari

No	Rekap Laporan	Aksi
1.	Mei - 2025	Lihat
2.	Juli - 2025	Lihat
3.	Juni - 2025	Lihat

Logout

Gambar 3. 37 UI Dashboard Laporan

Alindo Cargo Super Admin
Hi, Gladis...

Laporan

Laporan

Cetak

No	Resi	Tujuan	Tanggal	Nama Pengirim	Status	Total Biaya
1.	3425234647	Yogyakarta	22-12-2134	Jono	proses	130000
2.	3453456667	Jakarta	22-33-4352	Jono	proses	130000
3.	4564544568	Wonosobo	22-43-4353	Ghina	proses	130000

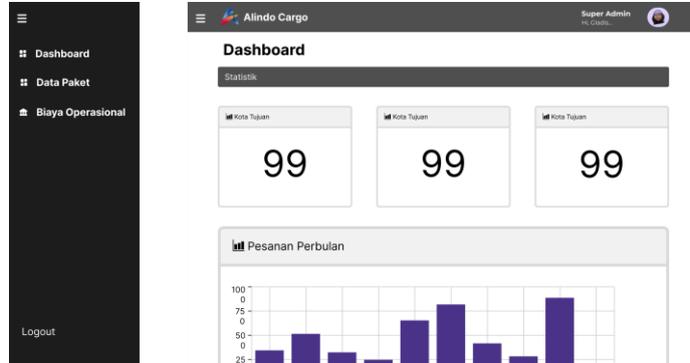
Kembali

Gambar 3. 38 UI Dashboard Cetak Laporan

c) Interface Admin

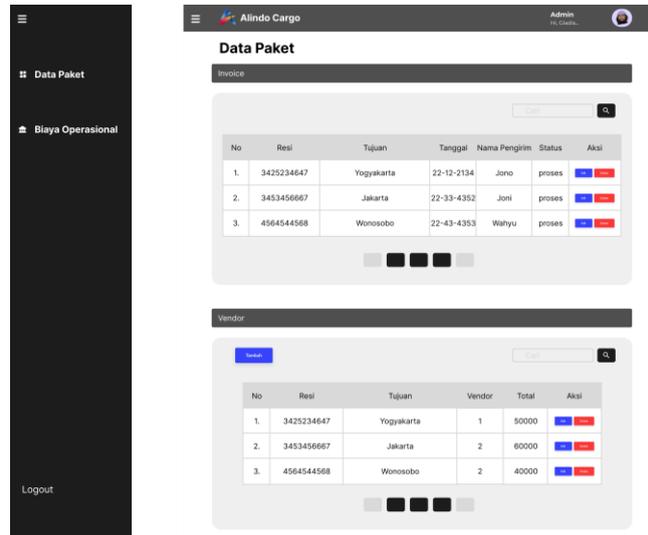
UI Admin menampilkan halaman dashboard, data paket (bisa edit invoice), vendor, dan input biaya operasional.

1) Halaman Dashboard

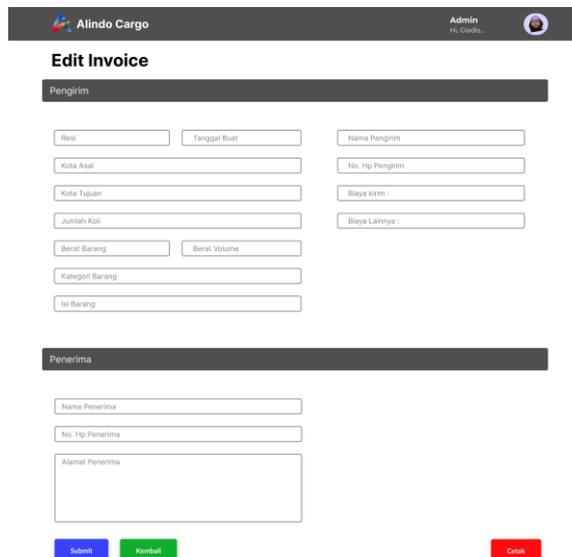


Gambar 3. 39 UI Dashboard Admin

2) Data paket



Gambar 3. 40 UI Dashboard Data Paket



Gambar 3. 41 UI Dashboard Edit Invoice

Vendor

Input Vendor

Resi

Tambah

Nama Vendor

Kota Vendor

Biaya Vendor

Total Biaya

Submit Kembali

Gambar 3. 42 UI Dashboard Tambah Vendor

3) Biaya Operasional

Biaya Operasional

Biaya Operasional

Tambah

Cari

No	Resi	Pengemasan	Bahan Bakar	Vendor	Total	Aksi
1.	3425234647	50000	40000	60000	1500000	Edit Hapus
2.	3453456667	40000	40000	58000	1380000	Edit Hapus
3.	4564544568	20000	50000	87000	1570000	Edit Hapus

Logout

Gambar 3. 43 UI Dashboard Biaya Operasional

Biaya Operasional

Input Biaya Operasional

Resi

Pengemasan

Bahan Bakar

Biaya Vendor

Total Biaya

Tambah Kembali

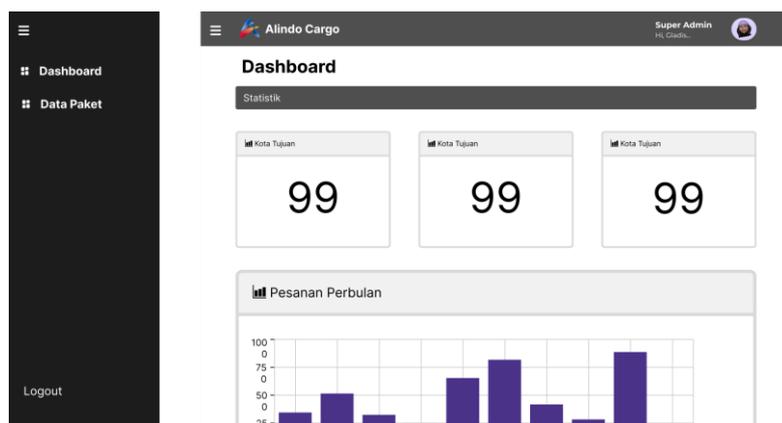
Gambar 3. 44 UI Dashboard Input Biaya Operasional

Gambar 3. 45 UI Dashboard Edit Biaya Operasional

d) Interface Tim Operasional

UI Tim Operasional memiliki dashboard sederhana untuk input invoice paket.

1) Dashboard Tim Operasional



Gambar 3. 46 UI Dashboard Tim Operasional

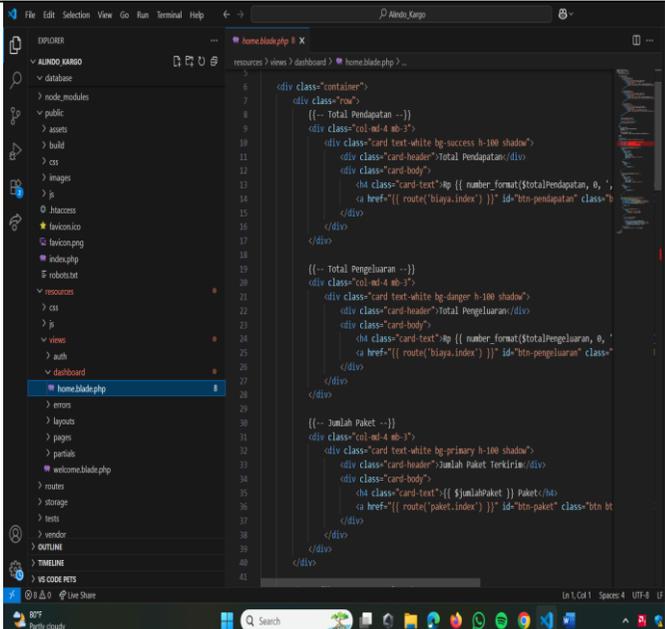
2) Invoice

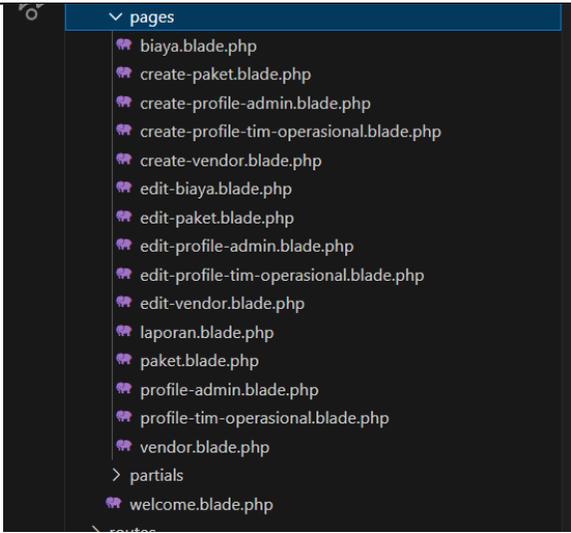
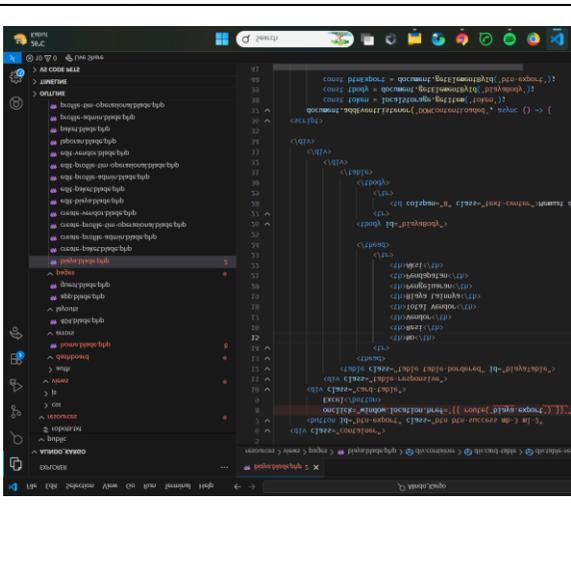
Gambar 3. 47 UI Dashboard Input Invoice

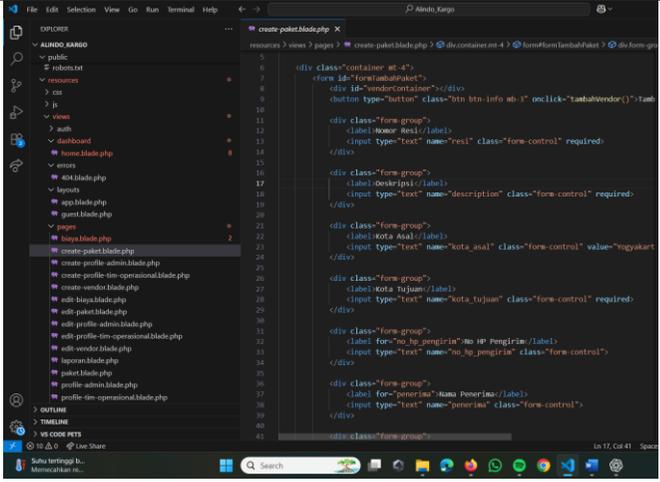
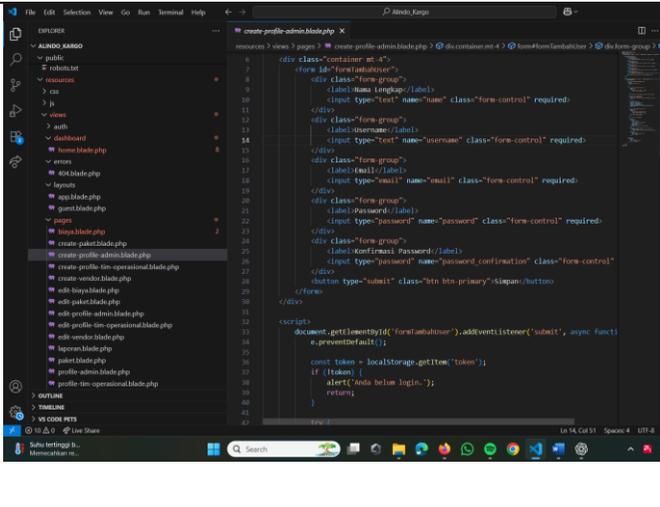
C. Tim Proyek Front End

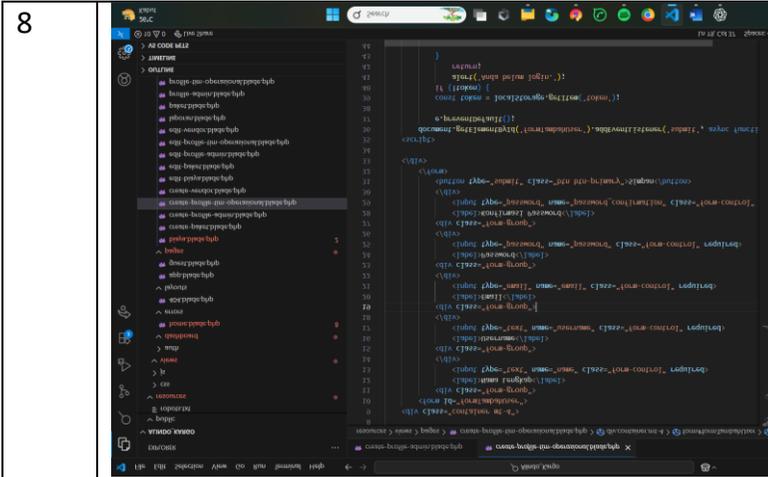
1. Membuat halaman pengguna
2. Menerapkan navigasi halaman
3. Membuat tampilan dashboard
4. Lampiran pekerjaan

Tabel 3. 4 Lampiran Kode Program Front-End

No.	Foto	Keterangan
1.		<p>Home.blade.php Berfungsi sebagai halaman dashboard utama dari sistem manajemen kargo. Ada 3 fungsi utama yaitu untuk mengarahkan ke halaman biaya.index selanjutnya untuk melihat detail pengeluaran pada biaya.index Dan yang terakhir untuk menunjukkan data jumlah paket ditampilkan dalam bentuk bilangan bulat pada paket.index Adapun fungsi lain untuk mengarahkan atau memindahkan laman dari home ke biaya.index dan paket.index</p>

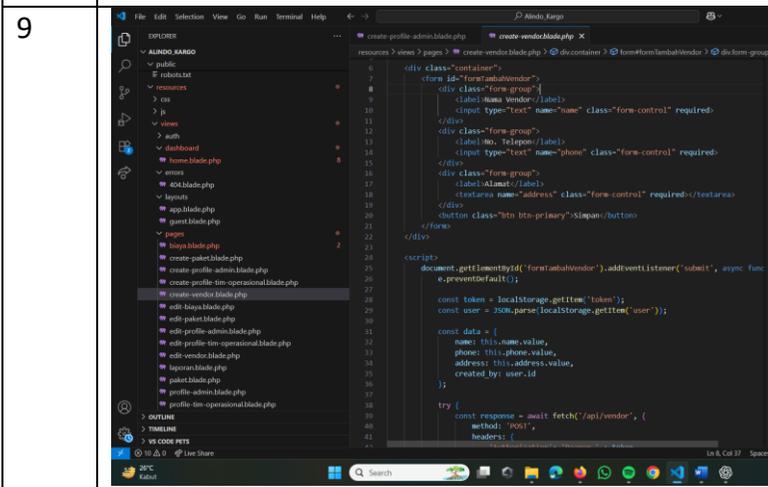
<p>4.</p>		<p>Gambar disamping menunjukkan page apa saja yang ada dalam web.</p>
<p>5.</p>		<p>Halaman ini adalah tampilan daftar Biaya Operasional dalam sistem Laravel yang menggunakan Blade dan JavaScript. Layout-nya mewarisi <i>layouts.app</i>, dengan judul "Biaya Operasional" dan tombol Export Excel yang hanya terlihat oleh pengguna dengan role <i>super-admin</i>.</p> <p>Data ditampilkan dalam tabel interaktif berisi kolom seperti Resi, Vendor, Biaya, Pengeluaran, dan Pendapatan. JavaScript mengambil token dari localStorage, lalu fetch data user dan data biaya dari API. Jika data ditemukan, tabel diisi otomatis dan dihitung total pengeluaran serta pendapatan tiap entri.</p> <p>Tabel ini kemudian diaktifkan dengan plugin DataTables untuk fitur pencarian dan pagination. Halaman ini mempermudah admin dalam mengelola dan meninjau biaya operasional secara dinamis dan efisien.</p>

6	 <pre> 6 <div class="container mt-4"> 7 <form id="formTambahpaket"> 8 <div class="form-group"> 9 <input type="text" name="resi" class="form-control" required> 10 </div> 11 <div class="form-group"> 12 <input type="text" name="description" class="form-control" required> 13 </div> 14 <div class="form-group"> 15 <input type="text" name="kota_asal" class="form-control" value="nyagakari"> 16 </div> 17 <div class="form-group"> 18 <input type="text" name="kota_tujuan" class="form-control" required> 19 </div> 20 <div class="form-group"> 21 <input type="text" name="no_hp_pengirim" class="form-control"> 22 </div> 23 <div class="form-group"> 24 <input type="text" name="nama_penerima" class="form-control"> 25 </div> 26 <div class="form-group"> 27 <input type="text" name="penerima" class="form-control"> 28 </div> 29 </form> 30 </div> 31 </div> 32 </div> 33 </div> 34 </div> 35 </div> 36 </div> 37 </div> 38 </div> 39 </div> 40 </div> 41 </div> </pre>	<p>Halaman ini adalah form untuk menambahkan data paket pada sistem Laravel. Pengguna dapat mengisi informasi seperti nomor resi, deskripsi, kota tujuan, data pengirim dan penerima, serta menambahkan beberapa vendor dan biaya mereka secara dinamis. Tombol “Cetak Invoice” digunakan untuk memverifikasi data dan wajib diklik sebelum tombol “Simpan” bisa diaktifkan. Setelah invoice dicetak, data akan dikirim ke server melalui API. Jika berhasil, sistem akan menampilkan notifikasi sukses dan mengarahkan ke halaman daftar paket. Semua proses menggunakan token autentikasi dan data user dari API.</p>
7	 <pre> 6 <div class="container mt-4"> 7 <form id="formTambahuser"> 8 <div class="form-group"> 9 <input type="text" name="name" class="form-control" required> 10 </div> 11 <div class="form-group"> 12 <input type="text" name="username" class="form-control" required> 13 </div> 14 <div class="form-group"> 15 <input type="text" name="email" class="form-control" required> 16 </div> 17 <div class="form-group"> 18 <input type="text" name="password" class="form-control" required> 19 </div> 20 <div class="form-group"> 21 <input type="text" name="password_confirmation" class="form-control" required> 22 </div> 23 <div class="form-group"> 24 <input type="submit" value="Simpan" class="btn btn-primary"> 25 </div> 26 </form> 27 </div> 28 </div> 29 </div> 30 </div> 31 </div> 32 </div> 33 </div> 34 </div> 35 </div> 36 </div> 37 </div> 38 </div> 39 </div> 40 </div> 41 </div> </pre>	<p>Halaman ini adalah form untuk menambahkan user admin baru dalam sistem Laravel. Pengguna diminta mengisi data seperti nama lengkap, username, email, password, dan konfirmasi password. Ketika form dikirim, sistem terlebih dahulu memverifikasi token dan mengecek apakah pengguna saat ini adalah super-admin. Jika bukan, maka akses ditolak.</p> <p>Jika valid, data akan dikirim ke endpoint <code>/api/register</code> menggunakan metode POST dengan role ditetapkan sebagai "admin". Jika pendaftaran berhasil, akan muncul notifikasi sukses dan pengguna diarahkan ke halaman daftar admin. Jika gagal, sistem menampilkan pesan error menggunakan SweetAlert.</p>



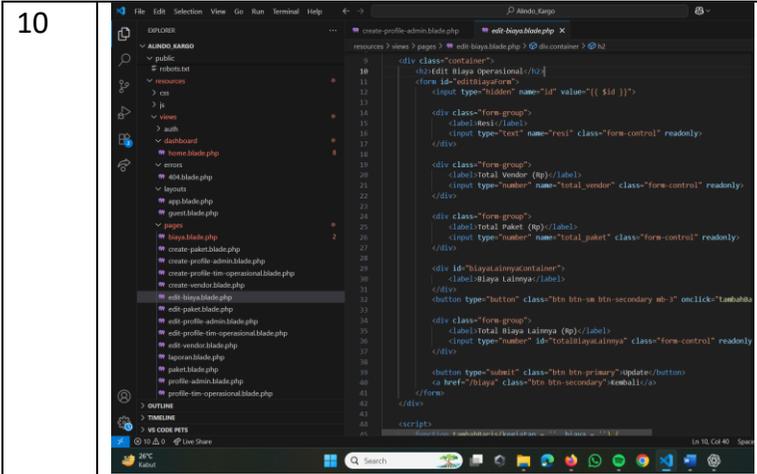
Halaman ini merupakan form tambah user untuk Tim Operasional pada sistem Laravel. Pengguna diminta mengisi data seperti nama lengkap, username, email, password, dan konfirmasi password. Saat form disubmit, sistem terlebih dahulu mengecek token login dan memastikan hanya pengguna dengan role super-admin yang dapat menambahkan akun.

Jika valid, data akan dikirim ke endpoint `/api/register` dengan role otomatis ditetapkan sebagai tim-operasional. Jika proses berhasil, akan muncul notifikasi sukses, lalu pengguna diarahkan ke halaman daftar tim operasional. Jika gagal, sistem akan menampilkan pesan kesalahan menggunakan SweetAlert.



Halaman ini adalah form tambah vendor dalam sistem Laravel. Pengguna diminta mengisi nama vendor, nomor telepon, dan alamat. Saat form disubmit, JavaScript akan mengambil token dan data user dari localStorage, lalu mengirim data ke endpoint `/api/vendor` melalui metode POST.

Jika berhasil, akan muncul notifikasi sukses dan pengguna diarahkan ke halaman daftar vendor. Jika terjadi kesalahan, sistem akan menampilkan pesan error menggunakan SweetAlert. Seluruh proses diautentikasi menggunakan token untuk memastikan hanya pengguna terdaftar yang bisa menambahkan data vendor.

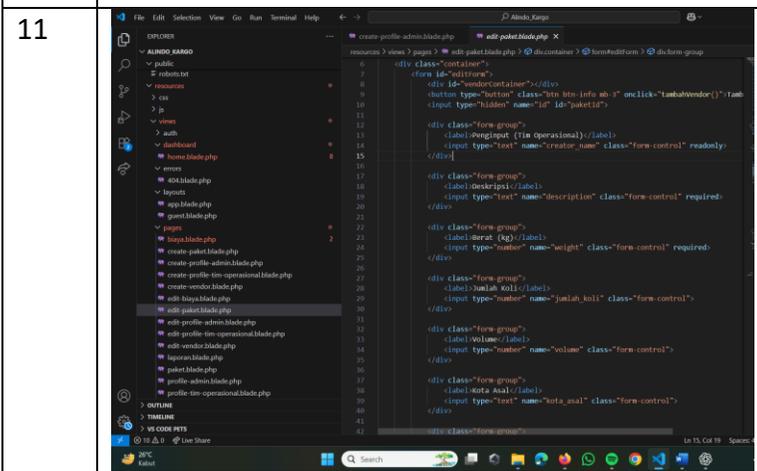


10

Halaman ini adalah form edit biaya operasional dalam sistem Laravel. Pengguna dapat melihat informasi resi, total vendor, dan total paket secara readonly, serta dapat menambahkan atau mengubah biaya lainnya melalui input dinamis.

Saat halaman dimuat, data biaya dari API /api/biaya/{id} akan diambil, kemudian ditampilkan ke dalam form. Pengguna bisa menambah baris biaya lain (berisi nama kegiatan dan jumlah biaya), lalu sistem otomatis menghitung total biaya lainnya.

Saat tombol Update diklik, data akan dikirim ke server melalui metode PUT. Jika berhasil, muncul notifikasi sukses dan pengguna diarahkan ke halaman daftar biaya. Jika gagal, error akan ditampilkan menggunakan SweetAlert. Semua proses ini memakai autentikasi token dan data user dari localStorage.



11

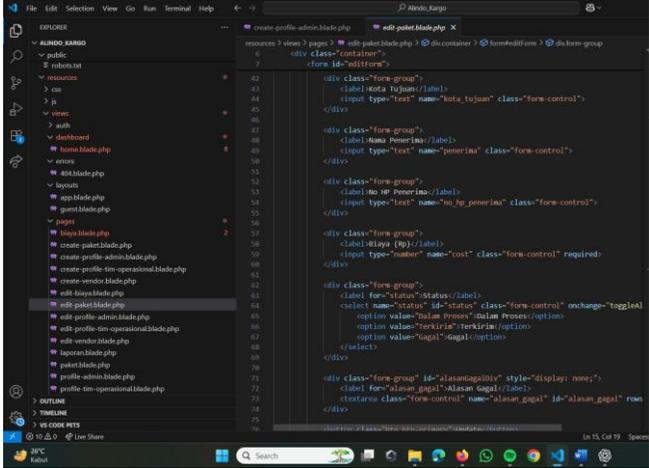
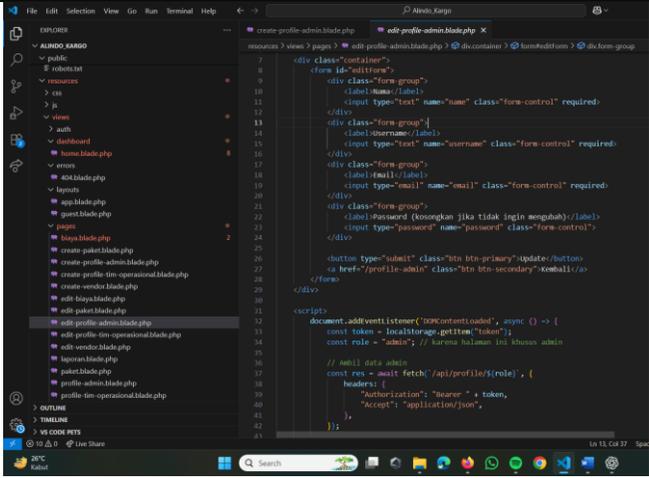
Halaman ini merupakan form edit paket dalam sistem Laravel yang memungkinkan pengguna memperbarui data paket pengiriman. Data yang dapat diedit meliputi deskripsi, berat, jumlah koli, volume, kota asal dan tujuan, data penerima, biaya pengiriman, vendor yang terlibat beserta biayanya, serta status pengiriman.

Saat halaman dimuat, data paket dan daftar vendor diambil dari API dan ditampilkan secara otomatis ke dalam form. Jika paket sebelumnya sudah memiliki vendor, data tersebut juga akan dimuat dan ditampilkan dalam form dinamis.

Form ini juga menyertakan dropdown status dengan tiga opsi: Dalam Proses, Terkirim, dan Gagal. Jika dipilih "Gagal", akan muncul kolom tambahan untuk mengisi alasan kegagalan.

Hak akses diatur berdasarkan role:

- Admin dapat mengubah status dan mengisi alasan gagal.
- Superadmin hanya bisa melihat, tidak bisa mengubah status.

	<p>Saat tombol Update ditekan, data akan dikirim ke endpoint API menggunakan metode PUT. Jika berhasil, akan muncul notifikasi sukses dan pengguna diarahkan ke halaman daftar paket. Jika gagal, akan muncul pesan error menggunakan SweetAlert. Semua proses terautentikasi melalui token dan validasi role user dari API.</p>
<p>12</p> 	<p>Halaman ini merupakan form edit profil untuk Admin dalam sistem Laravel. Saat halaman dimuat, sistem akan mengambil data profil admin dari endpoint <code>/api/profile/admin</code> menggunakan token autentikasi, lalu mengisi field name, username, dan email ke dalam form.</p> <p>Pengguna bisa memperbarui informasi tersebut, dan jika ingin mengganti password, cukup mengisi kolom password. Jika kolom password dikosongkan, maka password lama tetap dipertahankan.</p> <p>Setelah tombol Update diklik, data akan dikirim ke server menggunakan metode PUT. Jika update berhasil, akan muncul notifikasi sukses dan pengguna diarahkan kembali ke halaman profil admin. Jika gagal, sistem akan menampilkan pesan error dengan SweetAlert. Halaman ini ditujukan khusus untuk admin dan hanya bisa diakses jika token valid.</p>

13

```

11 <div class="container">
12 <form id="editform">
13 <div class="form-group">
14 <label>Nama /label>
15 <input type="text" name="name" class="form-control" required>
16 </div>
17 <div class="form-group">
18 <label>Username /label>
19 <input type="text" name="username" class="form-control" required>
20 </div>
21 <div class="form-group">
22 <label>Email /label>
23 <input type="email" name="email" class="form-control" required>
24 </div>
25 <div class="form-group">
26 <label>Password (kosongkan jika tidak ingin mengubah) /label>
27 <input type="password" name="password" class="form-control">
28 </div>
29 <button type="submit" class="btn btn-primary">Update /button>
30 <a href="/profile/tim-operasional" class="btn btn-secondary">Kembali /a>
31 </form>
32 </div>
33 <script>
34 document.addEventListener("DOMContentLoaded", async () => {
35   const token = localStorage.getItem("token");
36   const role = "tim-operasional"; // karena halaman ini khusus tim-operasional
37
38   // ambil data tim-operasional
39   const res = await fetch(`/api/profile/${role}`, {
40     headers: {
41       "Authorization": "bearer " + token,
42       "Accept": "application/json",
43     },
44   });
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Halaman ini adalah form edit profil untuk pengguna dengan role Tim Operasional dalam sistem Laravel. Saat halaman dimuat, data profil pengguna diambil dari endpoint /api/profile/tim-operasional menggunakan token yang tersimpan di localStorage. Data seperti nama, username, dan email otomatis diisi ke dalam form.

Pengguna bisa memperbarui data tersebut. Jika ingin mengganti password, cukup mengisi kolom password; jika dikosongkan, maka password lama tetap digunakan.

Setelah form dikirim, data akan dikirim ke server dengan metode PUT. Jika berhasil, akan muncul notifikasi sukses dan pengguna akan diarahkan ke halaman profil Tim Operasional. Jika gagal, akan muncul notifikasi error menggunakan SweetAlert. Halaman ini hanya ditujukan untuk pengguna dengan role tim-operasional dan memerlukan token autentikasi yang valid.

14

```

6 <div class="container">
7 <form id="formEditVendor">
8 <div class="form-group">
9 <label>Nama Vendor /label>
10 <input type="text" name="name" class="form-control" required>
11 </div>
12 <div class="form-group">
13 <label>No. Telepon /label>
14 <input type="text" name="phone" class="form-control" required>
15 </div>
16 <div class="form-group">
17 <label>Alamat /label>
18 <textarea name="address" class="form-control" required></textarea>
19 </div>
20 <div class="form-group">
21 <button class="btn btn-primary">Update /button>
22 <a href="/vendor" class="btn btn-secondary">Kembali /a>
23 </form>
24 </div>
25 <script>
26 document.addEventListener("DOMContentLoaded", async () => {
27   const token = localStorage.getItem("token");
28   const user = JSON.parse(localStorage.getItem("user"));
29   const id = [ $id ]; // property assignment expected
30
31   const res = await fetch(`/api/vendor/${id}`, {
32     headers: {
33       "Authorization": "bearer " + token,
34       "Accept": "application/json"
35     },
36   });
37
38   const vendor = await res.json();
39   const form = document.getElementById("formEditVendor");
40   form.name.value = vendor.name;
41   form.phone.value = vendor.phone;
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Halaman ini adalah form edit data vendor dalam sistem Laravel. Saat halaman dimuat, sistem mengambil data vendor dari API /api/vendor/{id} menggunakan token autentikasi. Data seperti nama, nomor telepon, dan alamat vendor diisi otomatis ke dalam form.

Pengguna dapat mengubah informasi tersebut, lalu mengklik tombol Update untuk menyimpan perubahan. Data yang diperbarui akan dikirim ke server melalui metode PUT, termasuk ID user (created_by) yang mengedit data.

Jika proses berhasil, muncul notifikasi sukses dan pengguna diarahkan ke halaman daftar vendor. Jika gagal, akan muncul pesan error menggunakan SweetAlert.

15

```

<code>
</code>

```

Halaman ini merupakan laporan paket bulanan yang menampilkan seluruh data pengiriman selama bulan dan tahun tertentu. Pengguna dapat memilih bulan dan tahun dari dropdown, lalu menampilkan data yang sesuai dengan periode tersebut. Data ditarik berdasarkan parameter bulan dan tahun, kemudian ditampilkan dalam bentuk tabel lengkap.

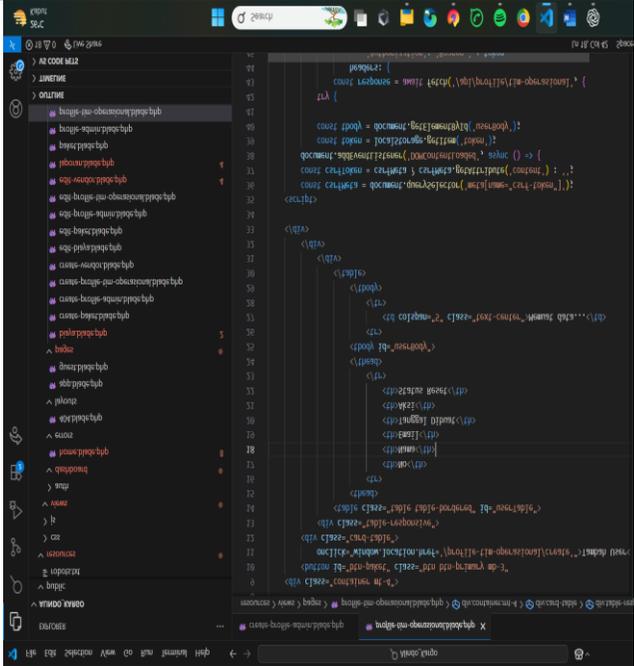
Tabel laporan mencakup informasi detail seperti:

- Nomor resi, deskripsi, berat, volume, jumlah koli
- Kota tujuan, nama dan kontak penerima
- Nama vendor, nama penginput (tim operasional)
- Total biaya vendor, biaya lainnya, total pengeluaran, dan pendapatan
- Tanggal input

Bagian bawah tabel (tfoot) menampilkan akumulasi total pengeluaran dan pendapatan bulan tersebut.

Tersedia juga tombol Export ke Excel, yang akan menghasilkan file Excel dari data laporan berdasarkan bulan dan tahun yang dipilih.

Semua nilai numerik diformat ke dalam bentuk mata uang Rupiah, dan data biaya lainnya diproses dari array atau nilai tunggal jika tersedia. Halaman ini menggunakan @forelse untuk menangani kasus jika data kosong, dan menghitung total menggunakan Collection::sum() dari Laravel.

		<p>Tabel dilengkapi dengan plugin DataTables untuk pencarian, pagination, dan info data. Seluruh aktivitas halaman ini diamankan dengan token dan CSRF, serta disesuaikan berdasarkan status pengguna.</p>
18		<p>Halaman ini adalah tampilan manajemen akun tim operasional dalam sistem Laravel. Di dalamnya, pengguna dapat melihat daftar user dengan role tim-operasional, serta melakukan aksi seperti menambah, mengedit, atau menghapus akun.</p> <p>Saat halaman dimuat, data pengguna diambil dari endpoint <code>/api/profile/tim-operasional</code> menggunakan token autentikasi. Data yang ditampilkan mencakup nama, email, tanggal dibuat, dan status permintaan reset password.</p> <p>Tiap user memiliki kolom Aksi yang berisi tombol Edit dan Hapus. Jika pengguna pernah mengajukan reset password, maka statusnya akan tampil:</p> <ul style="list-style-type: none"> • Belum ada permintaan reset • Menunggu persetujuan (dapat disetujui dengan tombol "Setujui") • Disetujui (diberikan link reset) • Sudah dipakai atau status tidak diketahui <p>Data ditampilkan dalam tabel interaktif menggunakan DataTables. Saat tombol Hapus diklik, sistem akan meminta konfirmasi, lalu menghapus user melalui permintaan DELETE jika disetujui. Semua proses diamankan menggunakan token dan CSRF.</p>

19

```

6 <div class="container">
7 <a href="/vendor/create" class="btn btn-primary mb-3">Tambah Vendor</a>
8 <div class="card table">
9 <div class="table-responsive">
10 <table class="table table-bordered" id="vendorTable">
11 <thead>
12 <tr>
13 <th>No</th>
14 <th>Nama</th>
15 <th>Telepon</th>
16 <th>Alamat</th>
17 <th>Aksi</th>
18 </tr>
19 </thead>
20 <tbody id="vendorbody">
21 <tr>
22 <td colspan="5" class="text-center">Memuat data...</td>
23 </tr>
24 </tbody>
25 </table>
26 </div>
27 </div>
28 </div>
29 </div>
30 <script>
31 document.addEventListener('DOMContentLoaded', async () => {
32   const token = localStorage.getItem('token');
33   const tbody = document.getElementById('vendorbody');
34
35   try {
36     const res = await fetch('/api/vendor', {
37       headers: {
38         'Authorization': 'Bearer ' + token,
39         'Accept': 'application/json'
40       }
41     });
42   }
43 });

```

Halaman ini adalah tampilan manajemen data vendor dalam sistem Laravel. Di dalamnya, pengguna dapat melihat daftar vendor yang tersedia, serta melakukan aksi seperti menambah, mengedit, atau menghapus vendor.

Saat halaman dimuat, data vendor diambil dari endpoint `/api/vendor` menggunakan token autentikasi. Data ditampilkan dalam tabel yang berisi kolom: Nomor, Nama, Telepon, Alamat, dan Aksi.

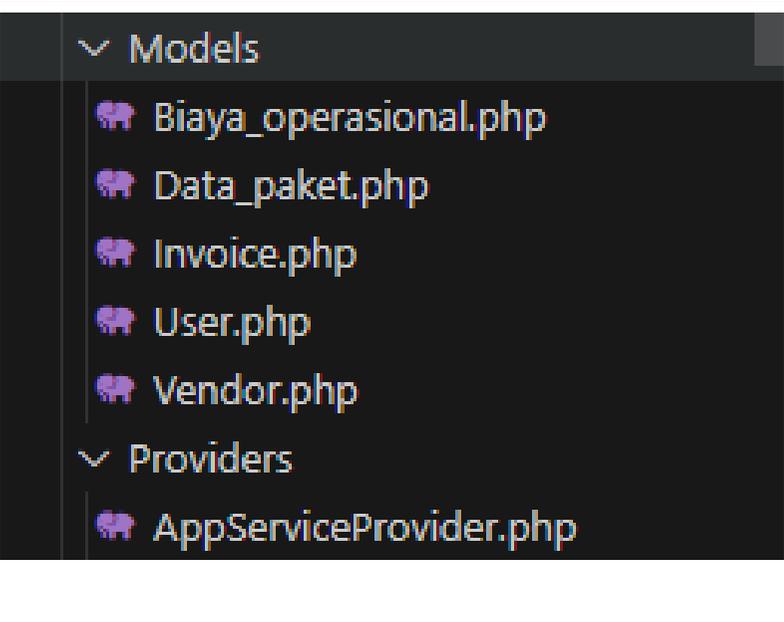
Jika data tersedia, tiap baris akan dilengkapi tombol Edit (menuju halaman `/vendor/edit/{id}`) dan tombol Hapus, yang akan meminta konfirmasi sebelum menghapus vendor lewat API.

Tabel ini menggunakan plugin DataTables untuk mendukung fitur pencarian, pagination, dan tampilan interaktif. Jika data gagal dimuat, akan ditampilkan pesan kesalahan. Semua proses diamankan dengan token dan validasi konfirmasi dari pengguna.

D. Tim Proyek Back End

1. Membuat database serta tabel-tabel
2. Membuat controller
3. Membuat API
4. Membuat autentikasi
5. Lampiran pekerjaan

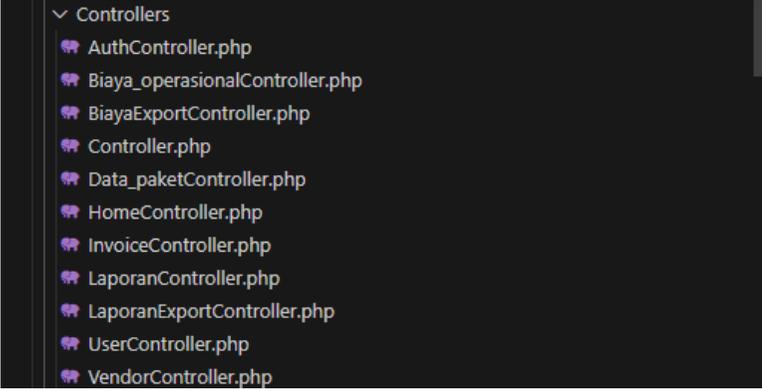
Tabel 3. 5 Lampiran Kode Program Back-End

1		<p>Folder ini berisi Model Eloquent, yaitu representasi dari tabel-tabel di database. Masing-masing file di sini digunakan untuk menangani logika data dan relasi antar tabel.</p> <p>Semua file dalam folder ini berperan penting dalam pengolahan logika backend dan relasi antar data dalam sistem manajemen perusahaan. Saat membuat API atau memproses data, model-model inilah yang akan digunakan oleh Controller untuk menyimpan, mengambil, dan memanipulasi data.</p>
---	--	---

2	<pre> class Biaya_operasional extends Model { use HasFactory; protected \$fillable = ['resi', 'total_vendor', 'total_paket', 'biaya_lainnya', 'total_keseluruhan', 'created_by',]; protected \$casts = ['biaya_lainnya' => 'array',]; public function creator() { return \$this->belongsTo(User::class, 'created_by'); } public function paket() { return \$this->belongsTo(\App\Models\Data_paket::class, 'resi', 'resi'); } } </pre>	<p>Model ini merepresentasikan tabel biaya_operasional di database dan digunakan Laravel untuk mengelola data biaya operasional perusahaan.</p> <p>Model ini digunakan untuk menyimpan dan mengambil data biaya operasional yang dibuat oleh user, serta menghubungkannya ke data paket melalui resi.</p>
3	<pre> public function creator() { return \$this->belongsTo(User::class, 'created_by'); } public function invoice() { return \$this->hasOne(Invoice::class, 'package_id'); } public function vendors() { return \$this->belongsToMany(\App\Models\Vendor::class, 'data_paket_vendor') ->withPivot('biaya_vendor') ->withTimestamps(); } public function Biaya_operasional() { return \$this->hasOne(Biaya_operasional::class, 'resi', 'resi'); } protected static function boot() { parent::boot(); static::deleting(function (\$paket) { \$paket->Biaya_operasional()?->delete(); }); } } </pre>	<p>Kode yang Anda tampilkan ini adalah bagian dari Model Data_paket di Laravel, yang menunjukkan relasi antar tabel dan logika khusus saat data dihapus.</p> <p>Juga memiliki fitur auto-delete biaya operasional saat paket dihapus, untuk menjaga integritas data.</p> <p>Jika Anda ingin, saya bisa bantu buat diagram visual relasinya (antara Data_paket, User, Vendor, Invoice, dll)</p>

4	<pre> public function vendor() { return \$this->belongsTo(Vendor::class); } public function package() { return \$this->belongsTo(Data_paket::class, 'package_id'); } public function creator() { return \$this->belongsTo(User::class, 'created_by'); } </pre>	<p>Kode yang ditampilkan adalah bagian dari sebuah model Laravel dari model Invoice.php atau Biaya_operasional.php yang menunjukkan relasi antar data. Code ini cocok untuk digunakan pada model seperti Invoice, Biaya_operasional, atau tabel pivot dengan informasi biaya.</p>
5	<pre> class User extends Authenticatable { /** @use HasFactory<\Database\Factories\UserFactory> */ use HasApiTokens, HasFactory, Notifiable; /** * The attributes that are mass assignable. * * @var list<string> */ protected \$fillable = ['name', 'username', 'email', 'password', 'role',]; } </pre>	<p>Kode ini adalah model User di Laravel, yang digunakan untuk mengelola data pengguna dalam sistem. Model ini memperluas Authenticatable, yang artinya ia dipakai untuk fitur autentikasi seperti login dan manajemen sesi pengguna.</p>

6	<pre> class Vendor extends Model { use HasFactory; protected \$fillable = ['name', 'phone', 'address', 'created_by']; public function creator() { return \$this->belongsTo(User::class, 'created_by'); } public function packages() { return \$this->hasMany(Data_paket::class); } public function invoices() { return \$this->hasMany(Invoice::class); } public function data_pakets() { return \$this->belongsToMany(Data_paket::class, 'data_paket_vendor')->withPivot('biaya_vendor'); } } </pre>	<p>Ini adalah model Laravel yang merepresentasikan data vendor di database. Mewakili entitas seperti supplier, mitra pengiriman, atau penyedia layanan lainnya.</p>
7	<pre> migrations ├── 0001_01_01_000000_create_users_table.php ├── 0001_01_01_000001_create_cache_table.php ├── 0001_01_01_000002_create_jobs_table.php ├── 2025_06_20_151347_create_personal_access_tokens_table.php ├── 2025_06_20_161444_create_biaya_operasionals_table.php ├── 2025_06_20_164202_create_vendors_table.php ├── 2025_06_20_164934_create_data_pakets_table.php ├── 2025_06_20_173650_add_role_to_users_table.php ├── 2025_06_20_174955_add_username_to_users_table.php ├── 2025_06_20_182123_add_biaya_operasional_id_and_resi_to_paket_table.php ├── 2025_06_20_182829_add_resi_to_biaya_operasionals_table.php ├── 2025_06_20_183009_remove_resi_from_data_pakets_table.php ├── 2025_06_20_184137_create_invoices_table.php ├── 2025_06_21_155053_add_resi_to_data_pakets_table.php ├── 2025_06_21_165519_add_detail_fields_to_data_pakets_table.php ├── 2025_06_24_092738_create_data_paket_vendor_table.php ├── 2025_06_24_181043_drop_vendor_id_from_data_pakets_table.php ├── 2025_06_25_050331_update_biaya_operasionals_structure.php ├── 2025_06_30_030934_update_biaya_operasionals_table.php └── 2025_07_13_091516_add_alasan_gagal_to_data_pakets_table.php </pre>	<p>Gambar disamping menampilkan file-file migrasi Laravel yang terletak di folder database/migrations. File migrasi digunakan untuk membuat dan mengelola struktur tabel database secara terstruktur dan otomatis menggunakan perintah Laravel seperti php artisan migrate.</p>

8		<p>Gambar tersebut menampilkan daftar file controller pada folder app/Http/Controllers dalam sebuah project Laravel. Folder ini berisi logika bisnis (business logic) yang menghubungkan model (data) dan view (tampilan antarmuka).</p>
9	<pre data-bbox="201 542 873 1197"> class AuthController extends Controller { public function login(Request \$request) { \$request->validate(['email' => 'required email', 'password' => 'required']); \$user = User::where('email', \$request->email)->first(); if (!\$user) { return response()->json(['message' => 'Email tidak ditemukan.'], 404); } if (!Hash::check(\$request->password, \$user->password)) { return response()->json(['message' => 'Password salah.'], 401); } // Jika berhasil Auth::login(\$user); // optional: kalau kamu ingin session return response()->json(['message' => 'Login berhasil', 'token' => \$user->createToken('api-token')->plainTextToken, 'user' => \$user]); } } </pre>	<p>AuthController adalah controller di Laravel yang digunakan untuk menangani proses login dan registrasi pengguna. Pada bagian login, sistem akan memeriksa apakah email dan password yang dimasukkan oleh pengguna sesuai dengan data yang ada di database. Jika cocok, maka pengguna akan diberikan token sebagai bukti bahwa dia berhasil login. Token ini bisa digunakan untuk mengakses fitur lain yang memerlukan otentikasi.</p>

10	<pre> public function index() { \$biayas = Biaya_operasional::all(); \$resis = \$biayas->pluck('resi')->filter(); \$pakets = Data_paket::with('vendors')->whereIn('resi', \$resis)->get()->keyBy('resi'); \$data = \$biayas->map(function (\$item) use (\$pakets) { \$item->vendor_names = []; \$item->total_vendor = 0; \$item->total_paket = 0; \$item->total_pengeluaran = 0; \$item->pendapatan = 0; \$paket = \$pakets[\$item->resi] ?? null; if (\$paket) { \$item->vendor_names = \$paket->vendors->pluck('name')->toArray(); // Hitung total biaya vendor dari pivot \$item->total_vendor = \$paket->vendors->sum(function (\$vendor) { return \$vendor->pivot->biaya_vendor ?? 0; }); // Biaya kirim asli dari kolom cost \$item->total_paket = \$paket->cost ?? 0; } // Hitung biaya lainnya \$totalBiayaLain = 0; if (is_array(\$item->biaya_lainnya)) { foreach (\$item->biaya_lainnya as \$biaya) { </pre>	<p>Biaya_operasionalController adalah bagian dari backend aplikasi yang bertanggung jawab untuk mengelola data terkait biaya operasional dalam proses pengiriman. Controller ini menghubungkan antara data biaya dengan paket pengiriman yang memiliki nomor resi tertentu.</p> <p>Saat data diminta (dengan fungsi index atau show), sistem akan mencocokkan data resi dari biaya operasional dengan data paket, lalu menghitung beberapa hal seperti total biaya vendor, biaya pengiriman paket, biaya tambahan lainnya, serta pendapatan yang dihasilkan (yaitu selisih antara biaya pengiriman dan total pengeluaran).</p>
11	<pre> class BiayaExportController extends Controller { public function export() { \$biayas = Biaya_operasional::all(); \$resis = \$biayas->pluck('resi')->filter(); \$pakets = Data_paket::with('vendors')->whereIn('resi', \$resis)->get()->keyBy('resi'); \$data = \$biayas->map(function (\$item) use (\$pakets) { \$item->vendor_names = []; \$item->total_vendor = 0; \$item->total_paket = 0; \$paket = \$pakets[\$item->resi] ?? null; if (\$paket) { \$item->vendor_names = \$paket->vendors->pluck('name')->toArray(); \$item->total_vendor = \$paket->vendors->sum(fn(\$vendor) => \$vendor->pivot->biaya_vendor ?? 0); \$item->total_paket = \$paket->cost ?? 0; } \$biayaLainnya = is_array(\$item->biaya_lainnya) ? \$item->biaya_lainnya : json_decode(\$item->biaya_lainnya, true); \$item->total_biaya_lainnya = is_array(\$biayaLainnya) ? collect(\$biayaLainnya)->sum(fn(\$b) => \$b['jumlah'] ?? 0) : 0; \$item->pengeluaran = \$item->total_vendor + \$item->total_biaya_lainnya; \$item->pendapatan = \$item->total_paket - \$item->pengeluaran; return \$item; </pre>	<p>Kode BiayaExportController ini digunakan untuk mengekspor data biaya operasional ke dalam file Excel (.xlsx). Secara sederhana, prosesnya dimulai dengan mengambil semua data biaya operasional beserta data paket yang memiliki nomor resi yang sesuai. Kemudian, data tersebut diproses untuk menghitung total biaya dari vendor, biaya lainnya, total pengeluaran, dan juga pendapatan yang dihasilkan.</p> <p>Setelah data dihitung dan disusun, sistem membuat spreadsheet baru menggunakan library PhpSpreadsheet.</p>

12	<pre> class Data_paketController extends Controller { public function index() { \$data = Data_paket::with(['vendors', 'creator'])->get(); return response()->json(\$data); } public function store(Request \$request) { \$validated = \$request->validate(['vendor_ids' => 'nullable array', 'vendor_ids.*' => 'exists:vendors,id', 'vendor_biayas' => 'nullable array', 'vendor_biayas.*' => 'nullable numeric min:0', 'resi' => 'required string unique:data_pakets,resi', 'description' => 'sometimes string', 'kota_asal' => 'sometimes string', 'kota_tujuan' => 'sometimes string', 'no_hp_pengirim' => 'nullable string', 'penerima' => 'nullable string', 'no_hp_penerima' => 'nullable string', 'alamat_penerima' => 'nullable string', 'weight' => 'sometimes numeric', 'jumlah_koli' => 'sometimes numeric', 'volume' => 'sometimes numeric', 'cost' => 'sometimes numeric', 'created_by' => 'sometimes exists:users,id', 'status' => 'sometimes string in:Terkirim,Dalam Proses,Gagal', 'alasan_gagal' => 'nullable string',]); } } </pre>	<p>Kode Data_paketController ini digunakan untuk mengelola data paket pengiriman dalam aplikasi, termasuk membuat, melihat, mengubah, dan menghapus data paket. Saat pengguna menambahkan data paket, sistem akan memvalidasi input seperti informasi pengiriman, vendor yang terlibat, dan biaya masing-masing vendor. Jika ada vendor yang dipilih, sistem juga akan menyimpan biaya vendor tersebut dan menghitung total biaya pengiriman dan pengeluaran.</p>
13	<pre> class HomeController extends Controller { public function index() { // Ambil semua data biaya \$biayas = Biaya_operasional::all(); // Ambil semua resi dari biaya dan cari data paket sesuai \$resis = \$biayas->pluck('resi')->filter(); \$pakets = Data_paket::with('vendors')->whereIn('resi', \$resis)->get()->keyBy('resi'); \$totalPendapatan = 0; \$totalPengeluaran = 0; foreach (\$biayas as \$item) { \$total_vendor = 0; \$total_paket = 0; // Cek apakah resi punya paket terkait \$paket = \$pakets[\$item->resi] ?? null; if (\$paket) { // Total biaya vendor dari relasi pivot \$total_vendor = \$paket->vendors->sum(function (\$vendor) { return \$vendor->pivot->biaya_vendor ?? 0; }); // Biaya pengiriman asli \$total_paket = \$paket->cost ?? 0; } } } } </pre>	<p>Kode HomeController ini digunakan untuk menampilkan data ringkasan keuangan dan statistik paket pada halaman dashboard aplikasi. Ketika halaman dashboard dibuka, controller ini mengambil seluruh data biaya operasional dan mencocokkannya dengan data paket berdasarkan nomor resi.</p> <p>Kemudian, sistem menghitung total pengeluaran (berdasarkan biaya vendor dan biaya tambahan lainnya) serta total pendapatan (dari biaya pengiriman dikurangi pengeluaran)</p>

14	<pre> class InvoiceController extends Controller { public function index() { \$invoices = Invoice::with(['vendor', 'package', 'creator'])->get(); return response()->json(\$invoices); } public function store(Request \$request) { \$validated = \$request->validate(['vendor_id' => 'required exists:vendors,id', 'package_id' => 'nullable exists:data_pakets,id', 'invoice_number' => 'required string unique:invoices', 'status' => 'in:submitted,reviewed,approved', 'total_amount' => 'required numeric', 'created_by' => 'required exists:users,id',]); \$invoice = Invoice::create(\$validated); return response()->json(\$invoice, 201); } } </pre>	<p>Kode ini merupakan bagian dari sistem manajemen logistik yang digunakan untuk mengelola data invoice atau tanda terima pengiriman barang. Fungsinya meliputi pembuatan, pembaruan, penampilan, serta pencetakan dokumen invoice dalam format PDF. Ketika pengguna ingin mencetak invoice berdasarkan data paket tertentu, sistem akan mengambil data terkait seperti nama pengirim, penerima, asal dan tujuan kota, serta informasi lengkap tentang barang yang dikirim. Selain itu, data vendor dan biaya juga disertakan untuk memperhitungkan total keseluruhan biaya pengiriman.</p>
15	<pre> class LaporanController extends Controller { public function index(Request \$request) { \$bulan = \$request->query('bulan', date('m')); \$tahun = \$request->query('tahun', date('Y')); \$startDate = Carbon::createFromDate(\$tahun, \$bulan, 1)->startOfMonth(); \$endDate = Carbon::createFromDate(\$tahun, \$bulan, 1)->endOfMonth(); \$paketList = Data_paket::with(['vendors', 'creator']) ->where('status', 'Ter kirim') ->whereBetween('created_at', [\$startDate, \$endDate]) ->get(); // Ambil biaya operasional berdasarkan resi \$biayaMap = Biaya_operasional::whereIn('resi', \$paketList->pluck('resi')) ->get() ->keyBy('resi'); // Gabungkan biaya operasional ke paket \$data = \$paketList->map(function (\$paket) use (\$biayaMap) { \$biaya = \$biayaMap[\$paket->resi] ?? null; \$paket->total_vendor = \$biaya->total_vendor ?? 0; \$paket->biaya_lainnya = is_array(\$biaya->biaya_lainnya) ? collect(\$biaya->biaya_lainnya)->sum('biaya') : (\$biaya->biaya_lainnya ?? 0); \$paket->pengeluaran = \$paket->total_vendor + \$paket->biaya_lainnya; \$paket->pendapatan = \$paket->cost - \$paket->pengeluaran; }); } } </pre>	<p>Kode ini berfungsi untuk menghasilkan laporan bulanan pengiriman paket yang berhasil terkirim. Saat pengguna mengakses halaman laporan, sistem akan memeriksa apakah ada parameter bulan dan tahun yang dikirim dari permintaan. Jika tidak, maka secara otomatis sistem menggunakan bulan dan tahun saat ini.</p> <p>Selanjutnya, sistem akan mengambil semua data paket yang memiliki status "Ter kirim" dan dibuat dalam rentang tanggal awal hingga akhir bulan tersebut. Setiap paket juga akan memuat informasi tambahan dari relasi vendor serta pengguna yang membuat data tersebut.</p>

16	<pre> class LaporanExportController extends Controller { public function export(Request \$request) { \$bulan = \$request->query('bulan', date('m')); \$tahun = \$request->query('tahun', date('Y')); \$startDate = Carbon::createFromDate(\$tahun, \$bulan)->startOfMonth(); \$endDate = Carbon::createFromDate(\$tahun, \$bulan)->endOfMonth(); \$paketList = Data_paket::with(['vendors', 'creator']) ->where('status', 'Terkirim') ->whereBetween('created_at', [\$startDate, \$endDate]) ->get(); \$biayaMap = Biaya_operasional::whereIn('resi', \$paketList->pluck('resi')) ->get() ->keyBy('resi'); \$spreadsheet = new Spreadsheet(); \$sheet = \$spreadsheet->getActiveSheet(); \$sheet->setCellValue('A1', 'Resi'); \$sheet->setCellValue('B1', 'Deskripsi'); \$sheet->setCellValue('C1', 'Berat'); \$sheet->setCellValue('D1', 'Volume'); \$sheet->setCellValue('E1', 'Jumlah Koli'); \$sheet->setCellValue('F1', 'Kota Tujuan'); \$sheet->setCellValue('G1', 'Nama Penerima'); \$sheet->setCellValue('H1', 'No HP Penerima'); \$sheet->setCellValue('I1', 'Vendor'); \$sheet->setCellValue('J1', 'Pengirim'); } } </pre>	<p>Kode ini digunakan untuk menghasilkan dan mengunduh laporan pengiriman paket dalam bentuk file Excel berdasarkan bulan dan tahun tertentu. Saat pengguna mengakses fungsi ini, sistem terlebih dahulu mengambil data paket yang berstatus "Terkirim" dalam rentang tanggal yang sesuai dengan bulan dan tahun yang dipilih. Setiap data paket juga dilengkapi dengan informasi vendor dan pembuat data.</p>
17	<pre> class UserController extends Controller { // ===== // // SEMUA PENGGUNA // // ===== // public function index() { return response()->json(User::all()); } public function show(\$id) { return response()->json(User::findOrFail(\$id)); } public function store(Request \$request) { \$validated = \$request->validate(['name' => 'required string', 'email' => 'required email unique:users', 'username' => 'required string unique:users,username', 'password' => 'required string min:6', 'role' => 'required in:super-admin,admin,operasional',]); \$validated['password'] = Hash::make(\$validated['password']); \$user = User::create(\$validated); return response()->json(\$user, 201); } } </pre>	<p>Kode di atas adalah implementasi dari UserController di Laravel, yang digunakan untuk mengelola data pengguna dalam aplikasi. Terdapat beberapa fungsi utama yang disediakan, seperti melihat semua pengguna, melihat detail pengguna berdasarkan ID, menambahkan pengguna baru, memperbaiki data pengguna, dan menghapus pengguna. Selain itu, controller ini juga membedakan pengguna berdasarkan peran mereka, yaitu super-admin, admin, dan tim-operasional, sehingga masing-masing peran dapat diatur secara spesifik.</p>

```

18 class VendorController extends Controller
{
    // Tampilkan semua vendor
    public function index()
    {
        return response()->json(Vendor::all());
    }

    // Simpan vendor baru
    public function store(Request $request)
    {
        $validated = $request->validate([
            'name' => 'required|string|max:255',
            'phone' => 'required|string|max:20',
            'address' => 'required|string',
            'created_by' => 'required|exists:users,id',
        ]);

        $vendor = Vendor::create($validated);
        return response($vendor, 201)
            ->header('Content-Type', 'application/json');
    }

    // Tampilkan vendor berdasarkan ID
    public function show($id)
    {
        $vendor = Vendor::findOrFail($id);
        return response()->json($vendor);
    }
}

```

Kode VendorController di atas adalah bagian dari aplikasi Laravel yang digunakan untuk mengelola data vendor atau rekanan jasa. Controller ini menyediakan fungsi-fungsi standar seperti menampilkan semua vendor, menyimpan data vendor baru, menampilkan data vendor berdasarkan ID tertentu, memperbarui data vendor, dan menghapus vendor dari database. Setiap permintaan yang masuk akan divalidasi terlebih dahulu untuk memastikan bahwa data yang dikirimkan sesuai dengan format yang diharapkan.

```

19 File Edit Selection View Go Run Terminal Help
ALINDO_KARGO
resources
views
layouts
app.blade.php
guest.blade.php
pages
biaya.blade.php
create-paket.blade.php
create-paket-admin.blade.php
create-profile-admin.blade.php
create-profile-tim-operasional.blade.php
create-vendor.blade.php
edit-biaya.blade.php
edit-paket.blade.php
edit-profile-admin.blade.php
edit-profile-tim-operasional.blade.php
edit-vendor.blade.php
laporan.blade.php
paket.blade.php
profile-admin.blade.php
profile-tim-operasional.blade.php
vendor.blade.php
partials
welcome.blade.php
routes
api.php
console.php
web.php
storage
tests
OUTLINE
TIMELINE
VS CODE PETS
Suhu tertinggi b...
Memecahkan re...

```

```

// Paket routes
Routes::prefix('paket')->group(function () {
    Routes::post('/register', [AuthController::class, 'register']);
    // User info & logout
    Routes::get('/user', function (Request $request) {
        return $request->user();
    });
    Routes::post('/logout', [AuthController::class, 'logout']);
});

// Biaya Operasional Routes
Routes::prefix('biaya')->group(function () {
    Routes::get('/', [BiayaOperasionalController::class, 'index']);
    Routes::get('{id}', [BiayaOperasionalController::class, 'show']);
    Routes::post('/', [BiayaOperasionalController::class, 'store']);
    Routes::put('{id}', [BiayaOperasionalController::class, 'update']);
    Routes::delete('{id}', [BiayaOperasionalController::class, 'destroy']);
});

// Laporan Routes
Routes::prefix('laporan')->group(function () {
    Routes::get('/', [LaporanController::class, 'index']);
    Routes::get('{id}', [LaporanController::class, 'show']);
    Routes::post('/', [LaporanController::class, 'store']);
    Routes::put('{id}', [LaporanController::class, 'update']);
    Routes::delete('{id}', [LaporanController::class, 'destroy']);
});

```

File ini berisi konfigurasi API routes Laravel untuk aplikasi manajemen paket dan keuangan. Route login bersifat publik, sementara semua route lainnya dilindungi oleh middleware auth:sanctum, sehingga hanya bisa diakses setelah login.

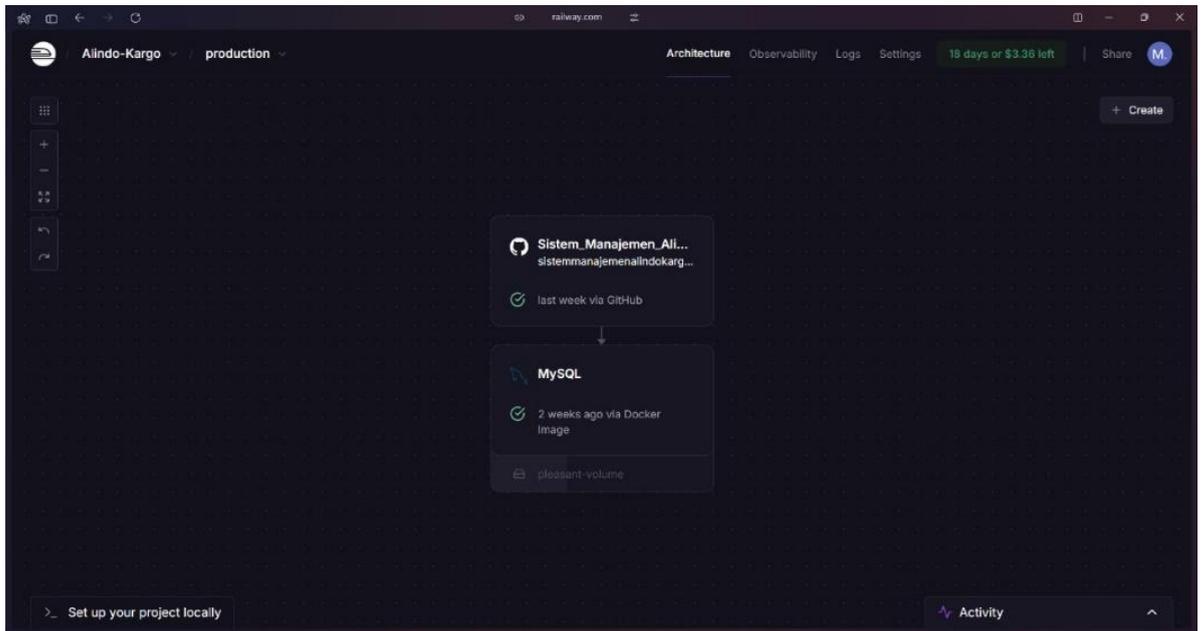
Rute yang tersedia mencakup:

- Autentikasi (login, logout, register, info user)
- Manajemen paket, biaya operasional, laporan, dan invoice
- CRUD vendor
- Manajemen user admin dan tim operasional
- Edit profil pengguna berdasarkan role
- Satu route /ping untuk pengecekan koneksi API.

3. 3. Penjamin kualitas proyek

A. Deployment

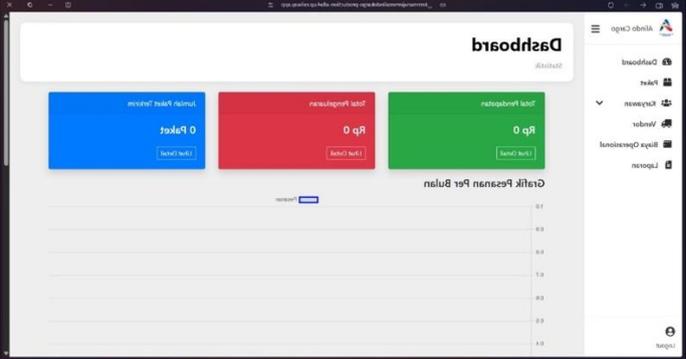
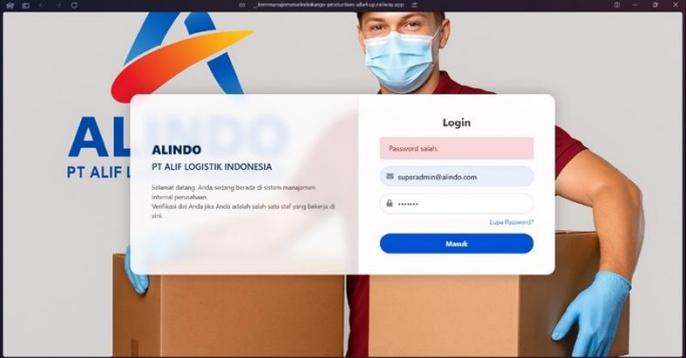
Proyek ini di deploy di platform hosting yang gratis (Railway)

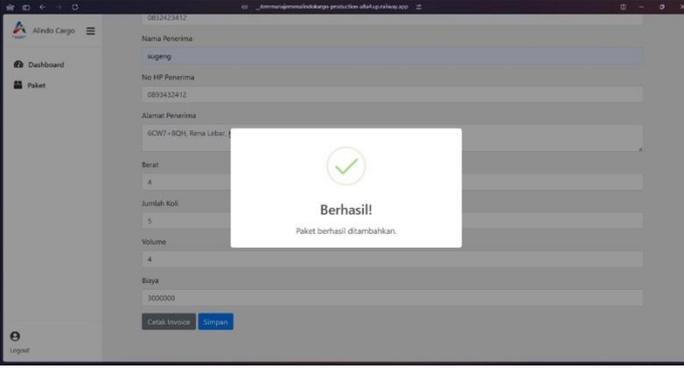
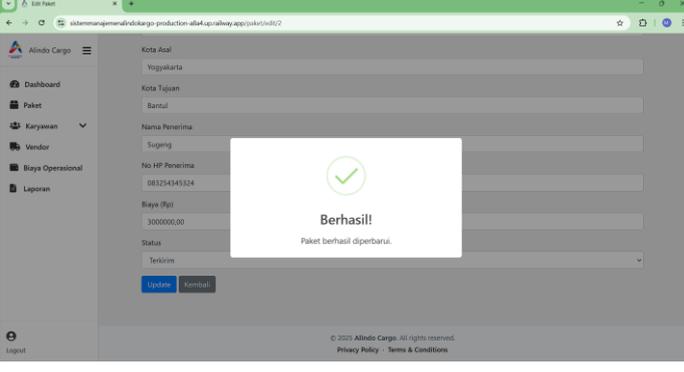
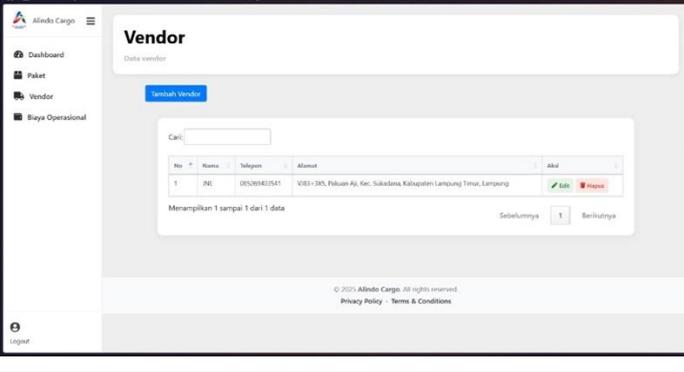
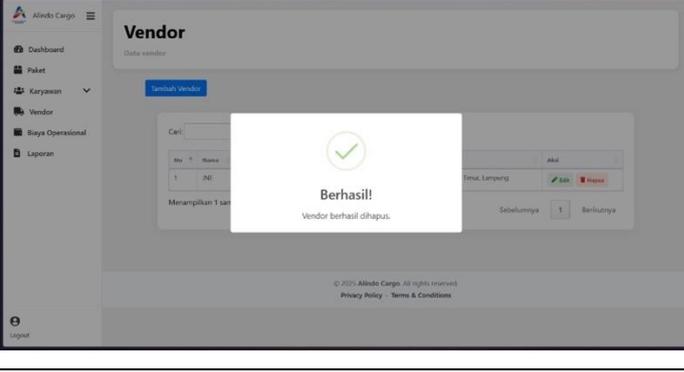


Gambar 3. 48 Deployment

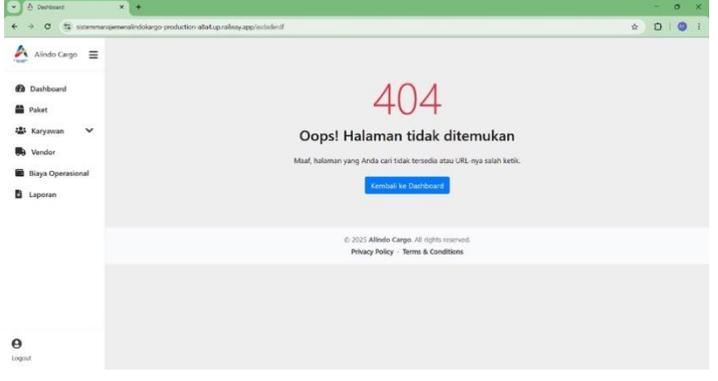
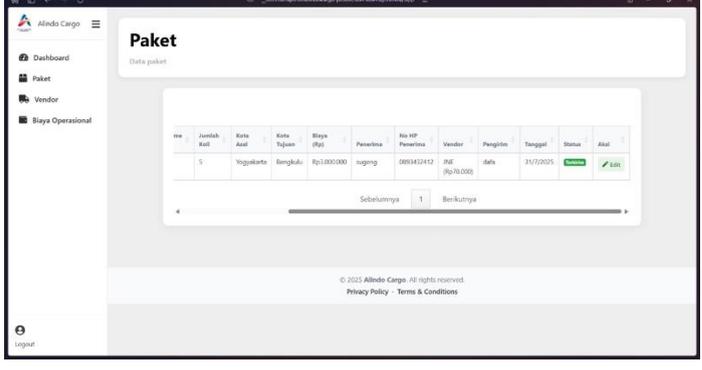
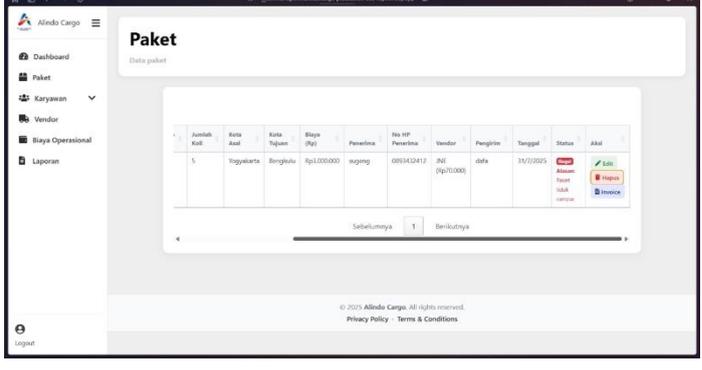
B. Testing Pengujian Luar

Tabel 3. 6 Hasil Testing

Fitur yang Diuji	Tampilan Website	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual
Login Super Admin		Masukkan email dan password yang valid	Berhasil masuk ke dashboard Super Admin	Valid
Login Gagal		Masukkan email/password yang salah	Muncul notifikasi kesalahan "Email atau Password salah"	Valid

<p>Input Data Paket (Tim Ops)</p>		<p>Isi semua form invoice dengan benar lalu submit</p>	<p>Data tampil di tabel Admin dan Super Admin</p>	<p>Valid</p>
<p>Edit Data Paket (Admin)</p>		<p>Klik tombol edit, ubah data tujuan pengiriman lalu simpan</p>	<p>Data berhasil diperbarui dan muncul alert sukses</p>	<p>Valid</p>
<p>Tambah Vendor (Admin)</p>		<p>Tambah nama vendor dan harga lalu simpan</p>	<p>Vendor baru muncul di tabel vendor</p>	<p>Valid</p>
<p>Hapus Vendor (Admin)</p>		<p>Klik tombol hapus pada salah satu vendor</p>	<p>Data vendor hilang dari tabel setelah konfirmasi</p>	<p>Valid</p>

<p>Cetak Laporan (Super Admin)</p>		<p>Akses halaman laporan, klik “Cetak”</p>	<p>File Excel terunduh dengan data rekap paket</p>	<p>Valid</p>
<p>Edit Biaya Operasional</p>		<p>Klik edit biaya, ubah nominal bahan bakar lalu simpan</p>	<p>Data baru muncul di tabel biaya operasional</p>	<p>Valid</p>
<p>Cek Total Keuntungan</p>		<p>Input biaya operasional, lihat perhitungan total di dashboard</p>	<p>Total rugi/laba muncul secara otomatis</p>	<p>Valid</p>
<p>Lupa Password</p>		<p>Klik “Lupa Password” di halaman login, masukkan email terdaftar, cek email, klik tautan reset, buat password baru</p>	<p>Email reset dikirim, tautan dapat digunakan, password berhasil diubah</p>	<p>Valid</p>

<p>URL Halaman Lain</p>		<p>Akses langsung URL admin/staf tanpa login (contoh: /admin, /dashboard)</p>	<p>Dialihkan ke halaman login jika belum autentikasi</p>	<p>Valid</p>
<p>Status Berubah Otomatis</p>		<p>Input biaya lainnya pada halaman edit paket</p>	<p>Status paket otomatis berubah jadi "Terkirim" setelah biaya lain disimpan</p>	<p>Valid</p>
<p>Mengubah Status gagal dan otomatis keluar alasan gagal.</p>		<p>Di halaman admin, ubah status paket menjadi "Gagal", lalu input alasan (contoh: "Alamat tidak ditemukan")</p>	<p>Status berubah ke "Gagal" dan alasan ditampilkan otomatis di tabel serta halaman detail</p>	<p>Valid</p>

3. Testing Webqual

(Evaluasi ini akan ditambahkan secara manual oleh pengguna berdasarkan hasil kuesioner 10 orang)

- a. Pada tahap Uji Normalitas dihasilkan nilai **Asymp. Sig. (2-tailed) = 0.563**, Karena nilai $p = 0.563 > 0.05$, maka data berdistribusi normal.

One-Sample Kolmogorov-Smirnov Test		
		Unstandardized Predicted Value
N		5
Normal Parameters ^a	Mean	3.4000000
	Std. Deviation	4.97995984
Most Extreme Differences	Absolute	.353
	Positive	.353
	Negative	-.247
Kolmogorov-Smirnov Z		.788
Asymp. Sig. (2-tailed)		.563
a. Test distribution is Normal.		

Gambar 3. 49 Uji Normalitas

- b. Uji Autokorelasi dihasilkan nilai **Durbin-Watson = 2.210** karena nilai diatas 2 maka **tidak terdapat autokorelasi**

Model Summary^b

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Durbin-Watson
1	1.000 ^a	1.000	1.000	.00000	2.210

a. Predictors: (Constant), total_x3, total_x2, total_x1

b. Dependent Variable: y

Gambar 3. 50 Uji Autokorelasi

- c. Uji Multikolinearitas dihasilkan nilai **VIF 40.200, 28.588, dan 28.588** jika VIF lebih besar dari 10 maka terdapat multikolinearitas yang tinggi karena diatas 10 **maka multikolinearitas tinggi**

Coefficients^a

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	Collinearity Statistics	
		B	Std. Error	Beta			Tolerance	VIF
1	(Constant)	7.930E-15	.000		.	.		
	total_x1	-2.911E-14	.000	.000	.	.	.025	40.200
	total_x2	3.273	.000	1.993	.	.	.035	28.588
	total_x3	-1.727	.000	-1.052	.	.	.035	28.588

a. Dependent Variable: y

Gambar 3. 51 Uji Multikolinearitas

- d. Uji Heteroskedastisitas dihasilkan nilai **Sig. 1.000** jika nilai $p < 0,05$ terjadi heteroskedastisitas, karena diatas 0,05 **maka tidak ada masalah heteroskedastisitas (data lolos uji).**

Coefficients^a

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	1.687E-12	.000		.000	1.000
	total_x1	-1.075E-11	.000	.000	.000	1.000
	total_x2	7396.909	.000	3.491	3.014E7	.000
	total_x3	-5948.091	.000	-2.807	-2.424E7	.000

a. Dependent Variable: ABS_RES

Gambar 3. 12 Uji Heteroskedastisitas

- e. Uji t dihasilkan nilai **Sig = 0.500, 0.260, 0.102, dan 0.201** jika nilai $p <$ dari 0.05 maka variable signifikan, karena diatas 0.05 **maka variable tidak signifikan**

Coefficients^a

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	2.000	2.000		1.000	.500
	total_x1	8.000	3.464	.464	2.309	.260
	total_x2	15.364	2.493	1.044	6.163	.102
	total_x3	-7.636	2.493	-.519	-3.063	.201

a. Dependent Variable: total_y

Gambar 3. 53 Uji T

3. 4. Kendala Proyek

Adapun kendala yang dihadapi dalam proyek ini adalah kami tidak mendapatkan informasi lebih dari client sehingga batal, mengapa bisa terjadi seperti itu karena pada saat pertemuan awal PM dengan client, client meminta untuk melihat website yang akan dibuat terlebih dahulu baru akan menandatangani MoU dan kami menyepakati itu dengan client. Lalu tim kami melanjutkan untuk membuat interface agar bisa diperlihatkan kepada client untuk melanjutkan proyek ini. Pada pertemuan kedua via zoom dengan client kami menyampaikan kembali hasil progres website kami namun client bilang ingin bertemu kami langsung, akhirnya pertemuan ketiga kami dengan client disitu membahas final fitur apa saja yang client butuhkan pada lalu client merevisi website yang sudah kami kerjakan sedikit untuk diperlihatkan kepada client, dan client minta untuk ditunjukkan kepadanya hasil dari yang ia minta. Setelah kami selesaikan website permintaan client yang terakhir dan meminta untuk ttd MoU client tiba tiba susah dihubungi dan jarang membalas cht, akhirnya client pada saat membalas cht ia mengatakan bahwa ia tidak mau melanjutkan proyek ini, bahkan tidak mau melihat website yang sudah kami buat sampai jadi.

Kesimpulannya proyek ini batal bukan karena kendala teknis dari tim kami, melainkan karena kurangnya komitmen dan itikad baik dari pihak client setelah kami memenuhi semua yang mereka minta. Hal ini menjadi pelajaran penting bahwa kesepakatan resmi dan MoU seharusnya disepakati terlebih dahulu sebelum pengerjaan dilakukan agar risiko seperti ini dapat diminimalisir.

3. 5. Keberlanjutan Proyek

Keberlanjutan dari proyek ini adalah tim tidak melanjutkan perjanjian dengan client, karena client menolak untuk melihat website yang telah jadi dan tidak mau menandatangani MoU, namun tim akan tetap mempresentasikan website ini dalam seminar MPTI.

BAB IV

PENUTUP

4. 1. Kesimpulan

Berdasarkan sistem yang sudah dibuat oleh tim kami pada program ini akan memberikan kemudahan bagi pengolahan paket yang akan dikirimkan juga pada pembuatan laporannya, dan akan menggantikan sistem manual pada proses pendataan menjadi digital, sehingga memberikan dampak positif baik dari segi waktu, biaya, tenaga dan tempat.

A. Sistem yang tim kembangkan merupakan hasil dari permintaan mitra berdasarkan kebutuhan fungsionalnya, yaitu sebagai berikut:

1. Sistem Manajemen Perusahaan
2. Pembuatan Invoice Digital
3. Sistem rekap laporan

B. Program yang telah di selesaikan merupakan program yang dikembangkan melalui beberapa tahapan diantaranya :

1. Analisis kebutuhan sistem,
2. Tahapan perencanaan,
3. Pemodelan,
4. Konstruksi

C. Penyerahan sistem kepada pelanggan dengan dukungan perangkat lunak yang telah tersedia.

1. Hingga Testing dan Pelatihan.

D. Dari langkah-langkah yang kami kerjakan secara berurutan menghasilkan program yang dapat digunakan oleh Alindo Cargo.

4. 2. Saran

Sistem yang kami buat sudah diselesaikan sebaik mungkin, pada website sistem harus dilakukan pemeliharaan agar dapat berfungsi dengan baik, dimulai dengan melakukan pemeriksaan, pemantauan, dan perbaikan data, agar sistem terus berjalan dengan baik dan tidak mengalami kendala

LAMPIRAN

1. Proposal

PROPOSAL**Solusi Sistem Manajemen yang lebih Efektif**

Anggota Kelompok :

M. Izza Mubarak 2200018432 (Project Manager)

Daffa Fachrezy 2200018433 (Back End)

Haura Faesa Syahputri A 2200018434 (System Analyst)

M. Sandy Alfachari 2200018479 (Front End)

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS AHMAD DAHLAN

2025

PENDAHULUAN

Sektor logistik dan ekspedisi kargo merupakan tulang punggung dalam sistem distribusi barang, baik di tingkat nasional maupun internasional. Setiap harinya, perusahaan ekspedisi menangani ratusan hingga ribuan transaksi pengiriman, yang masing-masing memerlukan dokumen pendukung seperti invoice atau faktur sebagai bukti pengiriman dan laporan keuangan.

Namun, dalam praktiknya, banyak perusahaan masih mengandalkan sistem manual atau semi-digital yang rentan terhadap kesalahan. Salah satu masalah yang kerap terjadi adalah duplikasi invoice, baik karena kelalaian dalam proses input maupun akibat tindakan manipulatif yang disengaja. Hal ini diperparah dengan tidak adanya sistem yang terintegrasi antar departemen serta lemahnya validasi dan pengawasan terhadap data transaksi.

Jika dibiarkan, permasalahan ini dapat menyebabkan kerugian yang cukup besar, seperti pembayaran ganda, pencatatan pengeluaran palsu, hingga ketidaksesuaian dalam laporan keuangan. Selain berpotensi merusak reputasi perusahaan di mata pelanggan dan mitra kerja, risiko semacam ini juga dapat menimbulkan konsekuensi hukum.

Untuk itu, perusahaan ekspedisi dituntut untuk segera melakukan transformasi digital — yaitu beralih dari sistem manual menuju sistem terintegrasi yang mampu mengelola proses pengiriman dan pencatatan keuangan secara otomatis, rapi, dan aman.

Salah satu kendala utama yang sering dirasakan oleh pemilik usaha adalah ketidakefisienan kerja tim, terutama saat harus melakukan verifikasi invoice secara manual, mengecek ulang dokumen satu per satu, dan melakukan konfirmasi silang antar divisi. Aktivitas ini menyita waktu dan energi, padahal seharusnya dapat dialihkan ke pekerjaan strategis yang lebih produktif.

Dengan hadirnya sistem digital yang tepat, proses seperti pembuatan invoice, pencatatan pengeluaran, dan penyusunan laporan keuangan dapat dilakukan secara otomatis dan terdokumentasi dengan baik. Hal ini tentu berdampak besar terhadap peningkatan efisiensi operasional, pengurangan potensi kesalahan, serta peningkatan transparansi dan akuntabilitas perusahaan.

ALUR KEGIATAN KERJA

Kegiatan pengerjaan sistem atau software diselesaikan sesuai dengan waktu yang telah disepakati antara tim dan klien setelah kontrak ditandatangani oleh kedua belah pihak dengan asumsi klien tidak melakukan proses keterlambatan dalam pengiriman data yang dibutuhkan dan keadaan-keadaan yang tak dapat disanggah (*force majeure*). Secara garis besar alur proses kegiatan kerja pembuatan system yang kita bangun dirincikan sebagai berikut :

Tahapan	Deskripsi
1. Order	Customer Melakukan Pemesanan
2. Follow Up	Kami merespon pemesanan dari customer
3. Konsultasi & Pemesanan	Klien menjelaskan kebutuhan fitur dan sistem
4. Analisis Sistem	Tim kami menganalisis sistem yang dibutuhkan customer
5. Fitur	Tim membuat detail fitur sistem sesuai kebutuhan customer
6. Meeting	Mempresentasikan hasil dan berdiskusi atas hasil fitur dan detail yang telah dibuat
7. Revisi	Customer merevisi (jika ada revisi) dan penyampaian hasil revisi yang telah tim buat.
8. Penawaran	Tim memberikan Proposal Penawaran kepada customer.
9. MoU & Pengembangan	Setelah penandatanganan MoU, Sistem akan di mulai develop.
10. Develop	Pengembangan dimulai setelah kontrak disetujui
11. Revisi	Customer diberikan waktu 2 minggu untuk melakukan 2x revisi
12. Final Version	Tim Wika Production memberikan Final Version kepada customer
13. Maintenance	Garansi bug dan revisi minor selama 4 bulan

Fitur Utama Sistem

1. Pencatatan Pengiriman

Tim operasional dapat menginput data pengiriman secara manual melalui form yang telah disediakan. Setiap entri mencakup informasi lengkap seperti tujuan, jenis paket, biaya pengiriman, serta vendor (jika menggunakan jasa eksternal).

2. Integrasi Otomatis ke Biaya Operasional

Sistem secara otomatis mengelompokkan jenis pengiriman berdasarkan metode (internal atau vendor) dan mencatatnya langsung sebagai bagian dari laporan pengeluaran operasional.

3. Penerbitan Invoice Otomatis (PDF)

Setiap transaksi pengiriman menghasilkan dokumen invoice dalam format PDF yang dapat dicetak langsung atau diunduh sebagai arsip keuangan.

4. Input dan Totalisasi Biaya Tambahan

Pengguna dapat menambahkan biaya lainnya secara manual, lengkap dengan deskripsi kegiatan dan nominal. Total biaya tambahan dihitung otomatis dan ditampilkan dalam tabel rekap.

5. Status Paket Otomatis "Ter kirim"

Setelah seluruh biaya operasional dan biaya lainnya terisi lengkap, sistem secara otomatis mengubah status paket menjadi "Ter kirim" tanpa perlu intervensi manual tambahan.

6. Penandaan Paket "Gagal" dengan Alasan

Jika suatu pengiriman gagal, pengguna dapat menandai paket tersebut sebagai "Gagal" dan mencantumkan alasan kegagalan. Status dan alasan ini akan muncul di seluruh tabel terkait: paket, vendor, staf, dan laporan.

7. Laporan Keuangan Bulanan Terperinci

Sistem menyusun laporan bulanan secara otomatis, menampilkan rincian total pemasukan, total pengeluaran, serta saldo bersih. Semua data dapat difilter per bulan dan diekspor jika dibutuhkan.

8. Dashboard Grafik Interaktif

Tersedia grafik visualisasi jumlah paket per kota tujuan dan jumlah pengiriman per bulan. Hal ini memudahkan manajemen dalam menganalisis performa logistik dan tren operasional.

9. Fungsi Export/Import Data Excel

Seluruh data transaksi dan laporan dapat diunduh dalam format Excel (.xlsx), dan pengguna juga dapat mengunggah data secara massal untuk mempercepat penginputan.

Struktur Hak Akses Pengguna

1. Super Admin

Memiliki kendali penuh atas seluruh sistem, termasuk pengelolaan user (admin dan tim operasional), pengaturan sistem, laporan, serta seluruh data pengiriman dan keuangan.

2. Admin

Bertugas mengelola data pengiriman, mencetak invoice, memverifikasi biaya, dan mengakses seluruh laporan keuangan. Admin tidak dapat mengelola pengguna atau mengakses konfigurasi sistem.

3. Tim Operasional

Fokus pada tugas-tugas operasional seperti input data pengiriman dan pengeluaran. Tidak memiliki akses ke laporan keuangan, grafik, atau hak untuk menghapus data.

Rincian Biaya

RENCANA ANGGARAN BIAYA					
Tahap Perencanaan Proyek					
Kegiatan	Durasi (Jam)	Upah	Biaya Personal	Biaya Bahan	Total Biaya
Analisis Kebutuhan Pengguna	10	Rp20.000	Rp200.000	Rp50.000	Rp250.000
Pembentukan Tim dan Penjadwalan	6	Rp15.000	Rp90.000	Rp-	Rp90.000
Penetapan Milestone proyek	5	Rp16.000	Rp80.000	Rp-	Rp80.000
Total Tahap Perencanaan Proyek	21		Rp370.000	Rp50.000	Rp420.000
Tahap Desain Aplikasi					
Kegiatan	Durasi (Jam)	Upah	Biaya Personal	Biaya Bahan	Total Biaya
Penentuan Fitur Aplikasi	5	Rp18.000	Rp90.000	Rp-	Rp90.000
Desain Antar Muka Pengguna UI/UX	6	Rp20.000	Rp120.000	Rp20.000	Rp140.000
Perancangan Sistem dan Database	6	Rp15.000	Rp90.000	Rp-	Rp90.000
Prototyping	5	Rp16.000	Rp80.000	Rp-	Rp80.000
Total Tahap Perancangan dan Desain Aplikasi	22		Rp380.000	Rp20.000	Rp400.000
Tahap Pengembangan Aplikasi					

Kegiatan	Durasi (Jam)	Upah	Biaya Personal	Biaya Bahan	Total Biaya
Front-End					
Implementasi Antar Muka UI / UX	3	Rp10.000	Rp30.000	Rp -	Rp30.000
Integrasi Fitur Interaktif	3	Rp10.000	Rp30.000	Rp -	Rp30.000
Pengembangan Fitur Notifikasi	3	Rp10.000	Rp30.000	Rp -	Rp30.000
Back-End					
Intregasi API	42	Rp30.000	Rp30.000	Rp -	Rp30.000
Pengelolaan Data	2	Rp10.000	Rp20.000	Rp -	Rp20.000
Total Tahap Pengembangan Aplikasi	53		Rp140.000	Rp -	Rp140.000
Tahap Pengujian					
Kegiatan	Durasi (Jam)	Upah	Biaya Personal	Biaya Bahan	Total Biaya
Pengujian Aplikasi	2	Rp10.000	Rp20.000	Rp -	Rp20.000
Pengujian Sistem	2	Rp10.000	Rp20.000	Rp -	Rp20.000
Pengujian Keamanan	2	Rp10.000	Rp20.000	Rp -	Rp20.000
Pengujian Performa	2	Rp10.000	Rp20.000	Rp -	Rp20.000
Total Tahap Pengujian	8		Rp80.000	Rp -	Rp80.000
Tahap Pemeliharaan					
Kegiatan	Durasi (Jam)	Upah	Biaya Personal	Biaya Bahan	Total Biaya

Monitoring Aplikasi	2	Rp10.000	Rp20.000	Rp -	Rp20.000
Pembaruan Fitur	2	Rp10.000	Rp20.000	Rp -	Rp20.000
Perbaiki Bug	2	Rp10.000	Rp20.000	Rp -	Rp20.000
Feedback	2	Rp10.000	Rp20.000	Rp	Rp20.000
Total Tahap Perencanaan Proyek	8		Rp80.000	Rp -	Rp80.000
Total Durasi dan Biaya Proyek	112				Rp 1.500.000,00

PENUTUP

Demikian proposal penawaran ini kami ajukan. Besar harapan kami untuk dapat menjalin kerja sama yang produktif dan berkelanjutan dengan perusahaan Anda dalam mendukung pengembangan sistem informasi, khususnya melalui pembuatan Website Sistem Manajemen Ekspedisi Kargo yang terintegrasi.

Sistem ini dirancang khusus untuk membantu perusahaan dalam menyederhanakan proses pengelolaan pengiriman, mencatat seluruh pengeluaran secara otomatis, serta menyusun laporan keuangan dan operasional secara akurat. Dengan fitur-fitur yang difokuskan pada efisiensi kerja tim, pengurangan kesalahan input, serta pelaporan yang terdokumentasi dengan baik, kami percaya sistem ini dapat menjadi solusi digital yang tepat guna bagi perusahaan ekspedisi Anda.

Melalui pengaturan hak akses yang terstruktur, seluruh aktivitas dalam sistem dapat dipantau dan diaudit dengan jelas, sehingga mendorong terciptanya transparansi, akuntabilitas, dan pengawasan internal yang kuat. Hal ini menjadi pondasi penting dalam membangun kepercayaan pelanggan dan mitra, sekaligus meminimalisir potensi risiko kecurangan yang selama ini menjadi tantangan serius di industri logistik.

Kami siap berkolaborasi dengan tim internal perusahaan — baik dari sisi operasional, keuangan, maupun manajerial — guna mewujudkan sistem manajemen yang modern, efisien, dan relevan dengan kebutuhan bisnis Anda.

Terima kasih atas perhatian dan kesempatan yang diberikan. Kami menantikan langkah awal kerja sama yang membawa manfaat besar bagi kedua belah pihak.

Daerah Istimewa Yogyakarta, 20 Mei 2025

Hormat kami,

Tim GEEK SQUAD

2. MoU

Memorandum Of Understanding

Nomor: JK-mrt /001/002/2011

Yang bertanda tangan dibawah ini :

1. Nama : Muhammad Izza Mubarak
Jabatan : Project Manager
Alamat : Jl. Wiroyudo III

Dalam hal ini bertindak untuk dan atas nama **GEEK SQUAD** untuk selanjutnya disebut **Pihak Pertama**.

2. Nama : Tengku Iswandi
Jabatan : Owner
Alamat : Jl. Nogosaren Baru No.60 A, Nogosaren, Nogotirto, Kec. Gamping, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55292

Dalam hal ini bertindak untuk sebagai Client untuk selanjutnya disebut **Pihak Kedua**. Pihak Pertama dan Pihak Kedua dengan ini menerangkan pembuatan **website** (untuk selanjutnya disebut perjanjian) dengan ketentuan-ketentuan sebagai berikut:

Pasal 1**Ruang Lingkup dan Penawaran Proyek**

PIHAK PERTAMA bersedia mengembangkan Website Sistem Manajemen Operasional Perusahaan sesuai permintaan PIHAK KEDUA.

I. Fitur-fitur utama yang ditawarkan dalam proyek ini meliputi:

1. Dashboard Interaktif untuk Admin dan Tim Operasional
2. Manajemen Data Paket, termasuk tambah/edit/hapus status pengiriman
3. Manajemen Vendor dan Karyawan
4. Input & Rekapitulasi Biaya Operasional
5. Fitur Export Laporan ke Excel/PDF
6. Fitur Cetak Invoice dan Nota
7. Otentikasi User (Login/Logout dan Role Access Control)
8. Desain Responsif & User Friendly
9. Garansi teknis selama 1 bulan setelah serah terima
10. Dokumentasi Sistem jika diminta

Teknologi utama yang digunakan: Laravel (PHP Framework), MySQL, dan Tailwind CSS/Vite.

Pasal 2

Hak dan Kewajiban Para Pihak

Dalam perjanjian ini Pihak Pertama dan Pihak Kedua memiliki kewajiban:

I. Hak dan Kewajiban Pihak Pertama :

1. Melaksanakan pembuatan website sesuai spesifikasi.
2. Menyerahkan produk dalam bentuk online dan source code di akhir masa kerja.
3. Menyediakan garansi teknis sesuai kesepakatan.
4. Memberi akses demo/testing sebelum peluncuran.

II. Hak dan Kewajiban Pihak Kedua :

1. Memberikan data dan informasi pendukung tepat waktu.
2. Melakukan pembayaran sesuai kesepakatan.
3. Berhak memberi masukan dan revisi wajar selama masa pengembangan.

Pasal 3

Nilai Perjanjian

Nilai proyek disepakati sebesar:

Rp. 1.500.000 (terbilang: [Satu Juta Rupiah])

Biaya ini mencakup seluruh pengerjaan dan fitur yang disebutkan di Pasal 1.

Pasal 4

Pembayaran

Pembayaran dilakukan saat website selesai 100% dan sebelum diserahkan sepenuhnya.

Pasal 5

Waktu Pengerjaan

1. Total waktu pengerjaan maksimal 30 hari kalender setelah MoU ditandatangani.
2. Dapat diperpanjang jika PIHAK KEDUA terlambat memberi data.

Pasal 6

Garansi dan Perbaikan

1. Garansi 1 bulan setelah serah terima untuk bug atau kesalahan teknis.
2. Tidak berlaku untuk penambahan fitur baru.

Pasal 7
Force Majeur

Perjanjian ini dapat ditunda jika terjadi hal di luar kendali kedua pihak (bencana, sakit keras, dll).

Pasal 8
Ketentuan Lain

Segala perubahan kesepakatan akan dicantumkan dalam addendum dan disetujui oleh kedua belah pihak.

Pasal 9
Perselisihan

Perselisihan yang tidak bisa diselesaikan secara damai akan dibawa ke Pengadilan Negeri Yogyakarta.

Demikian perjanjian ini disepakati dan ditandatangani tanpa paksaan oleh siapapun.

Kamis, 15 Juli 2025

Pihak Pertama :
Muhammad Izza Mubarak

Pihak Kedua:
Tengku Iswandi

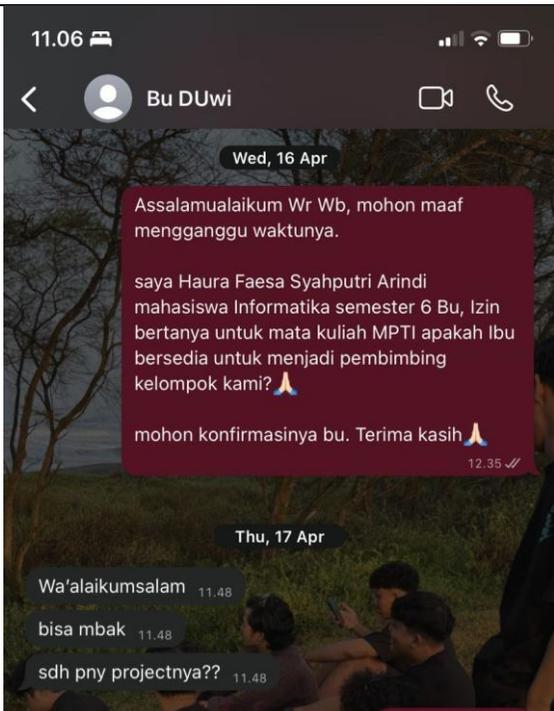
Paraf Pihak Pertama: _____

Paraf Pihak Kedua:

3. Dokumentasi

Tabel 4. 1 Dokumentasi

No.	Foto	Keterangan
1.	 A photograph showing two men standing in front of a white van. The van has "CARGO ALINDO" written on its side in blue and red letters. The man on the left is wearing a dark suit, and the man on the right is wearing a yellow t-shirt and a grey cap. They are standing in front of a building with a corrugated metal roof.	Pertemuan Pertama Di Kantor Alindo bersama client
2.	 A screenshot of a Zoom meeting interface. The main video feed shows a person with glasses and a dark shirt. The background of the video feed is a scenic view of the Golden Gate Bridge at sunset. The Zoom interface includes a top bar with "Zoom Workplace Meeting View Edit Window Help", a top right corner with "Recording..." and "View", and a bottom toolbar with "Audio", "Video", "Participants", "Chat", "React", "Share", "Apps", "Record", "More", and "Leave". A WhatsApp chat window is visible in the bottom right corner of the screenshot.	Meeting kedua dengan Client Melalui Zoom

3.		Pertemuan ketiga dengan client untuk memfixkan fitur aset
4.	 <p>11.06</p> <p>Bu DUwi</p> <p>Wed, 16 Apr</p> <p>Assalamualaikum Wr Wb, mohon maaf mengganggu waktunya.</p> <p>saya Haura Faesa Syahputri Arindi mahasiswa Informatika semester 6 Bu, Izin bertanya untuk mata kuliah MPTI apakah Ibu bersedia untuk menjadi pembimbing kelompok kami? 🙏</p> <p>mohon konfirmasinya bu. Terima kasih 🙏</p> <p>12.35 ✓</p> <p>Thu, 17 Apr</p> <p>Wa'alaikumsalam 11.48</p> <p>bisa mbak 11.48</p> <p>sdh pny projectnya?? 11.48</p>	Proses Pencarian Dosen pembimbing

5.



PT. ABIF LOGISTIK INDONESIA
 Jl. Nogosaren Baru No.60A Nogoirta
 Camping Sleman Jogjakarta
 Email : aliflogistikindonesia@gmail.com

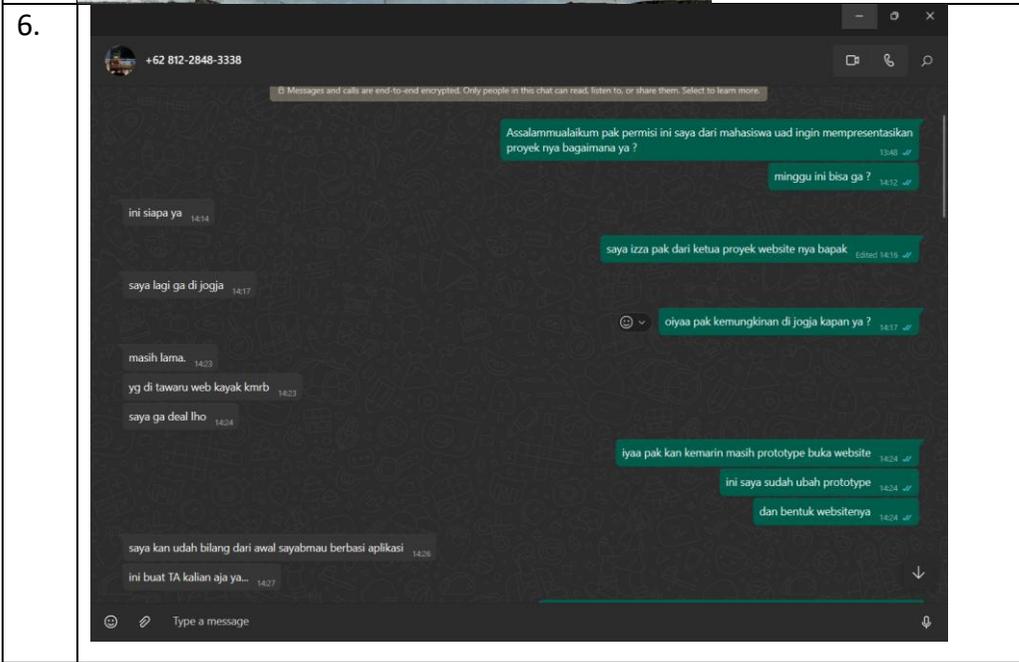
SURAT TANDA TERIMA

135 - 230850

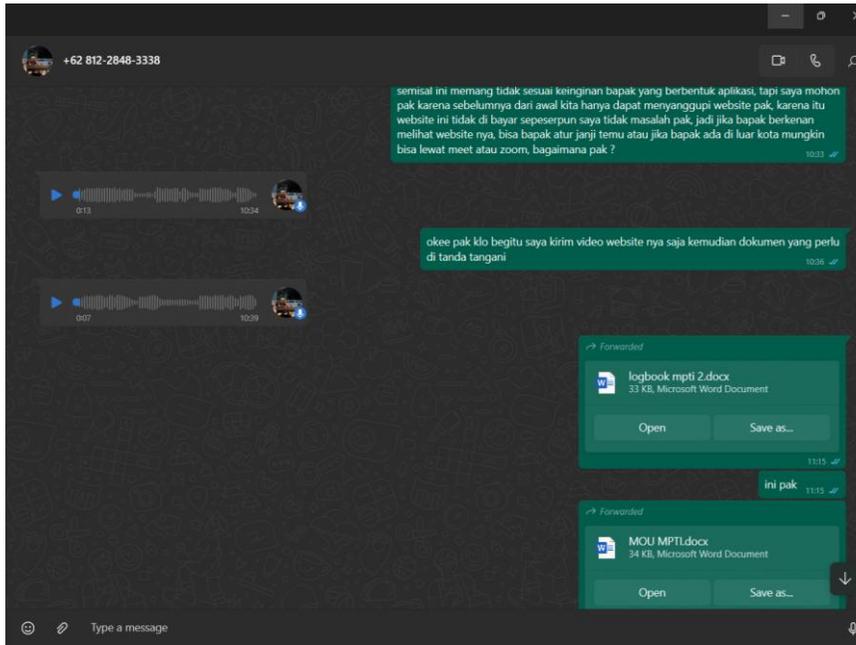
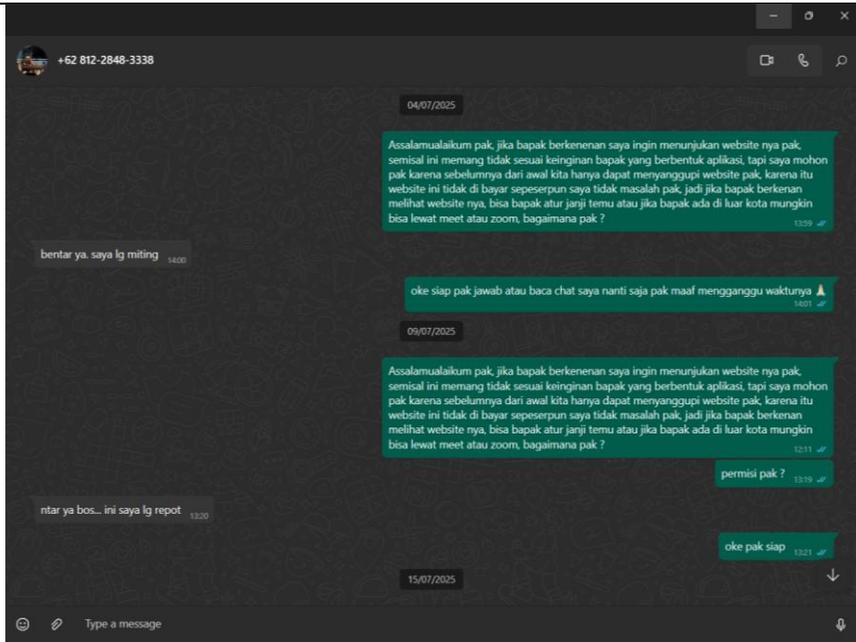
<input type="checkbox"/> DARAT <input type="checkbox"/> LAUT <input type="checkbox"/> UDARA <input type="checkbox"/> PORT TO PORT <input type="checkbox"/> DOOR TO DOOR KOTA ASAL : KOTA TUJUAN : TANGGAL KIRIM : UMLAH KOLI : ERAT : ERAT VOLUME : ISJ BARANG	PENGIRIM : No HP : PENERIMA : ALAMAT : No HP :	BEA KIRIM : ADM : PACKING : PICK UP : ASURANSI : LAIN - LAIN : TOTAL : NILAI BARANG YANG DI ASURANSIKAN ISJ BARANG <input type="checkbox"/> DIPERIKSA <input type="checkbox"/> TIDAK DIPERIKSA
--	--	---

PETUGAS () () () ()	PERNYATAAN PENGIRIM Kami menyatakan bahwa kandungan & isi barang pengiriman barang yang terdapat di lembar ini adalah benar & kami menyetujuinya PENGIRIM	PERNYATAAN PENERIMA Kiriannya ini saya / kami terima dengan baik & benar PENERIMA	KETENTUAN PENGIRIMAN 1. Tibaan paket / barang akibat force majeure (demonsi alam, pemampukan, kecelakaan, kebakaran) di luar tanggung jawab kami. 2. Isi Tibaan rusak / hilang akibat kemasan packing yang kurang baik menjadi tanggung jawab pengirim. 3. Isi Tibaan rusak pada waktu penerimaan harus dilakukan oleh pengirim kami dan dinyatakan secara tertulis atau dibuatkan berita acara. 4. Kami tidak bisa atau no to tidak bisa dibuang jika ada keterlambatan penanganan maka akan tanggung jawab pengirim. 5. Pengirim tidak bisa mena garansi rugi akibat keterlambatan proses pengiriman dengan waktu yang sudah ditetapkan. 6. Jika dalam waktu 1 (satu) bulan tidak ada pengakuan, kami dianggap sudah diterima dengan baik. 7. Garansi rugi yang dibebaskan karena rusak/hilang max 10 kali dari biaya kirim atau maksimal 1 (satu) juta rupiah. 8. Biaya kirim belum termasuk biaya asuransi barang.
----------------------------	---	---	--

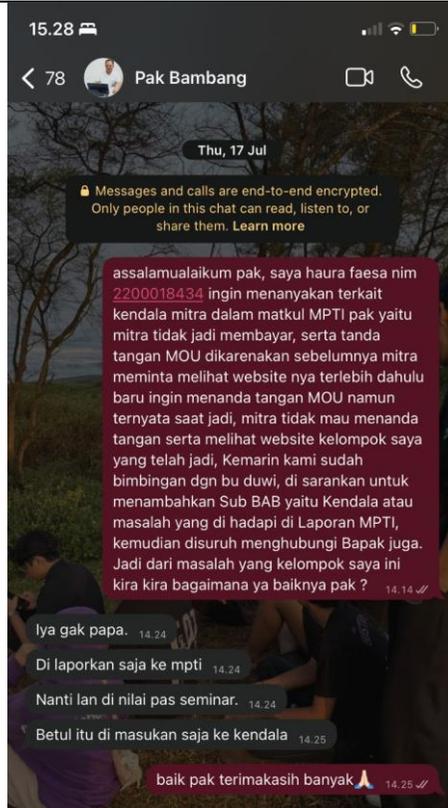
Invoice bisnis sebelum ada sistem



Bukti kendala pada klien kronologi secara chat atau dialog di aplikasi *Whatsapp*



7.



Pelaporan kendala kepada pak bambang selaku koordinator

MPTI

