

LAMPIRAN

Lampiran 1. Surat Kesepakatan Bimbingan Skripsi

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Arden Arjuna Mulvi
NIM : 21090128
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : Dega Surono Wibowo, S.T., M.Kom.
Status : Dosen
NIDN : 0607108202
Jabatan Fungsional : Lektor
Pangkat/Golongan : Penata /III/C

Pada hari ini Senin tanggal 17 Maret 2025 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Skripsi Pihak Pertama dengan syarat melakukan bimbingan 1 kali dalam seminggu atau setidaknya 3 kali bimbingan dalam satu bulan (dengan progress), apabila saya tidak memenuhi syarat tersebut maka saya tidak berhak meminta surat rekomendasi mengikuti sidang skripsi. Dan saya juga berjanji akan memenuhi persyaratan tersebut dan menyelesaikan skripsi baik produk maupun laporan tepat waktu.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

Tegal, 17 Maret 2025

Pihak Pertama



Arden Arjuna Mulvi

Pihak Kedua



Dega Surono Wibowo, S.T., M.Kom.
NIPY: 06.014.183

Mengetahui
Ketua Program Studi Sarjana Terapan
Teknik Informatika



Dyah Apriliani, S.T., M.Kom.
NIPY: 09.015.225

Lampiran 2. Surat Kesepakatan Bimbingan Tugas Akhir

SURAT KESEPAKATAN BIMBINGAN TUGAS AKHIR

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Arden Arjuna Mulvi
NIM : 21090128
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : Taufiq Abidin, S.Pd., M.Kom.
Status : Dosen
NIDN : 0603088305
Jabatan Fungsional : Lektor
Pangkat/Golongan : Penata /III/C

Pada hari ini Senin tanggal 17 Maret 2025 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing II Skripsi Pihak Pertama dengan syarat Pihak Pertama wajib melakukan bimbingan Skripsi minimal 8 kali kepada Pihak Kedua. Adapun waktu dan tempat pelaksanaan hari Rabu pukul 15.30 di ruang dosen.

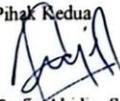
Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

Tegal, 17 Maret 2025

Pihak Pertama


Arden Arjuna Mulvi

Pihak Kedua


Taufiq Abidin, S.Pd., M.Kom.
NIPY: 06.014.184

Mengetahui
Ketua Program Studi Sarjana Terapan Teknik
Informatika


Dyah Anelham, S.T., M.Kom.
NIPY: 09.015.225

Lampiran 3. Surat Pengalihan Hak Cipta

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini :

1. Nama : Arden Arjuna Mulvi
Kewarganegaraan : Indonesia
Alamat : Jln.IR.H.Juanda No 91 RT.02 RW.04, Kecamatan Slawi, Kabupaten Tegal, Jawa Tengah, 52415
2. Nama : Dega Surono Wibowo, S.T., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Perumahan Sapphire Regency Blok H, No 1, RT.004, RW.001, Kelurahan Pulosari, Kecamatan Brebes, 52213
3. Nama : Taufiq Abidin, S.Pd., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Jl. Kaligangsa Asri Timur VIII No.33 RT. 3 RW. 7 Perum Kaligangsa Asri Kelurahan Kaligangsa, Kecamatan Margadana, Kota Tegal, 52147

Adalah Pihak I selaku pencipta, dengan ini menyerahkan karya ciptaan saya kepada :

Nama : Pusat Penelitian dan Pengabdian Masyarakat (P3M)
Alamat : Jl. Mataram No. 9 Pesurungan Lor Tegal

Adalah Pihak II selaku Pemegang Hak Cipta berupa Program Komputer dengan judul "Rancang Bangun Aplikasi dengan Metode Payment Gateway dan Customer Complaint Handling Model (CCHM) pada CV.Evan Network". untuk didaftarkan di Direktorat Hak Cipta dan Desain Industri, Direktorat Jenderal Kekayaan Intelektual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.

Demikianlah surat pengalihan hak ini kami buat, agar dapat dipergunakan sebagaimana mestinya.

Pemegang Hak Cipta
Kepala P3M


(Muhammad Fikri Hidayatullah, S.T., M.Kom.)

Tegal, 7 Juli 2025
Pencipta



(Arden Arjuna Mulvi)

(Dega Surono Wibowo, S.T., M.Kom.)

(Taufiq Abidin, S.Pd., M.Kom.)

Lampiran 4 Surat Pernyataan HKI

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini :

1. Nama : Arden Arjuna Mulvi
Kewarganegaraan : Indonesia
Alamat : Jln.IR.H.Juanda No 91 RT.02 RW.04, Kecamatan Slawi, Kabupaten Tegal, Jawa Tengah, 52415

2. Nama : Dega Suroño Wibowo, S.T., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Perumahan Sapphire Regency Blok H, No. 1, RT.004, RW.001, Kelurahan Pulosari, Kecamatan Brebes, 52213

3. Nama : Taufiq Abidin, S.Pd., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Jl. Kaligangsa Asri Timur VIII No.33 RT. 3 RW. 7 Perum Kaligangsa Asri Kelurahan Kaligangsa, Kecamatan Margadana, Kota Tegal, 52147

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:
Berupa : Program Komputer
Berjudul : Rancang Bangun Aplikasi dengan Metode Payment Gateway dan Customer Complaint Handling Model (CCHM) pada CV.Evan Network
 - Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyek kekayaan intelektual lainnya sebagaimana dimaksud dalam Pasal 68 ayat (2);
 - Bukan merupakan Ekspresi Budaya Tradisional sebagaimana dimaksud dalam Pasal 38;
 - Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
 - Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;
 - Bukan merupakan Ciptaan seni lukis yang berupa logo atau tanda pembeda yang digunakan sebagai merek dalam perdagangan barang/jasa atau digunakan sebagai lambang organisasi, badan usaha, atau badan hukum sebagaimana dimaksud dalam Pasal 65 dan;
 - Bukan merupakan Ciptaan yang melanggar norma agama, norma susila, ketertiban umum, pertahanan dan keamanan negara atau melanggar peraturan perundang-undangan sebagaimana dimaksud dalam Pasal 74 ayat (1) huruf d Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

2. Sebagai pemohon mempunyai kewajiban untuk menyimpan asli contoh ciptaan yang dimohonkan dan harus memberikan apabila dibutuhkan untuk kepentingan penyelesaian sengketa perdata maupun pidana sesuai dengan ketentuan perundang-undangan.

3. Karya Cipta yang saya mohonkan pada Angka 1 tersebut di atas tidak pernah dan tidak sedang dalam sengketa pidana dan/atau perdata di Pengadilan.

4. Dalam hal ketentuan sebagaimana dimaksud dalam Angka 1 dan Angka 3 tersebut di atas saya / kami langgar, maka saya / kami bersedia secara sukarela bahwa:
 - a. permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau
 - b. Karya Cipta yang telah terdaftar dalam Daftar Umum Ciptaan Direktorat Hak Cipta, Direktorat Jenderal Hak Kekayaan Intelektual, Kementerian Hukum Dan Hak Asasi Manusia R.I dihapuskan sesuai dengan ketentuan perundang-undangan yang berlaku.

- c. Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam berperkara dan/atau sedang dalam gugatan di Pengadilan maka status kepemilikan surat pencatatan elektronik tersebut ditangguhkan menunggu putusan Pengadilan yang berkekuatan hukum tetap.

Demikian Surat pernyataan ini saya/kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Tegal, 7 Juli 2025



(Arden Arjuna Mulvi)


(Dega Surono Wigow, S.T., M.Kom.)


(Taufiq Abidin, S.Pd., M.Kom.)

Aplikasi Web CCHM CV. Evan Network

Sistem Pengajuan, Pembayaran, Pengaduan Wifi Evan Net

x x x x x

Arden Arjuna Mulvi

Dega Surono Wibowo, S.T., M.Kom
Topik Abidin, S.Pd., M.Kom



Manual book ini disusun sebagai panduan penggunaan Aplikasi Pelayanan WiFi Berbasis Website pada CV. Evan Network. Aplikasi ini dikembangkan untuk mempermudah pelanggan dalam melakukan transaksi pembayaran layanan WiFi secara online serta memfasilitasi penyampaian keluhan secara efektif melalui fitur Customer Complaint Handling Model (CCHM). Dengan mengintegrasikan sistem Payment Gateway, aplikasi ini memungkinkan proses pembayaran dilakukan secara otomatis, aman, dan praktis tanpa perlu keterlibatan manual yang berisiko menimbulkan kesalahan pencatatan.

Selain itu, aplikasi ini dirancang dengan antarmuka yang ramah pengguna agar pelanggan dapat mengakses berbagai fitur dengan mudah. Bagi admin, aplikasi menyediakan sistem pencatatan dan pelaporan keuangan yang akurat dan real-time guna mendukung efisiensi operasional serta pengambilan keputusan. Melalui manual book ini, pengguna diharapkan dapat memahami cara kerja aplikasi secara menyeluruh, baik sebagai pelanggan maupun admin, sehingga proses pelayanan menjadi lebih cepat, efisien, dan terstruktur.



Gambar Logo CV. Evan Network

Aplikasi Pelayanan WiFi CV. Evan Network berbasis website dapat diakses dengan mudah melalui browser tanpa perlu instalasi tambahan. Pengguna cukup membuka URL aplikasi menggunakan perangkat yang kompatibel dan koneksi internet yang stabil.

Untuk mendukung pengalaman penggunaan yang optimal, berikut adalah perangkat dan spesifikasi minimum yang direkomendasikan:

1. Perangkat yang Didukung

Aplikasi Pelayanan WiFi CV. Evan Network dapat diakses melalui berbagai perangkat dengan koneksi internet yang stabil. Karena berbasis web, pengguna tidak perlu melakukan instalasi tambahan, cukup menggunakan browser yang kompatibel.

Spesifikasi Perangkat Minimum yang Direkomendasikan:

- a. Perangkat: Smartphone, tablet, laptop, atau komputer
- b. Sistem Operasi: Android 8.0 ke atas, iOS 12 ke atas, Windows 10 ke atas, atau macOS
- c. Browser: Google Chrome, Mozilla Firefox, Safari, atau Microsoft Edge (versi terbaru)
- d. Resolusi Layar: Disarankan minimal 1366 x 768 untuk tampilan optimal
- e. Koneksi Internet: Stabil (minimal jaringan 3G; disarankan 4G atau WiFi)

2. Akses Aplikasi

Aplikasi dapat digunakan langsung melalui browser web tanpa perlu instalasi.

- a. URL Akses: <https://evannetwork.example.com> (ganti sesuai domain resmi)
- b. Catatan: Pastikan koneksi internet aktif dan browser dalam versi terbaru untuk pengalaman terbaik.

Saat Anda pertama kali mengakses aplikasi layanan WiFi CV. Evan Network melalui browser, Anda akan disambut oleh tampilan halaman utama (homepage) yang menyerupai landing page modern. Halaman ini menampilkan pesan selamat datang seperti:

"Selamat Datang di Evan Network – Nikmati Internet Cepat, Stabil, dan Terjangkau untuk Kebutuhan Anda!"

Tampilan ini dirancang untuk memberikan kesan profesional dan menarik, dengan banner utama berisi slogan layanan, gambar ilustratif, serta tombol ajakan bertindak (Call to Action) seperti “Lihat Paket WiFi” atau “Langganan Sekarang”.

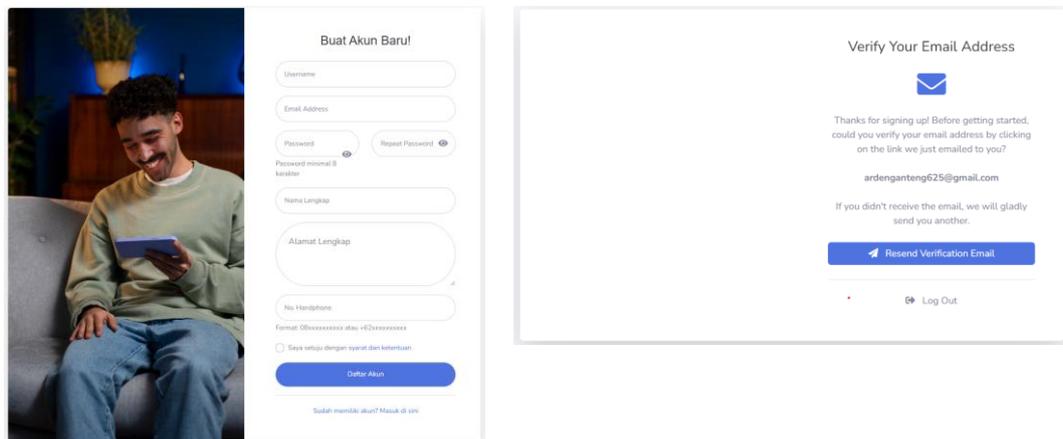


Di bawah banner utama, pengguna dapat langsung melihat daftar paket WiFi yang tersedia, lengkap dengan detail harga, kecepatan internet, dan fitur-fitur tambahan. Pengguna tidak perlu login untuk menjelajahi informasi ini. Setiap paket juga disertai tombol seperti “Pilih Paket” atau “Lihat Detail” yang akan mengarahkan ke halaman registrasi atau pembayaran.

1. User

a. Registrasi dan Login

- 1) Klik Tombol Buat Akun Baru di halaman Login
- 2) Isi form registrasi dengan informasi yang diperlukan:
 - Username
 - Email
 - Kata Sandi
 - Nama Lengkap
 - Nomor HP
- 3) Klik “Daftar Akun” untuk menyelesaikan registrasi
- 4) Tunggu verifikasi Email

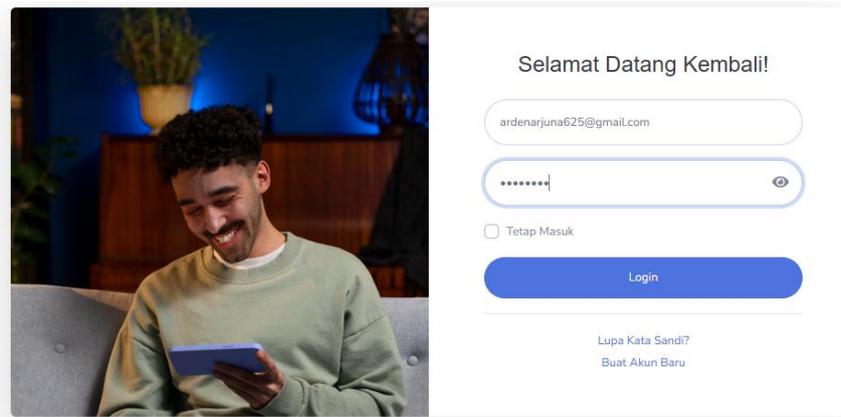


5) Setelah Selesai verifikasi lanjut ke Halaman Login

6) Cara Login :

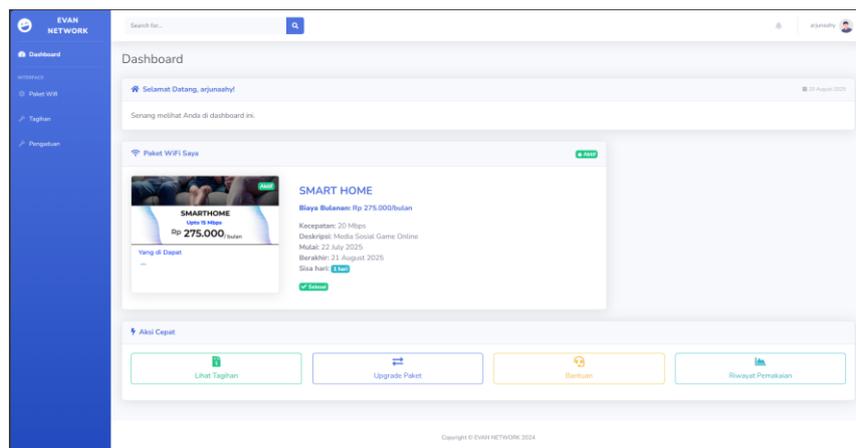
- Buka halaman login Pengguna
- Masukkan Email dan Kata Sandi yang telah terdaftar
- Klik tombol “Login”

- Jika berhasil, akan diarahkan ke dashboard pengguna



b. Dashboard Pengguna

Setelah berhasil login, pengguna akan diarahkan ke halaman Dashboard yang menampilkan antarmuka utama dengan fitur-fitur utama aplikasi layanan WiFi CV. Evan Network.



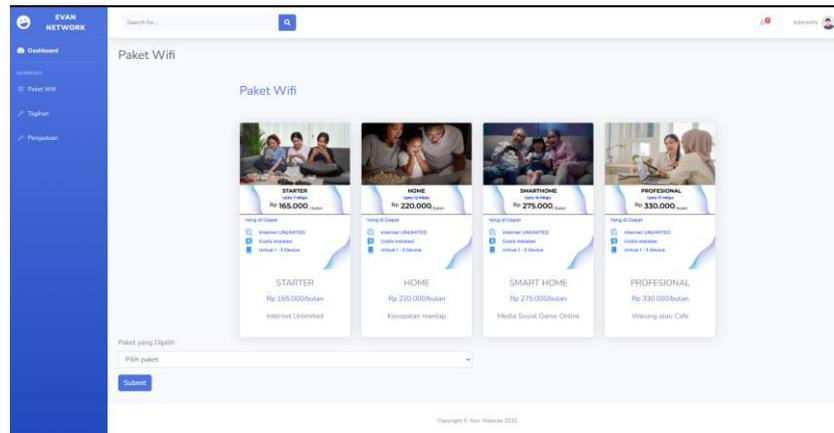
1) Sapaan Personal: Menampilkan ucapan selamat datang, misalnya:

“Selamat Datang, arjunaahy”

2) Menu Sidebar: Menyediakan akses cepat ke tiga fitur utama:

- Paket WiFi: Untuk melakukan pengajuan layanan internet.
- Tagihan: Untuk melihat dan membayar tagihan bulanan.
- Pengaduan: Untuk mengirimkan keluhan atau laporan masalah.

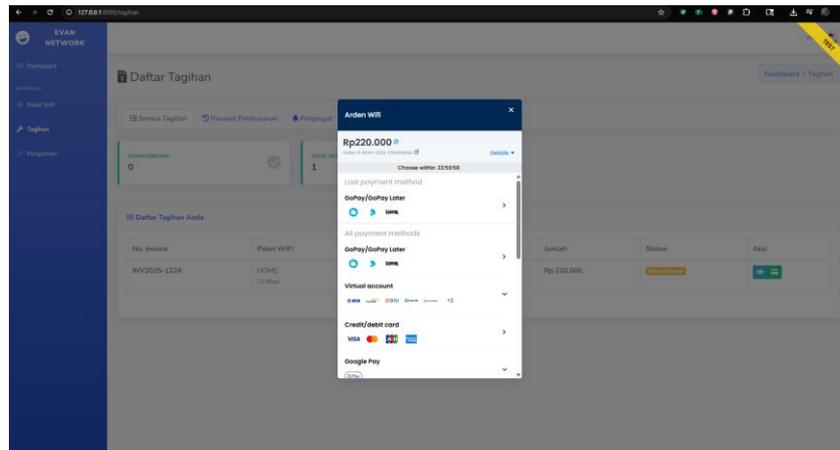
c. Pengajuan Paket WiFi



Pengguna dapat mengajukan permintaan berlangganan paket WiFi melalui menu "Paket WiFi" di sidebar. Langkah-langkah Pengajuan Paket:

- 1) Klik menu “Paket WiFi” pada sidebar.
- 2) Tinjau seluruh paket yang tersedia di halaman.
- 3) Pelajari detail dari masing-masing paket, seperti:
 - Nama Paket: STARTER, HOME, SMART HOME, PROFESIONAL
 - Kecepatan internet
 - Harga bulanan
 - Fitur tambahan
- 4) Pilih paket yang diinginkan melalui dropdown “Paket yang Dipilih”.
- 5) Klik tombol “Submit” untuk mengajukan permintaan.
- 6) Tunggu konfirmasi dari pihak Admin.
- 7) Daftar Paket yang Tersedia:
 - STARTER: Kecepatan 7 Mbps – Paket dasar
 - HOME: Kecepatan 12 Mbps – Untuk penggunaan rumahan
 - SMART HOME: Kecepatan 20 Mbps – Paket premium
 - PROFESIONAL: Untuk kebutuhan bisnis

d. Pembayaran Tagihan (Payment Gateway)



- 1) Menu “Tagihan” memungkinkan pengguna mengelola dan melakukan pembayaran tagihan dengan mudah. Langkah-langkah Pembayaran:
- 2) Klik menu “Tagihan” pada sidebar.
- 3) Anda akan diarahkan ke halaman Dashboard Tagihan, yang terdiri dari tiga tab:
 - Semua Tagihan
 - Riwayat Pembayaran
 - Penunggak
- 4) Pada bagian tabel “Daftar Tagihan Anda”, informasi tagihan ditampilkan secara rinci, meliputi:
 - Nomor Invoice
 - Nama Paket
 - Periode
 - Jatuh Tempo
 - Jumlah Tagihan
 - Status Pembayaran
 - Aksi (tindakan)
- 5) Contoh:
 - Nomor Invoice: INV2025-1224
 - Nama Paket: HOME
 - Jumlah Tagihan: Rp 220.000
 - Status: Belum Dibayar

- 6) Klik tombol “Bayar” pada baris tagihan yang ingin Anda lunasi.
- 7) Muncul popup pembayaran “Evan Network” yang menampilkan nominal tagihan.
- 8) Pilih metode pembayaran yang tersedia, seperti:
 - GoPay / GoPay Later
 - Virtual Account (BCA, Mandiri, BNI, Permata, dll)
 - Kartu Kredit / Debit (Visa, Mastercard, JCB, AmEx)
 - Google Pay
- 9) Sistem akan menampilkan hitung mundur waktu pembayaran, misalnya:
“Choose within 23:59:58”

e. Pengajuan Keluhan (Pengaduan)

Halaman Pengaduan merupakan antarmuka yang disediakan untuk menerima dan mengelola laporan keluhan atau masalah dari pengguna. Halaman ini dapat diakses melalui menu "Pengaduan" pada sidebar.

Langkah-langkah Penggunaan:

1) Mengakses Form Pengaduan.

Pengguna masuk ke halaman pengaduan dan akan melihat Form Pengaduan

2) Memilih Jenis Keluhan.

Pengguna diminta memilih jenis keluhan melalui dropdown bertuliskan placeholder:

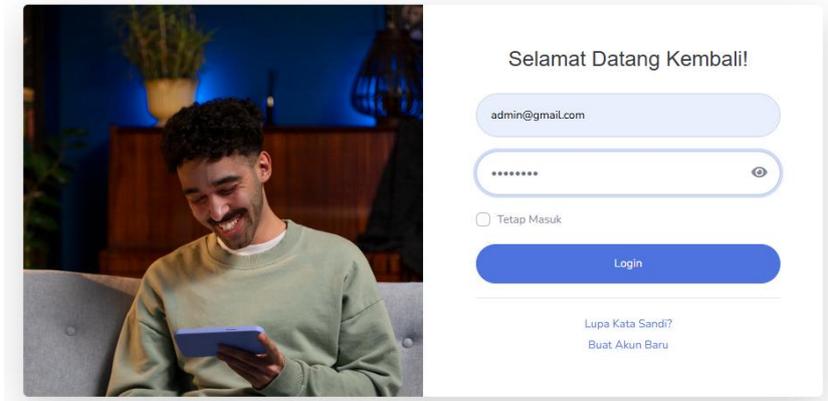
- “Pilih jenis keluhan”
- Dropdown ini berisi beberapa kategori masalah, seperti:
 - Gangguan koneksi
 - Tagihan tidak sesuai
 - Permintaan teknisi
 - Lainnya

3) Mengirimkan Pengaduan.

Setelah memilih jenis keluhan, pengguna dapat langsung menekan tombol “Kirim Pengaduan” untuk mengirim laporan tersebut ke sistem.

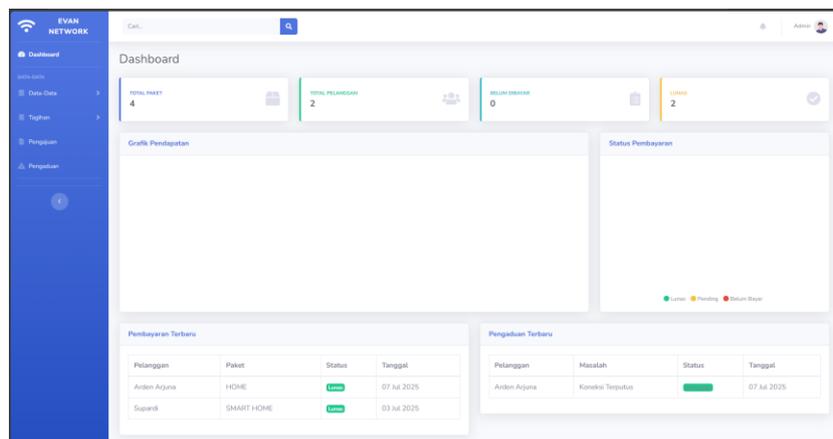
2. Admin

a. Login Admin



- 1) Buka Halaman login Admin Melalui Browser.
- 2) Masukkan:
 - Email yang telah terdaftar sebagai Admin
 - Kata Sandi sesuaikan akun Admin
- 3) Klik tombol “Login”
- 4) Jika login berhasil, Anda akan diarahkan ke halaman Dashboard Admin

b. Dashboard Admin



Setelah berhasil login, Admin akan diarahkan ke halaman Dashboard Admin. Pada halaman ini, terdapat dua bagian utama:

- 1) Ringkasan Statistik (Bagian Atas Dashboard)

Terdapat 4 kotak informasi utama:

- Total Paket

- Total Pelanggan
- Tagihan Belum Dibayar
- Tagihan Lunas

2) Tabel Pembayaran Terbaru (Bagian Bawah Dashboard)

Menampilkan data transaksi pelanggan terbaru dengan kolom:

- Nama Pelanggan
- Paket WiFi
- Status Pembayaran
- Tanggal Pembayaran
- Status Pengaduan

3) Tabel Pengaduan Terbaru

Menampilkan laporan pengaduan terbaru dari pelanggan, meliputi:

- Pelanggan
- Masalah
- Status
- Tanggal

c. Kelola Data Paket WiFi

Tampilan Halaman Data Paket WiFi. Menampilkan daftar seluruh paket yang tersedia, meliputi: ID Paket, Nama Paket, Kecepatan, Harga Bulanan, Deskripsi Paket, Gambar Paket, Aksi (Edit dan Hapus)

ID Paket	Nama Paket	Kecepatan	Harga Bulanan	Deskripsi Paket	Gambar	Aksi
5	STARTER	7 Mbps	Rp 145.000	Internet Unlimited		Edit Hapus
7	HOME	12 Mbps	Rp 220.000	Kecepatan Smartlap		Edit Hapus
8	SMART HOME	20 Mbps	Rp 275.000	Media Sosial Game Online		Edit Hapus

Tambah Data Paket:

- 1) Klik tombol “Tambah Paket” yang terletak di kanan atas tabel.
- 2) Isi form data paket yang terdiri dari:
 - Nama Paket
 - Kecepatan (misal: 10 Mbps)
 - Harga Bulanan
 - Deskripsi Paket
 - Upload Gambar Paket (opsional)
- 3) Klik tombol “Simpan” untuk menyimpan data dan menambahkan paket ke dalam sistem.

Edit Paket :

- 1) Pada tabel, cari paket yang ingin diedit.
- 2) Klik tombol Edit pada baris paket tersebut.
- 3) Ubah data sesuai kebutuhan.
- 4) Klik tombol Update/Simpan.

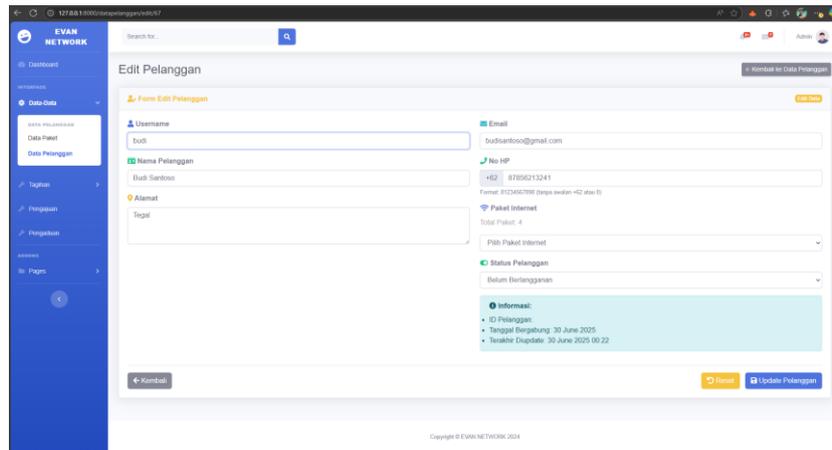
Hapus Paket :

- 1) Pada tabel, cari paket yang ingin dihapus.
- 2) Klik tombol Hapus di kolom Aksi.
- 3) Konfirmasi penghapusan jika muncul popup.

Generate Report

- 1) Klik tombol Generate Report untuk membuat laporan data paket dalam bentuk PDF atau Excel (tergantung implementasi sistem).
- 2) File akan terunduh atau tampil di tab baru.

d. Kelola Data Pelanggan



Halaman Awal kelola daa pelanggan :

- 1) Halaman akan menampilkan daftar pelanggan dalam bentuk tabel.
- 2) Untuk mencari data pelanggan:
 - Ketik nama, username, email, atau nomor HP di kolom pencarian.
 - Tekan ikon cari atau tombol Enter.
- 3) Untuk menyaring berdasarkan status:
 - Pilih status dari dropdown (Semua Status / Menunggu / Selesai).
 - Klik tombol Reset untuk mengembalikan tampilan ke semula.

Tambah data pelanggan :

- 1) Klik tombol + Tambah Pelanggan di kanan atas.
- 2) Isi form pelanggan:
 - Username
 - Nama Lengkap
 - Alamat
 - Email
 - Nomor HP
 - Pilih Paket Langganan
- 3) Klik Simpan untuk menyimpan data pelanggan.
- 4) Mengelola Data Pelanggan

Edit Data Pelanggan :

- 1) Klik tombol Edit
- 2) Ubah data
- 3) Lalu klik simpan

Hapus Data Pelanggan :

- 1) Klik tombol hapus
- 2) Konfirmasi data dan klik “Yes”
- 3) Data terhapus

e. Kelola Data Tagihan

ID Tagihan	Nama Pelanggan	Paket WiFi	Periode	Jumlah Tagihan	Jumlah Tempo	Status Tagihan	Aksi
1225	Andan Arjuna	10 Mbps	Aug 2025 10/08/2025	Rp 220.000	10/08/2025	Lunas	Detail, Edit, Hapus
1223	Supardi	20 Mbps	Aug 2025 10/08/2025	Rp 275.000	10/08/2025	Lunas	Detail, Edit, Hapus

Filter Data Tagihan:

- 1) Filter Berdasarkan Status: Pilih dari dropdown: Semua Status, Lunas, Belum Dibayar.
- 2) Filter Berdasarkan Bulan: Pilih bulan tagihan dari dropdown (misal: Agustus 2025).
- 3) Cari Pelanggan: kolom pencarian untuk mencari berdasarkan nama pelanggan.
- 4) Klik tombol Cari untuk menerapkan pencarian/filter.

Tombol Aksi Tagihan :

- 1) Detail lihat fungsi tagihan : Klik detail untuk melihat informasi lengkap
- 2) Edit ubah data tagihan : Klik edit ubah info, lalu simpan

3) Hapus tagihan : Klik hapus lalu konfirmasi penghapusan

Tambah Data Tagihan Manual :

- 1) Klik tombol Buat tagihan manual
- 2) Pilih pelanggan dan periode.
- 3) Masukkan jumlah tagihan dan jatuh tempo.
- 4) Klik Simpan untuk membuat tagihan secara manual.

Edit dan Hapus Tagihan

Fitur **Edit** dan **Hapus** pada data tagihan dilakukan dengan cara yang sama seperti pada pengelolaan **Data yang lain**.

f. Kelola Data Pengajuan

ID Pengajuan	Nama Pelanggan	Alamat Pengajuan	No Telp	Email	Paket WiFi	Status	Aksi
60	Andri Arcana	Slawi	087924034008	andriarcana28@gmail.com	HOME	Select	Detail Edit Hapus
70	Supardi	Bibis	087981263123	supardi@gmail.com	SMART HOME	Select	Detail Edit Hapus

1) Menampilkan daftar permintaan pemasangan WiFi dari calon pelanggan, dengan kolom sebagai berikut:

- ID Pengajuan
- Nama Pelanggan
- Alamat Pengajuan
- No Telp
- Email
- Paket WiFi
- Status
- Aksi

- 2) Dropdown Status mengubah status pengajuan : Ubah dari Menunggu ke Selesai, sistem akan otomatis membuat data pelanggan dan tagihan

Tombol Aksi Pengajuan :

- 1) Detail lihat fungsi pengajuan : Klik detail untuk melihat informasi lengkap
- 2) Edit ubah data tagihan : Klik edit ubah info, lalu simpan
- 3) Hapus tagihan : Klik hapus lalu konfirmasi penghapusan

g. Kelola Data Pengaduan

ID Pengaduan	Nama	Alamat Lengkap	Jenis Keluhan	Tanggal Pengaduan	Status	Aksi
17	Ardan Ajuna	Sirel	Kontrak Terputus	07-01-2025 14:02	Menunggu	Hapus

1) Tabel Tagihan Pengaduan

Menampilkan daftar keluhan pelanggan terkait layanan WiFi, dengan kolom berikut:

- ID Pengajuan
- Nama
- Alamat Lengkap
- Jenis Keluhan
- Tanggal Pengaduan
- Status
- Aksi

2) Mengelola Status Pengaduan

Pada Kolom Status, Admin dapat memilih status dari dropdown:

- Menunggu → Pengaduan baru diterima
- Diproses → Sedang ditindaklanjuti oleh teknis
- Selesai → Masalah telah diselesaikan

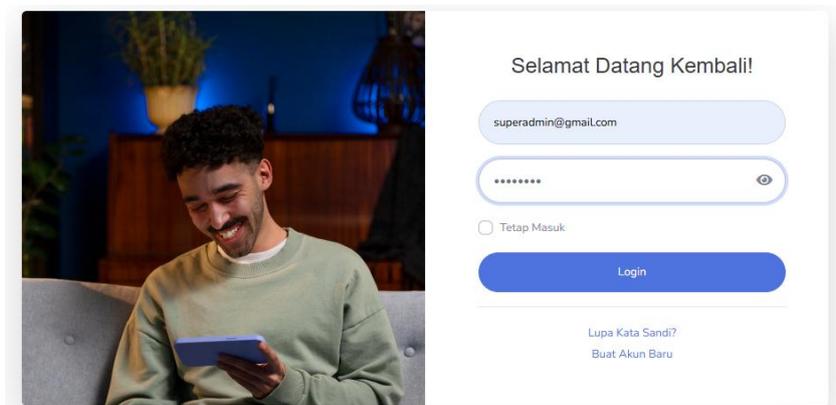
3) Menghapus Pengaduan

Klik tombol Hapus berwarna merah pada baris pengaduan.

- Akan muncul popup konfirmasi.
- Klik Ya/Hapus untuk menghapus data pengaduan secara permanen.

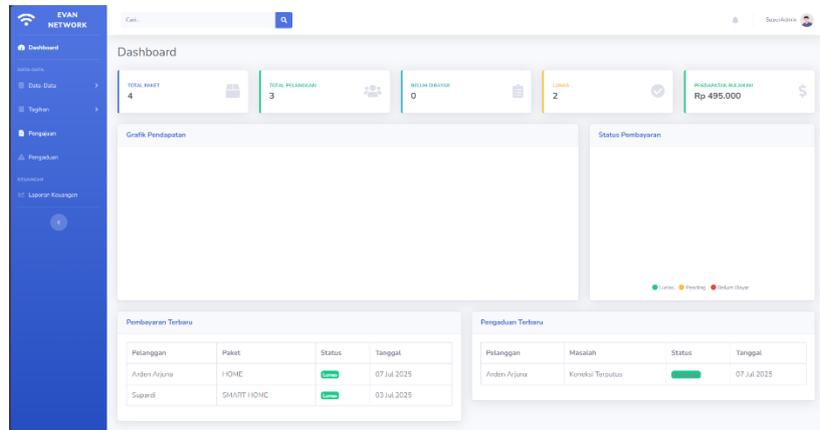
3. Super Admin

a. Login Super Admin



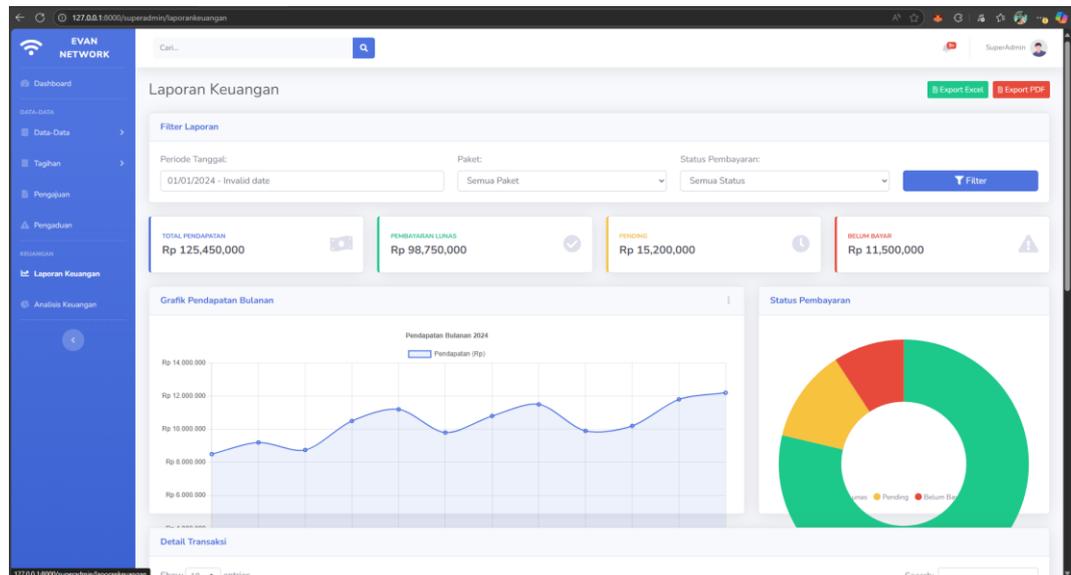
- 1) Buka Halaman login Admin Melalui Browser.
- 2) Masukkan:
 - Email yang telah terdaftar sebagai Super Admin
 - Kata Sandi sesuaikan akun Super Admin
- 3) Klik tombol “Login”
- 4) Jika login berhasil, Anda akan diarahkan ke halaman Dashboard Admin

b. Dashboard Super Admin



Setelah berhasil login, Super Admin akan diarahkan ke halaman Dashboard Super Admin, dimana dashboard hampir sama dengan Dashboard Admin. Pembedanya pada bagian pendapatan bulanan saja.

c. Kelola Laporan Keuangan



Super Admin dapat mengakses halaman Laporan Keuangan untuk melihat dan menganalisis transaksi secara menyeluruh. Berikut panduannya:

- 1) Masuk ke menu Laporan Keuangan dari sidebar.
- 2) Anda akan melihat total pendapatan serta status pembayaran pelanggan dalam bentuk grafik dan diagram.

3) Untuk melihat data yang lebih spesifik:

- Gunakan filter tanggal untuk menampilkan data dari periode tertentu.
- Pilih paket WiFi tertentu jika ingin menganalisis pendapatan per paket.
- Gunakan filter status pembayaran untuk menampilkan data berdasarkan kondisi (lunas, pending, belum dibayar).

4) Perhatikan grafik batang yang menunjukkan pendapatan bulanan.

5) Amati juga diagram pie yang memperlihatkan proporsi status pembayaran.

6) Untuk keperluan pelaporan atau dokumentasi:

- Klik tombol Ekspor Excel jika ingin menyimpan data dalam format .xlsx.
- Klik tombol Ekspor PDF jika ingin mencetak laporan dalam bentuk file PDF.

4. **Teknisi**

a. **Login Teknisi**

Teknisi dapat login melalui halaman login menggunakan akun yang telah didaftarkan oleh admin.

Langkah-langkah:

1) Buka Halaman login Admin Melalui Browser.

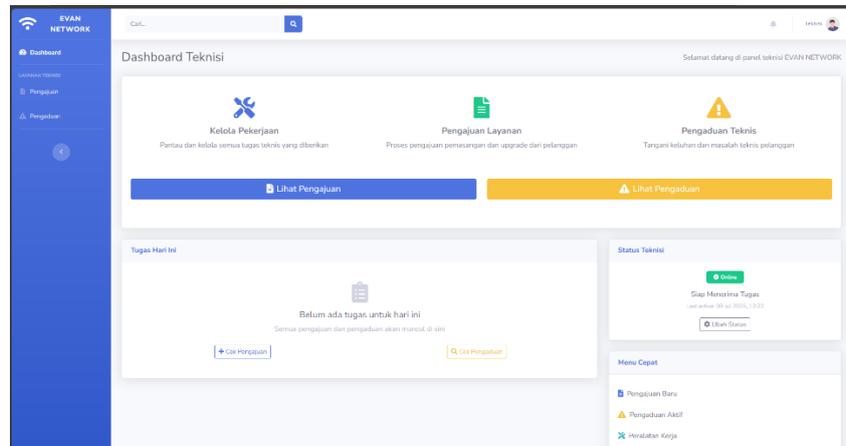
2) Masukkan:

- Email yang telah terdaftar sebagai Teknisi
- Kata Sandi sesuaikan akun Teknisi

3) Klik tombol “Login”

4) Jika login berhasil, Anda akan diarahkan ke halaman Dashboard Teknisi

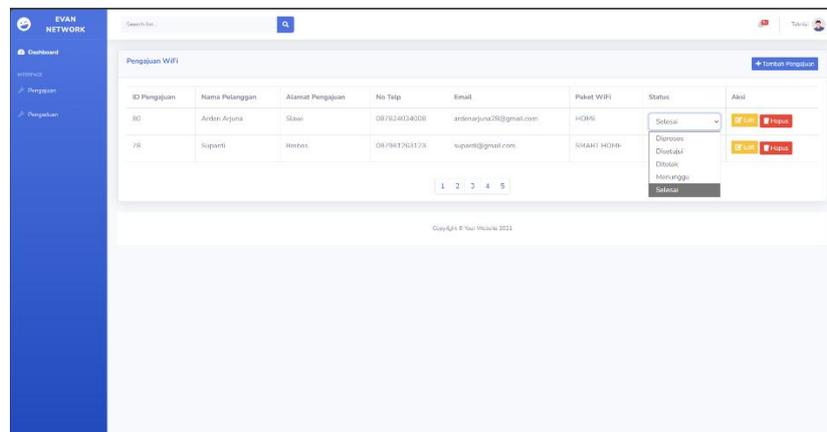
b. Dashboard Teknisi



Setelah berhasil login, Teknisi akan diarahkan ke halaman Dashboard Teknisi. Pada halaman ini, terdapat beberapa bagian:

- 1) Lihat Pengajuan
- 2) Lihat Pengaduan
- 3) Tugas Hari Ini

c. Update Status Pengajuan



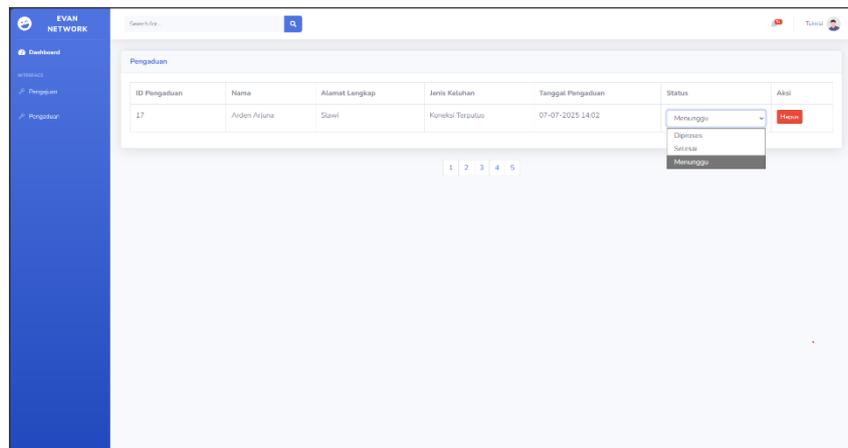
Teknisi dapat mengakses menu Pengajuan untuk melihat daftar permintaan pemasangan layanan dari pelanggan. Setelah proses pemasangan selesai di lapangan, teknisi harus memperbarui status sesuai hasil pekerjaan.

Langkah-langkah:

- 1) Buka menu Pengajuan.

- 2) Pilih data pengajuan yang ingin diperbarui.
- 3) Klik tombol Ubah Status.
- 4) Pilih status seperti Dalam Penanganan, Selesai, Menunggu Ditolak, Disetujui, dan Diproses (sesuai kondisi).
- 5) Klik Simpan untuk menyimpan perubahan status.

d. Update Status Pengaduan



Teknisi juga bertanggung jawab dalam menindaklanjuti pengaduan pelanggan yang diteruskan oleh admin. Setelah penanganan selesai, teknisi harus memperbarui status pengaduan agar user dan admin mengetahui progresnya.

Langkah-langkah:

- 1) Buka menu Pengaduan.
- 2) Pilih pengaduan yang ditugaskan.
- 3) Klik tombol Update Status.
- 4) Pilih status sesuai penanganan (Selesai, Menunggu, dan Diproses).
- 5) Klik Simpan untuk mengupdate data.

Berikut adalah beberapa masalah umum yang mungkin terjadi saat menggunakan aplikasi, beserta solusinya:

1. Tidak Bisa Login

- Periksa Koneksi Internet.
Pastikan perangkat Anda tersambung ke jaringan internet yang stabil.
- Cek Email dan Kata Sandi.
Pastikan email dan kata sandi yang Anda masukkan sudah benar, sesuai dengan data yang terdaftar. Perhatikan huruf besar/kecil dan tidak ada spasi berlebih.

2. Aplikasi Tidak Merespon

- Tutup dan Buka Kembali Aplikasi.
Coba tutup aplikasi dari browser sepenuhnya, lalu buka kembali dan login ulang.
- Periksa Versi Aplikasi.
Pastikan Anda mengakses versi terbaru dari aplikasi. Jika perlu, bersihkan cache browser atau gunakan mode incognito/private.



2025

**RANCANG BANGUN APLIKASI PELAYANAN WIFI
BERBASIS WEBSITE DENGAN METODE PAYMENT
GATEWAY DAN CUSTOMER COMPLAIN HANDLING
MODEL (CCHM) PADA CV. EVAN NETWORK**

Sistem Pengajuan, Pembayaran, Pengaduan Wifi Evan Net

x x x x x

Arden Arjuna Mulvi



DAFTAR ISI

DAFTAR ISI	ii
PENDAHULUAN	1
Controller	2
1. Controller Admin Paket Wifi	2
2. Controller Admin Pelanggan	3
3. Controller Admin Tagihan	4
4. Controller Admin Pengajuan	5
5. Controller Admin Pengaduan.....	6
6. Controller Super Admin Laporan Keuangan	8
7. Controller Teknisi Pengajuan.....	9
8. Controller Teknisi Pengaduan.....	9
9. Controller User Pengajuan Paket Wifi.....	10
10. Controller User Tagihan.....	12
11. Controller User Pengaduan.....	13
Model	14
1. Paket Wifi Model.....	15
2. Data Pelanggan Model.....	16
3. Tagihan Model	17
4. Pengajuan Model	18
5. Pengaduan Model	20

PENDAHULUAN

Perkembangan teknologi informasi telah membawa dampak signifikan terhadap berbagai aspek kehidupan, termasuk dalam hal pengelolaan layanan internet berbasis WiFi. Kebutuhan akan sistem yang terintegrasi dan efisien semakin meningkat seiring dengan bertambahnya jumlah pelanggan dan kompleksitas layanan yang ditawarkan oleh penyedia jaringan internet lokal.

CV. Evan Network sebagai salah satu penyedia layanan WiFi berbasis komunitas di daerah Brebes, menghadapi tantangan dalam mengelola data pelanggan, paket layanan, proses pengajuan, pembayaran, hingga penanganan pengaduan secara manual. Proses yang masih dilakukan secara konvensional menimbulkan risiko keterlambatan layanan, kesalahan pencatatan, dan kurangnya transparansi informasi kepada pelanggan.

Berdasarkan permasalahan tersebut, dikembangkanlah sebuah sistem informasi pelayanan WiFi berbasis website dengan memanfaatkan framework Laravel. Sistem ini dirancang untuk mendukung proses bisnis CV. Evan Network secara digital, meliputi manajemen paket WiFi, pengelolaan pelanggan, sistem tagihan dan pembayaran, serta layanan pengajuan dan pengaduan yang terhubung secara real-time. Pengembangan sistem ini diharapkan dapat meningkatkan efisiensi operasional, akurasi data, dan kepuasan pelanggan.

Dokumen ini menyajikan penjabaran teknis mengenai struktur dan implementasi sistem, termasuk penjelasan terkait controller dan model yang digunakan dalam pengembangan aplikasi. Dengan adanya dokumentasi ini, diharapkan dapat menjadi acuan dalam proses pemeliharaan dan pengembangan sistem di masa yang akan datang serta menjadi bagian dari pertanggungjawaban akademik dalam laporan tugas akhir.

Spesifikasi teknis meliputi :

1. Controller
2. Model

Spesifikasi uraian spesifikasi untuk membangun aplikasi:

1. Windows 11
2. Visual Studio Code
3. Web Browser

Controller

1. Controller Admin Paket Wifi



```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\PaketWifi;
6 use Illuminate\Http\Request;
7 use Illuminate\Support\Facades\Storage;
8 use Illuminate\Support\Facades\Validator;
9
10 class PaketWifiController extends Controller
11 {
12     /**
13      * Menampilkan daftar semua paket WiFi untuk Admin
14      */
15     public function index()
16     {
17         $paketWifi = PaketWifi::all();
18         $totalPaket = $paketWifi->count();
19
20         return view('dashboard.datapaket.index', compact('paketWifi', 'totalPaket'));
21     }
22 }
```

Controller ini adalah bagian dari sistem manajemen paket WiFi yang berfungsi untuk menampilkan dashboard admin. Ketika admin mengakses halaman daftar paket WiFi, controller ini akan mengambil semua data paket WiFi yang tersimpan di database, menghitung berapa total paket yang ada, kemudian menampilkan informasi tersebut dalam bentuk halaman web. Pada dasarnya, ini adalah fungsi "read" atau "view" dalam operasi CRUD yang memungkinkan admin melihat semua paket WiFi yang telah dibuat beserta jumlah totalnya di dashboard.

2. Controller Admin Pelanggan



```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\User;
6 use App\Models\PaketWifi;
7 use App\Models>DataPelanggan;
8 use Illuminate\Http\Request;
9
10 class PelangganController extends Controller
11 {
12     /**
13      * Menampilkan daftar pelanggan untuk Admin
14      */
15     public function index(Request $request)
16     {
17         $pelanggan = $this->getPelangganWithFilters($request);
18
19         return view('dashboard.datapelanggan.index', compact('pelanggan'));
20     }
21 }
```

Controller ini mengelola halaman daftar pelanggan untuk admin. Ketika admin mengakses halaman pelanggan, controller akan mengambil data pelanggan dari database dengan kemungkinan menerapkan filter tertentu (seperti pencarian, sorting, atau pagination), kemudian menampilkan daftar pelanggan tersebut di halaman dashboard admin.

Fungsi utamanya adalah menampilkan data pelanggan yang sudah diproses/difilter sesuai dengan parameter yang diterima dari request (misalnya jika admin melakukan pencarian nama pelanggan atau filter berdasarkan kriteria tertentu).

3. Controller Admin Tagihan

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\Tagihan;
6 use App\Models\User;
7 use App\Models\PaketWifi;
8 use App\Models>StatusTagihan;
9 use App\Models\Pengajuan;
10 use Illuminate\Http\Request;
11 use Carbon\Carbon;
12 use Illuminate\Support\Facades\DB;
13 use Illuminate\Support\Facades\Log;
14
15 class TagihanController extends Controller
16 {
17     /**
18      * Menampilkan daftar semua tagihan
19      */
20     public function index()
21     {
22         $tagihan = Tagihan::with(['user', 'paketWifi', 'statusTagihan', 'pengajuan', ])
23             ->orderBy('created_at', 'desc')
24             ->paginate(10); // tambahkan pagination untuk performa yang lebih baik
25
26         // Menghitung jumlah tagihan yang sudah dibayar (status "Dibayar" atau "Lunas")
27         $lunas = Tagihan::where('status_tagihan_id', 3)->count(); // status "Dibayar" atau "Lunas"
28
29         return view('dashboard.datatagihan.index', compact('tagihan'));
30     }
31 }
```

Controller ini mengelola halaman daftar tagihan untuk admin dengan fitur pagination dan statistik pembayaran. Ketika admin mengakses halaman tagihan, controller akan mengambil semua data tagihan dari database yang mencakup informasi user, paket WiFi, status tagihan, dan pengajuan, kemudian mengurutkan data berdasarkan tanggal pembuatan dengan yang terbaru ditampilkan terlebih dahulu. Data tersebut kemudian dibagi menjadi beberapa halaman dengan sistem pagination yang menampilkan 10 data per halaman untuk memudahkan navigasi. Selain itu, controller juga menghitung jumlah tagihan yang sudah lunas dengan status "Dibayar" atau "Lunas" untuk memberikan ringkasan statistik kepada admin. Semua informasi ini kemudian ditampilkan di halaman dashboard admin untuk membantu dalam memantau status pembayaran pelanggan dan mengelola tagihan dengan lebih efisien.

4. Controller Admin Pengajuan

```
1 //php
2
3 namespace App\Http\Controllers;
4
5 use Carbon\Carbon;
6 use App\Models\Pengajuan;
7 use App\Models\User;
8 use App\Models\PaketWiFi;
9 use App\Models>StatusPengajuan;
10 use App\Models\Tagihan;
11 use App\Models\Notification;
12 use Illuminate\Http\Request;
13 use Illuminate\Support\Facades\Auth;
14 use Illuminate\Support\Facades\Validator;
15 use Illuminate\Support\Facades\Log;
16 use Illuminate\Support\Facades\DB;
17
18 class PengajuanController extends Controller
19 {
20     protected $notificationController;
21
22     public function __construct()
23     {
24         $this->notificationController = new NotificationController();
25     }
26
27     /**
28      * Menampilkan daftar pengajuan untuk dashboard admin
29      */
30     public function index()
31     {
32         $pengajuan = Pengajuan::with(['user', 'paketWiFi', 'statusPengajuan'])
33             ->orderBy('created_at', 'desc')
34             ->get();
35
36         $statusPengajuan = StatusPengajuan::all();
37
38         return view('dashboard.pengajuan.index', compact('pengajuan', 'statusPengajuan'));
39     }
40 }
```

Controller ini mengelola halaman daftar pengajuan untuk admin dengan fitur notifikasi otomatis. Dalam constructor, controller menginisialisasi NotificationController yang berfungsi untuk mengirim notifikasi terkait pengajuan. Ketika admin mengakses halaman pengajuan, controller akan mengambil semua data pengajuan dari database yang mencakup informasi user, paket WiFi, dan status pengajuan, kemudian mengurutkan data berdasarkan tanggal pembuatan dengan yang terbaru ditampilkan terlebih dahulu dan mengubahnya menjadi collection untuk memudahkan manipulasi data. Controller juga mengambil semua data status pengajuan untuk keperluan filter atau dropdown. Semua informasi ini kemudian ditampilkan di halaman dashboard pengajuan admin, memungkinkan admin untuk memantau dan mengelola pengajuan pelanggan dengan lebih efektif

sambil mendapatkan notifikasi real-time untuk setiap perubahan status pengajuan.

5. Controller Admin Pengaduan

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\Pengaduan;
6 use App\Models\User;
7 use App\Models>StatusPengaduan;
8 use App\Models\Keluhan;
9 use App\Models\Notification;
10 use Illuminate\Http\Request;
11 use Illuminate\Support\Facades\Auth;
12 use Illuminate\Support\Facades\Log;
13
14 class PengaduanController extends Controller
15 {
16     protected $notificationController;
17
18     // Constants for better maintainability
19     const DEFAULT_STATUS_ID = 3;
20     const COMPLETED_STATUS_ID = 2;
21     const TEKNIISI_ROLE_ID = 4;
22     const ADMIN_ROLE = 1;
23     const SUPERADMIN_ROLE = 3;
24
25     public function __construct()
26     {
27         $this->notificationController = new NotificationController();
28     }
29
30     /**
31      * Display a listing of the resource based on user role
32      */
33     public function index()
34     {
35         $pengaduan = $this->getPengaduanWithRelations();
36         $statusPengaduan = StatusPengaduan::all();
37
38         $view = match (auth()->user()->role) {
39             self::SUPERADMIN_ROLE => 'superadmin.pengaduan.index',
40             'teknisi' => 'teknisi.pengaduan.index',
41             default => 'dashboard.pengaduan.index'
42         };
43
44         return view($view, compact('pengaduan', 'statusPengaduan'));
45     }
46 }
```

Controller ini mengelola halaman daftar pengaduan dengan sistem akses berbeda berdasarkan role pengguna. Controller memiliki konstanta untuk ID status dan role, serta NotificationController untuk notifikasi pengaduan.

Ketika diakses, controller mengambil data pengaduan beserta relasinya dan semua status pengaduan. Fitur utamanya adalah routing dinamis berdasarkan role dimana superadmin diarahkan ke halaman superadmin, teknisi ke halaman teknisi, dan role lainnya ke dashboard biasa. Data pengaduan dan status kemudian dikirim ke view yang sesuai dengan role pengguna untuk memastikan akses yang tepat sesuai wewenang masing-masing.x

6. Controller Super Admin Laporan Keuangan

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\PaketWifi;
7 use App\Models>StatusTagihan;
8 use Illuminate\Http\Request;
9 use Illuminate\Support\Facades\DB;
10 use Carbon\Carbon;
11 use PhpOffice\PhpSpreadsheet\Spreadsheet;
12 use PhpOffice\PhpSpreadsheet\Writer\Xlsx;
13 use Barryvdh\DomPDF\Facade\Pdf;
14
15 class LaporanKeuanganController extends Controller
16 {
17     // Status tagihan constants berdasarkan database
18     const STATUS_LUNAS = 1;
19     const STATUS_BELUM_BAYAR = 2;
20     const STATUS_SEBAGIAN = 3;
21     const STATUS_DIBATALKAN = 4;
22     const STATUS_TERLAMBAT = 5;
23
24     /**
25      * Display the financial report page
26      */
27     public function index()
28     {
29         $data = [
30             'title' => 'Laporan Keuangan',
31             'paketList' => $this->getPaketList(),
32             'statusList' => $this->getStatusList(),
33             'summaryCards' => $this->getSummaryCards(),
34             'chartData' => $this->getChartData(),
35             'agingReport' => $this->getAgingReport(),
36         ];
37
38         return view('superadmin.laporankeuangan.index', $data);
39     }
40 }
```

Controller ini mengelola halaman laporan keuangan untuk superadmin dengan fitur export ke Excel dan PDF. Controller memiliki konstanta untuk status tagihan seperti lunas, belum bayar, sebagian, dibatalkan, dan terlambat. Ketika diakses, controller mengumpulkan berbagai data keuangan termasuk daftar paket WiFi, status tagihan, kartu ringkasan statistik, data chart untuk visualisasi, dan laporan aging untuk analisis umur

piutang. Semua data tersebut dikemas dalam array dan dikirim ke view superadmin laporan keuangan, memberikan dashboard keuangan yang komprehensif dengan berbagai metrik dan visualisasi untuk membantu superadmin dalam pengambilan keputusan bisnis.

7. Controller Teknisi Pengajuan

```
1 public function indexTeknisi()
2 {
3     $pengajuan = Pengajuan::whereHas('user', function($query) {
4         $query->where('role_id', 4); // 4 adalah id untuk role 'teknisi'
5     }->get();
6
7     return view('teknisi.pengajuan.index', compact('pengajuan'));
8 }
```

Fungsi ini menampilkan halaman daftar pengajuan khusus untuk teknisi. Controller mengambil data pengajuan beserta informasi user yang terkait, namun dengan filter khusus yang hanya menampilkan pengajuan dari user yang memiliki role teknisi (role_id = 4). Data pengajuan yang sudah difilter kemudian dikirim ke view teknisi pengajuan untuk ditampilkan di dashboard teknisi.

8. Controller Teknisi Pengaduan

```
1 public function indexTeknisi()
2 {
3     $pengaduan = Pengaduan::whereHas('user', function($query) {
4         $query->where('role_id', self::TEKNISI_ROLE_ID);
5     }->with(['user', 'keluhan', 'statusPengaduan']->get();
6
7     return view('teknisi.pengaduan.index', compact('pengaduan'));
8 }
```

Fungsi ini menampilkan halaman daftar pengaduan khusus untuk teknisi. Controller mengambil data pengaduan beserta relasi user, keluhan, dan status pengaduan, namun dengan filter khusus yang hanya menampilkan

pengaduan dari user yang memiliki role teknisi (menggunakan konstanta `TEKNISI_ROLE_ID`). Data pengaduan yang sudah difilter kemudian dikirim ke view teknisi pengaduan untuk ditampilkan di dashboard teknisi.

9. Controller User Pengajuan Paket Wifi



```
1 public function showPackages()
2 {
3     $paketWifi = PaketWifi::all();
4     return view('user.paketwifi', compact('paketWifi'));
5 }
```

Fungsi ini menampilkan halaman daftar paket WiFi untuk user. Controller mengambil semua data paket WiFi yang tersedia dan mengirimkannya ke view user paket WiFi, memungkinkan user untuk melihat dan memilih paket WiFi yang ingin mereka ajukan.

```

1 public function store(Request $request)
2 {
3     $validated = $request->validate([
4         'paket_wifi_id.paket_wifi' => 'required|exists:paket_wifi,id.paket_wifi'
5     ]);
6
7     try {
8         // Cek apakah user sudah memiliki pengajuan yang sedang diproses
9         $existingPengajuan = Pengajuan::where('user_id.user' => Auth::id())
10             ->where('status_pengajuan_id', [4, 5]) ->first(); // Status Menunggu atau Selesai
11         if ($existingPengajuan) {
12             return redirect()->back()->with('error', 'Anda sudah memiliki pengajuan yang sedang diproses atau selesai.');

```

Fungsi ini mengelola proses penyimpanan pengajuan paket WiFi baru oleh user dengan validasi dan pengecekan duplikasi. Controller pertama memvalidasi input paket WiFi yang dipilih, kemudian mengecek apakah user sudah memiliki pengajuan yang sedang diproses atau selesai untuk menghindari duplikasi. Jika belum ada pengajuan sebelumnya, sistem akan membuat pengajuan baru dengan status "Menunggu" dan mengirim notifikasi kepada admin. Setelah berhasil disimpan, user akan diarahkan kembali ke halaman paket WiFi dengan pesan sukses. Jika terjadi error dalam proses penyimpanan, sistem akan mencatat error tersebut dan menampilkan pesan error kepada user untuk mencoba lagi.

10. Controller User Tagihan

```
1 public function userTagihan()
2 {
3     try {
4         $user = auth()-user();
5
6         if (!$user) {
7             return redirect()-route('login')->with('error', 'Silakan login terlebih dahulu. ');
8         }
9
10        $tagihan = Tagihan::select('paketid', 'statustagihan', 'tanggal')
11            -where('users_id_user', $user->id_user)
12            -orderBy('tanggal asc', 'desc')
13            -paginate(10);
14
15        return view('user-tagihan', compact('tagihan'));
16    } catch (Exception $e) {
17        $log = error('Error menampilkan tagihan user: ' . $e->getMessage());
18        return redirect()-back()-with('error', 'Terjadi kesalahan saat memuat tagihan. ');
19    }
20 }
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
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
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Controller ini untuk mengelola tagihan user. Fungsi `userTagihan()` menampilkan halaman daftar tagihan milik user yang sedang login dengan pagination 10 data per halaman, dilengkapi pengecekan autentikasi dan error handling. Fungsi `userTagihanShow()` menampilkan detail tagihan tertentu berdasarkan ID dengan validasi kepemilikan tagihan dan

pengecekan keberadaan data. Fungsi `userTagihanPayment()` menampilkan halaman pembayaran untuk tagihan yang statusnya belum dibayar atau terlambat, dengan validasi bahwa tagihan masih dapat dibayar. Semua fungsi dilengkapi dengan error handling yang akan mengarahkan user kembali dengan pesan error jika terjadi masalah dalam proses pengambilan data.

11. Controller User Pengaduan



```
1 public function create()
2 {
3     $keluhan = Keluhan::all();
4
5     $view = match (auth()->user()->role) {
6         self::ADMIN_ROLE => 'dashboard.pengaduan.create',
7         self::SUPERADMIN_ROLE => 'superadmin.pengaduan.create',
8         default => 'user.pengaduamifi'
9     };
10
11     return view($view, compact('keluhan'));
12 }
13
14 /**
15  * Store a newly created resource in storage
16  */
17 public function store(Request $request)
18 {
19     $validated = $this->validatePengaduanData($request);
20
21     try {
22         $pengaduan = $this->createPengaduan($validated);
23         $this->sendNewComplaintNotificationToAdmin($pengaduan);
24
25         return $this->getStoreSuccessResponse();
26     } catch (\Exception $e) {
27         return $this->handleStoreError($e, $request);
28     }
29 }
30 }
```

Kedua fungsi ini membentuk sistem pengaduan yang lengkap dengan routing dinamis berdasarkan role user. Fungsi `create` menampilkan halaman form pembuatan pengaduan dengan mengambil data keluhan dari database, kemudian menentukan view yang sesuai berdasarkan role pengguna dimana admin diarahkan ke dashboard pengaduan, superadmin ke form superadmin,

Model

1. Paket Wifi Model

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class PaketWifi extends Model
9 {
10     use HasFactory;
11
12     // Nama tabel yang digunakan oleh model
13     protected $table = 'paket_wifi';
14
15     // Nama kolom primary key adalah 'id', tentukan nama primary key
16     protected $primaryKey = 'id_paket_wifi'; // default adalah nama primary key yang benar
17
18     // Nama kolom yang dapat diisi secara mass assignment
19     protected $fillable = [
20         'nama_paket',
21         'kecepatan',
22         'harga_bulanan',
23         'deskripsi_paket',
24         'gambar',
25     ];
26
27     /**
28      * Relasi dengan model Pengajuan
29      * Setiap paket wifi dapat memiliki banyak pengajuan
30      */
31     public function pengajuan()
32     {
33         // Relasi dengan model yang akan di-relasikan
34         return $this->hasMany(Pengajuan::class, 'paket_wifi_id_paket_wifi', 'id_paket_wifi');
35     }
36
37     /**
38      * Relasi dengan model User
39      * Satu paket wifi bisa dimiliki oleh banyak pengguna (relasi belongsTo)
40      */
41     public function user()
42     {
43         // Relasi dengan model yang akan di-relasikan dengan User
44         return $this->belongsToMany(User::class, 'pengajuan', 'paket_wifi_id_paket_wifi', 'users_id_user');
45     }
46
47     // Relasi untuk User
48     public function users()
49     {
50         return $this->hasMany(User::class, 'paket_wifi_id_paket_wifi'); // Relasi paket_wifi ke tabel users
51     }
52 }
```

Model PaketWifi ini merupakan representasi tabel paket_wifi dalam database yang menggunakan Laravel Eloquent dengan fitur factory untuk data testing. Model ini memiliki konfigurasi dasar berupa nama tabel 'paket_wifi' dan primary key 'id_paket_wifi', serta mendefinisikan kolom-kolom yang dapat diisi secara mass assignment yaitu nama_paket, kecepatan, harga_bulanan, deskripsi_paket, dan gambar. Model ini dilengkapi dengan tiga relasi penting: fungsi pengajuan() yang menghubungkan dengan model Pengajuan melalui foreign key dan local key untuk menampilkan banyak pengajuan terkait paket tertentu, fungsi user() yang membuat relasi belongsToMany dengan model User

melalui tabel pivot 'pengajuan' untuk menunjukkan hubungan many-to-many antara paket dan user, dan fungsi users() yang menggunakan hasMany untuk menampilkan semua user yang terkait dengan paket WiFi tertentu melalui relasi di tabel users.

2. Data Pelanggan Model

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class DataPelanggan extends Model
9 {
10     use HasFactory;
11
12     // Nama tabel yang digunakan oleh model
13     protected $table = 'datapelanggan';
14
15     // Tentukan kolom-kolom apa yang akan diisi (mass assignment)
16     protected $fillable = [
17         'nama',
18         'email',
19         'alamat',
20         'no_telepon',
21     ];
22
23     // Jika menggunakan timestamps
24     public $timestamps = true;
25
26     /**
27      * Mengurutkan data berdasarkan 'created_at' secara descending (terbaru di atas).
28      */
29     protected static function booted()
30     {
31         parent::boot();
32
33         static::addGlobalScope('orderLatest', function ($builder) {
34             $builder->orderBy('created_at', 'desc'); // Atau bisa menggunakan 'id' untuk urutan berdasarkan ID
35         });
36     }
37 }
38
```

Model DataPelanggan ini merupakan representasi tabel datapelanggan dalam database yang menggunakan Laravel Eloquent dengan fitur factory untuk data testing. Model ini memiliki konfigurasi dasar berupa nama tabel 'datapelanggan' dan mendefinisikan kolom-kolom yang dapat diisi secara mass assignment yaitu nama, email, alamat, dan no_telepon untuk menyimpan informasi lengkap pelanggan. Model ini dilengkapi dengan fitur timestamps otomatis yang akan mencatat waktu pembuatan dan pembaruan data, serta memiliki fungsi booted() yang menerapkan global scope untuk mengurutkan data berdasarkan created_at secara descending, sehingga data pelanggan yang baru ditambahkan akan selalu tampil di posisi teratas ketika melakukan query ke database.

3. Tagihan Model

```
class Tagihan {
    private int id;
    private String user;
    private double jumlah;
    private Date tanggal;
    private boolean isPaid;

    public Tagihan(int id, String user, double jumlah, Date tanggal, boolean isPaid) {
        this.id = id;
        this.user = user;
        this.jumlah = jumlah;
        this.tanggal = tanggal;
        this.isPaid = isPaid;
    }

    // Getter dan Setter
    public int getId() { return id; }
    public String getUser() { return user; }
    public double getJumlah() { return jumlah; }
    public Date getTanggal() { return tanggal; }
    public boolean getIsPaid() { return isPaid; }

    public void setId(int id) { this.id = id; }
    public void setUser(String user) { this.user = user; }
    public void setJumlah(double jumlah) { this.jumlah = jumlah; }
    public void setTanggal(Date tanggal) { this.tanggal = tanggal; }
    public void setIsPaid(boolean isPaid) { this.isPaid = isPaid; }

    // Method utilitas
    public boolean isExpired() {
        Date now = new Date();
        return now.getTime() > tanggal.getTime();
    }

    public boolean isPaid() {
        return isPaid;
    }

    // Method untuk memeriksa status pembayaran
    public void checkPaymentStatus() {
        // Logika untuk memeriksa status pembayaran
    }
}
```

Model Tagihan ini mengelola sistem pembayaran tagihan WiFi dengan integrasi Midtrans, memiliki relasi lengkap ke User, PaketWifi, StatusTagihan, dan Payment. Model dilengkapi dengan method utilitas untuk mengecek status pembayaran (isPaid), jatuh tempo

(isJatuhTempo), generator nomor invoice otomatis, generator order ID Midtrans, serta scope queries untuk memfilter tagihan berdasarkan status lunas/belum lunas dan yang sudah jatuh tempo.

4. Pengajuan Model



```
1 <?php
2 namespace App\Models;
3 use Illuminate\Database\Eloquent\Factories\HasFactory;
4 use Illuminate\Database\Eloquent\Model;
5 class Pengajuan extends Model
6 {
7     use HasFactory;
8     protected $table = 'pengajuan';
9     protected $primaryKey = 'id'; //primary key
10    public $timestamps = false; //tidak ada createdAt dan updatedAt
11    protected $fillable = [
12        'user_id',
13        'paket_wifi_id',
14        'status_pengajuan'
15    ];
16    // Relasi dengan user
17    public function user()
18    {
19        return $this->belongsTo(User::class, 'user_id', 'id');
20    }
21    // Relasi dengan paketWifi
22    public function paketWifi()
23    {
24        return $this->belongsTo(PaketWifi::class, 'paket_wifi_id', 'id');
25    }
26    // Relasi dengan StatusPengajuan
27    public function statusPengajuan()
28    {
29        return $this->belongsTo(StatusPengajuan::class, 'status_pengajuan_id', 'id');
30    }
31    // Relasi dengan tagihan
32    public function tagihan()
33    {
34        return $this->hasMany(Tagihan::class, 'tagihan_id', 'id');
35    }
36    // Relasi ke tagihan dengan status tertentu (status_pengajuan_id = 1)
37    public function tagihanLunas()
38    {
39        return $this->tagihan()->where('status_pengajuan_id', 1);
40    }
41    // Relasi untuk mendapatkan user yang mengajukan atau user yang akan di bayar
42    public function getKasudatUser()
43    {
44        return $this->user ? $this->user->name : 'N/A';
45    }
46    // Relasi untuk mendapatkan paket wifi yang diajukan
47    public function getPaketWifiUser()
48    {
49        return $this->paket_wifi ? $this->paket_wifi->nama : 'N/A';
50    }
51    // Relasi untuk mendapatkan status pengajuan
52    public function getStatusPengajuan()
53    {
54        return $this->status_pengajuan ? $this->status_pengajuan->status : 'Belum';
55    }
56 }
```

Model Pengajuan ini mengelola sistem pengajuan layanan WiFi dengan relasi lengkap ke User, PaketWifi, StatusPengajuan, dan Tagihan. Model dilengkapi dengan relasi belongsTo untuk menghubungkan dengan entitas terkait, hasMany untuk tagihan yang terkait dengan pengajuan,

serta method khusus `tagihanSelesai()` untuk memfilter tagihan dengan status selesai. Model juga memiliki accessor methods untuk mendapatkan nama, alamat, dan nomor HP dari user terkait dengan fallback 'N/A' jika data tidak tersedia, serta fungsi `getStatusPengajuanName()` untuk menampilkan nama status pengajuan dengan default 'Unknown'.

5. Pengaduan Model

```
1 <!--
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Carbon\Carbon;
8
9 class Pengaduan extends Model
10 {
11     use HasFactory;
12
13     /**
14      * The database table for this model (e.g. "users").
15      *
16      * @var string
17      */
18     protected $table = 'pengaduan';
19
20     /**
21      * The primary key for the model (e.g. "id").
22      *
23      * @var string
24      */
25     protected $primaryKey = 'id';
26
27     /**
28      * The attributes that are mass assignable.
29      *
30      * @var array<string, bool>
31      */
32     protected $fillable = [
33         'user_id_user',
34         'keluhan_id_keluhan',
35         'status_id_status',
36     ];
37
38     /**
39      * The attributes that should be hidden for serialization.
40      *
41      * @var array<string>
42      */
43     protected $hidden = [
44         'created_at', 'updated_at', 'deleted_at',
45     ];
46
47     /**
48      * The attributes that should be cast to native types.
49      *
50      * @var array<string, string>
51      */
52     protected $casts = [
53         'created_at' => 'datetime-tz',
54         'updated_at' => 'datetime-tz',
55     ];
56
57     /**
58      * The relationships that are automatically eager loaded.
59      *
60      * @var array<string, string>
61      */
62     protected $with = [
63         'user',
64         'keluhan',
65         'statusPengaduan',
66     ];
67
68     /**
69      * The relationships that are automatically eager loaded.
70      *
71      * @var array<string, string>
72      */
73     protected $withDefault = [
74         'user',
75         'keluhan',
76         'statusPengaduan',
77     ];
78
79     /**
80      * The relationships that are automatically eager loaded.
81      *
82      * @var array<string, string>
83      */
84     protected $withDefault = [
85         'user',
86         'keluhan',
87         'statusPengaduan',
88     ];
89
90     /**
91      * The relationships that are automatically eager loaded.
92      *
93      * @var array<string, string>
94      */
95     protected $withDefault = [
96         'user',
97         'keluhan',
98         'statusPengaduan',
99     ];
100
101     /**
102      * The relationships that are automatically eager loaded.
103      *
104      * @var array<string, string>
105      */
106     protected $withDefault = [
107         'user',
108         'keluhan',
109         'statusPengaduan',
110     ];
111
112     /**
113      * The relationships that are automatically eager loaded.
114      *
115      * @var array<string, string>
116      */
117     protected $withDefault = [
118         'user',
119         'keluhan',
120         'statusPengaduan',
121     ];
122
123     /**
124      * The relationships that are automatically eager loaded.
125      *
126      * @var array<string, string>
127      */
128     protected $withDefault = [
129         'user',
130         'keluhan',
131         'statusPengaduan',
132     ];
133
134     /**
135      * The relationships that are automatically eager loaded.
136      *
137      * @var array<string, string>
138      */
139     protected $withDefault = [
140         'user',
141         'keluhan',
142         'statusPengaduan',
143     ];
144
145     /**
146      * The relationships that are automatically eager loaded.
147      *
148      * @var array<string, string>
149      */
150     protected $withDefault = [
151         'user',
152         'keluhan',
153         'statusPengaduan',
154     ];
155
156     /**
157      * The relationships that are automatically eager loaded.
158      *
159      * @var array<string, string>
160      */
161     protected $withDefault = [
162         'user',
163         'keluhan',
164         'statusPengaduan',
165     ];
166
167     /**
168      * The relationships that are automatically eager loaded.
169      *
170      * @var array<string, string>
171      */
172     protected $withDefault = [
173         'user',
174         'keluhan',
175         'statusPengaduan',
176     ];
177
178     /**
179      * The relationships that are automatically eager loaded.
180      *
181      * @var array<string, string>
182      */
183     protected $withDefault = [
184         'user',
185         'keluhan',
186         'statusPengaduan',
187     ];
188
189     /**
190      * The relationships that are automatically eager loaded.
191      *
192      * @var array<string, string>
193      */
194     protected $withDefault = [
195         'user',
196         'keluhan',
197         'statusPengaduan',
198     ];
199
200     /**
201      * The relationships that are automatically eager loaded.
202      *
203      * @var array<string, string>
204      */
205     protected $withDefault = [
206         'user',
207         'keluhan',
208         'statusPengaduan',
209     ];
210
211     /**
212      * The relationships that are automatically eager loaded.
213      *
214      * @var array<string, string>
215      */
216     protected $withDefault = [
217         'user',
218         'keluhan',
219         'statusPengaduan',
220     ];
221
222     /**
223      * The relationships that are automatically eager loaded.
224      *
225      * @var array<string, string>
226      */
227     protected $withDefault = [
228         'user',
229         'keluhan',
230         'statusPengaduan',
231     ];
232
233     /**
234      * The relationships that are automatically eager loaded.
235      *
236      * @var array<string, string>
237      */
238     protected $withDefault = [
239         'user',
240         'keluhan',
241         'statusPengaduan',
242     ];
243
244     /**
245      * The relationships that are automatically eager loaded.
246      *
247      * @var array<string, string>
248      */
249     protected $withDefault = [
250         'user',
251         'keluhan',
252         'statusPengaduan',
253     ];
254
255     /**
256      * The relationships that are automatically eager loaded.
257      *
258      * @var array<string, string>
259      */
260     protected $withDefault = [
261         'user',
262         'keluhan',
263         'statusPengaduan',
264     ];
265
266     /**
267      * The relationships that are automatically eager loaded.
268      *
269      * @var array<string, string>
270      */
271     protected $withDefault = [
272         'user',
273         'keluhan',
274         'statusPengaduan',
275     ];
276
277     /**
278      * The relationships that are automatically eager loaded.
279      *
280      * @var array<string, string>
281      */
282     protected $withDefault = [
283         'user',
284         'keluhan',
285         'statusPengaduan',
286     ];
287
288     /**
289      * The relationships that are automatically eager loaded.
290      *
291      * @var array<string, string>
292      */
293     protected $withDefault = [
294         'user',
295         'keluhan',
296         'statusPengaduan',
297     ];
298
299     /**
300      * The relationships that are automatically eager loaded.
301      *
302      * @var array<string, string>
303      */
304     protected $withDefault = [
305         'user',
306         'keluhan',
307         'statusPengaduan',
308     ];
309
310     /**
311      * The relationships that are automatically eager loaded.
312      *
313      * @var array<string, string>
314      */
315     protected $withDefault = [
316         'user',
317         'keluhan',
318         'statusPengaduan',
319     ];
320
321     /**
322      * The relationships that are automatically eager loaded.
323      *
324      * @var array<string, string>
325      */
326     protected $withDefault = [
327         'user',
328         'keluhan',
329         'statusPengaduan',
330     ];
331
332     /**
333      * The relationships that are automatically eager loaded.
334      *
335      * @var array<string, string>
336      */
337     protected $withDefault = [
338         'user',
339         'keluhan',
340         'statusPengaduan',
341     ];
342
343     /**
344      * The relationships that are automatically eager loaded.
345      *
346      * @var array<string, string>
347      */
348     protected $withDefault = [
349         'user',
350         'keluhan',
351         'statusPengaduan',
352     ];
353
354     /**
355      * The relationships that are automatically eager loaded.
356      *
357      * @var array<string, string>
358      */
359     protected $withDefault = [
360         'user',
361         'keluhan',
362         'statusPengaduan',
363     ];
364
365     /**
366      * The relationships that are automatically eager loaded.
367      *
368      * @var array<string, string>
369      */
370     protected $withDefault = [
371         'user',
372         'keluhan',
373         'statusPengaduan',
374     ];
375
376     /**
377      * The relationships that are automatically eager loaded.
378      *
379      * @var array<string, string>
380      */
381     protected $withDefault = [
382         'user',
383         'keluhan',
384         'statusPengaduan',
385     ];
386
387     /**
388      * The relationships that are automatically eager loaded.
389      *
390      * @var array<string, string>
391      */
392     protected $withDefault = [
393         'user',
394         'keluhan',
395         'statusPengaduan',
396     ];
397
398     /**
399      * The relationships that are automatically eager loaded.
400      *
401      * @var array<string, string>
402      */
403     protected $withDefault = [
404         'user',
405         'keluhan',
406         'statusPengaduan',
407     ];
408
409     /**
410      * The relationships that are automatically eager loaded.
411      *
412      * @var array<string, string>
413      */
414     protected $withDefault = [
415         'user',
416         'keluhan',
417         'statusPengaduan',
418     ];
419
420     /**
421      * The relationships that are automatically eager loaded.
422      *
423      * @var array<string, string>
424      */
425     protected $withDefault = [
426         'user',
427         'keluhan',
428         'statusPengaduan',
429     ];
430
431     /**
432      * The relationships that are automatically eager loaded.
433      *
434      * @var array<string, string>
435      */
436     protected $withDefault = [
437         'user',
438         'keluhan',
439         'statusPengaduan',
440     ];
441
442     /**
443      * The relationships that are automatically eager loaded.
444      *
445      * @var array<string, string>
446      */
447     protected $withDefault = [
448         'user',
449         'keluhan',
450         'statusPengaduan',
451     ];
452
453     /**
454      * The relationships that are automatically eager loaded.
455      *
456      * @var array<string, string>
457      */
458     protected $withDefault = [
459         'user',
460         'keluhan',
461         'statusPengaduan',
462     ];
463
464     /**
465      * The relationships that are automatically eager loaded.
466      *
467      * @var array<string, string>
468      */
469     protected $withDefault = [
470         'user',
471         'keluhan',
472         'statusPengaduan',
473     ];
474
475     /**
476      * The relationships that are automatically eager loaded.
477      *
478      * @var array<string, string>
479      */
480     protected $withDefault = [
481         'user',
482         'keluhan',
483         'statusPengaduan',
484     ];
485
486     /**
487      * The relationships that are automatically eager loaded.
488      *
489      * @var array<string, string>
490      */
491     protected $withDefault = [
492         'user',
493         'keluhan',
494         'statusPengaduan',
495     ];
496
497     /**
498      * The relationships that are automatically eager loaded.
499      *
500      * @var array<string, string>
501      */
502     protected $withDefault = [
503         'user',
504         'keluhan',
505         'statusPengaduan',
506     ];
507
508     /**
509      * The relationships that are automatically eager loaded.
510      *
511      * @var array<string, string>
512      */
513     protected $withDefault = [
514         'user',
515         'keluhan',
516         'statusPengaduan',
517     ];
518
519     /**
520      * The relationships that are automatically eager loaded.
521      *
522      * @var array<string, string>
523      */
524     protected $withDefault = [
525         'user',
526         'keluhan',
527         'statusPengaduan',
528     ];
529
530     /**
531      * The relationships that are automatically eager loaded.
532      *
533      * @var array<string, string>
534      */
535     protected $withDefault = [
536         'user',
537         'keluhan',
538         'statusPengaduan',
539     ];
540
541     /**
542      * The relationships that are automatically eager loaded.
543      *
544      * @var array<string, string>
545      */
546     protected $withDefault = [
547         'user',
548         'keluhan',
549         'statusPengaduan',
550     ];
551
552     /**
553      * The relationships that are automatically eager loaded.
554      *
555      * @var array<string, string>
556      */
557     protected $withDefault = [
558         'user',
559         'keluhan',
560         'statusPengaduan',
561     ];
562
563     /**
564      * The relationships that are automatically eager loaded.
565      *
566      * @var array<string, string>
567      */
568     protected $withDefault = [
569         'user',
570         'keluhan',
571         'statusPengaduan',
572     ];
573
574     /**
575      * The relationships that are automatically eager loaded.
576      *
577      * @var array<string, string>
578      */
579     protected $withDefault = [
580         'user',
581         'keluhan',
582         'statusPengaduan',
583     ];
584
585     /**
586      * The relationships that are automatically eager loaded.
587      *
588      * @var array<string, string>
589      */
590     protected $withDefault = [
591         'user',
592         'keluhan',
593         'statusPengaduan',
594     ];
595
596     /**
597      * The relationships that are automatically eager loaded.
598      *
599      * @var array<string, string>
600      */
601     protected $withDefault = [
602         'user',
603         'keluhan',
604         'statusPengaduan',
605     ];
606
607     /**
608      * The relationships that are automatically eager loaded.
609      *
610      * @var array<string, string>
611      */
612     protected $withDefault = [
613         'user',
614         'keluhan',
615         'statusPengaduan',
616     ];
617
618     /**
619      * The relationships that are automatically eager loaded.
620      *
621      * @var array<string, string>
622      */
623     protected $withDefault = [
624         'user',
625         'keluhan',
626         'statusPengaduan',
627     ];
628
629     /**
630      * The relationships that are automatically eager loaded.
631      *
632      * @var array<string, string>
633      */
634     protected $withDefault = [
635         'user',
636         'keluhan',
637         'statusPengaduan',
638     ];
639
640     /**
641      * The relationships that are automatically eager loaded.
642      *
643      * @var array<string, string>
644      */
645     protected $withDefault = [
646         'user',
647         'keluhan',
648         'statusPengaduan',
649     ];
650
651     /**
652      * The relationships that are automatically eager loaded.
653      *
654      * @var array<string, string>
655      */
656     protected $withDefault = [
657         'user',
658         'keluhan',
659         'statusPengaduan',
660     ];
661
662     /**
663      * The relationships that are automatically eager loaded.
664      *
665      * @var array<string, string>
666      */
667     protected $withDefault = [
668         'user',
669         'keluhan',
670         'statusPengaduan',
671     ];
672
673     /**
674      * The relationships that are automatically eager loaded.
675      *
676      * @var array<string, string>
677      */
678     protected $withDefault = [
679         'user',
680         'keluhan',
681         'statusPengaduan',
682     ];
683
684     /**
685      * The relationships that are automatically eager loaded.
686      *
687      * @var array<string, string>
688      */
689     protected $withDefault = [
690         'user',
691         'keluhan',
692         'statusPengaduan',
693     ];
694
695     /**
696      * The relationships that are automatically eager loaded.
697      *
698      * @var array<string, string>
699      */
700     protected $withDefault = [
701         'user',
702         'keluhan',
703         'statusPengaduan',
704     ];
705
706     /**
707      * The relationships that are automatically eager loaded.
708      *
709      * @var array<string, string>
710      */
711     protected $withDefault = [
712         'user',
713         'keluhan',
714         'statusPengaduan',
715     ];
716
717     /**
718      * The relationships that are automatically eager loaded.
719      *
720      * @var array<string, string>
721      */
722     protected $withDefault = [
723         'user',
724         'keluhan',
725         'statusPengaduan',
726     ];
727
728     /**
729      * The relationships that are automatically eager loaded.
730      *
731      * @var array<string, string>
732      */
733     protected $withDefault = [
734         'user',
735         'keluhan',
736         'statusPengaduan',
737     ];
738
739     /**
740      * The relationships that are automatically eager loaded.
741      *
742      * @var array<string, string>
743      */
744     protected $withDefault = [
745         'user',
746         'keluhan',
747         'statusPengaduan',
748     ];
749
750     /**
751      * The relationships that are automatically eager loaded.
752      *
753      * @var array<string, string>
754      */
755     protected $withDefault = [
756         'user',
757         'keluhan',
758         'statusPengaduan',
759     ];
760
761     /**
762      * The relationships that are automatically eager loaded.
763      *
764      * @var array<string, string>
765      */
766     protected $withDefault = [
767         'user',
768         'keluhan',
769         'statusPengaduan',
770     ];
771
772     /**
773      * The relationships that are automatically eager loaded.
774      *
775      * @var array<string, string>
776      */
777     protected $withDefault = [
778         'user',
779         'keluhan',
780         'statusPengaduan',
781     ];
782
783     /**
784      * The relationships that are automatically eager loaded.
785      *
786      * @var array<string, string>
787      */
788     protected $withDefault = [
789         'user',
790         'keluhan',
791         'statusPengaduan',
792     ];
793
794     /**
795      * The relationships that are automatically eager loaded.
796      *
797      * @var array<string, string>
798      */
799     protected $withDefault = [
800         'user',
801         'keluhan',
802         'statusPengaduan',
803     ];
804
805     /**
806      * The relationships that are automatically eager loaded.
807      *
808      * @var array<string, string>
809      */
810     protected $withDefault = [
811         'user',
812         'keluhan',
813         'statusPengaduan',
814     ];
815
816     /**
817      * The relationships that are automatically eager loaded.
818      *
819      * @var array<string, string>
820      */
821     protected $withDefault = [
822         'user',
823         'keluhan',
824         'statusPengaduan',
825     ];
826
827     /**
828      * The relationships that are automatically eager loaded.
829      *
830      * @var array<string, string>
831      */
832     protected $withDefault = [
833         'user',
834         'keluhan',
835         'statusPengaduan',
836     ];
837
838     /**
839      * The relationships that are automatically eager loaded.
840      *
841      * @var array<string, string>
842      */
843     protected $withDefault = [
844         'user',
845         'keluhan',
846         'statusPengaduan',
847     ];
848
849     /**
850      * The relationships that are automatically eager loaded.
851      *
852      * @var array<string, string>
853      */
854     protected $withDefault = [
855         'user',
856         'keluhan',
857         'statusPengaduan',
858     ];
859
860     /**
861      * The relationships that are automatically eager loaded.
862      *
863      * @var array<string, string>
864      */
865     protected $withDefault = [
866         'user',
867         'keluhan',
868         'statusPengaduan',
869     ];
870
871     /**
872      * The relationships that are automatically eager loaded.
873      *
874      * @var array<string, string>
875      */
876     protected $withDefault = [
877         'user',
878         'keluhan',
879         'statusPengaduan',
880     ];
881
882     /**
883      * The relationships that are automatically eager loaded.
884      *
885      * @var array<string, string>
886      */
887     protected $withDefault = [
888         'user',
889         'keluhan',
890         'statusPengaduan',
891     ];
892
893     /**
894      * The relationships that are automatically eager loaded.
895      *
896      * @var array<string, string>
897      */
898     protected $withDefault = [
899         'user',
900         'keluhan',
901         'statusPengaduan',
902     ];
903
904     /**
905      * The relationships that are automatically eager loaded.
906      *
907      * @var array<string, string>
908      */
909     protected $withDefault = [
910         'user',
911         'keluhan',
912         'statusPengaduan',
913     ];
914
915     /**
916      * The relationships that are automatically eager loaded.
917      *
918      * @var array<string, string>
919      */
920     protected $withDefault = [
921         'user',
922         'keluhan',
923         'statusPengaduan',
924     ];
925
926     /**
927      * The relationships that are automatically eager loaded.
928      *
929      * @var array<string, string>
930      */
931     protected $withDefault = [
932         'user',
933         'keluhan',
934         'statusPengaduan',
935     ];
936
937     /**
938      * The relationships that are automatically eager loaded.
939      *
940      * @var array<string, string>
941      */
942     protected $withDefault = [
943         'user',
944         'keluhan',
945         'statusPengaduan',
946     ];
947
948     /**
949      * The relationships that are automatically eager loaded.
950      *
951      * @var array<string, string>
952      */
953     protected $withDefault = [
954         'user',
955         'keluhan',
956         'statusPengaduan',
957     ];
958
959     /**
960      * The relationships that are automatically eager loaded.
961      *
962      * @var array<string, string>
963      */
964     protected $withDefault = [
965         'user',
966         'keluhan',
967         'statusPengaduan',
968     ];
969
970     /**
971      * The relationships that are automatically eager loaded.
972      *
973      * @var array<string, string>
974      */
975     protected $withDefault = [
976         'user',
977         'keluhan',
978         'statusPengaduan',
979     ];
980
981     /**
982      * The relationships that are automatically eager loaded.
983      *
984      * @var array<string, string>
985      */
986     protected $withDefault = [
987         'user',
988         'keluhan',
989         'statusPengaduan',
990     ];
991
992     /**
993      * The relationships that are automatically eager loaded.
994      *
995      * @var array<string, string>
996      */
997     protected $withDefault = [
998         'user',
999         'keluhan',
1000         'statusPengaduan',
1001     ];
1002
1003     /**
1004      * The relationships that are automatically eager loaded.
1005      *
1006      * @var array<string, string>
1007      */
1008     protected $withDefault = [
1009         'user',
1010         'keluhan',
1011         'statusPengaduan',
1012     ];
1013
1014     /**
1015      * The relationships that are automatically eager loaded.
1016      *
1017      * @var array<string, string>
1018      */
1019     protected $withDefault = [
1020         'user',
1021         'keluhan',
1022         'statusPengaduan',
1023     ];
1024
1025     /**
1026      * The relationships that are automatically eager loaded.
1027      *
1028      * @var array<string, string>
1029      */
1030     protected $withDefault = [
1031         'user',
1032         'keluhan',
1033         'statusPengaduan',
1034     ];
1035
1036     /**
1037      * The relationships that are automatically eager loaded.
1038      *
1039      * @var array<string, string>
1040      */
1041     protected $withDefault = [
1042         'user',
1043         'keluhan',
1044         'statusPengaduan',
1045     ];
1046
1047     /**
1048      * The relationships that are automatically eager loaded.
1049      *
1050      * @var array<string, string>
1051      */
1052     protected $withDefault = [
1053         'user',
1054         'keluhan',
1055         'statusPengaduan',
1056     ];
1057
1058     /**
1059      * The relationships that are automatically eager loaded.
1060      *
1061      * @var array<string, string>
1062      */
1063     protected $withDefault = [
1064         'user',
1065         'keluhan',
1066         'statusPengaduan',
1067     ];
1068
1069     /**
1070      * The relationships that are automatically eager loaded.
1071      *
1072      * @var array<string, string>
1073      */
1074     protected $withDefault = [
1075         'user',
1076         'keluhan',
1077         'statusPengaduan',
1078     ];
1079
1080     /**
1081      * The relationships that are automatically eager loaded.
1082      *
1083      * @var array<string, string>
1084      */
1085     protected $withDefault = [
1086         'user',
1087         'keluhan',
1088         'statusPengaduan',
1089     ];
1090
1091     /**
1092      * The relationships that are automatically eager loaded.
1093      *
1094      * @var array<string, string>
1095      */
1096     protected $withDefault = [
1097         'user',
1098         'keluhan',
1099         'statusPengaduan',
1100     ];
1101
1102     /**
1103      * The relationships that are automatically eager loaded.
1104      *
1105      * @var array<string, string>
1106      */
1107     protected $withDefault = [
1108         'user',
1109         'keluhan',
1110         'statusPengaduan',
1111     ];
1112
1113     /**
1114      * The relationships that are automatically eager loaded.
1115      *
1116      * @var array<string, string>
1117      */
1118     protected $withDefault = [
1119         'user',
1120         'keluhan',
1121         'statusPengaduan',
1122     ];
1123
1124     /**
1125      * The relationships that are automatically eager loaded.
1126      *
1127      * @var array<string, string>
1128      */
1129     protected $withDefault = [
1130         'user',
1131         'keluhan',
1132         'statusPengaduan',
1133     ];
1134
1135     /**
1136      * The relationships that are automatically eager loaded.
1137      *
1138      * @var array<string, string>
1139      */
1140     protected $withDefault = [
1141         'user',
1142         'keluhan',
1143         'statusPengaduan',
1144     ];
1145
1146     /**
1147      * The relationships that are automatically eager loaded.
1148      *
1149      * @var array<string, string>
1150      */
1151     protected $withDefault = [
1152         'user',
1153         'keluhan',
1154         'statusPengaduan',
1155     ];
1156
1157     /**
1158      * The relationships that are automatically eager loaded.
1159      *
1160      * @var array<string, string>
1161      */
1162     protected $withDefault = [
1163         'user',
1164         'keluhan',
1165         'statusPengaduan',
1166     ];
1167
1168     /**
1169      * The relationships that are automatically eager loaded.
1170      *
1171      * @var array<string, string>
1172      */
1173     protected $withDefault = [
1174         'user',
1175         'keluhan',
1176         'statusPengaduan',
1177     ];
1178
1179     /**
1180      * The relationships that are automatically eager loaded.
1181      *
1182      * @var array<string, string>
1183      */
1184     protected $withDefault = [
1185         'user',
1186         'keluhan',
1187         'statusPengaduan',
1188     ];
1189
1190     /**
1191      * The relationships that are automatically eager loaded.
1192      *
1193      * @var array<string, string>
1194      */
1195     protected $withDefault = [
1196         'user',
1197         'keluhan',
1198         'statusPengaduan',
1199     ];
1200
1201     /**
1202      * The relationships that are automatically eager loaded.
1203      *
1204      * @var array<string, string>
1205      */
1206     protected $withDefault = [
1207         'user',
1208         'keluhan',
1209         'statusPengaduan',
1210     ];
1211
1212     /**
1213      * The relationships that are automatically eager loaded.
1214      *
1215      * @var array<string, string>
1216      */
1217     protected $withDefault = [
1218         'user',
1219         'keluhan',
1220         'statusPengaduan',
1221     ];
1222
1223     /**
1224      * The relationships that are automatically eager loaded.
1225      *
1226      * @var array<string, string>
1227      */
1228     protected $withDefault = [
1229         'user',
1230         'keluhan',
1231         'statusPengaduan',
1232     ];
1233
1234     /**
1235      * The relationships that are automatically eager loaded.
1236      *
1237      * @var array<string, string>
1238      */
1239     protected $withDefault = [
1240         'user',
1241         'keluhan',
1242         'statusPengaduan',
1243     ];
1244
1245     /**
1246      * The relationships that are automatically eager loaded.
1247      *
1248      * @var array<string, string>
1249      */
1250     protected $withDefault = [
1251         'user',
1252         'keluhan',
1253         'statusPengaduan',
1254     ];
1255
1256     /**
1257      * The relationships that are automatically eager loaded.
1258      *
1259      * @var array<string, string>
1260      */
1261     protected $withDefault = [
1262         'user',
1263         'keluhan',
1264         'statusPengaduan',
1265     ];
1266
1267     /**
1268      * The relationships that are automatically eager loaded.
1269      *
1270      * @var array<string, string>
1271      */
1272     protected $withDefault = [
1273         'user',
1274         'keluhan',
1275         'statusPengaduan',
1276     ];
1277
1278     /**
1279      * The relationships that are automatically eager loaded.
1280      *
1281      * @var array<string, string>
1282      */
1283     protected $withDefault = [
1284         'user',
1285         'keluhan',
1286         'statusPengaduan',
1287     ];
1288
1289     /**
1290      * The relationships that are automatically eager loaded.
1291      *
1292      * @var array<string, string>
1293      */
1294     protected $withDefault = [
1295         'user',
1296         'keluhan',
1297         'statusPengaduan',
1298     ];
1299
1300     /**
1301      * The relationships that are automatically eager loaded.
1302      *
1303      * @var array<string, string>
1304      */
1305     protected $withDefault = [
1306         'user',
1307         'keluhan',
1308         'statusPengaduan',
1309     ];
1310
1311     /**
1312      * The relationships that are automatically eager loaded.
1313      *
1314      * @var array<string, string>
1315      */
1316     protected $withDefault = [
1317         'user',
1318         'keluhan',
1319         'statusPengaduan',
1320     ];
1321
1322     /**
1323      * The relationships that are automatically eager loaded.
1324      *
1325      * @var array<string, string>
1326      */
1327     protected $withDefault = [
1328         'user',
1329         'keluhan',
1330         'statusPengaduan',
1331     ];
1332
1333     /**
1334      * The relationships that are automatically eager loaded.
1335      *
1336      * @var array<string, string>
1337      */
1338     protected $withDefault = [
1339         'user',
1340         'keluhan',
1341         'statusPengaduan',
1342     ];
1343
1344     /**
1345      * The relationships that are automatically eager loaded.
1346      *
1347      * @var array<string, string>
1348      */
1349     protected $withDefault = [
1350         'user',
1351         'keluhan',
1352         'statusPengaduan',
1353     ];
1354
1355     /**
1356      * The relationships that are automatically eager loaded.
1357      *
1358      * @var array<string, string>
1359      */
1360     protected $withDefault = [
1361         'user',
1362         'keluhan',
1363         'statusPengaduan',
1364     ];
1365
1366     /**
1
```

menghubungkan dengan entitas terkait dan accessor `getCreatedAtAttribute()` untuk mengkonversi waktu `created_at` ke zona waktu Asia/Jakarta. Model juga dilengkapi dengan method `index()` yang mengambil data pengaduan dengan eager loading relasi user, keluhan, dan statusPengaduan, kemudian mengurutkan berdasarkan `created_at` secara descending dan membatasi hasil menjadi 5 data teratas untuk ditampilkan di dashboard superadmin.

Lampiran 7. Sertifikat HKI


REPUBLIK INDONESIA
KEMENTERIAN HUKUM

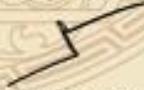
SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan	: ECU02025085900, 9 Juli 2025
Pencipta	
Nama	: Arden Arjuna Mulya, Dega Surana Wibowo, S.T., M.Kom. dkk
Alamat	: JL. IR.H.JUANDA NO.91 RT 002/RW 004 Desa Pakembaran, Slawi, Kab. Tegal, Jawa Tengah, 52413
Kewarganegaraan	: Indonesia
Pemegang Hak Cipta	
Nama	: Pusat Penelitian dan Pengabdian Masyarakat (P3M) Politeknik Harapan Bersama
Alamat	: Jalan Malantri No. 9, Pesurungan Lor, Kecamatan Margadana, Margadana, Kota Tegal, Jawa Tengah, 52142
Kewarganegaraan	: Indonesia
Jenis Ciptaan	: Program Komputer
Judul Ciptaan	: RANCANG BANGUN APLIKASI PELAYANAN WIFI BERBASIS WEBSITE DENGAN METODE PAYMENT GATEWAY DAN CUSTOMER COMPLAIN HANDLING MODEL (CCHM) PADA CV. EVAN NETWORK
Tanggal dan tempat ditandatangani untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	: 9 Juli 2025, di Kota Tegal
Jangka waktu perlindungan	: Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengantunan.
Nomor Pencatatan	: 000926251

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL
u.b
Direktur Hak Cipta dan Desain Industri


Agung Darmasasongko, S.H., M.H.
NIP. 196912261994031001





Disclaimer:
1. Untuk hal pemohon memberikan keterangan tidak benar dengan sumpah-penyataan, Misi ini beresung untuk menuliskan sumpah pemohon.
2. Surat Pencatatan ini valid digital secara elektronik menggunakan segel elektronik yang diterbitkan oleh Badan Besar Sertifikasi Elektronik, Badan Usaha dan Sanksi Negara.
3. Surat Pencatatan ini dapat dibuktikan keabsahannya dengan memindai kode QR pada dokumen ini dan informasi akan ditayangkan dalam browser.

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Arden Arjuna Mulvi	JL.IR.JEJUANDA NO.91 RT 002/RW 004 Desa Pakembaran Slawi, Kab. Tegal
2	Doga Saroni Wibowo, S.T., M.Kom.	Perumahan Sapphire Regency Blok H, No. 1, RT.004, RW.001, Kelurahan Pulosari Brebes, Kab. Brebes
3	Taufiq Abidin, S.Pl., M.Kom.	Jl. Kaligangsa Aeri Timur VIII No.33 RT. 3 RW. 7 Perum Kaligangsa Aeri Kelurahan Kaligangsa Magadana, Kota Tegal



Lampiran 8. Lembar Bimbingan



D IV TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Arden Arjuna Mulvi
 NIM : 21090128
 No. Ponsel : 087824034008
 Judul TA : RANCANG BANGUN APLIKASI PELAYANAN WIFI BERBASIS WEBSITE DENGAN METODE PAYMENT GATEWAY DAN CUSTOMER COMPLAINT HANDLING MODEL (CCHM) PADA CV. EVAN NETWORK
 Dosen Pembimbing I : Dega Suroño Wibowo, S.T., M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	25/2025 4	perbaiki use case sesuai dengan standar. baca jurnal/buku dari roni ratno wahono. 3..		<i>[Signature]</i>
2	2/5/2025	perbaiki use case sesuai dengan standar		<i>[Signature]</i>
3	16/5/2025	buat mock up dari aplikasi berdasarkan use case		<i>[Signature]</i>
4	27/5/2025	alur teknis ditambahkan proses bisnis ditambahkan		<i>[Signature]</i>

5.	3/6 2025	<ul style="list-style-type: none"> * Mulai develop skripsi * dilakukan uji fungsi * gunakan selenium 	f
6.	7/7 2025	<ul style="list-style-type: none"> * manah banyak fungsi yang tidak jalan * Verifikasi email sebelum atau g 	f
7.	8/7 2025	<p>Laporan kesalahan</p> <p>libint Hard file.</p> <ul style="list-style-type: none"> - Sitasi diintan - class diagram - data penelitian (rawan) - lengkap. 	f
8	9/7 2025	<p>Bab 1</p> <p>↓</p> <p>Bab akhir</p> <p>— de</p>	f



D IV TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Arden Arjuna Mulvi
NIM : 21090128
No. Ponsel : 087824034008
Judul TA : RANCANG BANGUN APLIKASI
PELAYANAN WIFI BERBASIS WEBSITE
DENGAN METODE PAYMENT GATEWAY
DAN CUSTOMER COMPLAINT
HANDLING MODEL (CCHM) PADA
CV. EVAN NETWORK

Dosen Pembimbing II : Taufiq Abidin, S.Pd., M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	2/5-25	Konsep	oke → lanjut ke ERD	
2.	7/5-25	ERD	di sempurnakan, & generate mjd desain database	
3.	8/5-25	Database	di sempurnakan dg desain yg sudah & buat sebelumnya - tambahkan data dummy >= 1 smt	

4.	2/5-25	Database	<ul style="list-style-type: none"> - hindari redundansi - dibuat sudah sdh berjalan >= 1 smt 	
5.	18/6-25	Database	<ul style="list-style-type: none"> - dibuat real time - tambahkan trigger untuk mengubah ledom update at 	
6.	24/6-25	Database	<ul style="list-style-type: none"> - ok - lanjut produk 80% 	
7.	7/7-25	Produk	<ul style="list-style-type: none"> - ok - lanjut laporan 	
8.	8/7-25	laporan	<ul style="list-style-type: none"> Ok, siap ujian 	

--	--	--	--	--

Tegal, 8 Juli 2015
Dosen Pembimbing II



Taufiq Abdin, S.Pd., M.Kom.
NIPY. 06.014.184