

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama pelaksanaan kerja magang di PT Braincode Digital Teknologi, mahasiswa magang ditempatkan pada posisi sebagai *Front-End Developer*. Dalam menjalankan peran tersebut, posisi ini berada di bawah koordinasi langsung *Technical Lead* yang sekaligus bertindak sebagai pembimbing lapangan selama kegiatan magang berlangsung. Posisi ini berada dalam struktur divisi *Development*, yang berfokus pada pengembangan dan optimalisasi antarmuka sistem berbasis web.

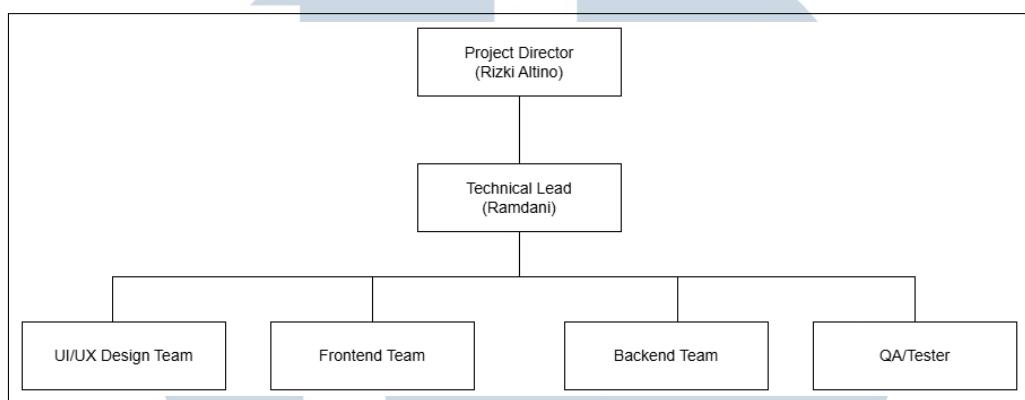
Sebagai *Front-End Developer*, tanggung jawab yang diemban mencakup mengimplementasikan serta mengembangkan komponen antarmuka pengguna (*user interface*) agar sesuai dengan kebutuhan sistem dan standar perusahaan. Selain itu, koordinasi juga dilakukan secara aktif dengan berbagai tim lainnya, termasuk *Backend Team* dan *UI/UX Design Team*, untuk memastikan integrasi antarmuka dengan layanan sistem berjalan dengan baik serta memberikan pengalaman pengguna (*user experience*) yang optimal.

Berikut merupakan daftar mahasiswa yang melaksanakan kegiatan magang pada periode yang sama di PT Braincode Digital Teknologi:

- Nama: Reinhard Javera Maheswara
Asal Kampus: Universitas Multimedia Nusantara
Jurusan: Informatika
Angkatan: 2022
- Nama: Muhammad Tristan Ajibrilyan Nandipinto
Asal Kampus: Universitas Multimedia Nusantara
Jurusan: Informatika
Angkatan: 2022

Salah satu proyek utama yang dikerjakan selama masa magang di PT Braincode Digital Teknologi adalah Core-AN Dashboard, khususnya pada Use Case Transport Failure Optimization (UC1). Proyek ini merupakan bagian dari sistem Core-AN yang dirancang untuk mendukung pemantauan serta optimalisasi penanganan kegagalan transportasi pada jaringan inti.

Dalam pelaksanaan proyek Core-AN Dashboard, struktur tim kerja disusun secara hierarkis dengan pembagian peran yang jelas untuk memastikan efisiensi pengembangan sistem. Struktur organisasi proyek dapat dilihat pada gambar berikut:



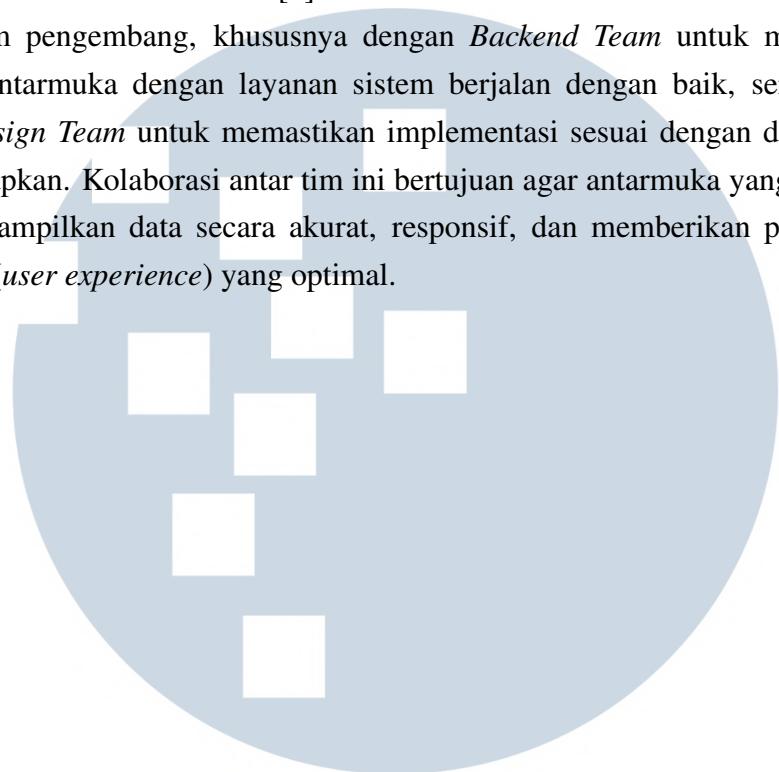
Gambar 3.1. Struktur organisasi proyek Core-AN Dashboard

Berdasarkan Gambar 3.1, struktur organisasi proyek dipimpin oleh *Project Director* yang bertanggung jawab atas arah strategis dan pengawasan keseluruhan proyek. Di bawahnya, terdapat *Technical Lead* yang berfungsi sebagai koordinator teknis dan penghubung antara manajemen dengan tim pengembangan. *Technical Lead* juga berperan sebagai pembimbing lapangan yang memberikan arahan teknis kepada seluruh anggota tim.

Tim pengembangan terbagi menjadi empat unit kerja utama, yaitu *UI/UX Design Team* yang bertanggung jawab dalam perancangan antarmuka dan pengalaman pengguna, *Frontend Team* yang mengimplementasikan desain menjadi kode antarmuka aplikasi, *Backend Team* yang mengembangkan logika bisnis dan integrasi sistem di sisi server, serta *QA/Tester* yang memastikan kualitas sistem melalui pengujian menyeluruh. Pendekatan berbasis komponen dalam pengembangan antarmuka telah menjadi praktik standar dalam pengembangan aplikasi web modern, di mana komponen dapat digunakan kembali dan dipelihara dengan lebih efisien [3].

Dalam proyek ini, mahasiswa magang ditempatkan pada *Frontend Team* dan berfokus pada pengembangan serta pengoptimalan sisi *front-end*. Tugas yang dikerjakan mencakup pembuatan dan penyesuaian tampilan antarmuka setiap halaman berdasarkan kebutuhan *use case*, implementasi komponen interaktif, serta peningkatan performa dan konsistensi tampilan sistem. Implementasi komunikasi antara *front-end* dan *back-end* menggunakan arsitektur RESTful Web Service

dengan autentikasi JWT telah terbukti efektif dalam menjaga keamanan dan integritas data sistem informasi [4]. Koordinasi dilakukan secara intensif dengan seluruh tim pengembang, khususnya dengan *Backend Team* untuk memastikan integrasi antarmuka dengan layanan sistem berjalan dengan baik, serta dengan *UI/UX Design Team* untuk memastikan implementasi sesuai dengan desain yang telah ditetapkan. Kolaborasi antar tim ini bertujuan agar antarmuka yang dibangun dapat menampilkan data secara akurat, responsif, dan memberikan pengalaman pengguna (*user experience*) yang optimal.



3.2 Tugas yang Dilakukan

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Mengikuti rangkaian <i>meeting</i> awal untuk menentukan proyek yang akan dikerjakan serta teknologi yang digunakan. Selain itu, dilakukan pembelajaran dokumentasi <i>SolidJS</i> sebagai <i>framework</i> utama yang digunakan dalam pengembangan <i>front-end</i> sistem.
2	Mempelajari struktur folder proyek yang telah ada serta mulai mengembangkan komponen awal pada halaman <i>Weekly Summary Report</i> , meliputi pembuatan komponen <i>Summary CNBQ</i> dan <i>UTIL</i> , serta pengembangan komponen <i>Achievement Port</i> menggunakan <i>AG Grid</i> agar dapat digunakan secara <i>reusable</i> .
3	Mengembangkan beberapa komponen tabel pada halaman <i>Weekly Summary Report</i> , yaitu <i>List Discard Datacomm</i> , <i>List Error Datacomm</i> , dan <i>List High Util Datacomm</i> . Selain itu, dilakukan proses <i>detailed</i> untuk menyesuaikan tampilan dengan desain <i>mock-up</i> pada <i>Figma</i> .
4	Melakukan perbaikan berbagai <i>bug</i> pada komponen tabel, khususnya terkait <i>layout</i> , <i>responsiveness</i> , dan konsistensi tampilan ketika ukuran tabel berubah. Proses <i>detailed</i> lanjutan juga dilakukan untuk menjaga kesesuaian dengan desain antarmuka.
5	Mempelajari sistem <i>ERPNext</i> sebagai referensi dashboard, kemudian mulai mengembangkan halaman Sanity Check dengan membuat <i>layout</i> , <i>filter dropdown</i> untuk <i>granularity</i> , <i>date range picker</i> , serta tombol <i>reset filter</i> .
6	Mengimplementasikan <i>trend chart</i> menggunakan <i>amCharts</i> serta tabel <i>NDM Data Source</i> menggunakan <i>AG Grid</i> . Selain itu, dilakukan sinkronisasi awal layanan data menggunakan <i>dummy data</i> dan perbaikan <i>bug</i> pada tabel dan <i>date picker</i> .

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan magang (lanjutan)

Minggu Ke -	Pekerjaan yang dilakukan
7	Melakukan penyempurnaan tampilan halaman <i>Sanity Check</i> , termasuk perbaikan <i>sidebar</i> , <i>light mode</i> , serta penyesuaian <i>hover interaction</i> pada grafik. Koordinasi rutin dilakukan melalui <i>daily meeting</i> .
8	Melakukan perubahan penamaan halaman dari <i>Sanity Check</i> menjadi Consistency Check , serta menyesuaikan seluruh teks dan <i>constant</i> agar konsisten. Selain itu, dilakukan refaktor komponen judul agar mendukung <i>dynamic title</i> .
9	Meningkatkan konfigurasi grafik dengan menambahkan <i>axis label</i> , rotasi teks, serta <i>formatter</i> tanggal. Dilakukan pula refaktor fungsi parsing tanggal untuk mendukung berbagai format data.
10	Mengembangkan fitur interaksi tabel seperti <i>row selection</i> , <i>highlight</i> , dan <i>event handler</i> untuk menjaga konsistensi pengalaman pengguna saat melakukan interaksi pada tabel data.
11	Mengimplementasikan logika pemilihan tanggal otomatis untuk <i>granularity daily</i> dan <i>weekly</i> , termasuk perhitungan <i>ISO week</i> , penentuan rentang minggu, serta penyederhanaan <i>data pipeline</i> dengan menghapus opsi <i>monthly</i> .
12	Melakukan evaluasi dan penyempurnaan aspek visual seperti kontras warna, <i>visual hierarchy</i> , serta perbaikan <i>tooltip</i> agar hanya menampilkan data yang relevan dengan kondisi aktif grafik.
13	Mengembangkan fitur Completeness Check (UC1) , meliputi pembuatan tombol khusus, pengelolaan <i>state</i> , integrasi dataset berbeda, serta pengujian <i>toggle mode</i> antara data normal dan data <i>completeness</i> .
14	Melakukan integrasi layanan <i>API</i> untuk UC1 , termasuk penyesuaian <i>endpoint</i> , pemetaan <i>response</i> , serta pengujian <i>fallback data</i> untuk menjaga stabilitas sistem ketika layanan belum sepenuhnya tersedia.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan magang (lanjutan)

Minggu Ke -	Pekerjaan yang dilakukan
15	Melakukan penyesuaian akhir agar seluruh data yang ditampilkan hanya berasal dari UC1 , serta memperbaiki perilaku <i>tooltip</i> dan <i>legend</i> agar tetap konsisten saat sebagian data disembunyikan.
16	Menyusun dan menyempurnakan dokumentasi <i>API</i> untuk beberapa halaman utama, meliputi <i>Home Page</i> , <i>Summary Symptom</i> , dan <i>Quality Event</i> . Dokumentasi dirapikan dengan menghilangkan bagian duplikatif tanpa mengurangi informasi penting.

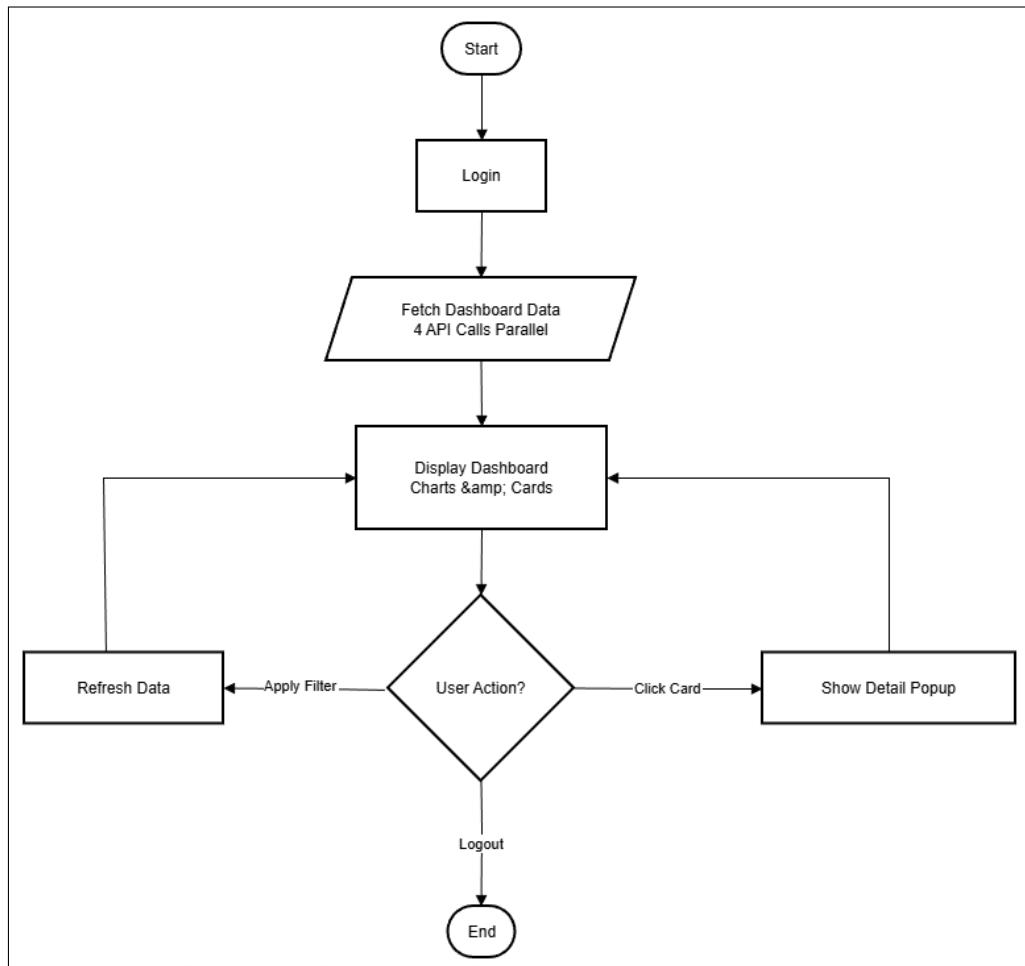
3.3 Uraian Pelaksanaan Magang

Pada bagian ini akan dijelaskan uraian pelaksanaan magang yang berfokus pada proses perancangan dan implementasi sistem yang dikembangkan selama kegiatan magang berlangsung. Pembahasan dimulai dari tahap perancangan sistem sebagai fondasi utama dalam pengembangan aplikasi, yang mencakup perancangan alur kerja (*flowchart*) serta visualisasi tampilan pada setiap halaman sistem. Perancangan ini bertujuan untuk menggambarkan alur interaksi pengguna, proses pengolahan data, serta hubungan antar komponen sistem secara terstruktur dan mudah dipahami sebelum masuk ke tahap implementasi.

3.3.1 Perancangan Sistem

A Flowchart Halaman Home

Berikut ini disajikan flowchart yang menggambarkan alur kerja halaman Home pada sistem *Transport Issue Detection*. Diagram ini menunjukkan proses mulai dari autentikasi pengguna hingga interaksi dengan berbagai komponen dashboard.



Gambar 3.2. Flowchart home

Berdasarkan Gambar 3.2, flowchart halaman Home menggambarkan alur penggunaan sistem sejak pengguna pertama kali mengakses aplikasi hingga sesi penggunaan berakhir. Proses diawali dengan tahap login dan autentikasi, di mana pengguna memasukkan *username* dan *password* untuk memperoleh JWT token sebagai penanda sesi yang disimpan pada *local storage*. Penggunaan JWT untuk autentikasi telah menjadi standar dalam pengembangan aplikasi web modern karena sifatnya yang *stateless*, aman, dan efisien untuk komunikasi antar layanan [5]. Setelah autentikasi berhasil, sistem melakukan pemuatan data dashboard dengan menjalankan empat pemanggilan API secara paralel untuk menampilkan berbagai informasi utama, meliputi *Nation Donut Chart*, *Trend Line Chart*, peta distribusi regional Indonesia, serta *group donut chart*. Pendekatan pemanggilan data secara paralel ini bertujuan untuk meningkatkan performa sistem, mempercepat waktu pemuatan halaman, serta memungkinkan proses *progressive rendering*.

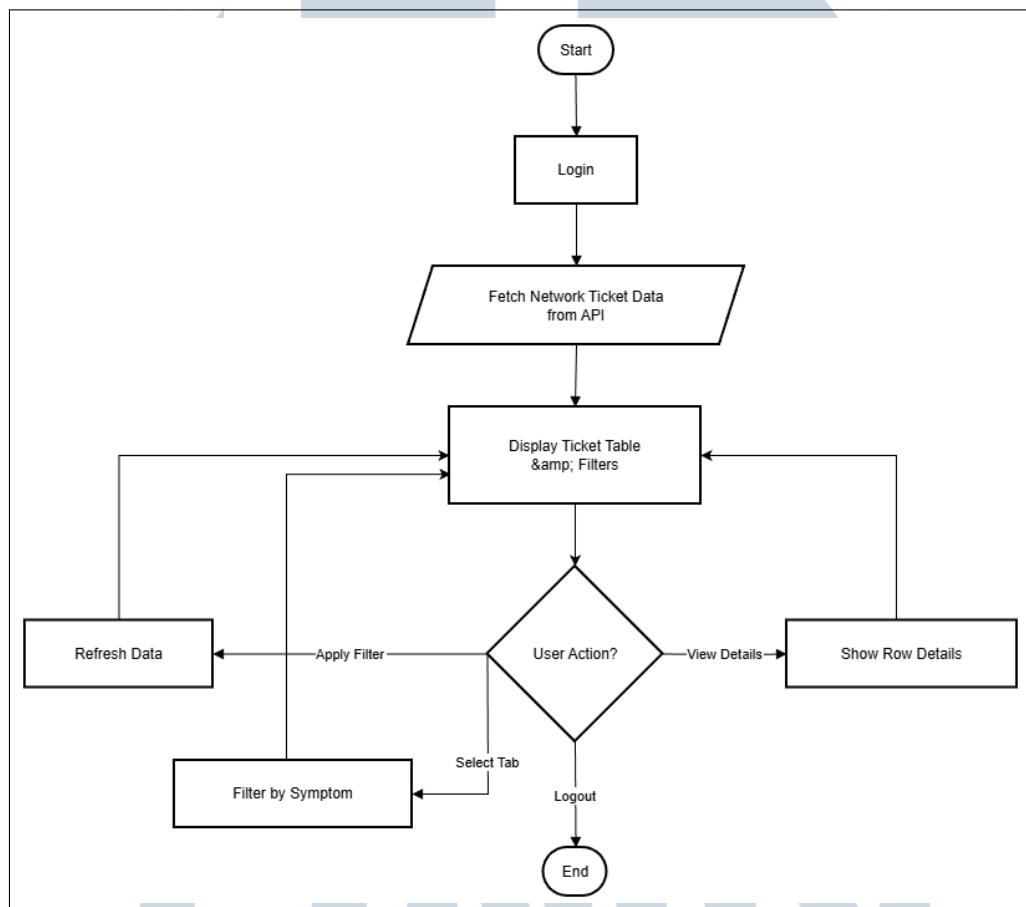
agar antarmuka dapat ditampilkan secara bertahap tanpa menghambat interaksi pengguna.

Setelah dashboard berhasil ditampilkan, pengguna dapat melakukan berbagai interaksi utama, seperti menerapkan *filter* berdasarkan wilayah dan rentang tanggal, memperluas (*expand*) kartu regional maupun grup untuk melihat detail data melalui modal pop-up, serta menyesuaikan tampilan untuk memaksimalkan area visualisasi tertentu. Desain dashboard yang responsif dan interaktif terbukti meningkatkan efektivitas pemantauan jaringan dengan menyajikan informasi yang mudah dipahami dan dapat diakses [6]. Setiap aksi pengguna akan memicu proses pembaruan data dan secara otomatis dicatat ke dalam sistem *activity logging*. Selama proses pengambilan data, sistem menampilkan indikator *loading* pada setiap komponen untuk memberikan umpan balik visual kepada pengguna.



B Flowchart Halaman Network Ticket

Berikut ini disajikan flowchart yang menggambarkan alur kerja halaman Network Ticket. Diagram ini menunjukkan proses pengelolaan tiket jaringan mulai dari autentikasi hingga berbagai interaksi pengguna dengan data tiket.



Gambar 3.3. Flowchart network ticket

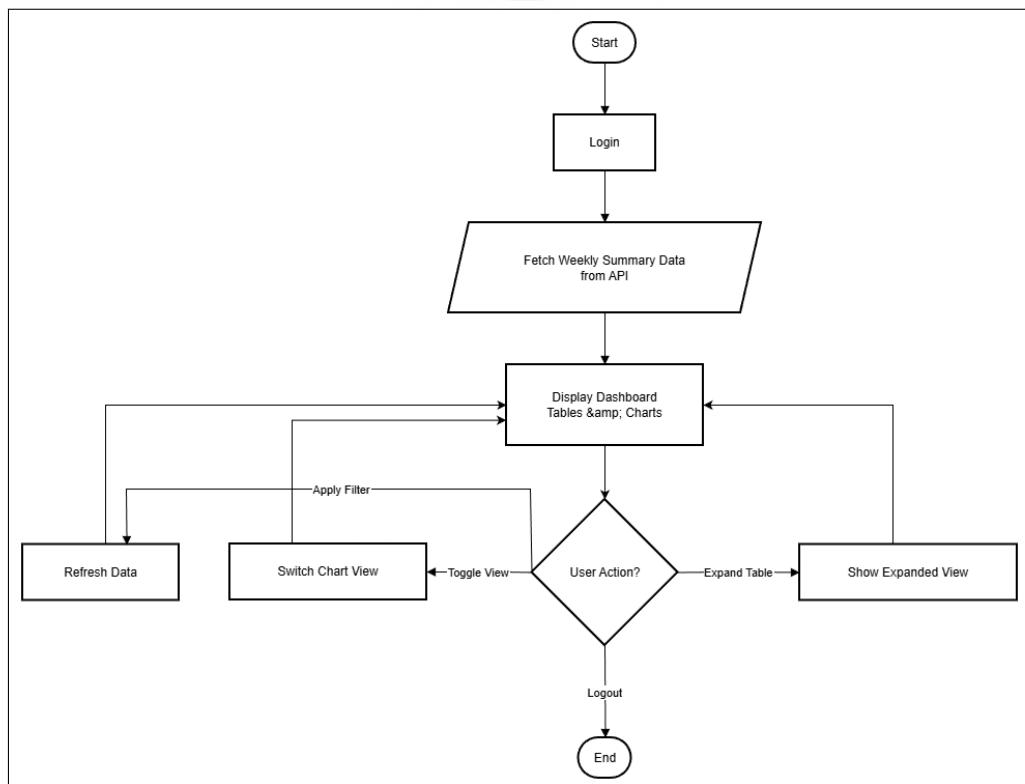
Berdasarkan Gambar 3.3, flowchart halaman Network Ticket menggambarkan alur kerja halaman tiket jaringan yang dimulai dari proses autentikasi pengguna hingga data tiket ditampilkan secara lengkap. Ketika pengguna mengakses halaman ini, sistem terlebih dahulu melakukan pengecekan token autentikasi melalui *AuthGuard* untuk memastikan sesi pengguna masih valid. Apabila token tidak valid atau telah kedaluwarsa, pengguna akan diarahkan kembali ke halaman login. Jika autentikasi berhasil, sistem melanjutkan proses dengan memuat halaman Network Ticket dan melakukan pemanggilan API untuk mengambil data tiket jaringan dari backend. Selama proses ini berlangsung, sistem menampilkan indikator *loading* agar pengguna mengetahui bahwa data sedang

diproses, sebelum akhirnya menampilkan dashboard utama berisi filter, tab jenis gangguan, serta tabel tiket jaringan.

Setelah data berhasil ditampilkan, pengguna dapat melakukan berbagai interaksi utama untuk menganalisis tiket jaringan secara fleksibel. Pengguna dapat menerapkan filter berdasarkan *status*, *granularity* waktu, dan rentang tanggal, memilih jenis gangguan melalui tab atau *dropdown symptom*, serta memanfaatkan tabel interaktif berbasis *AG-Grid* yang mendukung fitur sortir, pencarian, pagination, dan tampilan detail per baris tiket. Setiap aksi pengguna, seperti menerapkan filter, berpindah halaman, menyegarkan data, atau membuka detail tiket, akan memicu pembaruan tampilan secara reaktif tanpa harus memuat ulang halaman. Alur penggunaan diakhiri dengan proses logout, di mana sesi pengguna ditutup dan data sementara dihapus dari sistem.

C Flowchart Halaman Weekly Summary

Berikut ini disajikan flowchart yang menggambarkan alur kerja halaman Weekly Summary. Diagram ini menunjukkan proses pemuatan dan interaksi pengguna dengan ringkasan KPI mingguan jaringan.



Gambar 3.4. Flowchart weekly summary

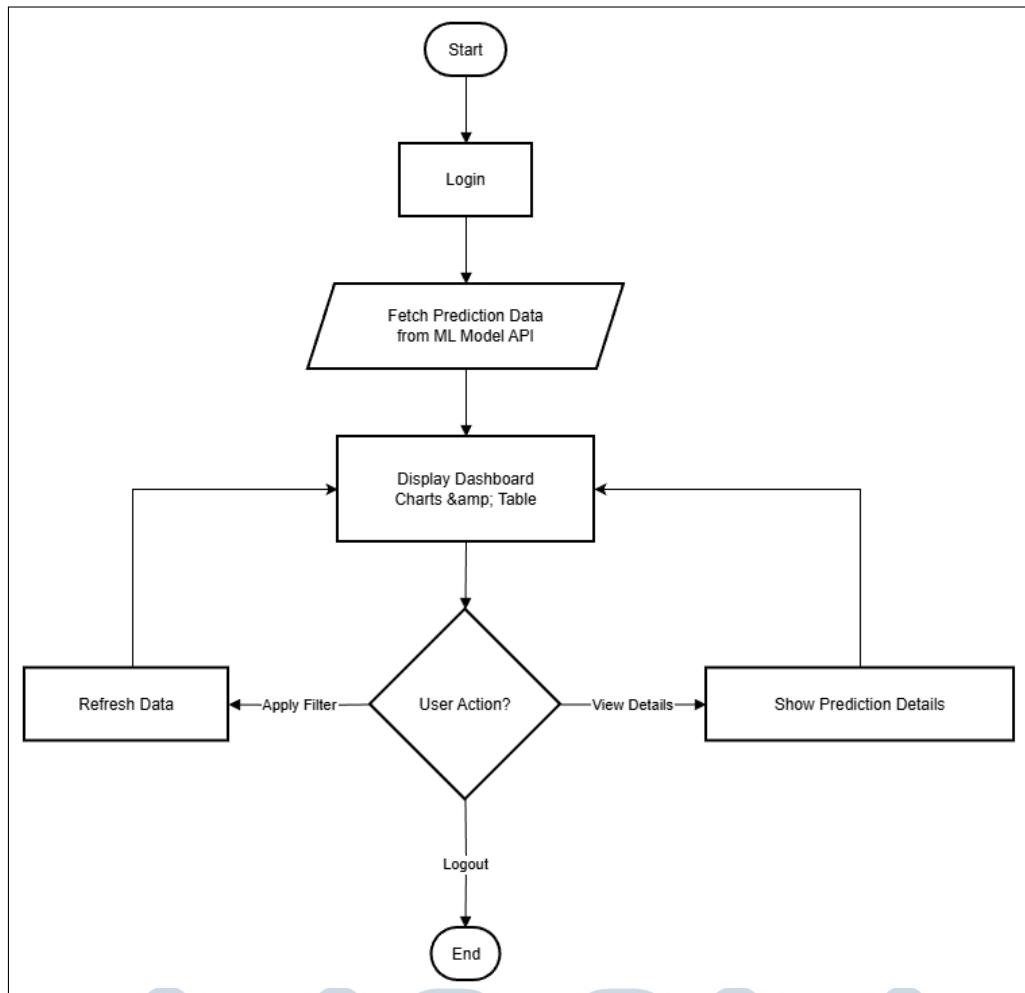
Berdasarkan Gambar 3.4, flowchart halaman Weekly Summary menggambarkan alur kerja sistem ketika pengguna mengakses halaman ringkasan KPI mingguan. Proses diawali dengan pengecekan autentikasi pengguna melalui *AuthGuard* yang memverifikasi *JWT token* pada *localStorage*. Apabila token tidak valid atau sudah kedaluwarsa, sistem akan mengarahkan pengguna kembali ke halaman login. Sebaliknya, jika token valid, sistem akan memuat halaman dan secara bersamaan mengambil berbagai data dari beberapa *Application Programming Interface* (API). Proses pengambilan data dilakukan secara paralel agar waktu pemuatan halaman lebih efisien, mencakup data ringkasan mingguan, pencapaian target, daftar port bermasalah, grafik penggunaan internet, serta grafik utilisasi jaringan.

Setelah data berhasil dimuat, sistem menampilkan dashboard yang terdiri dari beberapa bagian utama, yaitu filter data, tabel ringkasan, tabel pencapaian, daftar port, serta grafik performa jaringan. Pengguna dapat melakukan berbagai interaksi seperti menerapkan filter berdasarkan wilayah, grup layanan, jenis isu, dan minggu tertentu, mengganti tampilan grafik, melihat data berdasarkan rentang waktu, memperluas tabel atau grafik ke tampilan layar penuh, menyegarkan data, hingga keluar dari sistem. Setiap interaksi pengguna akan memicu pembaruan data atau tampilan secara dinamis tanpa harus memuat ulang seluruh halaman.

D Flowchart Halaman Prediction

Berikut ini disajikan flowchart yang menggambarkan alur kerja halaman Prediction. Diagram ini menunjukkan proses prediksi gangguan jaringan berbasis *machine learning* mulai dari autentikasi hingga interaksi pengguna dengan hasil prediksi.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.5. Flowchart prediction

Berdasarkan Gambar 3.5, flowchart halaman Prediction menggambarkan alur sistem ketika pengguna mengakses halaman prediksi gangguan jaringan berbasis *machine learning*. Model machine learning untuk prediksi kegagalan jaringan pada jaringan telekomunikasi heterogen telah menjadi area penelitian yang berkembang pesat, di mana kemampuan memprediksi kegagalan perangkat jaringan semakin diidentifikasi sebagai alat yang efektif untuk meningkatkan keandalan jaringan dan memungkinkan tindakan pemeliharaan preventif [7]. Proses dimulai saat pengguna membuka halaman prediksi, kemudian sistem melakukan pengecekan autentikasi melalui *AuthGuard* dengan memverifikasi *JWT token* pada *localStorage*. Apabila token tidak valid atau telah kedaluwarsa, pengguna akan diarahkan kembali ke halaman login. Jika autentikasi berhasil, sistem memuat halaman prediksi dan mengambil data dari *Prediction API* yang terhubung dengan model *machine learning*. Model ini memanfaatkan data historis gangguan jaringan

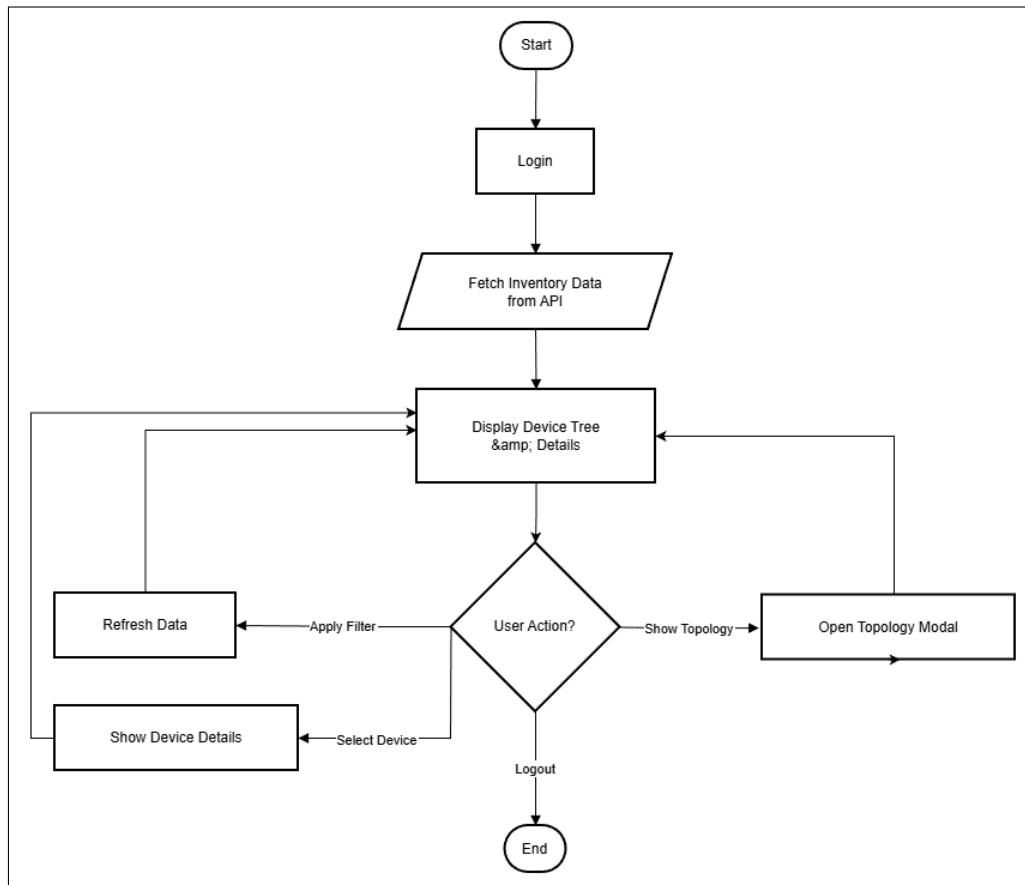
untuk memprediksi potensi masalah di masa mendatang, termasuk jenis gangguan serta perkiraan waktu terjadinya.

Setelah data berhasil dimuat, sistem menampilkan dashboard prediksi yang terdiri dari beberapa komponen utama, yaitu filter data, grafik tren prediksi, grafik distribusi prediksi, dan tabel hasil prediksi. Pengguna dapat menyaring data berdasarkan wilayah, jenis layanan jaringan, dan tanggal prediksi, lalu melihat perubahan hasil secara langsung pada grafik dan tabel. Grafik tren digunakan untuk melihat kecenderungan jumlah gangguan dari waktu ke waktu, sedangkan grafik distribusi menampilkan perbandingan prediksi gangguan antar wilayah. Tabel prediksi menyajikan informasi detail hingga tingkat perangkat dan antarmuka jaringan. Selain itu, pengguna dapat berinteraksi dengan sistem melalui berbagai aksi seperti melihat detail data, menyegarkan hasil prediksi, menghapus filter, hingga keluar dari sistem.

E Flowchart Halaman Inventory

Berikut ini disajikan flowchart yang menggambarkan alur kerja halaman Inventory. Diagram ini menunjukkan proses pengelolaan inventori perangkat jaringan mulai dari autentikasi hingga interaksi pengguna dengan data perangkat.





Gambar 3.6. Flowchart inventory

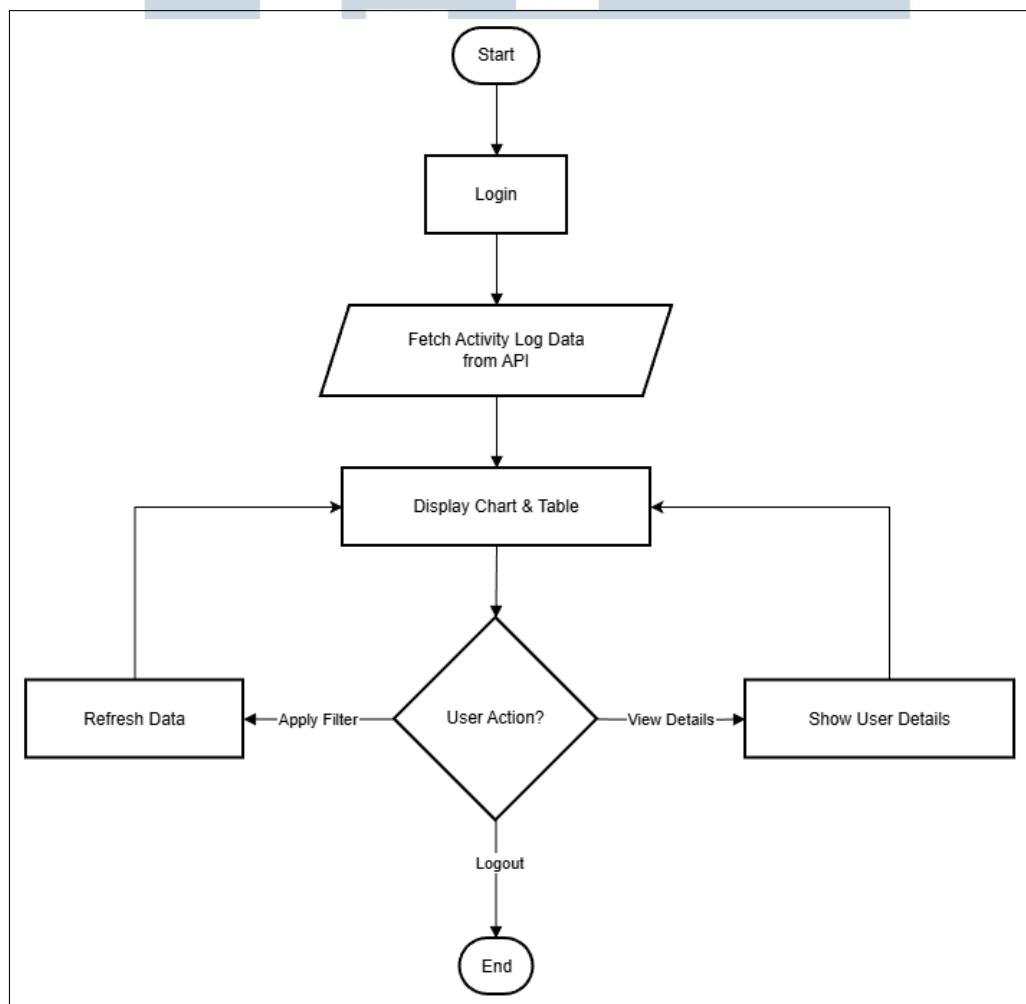
Berdasarkan Gambar 3.6, flowchart halaman Inventory menggambarkan alur sistem saat pengguna mengakses halaman inventori perangkat jaringan. Proses dimulai ketika pengguna membuka halaman inventory, kemudian sistem melakukan pengecekan autentikasi menggunakan *AuthGuard* dengan memverifikasi *JWT token* yang tersimpan pada *localStorage*. Mekanisme autentikasi berbasis token seperti *JWT* ini telah terbukti efektif dalam mengamanakan RESTful API dan sesi pengguna secara stateless, sehingga memungkinkan sistem memastikan hanya pengguna yang valid yang dapat melanjutkan [8]. Jika autentikasi gagal, pengguna akan diarahkan kembali ke halaman login. Apabila berhasil, sistem memuat halaman inventory dan mengambil data perangkat melalui *Inventory API*.

Setelah data berhasil dimuat, sistem menampilkan dashboard inventory yang terdiri dari filter data, sidebar inventori regional, serta detail perangkat. Pengguna dapat menyaring data berdasarkan wilayah, rentang waktu, dan tingkat granularitas, lalu memilih perangkat tertentu melalui struktur hierarki di sidebar. Informasi detail perangkat ditampilkan secara lengkap, mulai dari informasi umum,

daftar antarmuka fisik dan logikal, hingga riwayat gangguan yang pernah terjadi. Selain itu, pengguna juga dapat melakukan berbagai interaksi seperti melihat detail tabel, menyegarkan data, serta menampilkan visualisasi *network topology* dalam bentuk modal untuk memahami keterhubungan perangkat.

F Flowchart Halaman Activity Log

Berikut ini disajikan flowchart yang menggambarkan alur kerja halaman Activity Log. Diagram ini menunjukkan proses pencatatan dan pemantauan aktivitas pengguna dalam sistem.



N U S A N T A R A
Gambar 3.7. Flowchart activity log

Berdasarkan Gambar 3.7, flowchart halaman Activity Log menggambarkan alur sistem dalam mencatat dan menampilkan aktivitas pengguna di dalam aplikasi. Proses dimulai ketika pengguna mengakses halaman activity log,

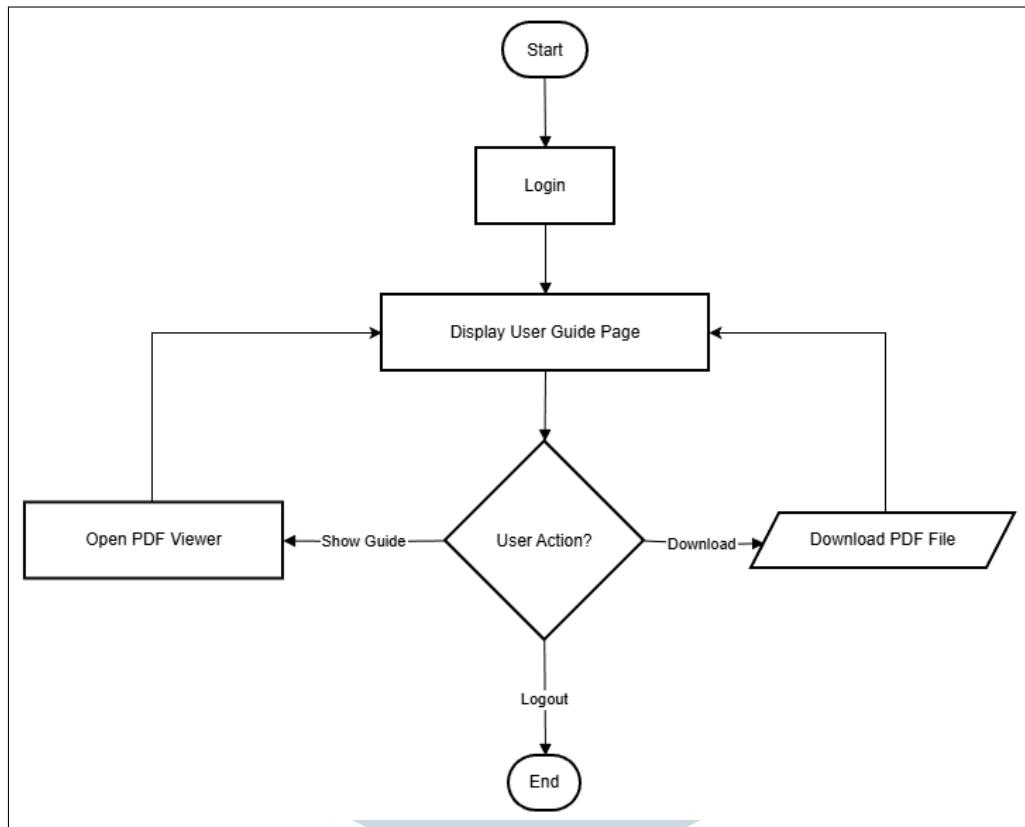
lalu sistem melakukan pemeriksaan autentikasi menggunakan *AuthGuard* dengan memverifikasi *JWT token*. Jika pengguna tidak terautentikasi, sistem akan mengarahkan kembali ke halaman login. Apabila autentikasi berhasil, halaman activity log akan dimuat dan sistem mengambil data aktivitas pengguna melalui *Activity Log API*. Mekanisme ini memastikan bahwa hanya pengguna dengan hak akses tertentu, seperti administrator, yang dapat melihat catatan aktivitas sebagai bagian dari kebutuhan keamanan dan audit sistem.

Setelah data berhasil dimuat, sistem menampilkan dashboard activity log yang terdiri dari filter rentang tanggal, grafik tren aktivitas, dan tabel aktivitas pengguna. Grafik digunakan untuk memperlihatkan pola penggunaan aplikasi dari waktu ke waktu, seperti hari dengan aktivitas tertinggi atau penurunan jumlah pengguna aktif. Sementara itu, tabel menampilkan informasi ringkas setiap pengguna, termasuk jumlah akses dan waktu akses terakhir, sehingga memudahkan identifikasi pengguna aktif maupun tidak aktif. Pengguna juga dapat melakukan interaksi sederhana seperti menyaring data berdasarkan tanggal, menyegarkan data terbaru, atau mengurutkan tabel.

G Flowchart Halaman User Guide

Berikut ini disajikan flowchart yang menggambarkan alur kerja halaman User Guide. Diagram ini menunjukkan proses akses dokumentasi penggunaan aplikasi oleh pengguna.





Gambar 3.8. Flowchart user guide

Berdasarkan Gambar 3.8, flowchart halaman User Guide menjelaskan alur pengguna dalam mengakses dokumentasi penggunaan aplikasi secara aman dan sederhana. Proses dimulai ketika pengguna membuka halaman *User Guide*, lalu sistem melakukan pengecekan autentikasi dengan memverifikasi *JWT token*. Jika pengguna belum login atau sesi telah berakhir, sistem akan mengarahkan kembali ke halaman login. Namun, jika autentikasi berhasil, halaman User Guide langsung ditampilkan tanpa memerlukan pemanggilan API ke backend karena seluruh konten bersifat statis. Halaman ini berfungsi sebagai titik awal dokumentasi dengan menampilkan judul, deskripsi singkat, serta dua tombol utama untuk mengakses panduan pengguna. Penyusunan antarmuka dan pengalaman pengguna yang intuitif seperti ini berpengaruh signifikan terhadap kemudahan penggunaan aplikasi secara keseluruhan [9].

Setelah halaman berhasil dimuat, pengguna dapat memilih beberapa aksi sesuai kebutuhan. Pengguna dapat menekan tombol *Show User Guide* untuk membuka dokumen panduan dalam bentuk PDF melalui *modal viewer*, sehingga isi panduan dapat dibaca langsung di dalam aplikasi dengan fitur navigasi halaman,

zoom, pencarian, cetak, dan unduh. Alternatif lainnya, pengguna dapat memilih Download User Guide untuk langsung mengunduh file PDF ke perangkat tanpa perlu membuka tampilan pratinjau. Selain itu, pengguna juga dapat keluar dari sistem melalui menu logout yang akan mengakhiri sesi dan menghapus data autentikasi.

3.3.2 Antarmuka Sistem dan Hasil Pengembangan

A Halaman Home

Berikut ini disajikan tampilan antarmuka halaman Home yang merupakan dashboard utama sistem Core-AN. Gambar ini menunjukkan tata letak komponen visualisasi dan informasi yang tersedia pada halaman utama.



Gambar 3.9. Halaman Home

Berdasarkan Gambar 3.9, halaman Home merupakan halaman utama pada sistem Core-AN *Use Case Transport Failure Optimization (UCI)* yang berfungsi sebagai dashboard pemantauan kondisi *transport network* secara nasional. Antarmuka ini dibangun menggunakan *framework* SolidJS dan dirancang untuk menampilkan ringkasan distribusi permasalahan transport berdasarkan wilayah *region* dan kelompok jaringan *group*. Melalui halaman ini, pengguna dapat memperoleh gambaran umum kondisi jaringan secara cepat melalui visualisasi peta Indonesia, grafik, serta kartu informasi yang bersifat interaktif.

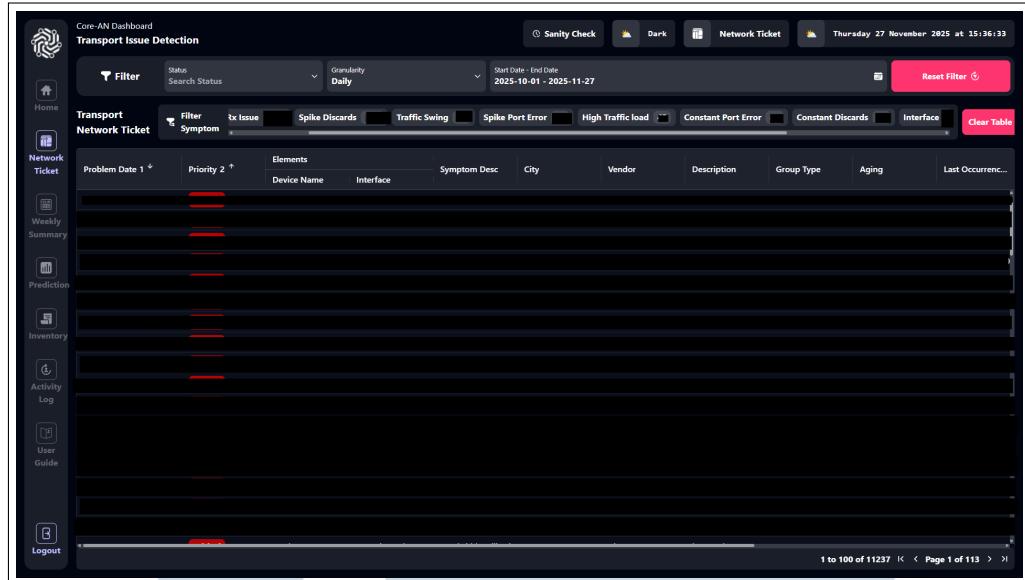
Secara fungsional, halaman Home bekerja dengan melakukan pengambilan data secara terpusat ketika halaman pertama kali diakses. Data yang ditampilkan meliputi distribusi kegagalan transport pada tingkat nasional, regional, dan grup, serta tren permasalahan dalam rentang waktu tertentu. Seluruh data tersebut dikelola melalui global state management sehingga perubahan filter, seperti rentang tanggal, tingkat granularitas (*daily*, *weekly*, atau *monthly*), serta pemilihan region, akan secara otomatis memperbarui seluruh komponen visual yang ditampilkan. Pendekatan ini memungkinkan sistem untuk menyajikan informasi yang konsisten dan responsif tanpa perlu memuat ulang halaman.

Dari sisi interaksi, halaman Home mendukung fitur drill-down analysis, di mana pengguna dapat menelusuri data secara lebih mendalam dengan memilih region atau group tertentu. Interaksi tersebut akan menampilkan *popup* berisi detail tambahan berupa grafik tren dan tabel pendukung. Selain itu, pengguna juga dapat mengatur visibilitas kategori data melalui legenda interaktif pada grafik dan peta. Dengan kombinasi visualisasi geografis, grafik tren, serta pencatatan aktivitas pengguna *activity logging*, halaman Home berperan penting dalam mendukung proses monitoring real-time, analisis perbandingan antar wilayah, serta pengambilan keputusan berbasis data pada sistem Core-AN.

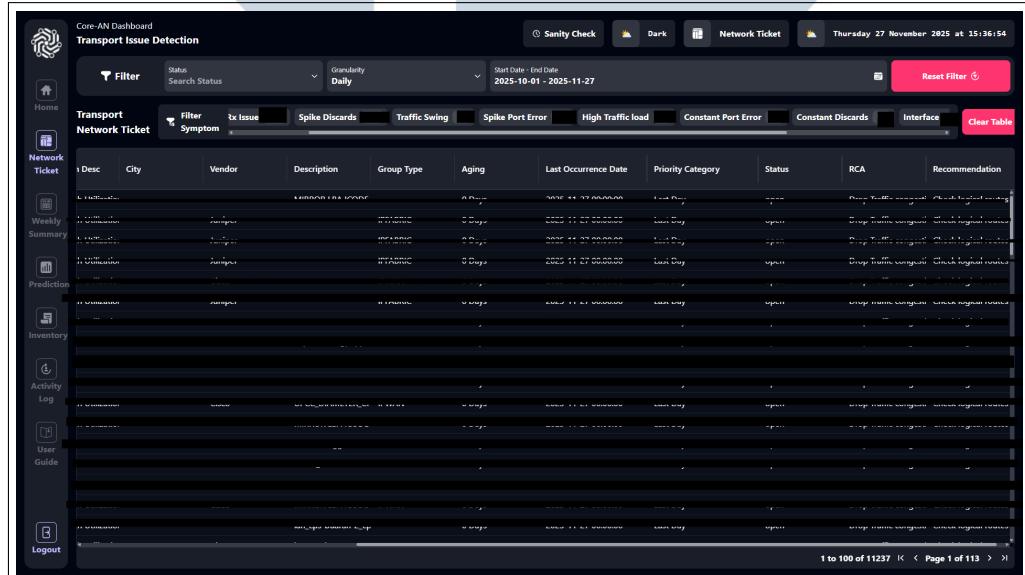
B Halaman Network Ticket

Berikut ini disajikan tampilan antarmuka halaman Network Ticket yang menampilkan data tiket permasalahan jaringan. Gambar dibagi menjadi dua bagian untuk menunjukkan keseluruhan tampilan tabel yang lebar.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.10. Halaman network ticket (sisi kiri)



Gambar 3.11. Halaman network ticket (sisi kanan)

Berdasarkan Gambar 3.10 dan Gambar 3.11, halaman Network Ticket merupakan antarmuka yang digunakan untuk menampilkan data permasalahan jaringan transport secara terperinci dalam bentuk tabel. Halaman ini menyajikan informasi penting seperti tanggal terjadinya masalah, tingkat prioritas, perangkat dan *interface* yang terdampak, deskripsi gangguan, lokasi, vendor, status tiket, hingga rekomendasi penanganan. Dengan penyajian data secara tabular, halaman

ini memungkinkan pengguna untuk melakukan *monitoring* kondisi jaringan secara menyeluruh dan sistematis, terutama pada skala data yang besar.

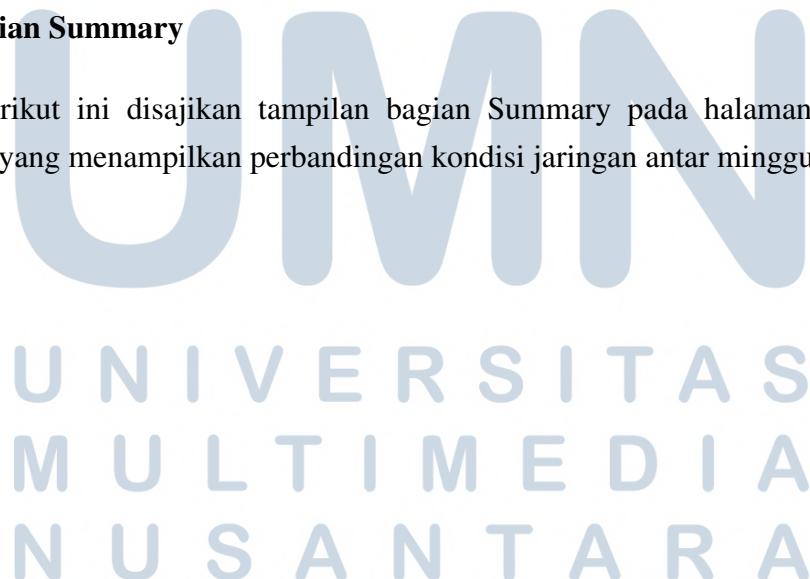
Secara fungsional, halaman Network Ticket dilengkapi dengan berbagai fitur pendukung analisis data, seperti filter, pencarian, dan pagination. Pengguna dapat memfilter tiket berdasarkan status, kategori gangguan *symptom*, serta rentang waktu tertentu sehingga data yang ditampilkan menjadi lebih relevan dengan kebutuhan analisis. Integrasi dengan API backend mengikuti prinsip RESTful yang memastikan komunikasi data yang efisien, terstruktur, dan mudah dipelihara [3]. Selain itu, pengurutan data pada kolom tertentu, seperti tanggal kejadian dan prioritas, memudahkan pengguna dalam mengidentifikasi tiket yang bersifat kritis atau memerlukan penanganan segera.

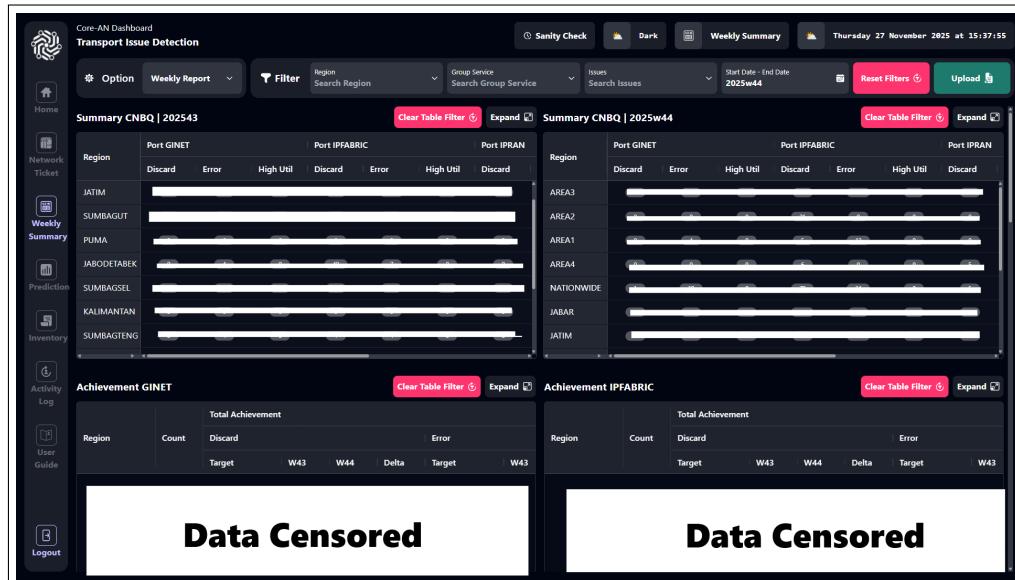
Dari sisi penggunaan, halaman Network Ticket berperan penting bagi berbagai pemangku kepentingan. Tim Network Operation Center (NOC) dapat menggunakan halaman ini untuk melakukan pelacakan dan *monitoring* tiket secara detail, sementara tim *engineering* memperoleh informasi yang diperlukan untuk proses analisis dan penanganan gangguan. Bagi pihak manajemen, data yang tersaji pada halaman ini dapat digunakan sebagai dasar evaluasi kondisi jaringan dan penyusunan laporan.

C Halaman Weekly Summary

C.1 Bagian Summary

Berikut ini disajikan tampilan bagian Summary pada halaman Weekly Summary yang menampilkan perbandingan kondisi jaringan antar minggu.





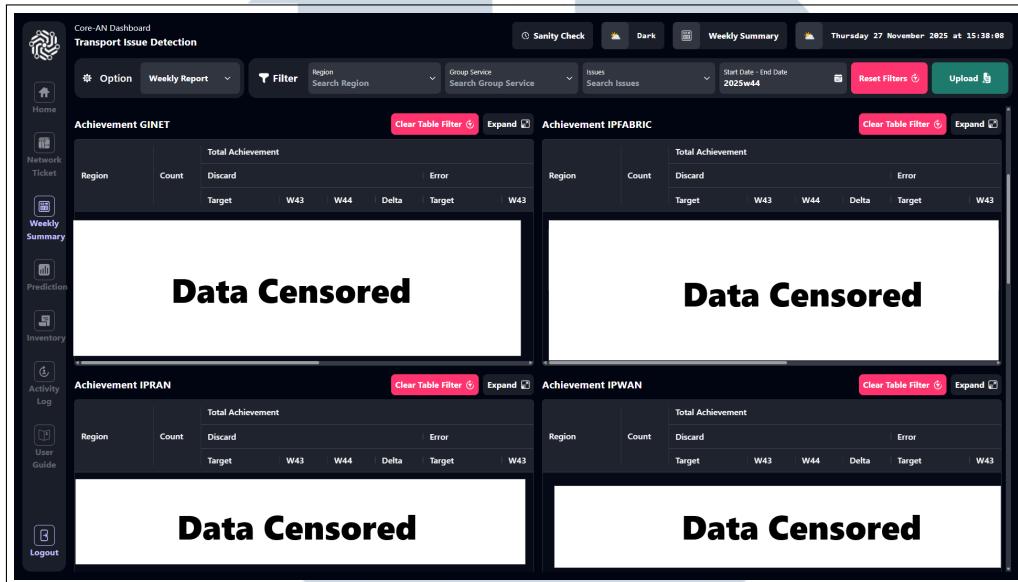
Gambar 3.12. Halaman weekly summary - bagian summary

Berdasarkan Gambar 3.12, bagian Summary merupakan komponen utama pada halaman *Weekly Summary* yang berfungsi untuk menampilkan ringkasan kondisi jaringan dalam bentuk perbandingan antar minggu (*week-over-week comparison*). Pada bagian ini ditampilkan dua tabel yang disusun secara berdampingan, di mana tabel sebelah kiri merepresentasikan data minggu sebelumnya dan tabel sebelah kanan menampilkan data minggu berjalan. Pola desain dashboard untuk sistem monitoring yang efektif memerlukan perhatian khusus terhadap target pengguna, konten yang sesuai, dan presentasi data yang tepat [6]. Setiap tabel menyajikan jumlah isu jaringan berdasarkan region serta tiga kelompok infrastruktur utama, yaitu Port GINET, Port IPFABRIC, dan Port IPRAN, dengan tiga jenis metrik yang diamati, yaitu Discard, Error, dan *High Utilization*.

Secara visual, tabel pada bagian Summary menggunakan tampilan *dark theme* dengan nilai numerik yang disajikan dalam bentuk *circular badge* sehingga lebih mudah dibaca dan dipindai. Nilai yang lebih dari nol akan diberikan penanda visual khusus untuk menyoroti adanya potensi permasalahan yang memerlukan perhatian lebih lanjut. Selain itu, setiap tabel dilengkapi dengan fitur interaktif seperti Clear Table Filter untuk mengembalikan tampilan ke kondisi awal dan Expand untuk menampilkan tabel dalam ukuran penuh dengan informasi yang lebih detail.

C.2 Bagian Achievement

Berikut ini disajikan tampilan bagian Achievement pada halaman Weekly Summary yang menampilkan evaluasi pencapaian kinerja jaringan.



Gambar 3.13. Halaman weekly summary - bagian achievement

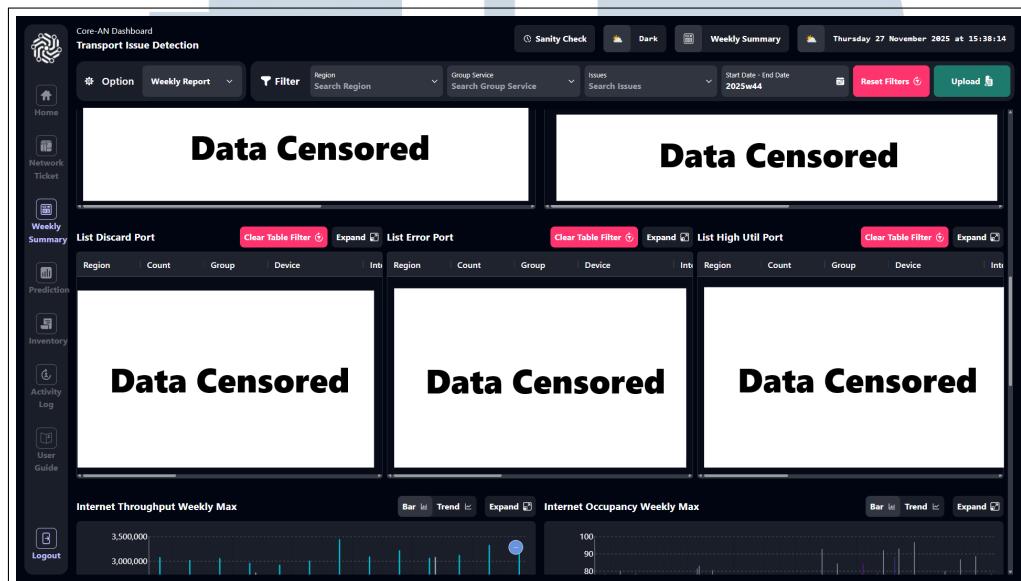
Berdasarkan Gambar 3.13, bagian Achievement merupakan bagian pada halaman *Weekly Summary* yang berfungsi untuk menampilkan evaluasi pencapaian kinerja jaringan dalam periode mingguan. Pada bagian ini ditampilkan empat tabel *Key Performance Indicator* (KPI) yang masing-masing merepresentasikan group service GINET, IPFABRIC, IPRAN, dan IPWAN dalam susunan *grid* 2x2. Setiap tabel menyajikan data berdasarkan region dengan struktur perbandingan antara nilai aktual minggu sebelumnya (W43) dan minggu berjalan (W44) terhadap target yang telah ditentukan. Parameter yang ditampilkan meliputi Count (jumlah data), serta dua metrik utama, yaitu Discard dan Error.

Untuk mempermudah pemahaman pengguna, tabel pada Bagian Achievement dilengkapi dengan indikator visual berbasis warna. Warna hijau menandakan kondisi ideal tanpa permasalahan (nilai nol), warna kuning menunjukkan kondisi yang masih berada dalam batas target, sedangkan warna merah mengindikasikan nilai yang melebihi target dan memerlukan perhatian lebih lanjut. Selain itu, kolom Delta digunakan untuk menunjukkan perubahan kondisi antar minggu, di mana nilai positif (ditandai warna merah) menunjukkan peningkatan jumlah isu, nilai negatif (warna hijau) menunjukkan perbaikan, dan

nilai nol menandakan kondisi yang stabil. Setiap tabel juga menyediakan fitur interaktif seperti Clear Table Filter untuk mengatur ulang data dan Expand untuk menampilkan tabel dalam ukuran penuh.

C.3 Bagian List Port

Berikut ini disajikan tampilan bagian List Port pada halaman Weekly Summary yang menampilkan daftar port jaringan bermasalah.



Gambar 3.14. Halaman weekly summary - bagian list port

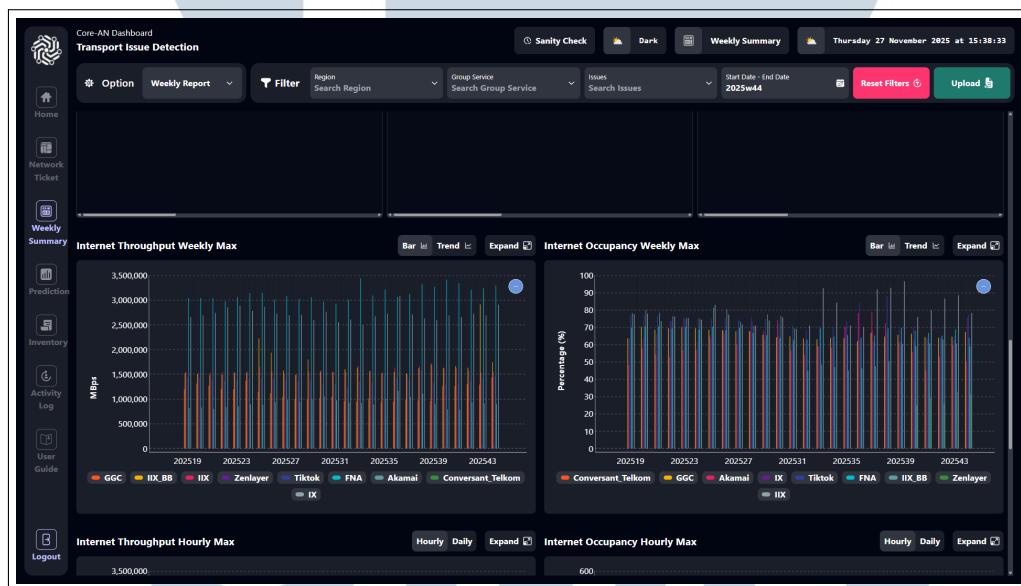
Berdasarkan Gambar 3.14, bagian List Port merupakan bagian pada halaman *Weekly Summary* yang berfungsi untuk menampilkan daftar port jaringan yang terindikasi bermasalah dan memerlukan perhatian lebih lanjut. Pada bagian ini disajikan tiga tabel utama dalam susunan horizontal, yaitu List Discard Port, List Error Port, dan List High Util Port. Masing-masing tabel mengelompokkan port berdasarkan jenis permasalahan, di mana Discard Port menunjukkan port dengan tingkat pembuangan paket (*packet discard*) yang tinggi, Error Port menunjukkan port dengan tingkat kesalahan (*error rate*) yang tinggi, serta High Util Port menunjukkan port dengan tingkat penggunaan bandwidth yang mendekati atau melebihi batas aman. Informasi yang ditampilkan pada setiap tabel meliputi Region, Count (jumlah port bermasalah), Group Service, Device, dan Interface.

Untuk meningkatkan kemudahan analisis, bagian List Port dilengkapi dengan berbagai elemen visual dan fitur interaktif. Kolom Count ditampilkan

dalam bentuk *badge* berwarna untuk menandakan tingkat urgensi, seperti warna kuning untuk peringatan, merah untuk tingkat keparahan tinggi, dan oranye untuk kondisi penggunaan tinggi. Selain itu, setiap tabel menyediakan tombol Clear Table Filter untuk menghapus filter yang diterapkan dan memuat ulang data, serta tombol Expand untuk menampilkan tabel dalam tampilan penuh dengan informasi tambahan. Baris tabel juga dapat diklik untuk menampilkan detail port secara lebih lengkap melalui jendela *popup*, termasuk riwayat kejadian dan grafik tren historis.

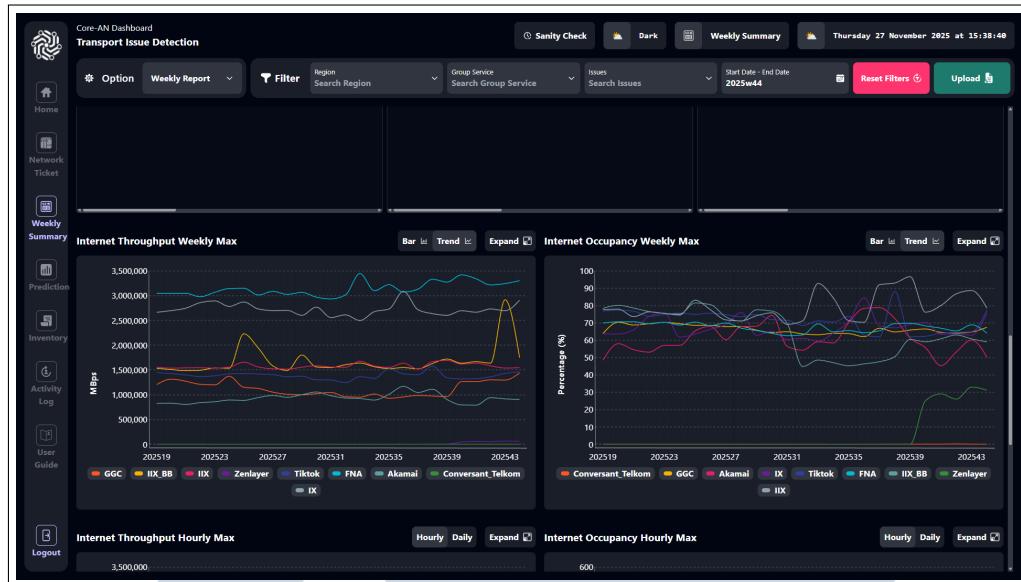
C.4 Bagian Internet Weekly

Berikut ini disajikan tampilan bagian Internet Weekly pada halaman Weekly Summary yang menampilkan ringkasan kondisi trafik jaringan internet mingguan dalam dua format visualisasi.



Gambar 3.15. Halaman weekly summary - bagian internet weekly (bar)

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.16. Halaman weekly summary - bagian internet weekly (trend)

Berdasarkan Gambar 3.15 dan Gambar 3.16, bagian Internet Weekly merupakan bagian pada halaman *Weekly Summary* yang berfungsi sebagai ringkasan kondisi trafik jaringan internet dalam periode mingguan. Section ini menampilkan dua indikator utama, yaitu Internet Throughput Weekly Max dan Internet Occupancy Weekly Max, yang masing-masing merepresentasikan nilai throughput maksimum serta tingkat pemanfaatan kapasitas bandwidth tertinggi dalam satu minggu. Informasi ini disajikan dalam bentuk grafik ringkas untuk memberikan gambaran cepat mengenai performa jaringan secara keseluruhan.

Melalui penyajian data tersebut, pengguna dapat dengan mudah mengidentifikasi minggu dengan aktivitas trafik tertinggi serta mengevaluasi apakah penggunaan bandwidth masih berada dalam batas normal atau mendekati kapasitas maksimum. Bagian ini bersifat sebagai *overview* awal yang membantu pengguna memahami kondisi umum jaringan sebelum melanjutkan ke analisis yang lebih rinci pada bagian lain.

C.5 Bagian Internet Hourly

Berikut ini disajikan tampilan bagian Internet Hourly pada halaman Weekly Summary yang menyajikan visualisasi trafik jaringan internet berdasarkan interval waktu per jam.



Gambar 3.17. Halaman weekly summary - bagian internet hourly

Berdasarkan Gambar 3.17, bagian Internet Hourly merupakan bagian pada halaman *Weekly Summary* yang menyajikan visualisasi trafik jaringan internet berdasarkan interval waktu per jam. Pada bagian ini ditampilkan dua grafik utama, yaitu Internet Throughput Hourly Max dan Internet Occupancy Hourly Max, yang masing-masing menggambarkan nilai throughput maksimum dan tingkat pemanfaatan kapasitas bandwidth setiap jam dalam rentang beberapa hari. Melalui tampilan ini, pengguna dapat memperoleh gambaran yang lebih rinci mengenai pola trafik harian, termasuk jam-jam dengan aktivitas tertinggi (*peak hours*) serta fluktuasi trafik yang terjadi dalam satu hari.

Bagian ini juga dilengkapi dengan fitur interaktif untuk membantu analisis data, seperti tombol Hourly dan Daily yang memungkinkan pengguna berpindah antara tampilan per jam dan navigasi ke tampilan harian, serta tombol Expand untuk melihat grafik dalam ukuran layar penuh. Dengan adanya fitur *hover* pada grafik, pengguna dapat melihat nilai detail pada waktu tertentu, sehingga memudahkan identifikasi waktu terjadinya lonjakan trafik atau kondisi utilisasi yang mendekati batas kapasitas.

C.6 Bagian Internet Daily

Berikut ini disajikan tampilan bagian Internet Daily pada halaman Weekly Summary yang berfungsi sebagai ringkasan performa jaringan internet dalam skala harian.



Gambar 3.18. Halaman weekly summary - bagian internet daily

Berdasarkan Gambar 3.18, bagian Internet Daily berfungsi sebagai ringkasan performa jaringan internet dalam skala *harian* untuk periode yang lebih panjang (umumnya 1–3 bulan). Pada bagian ini ditampilkan dua grafik utama dalam satu tampilan berdampingan, yaitu Internet Throughput Daily Max yang menunjukkan nilai throughput maksimum per hari dalam satuan Mbps, serta Internet Occupancy Daily Max yang menampilkan persentase pemakaian kapasitas jaringan per hari. Kedua grafik ini membantu pengguna memahami pola penggunaan internet dari waktu ke waktu, membandingkan aktivitas pada hari kerja dan akhir pekan, serta mengidentifikasi tren jangka panjang yang berguna untuk *capacity planning* dan evaluasi performa jaringan.

Bagian ini juga dilengkapi dengan berbagai fitur interaktif untuk mempermudah analisis, seperti tombol Hourly dan Daily untuk berpindah tingkat granularitas data, serta tombol Expand yang membuka tampilan grafik layar penuh dengan informasi analisis tambahan. Pada mode expanded, pengguna dapat melihat informasi penting seperti *peak day* (hari dengan trafik tertinggi), rata-rata penggunaan pada hari kerja dan akhir pekan, serta peringatan apabila tingkat okupansi melewati ambang batas tertentu.

C.7 Bagian Utilization

Berikut ini disajikan tampilan bagian Utilization pada halaman Weekly Summary yang digunakan untuk memantau tingkat pemakaian kapasitas jaringan.



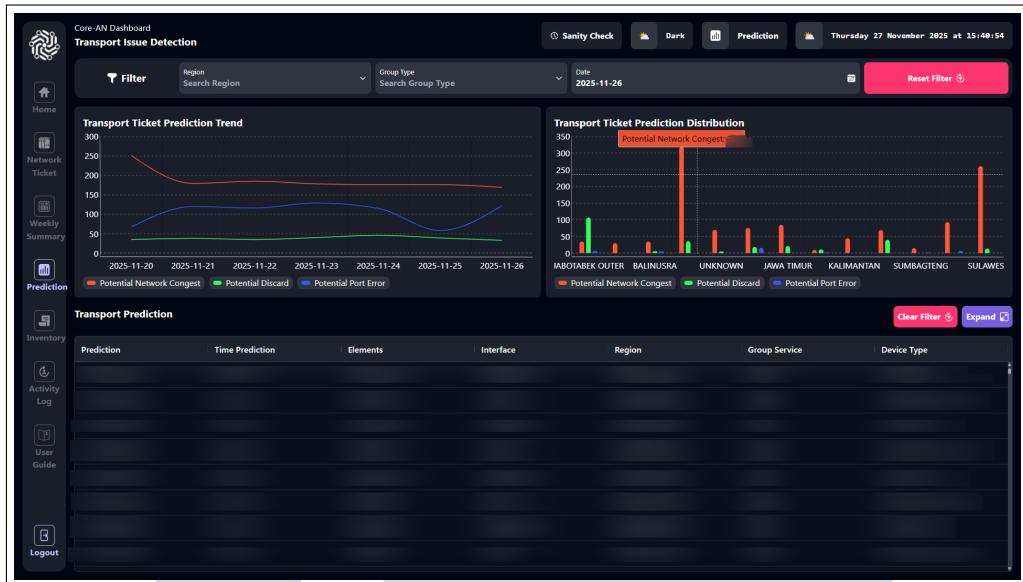
Gambar 3.19. Halaman weekly summary - bagian utilization

Berdasarkan Gambar 3.19, bagian Utilization merupakan bagian pada halaman *Weekly Summary* yang digunakan untuk memantau tingkat pemakaian kapasitas jaringan dengan menampilkan dua grafik batang horizontal, yaitu Utilization Transport dan Utilization Internet. Grafik ini menunjukkan persentase penggunaan bandwidth (0–100%) dibandingkan dengan total kapasitas yang tersedia untuk setiap *group service*. Pada sisi Transport, data mencakup infrastruktur seperti GINET, IPFABRIC, IPRAN, dan IPWAN, sedangkan pada sisi Internet mencakup layanan seperti IIX, GGC, Zenlayer, Tiktok, Akamai, dan lainnya. Setiap batang diberi warna berbeda untuk memudahkan pemahaman kondisi jaringan, yaitu hijau untuk kondisi normal (<50%), kuning untuk peringatan (50–80%), dan merah untuk kondisi kritis (>80%).

Bagian ini juga menyediakan berbagai fitur pendukung analisis, seperti garis ambang batas pada 50% dan 80% sebagai referensi visual, label persentase langsung pada batang grafik, serta *tooltip* interaktif yang menampilkan detail pemakaian. Selain tampilan utama, tersedia tombol Expand untuk membuka grafik dalam mode layar penuh yang dilengkapi fitur tambahan seperti pengurutan data, filter berdasarkan status, tabel rincian penggunaan, serta ringkasan statistik.

D Halaman Prediction

Berikut ini disajikan tampilan halaman Prediction yang berfungsi sebagai halaman analitik prediktif untuk mendukung pengelolaan jaringan secara proaktif.



Gambar 3.20. Halaman prediction

Berdasarkan Gambar 3.20, halaman Prediction merupakan halaman analitik prediktif yang dirancang untuk mendukung pengelolaan jaringan secara proaktif melalui pemanfaatan *machine learning*. Halaman ini berfungsi sebagai early warning system dengan menampilkan hasil prediksi potensi gangguan jaringan, seperti Network Congestion, Discard, dan Port Error, sebelum gangguan tersebut benar-benar terjadi. Kemampuan sistem untuk menganalisis data jaringan secara *real-time* dan memprediksi potensi masalah merupakan keunggulan penting dalam pengelolaan infrastruktur jaringan modern [10]. Prediksi dihasilkan berdasarkan analisis data historis, pola penggunaan jaringan, serta tren performa jaringan yang diproyeksikan dalam rentang waktu tertentu.

Informasi prediksi pada halaman ini disajikan dalam bentuk visualisasi grafik dan tabel yang memudahkan pengguna dalam memahami kondisi jaringan di masa mendatang. Melalui tampilan tersebut, pengguna dapat mengidentifikasi elemen jaringan, wilayah, maupun layanan yang memiliki tingkat risiko tinggi terhadap gangguan. Secara fungsional, Halaman Prediction mendukung proses pengambilan keputusan operasional dan perencanaan jaringan berbasis data, memungkinkan tim operasional untuk melakukan tindakan preventif sebelum gangguan berdampak pada kualitas layanan.

E Halaman Inventory

E.1 Gambaran Umum Halaman Inventory

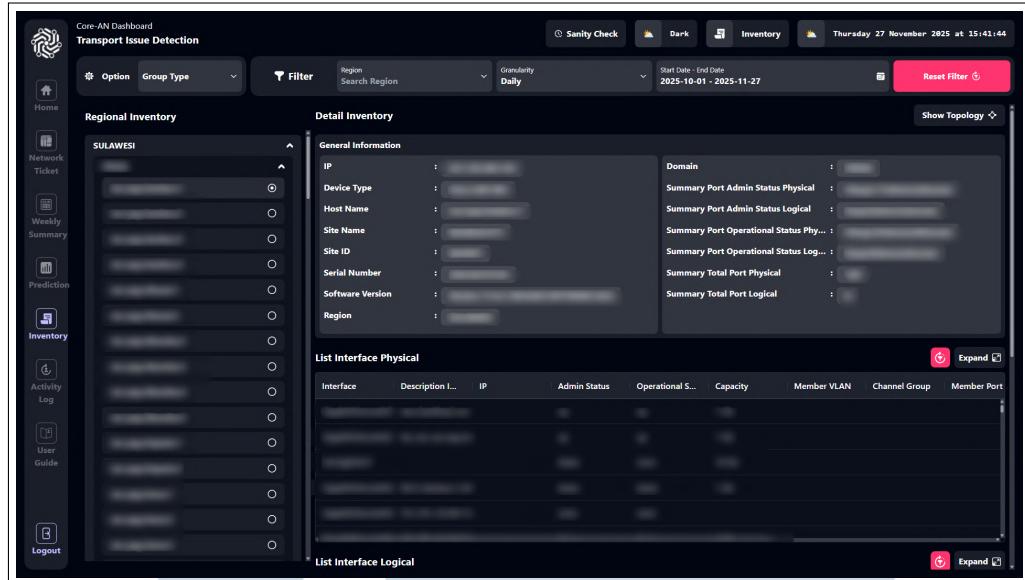
Halaman Inventory merupakan halaman manajemen *device* dan *network infrastructure* yang berfungsi sebagai pusat informasi perangkat jaringan pada sistem *Core AN Dashboard*. Halaman ini menyediakan tampilan komprehensif terhadap data perangkat jaringan, konfigurasi *interface*, riwayat gangguan, serta visualisasi topologi jaringan yang mendukung aktivitas pemantauan dan *troubleshooting* jaringan secara terintegrasi. Melalui halaman ini, pengguna dapat mengakses informasi detail perangkat secara terpusat tanpa perlu berpindah ke halaman lain.

Selain sebagai repositori data perangkat, halaman Inventory juga dirancang untuk mendukung dua pendekatan analisis, yaitu device-centric dan service-centric. Pendekatan device-centric digunakan untuk memantau kondisi dan konfigurasi perangkat secara individual, sedangkan pendekatan service-centric difokuskan pada analisis trafik berdasarkan kelompok layanan. Dengan adanya dua mode tampilan ini, halaman Inventory mampu memberikan fleksibilitas bagi pengguna dalam memahami kondisi jaringan dari berbagai perspektif operasional.

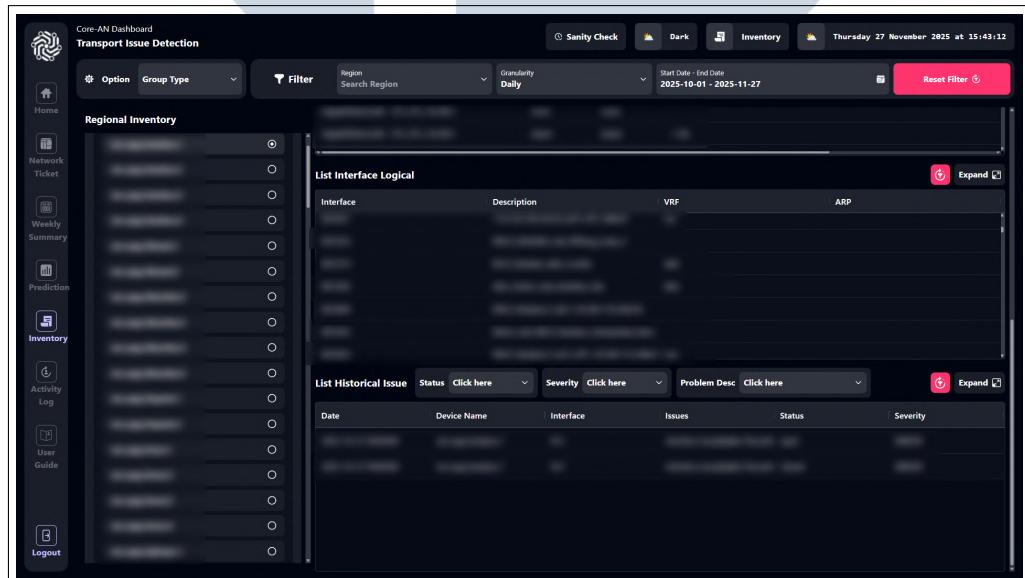
E.2 Komponen Halaman Inventory

Berikut ini disajikan tampilan komponen-komponen utama pada halaman Inventory yang saling terintegrasi untuk mendukung aktivitas pemantauan perangkat jaringan.





Gambar 3.21. Halaman inventory



Gambar 3.22. Halaman inventory (lanjutan)

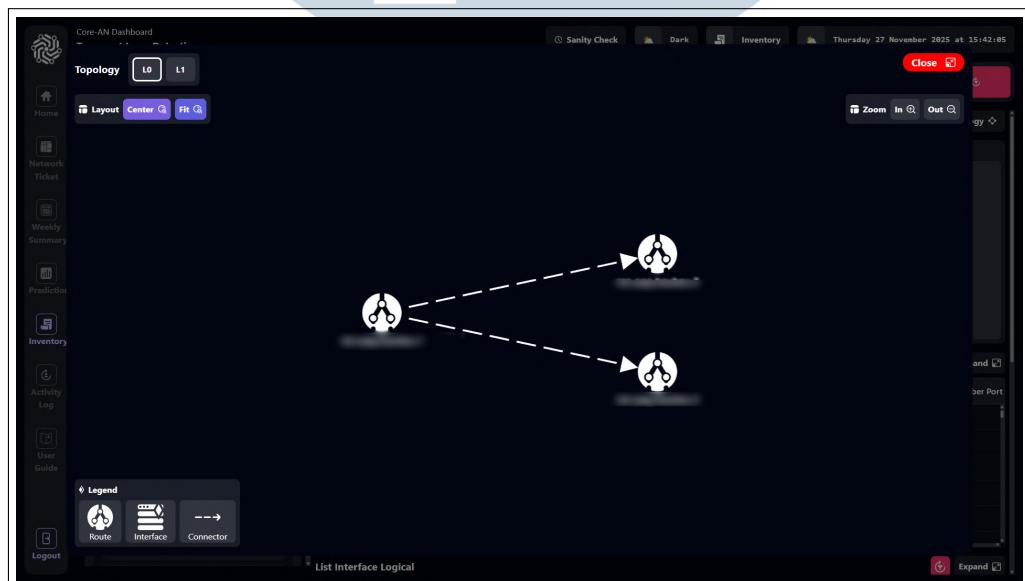
Berdasarkan Gambar 3.21 dan Gambar 3.22, halaman Inventory terdiri dari beberapa komponen utama yang saling terintegrasi. Pada mode Group Type View, halaman dibagi menjadi dua bagian utama, yaitu Regional Inventory pada sisi kiri dan Detail Inventory pada sisi kanan. Regional Inventory menampilkan struktur hierarki perangkat berdasarkan *Region*, *Group Type*, dan *Device Name* dalam bentuk *collapsible tree*, sehingga memudahkan navigasi dan pemilihan perangkat.

Setelah perangkat dipilih, Detail Inventory akan menampilkan informasi umum perangkat, daftar *interface physical*, daftar *interface logical*, serta riwayat gangguan perangkat.

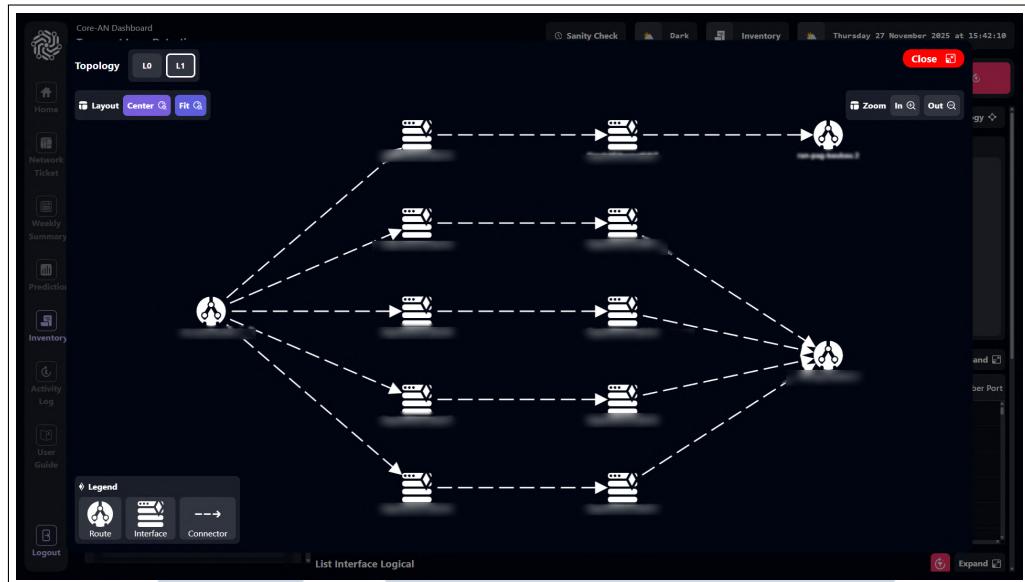
Selain mode tersebut, halaman Inventory juga menyediakan Group Service View sebagai mode alternatif. Mode ini menyajikan data jaringan berdasarkan kelompok layanan seperti *IIX*, *GGC*, dan layanan lainnya dalam bentuk tabel statistik trafik yang komprehensif. Seluruh komponen pada halaman Inventory dilengkapi dengan fitur pendukung seperti *filtering*, *sorting*, tampilan responsif, serta integrasi dengan *API* untuk memastikan data yang ditampilkan selalu bersifat *real-time* dan akurat.

E.2.1 Topology View (Modal)

Berikut ini disajikan tampilan Topology View yang merupakan fitur visualisasi jaringan untuk menampilkan hubungan antar perangkat dalam dua lapisan visualisasi.



Gambar 3.23. Bagian topology - L0

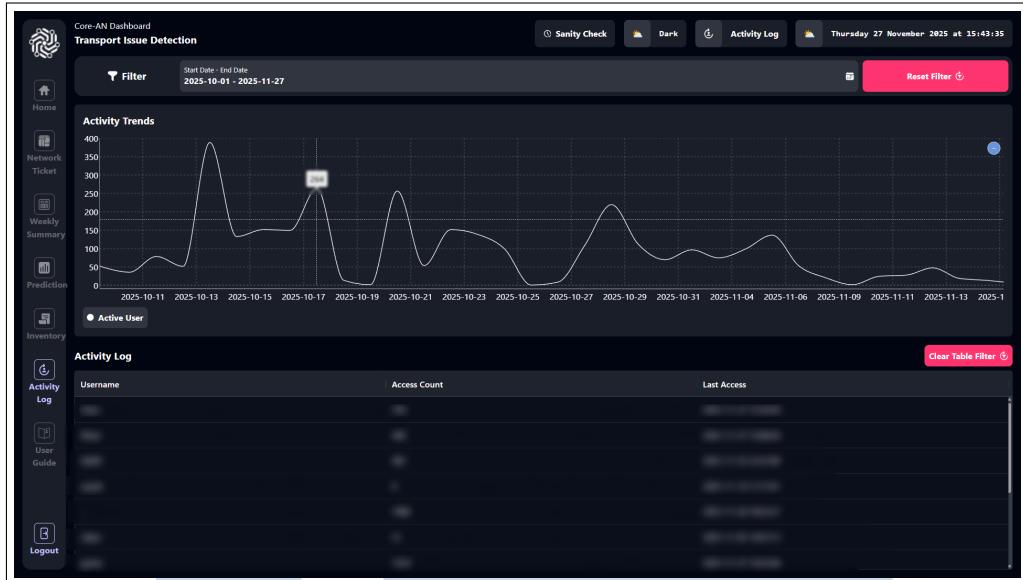


Gambar 3.24. Bagian topology - L1

Berdasarkan Gambar 3.23 dan Gambar 3.24, *Topology View* merupakan fitur visualisasi jaringan yang diakses melalui halaman *Inventory* dalam bentuk modal. Fitur ini digunakan untuk menampilkan hubungan antar perangkat jaringan dalam dua lapisan visualisasi, yaitu *Layer L0 (Physical)* dan *Layer L1 (Logical)*. Pada *Layer L0*, visualisasi menampilkan detail koneksi fisik antar perangkat beserta *interface* dan *connector*, sedangkan pada *Layer L1* ditampilkan hubungan logis antar perangkat dalam bentuk koneksi yang lebih sederhana. Visualisasi topologi ini bersifat interaktif, memungkinkan pengguna melakukan *zoom*, *pan*, serta pengaturan tata letak graf, sehingga membantu proses analisis konektivitas jaringan, dokumentasi infrastruktur, dan kegiatan *troubleshooting*.

F Halaman Activity Log

Berikut ini disajikan tampilan halaman *Activity Log* yang merupakan halaman monitoring untuk menampilkan riwayat aktivitas pengguna dalam sistem.



Gambar 3.25. Halaman activity log

Berdasarkan Gambar 3.25, halaman Activity Log merupakan halaman monitoring yang digunakan untuk menampilkan riwayat aktivitas pengguna dalam sistem *Transport Issue Detection*. Implementasi sistem pencatatan aktivitas pengguna memerlukan mekanisme autentikasi yang aman dan efisien, di mana JWT menyediakan solusi yang optimal untuk mengelola sesi pengguna tanpa membebani server dengan penyimpanan *state* [11]. Halaman ini memberikan gambaran mengenai siapa saja pengguna yang mengakses sistem, kapan aktivitas tersebut dilakukan, serta seberapa sering interaksi terjadi dalam periode waktu tertentu.

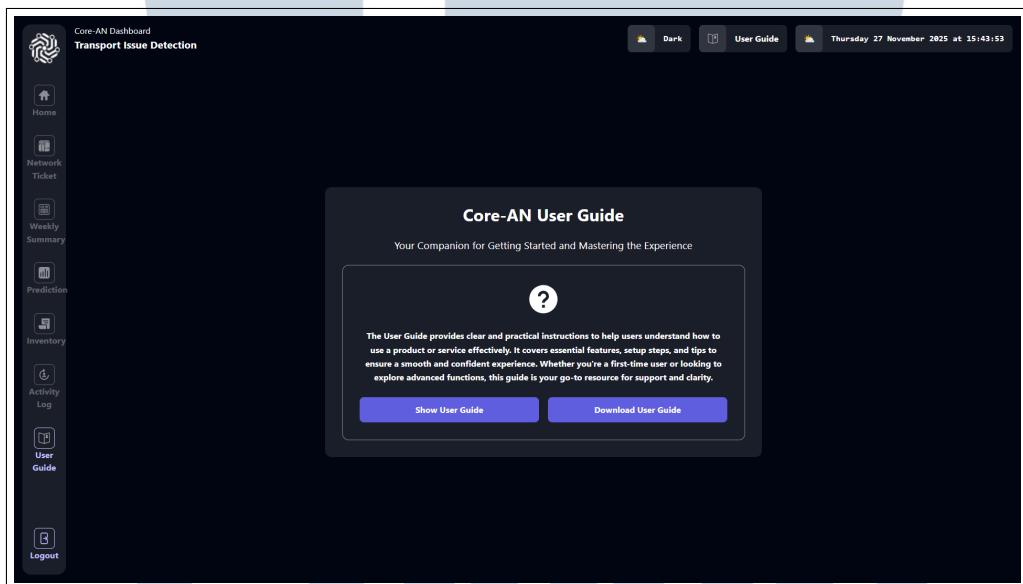
Secara struktur, halaman Activity Log terdiri dari tiga bagian utama, yaitu filter section, trend chart, dan tabel aktivitas. Pada bagian atas, filter date range disediakan untuk memungkinkan pengguna memilih rentang waktu tertentu. Bagian tengah menampilkan trend chart berupa grafik garis yang menunjukkan jumlah akses harian pengguna dalam periode yang dipilih, dilengkapi dengan fitur *pan*, *zoom*, *tooltip*, serta *legend toggle*. Sementara itu, pada bagian bawah terdapat tabel aktivitas yang menampilkan detail pengguna seperti *username*, jumlah akses, dan waktu akses terakhir, yang dilengkapi dengan fitur *filtering* dan *sorting*.

Sistem pencatatan aktivitas pada halaman ini bekerja secara otomatis dan manual. Pencatatan otomatis dilakukan setiap kali pengguna melakukan navigasi ke halaman tertentu, di mana sistem akan mencatat nama pengguna dan halaman yang diakses. Selain itu, pencatatan manual juga diterapkan pada aksi-aksi

spesifik pengguna seperti membuka detail data, mengunggah atau mengunduh file, serta menerapkan filter. Seluruh data aktivitas tersebut disimpan ke dalam basis data dan ditampilkan kembali pada halaman Activity Log secara *real-time*. Dengan mekanisme ini, halaman Activity Log berperan penting dalam mendukung kebutuhan *audit trail*, pemantauan perilaku pengguna, analisis keamanan, serta penyusunan laporan penggunaan sistem.

G User Guide

Berikut ini disajikan tampilan halaman User Guide yang merupakan halaman dokumentasi untuk menyediakan panduan penggunaan sistem.



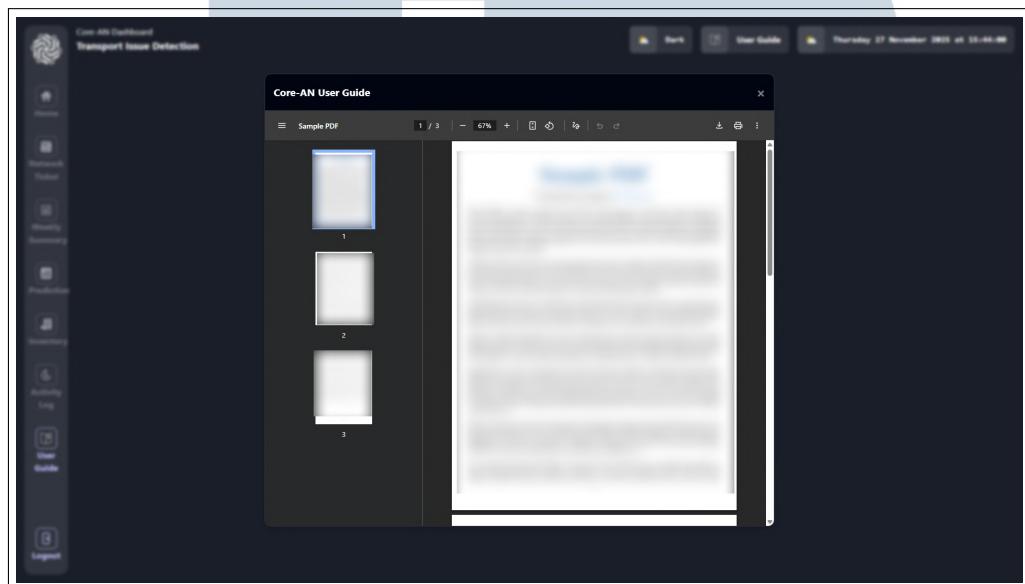
Gambar 3.26. Halaman user guide

Berdasarkan Gambar 3.26, Halaman User Guide merupakan halaman dokumentasi yang menyediakan panduan penggunaan sistem *Transport Issue Detection* dalam format dokumen *PDF*. Halaman ini dirancang sebagai sumber referensi utama bagi pengguna untuk memahami fitur-fitur aplikasi, tahapan penggunaan, serta tips operasional sistem secara menyeluruh. Dengan adanya User Guide, pengguna dapat mempelajari cara kerja sistem secara mandiri tanpa harus bergantung pada bantuan langsung dari administrator atau tim pengembang.

Pada tampilan awal, halaman User Guide menampilkan judul dan deskripsi singkat yang menjelaskan tujuan dari panduan tersebut. Konten utama ditampilkan dalam sebuah kartu informasi yang berfungsi untuk memusatkan perhatian

pengguna pada dokumentasi yang tersedia, dilengkapi ikon bantuan sebagai penanda halaman dokumentasi. Selain itu, tersedia dua tombol utama, yaitu tombol untuk menampilkan panduan secara langsung di dalam sistem dan tombol untuk mengunduh dokumen *PDF*. Desain halaman ini juga menyesuaikan tampilan *light mode* dan *dark mode* guna menjaga kenyamanan visual pengguna.

Berikut ini disajikan tampilan PDF Viewer Modal ketika pengguna memilih opsi untuk menampilkan panduan.



Gambar 3.27. User guide - PDF

Berdasarkan Gambar 3.27, ketika pengguna memilih opsi untuk menampilkan panduan dengan menekan *button* Show User Guide, sistem akan menampilkan PDF Viewer Modal. Tampilan ini memungkinkan pengguna membaca dokumen User Guide secara langsung tanpa meninggalkan halaman aplikasi. Dokumen ditampilkan menggunakan *PDF viewer* bawaan peramban yang mendukung navigasi halaman, pencarian teks, pengaturan *zoom*, serta opsi unduh dan cetak. Modal ini dilengkapi dengan tombol penutup untuk kembali ke halaman utama User Guide.

3.4 Kendala dan Solusi yang Ditemukan

Selama proses pengembangan proyek *Transport Failure Optimization Use Case Dashboard*, terdapat berbagai kendala teknis maupun non-teknis yang muncul akibat kompleksitas sistem, banyaknya halaman dashboard, serta

penggunaan beragam teknologi pendukung. Kendala-kendala tersebut menjadi tantangan dalam memastikan sistem dapat berjalan secara optimal, stabil, dan mudah digunakan. Oleh karena itu, diperlukan solusi yang tepat agar proses pengembangan tetap berjalan sesuai dengan kebutuhan sistem dan pengguna.

3.4.1 Kendala

Kendala utama yang dihadapi adalah tingginya kompleksitas integrasi *multi-library* pada sisi *frontend*. Dashboard memanfaatkan beberapa *library* visualisasi seperti AmCharts 5, AG-Grid, dan Cytoscape.js yang masing-masing memiliki *API* dan *lifecycle* berbeda. Perbedaan paradigma ini berpotensi menimbulkan konflik dependensi serta risiko *memory leak* apabila tidak dilakukan pengelolaan *instance* dengan baik.

Selain itu, pengelolaan *state* pada banyak halaman menjadi tantangan tersendiri. Terdapat beberapa halaman yang saling berbagi data seperti filter tanggal, *region*, dan preferensi pengguna, sehingga berpotensi menyebabkan duplikasi pemanggilan *API* dan kesulitan sinkronisasi antar komponen. Kondisi ini berdampak pada efisiensi aplikasi serta meningkatkan kompleksitas pemeliharaan kode.

Kendala lain yang signifikan berkaitan dengan performa sistem dan stabilitas sesi pengguna. Pengolahan *dataset* berukuran besar menyebabkan penurunan performa pada tabel dan visualisasi grafik, terutama pada perangkat dengan spesifikasi rendah. Selain itu, masalah *session timeout*, *error API*, serta perbedaan perilaku antar browser juga menjadi hambatan dalam menjaga pengalaman pengguna yang konsisten.

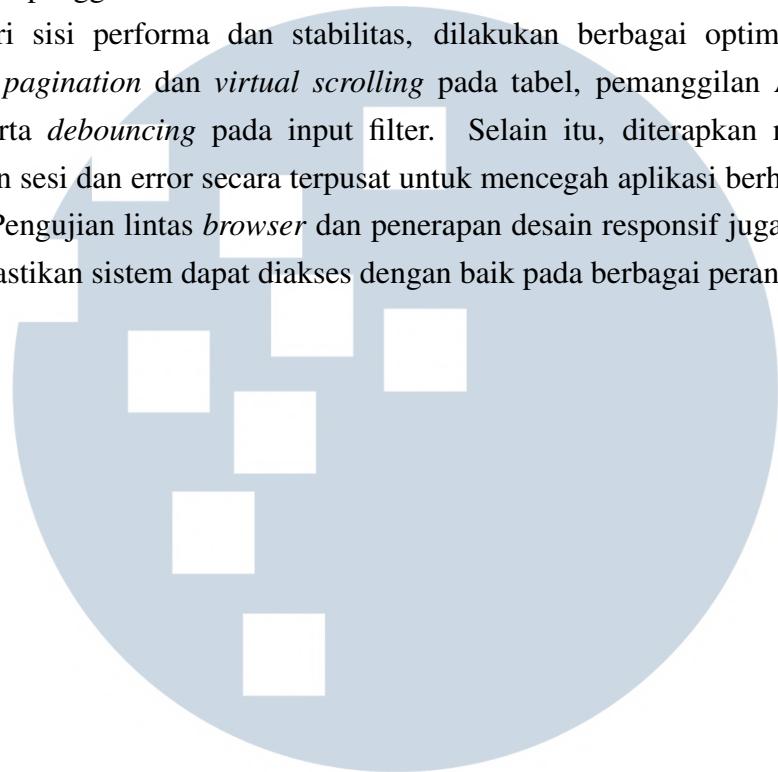
3.4.2 Solusi

Untuk mengatasi kompleksitas integrasi *library*, diterapkan pendekatan *component wrapping* dengan memanfaatkan *lifecycle* SolidJS. Setiap *library* dibungkus dalam komponen terpisah dan dilengkapi proses *cleanup* menggunakan `onCleanup()` agar penggunaan memori tetap terkontrol. Selain itu, *lazy loading* diterapkan agar *library* berat hanya dimuat saat diperlukan.

Permasalahan manajemen *state* diselesaikan dengan menerapkan *centralized state management* menggunakan *store* berbasis SolidJS. Pendekatan ini memungkinkan sinkronisasi data antar halaman secara konsisten, mengurangi duplikasi *API call*, serta menyederhanakan pengelolaan filter global. Penyimpanan

preferensi pengguna juga dilakukan melalui *localStorage* untuk meningkatkan kenyamanan penggunaan.

Dari sisi performa dan stabilitas, dilakukan berbagai optimasi seperti penerapan *pagination* dan *virtual scrolling* pada tabel, pemanggilan API secara paralel, serta *debouncing* pada input filter. Selain itu, diterapkan mekanisme penanganan sesi dan error secara terpusat untuk mencegah aplikasi berhenti secara tiba-tiba. Pengujian lintas *browser* dan penerapan desain responsif juga dilakukan guna memastikan sistem dapat diakses dengan baik pada berbagai perangkat.



UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA