

## BAB 3

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Koordinasi

Selama pelaksanaan kegiatan magang, posisi yang ditempati mencakup dua peran, yaitu sebagai Product Owner dan Front End Web Developer di bawah naungan PT. Moonlay Technologies. Kegiatan magang dilakukan pada divisi Teknologi (*Technology Division*) yang berada langsung di bawah pengawasan Head of Technology.

Sebagai Product Owner, koordinasi dilakukan langsung dengan Product Owner Lead, yang memberikan arahan terkait pengelolaan backlog, pengumpulan kebutuhan pengguna, penyusunan prioritas pengembangan fitur, serta komunikasi antara tim pengembang dan pemangku kepentingan lainnya. Dalam pelaksanaannya, terdapat interaksi dengan tim-tim terkait seperti Project Manager dan tim pengembang untuk memastikan spesifikasi produk dapat dikembangkan secara tepat.

Sementara itu, dalam peran sebagai Front End Web Developer, posisi berada di bawah struktur Software Development Manager, dengan koordinasi erat bersama Backend Developer dalam proses pengembangan aplikasi *web*. Keterampilan dan alat seperti TypeScript dan React JS digunakan untuk menghasilkan pengalaman pengguna yang optimal. Tanggung jawab mencakup implementasi desain antarmuka pengguna ke dalam kode program, integrasi API dari tim backend, serta memastikan antarmuka berjalan dengan baik di berbagai perangkat dan browser.

Kedua peran tersebut memberikan kesempatan untuk memahami alur kerja produk dari sisi manajerial maupun teknis. Partisipasi aktif dilakukan dalam rapat harian (*daily sprint*), *sprint planning*, serta review dan *retrospective* dalam siklus pengembangan agile yang diterapkan di perusahaan. Komunikasi dan kolaborasi lintas tim menjadi elemen penting dalam menjaga keberhasilan proses pengembangan produk yang sedang dikerjakan.

### 3.2 Tugas yang Dilakukan

Selama pelaksanaan magang, berbagai tugas telah diselesaikan secara bertahap sesuai dengan kebutuhan proyek Visee. Tahapan awal dimulai dengan memahami proses bisnis dan alur kerja proyek, termasuk teknologi yang digunakan dalam pengembangannya. Setelah itu, dilakukan penyusunan dan pembahasan *product backlog* serta mempelajari metode kerja *sprint* menggunakan Jira sebagai alat bantu manajemen proyek. Untuk mendukung kebutuhan basis data, dibuat ERD dan tabel untuk basis data PostgreSQL menggunakan *tool* DBeaver.

Fokus kemudian beralih pada pengembangan halaman *Store Management* untuk admin, dimulai dari pembuatan tampilan hingga pengintegrasian fitur CRUD agar data dapat dikelola langsung melalui antarmuka pengguna. Fitur tambahan seperti *filter* pada halaman *Store Management* dan *store selection* pada *sidebar* juga dikembangkan untuk mempermudah navigasi dan pencarian data.

Tahap selanjutnya mencakup penambahan *field input store name* pada halaman *zone* dan *workhour*, yang masing-masing juga diintegrasikan ke dalam sistem. Setelah bagian *Store Management* selesai, pekerjaan dilanjutkan dengan mengintegrasikan halaman *live visitor*, *live demographic*, *historical visitor*, dan *historical demographic* sebagai bagian dari fitur analitik pada dashboard Visee. Terakhir, dilakukan pembuatan tampilan UI dan integrasi untuk halaman perbandingan toko (*store comparison*).

Rincian tugas-tugas yang telah dikerjakan selama masa magang tersebut dapat dilihat pada Tabel 3.1 yang berisi daftar pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang.

U M W N  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

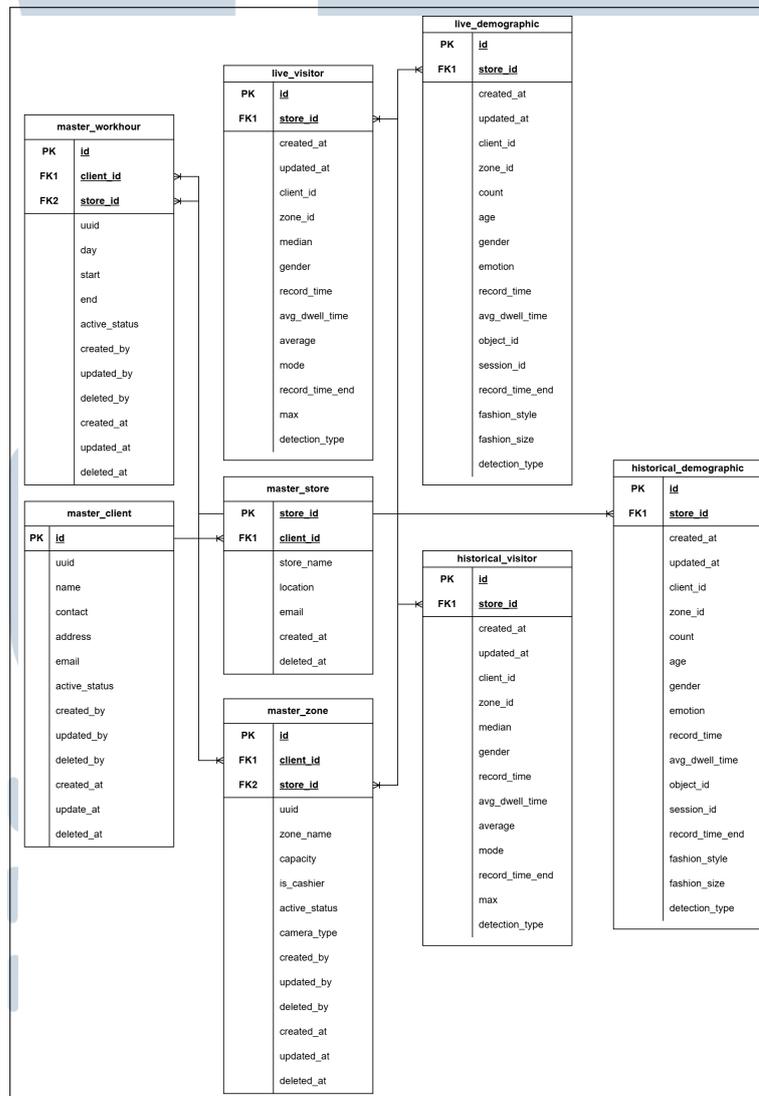
Minggu Ke -	Pekerjaan yang dilakukan
1	Memahami proses bisnis dan alur kerja proyek Visee, termasuk teknologi yang digunakan
2	Menyusun dan membahas <i>product backlog</i> dan mempelajari <i>sprint</i> menggunakan Jira
3	Membuat ERD dan tabel untuk basis data PostgreSQL Proyek Visee menggunakan <i>tool</i> DBeaver
4	Membuat tampilan halaman <i>Store Management</i> untuk admin
5	Mengintegrasikan fitur CRUD yang ada di halaman <i>store management</i>
6	Membuat fitur <i>filter</i> pada <i>Store Management</i> dan fitur <i>store selection</i> pada navbar
7	Menambahkan <i>field input store name</i> pada halaman <i>zone</i> dan mengintegrasikannya
8	Menambahkan <i>field input store name</i> pada halaman <i>workhour</i> dan mengintegrasikannya
9	Mengintegrasikan halaman <i>live visitor</i>
10	Mengintegrasikan halaman <i>live demographic</i>
11	Mengintegrasikan halaman <i>historical visitor</i>
12	Mengintegrasikan halaman <i>historical demographic</i>
13	Membuat tampilan UI dan mengintegrasikan halaman <i>store comparison</i>

### 3.3 Uraian Pelaksanaan Magang

Pelaksanaan magang di PT. Moonlay Technologies diawali dengan proses orientasi terhadap alur kerja internal perusahaan, termasuk pengenalan metode pengembangan perangkat lunak menggunakan pendekatan Agile. Proyek utama yang dikerjakan adalah Visee, sebuah sistem kamera pengawas yang dilengkapi fitur analisis demografis secara *real-time* dan historis. Tahapan awal difokuskan pada pemahaman sistem serta perencanaan kebutuhan pengembangan secara bertahap menggunakan manajemen proyek berbasis *sprint*.

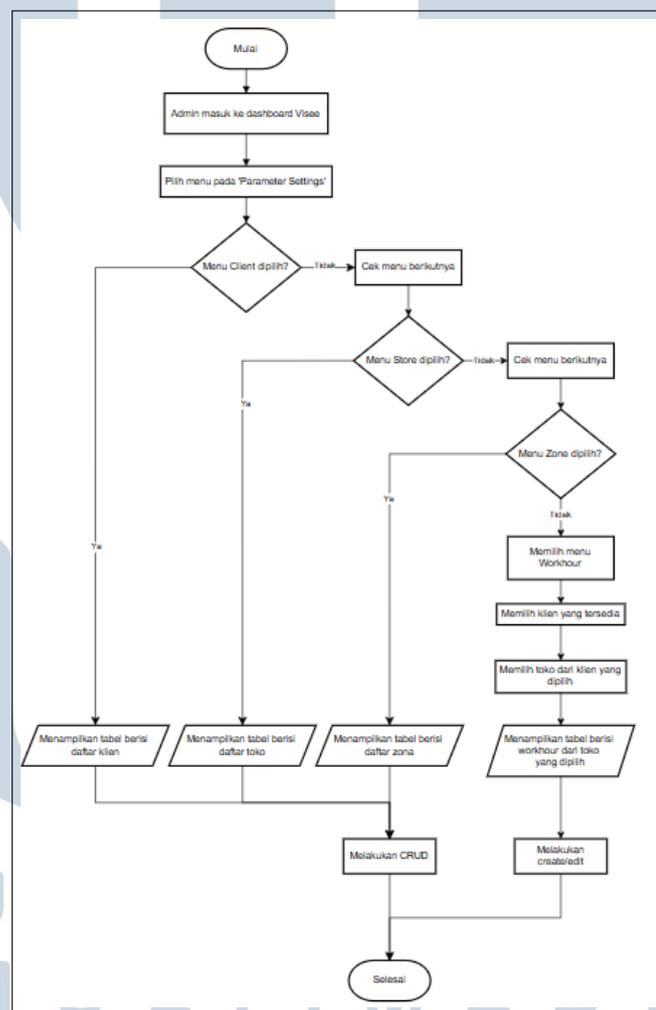
### 3.3.1 Perancangan

Tahapan awal difokuskan pada pemahaman proses bisnis dan alur kerja proyek Visee, termasuk teknologi yang digunakan dalam pengembangannya. Setelah itu, dilakukan penyusunan *product backlog* serta pembelajaran metode kerja *sprint* menggunakan Jira sebagai alat bantu manajemen proyek. Setelah *backlog* tersusun, dilakukan perancangan struktur basis data PostgreSQL untuk menyimpan informasi pengunjung, toko, zona, dan statistik. Perancangan ini meliputi pembuatan *Entity Relationship Diagram* serta tabel awal menggunakan *tool* DBeaver. Untuk detail struktur basis data, dapat dilihat pada Gambar 3.1, yang menyajikan *Entity Relationship Diagram* sistem Visee.



Gambar 3.1. *Entity Relationship Diagram* Visee

Selain perancangan struktur basis data, alur kerja admin untuk mengelola data-data *master* tersebut juga dirancang pada tahap ini. Perancangan alur kerja ini divisualisasikan menggunakan *flowchart* untuk memberikan gambaran yang jelas mengenai langkah-langkah yang perlu dilakukan pengguna, mulai dari memilih menu pada *sidebar* hingga melakukan operasi *Create, Read, Update, Delete* (CRUD) untuk data *Client, Store, dan Zone*, serta operasi *create/edit* untuk *Workhour*. *Flowchart* ini menjadi landasan utama dalam pengembangan antarmuka pengguna dan logika navigasi pada fitur pengelolaan data. *Flowchart* admin secara lengkap dapat dilihat pada Gambar 3.2.



Gambar 3.2. *Flowchart* Proses Pengelolaan Data *Master* oleh Admin

### 3.3.2 Pengembangan Halaman *Store Management*

Pengembangan selanjutnya difokuskan pada halaman *Store Management* yang dirancang untuk admin. Tahap awal meliputi perancangan dan implementasi

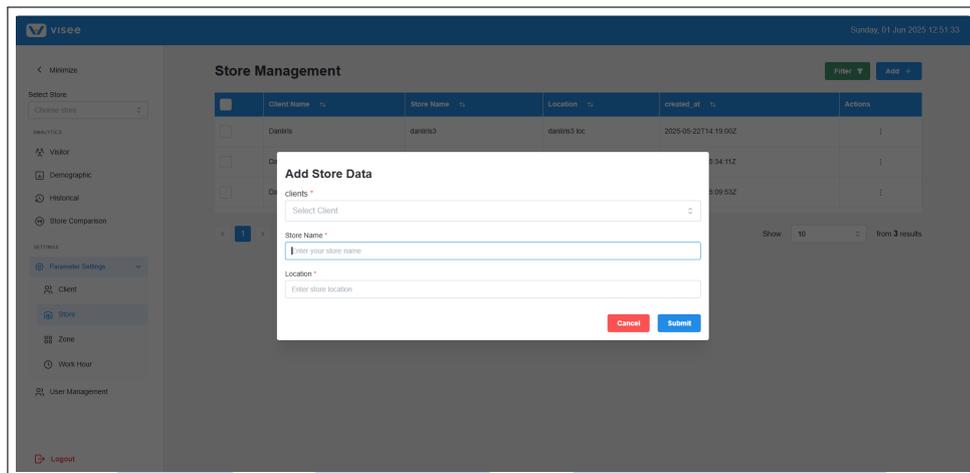
antarmuka pengguna (UI) yang intuitif guna memudahkan pengelolaan data toko. Halaman utama *Store Management* menampilkan daftar toko dalam format tabel yang informatif, menyajikan kolom-kolom data seperti *Client Name*, *Store Name*, *Location*, dan *created\_at*. Setiap baris data toko dilengkapi dengan menu *Actions* yang memungkinkan admin melakukan berbagai operasi pengelolaan. Untuk menangani volume data yang besar, halaman ini juga dilengkapi dengan fungsionalitas paginasi yang memungkinkan admin mengatur jumlah data yang ditampilkan per halaman dan bernavigasi antar halaman. Tampilan utama dari halaman *Store Management* dapat dilihat pada Gambar 3.3.

<input type="checkbox"/>	Client Name <sup>1</sup>	Store Name <sup>1</sup>	Location <sup>1</sup>	created_at <sup>1</sup>	Actions
<input type="checkbox"/>	Danirs	danirs3	danirs3 loc	2025-05-22T14:19:00Z	<a href="#">⋮</a>
<input type="checkbox"/>	Danirs	danirs1	danirs1 loc	2025-03-24T10:34:11Z	<a href="#">⋮</a>
<input type="checkbox"/>	Danirs	danirs2	danirs2 loc	2025-05-14T10:09:53Z	<a href="#">⋮</a>

Gambar 3.3. Tampilan utama halaman Store Management

Untuk mengelola data toko secara efektif, fitur-fitur inti CRUD (*Create*, *Read*, *Update*, *Delete*) diimplementasikan secara komprehensif pada halaman ini:

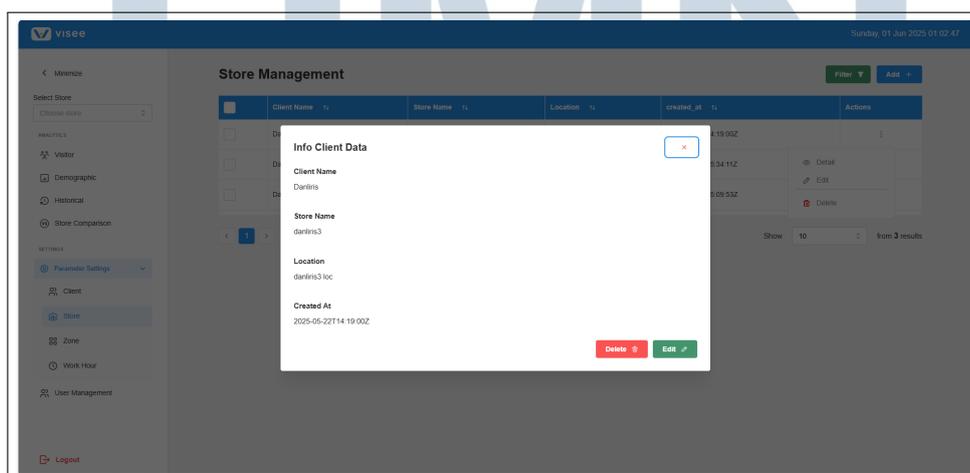
1. *Create* (Tambah Toko Baru): Admin dapat menambahkan data toko baru dengan mengklik tombol *Add* yang terletak di bagian atas tabel. Setelah diklik, akan muncul sebuah jendela modal (*pop-up*) berjudul *Add Store Data*, seperti yang ditunjukkan pada Gambar 3.4.



Gambar 3.4. Modal *add Store Data* untuk menambah toko baru

Di dalam modal ini, admin diminta untuk mengisi beberapa informasi, yaitu memilih *clients* dari daftar klien yang sudah terdaftar (*dropdown*), memasukkan *Store Name*, dan *Location*. Setelah semua data terisi, admin dapat menyimpannya ke dalam sistem melalui tombol *Submit*.

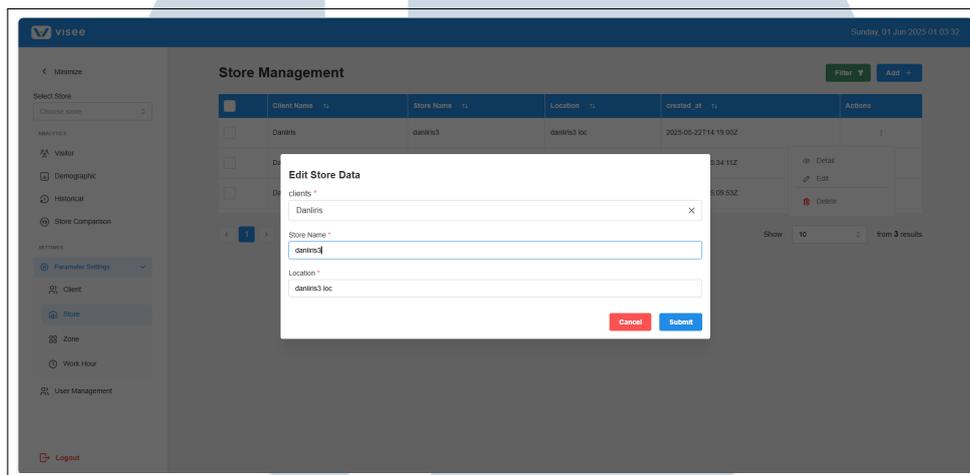
2. *Read* (Lihat Detail Toko): Untuk melihat detail informasi dari sebuah toko, admin dapat menggunakan menu *Actions* yang tersedia pada setiap baris data toko di tabel. Dengan memilih opsi *Detail* dari menu tersebut, sebuah modal *Info Client Data* akan ditampilkan (lihat Gambar 3.5). Modal ini menyajikan informasi lengkap mengenai toko yang dipilih, meliputi *Client Name*, *Store Name*, *Location*, dan *Created At*



Gambar 3.5. Modal *Info Client Data* yang menampilkan detail toko

3. *Update* (Ubah Data Toko): Proses pembaruan data toko dapat diakses melalui dua cara. Pertama, dari modal *Info Client Data* (setelah memilih *Detail*),

admin dapat mengklik tombol *Edit*. Kedua, opsi *Edit* juga dapat diakses secara langsung dari menu *Actions* pada tabel daftar toko. Ketika opsi *Edit* dipilih, akan muncul modal *Edit Store Data*, sebagaimana terlihat pada Gambar 3.6. Di sini, admin dapat mengubah data *clients*, *Store Name*, dan *Location* dari toko yang bersangkutan. Perubahan disimpan dengan mengklik tombol *Submit*.



Gambar 3.6. Modal *Edit Store Data* untuk mengubah informasi toko

4. *Delete* (Hapus Data Toko): Penghapusan data toko juga dapat dilakukan melalui modal *Info Client Data*, di mana terdapat tombol *Delete*. Selain itu, opsi *Delete* juga tersedia langsung pada menu *Actions* di setiap baris tabel. Setelah admin memilih untuk menghapus dan melakukan konfirmasi, data toko tersebut akan dihapus secara permanen dari sistem.

Selain fungsionalitas CRUD ini tersebut, pada halaman *Store Management* juga dikembangkan tombol *Filter*. Fitur ini bertujuan untuk memberikan kemudahan kepada admin dalam melakukan pencarian atau penyaringan daftar toko berdasarkan kriteria tertentu, sehingga pengelolaan data menjadi lebih efisien. Lebih lanjut, untuk meningkatkan pengalaman pengguna dalam navigasi antar toko pada keseluruhan dasbor Visee, diimplementasikan pula fitur *store selection* pada bagian *sidebar*.

### 3.3.3 Penambahan *Field* dan Integrasi Halaman *Zone & Workhour*

Setelah pengembangan halaman *Store Management*, tahap selanjutnya adalah melakukan penyesuaian pada halaman pengelolaan *Zone* dan *Workhour*.

Penyesuaian utama pada halaman *Zone* adalah penambahan field *Store Name* untuk mengasosiasikan setiap zona dengan toko spesifik. Hal ini penting untuk memastikan bahwa konfigurasi zona dan jam kerja dapat dikelola secara terperinci untuk masing-masing toko yang terdaftar dalam sistem Visee.

Pada halaman *Zone Data*, tabel utama kini menampilkan kolom *Store Name* di samping *Client Name*, *Zone*, *Camera Type*, *Capacity*, dan *Cashier*. Tampilan halaman *Zone Data* yang telah diperbarui dapat dilihat pada Gambar 3.7.

	Client Name	Store Name	Zone	Camera Type	Capacity	Cashier	Actions
<input type="checkbox"/>	Daniris	daniris3	daniris1demo	DEMOGRAPHIC CAMERA	1	NO	⋮
<input type="checkbox"/>	Daniris	daniris2	daniris1demo	DEMOGRAPHIC CAMERA	1	NO	⋮
<input type="checkbox"/>	Daniris	daniris1	daniris1demo	DEMOGRAPHIC CAMERA	1	NO	⋮
<input type="checkbox"/>	Daniris	daniris1	daniris11demo	DEMOGRAPHIC CAMERA	1	NO	⋮
<input type="checkbox"/>	Daniris	daniris1	daniris12	HEATMAP CAMERA	1	NO	⋮
<input type="checkbox"/>	Daniris	daniris3	daniris31	HEATMAP CAMERA	1	NO	⋮
<input type="checkbox"/>	Daniris	daniris2	daniris21	HEATMAP CAMERA	1	NO	⋮
<input type="checkbox"/>	Daniris	daniris1	daniris11	HEATMAP CAMERA	1	NO	⋮

Gambar 3.7. Tampilan utama halaman *Zone Data* dengan penambahan kolom *Store Name*

Proses pengelolaan data zona juga mengikuti fungsionalitas CRUD yang serupa dengan halaman *Store Management*:

1. *Create* (Tambah Zona Baru): Untuk menambahkan data zona baru, admin dapat mengklik tombol *Add*. Sebuah modal *Add Zone Data* akan muncul, seperti yang terlihat pada Gambar 3.8.

**Add Zone Data**

Client Name \*

Store Name \*

Camera Type \*

Zone Name \*

Capacity \*

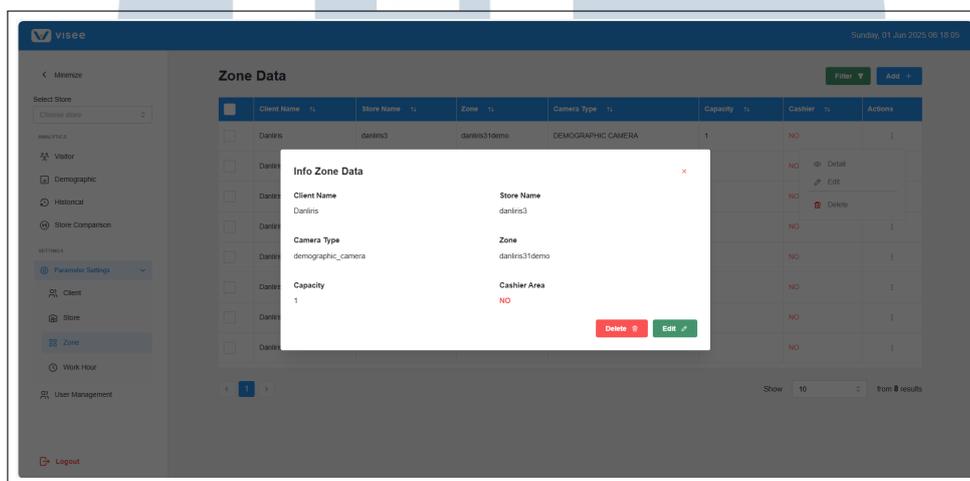
Cashier  No  Yes

Cancel Submit

Gambar 3.8. Modal *Add Zone Data* dengan field *Store Name*

Pada modal ini, admin harus memilih *Client Name* dan *Store Name* dari daftar *dropdown* yang tersedia. Selanjutnya, admin perlu menginput *Camera Type*, *Zone Name*, *Capacity*, dan menentukan status *Cashier* (*No/Yes*). Data disimpan dengan tombol *Submit*.

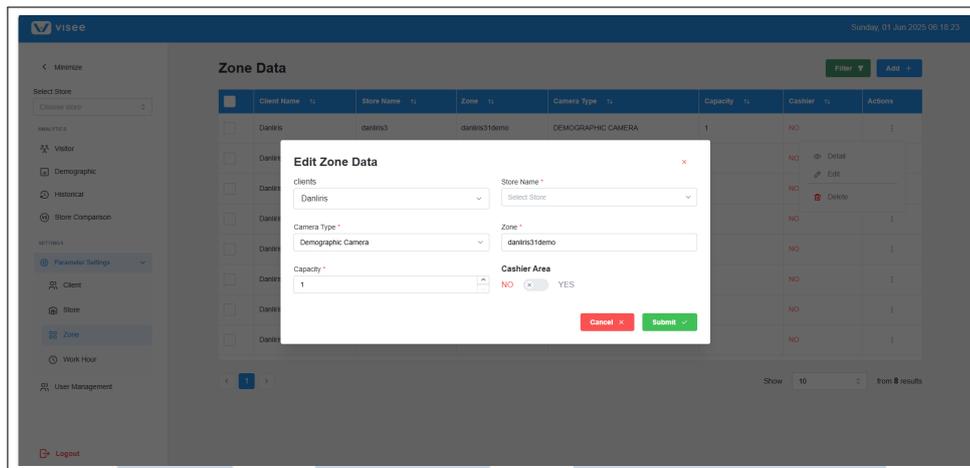
2. *Read* (Lihat Detail Zona): Detail informasi zona dapat diakses melalui menu *Actions* pada setiap baris data di tabel, kemudian memilih opsi *Detail*. Modal *Info Zone Data* akan menampilkan semua informasi terkait zona tersebut, termasuk *Client Name* dan *Store Name* yang terasosiasi (lihat Gambar 3.9).



Gambar 3.9. Modal *Info Zone Data* yang menampilkan detail zona termasuk *Store Name*

3. *Update* (Ubah Data Zona): Perubahan data zona dilakukan melalui modal *Edit Zone Data* yang dapat diakses dari tombol *Edit* di modal *Info Zone Data* atau langsung dari menu *Actions* di tabel. Seperti yang ditunjukkan pada Gambar 3.10, admin dapat mengubah *Client Name*, *Store Name*, *Camera Type*, *Zone Name*, *Capacity*, dan status *Cashier*. Perubahan disimpan melalui tombol *Submit*.

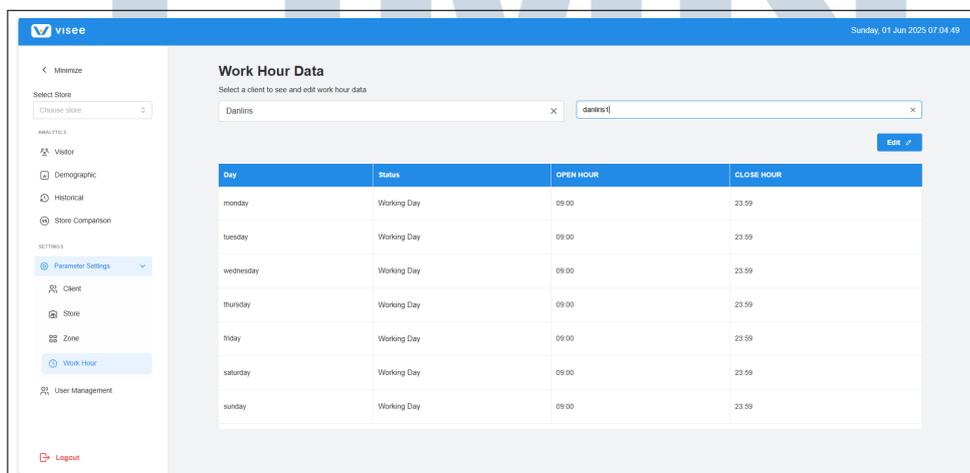
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.10. Modal *Edit Zone Data* dengan *field Store Name* yang dapat diubah

4. *Delete* (Hapus Data Zona): Opsi penghapusan zona juga tersedia melalui tombol *Delete* pada modal *Info Zone Data* atau langsung dari menu *Actions*.

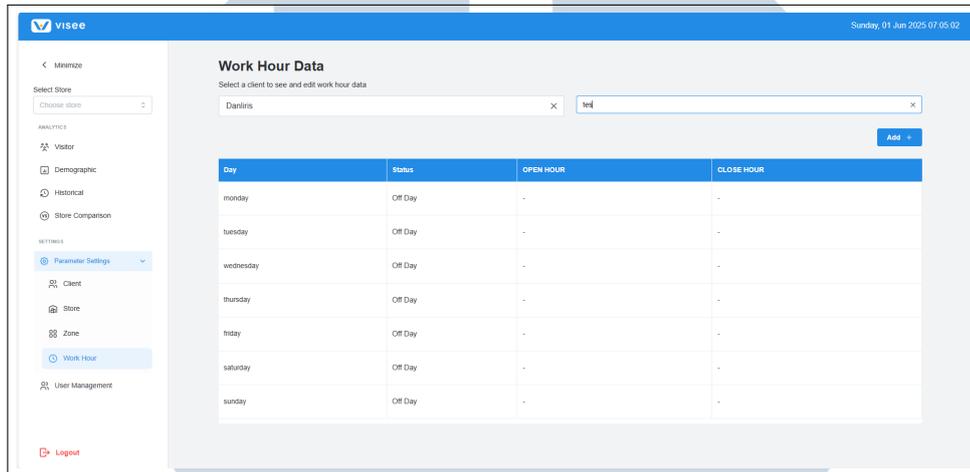
Integrasi serupa dengan penambahan keterkaitan pada data toko juga dilakukan pada halaman *Workhour*. Untuk mengakses atau mengubah data jam kerja, admin kini harus terlebih dahulu memilih *Client Name* dan kemudian *Store Name* yang spesifik melalui dua buah dropdown yang saling bergantung. Setelah kedua pilihan ini ditentukan, sistem akan menampilkan data jam kerja untuk toko yang dipilih, yang terdiri dari daftar hari (Senin hingga Minggu) beserta statusnya (*Working Day* atau *Off Day*), jam buka (*OPEN HOUR*), dan jam tutup (*CLOSE HOUR*). Tampilan halaman *Workhour* setelah pemilihan klien dan toko dapat dilihat pada Gambar 3.11.



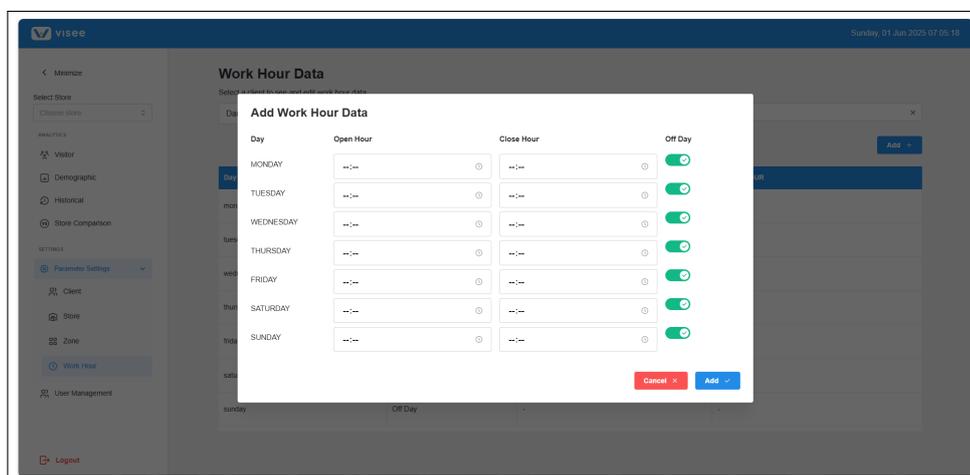
Gambar 3.11. Tampilan halaman *Workhour* setelah memilih *Client* dan *Store*

Pengelolaan data jam kerja pada halaman ini meliputi:

1. *Create/Add* (Tambah Data Jam Kerja): Jika sebuah toko yang dipilih belum memiliki konfigurasi jam kerja, halaman akan menampilkan status *Off Day* untuk semua hari tanpa jam operasional, seperti yang ditunjukkan pada Gambar 3.12.

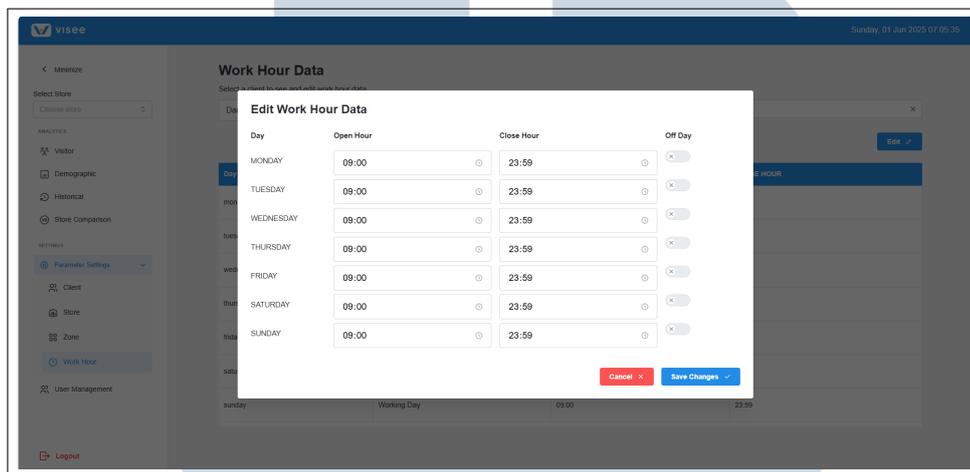


Gambar 3.12. Tampilan halaman *Workhour* untuk toko tanpa konfigurasi jam kerja Admin dapat mengklik tombol *Add* untuk membuka modal *Add Work Hour Data* (lihat Gambar 3.13).



Gambar 3.13. Modal *Add Work Hour Data* untuk mengatur jam kerja mingguan Dalam modal ini, untuk setiap hari (*MONDAY* hingga *SUNDAY*), admin dapat mengatur *Open Hour* dan *Close Hour* menggunakan input waktu. Terdapat juga *toggle switch Off Day* untuk setiap hari, yang jika diaktifkan, akan menonaktifkan input jam buka dan tutup, menandakan hari tersebut sebagai hari libur. Setelah konfigurasi selesai, admin menyimpannya dengan tombol *Add*.

2. *Update/Edit* (Ubah Data Jam Kerja): Untuk toko yang sudah memiliki konfigurasi jam kerja, admin dapat mengklik tombol *Edit* yang terletak di samping *dropdown* pemilihan toko. Tindakan ini akan membuka modal *Edit Work Hour Data*, seperti yang diilustrasikan pada Gambar 3.14.



Gambar 3.14. Modal *Edit Workhour Data* untuk mengubah konfigurasi *workhour* mingguan

Antarmuka pada modal ini serupa dengan modal tambah data, memungkinkan admin untuk mengubah jam buka, jam tutup, atau status *Off Day* untuk setiap hari dalam seminggu. Perubahan disimpan dengan mengklik tombol *Save Changes*.

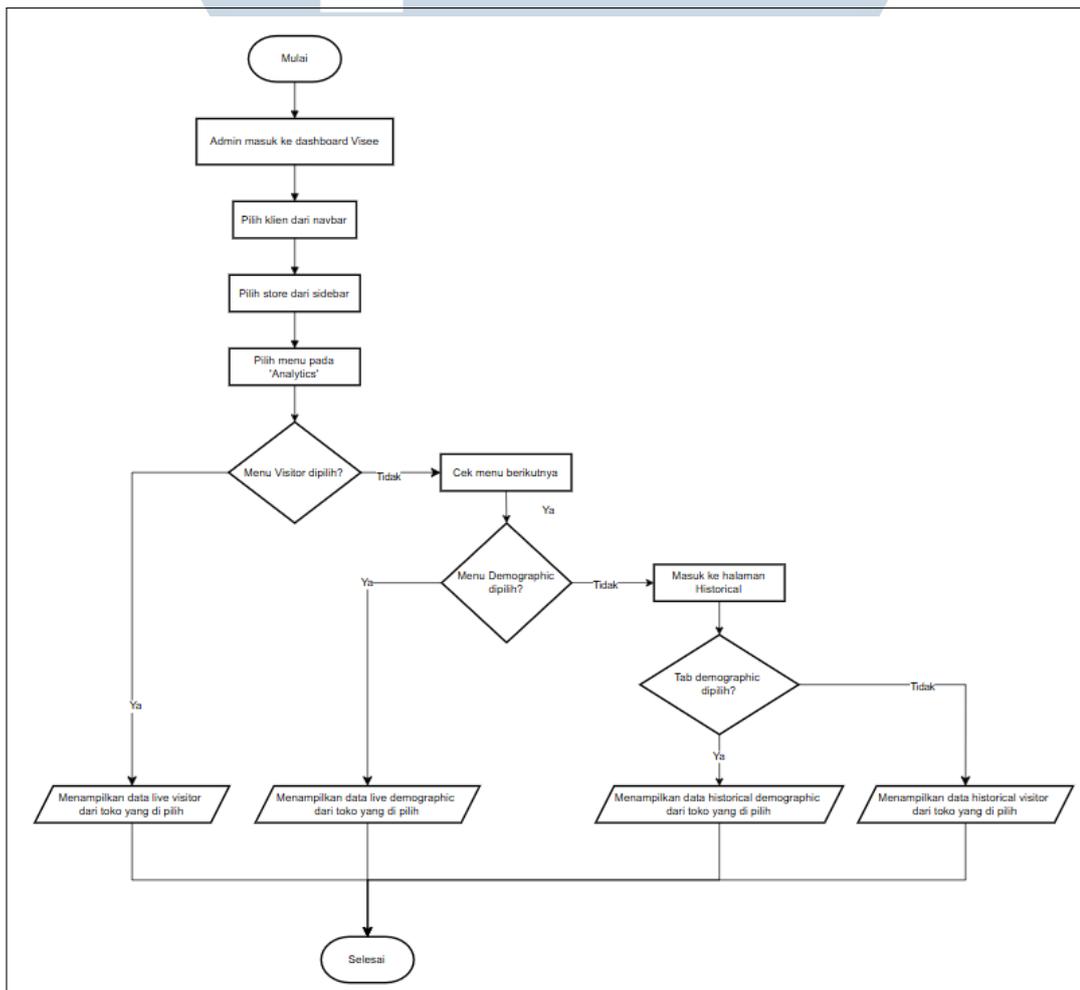
Penambahan *field Store Name* sebagai *filter* utama pada halaman *Zone* dan *Workhour*, serta mekanisme pemilihan berjenjang (*Client* lalu *Store*) pada halaman *Workhour*, memastikan bahwa semua konfigurasi terkait operasional toko dapat dikelola secara terperinci dan akurat. Setelah *field Store Name* diintegrasikan dan alur pengelolaan disesuaikan, dilakukan pengujian menyeluruh untuk memastikan informasi toko tersimpan, ditampilkan, dan dapat dikelola secara konsisten di seluruh bagian sistem yang relevan.

### 3.3.4 Integrasi Halaman Analitik Pengunjung

Setelah fungsionalitas pengelolaan data *master* (*Store*, *Zone*, *Workhour*) selesai dikembangkan dan terintegrasi, fokus pekerjaan beralih pada integrasi halaman-halaman analitik pengunjung ke dalam dasbor Visee. Tujuan utama dari tahap ini adalah menyediakan visualisasi data yang komprehensif bagi pengguna untuk memantau dan menganalisis aktivitas pengunjung di toko mereka.

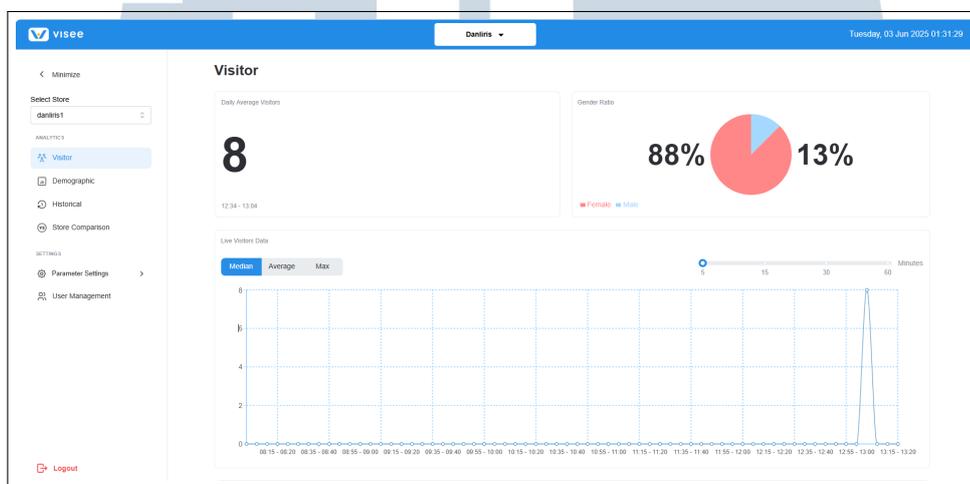
Integrasi ini mencakup beberapa halaman analitik utama, yaitu *Live Visitor*, *Live Demographic*, *Historical Visitor*, dan *Historical Demographic*. Penting untuk dicatat bahwa zona yang ditampilkan pada halaman *Visitor* dan *Demographic* (baik *live* maupun *historical*) berbeda berdasarkan tipe kamera yang terpasang pada zona tersebut.

Alur kerja untuk mengakses halaman analitik dimulai setelah pengguna memilih klien dan toko yang spesifik. Dari *sidebar*, pengguna dapat memilih menu analitik yang diinginkan, yaitu *Visitor*, *Demographic*, atau *Historical*. Sistem kemudian akan langsung menampilkan data yang relevan sesuai dengan menu yang dipilih oleh pengguna untuk toko tersebut. Proses sederhana ini memastikan pengguna dapat dengan cepat mengakses berbagai jenis data analitik yang tersedia. Diagram alir yang menggambarkan proses tampilan data analitik ini dapat dilihat pada Gambar 3.15.



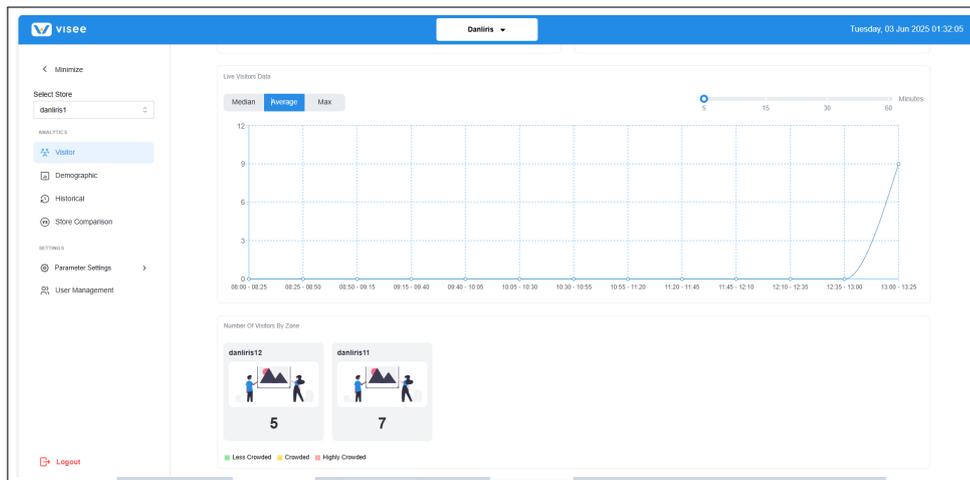
Gambar 3.15. *Flowchart* Proses Tampilan Data Analitik

Salah satu halaman analitik kunci yang diintegrasikan adalah halaman *Live Visitor*. Pada halaman ini, pengguna dapat memilih toko spesifik melalui *dropdown Select Store* yang terletak di *sidebar*. Setelah toko dipilih, halaman akan menampilkan berbagai metrik terkait pengunjung toko tersebut. Sebagai contoh, pada Gambar 3.16, ditampilkan halaman *Live Visitor* untuk toko "danliris1". Informasi yang disajikan meliputi ringkasan jumlah rata-rata pengunjung harian (*Daily Average Visitors*) dan rasio gender pengunjung (*Gender Ratio*) yang divisualisasikan dalam bentuk diagram lingkaran.



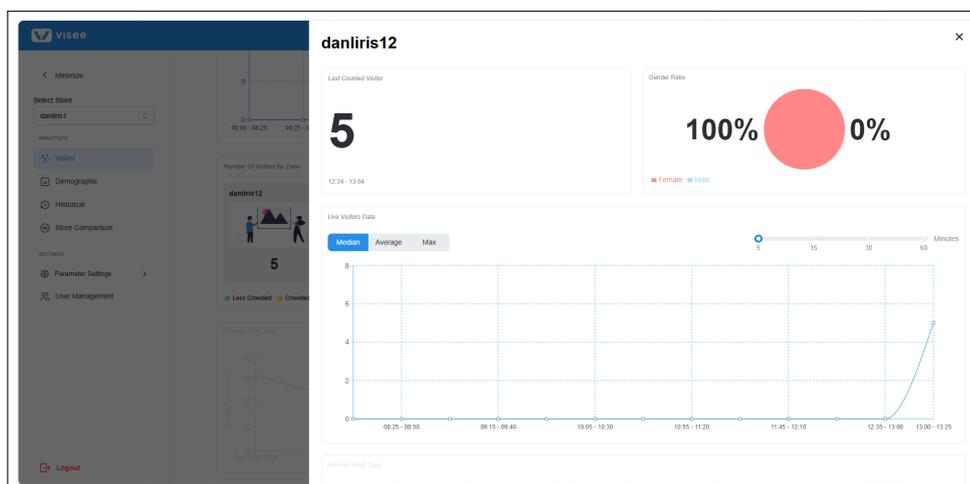
Gambar 3.16. Tampilan utama halaman *Live Visitor* setelah memilih toko

Di bawah ringkasan tersebut, terdapat grafik garis interaktif *Live Visitors Data* yang menampilkan tren jumlah pengunjung secara *real-time* atau dalam interval waktu tertentu (misalnya, per 5, 15, 30, atau 60 menit). Pengguna dapat memilih untuk melihat data berdasarkan nilai median, rata-rata (*average*), atau maksimum. Lebih lanjut, seperti yang ditunjukkan pada Gambar 3.17, halaman ini juga menyajikan informasi *Number Of Visitors By Zone*. Bagian ini menampilkan kartu-kartu untuk setiap zona yang terdaftar di toko yang dipilih dan memiliki tipe kamera *Heatmap Camera* (misalnya, "danliris12", "danliris11"), lengkap dengan jumlah pengunjung saat ini di zona tersebut dan indikator tingkat keramaian (misalnya, *Less Crowded*, *Crowded*, *Highly Crowded*). Zona dengan tipe kamera selain *Heatmap Camera* tidak akan ditampilkan pada halaman *Live Visitor* ini.



Gambar 3.17. Tampilan bagian *Live Visitors Data* dan *Number Of Visitors By Zone* pada halaman *Live Visitor*

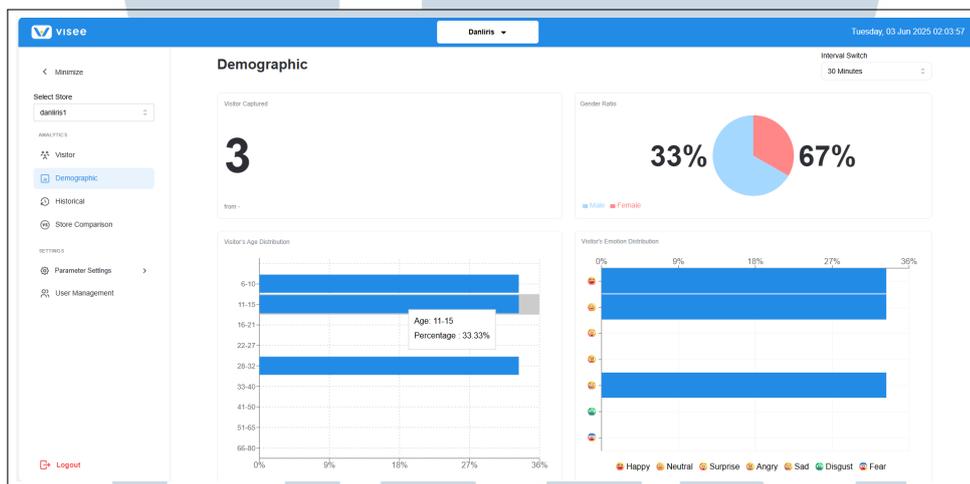
Untuk analisis yang lebih mendalam per zona, pengguna dapat mengklik salah satu kartu zona. Tindakan ini akan memunculkan sebuah jendela modal yang menampilkan detail analitik khusus untuk zona yang dipilih, sebagaimana diilustrasikan pada Gambar 3.18 untuk zona "danliris12". Modal ini menyajikan informasi seperti *Last Counted Visitor* di zona tersebut, *Gender Ratio* spesifik zona, serta grafik *Live Visitors Data* yang hanya mencakup data dari zona yang bersangkutan.



Gambar 3.18. Tampilan modal analitik detail untuk zona spesifik pada halaman *Live Visitor*

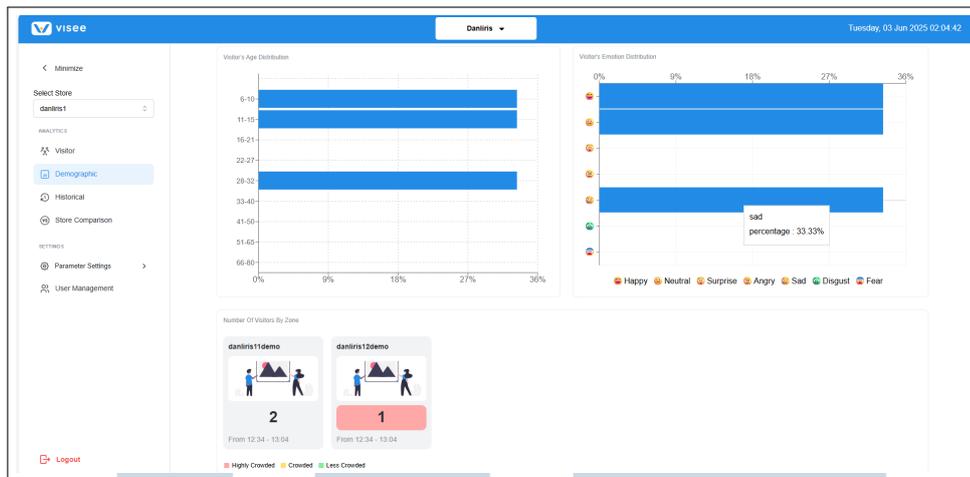
Selanjutnya, integrasi dilakukan untuk halaman *Live Demographic*. Serupa dengan halaman *Live Visitor*, pengguna pertama-tama memilih toko yang ingin dianalisis. Halaman *Live Demographic* kemudian menyajikan data demografis pengunjung secara *real-time* atau berdasarkan interval waktu yang dapat dipilih

melalui *dropdown Interval Switch* (misalnya, per 30 menit). Seperti terlihat pada Gambar 3.19, setelah toko "danliris1" dipilih, halaman menampilkan jumlah total pengunjung yang datanya terekam (*Visitor Captured*) dan *Gender Ratio* keseluruhan pengunjung toko dalam bentuk diagram lingkaran. Di bawahnya, terdapat dua grafik utama: *Visitor's Age Distribution* yang menampilkan distribusi usia pengunjung dalam berbagai rentang usia (misalnya, 6-10, 11-15, dst.) dalam format diagram batang horizontal, dan *Visitor's Emotion Distribution* yang menyajikan persentase berbagai emosi yang terdeteksi dari pengunjung (seperti *Happy, Neutral, Surprise, Angry, Sad, Disgust, Fear*) juga dalam bentuk diagram batang horizontal. Pengguna dapat mengarahkan kursor ke setiap batang untuk melihat persentase detailnya.



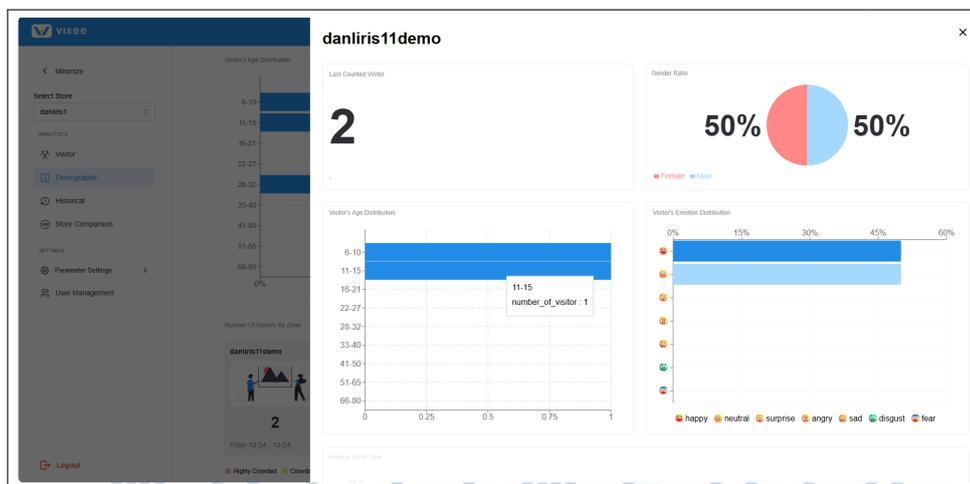
Gambar 3.19. Tampilan utama halaman *Live Demographic* setelah memilih toko

Di bagian bawah halaman *Live Demographic*, juga terdapat bagian *Number Of Visitors By Zone*, seperti yang ditunjukkan pada Gambar 3.20. Bagian ini menampilkan jumlah pengunjung yang terdeteksi di setiap zona spesifik yang memiliki tipe kamera *Demographic Camera* (misalnya, "danliris11demo", "danliris12demo") beserta indikator keramaian masing-masing zona. Zona dengan tipe kamera selain *Demographic Camera* tidak akan muncul di sini.



Gambar 3.20. Tampilan bagian *Number Of Visitors By Zone* pada halaman *Live Demographic*

Ketika pengguna mengklik salah satu kartu zona pada bagian *Number Of Visitors By Zone*, sebuah modal akan muncul menampilkan data demografis yang lebih terperinci khusus untuk zona tersebut. Sebagai contoh, Gambar 3.21 menampilkan modal untuk zona "danliris11demo". Modal ini menyajikan *Last Counted Visitor* di zona tersebut, *Gender Ratio* spesifik zona, *Visitor's Age Distribution* di zona tersebut, dan *Visitor's Emotion Distribution* yang hanya mencakup data dari pengunjung yang terdeteksi di zona "danliris11demo".

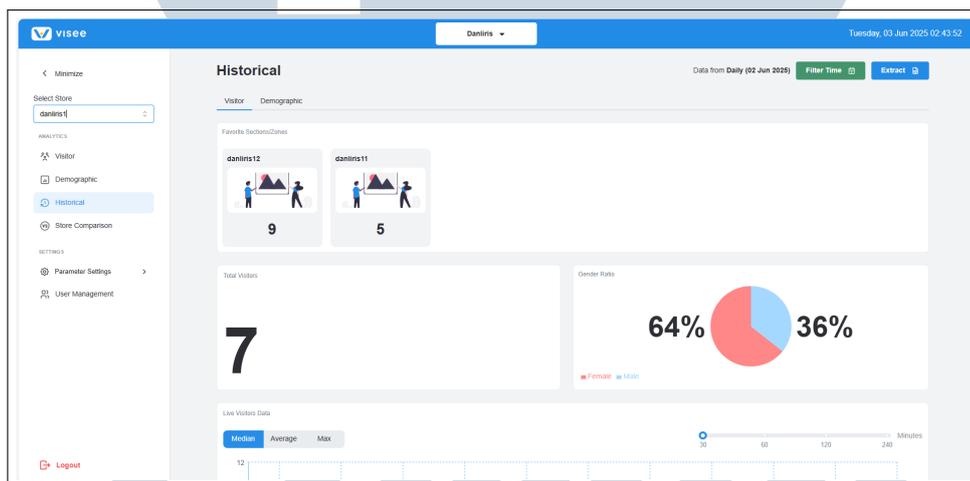


Gambar 3.21. Tampilan modal *Live Demographic* detail untuk zona spesifik

Integrasi halaman *Historical* memberikan kemampuan kepada pengguna untuk menganalisis data pengunjung dan demografis berdasarkan rentang waktu historis. Halaman ini memiliki dua tab utama, yaitu *Visitor* dan *Demographic*. Pada tab *Visitor* di halaman *Historical*, setelah pengguna memilih toko (misalnya,

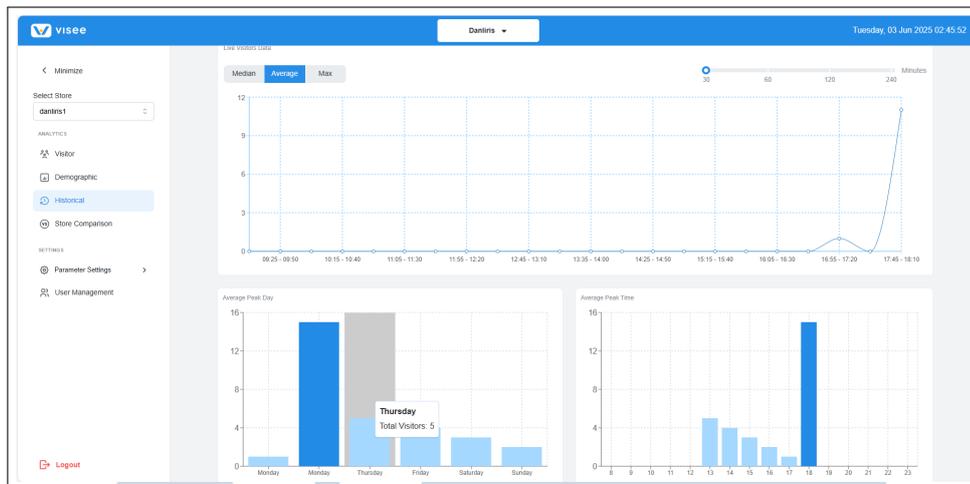
”danliris1”) melalui *dropdown Select Store*, akan ditampilkan data historis sesuai dengan *filter* waktu yang diterapkan. Pengguna dapat memilih rentang waktu (harian) melalui tombol *Filter Time*. Data yang telah difilter kemudian dapat diekstraksi atau diunduh melalui tombol ‘*Extract*’.

Seperti yang terlihat pada Gambar 3.22, halaman *Historical Visitor* menampilkan beberapa metrik kunci. Di bagian atas, terdapat ‘*Favorite Sections/Zones* yang menunjukkan zona-zona dengan tipe kamera *Heatmap* yang paling banyak dikunjungi (misalnya, ”danliris12” dengan 9 pengunjung dan ”danliris11” dengan 5 pengunjung) selama periode waktu yang dipilih. Di bawahnya, ditampilkan ringkasan ‘*Total Visitors* dan *Gender Ratio* untuk periode tersebut. Grafik garis *Live Visitors Data* (dalam konteks historis) juga tersedia, menampilkan tren jumlah pengunjung dari waktu ke waktu selama periode yang dipilih, dengan opsi untuk melihat median, rata-rata, atau maksimum.



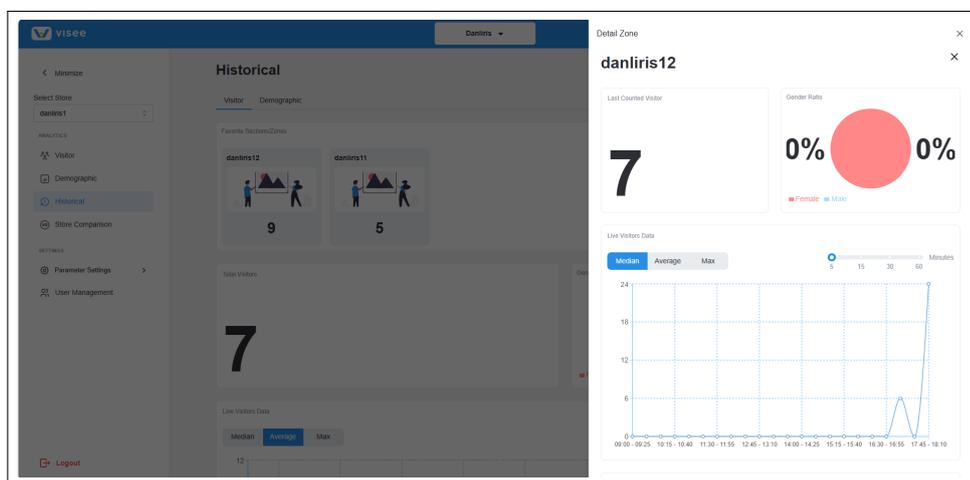
Gambar 3.22. Tampilan utama tab *Visitor* pada halaman *Historical* setelah memilih toko

Lebih lanjut, halaman *Historical Visitor* juga menyajikan analisis tren pengunjung yang lebih mendalam. Sebagaimana diilustrasikan pada Gambar 3.23, terdapat grafik batang *Average Peak Day* yang menunjukkan rata-rata jumlah pengunjung tertinggi pada setiap hari dalam seminggu (misalnya, Senin, Selasa, dst.) selama periode waktu yang dipilih. Di sebelahnya, grafik batang *Average Peak Time* menampilkan rata-rata jumlah pengunjung tertinggi pada setiap jam dalam sehari. Informasi ini sangat berguna untuk memahami pola keramaian toko.



Gambar 3.23. Tampilan bagian grafik *Average Peak Day* dan *Average Peak Time* pada tab *Visitor* halaman *Historical*

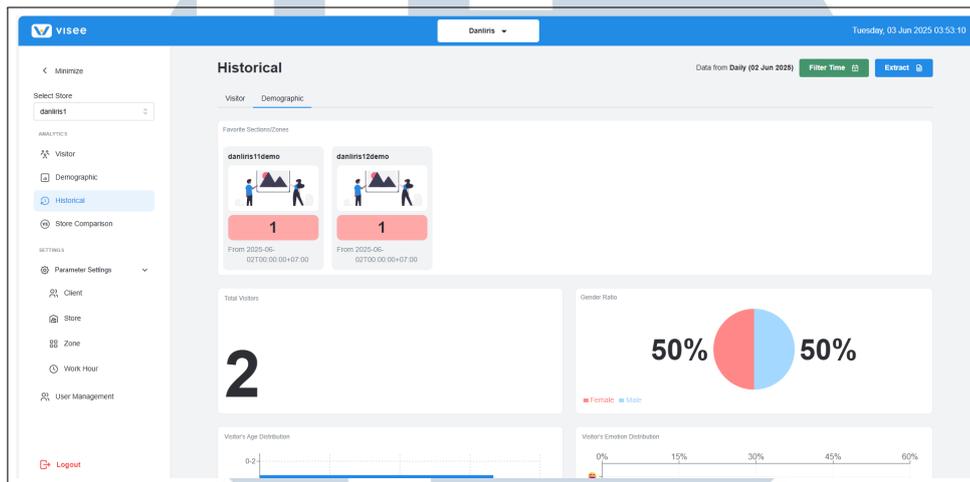
Serupa dengan halaman analitik *live*, pengguna juga dapat melihat detail historis per zona (dengan tipe kamera *Heatmap*). Dengan mengklik salah satu kartu pada bagian *Favorite Sections/Zones*, sebuah modal *Detail Zone* akan muncul. Sebagai contoh, Gambar 3.24 menunjukkan modal untuk zona "danliris12", yang menampilkan *Last Counted Visitor* (untuk periode historis yang dipilih), *Gender Ratio* spesifik zona, dan grafik *Historical Visitors Data* untuk zona tersebut.



Gambar 3.24. Tampilan modal detail zona historis pada tab *Visitor* halaman *Historical*

Tab *Demographic* pada halaman *Historical* menyajikan analisis data demografis pengunjung berdasarkan periode waktu historis yang telah dipilih. Setelah pengguna memilih toko dan menerapkan *filter* waktu, halaman akan menampilkan informasi serupa dengan halaman *Live Demographic*, namun dalam konteks data historis. Seperti yang diilustrasikan pada Gambar 3.25, bagian

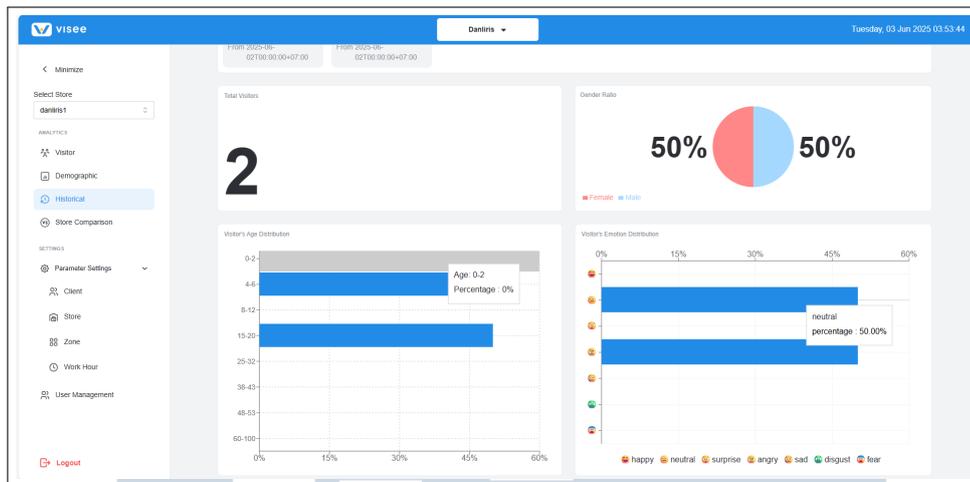
atas menampilkan *Favorite Sections/Zones* yang menunjukkan jumlah pengunjung unik per zona dengan tipe kamera *Demographic Camera* selama periode tersebut. Di bawahnya, terdapat ringkasan *Total Visitors* (atau total data demografis yang terekam) dan *Gender Ratio* untuk keseluruhan pengunjung selama periode waktu yang dipilih.



Gambar 3.25. Tampilan utama tab *Demographic* pada halaman *Historical* setelah memilih toko

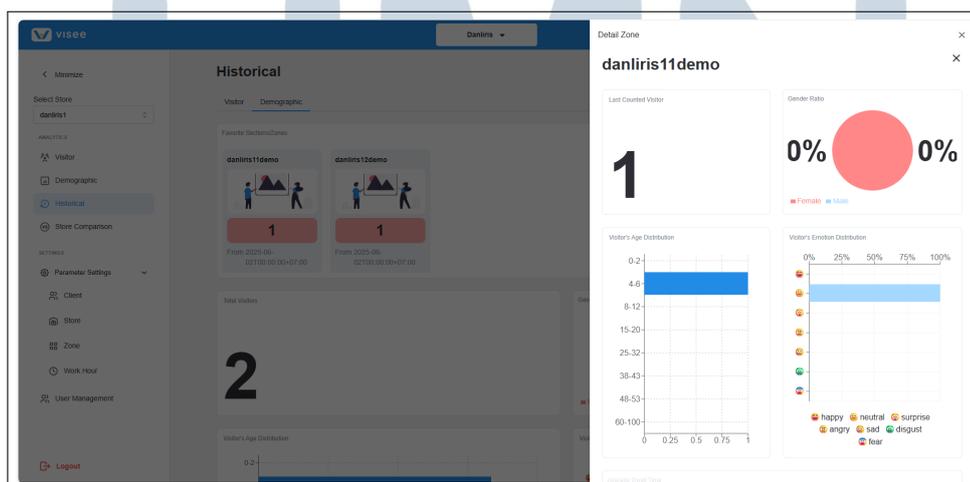
Selanjutnya, disajikan grafik *Visitor's Age Distribution* dan *Visitor's Emotion Distribution* yang merangkum data demografis selama periode waktu yang dipilih, seperti yang terlihat pada Gambar 3.26. Grafik distribusi usia menunjukkan persentase pengunjung dalam berbagai rentang usia, sementara grafik distribusi emosi menampilkan persentase emosi yang terdeteksi. Pengguna dapat mengarahkan kursor pada setiap batang grafik untuk melihat detail persentase spesifik, misalnya persentase pengunjung dalam rentang usia "0-2" atau persentase emosi "neutral".

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.26. Tampilan grafik *Visitor's Age Distribution* dan *Visitor's Emotion Distribution* pada tab *Demographic* halaman *Historical*

Sama seperti tab *Visitor* dan halaman *Live Demographic*, pengguna dapat mengklik salah satu kartu zona (dengan tipe kamera *Demographic Camera*) pada bagian *Favorite Sections/Zones* untuk melihat detail demografis historis khusus untuk zona tersebut. Modal *Detail Zone* yang muncul (lihat Gambar 3.27 untuk contoh zona "danliris11demo") akan menampilkan *Last Counted Visitor* (jumlah data demografis dari zona tersebut selama periode historis), *Gender Ratio* spesifik zona, serta grafik *Visitor's Age Distribution* dan *Visitor's Emotion Distribution* yang hanya mencakup data dari zona yang dipilih selama periode waktu yang ditentukan.



Gambar 3.27. Tampilan modal detail demografis zona historis pada tab *Demographic* halaman *Historical*

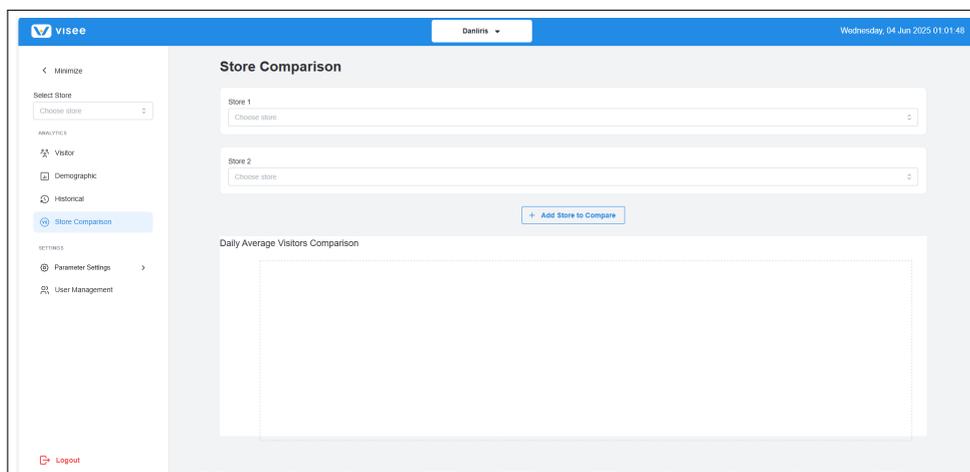
Keseluruhan integrasi halaman analitik ini, baik *live* maupun historis, bertujuan untuk memberdayakan pengguna dengan wawasan yang *actionable* guna

mendukung pengambilan keputusan bisnis yang lebih baik, seperti optimalisasi tata letak toko, penyesuaian strategi pemasaran, dan peningkatan pengalaman pelanggan secara keseluruhan.

### 3.3.5 Pengembangan Fitur Perbandingan Toko (*Store Comparison*)

Setelah integrasi halaman-halaman analitik utama, pengembangan dilanjutkan dengan pembuatan fitur Perbandingan Toko (*Store Comparison*). Fitur ini dirancang untuk memberikan kemudahan kepada pengguna dalam membandingkan kinerja beberapa toko secara langsung dalam satu tampilan. Halaman *Store Comparison* memungkinkan pengguna untuk memilih beberapa toko yang ingin dibandingkan metrik utamanya.

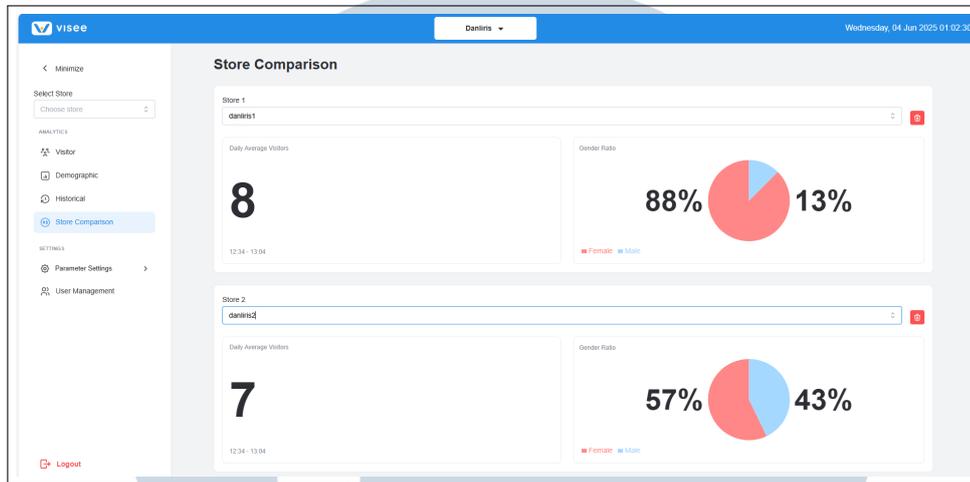
Pada tahap awal, pengguna disajikan dengan dua *dropdown* untuk memilih *Store 1* dan *Store 2* dari daftar toko yang tersedia, seperti yang ditunjukkan pada Gambar 3.28. Terdapat juga tombol *+ Add Store to Compare* yang memungkinkan pengguna untuk menambahkan lebih banyak toko ke dalam perbandingan jika diperlukan. Di bawah pilihan toko, terdapat area yang akan menampilkan grafik perbandingan rata-rata pengunjung harian.



Gambar 3.28. Tampilan awal halaman *Store Comparison*

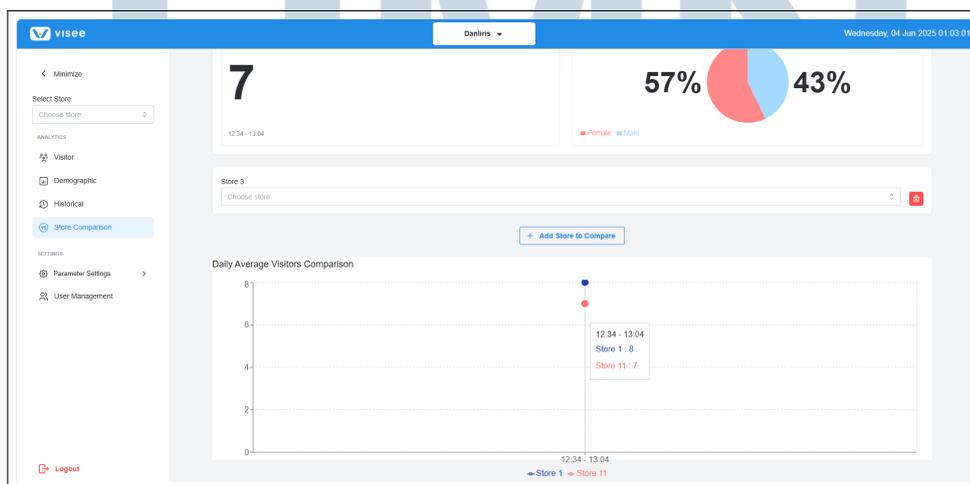
Setelah pengguna memilih toko pada masing-masing *dropdown*, halaman akan menampilkan data ringkas untuk setiap toko yang dipilih. Sebagai contoh, pada Gambar 3.29, setelah "danliris1" dipilih sebagai *Store 1* dan "danliris2" sebagai *Store 2*, akan muncul kartu informasi untuk masing-masing toko. Setiap kartu menampilkan metrik *Daily Average Visitors* (Rata-Rata Pengunjung Harian) dan *Gender Ratio* (Rasio Gender) yang divisualisasikan dalam bentuk diagram

lingkaran. Pengguna juga dapat menghapus toko dari perbandingan dengan mengklik ikon tempat sampah yang tersedia di sebelah kanan setiap pilihan toko.



Gambar 3.29. Tampilan halaman *Store Comparison* setelah memilih dua toko

Ketika data untuk toko-toko yang dipilih telah dimuat, grafik *Daily Average Visitors Comparison* di bagian bawah halaman akan terisi. Grafik ini menampilkan perbandingan langsung rata-rata pengunjung harian antar toko yang dipilih dalam bentuk diagram titik, memudahkan pengguna untuk melihat toko mana yang memiliki kinerja lebih baik dalam hal jumlah rata-rata pengunjung. Pengguna dapat terus menambahkan toko untuk dibandingkan (misalnya, *Store 3* seperti yang terlihat pada Gambar 3.30) dan grafik akan secara dinamis memperbarui data perbandingan.



Gambar 3.30. Tampilan grafik *Daily Average Visitors Comparison*

Fitur *Store Comparison* ini memberikan nilai tambah yang signifikan bagi pengguna yang mengelola banyak cabang toko, karena memungkinkan mereka

untuk dengan cepat mengidentifikasi tren, menganalisis perbedaan kinerja, dan membuat keputusan strategis berdasarkan perbandingan data yang relevan.

### 3.4 Kendala dan Solusi yang Ditemukan

Selama pelaksanaan kerja magang dalam pengembangan fitur *Store Management* pada *website* Visee di Moonlay Technologies, ditemukan beberapa kendala, baik dari aspek non-teknis maupun teknis. Berbagai tantangan ini memerlukan pendekatan yang adaptif dan proaktif agar tidak menghambat jalannya pengembangan proyek. Berikut adalah rincian kendala yang ditemukan selama periode magang.

1. Kurangnya Kesempatan *Networking*: Interaksi yang terbatas dengan staf senior atau profesional di luar tim langsung mengurangi peluang untuk membangun jaringan profesional yang lebih luas dan mendapatkan wawasan mendalam mengenai berbagai aspek industri.
2. Kebutuhan Penguasaan Teknologi Baru: Terdapat kebutuhan untuk menguasai teknologi spesifik yang digunakan dalam proyek Visee, seperti TypeScript dan Golang, yang belum dikuasai secara mendalam sebelumnya. Kurangnya akses terhadap pelatihan formal internal menjadi tantangan tambahan.

Untuk mengatasi berbagai kendala tersebut, serangkaian solusi diterapkan secara efektif. Proses adaptasi dan penyelesaian masalah ini tidak hanya memastikan pengembangan fitur berjalan sesuai target, tetapi juga memberikan pembelajaran berharga dalam menghadapi tantangan di lingkungan kerja profesional. Berikut adalah solusi yang berhasil diimplementasikan.

1. Memanfaatkan Momen Interaksi Secara Proaktif: Setiap sesi pertemuan harian (*daily meeting*) dan sesi *review* dimanfaatkan secara maksimal untuk bertanya, berdiskusi, dan berinteraksi dengan anggota tim yang lebih senior guna memperluas jaringan dan menggali wawasan industri.
2. Pembelajaran Mandiri Secara Intensif: Melakukan studi mendalam secara mandiri melalui berbagai sumber seperti dokumentasi resmi, tutorial daring, dan platform pembelajaran relevan untuk mempercepat penguasaan teknologi yang dibutuhkan (TypeScript dan Golang) dan meningkatkan keterampilan teknis.