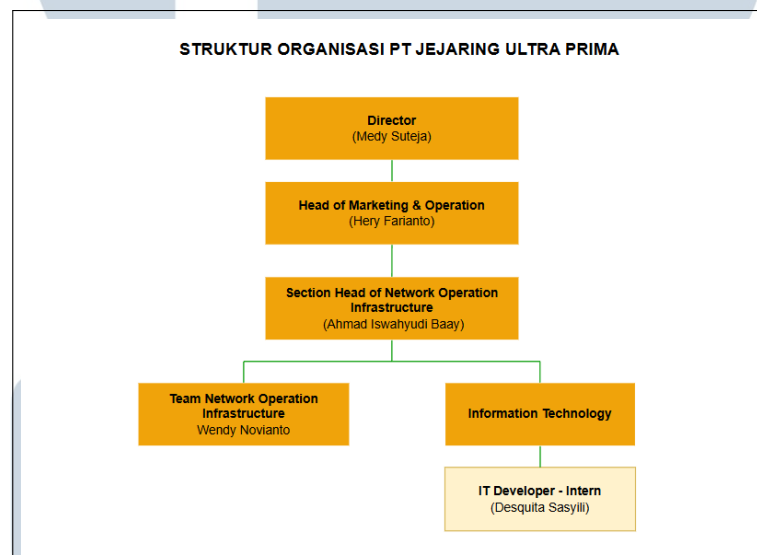


BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Pelaksanaan magang di PT Summarecon Agung Tbk ditugaskan pada anak perusahaannya, yaitu PT Jejaring Ultra Prima, berfokus pada operasional proyek infrastruktur jaringan untuk mendukung penyediaan layanan jaringan pada proyek-proyek *real estate* Summarecon. Kegiatan operasional ini mencakup penyediaan dan pengelolaan jaringan berbasis kabel *fiber optic*, seperti layanan FTTH (*Fiber to the Home*) maupun jaringan seluler. Gambar 3.1 adalah struktur dari perusahaan Jejaring Ultra Prima.



Gambar 3.1. Struktur Organisasi Perusahaan

Sumber: Data Internal Perusahaan [14]

Struktur tim yang tertera pada Gambar 3.1 berada di bawah naungan *Corporate Engineering* yang secara struktural dipimpin oleh Bapak Medy Suteja selaku *Director*. Pada tingkat operasional, unit dikoordinasikan oleh Bapak Hery Farianto sebagai *Head of Marketing & Operation*, serta Bapak Ahmad Iswahyudi Baay sebagai *Section Head of Network Operation Infrastructure*. Tim *Network Operation Infrastructure* bertanggung jawab dalam memantau dan mengelola operasional jaringan agar berjalan secara optimal.

Selama pelaksanaan kerja magang, penulis menempati posisi sebagai *Web*

Developer di bawah naungan divisi *Information Technology* (IT) pada unit *Corporate Engineering*. Pembimbing lapangan atau *supervisor* adalah Bapak Ahmad Iswahyudi Baay, yang berperan dalam memberikan tugas, arahan, serta bimbingan selama proses pengembangan sistem guna memastikan pekerjaan yang dilakukan sesuai dengan kebutuhan dan standar perusahaan.

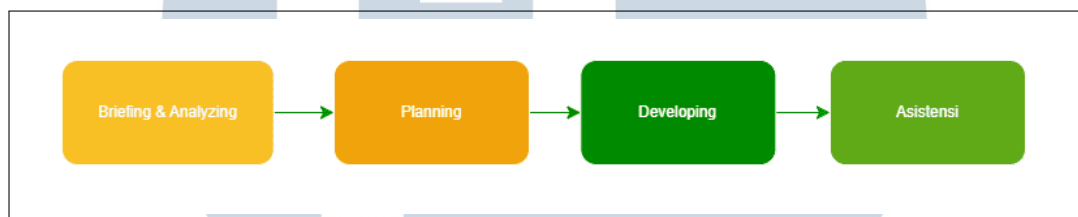
3.2 Tugas yang Dilakukan

Tugas yang dilakukan selama pelaksanaan magang yaitu terlibat langsung dalam kegiatan pengembangan sistem *monitoring* manajemen operasional proyek infrastruktur jaringan berbasis *website dashboard*. *Website* ini mencakup pengelolaan data internal perusahaan seperti dokumen perjanjian kerja sama dengan berbagai penyedia layanan ISP (*Internet Service Provider*), data *homepass*, *design drawing*, *core management* serta data *performance* mengenai keuangan. *website* ini yaitu mempermudah pengelompokan informasi jaringan dari banyaknya proyek infrastruktur yang dikembangkan, serta membantu beberapa pihak untuk mempermudah *monitoring* informasi data yang dimiliki oleh jejaring ultra prima. Hal-hal yang dilakukan pada pelaksanaan kerja magang sebagai berikut.

1. Mendesain tampilan antara muka disesuaikan dengan keinginan yang diminta oleh pihak perusahaan.
2. Merancang dan membuat *schema structure database*. *Database* yang digunakan untuk menyimpan seluruh data, nantinya akan dimasukkan atau tersimpan menggunakan *MySQL Server*.
3. Melakukan pengembangan fungsi pada sisi *frontend* dan *backend* sesuai dengan perancangan yang telah ditetapkan.
4. Mengembangkan pembuatan operasi CRUD (*Create, Read, Update, Delete*) pada *dashboard* admin, serta melakukan *debugging* selama proses uji memastikan program berjalan sesuai dengan fungsinya.
5. Mengembangkan tampilan visualisasi data yang informatif sehingga data dapat dikelompokkan dengan jelas, mudah dipahami, serta sesuai dengan konteksnya.
6. Mengikuti rapat untuk mendapatkan *requirement* yang diperlukan dan mempresentasikan tampilan pembuatan *website* yang dikembangkan.

3.3 Uraian Pelaksanaan Magang

Pada Gambar 3.2 ditunjukkan setiap tugas dimulai dengan *briefing* dan analisis kebutuhan untuk memahami pekerjaan serta menentukan solusi sebelum implementasi. Hasil analisis digunakan dalam perencanaan sistem dan strategi pengembangan. Tahap selanjutnya adalah pengembangan kode, pengujian fungsi, dan penyesuaian tampilan sesuai standar perusahaan. Setelah selesai, dilakukan asistensi dengan *supervisor* untuk evaluasi, validasi, dan *approval* agar hasil akhir dinilai sudah sesuai dengan kebutuhan sistem.



Gambar 3.2. Flow Pengerjaan Fitur

Berdasarkan alur kerja tersebut, setiap tugas memiliki tahapan yang menyesuaikan jenis pekerjaannya. Jika berkaitan dengan perbaikan atau *debugging*, dilakukan analisis mendalam terhadap *log sistem*, *database query*, dan relasi antar data untuk menemukan penyebab masalah, lalu melakukan perbaikan berdasarkan *feedback* perusahaan tanpa mengganggu fungsi sistem lainnya.

Untuk tugas pengembangan fitur baru, kegiatan dimulai dengan penyusunan konsep dan diskusi bersama *supervisor* dan tim terkait agar perancangan sesuai dengan kebutuhan sistem. Secara umum, pelaksanaan magang mengikuti alur kerja yang ditetapkan oleh *supervisor*, mencakup *briefing* dan analisis kebutuhan, perencanaan, pengembangan, serta asistensi dalam evaluasi hasil.

Dengan alur kerja tersebut, setiap tugas dapat dilaksanakan secara terarah dan sistematis, sehingga hasil pengembangan tidak hanya sesuai kebutuhan teknis, tetapi juga membantu meningkatkan kinerja operasional perusahaan. Rincian kegiatan magang diuraikan pada Tabel 3.1.

Tabel 3.1. Pekerjaan Mingguan Selama Magang

Minggu ke-	Pekerjaan yang dilakukan
1	Mengikuti <i>briefing</i> awal terkait pengembangan kebutuhan sistem, implementasi fitur CRUD, penyesuaian tampilan desain CSS, penambahan fitur <i>export file data table</i> , melakukan perbaikan <i>error handling</i> pada <i>performance dashboard</i> serta pengoptimalan bagian <i>frontend</i> dan <i>backend</i> untuk daftar mitra kerja sama dan aset-aset gambar yang dibutuhkan.
2	Pengembangan fungsi <i>search live</i> di semua halaman, implementasi fungsi <i>logic backend</i> mencakup pembuatan struktur <i>database</i> , CRUD beserta tampilan <i>iFrame PDF</i> . Melakukan perbaikan <i>breadcrumb</i> , dan penyesuaian tampilan format <i>export file</i> .
3	Melakukan perbaikan <i>generate password</i> , perbaikan <i>session</i> dan tampilan <i>donut chart</i> , serta <i>query</i> relasi <i>database maintenance</i> . Pembaruan pada hosting dan perancangan fitur baru berdasarkan hasil <i>meeting online</i> dan revisi tampilan bersama <i>supervisor</i> dan tim terkait.
4	Perancangan dan pengembangan <i>mockup gauge chart</i> , <i>query</i> perbandingan <i>budget</i> dan realisasi pada seluruh data finansial beserta integrasi visualisasi data pada <i>bar chart</i> . Perancangan alur sistem pengiriman <i>email</i> otomatis dan perapihan struktur <i>database</i> .
5	Pengembangan format <i>field</i> dan penambahan nilai <i>enum</i> pada fitur tambahan, implementasi pengiriman <i>email</i> otomatis dengan penyesuaian tampilan <i>template email</i> saat terdapat unggahan baru. Melakukan revisi pada <i>script</i> pengiriman <i>email</i> agar terkirim ke bagian <i>finance</i> , konfigurasi mengatur <i>Cron Job</i> untuk pengiriman otomatis dengan jadwal yang ditentukan.
6	Melakukan diskusi dan melaporkan progres bersama tim, terkait tampilan fitur email otomatis yang sudah diimplementasikan, melanjutkan pengembangan kode kolom BAST (Berita Acara Serah Terima). Analisis dan perancangan <i>query</i> CRUD ' <i>ongoing project</i> ', perancangan <i>database</i> baru dan melakukan revisi pada tampilan <i>card summary financial</i> .

Lanjut pada halaman berikutnya

Lanjutan Tabel 3.1. Pekerjaan Mingguan Selama Magang

Minggu Ke-	Pekerjaan yang dilakukan
7	Pengembangan dan perbaikan kode <i>card Key Performance Indicator</i> , mulai melakukan pengembangan <i>CRUD ongoing project</i> , pembuatan desain antar muka serta <i>database PKS ongoing project</i> , perbaikan <i>error</i> input data, disertai diskusi pengembangan fitur bersama tim.
8	Melakukan diskusi bersama <i>supervisor</i> dan tim terkait pengembangan fungsi status PKS. Memperbaiki tampilan <i>card</i> dan ilustrasi agar lebih interaktif, menambahkan tipe <i>enum</i> pada status PKS.
9	Melanjutkan pengembangan fitur <i>CRUD</i> dan tampilan responsif pada <i>dashboard JJUP</i> beserta perancangan <i>logic</i> penyimpanan alamat <i>email</i> mitra.
10	Melakukan diskusi mingguan mengenai pengembangan fitur <i>email reminder by system</i> , pengembangan perbandingan tiga <i>donut chart</i> dengan menampilkan data <i>ascending</i> , perancangan pengiriman <i>email reminder</i> PKS dan <i>revenue sharing</i> otomatis per kuartal, menyesuaikan pengambilan data dari <i>database</i> .
11	Melakukan penyesuaian batas ukuran <i>file</i> yang akan diunggah, menambahkan " <i>progress bar</i> " pada input data, membuat struktur <i>database</i> baru penyimpanan data <i>Utilization</i> beserta <i>logic backend</i> , memperbarui visualisasi <i>gauge chart</i> menampilkan data berdasarkan tahun berjalan dengan tambahan <i>dropdown</i> pemilihan tahun-tahun sebelumnya.
12	Melakukan perbaikan fitur PKS dan <i>email reminder</i> , mengembangkan fungsi <i>search engine</i> dan elemen tampilan, serta merancang dan mengimplementasikan fitur baru terkait data mitra dan PKS pada sistem.
13	Melakukan perbaikan dan debugging <i>donut chart</i> , <i>search engine</i> , mengembangkan halaman <i>Partners</i> serta penambahan <i>enum</i> baru pada fitur PKS dan <i>ongoing project</i> , serta merancang visualisasi <i>gauge chart</i> untuk status PKS Mitra dan Sewa.

Lanjut pada halaman berikutnya

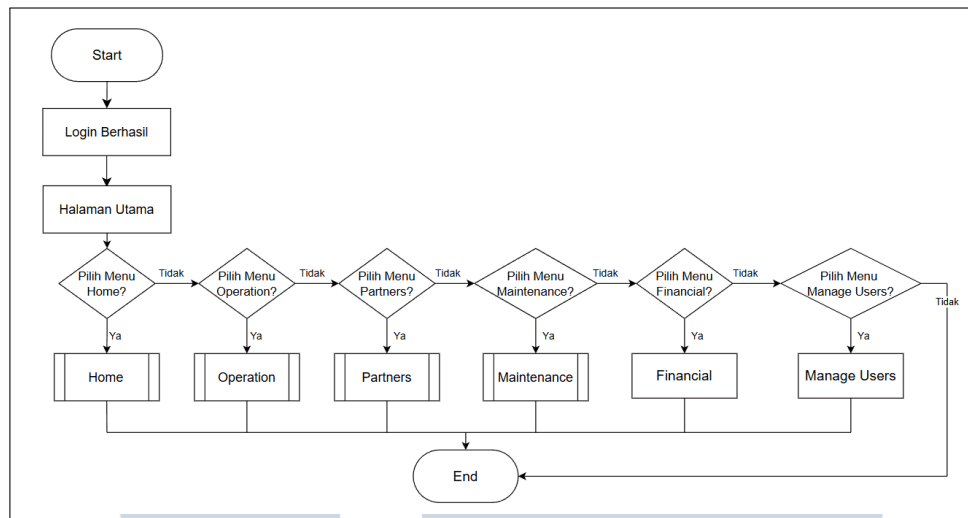
Lanjutan Tabel 3.1. Pekerjaan Mingguan Selama Magang

Minggu Ke-	Pekerjaan yang dilakukan
14	Menyempurnakan tampilan <i>donut chart</i> dan <i>DataTable</i> , memperbaiki elemen UI, menambahkan navigasi dari <i>modal</i> ke halaman data.
15	Memperbaiki <i>error</i> tombol pada <i>modal</i> edit <i>speedometer</i> “ <i>Utilization</i> (RGH)”, merapikan seluruh tampilan <i>frontend</i> , meningkatkan fungsi <i>search live</i> untuk user dan admin.
16	Memperbaiki tampilan <i>card</i> dengan penambahan <i>badge</i> , merapikan CSS <i>card</i> data <i>financial</i> , mengubah halaman <i>maintenance-detail</i> menjadi tabel.
17	Memperbaiki <i>error</i> pada edit data PKS, merapikan halaman data <i>maintenance</i> dengan menyesuaikan format template export PDF. Diskusi terkait fitur upload dokumen <i>multiple</i> , serta merancang <i>database</i> dan alur fitur baru sekaligus merapikan <i>card</i> edit PKS untuk mendukung penambahan dokumen.

3.4 Proses Pengembangan Sistem

Gambar 3.3 menampilkan diagram alur (*flowchart*) yang memberikan gambaran menyeluruh terkait proses utama dalam sistem yang digunakan dan dikembangkan selama kegiatan magang. Diagram ini memperlihatkan alur interaksi pengguna dengan sistem, dimulai dari proses *login* hingga navigasi menuju berbagai menu utama, seperti *Home*, *Operation*, *Partners*, *Maintenance*, *Financial*, dan *Manage Users*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.3. Flowchart Keseluruhan Sistem

Setiap menu memiliki fungsinya masing-masing dan membawa pengguna ke fitur pengelolaan data melalui operasi CRUD. *Flowchart* tersebut juga menampilkan modul-modul yang menjadi fokus pengembangan, seperti pengalihan otomatis ke halaman *Home* saat pertama kali *login*, pengelolaan data PKS beserta fitur *email blast* dan *email reminder* pada menu *Operation*, daftar mitra kerja sama pada *Partners*, pengelolaan data perawatan pada *Maintenance*, serta pengembangan fungsi pada menu *Financial*.

Pembahasan lebih mendalam mengenai masing-masing modul yang ditampilkan dalam *flowchart* akan dijelaskan lebih lanjut pada subbab berikutnya, dengan fokus pada komponen yang berhubungan langsung dengan proses pengembangan yang dilakukan.

3.4.1 Pengembangan Halaman Home

A Analisis Kebutuhan

Dalam proses pengembangan halaman *Home*, diperlukan sebuah halaman khusus yang menampilkan *Key Performance Indicator* (KPI) dalam bentuk visualisasi *gauge chart* interaktif. Pada halaman ini terdapat tiga *gauge chart* utama, yaitu *Utilization* (RGH), *Revenue*, dan *CAPEX*. Penjelasan lebih lanjut mengenai masing-masing *gauge chart* dijabarkan sebagai berikut.

- Utilization

Data diinput secara manual oleh admin melalui CRUD, mencakup nilai

utilization dan persentasenya, memiliki rentang nilai persentase dari 0% hingga 100%

- Revenue dan CAPEX

Data diambil secara otomatis dari tabel *financial* pada basis data *revenue* dan *capex*. Saat halaman pertama kali dibuka, sistem menampilkan tahun berjalan atau tahun teratas sebagai nilai *default*, dengan rentang 0% hingga 120% untuk mengakomodasi kemungkinan realisasi melebihi nilai *budget*. Selain itu, tersedia fitur *dropdown* pemilihan tahun untuk menampilkan data tahun sebelumnya sebagai bahan analisis perbandingan antar tahun.

Gauge Chart Revenue dan *CAPEX* menghitung persentase perbandingan antara *Budget* dan Realisasi untuk menentukan apakah kinerja meningkat, stabil, atau menurun. Warna indikator pada *speedometer* akan menyesuaikan tampilan dari hasil perhitungan.

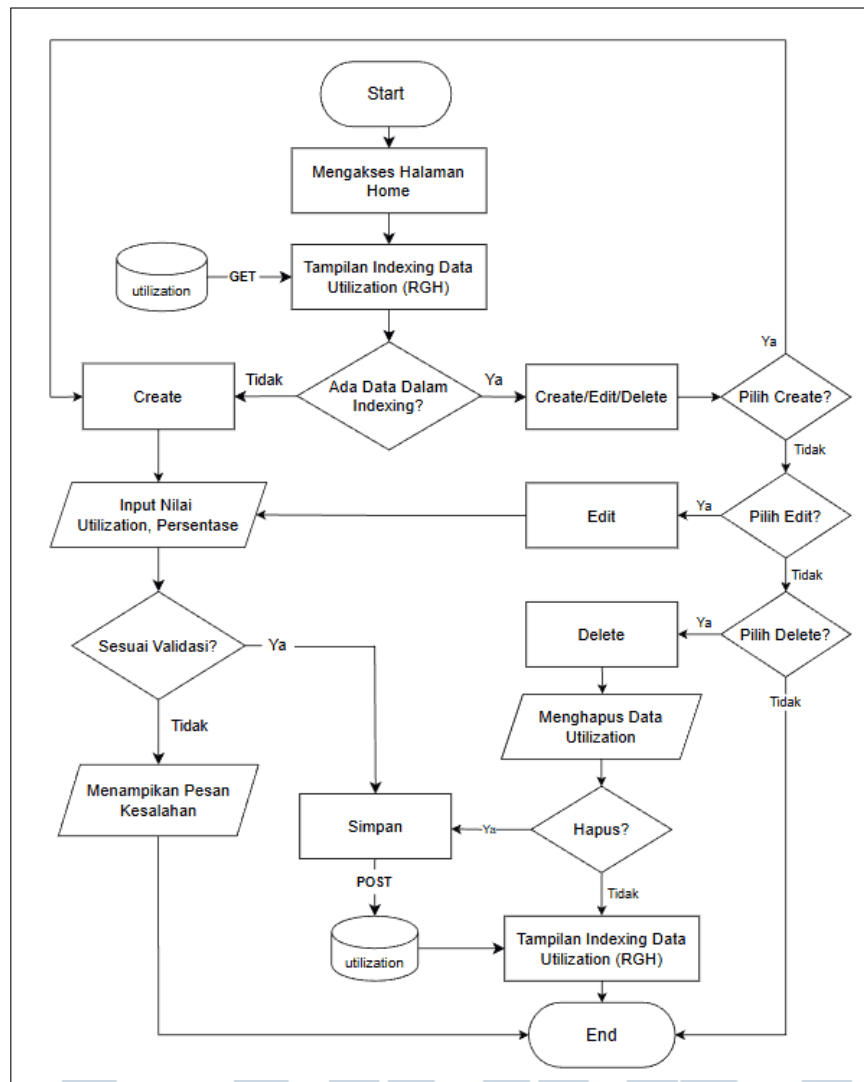
- Utilization dan Revenue: merah → oranye → biru → hijau.
- CAPEX: hijau → biru → oranye → merah (dimulai dari warna hijau karena semakin tinggi persentase *CAPEX*, semakin berisiko dan menunjukkan bahwa realisasi telah melebihi *budget*).

Dari sisi tampilan dan pengalaman pengguna, halaman ini dirancang agar memudahkan pengguna dalam mengakses data penting secara visual dan mudah dipahami. Integrasi langsung dengan tabel basis data *revenue* dan *capex* menjadi aspek penting untuk mendukung kemudahan pengelolaan data perusahaan serta pemantauan informasi secara *real-time*.

B Flowchart

- Flowchart Proses CRUD Data Utilization

Proses CRUD data utilization akan dijelaskan melalui *Flowchart* pada Gambar 3.4 di bawah ini.



Gambar 3.4. Flowchart Proses CRUD Data Utilization

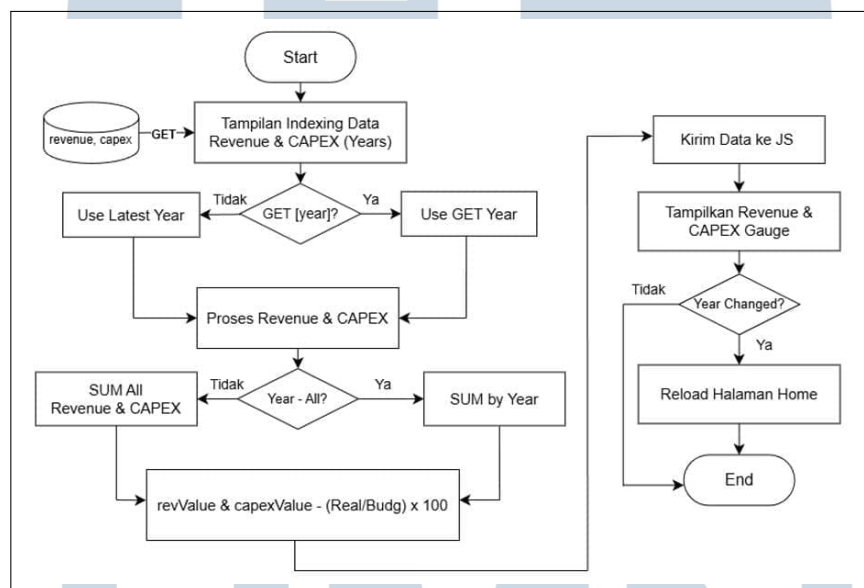
Gambar 3.4 diatas menjelaskan alur proses CRUD untuk fitur manajemen data *Utilization* (RGH). Proses dimulai saat admin mengakses halaman *Home*, di mana sistem langsung menampilkan indeks data dan melakukan permintaan *GET* ke basis data *utilization* untuk menampilkan nilai dalam bentuk visualisasi *gauge chart*. Jika tidak ada data, admin dapat menambahkan entri baru melalui fitur *Create*, dengan menginput nilai dan persentase data *Utilization*, kemudian melewati proses validasi. Jika terdapat kesalahan dalam inputan, sistem akan menampilkan pesan kesalahan agar dapat memperbaiki input. Bila validasi sukses, data disimpan melalui metode *POST* dan halaman indeks ditampilkan kembali.

Setelah data tersedia, admin dapat melakukan *Create*, *Edit*, atau *Delete*.

Pada proses *Edit*, data akan diverifikasi lalu disimpan ke basis data sebelum kembali ke halaman indeks. Aksi *Delete* langsung menghapus data dan menampilkan kembali halaman indeks. Setiap proses CRUD selalu diarahkan ke halaman indeks agar perubahan data dapat langsung terlihat dan alur navigasi tetap konsisten.

- Flowchart Integrasi Gauge Chart Revenue dan CAPEX

Pada Gambar 3.5 di bawah ini akan menjelaskan alur bagaimana sistem mengolah dan menampilkan *gauge chart* untuk data *Revenue* dan *CAPEX*.



Gambar 3.5. Flowchart Integrasi Gauge Chart Revenue dan CAPEX

Prosesnya dimulai ketika sistem menampilkan data berdasarkan tahun yang diambil melalui permintaan *GET* pada basis data *revenue* dan *capex*. Pada tahap awal, sistem melakukan pengecekan apakah parameter tahun sudah tersedia pada URL. Jika ada, sistem langsung menggunakan tahun tersebut, namun jika tidak ada, sistem akan memilih tahun terbaru atau teratas sebagai nilai *default*.

Setelah itu, untuk fitur *dropdown* pemilihan data pertahun, sistem memproses data *Revenue* dan *CAPEX* sesuai dengan tahun yang dipilih. Jika pengguna memilih satu tahun tertentu, maka perhitungan hanya dilakukan pada data di tahun tersebut. Namun apabila pengguna memilih opsi “*All*”, sistem akan menjumlahkan semua data *Revenue* dan *CAPEX* dari berbagai tahun. Nilai persentasenya dihitung menggunakan Rumus di bawah ini.

Rumus 3.1 menunjukkan cara perhitungan persentase.

$$value = \left(\frac{Realisasi}{Budget} \right) \times 100 \quad (3.1)$$

Rumus Persamaan 3.1 digunakan untuk mengetahui seberapa besar capaian (realisasi) dibandingkan dengan anggaran yang sudah direncanakan (*budget*). Jika nilai realisasi semakin mendekati atau melebihi *budget*, maka persentasenya juga akan semakin besar.

Setelah perhitungan selesai, hasilnya akan dikirim ke sisi *frontend* dan ditampilkan dalam bentuk *gauge chart*. Jika pengguna mengganti tahun, halaman *Home* akan dimuat ulang untuk menyesuaikan data. Jika tidak ada perubahan, tampilan tetap menggunakan data yang sedang aktif, sehingga grafik selalu sesuai dengan tahun yang dipilih.

C Perancangan Basis Data

Pada Tabel 3.2 di bawah menjelaskan rincian struktur basis data *utilization* berfungsi sebagai komponen penyimpanan data terkait tingkat pemanfaatan pada sistem.

Tabel 3.2. Struktur Tabel Utilization

Nama Kolom	Type Data	NULL	Keterangan
id	INT (11)	Tidak	Primary Key
nilai	DECIMAL (15, 2)	Tidak	Nilai Utilization (angka)
persen	DECIMAL (5, 2)	Tidak	Persentase Utilization
updated_at	TIMESTAMP	Tidak	Mencatat waktu

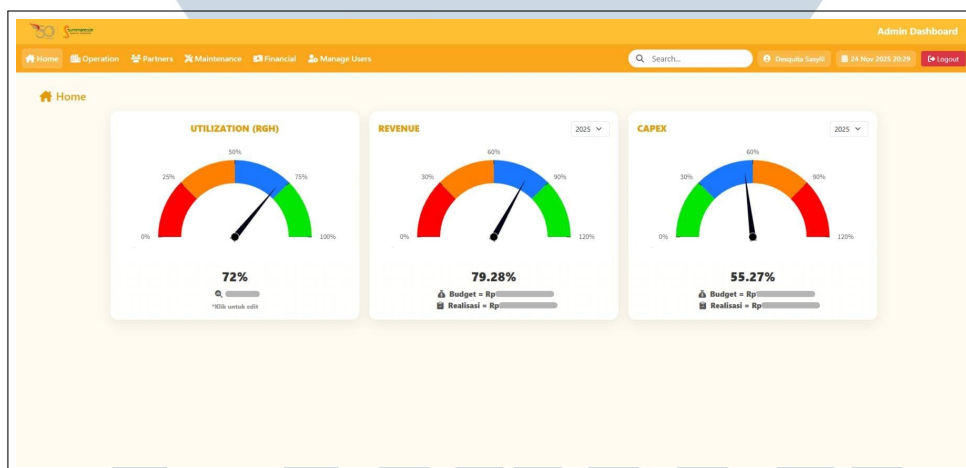
Tabel ini merekam informasi utama berupa nilai *utilization* dan persentasenya, yang digunakan untuk mendukung proses monitoring maupun analisis kinerja. Kolom *id* bertipe *int (11)* digunakan sebagai *primary key* dan menjadi identitas unik setiap data *utilization* yang tersimpan. Kolom *nilai* berisi angka *utilization* dalam format *decimal (15, 2)*, sedangkan kolom *persen* menyimpan persentase

pemanfaatan dalam format `decimal(5,2)`, keduanya diset sebagai *NOT NULL* dengan nilai awal (*default*) 0.00 untuk memastikan konsistensi dan validitas data.

Tabel ini dilengkapi kolom `updated_at` bertipe `timestamp` yang diperbarui otomatis saat data berubah, sehingga memudahkan pelacakan pembaruan. Struktur tabel dirancang agar pencatatan data *utilization* akurat dan siap digunakan untuk kebutuhan operasional, visualisasi, serta penyusunan laporan.

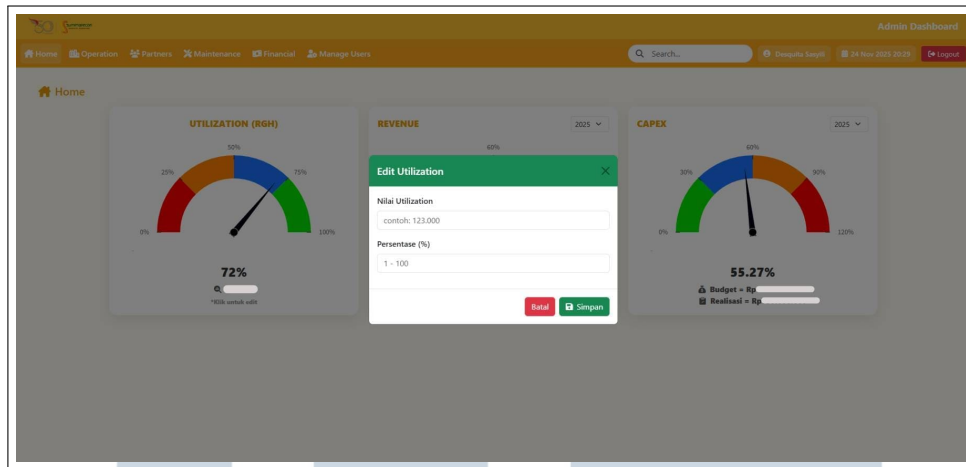
D Implementasi

Halaman *home* menjadi salah satu bagian yang penting karena berfungsi sebagai pusat informasi terkait *key performance indicator* (KPI), seperti data *Utilization* (RGH), *Revenue*, dan *CAPEX*. Dimana pada awal pengguna berhasil melakukan login akan langsung diarahkan ke halaman ini. Tampilan lengkap halaman ini dapat dilihat pada Gambar 3.6.



Gambar 3.6. Tampilan Utama dan Indexing Halaman Home

Pada tampilan utama Gambar 3.6, sistem menampilkan visualisasi data dalam bentuk *gauge chart*. Setiap *gauge* menampilkan informasi yang berbeda, untuk *gauge utilization* menampilkan nilai dan persentasenya, sedangkan *gauge revenue* dan *CAPEX* menampilkan nilai *budget*, realisasi, serta persentase hasil perbandingan antara keduanya. Selain itu, setiap *gauge* juga dilengkapi indikator berwarna seperti *speedometer*, lengkap dengan rentang persentase masing-masing. Pengguna juga dapat memilih tahun tertentu untuk *filter* data *revenue* dan *CAPEX*.



Gambar 3.7. Tampilan Modal Form CRUD Data Utilization (RGH)

Gambar 3.7 menampilkan modal form yang digunakan admin untuk melakukan CRUD (*Create, read, update, delete*) data *utilization*. Form ini berisi dua input, yaitu nilai dan persentase *utilization*. Setelah data diisi dan dikirim, sistem akan memvalidasi input terlebih dahulu. Jika data valid, informasi akan disimpan ke tabel *utilization* melalui proses di sisi *server*, seperti ditunjukkan pada Kode 3.1.

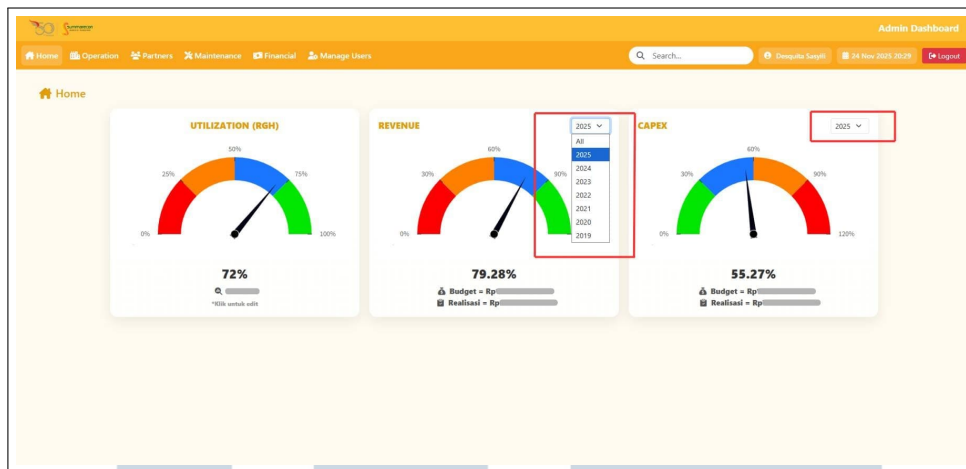
```

1  /* UTILIZATION */
2  $utilRow = $con->query("SELECT * FROM utilization ORDER BY id DESC
    LIMIT 1")->fetch_assoc();
3  $utilValue = $utilRow['persen'] ?? 0;
4  $utilTextValue = number_format($utilRow['nilai'] ?? 0, 0, ',', '.');
5
6  if (isset($_POST['update_util'])) {
7      $nilai = str_replace('.', '', $_POST['util_nilai']);
8      $persen = $_POST['util_persen'];
9      $con->query("UPDATE utilization
10         SET nilai='$nilai', persen='$persen'
11         ORDER BY id DESC LIMIT 1");
12     echo "<meta http-equiv='refresh' content='0'>";
13     exit;
14 }

```

Kode 3.1: Proses Penyimpanan Data Utilization

Untuk fitur filter data *revenue* dan *CAPEX* per tahun, maupun untuk menampilkan total dari semua tahun yang sudah tersimpan pada basis data *revenue* dan *capex*, dapat dilihat pada Gambar 3.8.



Gambar 3.8. Tampilan Filter Tahun Data Revenue dan CAPEX

Pada proses pemilihan tahun, pengguna hanya perlu menekan *dropdown* yang terdapat di bagian kanan atas pada *card gauge chart*. Setelah *dropdown* terbuka, sistem akan menampilkan daftar tahun yang tersedia sesuai data yang ada di dalam basis data. Ketika pengguna memilih salah satu tahun, sistem akan otomatis memuat ulang halaman dan memperbarui tampilan data, mulai dari nilai *budget*, realisasi *revenue* dan *CAPEX*, hingga indikator warna pada *speedometer*. Semua informasi tersebut akan menyesuaikan dengan data yang tersimpan pada tabel *revenue* dan *capex*, sehingga pengguna bisa langsung melihat perubahan datanya secara *real-time*.

3.4.2 Pengembangan Halaman Operation

A Analisis Kebutuhan

Dalam pengembangan halaman *Operation*, sistem menampilkan berbagai *summary* yang diambil dari beberapa halaman melalui operasi CRUD. Pada *dashboard* ini terdapat *card summary* untuk *Ongoing Project*, *Task*, dan *Network Issues*. Selain itu, ditampilkan juga *bar chart* jumlah data terkait serta *donut chart* yang menampilkan total PKS dari basis data pks. Tiga *donut chart* menunjukkan total PKS berdasarkan statusnya. *Detail* setiap konten pada *dashboard* dijelaskan sebagai berikut.

- Card Summary Ongoing Project

Card ini menampilkan daftar PKS *Ongoing Project* dalam bentuk *modal* tabel interaktif. Data yang ditampilkan mencakup PKS dengan status *active*

dan *will expire*, serta dilengkapi fitur *pagination* dan pencarian. Saat salah satu baris dipilih, sistem akan mengarahkan pengguna ke halaman *detail* berdasarkan *property_id*.

- Card Summary Task

Card ini menampilkan jumlah *task* berstatus *Ongoing* dari tabel *maintenance*. Saat ditekan, pengguna diarahkan ke halaman *maintenance* untuk melihat detail *task* yang aktif.

- Card Summary Network Issues

Card ini menampilkan jumlah *network issues* 'Ongoing' dari tabel *network_issues* dan mengarahkan ke halaman CRUD untuk pengelolaan data.

- Visualisasi Bar Chart

Bar chart ini menampilkan jumlah kawasan per unit, diambil dari tabel unit dan kawasan melalui relasi *unit_id*, dengan tinggi bar menunjukkan total kawasan terdaftar.

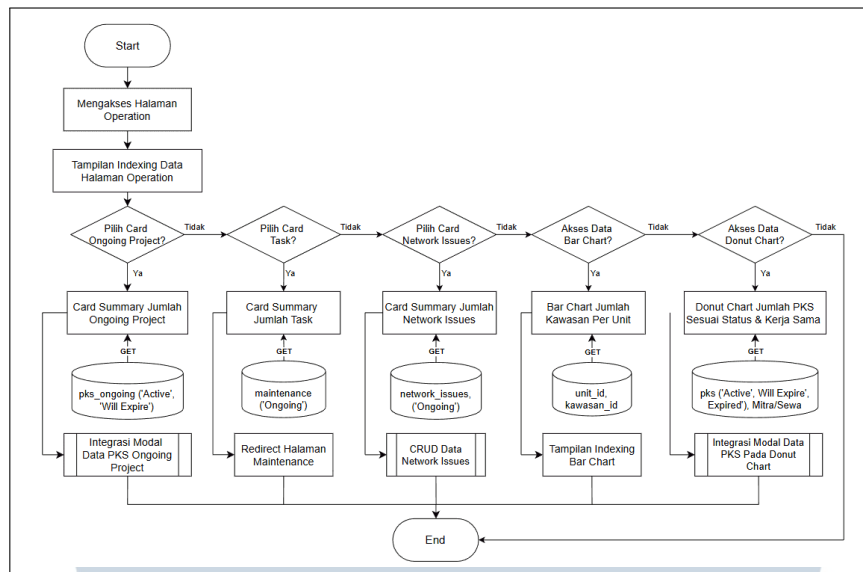
- Visualisasi Donut Chart

Visualisasi ini menampilkan tiga *donut chart* PKS, yaitu *Active*, *Will Expire*, dan *Expired*. Pengguna dapat memilih jenis visualisasi melalui *dropdown* berdasarkan status PKS Mitra atau Sewa. Saat *donut chart* ditekan, sistem menampilkan *modal* tabel sesuai pilihan, dan pengguna dapat menuju halaman *detail* PKS berdasarkan *property_id*.

B Flowchart

- Flowchart Keseluruhan Halaman Operation

Gambar 3.9 berikut memperlihatkan alur keseluruhan halaman *Operation*. Diagram ini menampilkan rangkaian proses serta penyajian informasi dalam tampilan interaktif untuk memudahkan pengguna dalam memahami informasi, termasuk *card summary*, *bar chart*, dan *donut chart*.



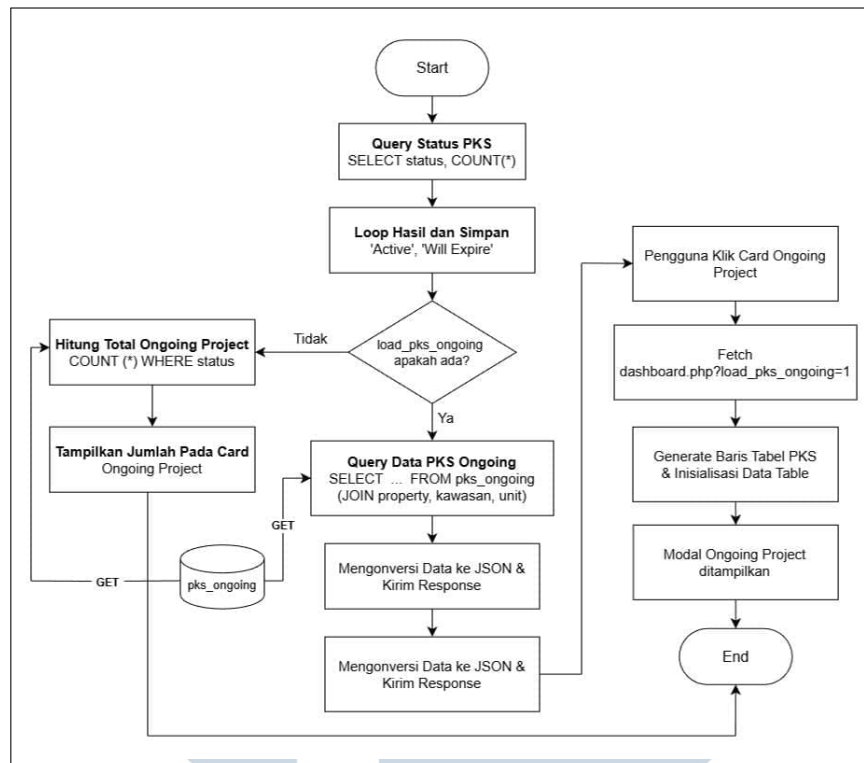
Gambar 3.9. Flowchart Keseluruhan Halaman Operation

Flowchart ini menggambarkan alur saat pengguna membuka halaman *Operation*. Sistem langsung menampilkan ringkasan data dengan mengambil PKS yang berstatus *Active* dan *Will Expire* untuk ditampilkan pada *card summary* dan dapat dilihat detailnya melalui *modal*. Selain itu, sistem juga menampilkan data *maintenance* dengan status *Ongoing* serta menyediakan akses ke halaman *maintenance* untuk melihat atau memperbarui data.

Sistem menampilkan jumlah *network issues Ongoing* dari tabel *network_issues* dengan akses *CRUD*, *bar chart* persebaran kawasan per unit, dan *donut chart* status PKS (*Active*, *Will Expire*, *Expired*) beserta jenis kerja sama, yang dapat dibuka melalui *modal*. Subproses ditampilkan terpisah pada *flowchart*.

- Subproses Flowchart Integrasi Modal DataTable PKS Ongoing Project

Gambar 3.10 menampilkan alur subproses ketika pengguna membuka *card summary* PKS *ongoing project* melalui halaman *Operation*. *Flowchart* ini menjelaskan bagaimana sistem mengambil data dari *server* hingga data tersebut ditampilkan ke dalam *modal* secara lengkap.



Gambar 3.10. Flowchart Integrasi Modal DataTable PKS Ongoing Project

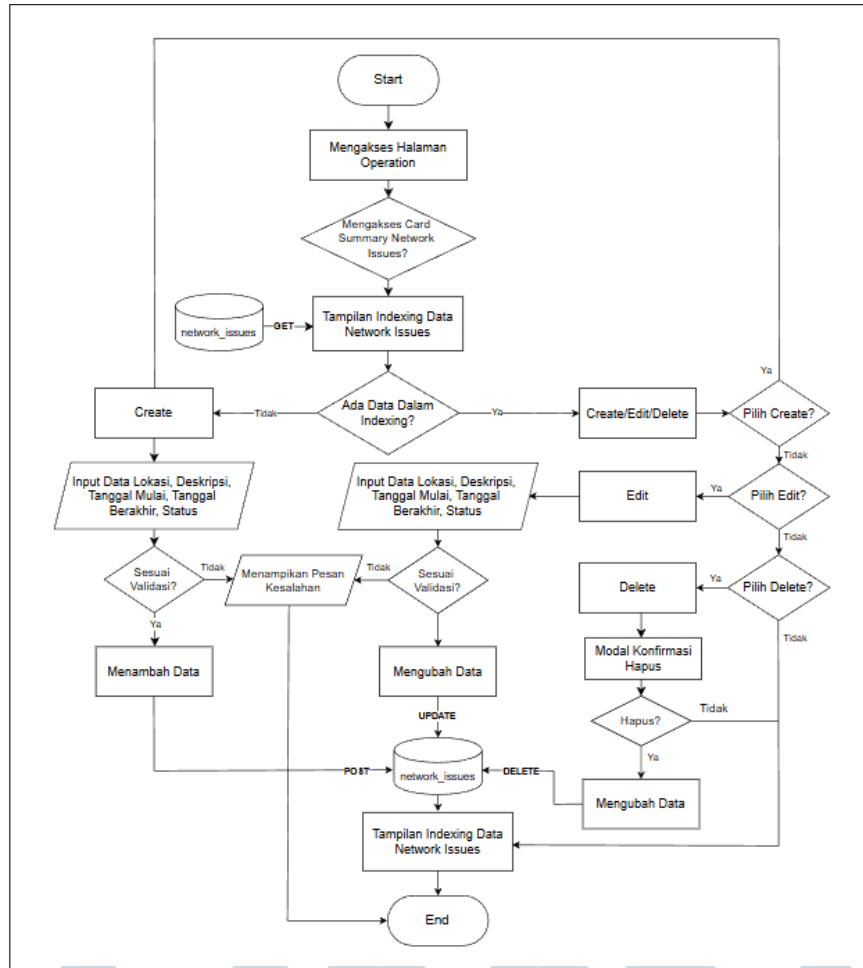
Gambar 3.10 menunjukkan alur sistem menghitung status PKS (*Active*, *Will Expire*) untuk ditampilkan pada *card summary*. Jika *card Ongoing Project* ditekan, sistem mengambil data lengkap PKS dengan *JOIN* ke tabel *property*, *kawasan*, dan *unit*, lalu mengirim hasil dalam format JSON ke *frontend*.

Di sisi pengguna, data JSON diproses untuk membentuk tabel secara dinamis dan ditampilkan menggunakan *DataTable* dengan fitur pencarian dan *pagination*. Setelah itu, *modal Ongoing Project* akan menampilkan data PKS yang masih aktif atau mendekati masa berakhir. *Flowchart* ini menjelaskan alur dari pengambilan data di *server* hingga ditampilkan ke pengguna, dan disajikan terpisah agar alurnya lebih mudah dipahami.

• Subproses Flowchart CRUD Data Network Issues

Flowchart pada Gambar 3.11 menggambarkan proses pengelolaan data *network issues* melalui halaman *Operation*. Setelah pengguna membuka *card Network Issues*, sistem mengambil seluruh data dari tabel *network_issues* dan menampilkannya dalam bentuk tabel. Dari tampilan ini, pengguna dapat melakukan tiga aksi utama, yaitu menambah data (*create*), mengubah data

(edit), dan menghapus data (delete).



Gambar 3.11. Flowchart Proses CRUD Data Network Issues

1. Menambah Data (Create)

Pada proses ini, pengguna mengisi *form* berisi lokasi, deskripsi, tanggal mulai, tanggal berakhir, dan status. Sistem melakukan validasi input terlebih dahulu. Jika valid, data disimpan ke *database* menggunakan perintah berikut.

```
1 INSERT INTO network_issues
2     (deskripsi, lokasi, tanggal_mulai, tanggal_berakhir,
3      status) VALUES ('...', '...', '...', '...', '...');
```

Kode 3.2: Proses Insert Data Network Issues

Jika validasi gagal, sistem menampilkan pesan kesalahan dan pengguna perlu memperbaiki input.

2. Mengubah Data (*Edit*)

Pada proses *edit*, pengguna memilih data yang ingin diperbarui dan mengisi ulang *form*. Setelah lolos validasi, sistem menjalankan perintah:

```
1 UPDATE network_issues
2     SET deskripsi='...', lokasi='...', tanggal_mulai='...'
3     , tanggal_berakhir='...', status='...'
4     WHERE id = ...;
```

Kode 3.3: Proses Update Data Network Issues

Jika proses berhasil, sistem menampilkan notifikasi dan memuat ulang data terbaru.

3. Menghapus Data (*Delete*)

Sebelum data dihapus, sistem menampilkan *modal* konfirmasi. Jika pengguna menyetujui, sistem menjalankan perintah:

```
1 DELETE FROM network_issues
2 WHERE id = ...;
```

Kode 3.4: Proses Update Data Network Issues

Jika dibatalkan, sistem kembali ke tampilan awal tanpa perubahan apapun.

Setelah setiap aksi selesai, halaman akan memuat ulang dan menampilkan data terbaru berdasarkan perintah:

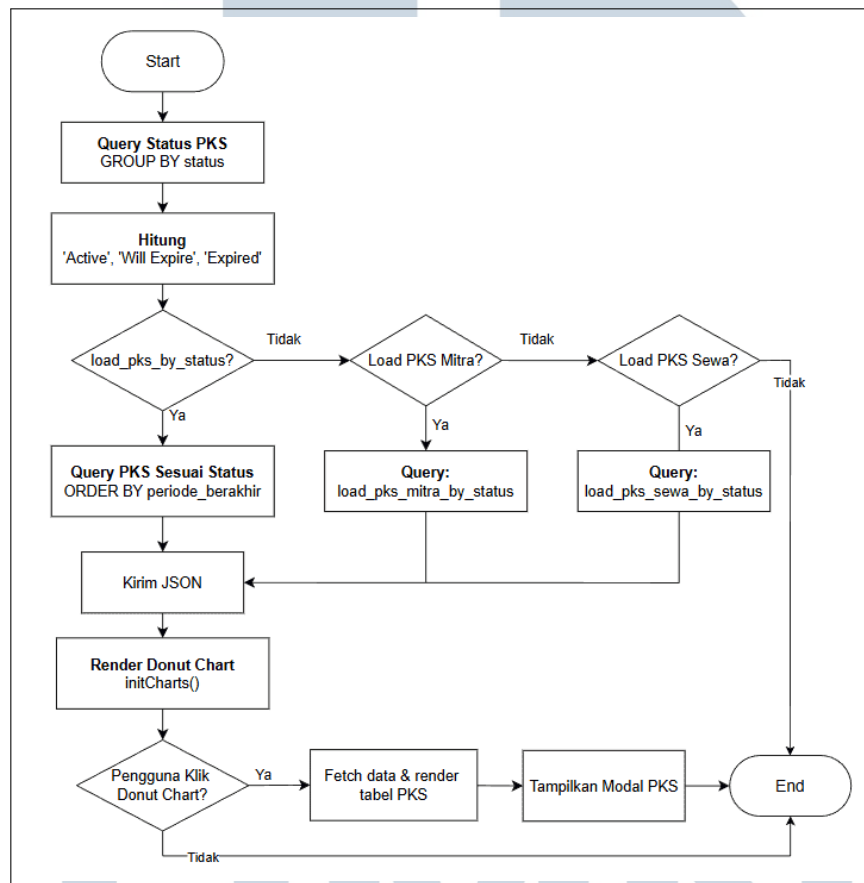
```
1 SELECT * FROM network_issues ORDER BY id DESC;
```

Kode 3.5: Proses Update Data Network Issues

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

- Subproses Flowchart Integrasi Modal DataTable Donut Chart

Gambar 3.12 menunjukkan alur proses pada fitur *donut chart* untuk status PKS dan mengelompokkan berdasarkan kategori *Active*, *Will Expire*, dan *Expired*.



Gambar 3.12. Flowchart Integrasi Modal DataTable Donut Chart

Flowchart ini menggambarkan bagaimana sistem memulai permintaan data ke *server* saat pengguna membuka modal, kemudian menerima *respons* berupa data yang telah difilter, dan menampilkannya dalam bentuk visual grafik. Alur ini memastikan penyajian informasi dapat berlangsung setiap kali *modal* diakses.

1. Mengambil dan Menghitung Status PKS

Sistem menjalankan *query* untuk menghitung jumlah PKS berdasarkan status, lalu menyimpannya sebagai data yang akan ditampilkan pada *chart*.

2. Memuat Data Sesuai Parameter

Jika permintaan data dilakukan melalui parameter *load_pks_by_status*, sistem mengambil data PKS sesuai status dan mengurutkannya berdasarkan periode berakhir. Pada kondisi tertentu, data juga dipisahkan berdasarkan jenis kerja sama, yaitu Mitra atau Sewa.

3. Mengirim Data dalam Format JSON

Setelah data berhasil diproses, sistem mengirimkan *respons* dalam bentuk JSON untuk digunakan pada proses *rendering chart*.

4. Menampilkan Donut Chart

Donut chart ditampilkan melalui fungsi *initCharts()* sebagai visualisasi status PKS agar pengguna bisa melihat kondisi secara cepat.

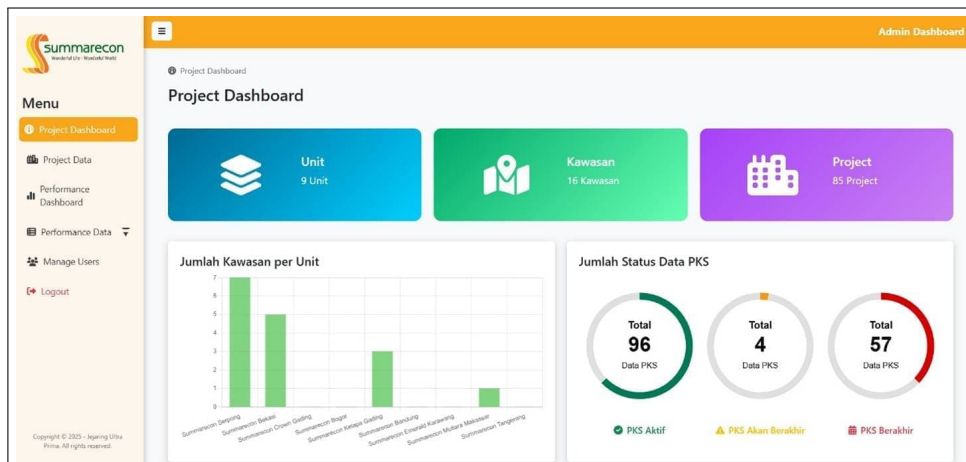
5. Menampilkan Detail Melalui Modal

Ketika pengguna menekan *chart*, sistem melakukan pengambilan data tambahan lalu menampilkan tabel PKS dalam bentuk *modal datatable* untuk melihat informasi lebih lengkap.

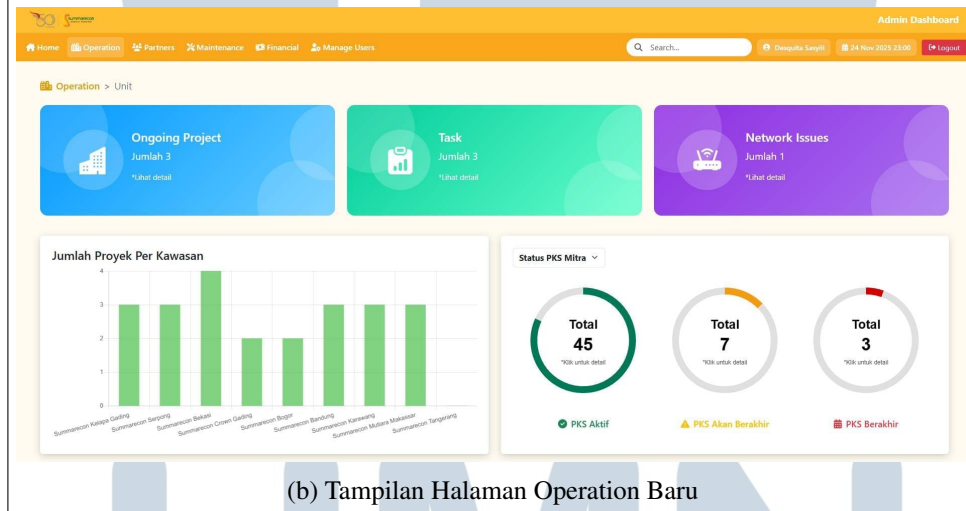
Dengan alur ini, pengguna dapat memahami status PKS secara cepat dan ringkas dari *chart*, kemudian melihat *detail* data hanya dengan satu tekan tanpa perlu berpindah halaman.

C Implementasi

Pengembangan pada halaman *Operation* yang ditampilkan pada Gambar 3.13 merupakan hasil penyempurnaan dari tampilan sebelumnya pada Gambar 3.13a dan tahap pengembangan pada Gambar 3.13b. Pada versi awal, bagian *card summary* hanya menampilkan tiga *card* yang berisi data unit, kawasan, dan proyek.



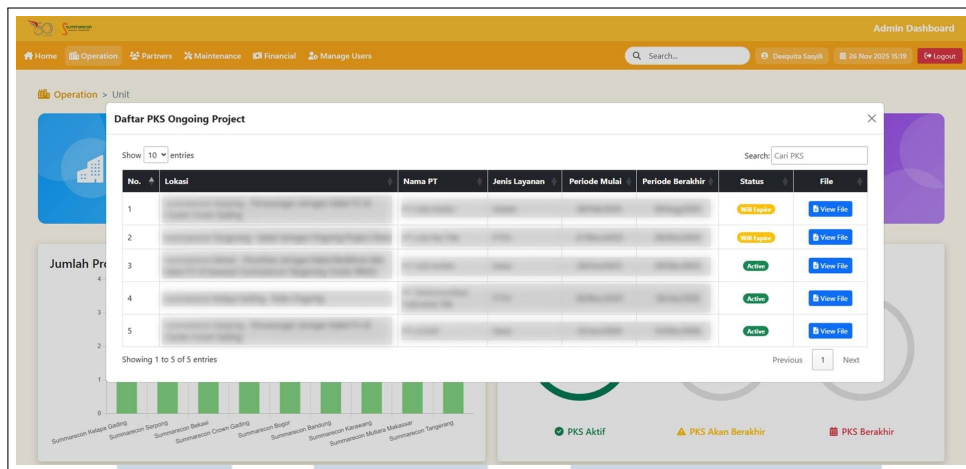
(a) Tampilan Halaman Operation Sebelumnya



(b) Tampilan Halaman Operation Baru

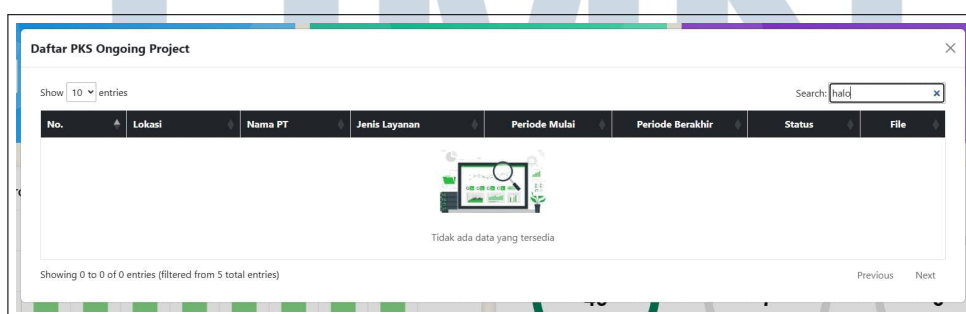
Gambar 3.13. Tampilan Halaman Operation

Pada tahap pengembangan, ketiga *card* diperbarui menjadi *card ongoing project*, *task*, dan *network issues* untuk menampilkan informasi lebih relevan dan *detail*. Visualisasi *donut chart* ditingkatkan dengan pemilihan status ‘Mitra’ dan ‘Sewa’, serta menampilkan jumlah PKS berdasarkan status *Active*, *Will Expire*, dan *Expired*. Panel navigasi vertikal diganti *navbar* horizontal yang memuat menu utama, nama pengguna, tanggal, dan waktu, sehingga tampilan lebih informatif dan mendukung analisis data.



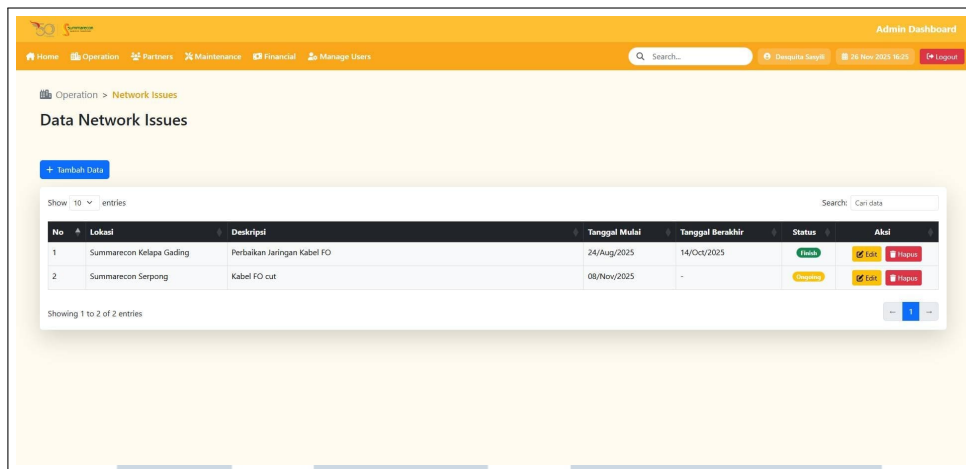
Gambar 3.14. Tampilan Modal DataTable Card Ongoing Project

Gambar 3.14 menampilkan *modal datatable* yang muncul saat *card ongoing project* ditekan. Pada tampilan ini, pengguna dapat melihat data secara lengkap, mulai dari *detail* lokasi, nama PT, jenis layanan, periode mulai hingga berakhir, serta status PKS. Selain itu, disediakan tombol aksi (*view file*) untuk melihat dokumen sesuai dengan data PKS masing-masing. *Modal* ini memiliki opsi jumlah entri (5, 10, 25, atau semua) dengan *default* 10 baris, serta dilengkapi fitur *pagination* dan pencarian untuk memudahkan pengguna dalam menemukan data tertentu. Apabila tidak terdapat data atau *card ongoing project* dalam keadaan kosong, sistem akan menampilkan tampilan seperti pada Gambar 3.15 di bawah ini.



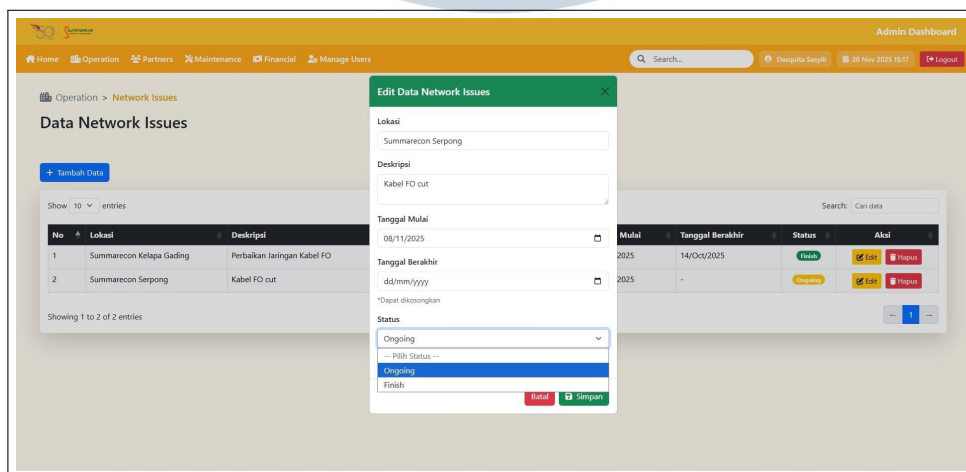
Gambar 3.15. Tampilan Tidak Ada Data Tersedia

Saat *card network issues* ditekan, pengguna akan diarahkan ke halaman data *network issues*. Pada halaman ini, pengguna bisa melakukan operasi CRUD seperti menambah, mengubah, atau menghapus data, seperti yang ditunjukkan pada Gambar 3.16.



Gambar 3.16. Tampilan Data Network Issues

Tampilan pada Gambar 3.16 menampilkan daftar *network issues* dalam *DataTable*, meliputi lokasi, deskripsi, tanggal mulai dan berakhir, serta status *ongoing* atau *finish*. *DataTable* dilengkapi fitur *show entries*, *pagination*, dan pencarian. Tombol biru di kiri atas digunakan untuk menambah data, sedangkan tombol kuning pada kolom aksi untuk mengedit data.

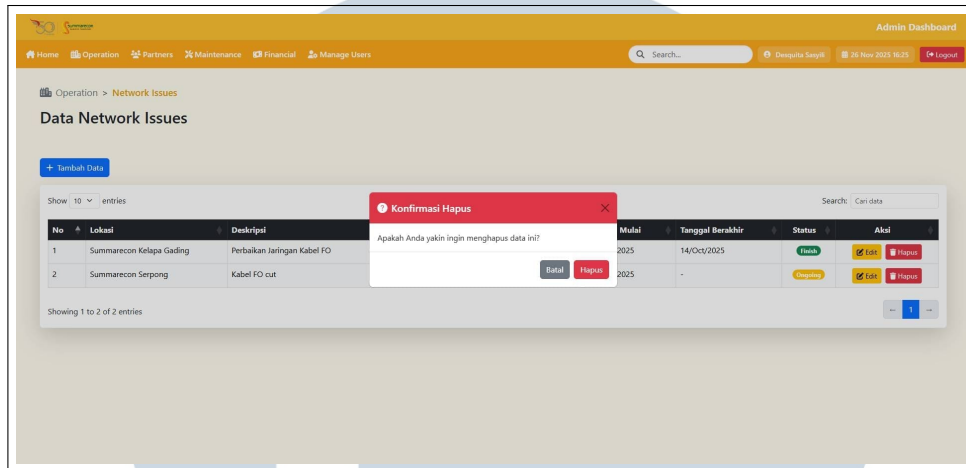


Gambar 3.17. Tampilan Modal Form Network Issues

Pada Gambar 3.17 merupakan tampilan *modal form* yang digunakan untuk menambah atau mengubah data *network issues*. Pada *modal* ini, pengguna dapat mengisi atau memperbarui beberapa *field*, seperti lokasi, deskripsi, tanggal mulai, tanggal berakhir, serta memilih status apakah masih *ongoing* atau sudah *finish* melalui *dropdown*.

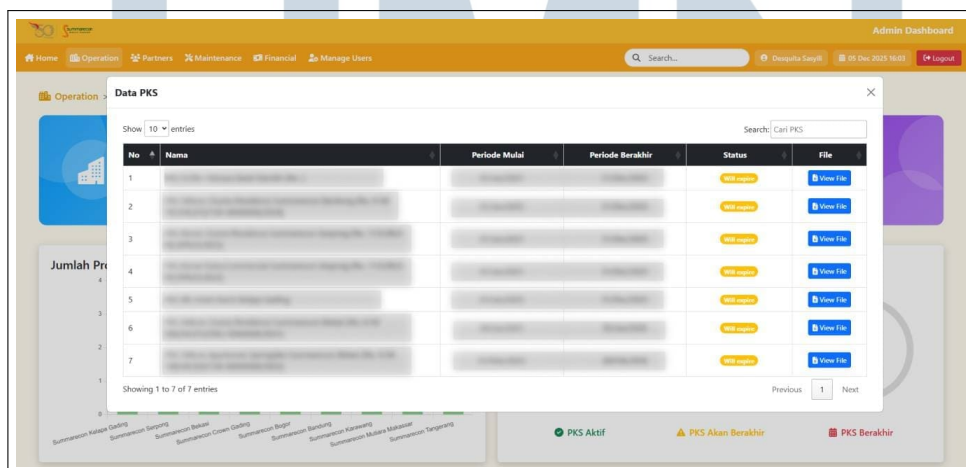
Untuk *field* tanggal berakhir, bertipe *NULL* sehingga dapat dikosongkan jika

masalah jaringan tersebut masih dalam proses penanganan. *Field* ini dapat diisi ketika proses pemeliharaan sudah selesai dan statusnya berubah menjadi *finish*.



Gambar 3.18. Tampilan Modal Konfirmasi Hapus

Pada Gambar 3.18 ditampilkan *modal* konfirmasi yang muncul ketika pengguna akan menghapus data *network issues*. Untuk menghapus data, dapat menekan tombol hapus berwarna merah pada kolom aksi di *DataTable*, seperti terlihat pada Gambar 3.16. Setelah tombol tersebut ditekan, sistem akan menampilkan *modal* konfirmasi seperti pada Gambar 3.18 sebagai proses memastikan apakah data tersebut benar-benar akan dihapus.



Gambar 3.19. Tampilan Modal PKS Pada Donut Chart

Pada Gambar 3.19 merupakan tampilan *modal datatable* ketika *donut chart* ditekan. *Modal* yang muncul akan menyesuaikan dengan jenis *donut chart* yang

dipilih, yaitu *Active*, *Will Expire*, atau *Expired*. Status *badge* yang tampil pada kolom juga akan mengikuti jenis *donut chart* yang dipilih.

3.4.3 Penambahan Field Email dan Integrasi Kirim Email Blast

A Analisis Kebutuhan

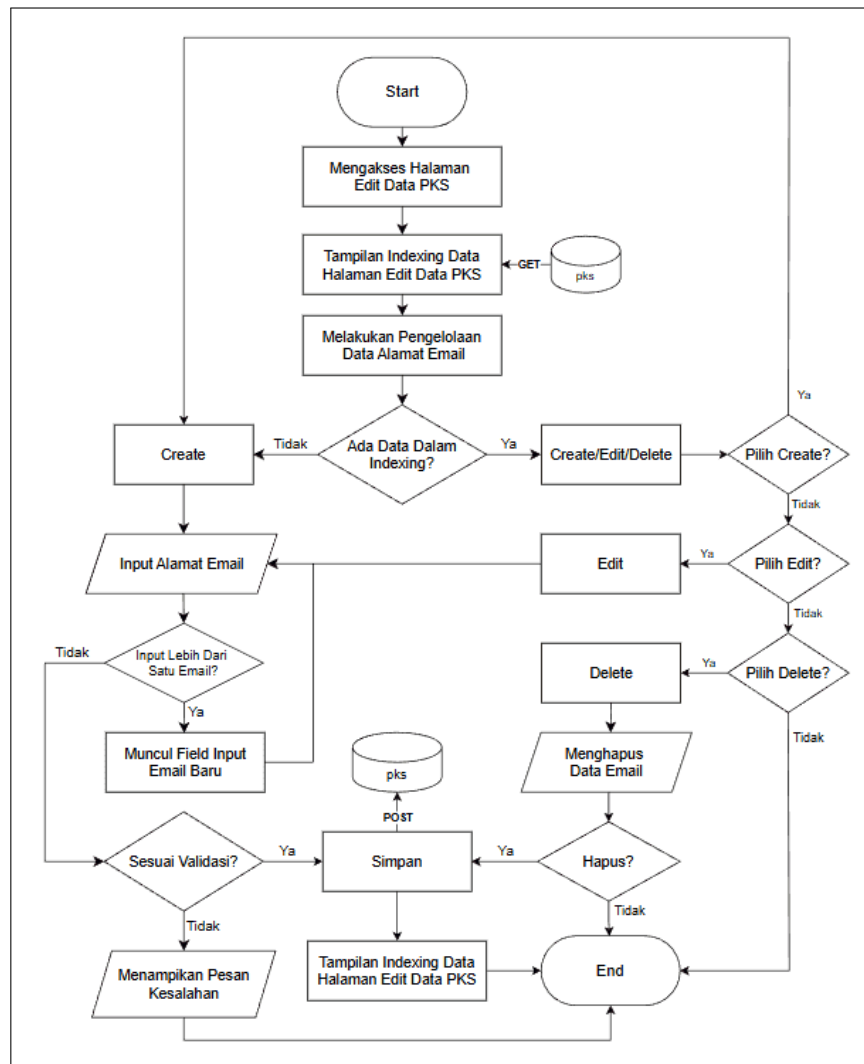
Pada halaman *project data* yang digunakan untuk menyimpan data-data PKS di dalam sistem, melakukan fitur baru yaitu daftar *email* mitra, di mana admin dapat memasukkan lebih dari satu alamat *email*. Setelah *email-email* tersebut tersimpan, admin dapat mengirim *email blast* ke seluruh alamat yang terdaftar. Jika ada *email* yang tidak ingin ikut dikirim *email blast* tersebut, admin dapat menghapus centang melalui tampilan *checkbox*.

Selain itu, tersedia dua jenis *email blast*, yaitu "*Reminder PKS*" dan "*Reminder Revenue Sharing*", masing-masing dengan *template* pesan yang berbeda sesuai kebutuhan. Sebelum melakukan pengiriman, admin harus memastikan bahwa *field email* mitra sudah terisi. Jika ingin menambahkan *email* baru, admin dapat memasukkannya melalui *field email* yang sudah disediakan.

B Flowchart

- Flowchart Pengelolaan Alamat Email Mitra

Gambar 3.20 menunjukkan alur proses saat pengguna mengelola alamat *email* pada halaman Edit Data PKS. Alur dimulai ketika pengguna membuka halaman tersebut. Pada tahap awal, sistem mengambil data *email* dari basis data dan menampilkannya sebagai daftar *indexing email*. Setelah data ditampilkan, sistem melakukan pengecekan apakah sudah ada *email* yang tersimpan. Jika belum ada, pengguna dapat melakukan proses pembuatan data baru dengan mengisi alamat *email*.



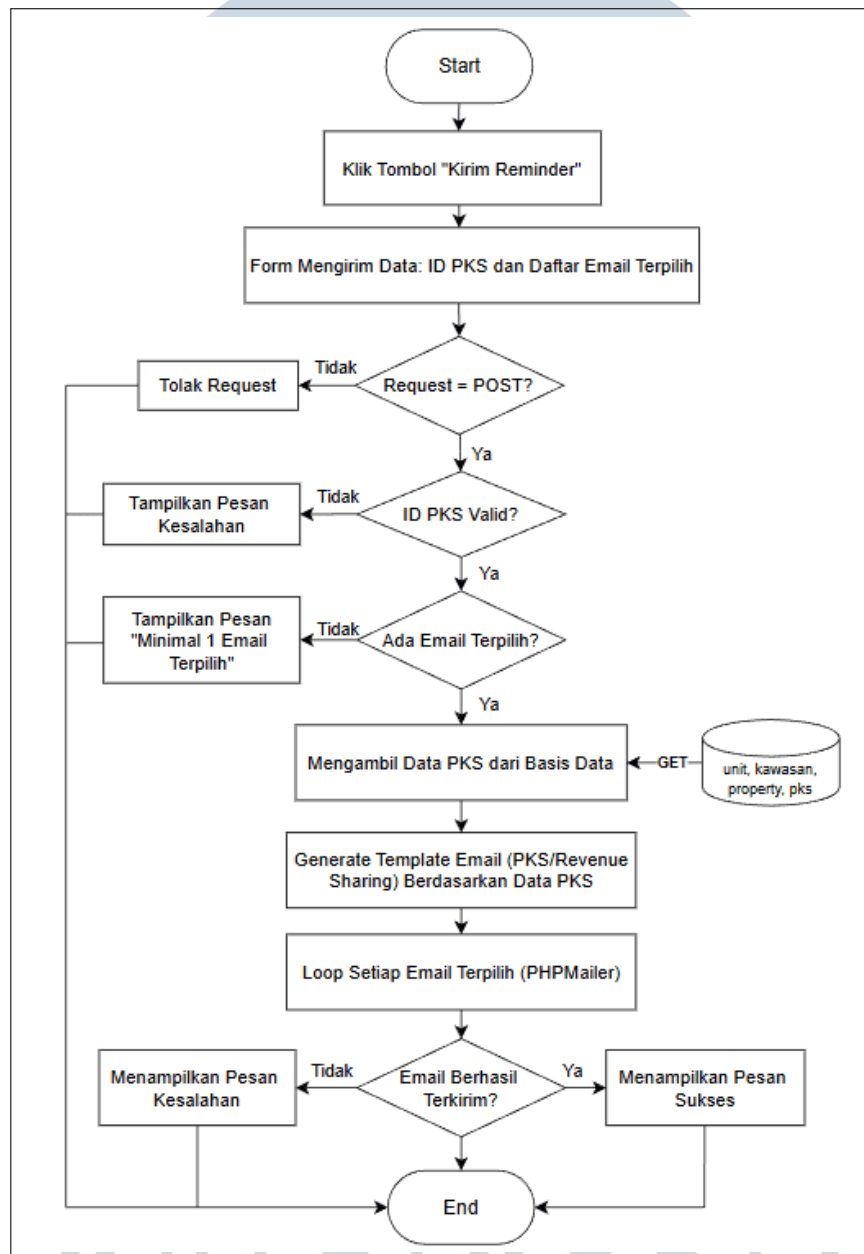
Gambar 3.20. Flowchart Pengelolaan Alamat Email Mitra

Pengguna dapat menambahkan lebih dari satu *email* melalui tombol tambah *email*. Setelah semua *email* diisi, sistem melakukan validasi dan menyimpan data jika valid, atau menampilkan pesan kesalahan jika terdapat input yang tidak sesuai. Untuk data *email* yang sudah ada, pengguna dapat melakukan *create*, *edit*, atau *delete*. Setelah proses selesai, sistem akan kembali ke halaman *indexing* dan menampilkan data *email* terbaru.

- Flowchart Integrasi Pengiriman Email Blast Reminder

Gambar 3.21 menampilkan alur pengiriman *email blast* PKS atau *Revenue Sharing* saat admin menekan tombol *Kirim Reminder*. Proses dimulai dari tombol ditekan hingga *email* terkirim, meliputi pengiriman

ID PKS dan daftar *email*, pengecekan metode POST, validasi ID PKS, dan memastikan minimal satu *email* dipilih. Jika pengecekan gagal, sistem menampilkan pesan kesalahan.



Gambar 3.21. Flowchart Integrasi Pengiriman Email Blast Reminder

Setelah semua pengecekan sudah aman, sistem akan mengambil data PKS dari basis data. Data ini diambil dengan melakukan *query* ke tabel-tabel terkait seperti *unit*, *kawasan*, *property*, dan *pks*. Pada saat data PKS berhasil diambil, sistem membuat *template email* untuk *Reminder* PKS atau

Revenue Sharing berdasarkan data yang sudah diambil. *Template* ini berisi informasi penting yang akan dikirimkan ke penerima.

Sistem kemudian melakukan *looping* untuk mengirim setiap *email* menggunakan PHPMailer, mengecek keberhasilan pengiriman. Jika ada *email* gagal, muncul pesan kesalahan; jika semua berhasil, sistem menampilkan pesan sukses. Alur ini memastikan *email reminder* terkirim dengan benar dan meminimalkan kesalahan.

C Perancangan Basis Data

Tabel 3.3 berikut menampilkan struktur basis data pks yang digunakan sebagai penyimpanan data kerja sama (PKS) pada sistem.

Tabel 3.3. Struktur Tabel PKS

Kolom	Tipe Data	NULL	Keterangan
id	INT (11)	Tidak	Primary key
property_id	INT (11)	Tidak	Relasi properti
nama	VARCHAR (255)	Tidak	Nama PKS
nama_pt	VARCHAR (255)	Tidak	Nama perusahaan
jenis_layanan	ENUM	Tidak	Jenis layanan mitra
revenue_sharing	VARCHAR (255)	Tidak	Skema bagi hasil
homepass	INT (11)	Ya	Default 0
revenue	BIGINT (20)	Ya	Default 0
periode_mulai	DATE	Tidak	Periode mulai
periode_berakhir	DATE	Tidak	Periode akhir
bast	DATE	Ya	Tanggal BAST
status	ENUM	Ya	Status PKS
file_path	VARCHAR (255)	Tidak	Lokasi file
emails	TEXT	Ya	Email mitra

Pada Tabel 3.3 digunakan untuk menyimpan seluruh data Perjanjian Kerja Sama (PKS) dalam sistem. Kolom *id* berfungsi sebagai *primary key* dan menjadi identitas unik setiap PKS. Kolom *property_id* menyimpan relasi ke data properti

terkait, sedangkan kolom `nama`, `nama_pt`, dan `jenis_layanan` mencatat informasi dasar mengenai nama PKS, perusahaan mitra, dan jenis layanan yang diberikan.

Tabel ini menyimpan informasi tambahan seperti skema bagi hasil (`revenue_sharing`), jumlah `homepass`, dan nilai `revenue`. Masa berlaku PKS dicatat melalui kolom `periode_mulai`, `periode_berakhir`, dan `bast`, sedangkan status PKS disimpan pada kolom `status`. Selain itu, kolom `file_path` digunakan untuk menyimpan lokasi dokumen PKS dan `emails` untuk menyimpan *email* mitra, sehingga pengelolaan data kerja sama dapat dilakukan dengan rapi dan mudah diakses.

D Implementasi

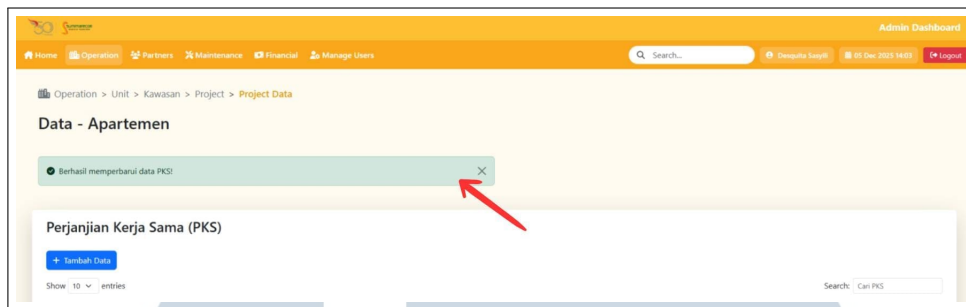
Penambahan *field* alamat *email* pada data PKS dibuat sebagai bagian dari sistem *Email Blast* untuk mempermudah pengelolaan dan pengiriman *email* massal ke mitra kerja sama. Tujuan pengembangan ini agar proses komunikasi jadi lebih fleksibel, personal, dan terhubung langsung dengan data PKS yang sudah ada. Fitur ini juga dirancang dengan tampilan yang mudah digunakan, pilihan data penerima yang lebih fleksibel, dan kemampuan untuk menyesuaikan isi *email* sesuai kebutuhan.



The screenshot shows the 'Edit Perjanjian Kerja Sama (PKS)' form. The 'Alamat Email Mitra' section is highlighted with a red box. It contains a list of email addresses: sasyilikheylya@gmail.com, desquitasasyili@gmail.com, and example@gmail.com. There are buttons for '+ Tambah Email' and 'Simpan Perubahan'. A red arrow points to the 'Simpan Perubahan' button. Below the form are two reminder sections: 'Reminder PKS' and 'Reminder Revenue Sharing'.

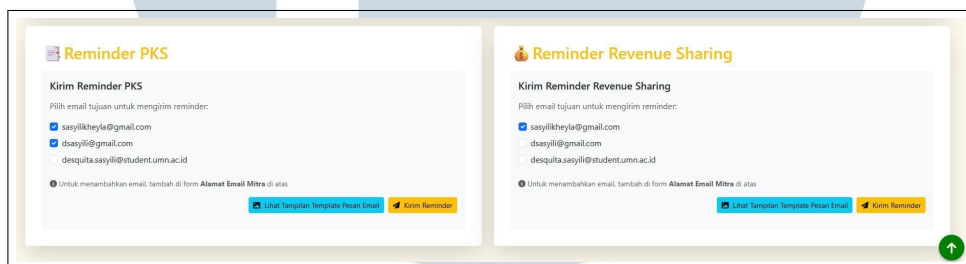
Gambar 3.22. Tampilan Halaman Data PKS Penambahan Field Alamat Email

Gambar 3.22 menampilkan halaman data PKS. Admin dapat menambahkan *email* baru dengan tombol *Tambah Email*, mengedit isi *field*, atau menghapus *email* dengan tombol *close* (x). Setelah selesai, tekan *Simpan Perubahan* untuk menyimpan data, dan sistem menampilkan pesan sukses seperti pada Gambar 3.23.



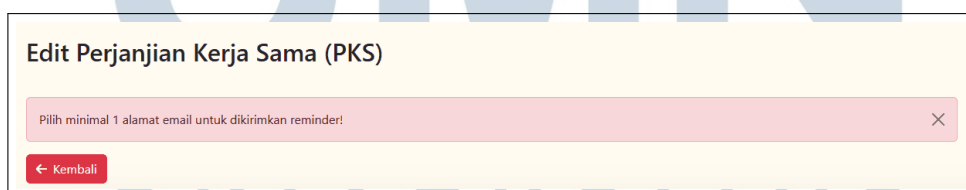
Gambar 3.23. Tampilan Pesan Sukses Data PKS

Saat melakukan pengiriman *email blast reminder* PKS maupun *Revenue Sharing*, admin dapat memilih alamat *email* mana saja yang akan dikirim pesan dengan mencentang *checkbox* yang tersedia, seperti terlihat pada Gambar 3.24.



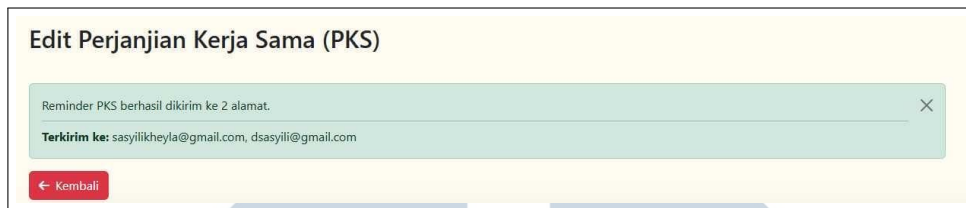
Gambar 3.24. Tampilan Pilih Alamat Email Akan Kirim Email Blast

Sesuai dengan analisis kebutuhan, saat mengirim *email blast reminder* harus terdapat setidaknya satu alamat *email* yang dipilih. Jika admin mencoba mengirim *email blast* tanpa mencentang alamat *email* satu pun pada *checkbox*, maka sistem akan menampilkan pesan gagal mengirim seperti pada Gambar 3.25.



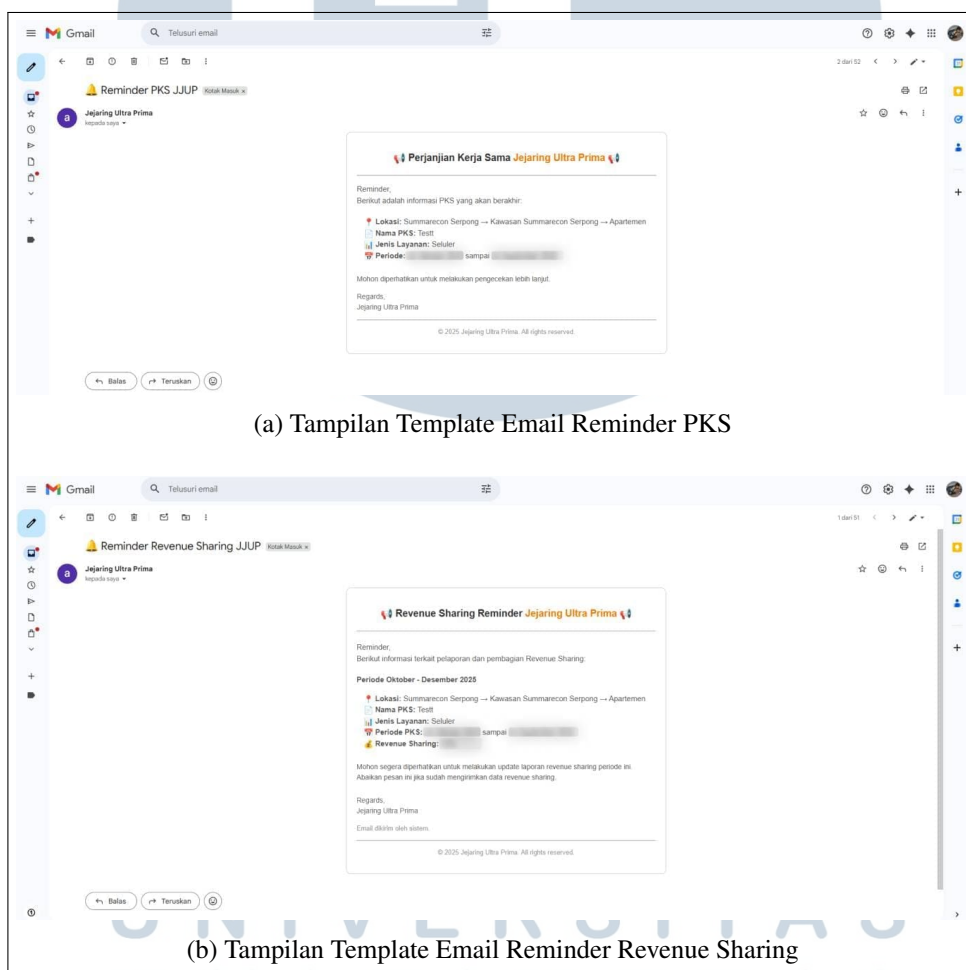
Gambar 3.25. Tampilan Pesan Gagal Kirim Email Blast

Pada saat melakukan pengiriman *email blast* berhasil dilakukan tanpa adanya kesalahan maka akan menampilkan pesan sukses seperti pada Gambar 3.26



Gambar 3.26. Tampilan Pesan Sukses Kirim Email Blast

Tampilan *template* pesan *email blast* untuk *reminder* PKS dan *Revenue Sharing* memiliki tampilan yang berbeda. Tampilannya seperti pada Gambar 3.27



(a) Tampilan Template Email Reminder PKS

(b) Tampilan Template Email Reminder Revenue Sharing

Gambar 3.27. Tampilan Template Email Blast Reminder

Data yang ditampilkan pada *template email blast* mencakup informasi seperti lokasi, nama PKS, jenis layanan, serta periode mulai hingga berakhir. Seluruh informasi tersebut diambil dari relasi antar tabel di basis data sehingga data yang masuk tetap konsisten dan akurat. Penyesuaian data juga dilakukan sesuai

permintaan dan kebutuhan perusahaan agar informasi yang dikirim benar-benar relevan. Dengan penyajian data yang lebih lengkap dan tepat, proses pengingat dapat berjalan lebih efektif dan membantu pihak terkait dalam memantau masa berlaku kerja sama dengan lebih mudah.

3.4.4 Pengembangan Halaman Partners

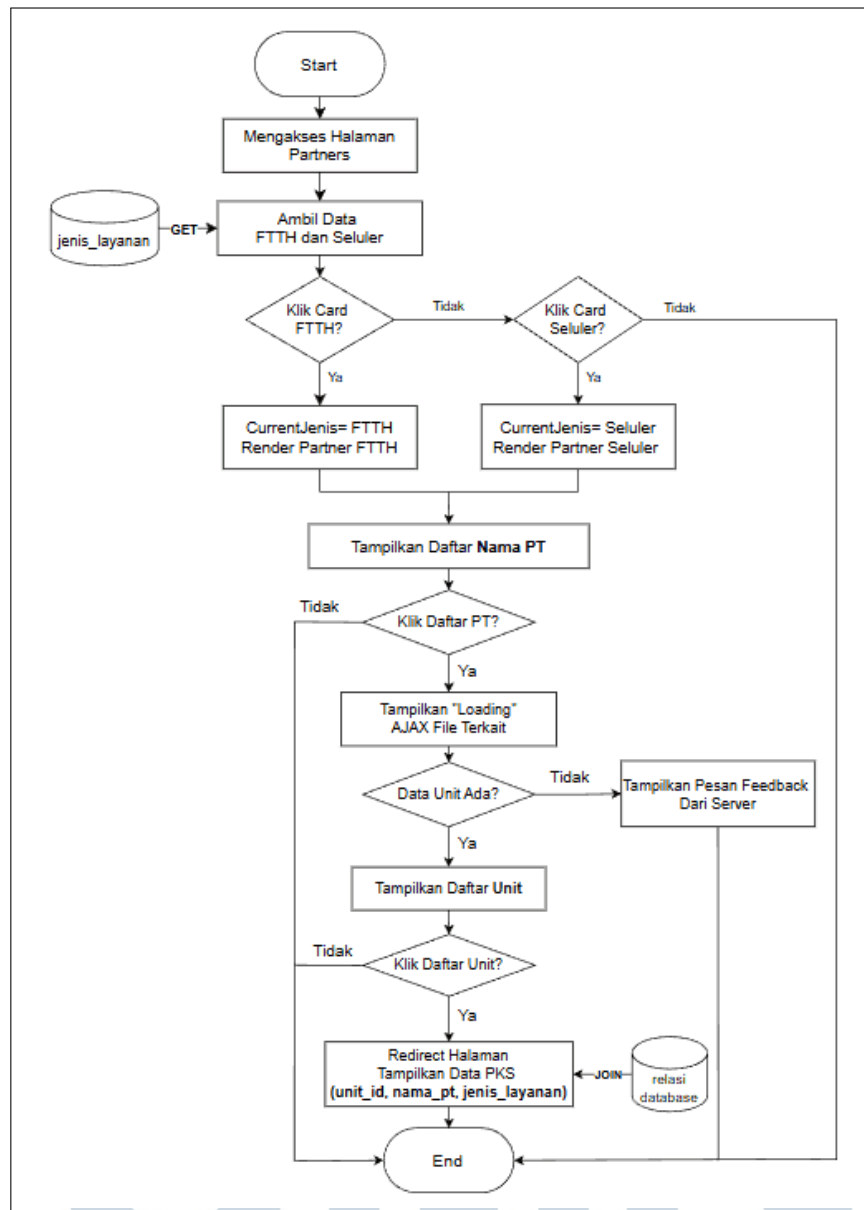
A Analisis Kebutuhan

Halaman *partners* digunakan untuk menampilkan daftar mitra yang terdaftar. Data mitra dikelompokkan berdasarkan jenis layanan, yaitu FTTH dan Seluler, dan ditampilkan dalam bentuk *card*. Saat salah satu *card* dipilih, sistem menampilkan daftar mitra sesuai layanan, dan setiap mitra dapat dipilih untuk melihat unit yang terkait. Selain itu, daftar nama unit yang muncul dari mitra tersebut juga dapat ditekan. Saat unit dipilih, akan *redirect* halaman dan menampilkan data PKS yang berkaitan dengan unit dan mitra yang dipilih. Proses ini mengambil data berdasarkan `unit_id`, `nama_pt`, dan `jenis_layanan`. Data yang ditampilkan pada halaman ini diambil dari relasi tabel PKS di basis data, dengan memanfaatkan kolom `jenis_layanan` dan `nama_pt` sesuai struktur pada Tabel 3.3.

B Flowchart

Gambar 3.28 menunjukkan alur kerja pada halaman *Partners*, mulai dari mengakses halaman hingga data PKS ditampilkan berdasarkan unit dan mitra yang dipilih. *Flowchart* ini membantu menjelaskan bagaimana sistem menampilkan daftar mitra berdasarkan jenis layanan, lalu menelusuri data unit hingga akhirnya menampilkan informasi PKS yang relevan.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.28. Flowchart Integrasi Halaman Partners

Proses dimulai saat pengguna membuka halaman Partners. Sistem menampilkan data mitra berdasarkan layanan FTTH dan Seluler dalam bentuk card. Ketika salah satu card dipilih, sistem menyimpan jenis layanan tersebut dan menampilkan daftar mitra yang dapat dipilih oleh pengguna.

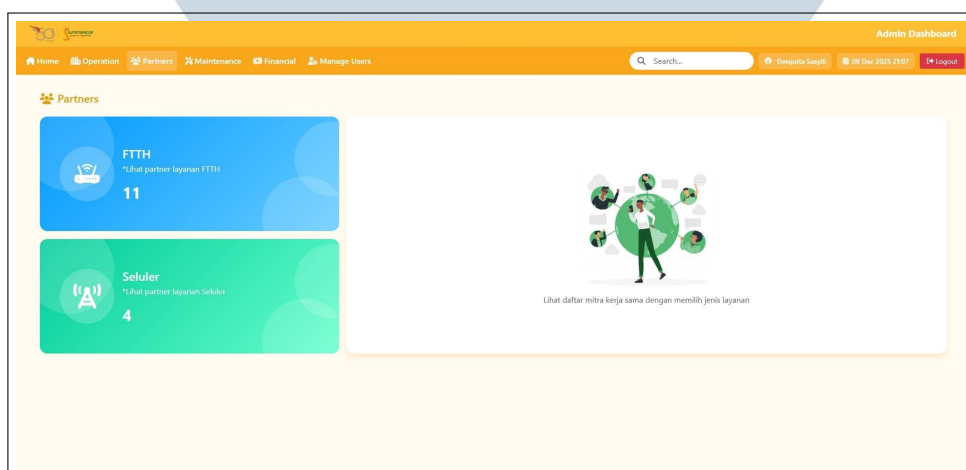
Jika pengguna menekan salah satu nama mitra, sistem akan menampilkan indikator “Loading” dan melakukan *request* menggunakan AJAX untuk mengambil data unit yang terkait dengan mitra tersebut. Setelah itu, sistem mengecek apakah data unit tersedia. Jika tidak ada unit yang ditemukan, sistem akan menampilkan

pesan *feedback* sebagai respon dari *server*.

Jika data unit tersedia, sistem menampilkan daftar unit milik mitra untuk dipilih. Saat unit dipilih, sistem akan melakukan *redirect* ke halaman PKS berdasarkan *unit_id*, *nama_pt*, dan *jenis_layanan* melalui proses *JOIN* tabel. Alur ini menyajikan data secara terstruktur sehingga memudahkan pengguna menelusuri informasi kerja sama.

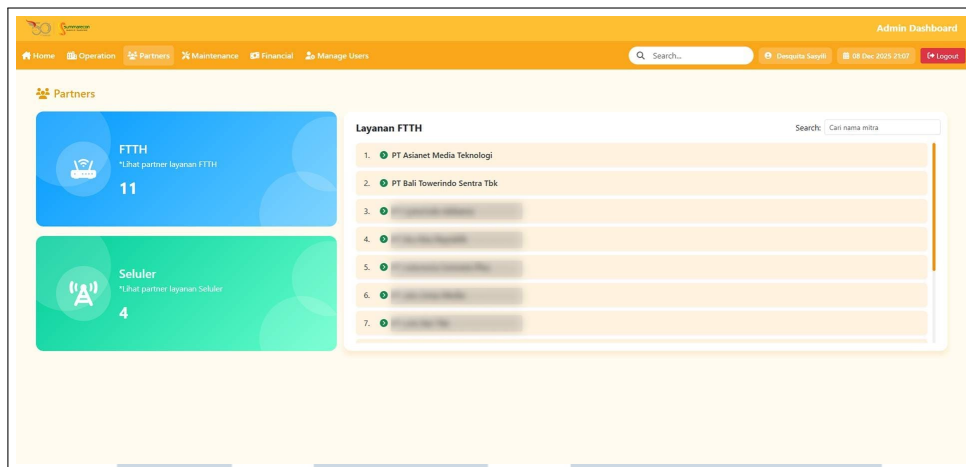
C Implementasi

Pengembangan pada halaman *partners* menjadi salah satu bagian dari proses pengembangan tampilan *dashboard* agar data terlihat lebih terstruktur dan dikelompokkan dengan rapi. Dengan pengelompokan ini, pengguna dapat lebih mudah mencari data yang saling berkaitan tanpa harus mencari satu per satu. Saat halaman *partners* diakses, tampilan awalnya akan terlihat seperti pada Gambar 3.29.



Gambar 3.29. Tampilan Utama Halaman Partners

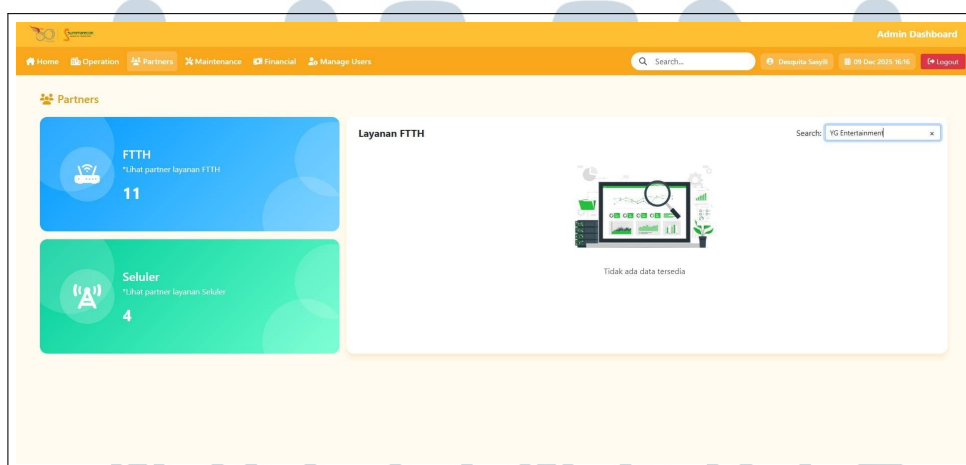
Gambar di atas menampilkan dua *card* yang mewakili masing-masing jenis layanan, yaitu FTTH dan Seluler. Setiap *card* memuat informasi mengenai total jumlah mitra yang termasuk dalam kategori layanan tersebut. Sementara itu, *card* berwarna putih di bagian kanan dengan tampilan aset gambar merupakan tampilan *default*. Nantinya, daftar nama mitra akan ditampilkan pada *card* tersebut setelah pengguna memilih jenis layanan yang ingin ditampilkan. Pada Gambar 3.30 ditampilkan daftar nama mitra ketika salah satu *card* jenis layanan dipilih.



Gambar 3.30. Tampilan Daftar Nama Mitra Berdasarkan Jenis Layanan

Daftar nama mitra tersebut ditampilkan dalam bentuk *list* yang dapat di *scroll*. Untuk memudahkan pengguna mengetahui layanan apa yang sedang dibuka, terdapat informasi di bagian atas *list*, misalnya jika pengguna memilih layanan FTTH maka teks yang muncul adalah “Layanan FTTH”, dan jika memilih layanan Seluler maka teksnya menjadi “Layanan Seluler”.

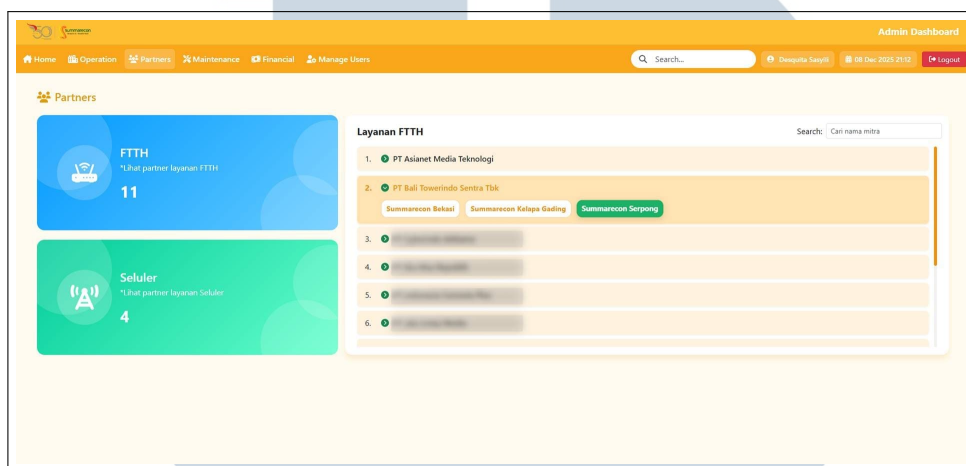
Selain itu, tersedia fitur *live search* yang berada di sisi kanan atas untuk memudahkan pencarian nama mitra tertentu. Jika hasil pencarian tidak menemukan data apa pun, tampilan yang muncul akan terlihat seperti pada Gambar 3.31.



Gambar 3.31. Tampilan Hasil Live Search Tidak Ada Data Mitra

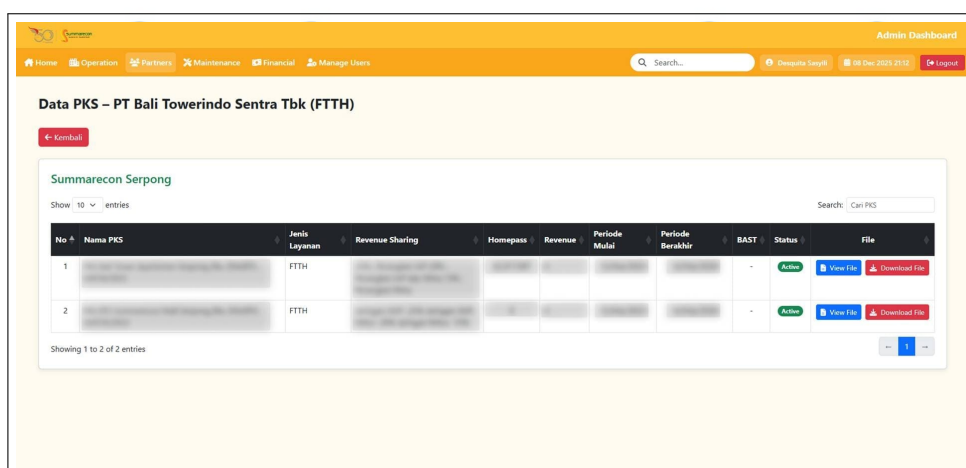
Pada Gambar 3.32 di bawah ditunjukkan tampilan saat pengguna menekan salah satu nama mitra yang terdaftar. Setelah ditekan, *server* akan menampilkan daftar unit yang terkait dengan mitra tersebut. Pada sisi *backend*, proses ini

dilakukan menggunakan JOIN antara tabel unit, kawasan, dan property untuk mendapatkan data unit_id yang sesuai. Daftar unit yang ditampilkan juga dapat ditekan, saat kursor diarahkan ke salah satu unit, tampilannya berubah dengan efek *hover* berwarna hijau.



Gambar 3.32. Tampilan Daftar Unit Berdasarkan Nama Mitra

Gambar 3.33 di bawah ini menunjukkan tampilan halaman yang muncul setelah pengguna menekan salah satu daftar unit dari mitra yang dipilih. Pada halaman ini, data PKS ditampilkan dalam bentuk *datatable*.



Gambar 3.33. Redirect Halaman Menampilkan Data PKS Berdasarkan Unit, Nama Mitra, dan Jenis Layanan

Pada Gambar 3.33, pengguna dapat melihat informasi mengenai data apa yang sedang dibuka melalui judul halaman, yaitu terdapat nama mitra, jenis layanan, dan nama unit yang dipilih. Informasi tersebut ditampilkan karena proses pengambilan

datanya menggunakan tiga parameter utama hasil JOIN, yaitu *unit_id*, *nama_pt*, dan *jenis_layanan*.

Data ditampilkan dalam *datatable* yang memuat informasi PKS, jenis layanan, *revenue sharing*, *homepass*, *revenue*, periode, BAST, status (*Active*, *Will Expire*, *Expired*), dan *file*. Tersedia fitur unduh/lihat *file*, jumlah entri, *pagination*, dan pencarian untuk memudahkan penelusuran data.

3.4.5 Pengembangan Halaman Maintenance

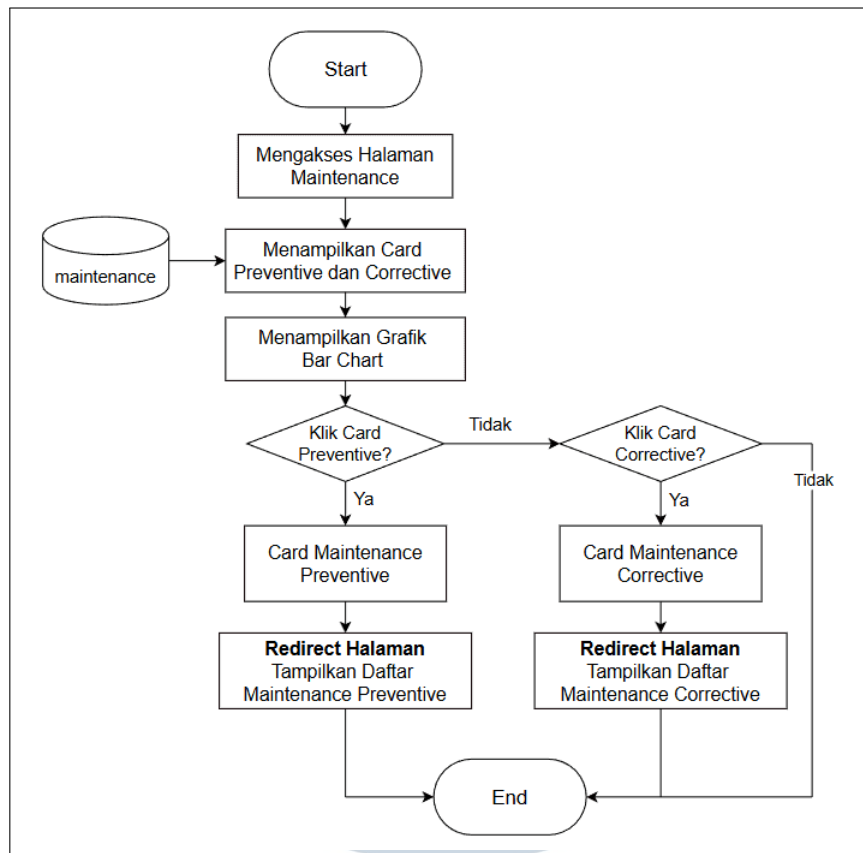
A Analisis Kebutuhan

Pada proses pengembangan halaman *maintenance*, halaman ini berfungsi menampilkan dan menyimpan data aktivitas *maintenance* untuk permasalahan jaringan. Tampilan awal menampilkan *dashboard* berupa *summary card* untuk jenis *Preventive* dan *Corrective*, serta *bar chart* jumlah kegiatan berdasarkan *unit_id*. Menyediakan *breadcrumb* sebagai navigasi ke halaman pengelolaan data. Pengguna memilih unit terlebih dahulu, kemudian diarahkan ke halaman CRUD untuk melakukan pengelolaan data *maintenance*.

Data *maintenance* yang sudah tersimpan akan ditampilkan dalam bentuk *datatable* dengan fitur *filter* berdasarkan bulan dan tahun, opsi jumlah data (*show entries*), pencarian, serta *pagination*. Terdapat juga *badge* yang menampilkan total biaya, yaitu penjumlahan seluruh nilai biaya pada data yang terdaftar. Selain itu, data *maintenance* dapat diekspor dalam format Excel (.xlsx) dan PDF (.pdf), sehingga memudahkan pengguna dalam pengolahan atau pelaporan data.

B Flowchart

Gambar 3.34 menunjukkan alur proses saat pengguna mengakses halaman *maintenance*.

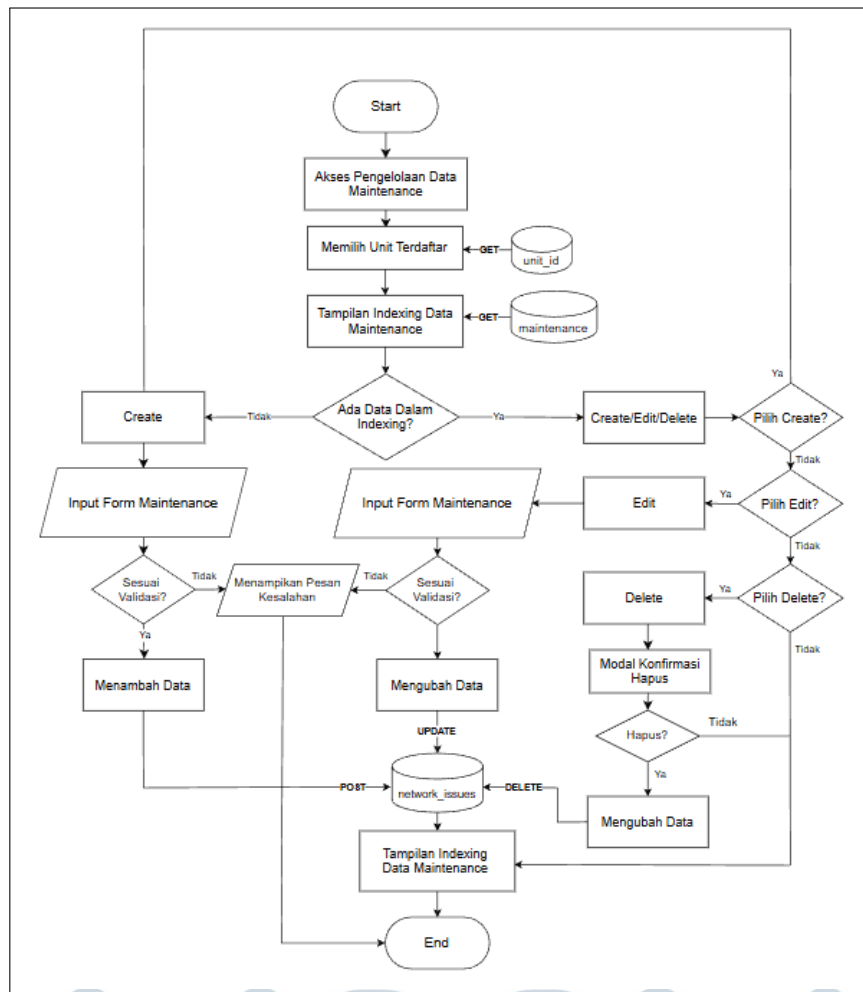


Gambar 3.34. Flowchart Akses Halaman Maintenance

Alur halaman *maintenance* (Gambar 3.34) dimulai saat pengguna mengakses halaman. Sistem menampilkan dua *card* (*Preventive* dan *Corrective*) serta *bar chart* jumlah kegiatan *maintenance* per unit.

Selanjutnya, pengguna dapat memilih salah satu *card* yang ditampilkan. Jika pengguna menekan *card Preventive*, sistem akan menampilkan halaman daftar *maintenance* dengan filter khusus jenis *Preventive*. Sebaliknya, apabila *card Corrective* yang dipilih, pengguna akan diarahkan ke halaman daftar *maintenance* untuk jenis *Corrective*. Pada tahap ini sistem melakukan proses *redirect* menuju halaman daftar yang sesuai. Alur kemudian berakhir setelah daftar data *maintenance* tampil pada halaman yang dipilih.

Flowchart pada Gambar 3.35 menggambarkan proses pengelolaan data *maintenance*. Saat pengguna berada di halaman ini dapat melakukan tambah data (*create*), mengubah data (*edit*), menghapus data (*delete*) ataupun melihat detail data *maintenance*.



Gambar 3.35. Flowchart CRUD Data Maintenance

Alur kerja halaman pengelolaan data pada Gambar 3.35 dimulai saat pengguna memilih unit dan akan mengambil relasi `unit_id`, kemudian sistem menampilkan daftar maintenance dari tabel `maintenance`. Jika data belum ada, pengguna dapat menambahkan (*create*) melalui formulir dengan validasi sebelum disimpan. Data yang tersedia dapat dilihat (*detail*), diubah (*edit*), atau dihapus (*delete*) melalui formulir dan konfirmasi, dengan semua perubahan tersimpan langsung ke basis data. Seluruh proses *create*, *update*, dan *delete* dilakukan langsung pada basis data, sehingga memastikan bahwa setiap perubahan tercatat dengan baik.

C Perancangan Basis Data

Tabel 3.4 berikut menampilkan struktur basis data maintenance yang digunakan sebagai penyimpanan data maintenance pada sistem.

Tabel 3.4. Struktur Tabel Maintenance

Kolom	Type Data	NULL	Keterangan
id	INT (11)	Tidak	Primary key
unit_id	INT (11)	Tidak	Relasi ID unit
property_id	INT (11)	Tidak	Relasi ID properti
proyek_nama	VARCHAR (255)	Tidak	Nama proyek maintenance
tanggal	DATE	Tidak	Tanggal pelaksanaan
alamat	TEXT	Tidak	Lokasi maintenance
issue	TEXT	Tidak	Masalah yang ditangani
waktu_mulai	DATETIME	Tidak	Waktu mulai kegiatan
waktu_selesai	DATETIME	Ya	Waktu selesai kegiatan
durasi	INT (10) UNSIGNED	Ya	Durasi dalam menit
pic	VARCHAR (255)	Tidak	Penanggung jawab
biaya	DECIMAL (15,2)	Ya	Total biaya maintenance
jenis	ENUM('preventive', 'corrective')	Tidak	Jenis maintenance
status	ENUM('Finish', 'On Going')	Tidak	Status pengerjaan
created_at	DATETIME	Tidak	Tanggal dibuat (auto)
updated_at	DATETIME	Tidak	Tanggal diperbarui (auto)

Tabel 3.4 menyimpan data kegiatan *maintenance* yang mencakup identitas unit, detail proyek, waktu pelaksanaan, durasi, biaya, penanggung jawab, serta jenis dan status pemeliharaan. Kolom id berfungsi sebagai *primary key*, sedangkan *unit_id* dan *property_id* sebagai relasi ke tabel unit dan properti.

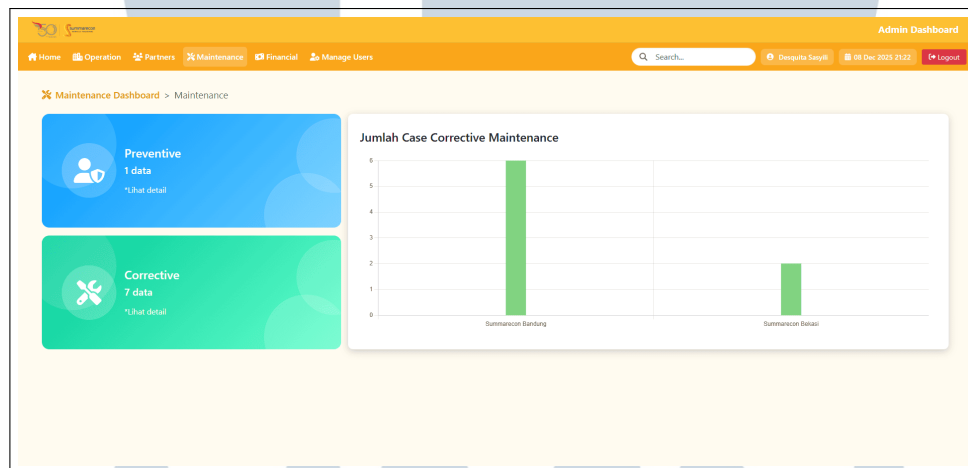
Tabel ini juga mencatat informasi operasional, biaya, jenis *maintenance*

(*preventive* atau *corrective*), status pengerjaan, serta waktu pembuatan dan pembaruan data melalui kolom `created_at` dan `updated_at`.

Jika pekerjaan masih berstatus *On Going*, kolom waktu selesai dan biaya diperbolehkan kosong. Nilai durasi akan terisi 0 jika waktu selesai belum diinput. Namun, ketika waktu mulai dan waktu selesai telah diisi, sistem akan otomatis menghitung durasi berdasarkan selisih kedua waktu tersebut.

D Implementasi

Pada proses pengembangan halaman *maintenance*, ketika halaman ini pertama kali diakses akan ditampilkan antarmuka seperti terlihat pada Gambar 3.36.



Gambar 3.36. Tampilan Utama Halaman Maintenance

Pada Gambar 3.36 ditampilkan dua *card* yang merepresentasikan jenis *maintenance*, yaitu *preventive* dan *corrective*. Setiap *card* menampilkan total jumlah data sesuai kategori, yang diperoleh dari tabel *maintenance* berdasarkan nilai ENUM pada kolom jenis *detail* struktur basis data terlihat pada Tabel 3.4.

Di sisi kanan kedua *card* tersebut terdapat visualisasi berupa grafik *bar chart* yang menampilkan jumlah kasus *corrective maintenance* untuk setiap *unit_id*. Ketika salah satu *card* ditekan, pengguna akan diarahkan ke halaman daftar *maintenance* sesuai jenis yang dipilih. Contoh tampilan halaman tersebut ditunjukkan pada Gambar 3.37.

Corrective Maintenance

Total Biaya: **Rp 2.560.000**

No	Tanggal	Project	Alamat	Issue	Waktu Mulai	Waktu Selesai	Durasi	PIC	Status	Biaya
1	19/11/2025	Summarecon Mall Kelapa Gading	Mall Kelapa Gading	test kabel fiber optic	19/Nov/2025 10:53	-	-	Desquita Sasyili	Unhappy	Rp 0
2	07/07/2025	Backbone Kawasan Halte Lotus	Cluster Lotus	FO Cut	07/Jul/2025 10:30	08/Jul/2025 14:00	1 hari 3 jam 30 menit	Kim Jennie	Happy	Rp 0
3	01/06/2025	Kawasan Summarecon Bekasi	Kawasan Summarecon Bekasi	Perbaikan Jaringan Kabel FO	01/Jun/2025 11:11	30/Jun/2025 17:00	29 hari 5 jam 49 menit	Bogummy	Happy	Rp 900.000
4	30/03/2025	Summarecon Bandung	Cluster Btari	FO cut: digigit tikus	29/Mar/2025 02:00	30/Mar/2025 09:00	1 hari 7 jam	Kim Jennie	Unhappy	Rp 1.660.000

Total Biaya: **Rp 2.560.000**

Gambar 3.37. Tampilan Data Maintenance Berdasarkan Jenis Card

Gambar di atas menampilkan halaman setelah pengguna memilih jenis *maintenance*. Halaman ini menampilkan seluruh data *maintenance* berdasarkan *unit_id* sesuai jenis yang dipilih (*preventive* atau *corrective*) dalam bentuk *datatable*. Informasi yang ditampilkan meliputi tanggal, proyek, alamat, *issue*, waktu, durasi, *PIC*, status, dan biaya. Tersedia fitur *filter* bulan dan tahun, *show entries*, pencarian, serta *pagination* untuk memudahkan pengelolaan data.

Pada Gambar 3.38 ditampilkan data *maintenance* dalam bentuk *datatable* yang dilengkapi dengan kolom-kolom informasi *maintenance*.

Data Maintenance - Summarecon Bandung

Total Biaya: **Rp 1.660.000**

No	Tanggal	Project	Alamat	Issue	Waktu Mulai	Waktu Selesai	Durasi	PIC	Status	Biaya	Aksi
1	19/11/2025	Summarecon Mall Kelapa Gading	Mall Kelapa Gading	test kabel fiber optic	19/Nov/2025 10:53	-	-	Desquita Sasyili	Unhappy	Rp 0	Detail Edit
2	08/05/2025	Summarecon Bandung	Cluster Betari	Redaman Tinggi	08/May/2025 15:30	08/May/2025 21:30	6 jam	Desquita	Happy	Rp 760.000	Detail Edit
3	30/03/2025	Summarecon Bandung	Cluster Btari	FO cut: digigit tikus	29/Mar/2025 02:00	30/Mar/2025 09:00	1 hari 7 jam	Kim Jennie	Unhappy	Rp 1.660.000	Detail Edit

Total Biaya: **Rp 1.660.000**

Gambar 3.38. Tampilan Pengelolaan CRUD Data Maintenance

Pada halaman data *maintenance* ini tersedia fitur penyaringan data berdasarkan bulan dan tahun. Selain itu, sistem menampilkan total biaya *maintenance* yang diperoleh dari hasil penjumlahan kolom biaya. Halaman ini juga

dilengkapi dengan tombol untuk melakukan proses ekspor data ke dalam file.

Pada Gambar 3.39 ditampilkan *modal form* untuk pembuatan (*create*) data *maintenance* yang muncul setelah pengguna menekan tombol “Tambah Data” berwarna biru.

Gambar 3.39. Tampilan Modal Form Tambah Data Maintenance

Data yang diinput pada *form* ini meliputi tanggal input, nama proyek, alamat, *issue*, waktu mulai, waktu selesai (opsional), penanggung jawab, biaya (opsional), jenis *maintenance*, dan status. Pada Gambar 3.40 ditampilkan halaman *detail* data *maintenance*. Halaman ini dapat diakses ketika pengguna menekan tombol “Detail” pada kolom aksi yang terdapat pada *datatable*.

Gambar 3.40. Tampilan Detail Data Maintenance

Data yang ditampilkan pada halaman *detail maintenance* ini mencakup seluruh informasi yang tersimpan di dalam basis data. Apabila pengguna ingin

melakukan perubahan terhadap data tersebut, pengguna dapat menekan tombol “Edit” berwarna kuning yang terletak di samping tombol “Batal”.

Pada Gambar 3.41 ditampilkan halaman edit data *maintenance* yang menampilkan *card* beserta data *maintenance* yang telah tersimpan sebelumnya.

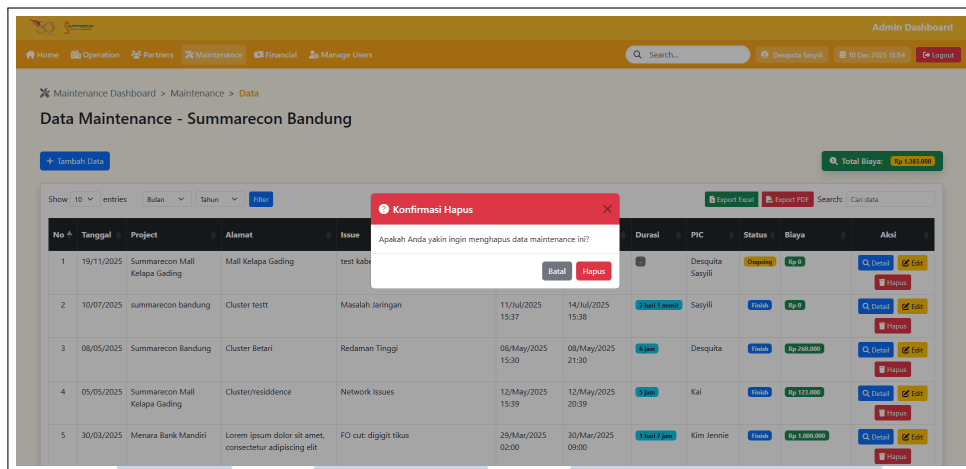
The screenshot shows a web application interface for editing maintenance data. The top navigation bar includes links for Home, Operation, Partners, Maintenance, Financial, and Manage Users. The main content area is titled 'Edit Data Maintenance' and contains a form with the following fields: 'Tanggal Input' (30/03/2025), 'Pilih Proyek' (Menara Bank Mandiri), 'Alamat' (Lorem ipsum dolor sit amet, consectetur adipiscing elit), 'Issue' (FO cut: digigit tikus), 'Waktu Mulai' (29/03/2025 02:00), 'Waktu Selesai' (30/03/2025 09:00), 'PIC' (Kim Jennie), 'Biaya' (1.000.000), 'Jenis' (Corrective), and 'Status' (Finish). At the bottom right of the form are two buttons: 'Batal' (Cancel) and 'Simpan Perubahan' (Save Changes).

Gambar 3.41. Tampilan Edit Data Maintenance

Setelah seluruh perubahan selesai dilakukan, pengguna dapat menekan tombol “Simpan Perubahan” untuk menyimpan data. Apabila pengguna tidak ingin mengubah data atau ingin membatalkan proses pengeditan, pengguna dapat menekan tombol “Batal”.

Pada Gambar 3.42 ditampilkan *modal* konfirmasi penghapusan yang muncul ketika pengguna menekan tombol “Hapus” untuk menghapus data *maintenance*. *Modal* konfirmasi ini berfungsi untuk memverifikasi apakah pengguna benar-benar ingin melakukan penghapusan data.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.42. Tampilan Hapus Data Maintenance

Pada Gambar 3.43 di bawah, ditunjukkan hasil ekspor data ke dalam format PDF yang memuat seluruh informasi terkait, termasuk kolom-kolom penting, sehingga dapat melihat, menyimpan, dan membagikan data secara lengkap sesuai dengan tampilan pada sistem.

No	Tanggal	Project	Alamat	Issue	Waktu Mulai	Waktu Selesai	Durasi	PIC	Status	Biaya
1	19/11/2025	Summarecon Mall Kelapa Gading	Mall Kelapa Gading	test kabel fiber optic	19/Nov/2025 10:53	-	-	Desquita Sasyili	Ongoing	Rp 0
2	10/07/2025	summarecon bandung	Cluster testt	Masalah Jaringan	11/Jul/2025 15:37	14/Jul/2025 15:38	3 hari 1 menit	Sasyili	Finish	Rp 0
3	08/05/2025	Summarecon Bandung	Cluster Betari	Redaman Tinggi	08/May/2025 15:30	08/May/2025 21:30	6 jam	Desquita	Finish	Rp 260.000
4	05/05/2025	Summarecon Mall Kelapa Gading	Cluster/residence	Network Issues	12/May/2025 15:39	12/May/2025 20:39	5 jam	Kai	Finish	Rp 123.000
5	30/03/2025	Menara Bank Mandiri	Lorem ipsum dolor sit amet, consectetur adipiscing elit	FO cut: digitik tikus	29/Mar/2025 02:00	30/Mar/2025 09:00	1 hari 7 jam	Kim Jennie	Finish	Rp 1.000.000
										Total Biaya: Rp 1.383.000

© Jejaring Ultra Prima

Gambar 3.43. Hasil Export Data Maintenance Format PDF

Hasil ekspor data dalam format PDF menampilkan tabel *maintenance* sesuai dengan data yang dipilih pada halaman *maintenance*. Dokumen ini dilengkapi dengan judul (*title*) yang diambil dari nama unit pada basis data.

Pada Gambar 3.44 ditampilkan hasil ekspor data dalam bentuk XLSX.

No	Tanggal	Project	Alamat	Issue	Waktu Mulai	Waktu Selesai	Durasi	PIC	Status	Biaya
1	1/11/2025	Summarecon Mall Kelapa Gading	Mall Kelapa Gading	test kabel fiber optic	28/Nov/2025 15:53	-	-	Desquitta Sasyli	Ongoing	Rp 0
2	2/10/2025	Summarecon Bandung	Cluster test	Masalah jaringan	11/Jul/2025 15:37	14/Jul/2025 15:38	3 hari 1 menit	Sasyli	Finish	Rp 0
3	08/05/2025	Summarecon Bandung	Cluster Betari	Redaman Tengg	06/May/2025 15:30	06/May/2025 21:30	6 jam	Desquitta	Finish	Rp 200.000
4	05/05/2025	Summarecon Mall Kelapa Gading	Cluster/Residence	Network issues	12/May/2025 15:39	12/May/2025 20:39	5 jam	Kai	Finish	Rp 123.000
5	30/01/2025	Menara Bank Mandiri	Lorem ipsum dolor sit amet, consectetur adipiscing elit	FO cut: digitik tikus	29/Mar/2025 02:00	30/Mar/2025 09:00	1 hari 7 jam	Kim Jennie	Finish	Rp 1.000.000
Total Biaya:										Rp 1.183.000

Gambar 3.44. Hasil Export Data Maintenance Format XLSX

Hasil ekspor data dalam format XLSX menampilkan tabel *maintenance* sesuai dengan data yang dipilih pada halaman *maintenance*. Dokumen ini juga dilengkapi dengan judul (*title*) yang diambil dari nama unit pada basis data.

3.5 Kendala dan Solusi yang Ditemukan

3.5.1 Kendala

Selama program magang berlangsung, terdapat beberapa kendala yang dihadapi dalam proses perancangan dan pengembangan, baik itu besar maupun kecil. Kendala yang dihadapi selama proses kegiatan kerja magang ini antara lain sebagai berikut.

1. Perbedaan jenis data per kelompok dan per jenis, yang diambil melalui relasi *database* antar tabel, menjadi kendala karena data yang ditampilkan dan disimpan memiliki format dan input yang beragam.
2. Penerapan *email blast* untuk data yang dimasukkan menjadi kendala, karena pada pengembangan awal tidak terdapat ketentuan untuk mengirim *email blast* saat data baru diunggah, sehingga perlu sedikit penyesuaian pada kode.

3.5.2 Solusi

Untuk mengatasi kendala-kendala tersebut, solusi yang diterapkan selama pelaksanaan magang agar pengembangan tetap dapat diselesaikan dengan baik. Berikut adalah solusi yang ditemukan.

1. Melakukan standarisasi format data saat pengambilan dari *database*, menggunakan *query* yang terstruktur dan *join* antar tabel yang konsisten.

Selain itu, menerapkan validasi data pada sisi *backend* sebelum ditampilkan atau disimpan, sehingga data tetap konsisten dan dapat diolah dengan baik.

2. Menambahkan logika *email blast* di *backend* saat data baru diunggah, memeriksa validitas *email* dan *role* penerima, lalu mengirim otomatis dengan PHPMailer beserta notifikasi sukses atau gagal.

