

## BAB 3

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Koordinasi

Pelaksanaan kerja magang di Dinas Komunikasi dan Informatika (Diskominfo) Kota Balikpapan menempatkan penulis pada posisi sebagai *Web Developer* dan *Flutter Developer* di bawah supervisi Bidang *e-Government*. Kegiatan magang ini berfokus pada pengembangan tiga produk utama: Sistem Administrasi Kenaikan Gaji Berkala berbasis web, UI Aplikasi Mobile *Superapp* Balikpapan, dan Layanan *E-Signature* dokumen digital.

Bimbingan teknis diberikan secara langsung oleh pembimbing lapangan yang merupakan pegawai senior Bidang *e-Government*, dengan koordinasi rutin bersama staf kepegawaian terkait kebutuhan spesifik sistem administrasi. Seluruh progres pekerjaan dilaporkan dalam forum rapat mingguan setiap hari Jumat pukul 09.00-11.00 WITA. Pertemuan ini berfungsi sebagai media evaluasi capaian mingguan, identifikasi kendala teknis, serta penyusunan rencana kerja untuk minggu-minggu berikutnya. Mekanisme koordinasi dilakukan melalui kombinasi komunikasi luring dan daring. Untuk konsultasi rutin dan berbagi dokumen teknis difasilitasi melalui *WhatsApp*.

Selain tanggung jawab utama dalam ketiga proyek inti, penulis juga dilibatkan dalam beberapa tugas pendukung seperti perbaikan tampilan aplikasi pemerintahan yang sudah ada. Keterlibatan dalam aktivitas tambahan ini memberikan perspektif holistik tentang ekosistem teknologi informasi di lingkungan pemerintah daerah, sekaligus memperluas wawasan tentang standar dan regulasi pengembangan sistem di sektor publik.

Struktur pengawasan dalam pelaksanaan magang mengikuti hierarki organisasi Diskominfo secara ketat. Laporan kemajuan teknis disampaikan secara berkala kepada pembimbing lapangan. Proses ini memastikan seluruh pengembangan sistem selaras dengan visi transformasi digital pemerintah kota dan memenuhi standar tata kelola.

Dengan adanya kerja sama yang baik antara pembimbing lapangan, tim teknis, dan pegawai kepegawaian, serta ditunjang oleh koordinasi yang rutin dan efektif, pelaksanaan magang dapat berjalan dengan lancar.

### 3.2 Tugas yang Dilakukan

Selama pelaksanaan magang, penulis terlibat dalam pengembangan tiga produk teknologi utama yang mencakup *web administration*, *mobile application*, dan layanan *e-signature*. Ketiga proyek tersebut dikembangkan dengan pendekatan *system development life cycle* yang terdiri atas fase analisis kebutuhan, perancangan arsitektur, implementasi teknis, pengujian fungsional, dan optimisasi akhir.

Pengembangan sistem dalam magang ini mengikuti metodologi System Development Life Cycle (SDLC) dengan pendekatan waterfall yang disesuaikan dengan kebutuhan lingkungan pemerintahan. Setiap proyek dimulai dengan tahap analisis kebutuhan melalui diskusi intensif dengan pembimbing lapangan dan staf terkait untuk memahami requirements fungsional dan non-fungsional. Tahap perancangan meliputi pembuatan database schema, dan arsitektur sistem yang sesuai dengan standar pemerintah. Implementasi dilakukan secara iteratif dengan review berkala setiap minggu untuk memastikan output sesuai ekspektasi. Fase testing mencakup functional testing, user acceptance testing, dan performance testing sebelum deployment ke environment production. Maintenance dan dokumentasi dilakukan pada tahap akhir untuk memastikan sustainabilitas sistem.

Pengembangan *web-based administration system* untuk manajemen Kenaikan Gaji Berkala (KGB) dan Kenaikan Pangkat (KP) menggunakan *framework* Laravel dengan basis data MySQL. Tahap awal meliputi analisis *business process* sistem remunerasi pemerintah daerah dan identifikasi kebutuhan fungsional seperti *role-based access control*, *filtering* data berdasarkan Satuan Kerja Perangkat Daerah (SKPD), serta pelacakan riwayat perubahan. Implementasi teknis mencakup pembuatan modul CRUD dengan integrasi *Yajra DataTables*, dan mekanisme *history tracking* perubahan data. Pada fase pengujian dilakukan validasi konsistensi *filter* antar-modul dan penanganan *edge case* seperti pengecekan *null pointer exception*. Arsitektur sistem menerapkan pola *Controller-Service-Repository* untuk meningkatkan modularitas kode.

Perancangan dan implementasi aplikasi mobile berbasis *Flutter framework* dengan pendekatan *component-based architecture*. Fase desain meliputi studi komparatif antarmuka *superapp* terkini. Pengembangan antarmuka pengguna menerapkan prinsip *responsive design* dengan komponen *reusable widget* seperti *carousel hero*, *news card*, dan *button row*. Integrasi teknologi pendukung meliputi *GetX* untuk *state management*, dan *WebView* untuk konten eksternal. Optimisasi akhir mencakup penyesuaian *layout* untuk berbagai *device breakpoint*, penerapan

*vector assets*, serta penyempurnaan *user experience* pada alur autentikasi (*login*, pendaftaran, dan reset kata sandi).

Pengembangan platform penandatanganan dokumen digital dengan integrasi *third-party signing API*. Arsitektur sistem menggabungkan *PDF.js* untuk *rendering* dokumen, *Fabric.js* untuk manipulasi *canvas*, dan *Three.js* untuk elemen visual 3D. Tahap implementasi meliputi pembuatan mekanisme *overlay positioning* tanda tangan, validasi NIK dan *passphrase*, serta *workflow* penandatanganan multi-tahap. Versi awal sistem (*v1*) ditingkatkan ke *v2* dengan penambahan mekanisme keamanan *payload signing* dan verifikasi integritas dokumen. Pengujian fungsional mencakup simulasi *end-to-end process* dari unggah dokumen, penempatan tanda tangan, hingga *download* hasil tertandatangani. Dokumentasi teknis berisi spesifikasi *API payload* dan mekanisme *error handling* disusun untuk kebutuhan pemeliharaan sistem.

Seluruh proyek dikembangkan dengan memanfaatkan *version control system* berbasis Git dan *continuous integration* melalui platform GitHub. *Integrated Development Environment (IDE)* utama yang digunakan adalah Visual Studio Code dengan ekstensi pendukung produktivitas seperti *ESLint* dan *Prettier* untuk menjaga kualitas kode. Pengujian performa dilakukan dengan alat *Chrome DevTools* dan *Lighthouse* untuk mengoptimalkan *loading time* dan *resource utilization*.

### 3.3 Uraian Pelaksanaan Magang

Program magang Merdeka Belajar Kampus Merdeka (MBKM) di Dinas Komunikasi dan Informatika (Diskominfo) Kota Balikpapan dilaksanakan selama periode 17 Februari 2025 hingga 28 Juni 2025. Selama masa magang ini, penulis ditempatkan di Bidang *E-Government* dengan peran ganda sebagai *Web Developer* dan *Flutter Developer*, bertanggung jawab penuh dalam pengembangan tiga proyek sistem informasi strategis untuk mendukung transformasi digital pemerintah kota, yaitu Sistem Administrasi Kenaikan Gaji Berkala berbasis web, UI Aplikasi Mobile *Superapp* Balikpapan, dan Layanan *E-Signature* dokumen digital. Berbagai tanggung jawab penulis, di antara lain dituliskan seperti pada tabel:

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

Minggu Ke-	Pekerjaan yang dilakukan
1	Setup proyek Laravel dan struktur database; Implementasi CRUD dasar untuk entitas Eselon, Jabatan, Golongan, dan SKPD; Pengembangan modul Pegawai dengan validasi NIP; Integrasi template admin dan modularisasi komponen UI; Pembuatan database seeder untuk data uji
2	Implementasi sistem autentikasi dan role management; Penerapan pola CSR (Controller-Service-Repository); Integrasi Yajra DataTables dengan terjemahan Indonesia; Pengembangan sistem notifikasi untuk operasi CRUD; Perbaikan UI: breadcrumb, responsive design, dan format tanggal Indonesia
3	Pengembangan modul KGB (Kenaikan Gaji Berkala) dan KP (Kenaikan Pangkat); Implementasi fitur history tracking untuk KGB/KP; Perbaikan tampilan tabel dengan rowspan dan pewarnaan; Penambahan validasi: NIP leading zero, penanganan data kosong; Implementasi filter berdasarkan SKPD untuk modul KGB/KP; Pembaruan seeder untuk data yang lebih akurat
4	Implementasi sistem filtering berdasarkan SKPD user login; Pengembangan logika akses terbatas per SKPD; Perbaikan bug terkait konsistensi filter dan pencarian; Penambahan fitur "Semua SKPD" dengan mekanisme filter; Refactor kode dan penanganan exception melalui modal notifikasi; Perbaikan akhir (bugfix) sebelum deployment; Memulai pembelajaran mandiri Flutter setelah deployment

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

Tabel 3.2. Kelanjutan Pekerjaan Mingguan Pelaksanaan Kerja Magang

Minggu Ke-	Pekerjaan yang dilakukan
5	Pembelajaran Flutter dasar hingga menengah melalui seri Flutter Fasttrack; Inisialisasi proyek aplikasi mobile; Pengembangan halaman home dengan carousel hero dan section widget; Implementasi reusable components (button row, section title); Pengembangan halaman login awal dengan widget input terpisah; Pengaturan dasar aplikasi (versioning, gradient theme)
6	Redesain UI/UX aplikasi menggunakan Figma; Studi komparatif UI superapp lainnya; Perancangan UI baru; Pengembangan konsep visual identity; Penyusunan design system untuk konsistensi UI
7	Penyempurnaan desain Figma; Optimisasi layout untuk berbagai ukuran layar; Pemilihan palette warna final dan tipografi; Penyiapan asset visual (icon, illustration)
8	Implementasi desain Figma ke kode Flutter; Refactor kode dengan GetX untuk state management; Perbaikan responsive design (carousel, cards); Penerapan vector background pada halaman login; Pengembangan news card dengan fixed height dan warna terkustomisasi; Implementasi button row dengan data passing; Perbaikan tata letak (padding, overflow)
9	Penyempurnaan UI akhir: card design, vector backgrounds; Implementasi YouTube cards dan latar belakang vektor; Penyesuaian warna (darker font, gradient improvement); Webview integration untuk card content; Final testing dan perbaikan responsivitas; Dokumentasi setting dan finalisasi UI sebelum integrasi backend

Tabel 3.3. Kelanjutan Pekerjaan Mingguan Pelaksanaan Kerja Magang

Minggu Ke-	Pekerjaan yang dilakukan
10	Penelitian arsitektur sistem e-signature; Studi komparatif implementasi esign website lain; Analisis payload signing dan mekanisme keamanan; Evaluasi library: PDF.js, Fabric.js, Signature Pad; Perencanaan alur kerja dan desain UI awal; Persiapan lingkungan pengembangan
11	Implementasi halaman utama dengan background 3D (Three.js); Pengembangan halaman sign: upload PDF dan render halaman (PDF.js); Implementasi canvas untuk penempatan tanda tangan (Fabric.js); UI untuk positioning image/signature dan transformasi; Validasi form: disable button sebelum upload PDF; Sistem penamaan file dan error handling
12	Penyempurnaan manajemen canvas multi-halaman; Implementasi penempatan tanda tangan di berbagai halaman; Perbaikan UI: sticky navbar, responsive design; Penanganan bug transformasi dan ukuran canvas; Pengembangan sistem penyimpanan sementara dokumen; Optimisasi kode dan modularisasi fungsi
13	Implementasi form konfirmasi NIK dan passphrase; Integrasi API signing pertama (Laravel HTTP); Logika penggabungan upload dan signing dalam satu request; Penanganan response API dan notifikasi error; Pengembangan fallback rendering untuk testing; Perbaikan bug rendering PDF dari API
14	Pengembangan halaman after-signing (tanpa validasi awal); Implementasi progress bar untuk proses signing; Sistem verifikasi tanda tangan awal; Fungsi download dokumen yang sudah ditandatangani; Optimisasi performa rendering dokumen besar; Penanganan jenis response dari API
15	Implementasi penuh sistem verifikasi tanda tangan; Penyempurnaan UI progress bar dan status; Perbaikan bug download dan penamaan file; Optimisasi ukuran overlay sebelum API call; Persiapan upgrade ke API v2 signing; Testing lintas browser dan perangkat

Tabel 3.4. Kelanjutan Pekerjaan Mingguan Pelaksanaan Kerja Magang

Minggu Ke-	Pekerjaan yang dilakukan
16	Implementasi penuh API v2 signing dengan mekanisme keamanan baru; Pengembangan sistem verifikasi tanda tangan versi 2; Manipulasi teknis ukuran overlay sebelum pengiriman API; Penyempurnaan halaman post-sign details; Final testing alur end-to-end; Dokumentasi payload dan mekanisme signing
17	Perubahan struktur database dengan penambahan kolom NIK dan nama; Penyempurnaan UI modal signing dan implementasi fitur unzip; Penerjemahan antarmuka ke Bahasa Indonesia; Pengembangan filter status dan fitur edit/cancel request; Penyelesaian CRUD user dengan perbaikan proses signing
18	Implementasi role-based access control dan dashboard admin; Pengembangan API v2 dengan verifikasi tanda tangan; Optimasi bulk signing PDF dan manipulasi overlay; Testing end-to-end alur signing; Finalisasi fitur utama sebelum deployment; Penutupan magang

### 3.4 Pengenalan Magang

Kegiatan magang dimulai pada tanggal 17 Februari 2025 pukul 08.00 WITA di Dinas Komunikasi dan Informatika (Diskominfo) Kota Balikpapan. Lokasi kantor Diskominfo terletak di lantai dua gedung pemerintahan kota, tepatnya di Jalan Jenderal Sudirman No. 86, Klandasan Ulu. Diskominfo memiliki dua kantor terpisah, yaitu satu unit di seberang kompleks gedung pemerintahan kota dan satu unit lagi di dalam gedung utama pemerintahan yang khusus menangani bagian e-Government. Penempatan magang dilakukan di divisi e-Government dalam gedung utama, khususnya di ruangan DATSI (Data dan Sistem Informasi).

Pada hari pertama, dilakukan pengenalan terhadap lingkungan kerja pemerintah kota, termasuk struktur organisasi dan peran strategis Diskominfo dalam mendukung digitalisasi pemerintahan. Diadakan pula perkenalan dengan beberapa peserta magang lainnya yang sedang menjalani program magang di instansi yang sama, berasal dari beberapa universitas dan program studi. Selain itu, dilakukan pertemuan dengan staf-staf Diskominfo, meliputi tim developer, pranala komputer,

dan admin yang beberapa akan membantu dan membimbing dalam proyek pertama, website KGB.

Pada tahap pengenalan, penulis diperkenalkan dengan beberapa figur penting yang akan membimbing selama periode magang. Supervisor utama magang yang bertanggung jawab atas koordinasi umum dan evaluasi kinerja selama program berlangsung. Selain itu, penulis juga berkenalan dengan dua pembimbing teknis utama yang akan mengawasi langsung pengembangan proyek, dua orang yang biasa menjadi lead developer dan technical advisor di DATSI Diskominfo. Kedua pembimbing teknis ini memiliki peran strategis dalam memberikan briefing detail mengenai spesifikasi proyek, melakukan review berkala terhadap progress pengembangan, serta memberikan masukan teknis dan revisi yang diperlukan untuk memastikan kualitas hasil akhir sesuai dengan standar yang ditetapkan oleh Diskominfo Balikpapan.

Pertemuan pertama dengan supervisor magang dilaksanakan untuk menjelaskan detail peraturan magang, jam kerja, dan tanggung jawab selama periode magang. Jam kerja magang mengikuti ketentuan pemerintah kota, yaitu Senin-Kamis pukul 07.00-16.30 WITA dengan waktu istirahat pukul 12.00-13.00 WITA, sedangkan hari Jumat pukul 07.00-11.30 WITA tanpa waktu istirahat. Supervisor bertanggungjawab untuk mengawasi, dan akan melakukan update *progress* dengan tim pengembang pada setiap hari Jum'at.

### **3.5 Website Administrasi KGB**

Website Administrasi Kenaikan Gaji Berkala (KGB) adalah sistem berbasis web untuk mengelola kenaikan gaji dan pangkat Pegawai Negeri Sipil (PNS) di Pemerintah Kota Balikpapan. Sistem ini menggantikan proses manual dengan fitur digital seperti input data pegawai (CRUD), pengingat otomatis jadwal kenaikan gaji/pangkat, dan pembatasan akses berdasarkan peran pengguna. Tujuannya adalah meningkatkan efisiensi, mengurangi kesalahan, dan mempercepat proses administrasi kepegawaian secara terpusat dan transparan.

#### **3.5.1 Analisis Kebutuhan & Desain Sistem**

Setelah proses pengenalan, dilakukan perencanaan teknis pengembangan website. Tim pengembang menjelaskan bahwa bahasa pemrograman utama yang akan digunakan adalah Laravel dengan fleksibilitas dalam pemilihan framework

CSS. Untuk mempercepat proses pengembangan, diperbolehkan penggunaan template admin page gratis sebagai boilerplate. Dipilih template Sneat Dashboard untuk Laravel yang dikembangkan oleh Themeselection, yang telah dilengkapi dengan Laravel Breeze untuk autentikasi dan Yajra Datatables untuk manajemen tabel interaktif. Template ini kemudian dimodifikasi dan disesuaikan dengan kebutuhan proyek.

Tim pengembang memberikan gambaran menyeluruh tentang sistem yang akan dibangun, mencakup tujuan penggunaan website dan mekanisme kerjanya. Website ini dirancang untuk mengelola administrasi Kenaikan Gaji Berkala (KGB) dan kenaikan pangkat Pegawai Negeri Sipil (PNS) di lingkungan pemerintah Kota Balikpapan. Penjelasan fitur dilakukan secara bertahap per bagian, dengan pemberian tugas baru yang diberikan setiap kali tugas sebelumnya telah selesai dikerjakan. Proses update progress dan penugasan baru biasanya dilakukan setiap hari Jumat melalui rapat mingguan.

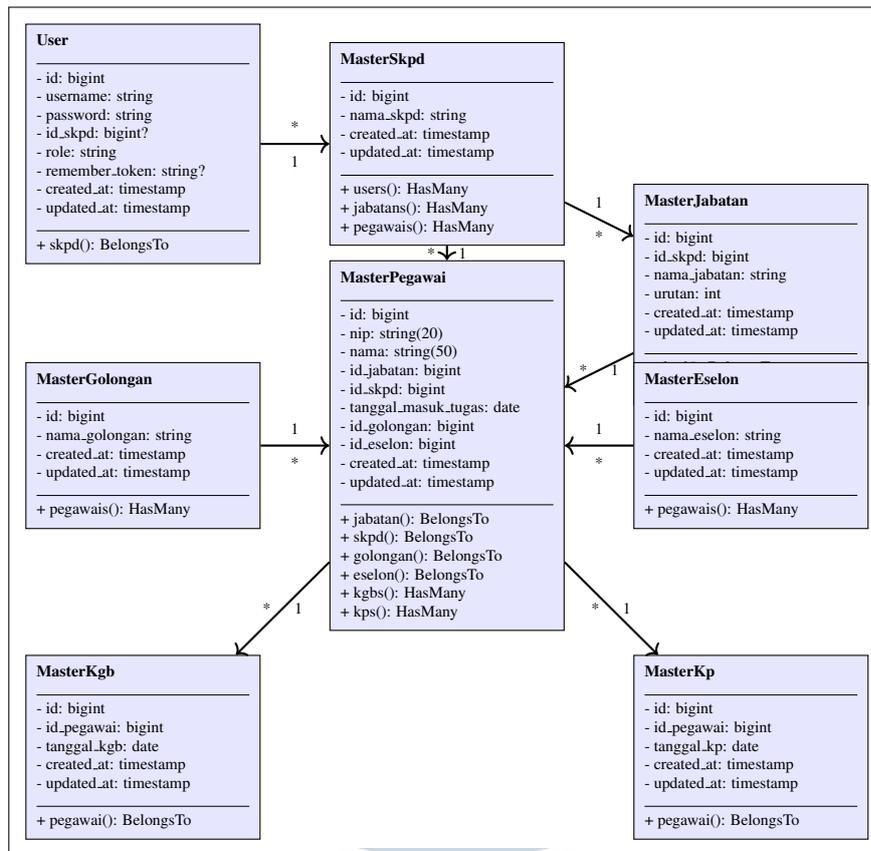
Selama tahap analisis kebutuhan, dilakukan identifikasi fitur-fitur utama yang diperlukan dalam sistem, antara lain:

- Sistem autentikasi dengan multi-role (admin dan pegawai)
- Manajemen data pegawai (CRUD)
- Pengajuan dan persetujuan KGB
- Riwayat kenaikan pangkat
- Filtering data

Untuk menjaga kolaborasi dan version control, seluruh kode proyek diunggah ke platform GitHub. Seluruh tim pengembang diundang ke dalam repository yang sama sehingga dapat memantau perkembangan kode melalui fitur commit history. Pendekatan ini memungkinkan tracking perubahan, kolaborasi yang lebih terstruktur, serta memudahkan proses debugging dan pengembangan fitur baru.

## **A Class Diagram**

Aplikasi KGB Diskominfo memiliki struktur class yang saling terhubung melalui berbagai metode relasi Eloquent. Pada class User, metode `skpd()` menggunakan relasi `BelongsTo` untuk menghubungkan user dengan SKPD



Gambar 3.1. Class Diagram untuk Website KGB Diskominfo

tertentu, memungkinkan sistem role-based access control berdasarkan unit kerja. Class MasterSkpd sebagai entitas sentral memiliki tiga metode relasi utama: `users()` dengan relasi `HasMany` untuk mengelola user yang terdaftar di SKPD tersebut, `jabatans()` untuk mengakses daftar jabatan yang tersedia di SKPD, dan `pegawais()` untuk mengelola data pegawai yang bertugas di unit kerja tersebut.

Class MasterJabatan mengimplementasikan metode `skpd()` dengan relasi `BelongsTo` untuk mereferensikan SKPD induk, serta `pegawais()` dengan relasi `HasMany` untuk mengelola pegawai yang menduduki jabatan tersebut. Sebagai core entity, MasterPegawai memiliki lima metode relasi `BelongsTo`: `jabatan()`, `skpd()`, `golongan()`, dan `eselon()` untuk mengakses data referensi terkait, serta dua metode `HasMany`: `kgbs()` dan `kps()` untuk mengelola riwayat kenaikan gaji berkala dan kenaikan pangkat pegawai. Class referensi MasterGolongan dan MasterEselon masing-masing memiliki metode `pegawais()` dengan relasi `HasMany` untuk mengakses daftar pegawai berdasarkan golongan dan eselon tertentu.

Untuk pengelolaan transaksi, class MasterKgb dan MasterKp menggunakan metode pegawai () dengan relasi BelongsTo untuk mereferensikan data pegawai yang bersangkutan. Struktur relasi ini memungkinkan aplikasi untuk melakukan operasi CRUD yang efisien, pelacakan riwayat kenaikan gaji dan pangkat, serta implementasi business logic yang kompleks seperti perhitungan otomatis tanggal KGB berikutnya dan validasi kelayakan kenaikan pangkat berdasarkan kriteria yang telah ditetapkan.

## **B Skema Database**

Perancangan struktur database dilakukan secara bertahap sesuai dengan kebutuhan pengembangan sistem. Pada tahap awal, hanya dibutuhkan tabel-tabel dasar yaitu master\_pegawai, master\_eselon, master\_skpd, master\_golongan, dan master\_jabatan. Seluruh tabel sengaja diberi prefix "master\_" sesuai dengan naming convention yang diterapkan tim pengembang. Tabel master\_pegawai awalnya dirancang dengan relasi ke tabel master\_jabatan, dimana data SKPD pegawai diperoleh melalui relasi jabatan yang dimiliki (melalui id\_jabatan).

Seiring perkembangan sistem, dilakukan beberapa penyesuaian struktur database. Salah satu perubahan signifikan adalah penambahan field id\_skpd langsung pada tabel master\_pegawai untuk mempermudah query data, dimana sebelumnya relasi SKPD harus melalui tabel jabatan. Perubahan semacam ini biasanya diberikan oleh tim pengembang saat dibutuhkan dalam proses pengembangan fitur tertentu.

Tabel users kemudian ditambahkan untuk menangani sistem autentikasi dengan field dasar seperti username, password, dan role. Field id\_skpd juga dimasukkan dalam tabel users untuk memungkinkan filtering data berdasarkan SKPD pengguna yang login.

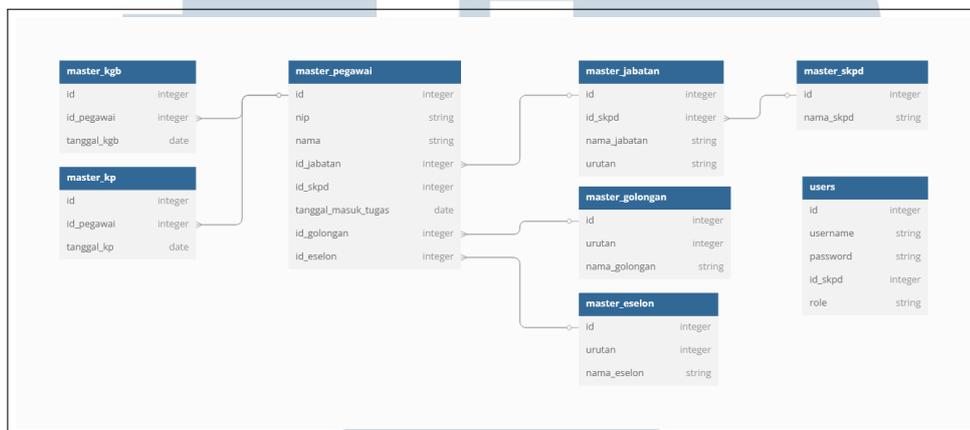
Pada tahap berikutnya, dikembangkan tabel master.kgb dan master.kp untuk mencatat riwayat Kenaikan Gaji Berkala dan Kenaikan Pangkat. Kedua tabel ini memiliki struktur serupa dengan field id\_pegawai sebagai foreign key ke tabel master\_pegawai dan field tanggal untuk mencatat waktu transaksi. Relasi antar tabel dirancang sedemikian rupa untuk mendukung fitur-fitur utama sistem seperti:

- Pelacakan riwayat KGB dan KP per pegawai
- Filtering data berdasarkan SKPD
- Penentuan KGB/KP berikutnya berdasarkan tanggal terakhir

- *Priveleges checking* dalam website

Struktur database terus disempurnakan selama proses pengembangan, termasuk penambahan field urutan pada tabel master untuk keperluan sorting data. Seluruh migrasi database diimplementasikan menggunakan fitur migration Laravel, memungkinkan perubahan struktur dapat dilakukan secara terkontrol dan terdokumentasi dengan baik.

Pada akhir pengembangan, dihasilkan struktur database berikut:



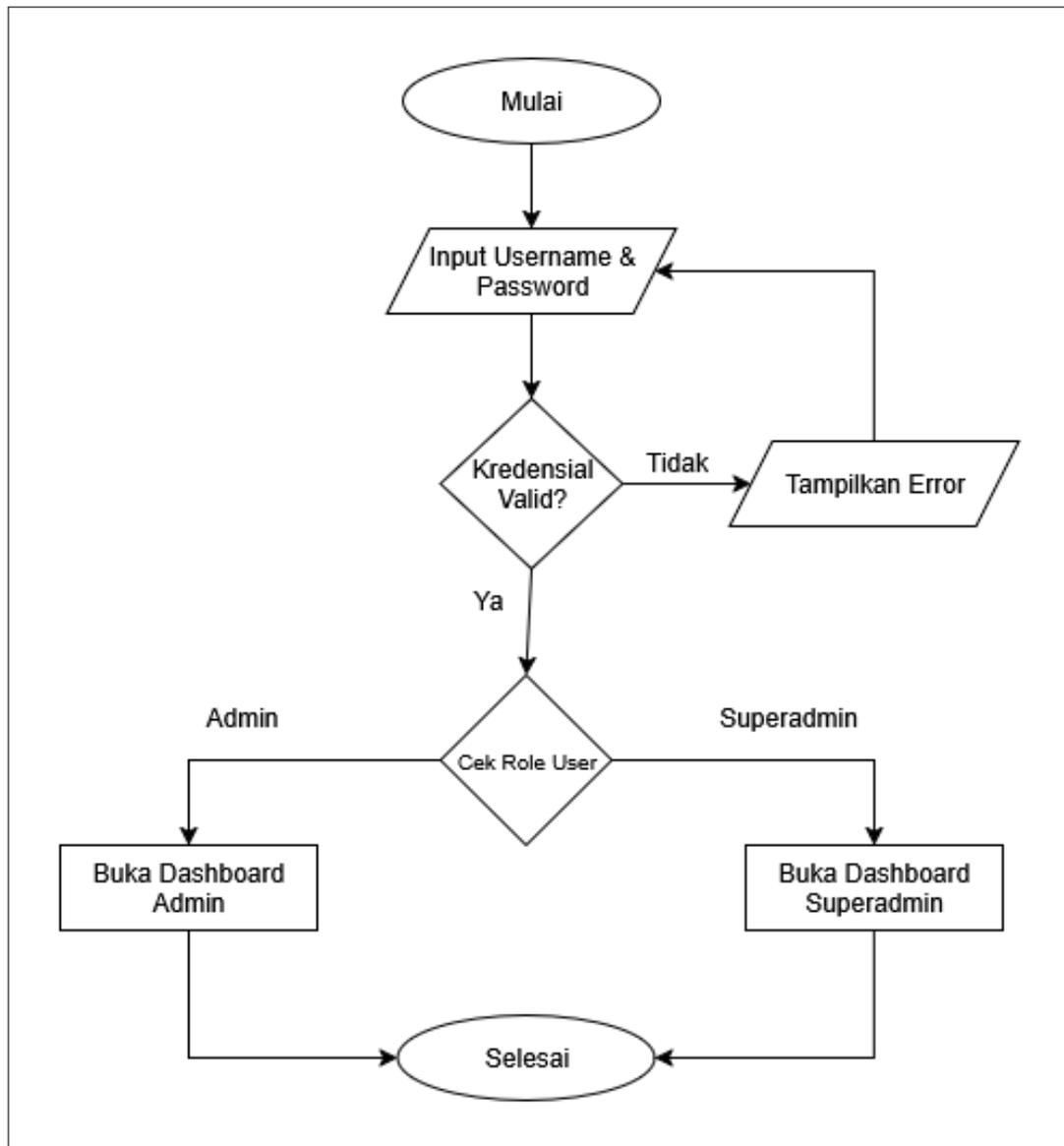
Gambar 3.2. Struktur Database Website Administrasi KGB

### C Flowchart

Flowchart adalah alat penting dalam perancangan sistem yang membantu memvisualisasikan alur kerja dan interaksi antar komponen. Flowchart untuk website KGB dirancang untuk menggambarkan proses utama, mulai dari autentikasi pengguna hingga pengelolaan data pegawai, KGB, dan KP. Flowchart ini mencakup beberapa komponen utama:

- **Autentikasi Pengguna:** Proses login yang memeriksa kredensial pengguna dan menentukan peran (admin atau pegawai).

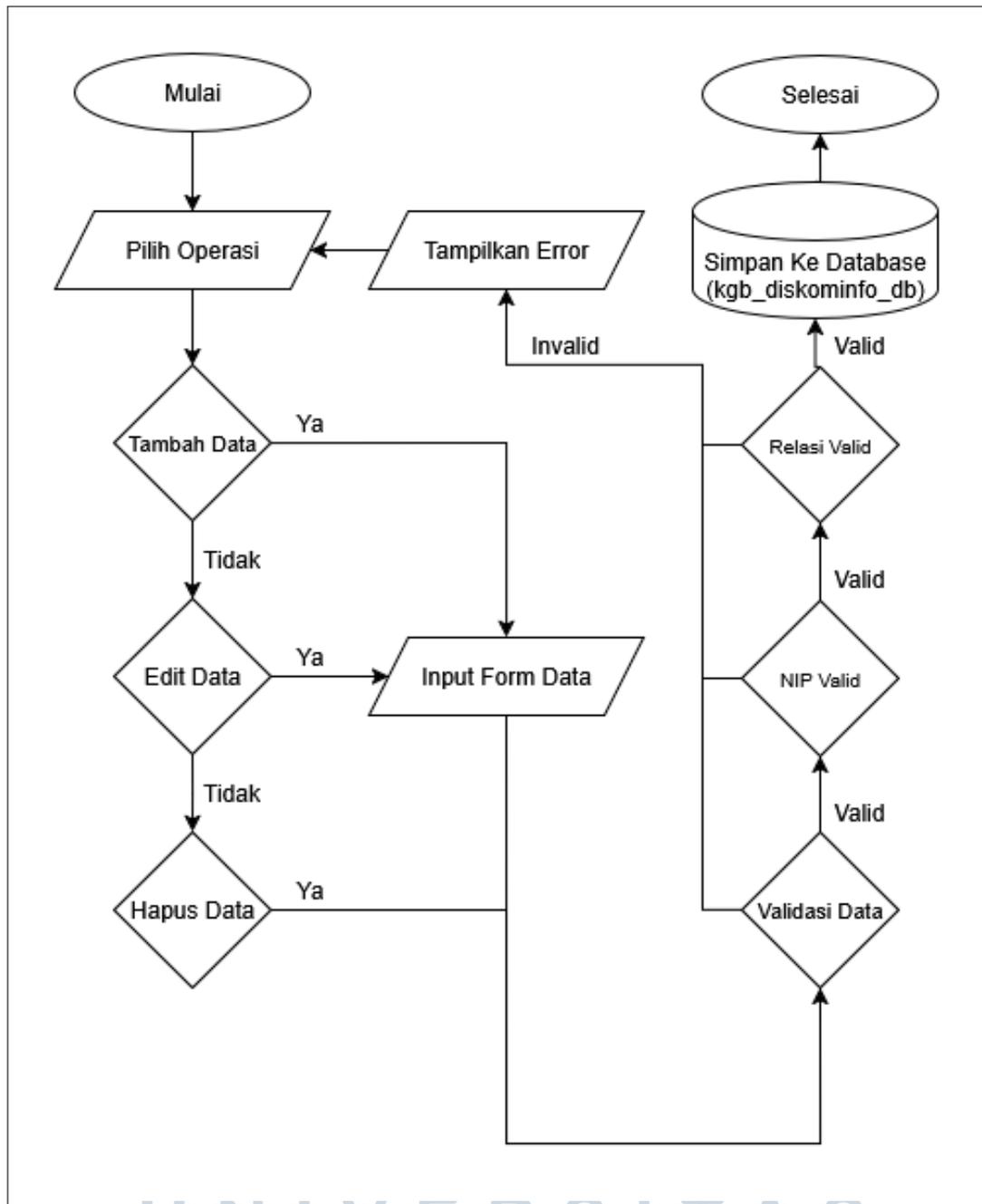
Proses login dibagi menjadi dua role, admin dan superadmin. Berikut adalah prosesnya:



Gambar 3.3. Flowchart Sistem Autentikasi Website KGB

- Manajemen Data Pegawai: Proses CRUD untuk data pegawai, termasuk penambahan, pengeditan, dan penghapusan data.

Proses manajemen data pegawai digunakan untuk pencatatan dan pengelolaan informasi pegawai dalam sistem KGB, untuk mengecek validitas data pegawai, serta untuk memastikan integritas data yang berkaitan dengan kenaikan gaji dan pangkat. Proses ini meliputi beberapa langkah penting, yaitu:



Gambar 3.4. Flowchart Manajemen Data Pegawai KGB

Flowchart manajemen data pegawai menggambarkan alur kerja sistem CRUD untuk tabel `master_pegawai`. Proses dimulai dengan pemilihan operasi yang akan dilakukan (Create, Read, Update, Delete). Untuk operasi Create dan Update, sistem mengarahkan pengguna ke form input yang mencakup field-field utama seperti NIP, nama, tanggal masuk tugas, serta relasi ke tabel

master lainnya (jabatan, SKPD, golongan, eselon).

Tahap validasi data meliputi beberapa pengecekan penting:

- Validasi NIP: Memastikan format NIP sesuai standar (18 digit) dan tidak duplikat dalam database
- Validasi Relasi: Memverifikasi bahwa `id_jabatan`, `id_skpd`, `id_golongan`, dan `id_eselon` yang dipilih valid dan terdapat dalam tabel master terkait
- Validasi Format: Mengecek format tanggal masuk tugas dan field wajib lainnya

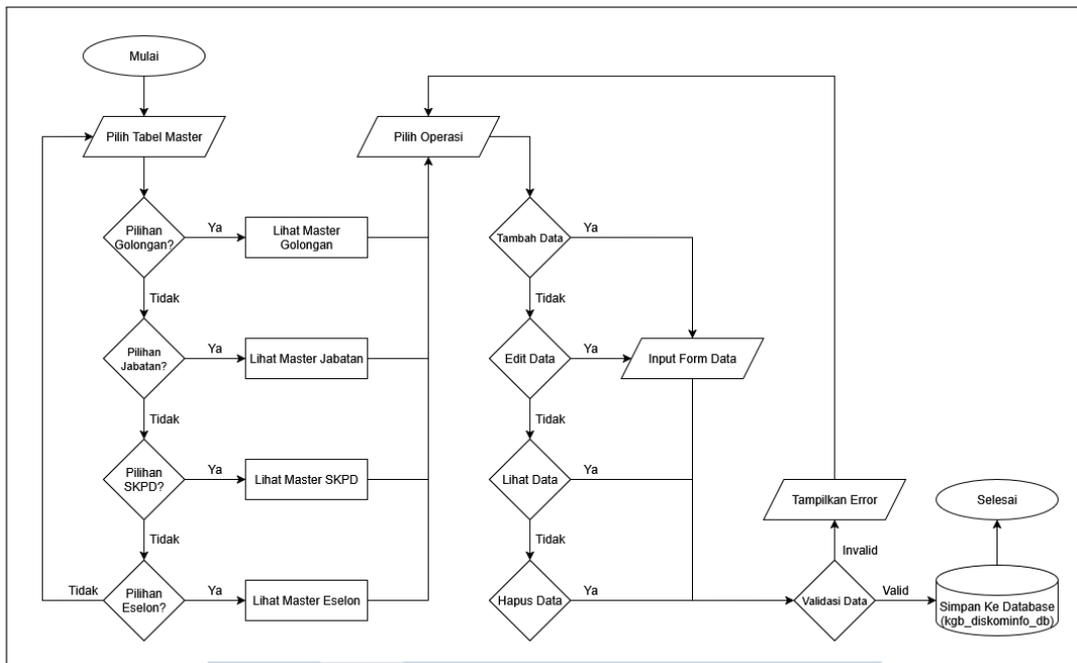
Jika validasi berhasil, data disimpan ke database dan sistem menampilkan notifikasi sukses. Apabila terjadi error pada tahap validasi, sistem akan menampilkan pesan error yang sesuai dan mengarahkan kembali ke form input untuk perbaikan. Untuk operasi Read dan Delete, sistem langsung melakukan validasi sebelum eksekusi query ke database.

Struktur tabel `master_pegawai` dengan relasi ke tabel master lainnya memungkinkan sistem melakukan pengecekan integritas referensial, memastikan data pegawai selalu konsisten dengan data master yang tersedia. Hal ini penting untuk menjaga kualitas data dalam sistem administrasi kepegawaian.

- Manajemen Data SKPD, Jabatan, Golongan, Eselon:

Proses manajemen data master meliputi operasi CRUD untuk tabel-tabel referensi utama dalam sistem, berikut flowchartnya:

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

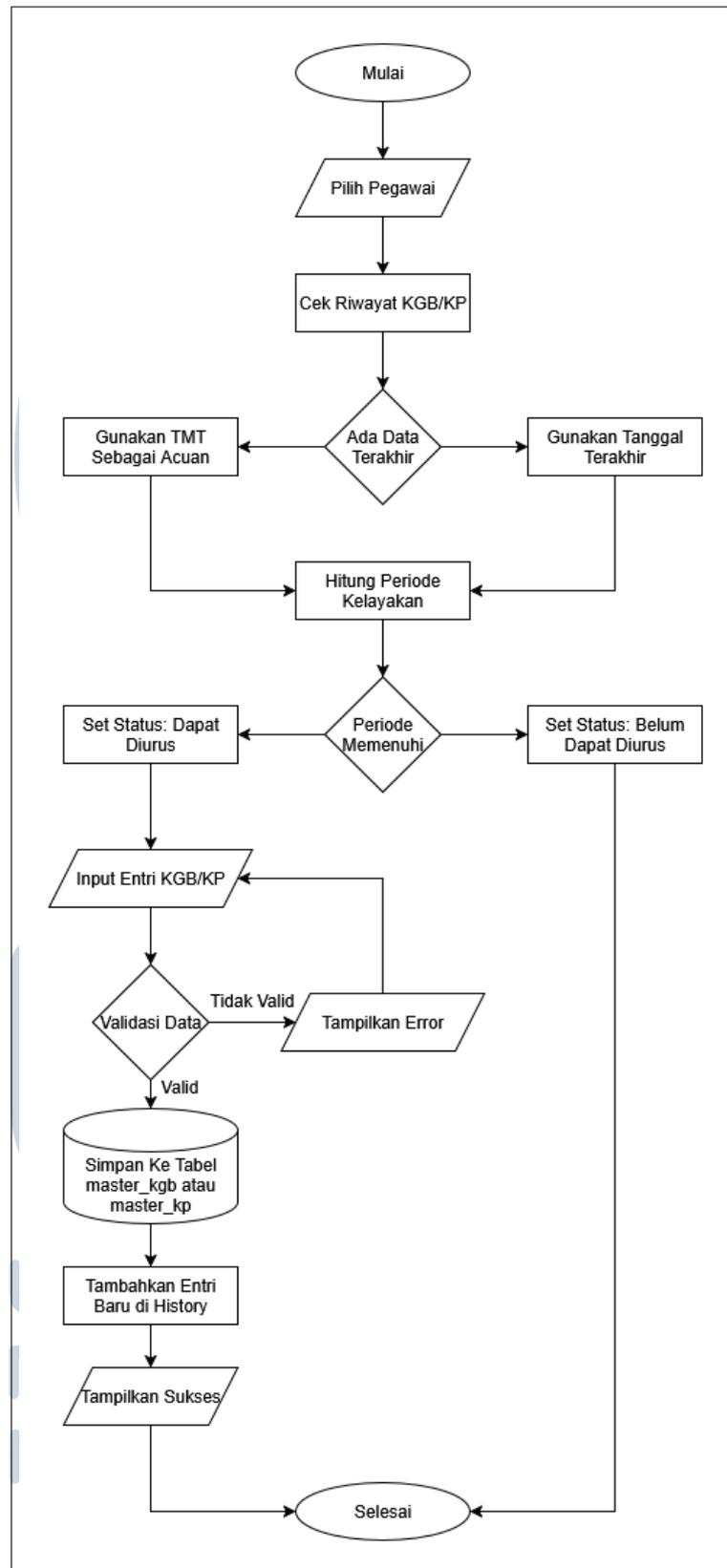


Gambar 3.5. Flowchart Manajemen Data Master (SKPD, Jabatan, Golongan, Eselon)

Flowchart ini menggambarkan proses pemilihan tabel master yang akan dikelola, diikuti dengan operasi CRUD yang dapat dilakukan pada masing-masing tabel. Setiap operasi memiliki validasi data sebelum disimpan ke database, dan penanganan error jika terjadi kesalahan. Karena data-data ini merupakan data referensi yang akan digunakan tabel pegawai, maka data-data ini harus di masukkan sebelum tabel pegawai. Tabel Jabatan harus dimasukkan setelah Tabel Golongan karena relasi antara keduanya, dimana setiap jabatan harus memiliki golongan yang sesuai. Tabel SKPD harus dimasukkan sebelum tabel Pegawai karena relasi antara keduanya, dimana setiap pegawai harus memiliki SKPD yang sesuai.

- Pengajuan KGB/KP: Proses pengajuan Kenaikan Gaji Berkala dan Kenaikan Pangkat, termasuk validasi dan persetujuan.

Proses pengajuan KGB dan KP merupakan alur bisnis inti dalam sistem administrasi kepegawaian. Sistem dirancang untuk mengotomatisasi perhitungan kelayakan pengajuan berdasarkan periode yang telah ditentukan, yaitu 2 tahun untuk KGB dan 4 tahun untuk KP. Berikut adalah flowchart untuk proses pengajuan KGB/KP:



Gambar 3.6. Flowchart Proses Pengajuan KGB/KP - Lengkap

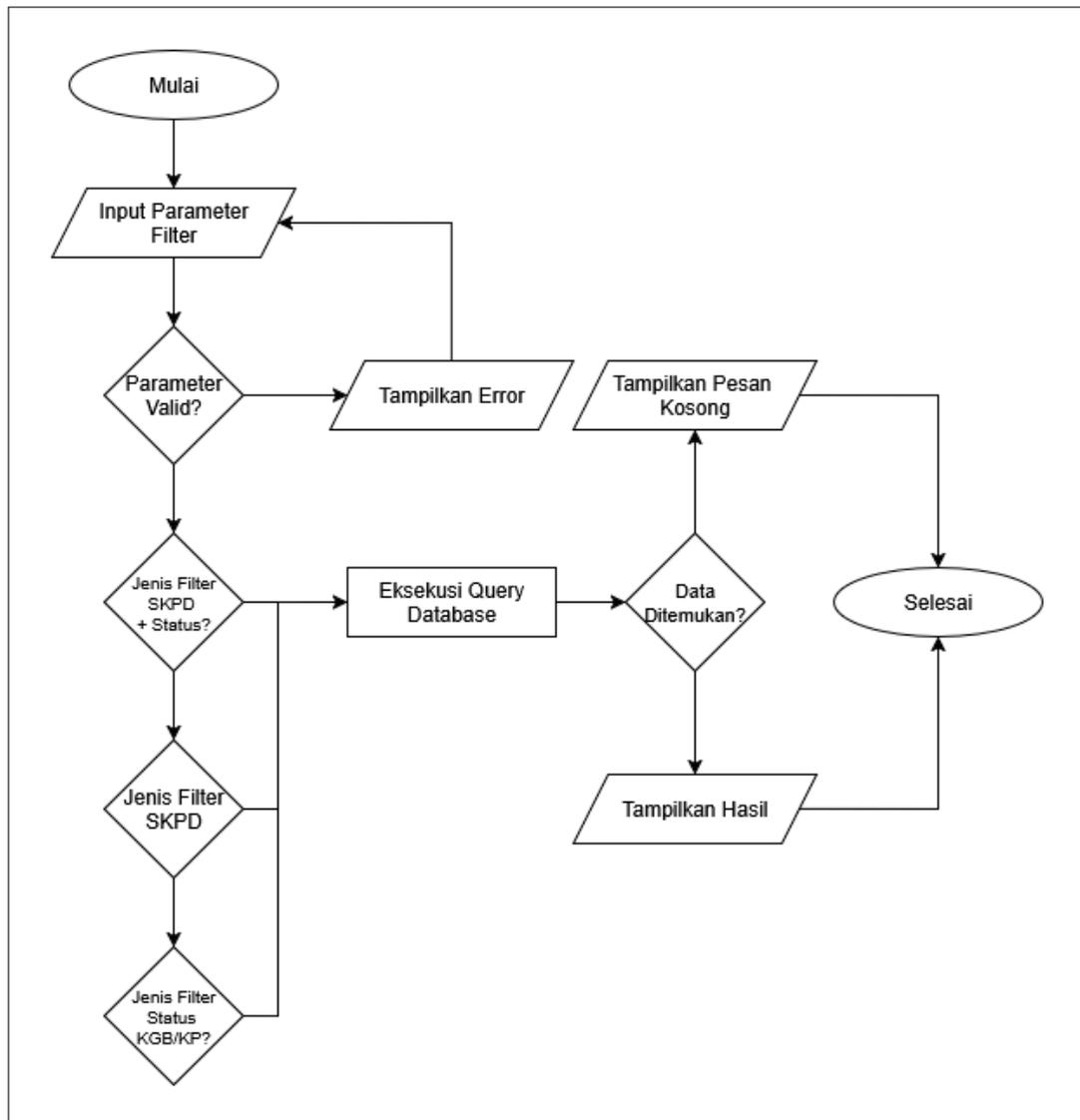
Flowchart pengajuan KGB/KP menggambarkan alur sistematis dalam mengelola proses kenaikan gaji dan pangkat pegawai. Proses dimulai dengan pemilihan pegawai yang akan diajukan KGB atau KP. Sistem kemudian melakukan pengecekan riwayat untuk menentukan acuan perhitungan periode kelayakan.

Untuk pegawai yang belum memiliki riwayat KGB/KP sebelumnya, sistem menggunakan Tanggal Masuk Tugas (TMT) sebagai acuan perhitungan. Sementara untuk pegawai yang sudah memiliki riwayat, sistem akan menggunakan tanggal terakhir pengajuan yang tercatat dalam database. Hal ini memastikan tidak ada duplikasi pengajuan dalam periode yang sama. Tahap perhitungan periode kelayakan melibatkan algoritma yang membandingkan tanggal acuan dengan tanggal saat ini. Untuk KGB, periode minimal adalah 2 tahun, sedangkan untuk KP adalah 4 tahun. Sistem akan menampilkan status "Dapat Diurus" jika periode telah memenuhi syarat, atau "Belum Dapat Diurus" jika masih dalam masa tunggu.

Jika pengajuan memenuhi syarat, sistem akan membuat entri baru dalam tabel KGB atau KP. Proses validasi data mencakup pengecekan format tanggal, kelengkapan informasi pegawai, dan konsistensi data dengan riwayat sebelumnya. Setelah validasi berhasil, data disimpan ke database dan riwayat diperbarui secara otomatis. Sistem memberikan feedback kepada pengguna melalui pesan sukses atau error yang sesuai dengan hasil operasi. Jika terjadi error pada tahap validasi, sistem akan menampilkan pesan kesalahan yang spesifik dan mengarahkan kembali ke form input untuk perbaikan data. Hal ini membantu dalam tracking dan monitoring proses administrasi kepegawaian.

- **Filtering Data:** Fitur untuk menyaring data berdasarkan SKPD atau status KGB/KP.

Sistem filtering data dirancang untuk memungkinkan pengguna menyaring informasi berdasarkan kriteria tertentu seperti SKPD atau status KGB/KP. Proses ini melibatkan validasi parameter filter, penerapan kondisi pencarian, dan pengembalian hasil yang sesuai. Berikut adalah flowchart untuk proses filtering data:



Gambar 3.7. Flowchart Sistem Filtering Data

Flowchart filtering data menggambarkan proses penyaringan informasi dalam sistem administrasi KGB. Proses dimulai dengan input parameter filter dari pengguna, yang dapat berupa SKPD tertentu, status KGB/KP (seperti "Dapat Diurus" atau "Belum Dapat Diurus"), atau kombinasi keduanya. Tahap validasi parameter mencakup pengecekan format dan keberadaan nilai filter yang dipilih. Sistem akan memverifikasi apakah SKPD yang dipilih valid dan terdapat dalam database, serta memastikan status yang dipilih sesuai dengan nilai yang diizinkan.

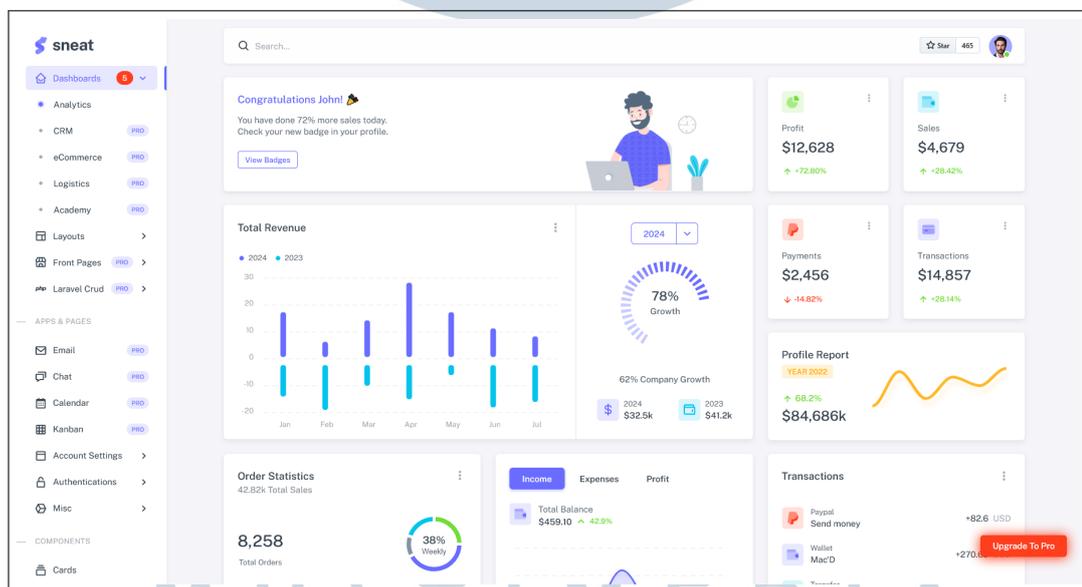
Berdasarkan jenis filter yang dipilih, sistem akan membuat query database yang sesuai:

- Filter SKPD: Query akan menambahkan kondisi `WHERE id_skpd = ?` untuk membatasi data pada SKPD tertentu
- Filter Status KGB/KP: Query akan menambahkan kondisi berdasarkan perhitungan tanggal untuk menentukan kelayakan pengajuan
- Filter Kombinasi: Query akan menggabungkan kedua kondisi dengan operator `AND`

Setelah eksekusi query, sistem akan memeriksa apakah data ditemukan. Jika ada hasil, data akan ditampilkan dalam format tabel dengan informasi lengkap pegawai dan status KGB/KP-nya. Jika tidak ada data yang sesuai dengan filter, sistem akan menampilkan pesan bahwa tidak ada data yang ditemukan dengan kriteria yang ditentukan.

## D Desain Website KGB

Perancangan UI/UX website KGB dilakukan dengan pendekatan *mobile-first design* untuk memastikan responsivitas yang optimal di berbagai perangkat.



Gambar 3.8. Demo Laravel Sneat - Free Version

Sumber: [19]

Laravel Sneat adalah template admin yang digunakan sebagai dasar pengembangan website administrasi KGB. Template ini menyediakan berbagai komponen UI yang siap pakai, seperti tabel, form, dan notifikasi, sehingga

mempercepat proses pengembangan. Selain itu, Laravel Sneat telah terintegrasi dengan Laravel Breeze untuk sistem autentikasi dan Yajra DataTables untuk manajemen tabel interaktif.

Pemilihan template Sneat didasarkan pada beberapa pertimbangan teknis yang mendukung kebutuhan proyek. Template ini dikembangkan oleh Themeselection dengan fokus pada pengalaman pengembang yang optimal dan dokumentasi yang lengkap. Arsitektur template mengikuti best practice Laravel dengan struktur folder yang terorganisir dan penggunaan komponen Blade yang modular. Hal ini memungkinkan kustomisasi yang fleksibel tanpa mengganggu integritas sistem keseluruhan. Integrasi dengan Laravel Breeze memberikan foundation autentikasi yang solid dengan fitur login dan register untuk website KGB. Sistem ini menggunakan session-based authentication yang sesuai dengan kebutuhan aplikasi web internal pemerintah. Selain itu, middleware autentikasi telah dikonfigurasi untuk mendukung sistem role-based access control yang diperlukan dalam lingkungan administratif. Selain itu, template ini dilengkapi dengan fitur-fitur unggulan yang mendukung pengembangan aplikasi berkualitas tinggi:

- Berbasis Bootstrap 5 dengan desain responsif penuh
- Kompatibel dengan Laravel 11 terbaru
- *Vertical layout* yang intuitif dan mudah dinavigasi
- Struktur folder yang terorganisir dengan kode yang bersih dan terdokumentasi
- Dukungan SASS untuk kustomisasi styling yang fleksibel
- Halaman autentikasi yang sudah terintegrasi

Yajra DataTables yang sudah terintegrasi dalam template menyediakan *server-side processing* untuk menangani data dalam jumlah besar. *Package* ini secara otomatis menangani *pagination*, *sorting*, dan *searching* tanpa memerlukan konfigurasi tambahan yang kompleks. Fitur ini sangat penting mengingat volume data pegawai yang cukup besar di lingkungan pemerintah kota.

Versatilitas Sneat memungkinkan pengembangan berbagai jenis aplikasi web, mulai dari platform SaaS, aplikasi manajemen proyek, *backend e-commerce*, sistem CRM, hingga aplikasi analitik. Dengan desain yang sepenuhnya responsif, aplikasi yang dibangun menggunakan template ini akan tampil optimal di berbagai perangkat, mulai dari desktop, tablet, hingga perangkat mobile.

### 3.5.2 Implementasi Website KGB

Dalam implementasi website administrasi KGB, beberapa fitur utama yang dikembangkan meliputi:

#### A Manajemen Tabel Non-KGB dan Non-KP CRUD dengan *filtering* dan *search*

Implementasi tabel CRUD dilakukan dengan memanfaatkan Yajra DataTables, sebuah paket Laravel yang menangani proses server-side untuk DataTables jQuery. Paket ini dipilih karena kemampuannya dalam menangani data besar secara efisien sekaligus menyediakan berbagai fitur interaktif seperti pagination, sorting, dan pencarian otomatis. Proses pencarian (search) telah tersedia secara bawaan dalam DataTables, memungkinkan pengguna melakukan pencarian cepat di seluruh kolom tabel tanpa perlu pengembangan tambahan. Lalu, untuk semua DataTables diimplementasikan perubahan bahasa menjadi Bahasa Indonesia, memungkinkan antarmuka lebih mudah digunakan.

Untuk kebutuhan filtering yang lebih spesifik, dikembangkan mekanisme custom menggunakan Laravel. Filtering ini dirancang untuk menangani kebutuhan kompleks seperti penyaringan data berdasarkan SKPD tertentu atau rentang tanggal. Implementasinya dilakukan dengan menambahkan parameter tambahan pada query builder Laravel yang kemudian diproses di server sebelum data dikembalikan ke client-side DataTables. Pendekatan ini memungkinkan filtering yang lebih terarah dibandingkan fitur search bawaan yang bersifat global.

Integrasi Yajra DataTables dilakukan dengan membuat kode khusus yang menangani request AJAX dari DataTables. Controller ini bertanggung jawab untuk:

- Mengambil data dari database menggunakan Eloquent ORM
- Menerapkan parameter filtering yang dikirim dari client
- Memproses request sorting dan pagination
- Mengembalikan data dalam format yang diharapkan DataTables

Pada sisi tampilan, tabel-tabel utama seperti daftar pegawai, jabatan, dan SKPD diimplementasikan dengan antarmuka yang konsisten. Setiap tabel dilengkapi dengan:

- Tombol aksi untuk operasi CRUD
- Indikator visual untuk sorting aktif
- Kontrol pagination yang responsif
- Input search bawaan DataTables
- Komponen filtering custom yang dikembangkan khusus

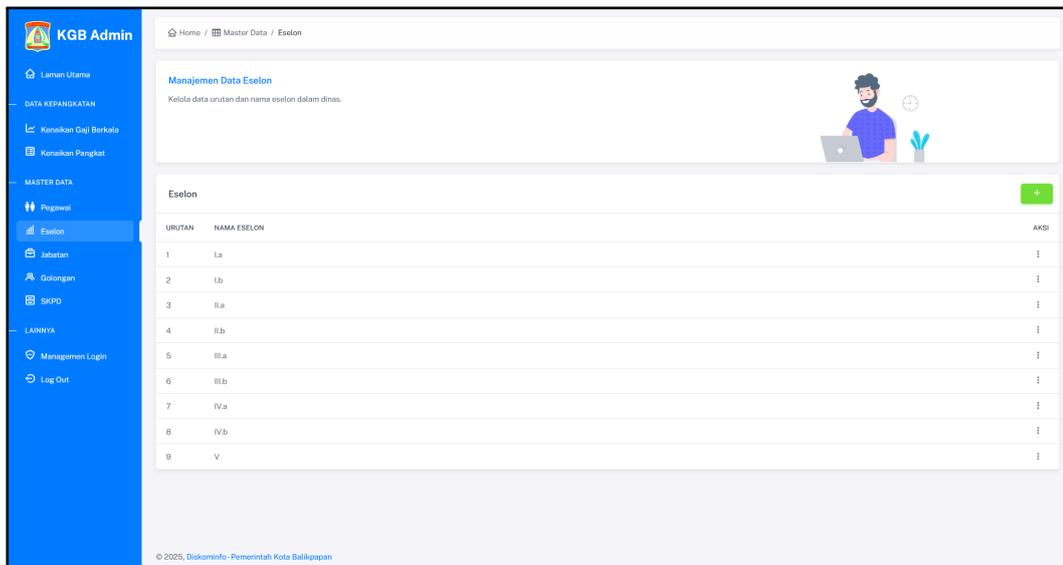
Penggunaan Yajra DataTables memberikan beberapa keuntungan signifikan. Pertama, beban pemrosesan data dialihkan ke server sehingga mengurangi beban di sisi client. Kedua, implementasi yang terstruktur melalui paket Laravel memudahkan maintenance dan pengembangan fitur tambahan. Terakhir, antarmuka yang dihasilkan tetap responsif meskipun menangani data dalam jumlah besar.

Struktur data dasar untuk tabel eselon dan golongan mengacu pada peraturan resmi pemerintah, khususnya jenjang eselon berdasarkan ketentuan birokrasi pemerintah Indonesia dan Golongan/Ruang mengacu pada PP No. 100 Tahun 2000 jo. PP No. 13 Tahun 2002 tentang Kenaikan Pangkat PNS [20].

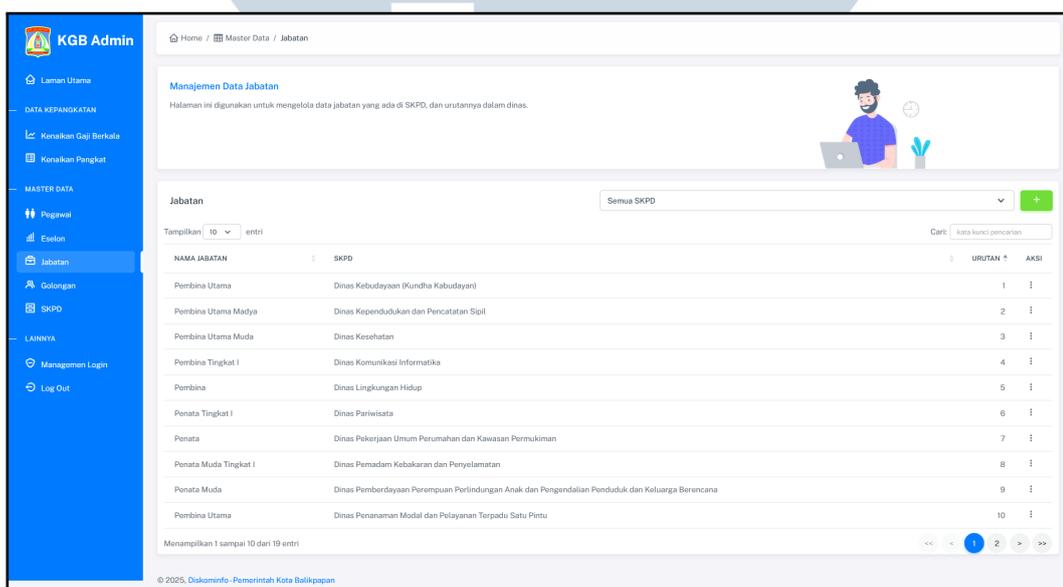
Berikut tampilan Data Table untuk berbagai tabel:

NIP	NAMA	JABATAN	SKPD	TANGGAL MASUK TUGAS	GOLONGAN	ESELON	KGB TERAKHIR	AKSI
00000000000000000001	Mr. Barrett Gleichner	Penata	Dinas Penanaman Modal dan Pelayanan Terpadu Satu Pintu	18 Oktober 2023	IV/a	IV.a	18 Oktober 2023	⋮
00000000000000000002	Prof. Flavio Hintz DDS	Penata Muda	Dinas Kebudayaan (Kuntha Kabudayan)	19 September 2022	III/a	III.b	19 September 2022	⋮
00000000000000000003	Prof. Samson Osinski	Penata Tingkat I	Dinas Pendidikan Pemuda dan Olahraga	07 Oktober 2020	IV/c	I.a	07 Oktober 2020	⋮
00000000000000000004	Mrs. Vickie Purdy DDS	Pembina Utama	Dinas Pendidikan Pemuda dan Olahraga	25 Juni 2024	IV/a	III.b	25 Juni 2024	⋮
00000000000000000005	Ayla Macejovic	Pembina	Dinas Perindustrian Koperasi Usaha Kecil dan Menengah	09 Januari 2024	IV/e	II.a	09 Januari 2024	⋮
00000000000000000006	Dr. Jerrod Glover PhD	Pembina Utama	Dinas Pemadam Kebakaran dan Penyelamatan	22 Desember 2021	III/b	V	22 Desember 2021	⋮
00000000000000000007	Rita Dicki	Penata Muda Tingkat I	Dinas Lingkungan Hidup	06 November 2023	III/c	II.b	06 November 2023	⋮
00000000000000000008	Verla Koopp	Pembina Tingkat I	Dinas Pemberdayaan Perempuan Perlindungan Anak dan Pengendalian Penduduk dan Keluarga Berencana	14 Maret 2022	IV/d	IV.b	14 Maret 2022	⋮

Gambar 3.9. Tampilan Tabel Pegawai

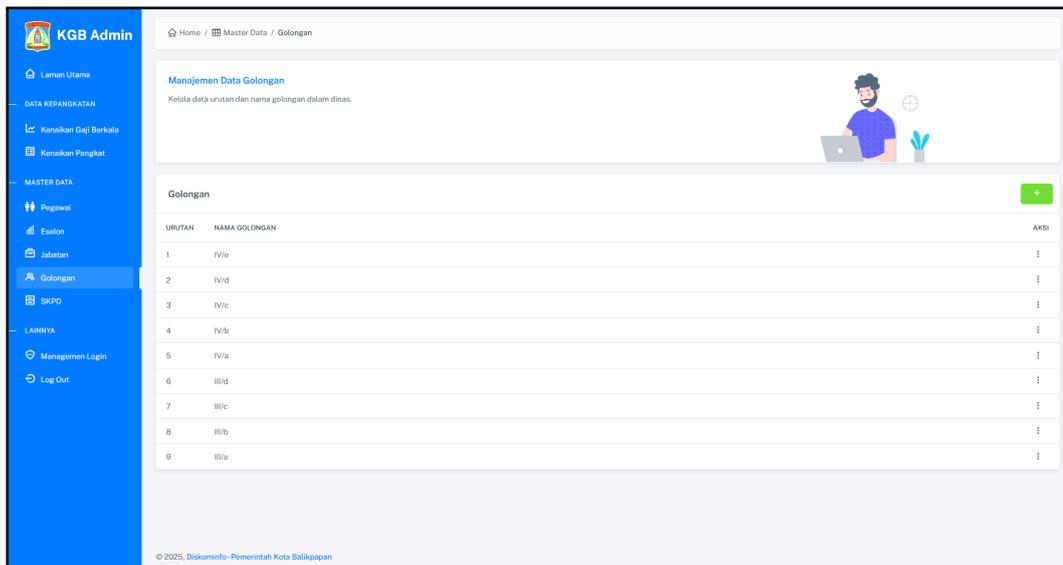


Gambar 3.10. Tampilan Tabel Eselon

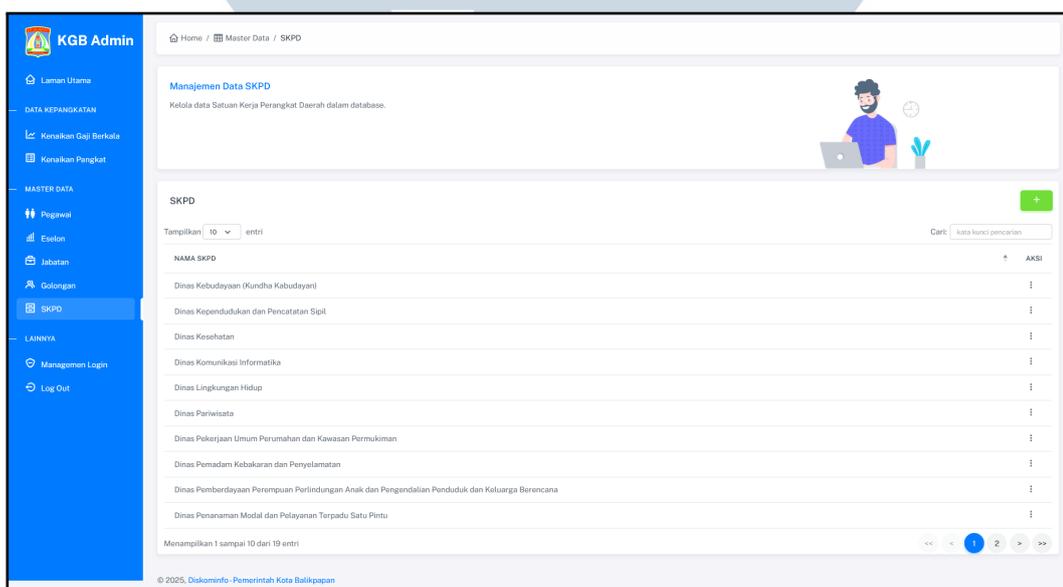


Gambar 3.11. Tampilan Tabel Jabatan

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.12. Tampilan Tabel Golongan



Gambar 3.13. Tampilan Tabel SKPD

## B CRUD KGB dan KP dengan fitur *history*

Pengembangan tabel Kenaikan Gaji Berkala (KGB) dan Kenaikan Pangkat (KP) dilengkapi dengan sistem manajemen riwayat yang komprehensif. Implementasi ini mencakup mekanisme otomatis untuk menentukan kelayakan pengajuan berdasarkan periode waktu, serta penyediaan riwayat lengkap setiap pencatatan yang dilakukan.

Sistem dilengkapi dengan *dropdown filter* yang memungkinkan pengguna untuk menyaring data berdasarkan status KGB/KP, khususnya untuk melihat apakah suatu pengajuan sudah dapat diurus atau belum. Filter ini bekerja dengan menganalisis tanggal terakhir pengajuan dan membandingkannya dengan periode berlaku (2 tahun untuk KGB dan 4 tahun untuk KP).

- Untuk pegawai tanpa data KGB/KP, sistem menghasilkan entri virtual berdasarkan TMT (Tanggal Masuk Tugas)
- Data virtual ditampilkan di *Datatables* tanpa disimpan ke database
- Berfungsi sebagai pengingat otomatis untuk pegawai baru

Fitur CRUD menyediakan fungsi manajemen riwayat yang lengkap:

Tabel 3.5. Fitur Manajemen Riwayat KGB/KP

<b>Fungsi</b>	<b>Deskripsi</b>
Tambah	Membuat entri baru saat proses KGB/KP selesai
Edit	Memperbaiki data yang sudah ada
Hapus	Menghapus entri yang tidak valid

Mekanisme perhitungan otomatis mencakup:

- Evaluasi periodik berdasarkan tanggal terakhir pengajuan
- Pencatatan lengkap setiap transaksi yang dilakukan
- Validasi ketat untuk mencegah duplikasi pengajuan

Antarmuka pengguna didesain dengan:

- Kode warna berbeda untuk berbagai status
- Tampilan kronologis riwayat pengajuan
- Informasi *countdown* pengajuan berikutnya

Berikut tampilan KGB dan KP dengan *history*:

**Manajemen Data KGB**  
Kelola data Kenaikan Gaji Berkala yang terjadi setiap dua tahun dalam database.

Kenaikan Gaji Berkala: Semua SKPD, Semua Status

Tampilkan: 10 entri

NAMA PEGAWAI	TANGGAL	KGB SELANJUTNYA	STATUS KGB	SISA HARI	AKSI
Aaa	31 Maret 2027	31 Maret 2029	Belum Waktunya	1451 hari	⋮
Abraham Pacocha	17 Juli 2023	17 Juli 2025	Belum Waktunya	98 hari	⋮
Adan Stokes	14 Maret 2023	14 Maret 2025	Harus Diurus	-	⋮
Adella Schulist Jr.	15 Agustus 2021	15 Agustus 2023	Harus Diurus	-	⋮
Ahmad Lebsack Jr.	20 November 2022	20 November 2024	Harus Diurus	-	⋮
Aisha Hauck	01 Maret 2023	01 Maret 2025	Harus Diurus	-	⋮
Alec Beahan	14 Maret 2023	14 Maret 2025	Harus Diurus	-	⋮

Gambar 3.14. Tampilan Tabel Kenaikan Gaji Berkala

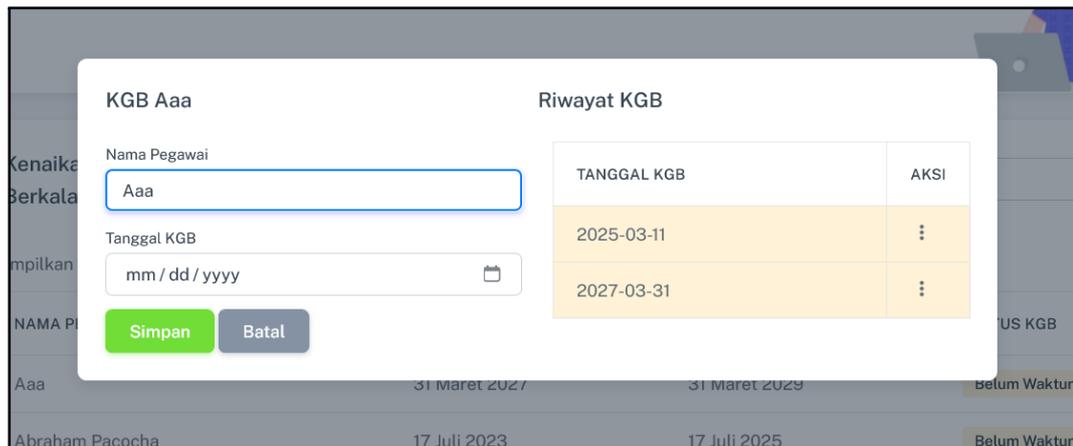
**Manajemen Data Kenaikan Pangkat**  
Kelola data Kenaikan Pangkat yang terjadi setiap empat tahun dalam database.

Kenaikan Pangkat: Semua SKPD, Semua Status

Tampilkan: 10 entri

NAMA PEGAWAI	TANGGAL	KENAIKAN PANGKAT SELANJUTNYA	STATUS KENAIKAN PANGKAT	SISA HARI	AKSI
Aaa	12 Maret 2025	12 Maret 2029	Belum Waktunya	1432 hari	⋮
Abraham Pacocha	11 Maret 2025	11 Maret 2029	Belum Waktunya	1431 hari	⋮
Adan Stokes	14 Maret 2023	14 Maret 2027	Belum Waktunya	703 hari	⋮
Adella Schulist Jr.	15 Agustus 2021	15 Agustus 2025	Belum Waktunya	127 hari	⋮
Ahmad Lebsack Jr.	20 November 2022	20 November 2026	Belum Waktunya	589 hari	⋮
Aisha Hauck	01 Maret 2023	01 Maret 2027	Belum Waktunya	690 hari	⋮
Alec Beahan	14 Maret 2023	14 Maret 2027	Belum Waktunya	703 hari	⋮

Gambar 3.15. Tampilan Tabel Kenaikan Pangkat



Gambar 3.16. Tampilan tabel History untuk setiap KGB dan KP

### C Sistem login dan roles users

Sistem login dikembangkan dengan menggunakan Laravel Breeze sebagai starter kit untuk autentikasi. Terdapat dua jenis role yang diimplementasikan, yaitu admin dan superadmin, dengan level akses yang berbeda sesuai kebutuhan operasional sistem.

Role admin dirancang dengan hak akses terbatas untuk keperluan monitoring data. Admin hanya dapat mengakses data kepangkatan, termasuk tabel Kenaikan Gaji Berkala (KGB) dan Kenaikan Pangkat (KP). Pada bagian master data, admin diperbolehkan melihat data pegawai dan jabatan.

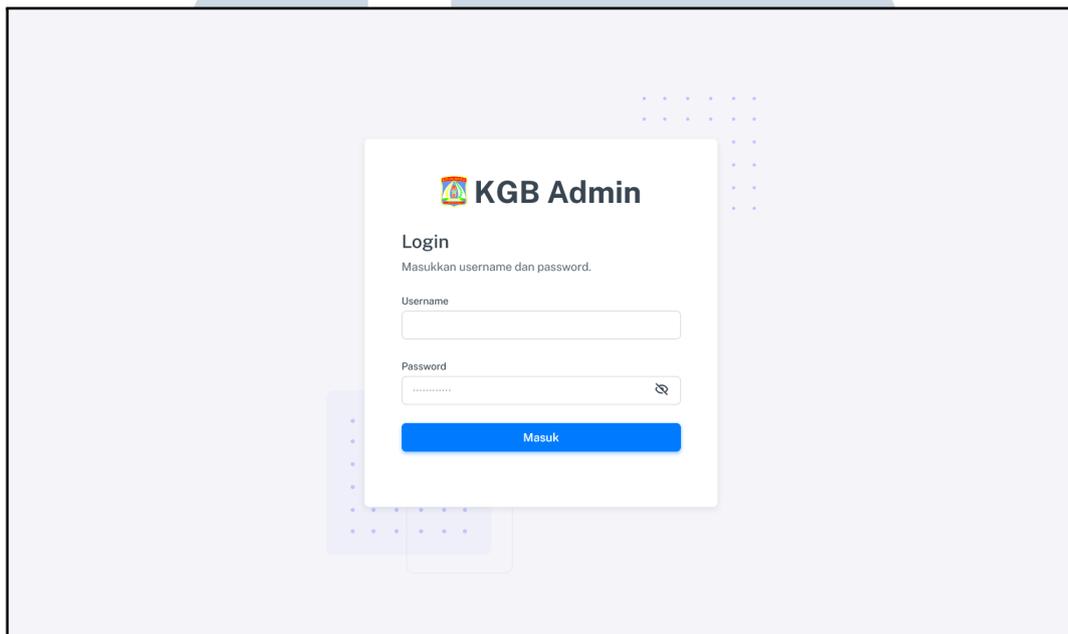
Sementara itu, role superadmin memiliki hak akses penuh terhadap seluruh fitur sistem. Superadmin dapat mengelola semua data, termasuk menambah, mengedit, dan menghapus entri pada tabel master data seperti pegawai, jabatan, eselon, golongan, dan SKPD. Selain itu, superadmin memiliki akses eksklusif untuk mengelola tabel users, yang meliputi:

- Membuat akun login baru
- Menentukan role (admin atau superadmin) untuk setiap user
- Mengatur privileges atau hak akses
- Melakukan update atau penghapusan akun

Implementasi pembatasan akses dilakukan melalui middleware Laravel yang memeriksa role user sebelum mengizinkan akses ke suatu route. Setiap

halaman dan fungsi dalam sistem telah dilengkapi dengan pengecekan role, sehingga mencegah akses tidak sah ke fitur yang tidak diperbolehkan. Selain itu, antarmuka sidebar dan menu navigasi secara otomatis menyesuaikan dengan role yang dimiliki oleh user yang login, sehingga hanya menampilkan opsi-opsi yang relevan.

Berikut tampilan login dan perbedaan *sidebar* antara admin dan superadmin:



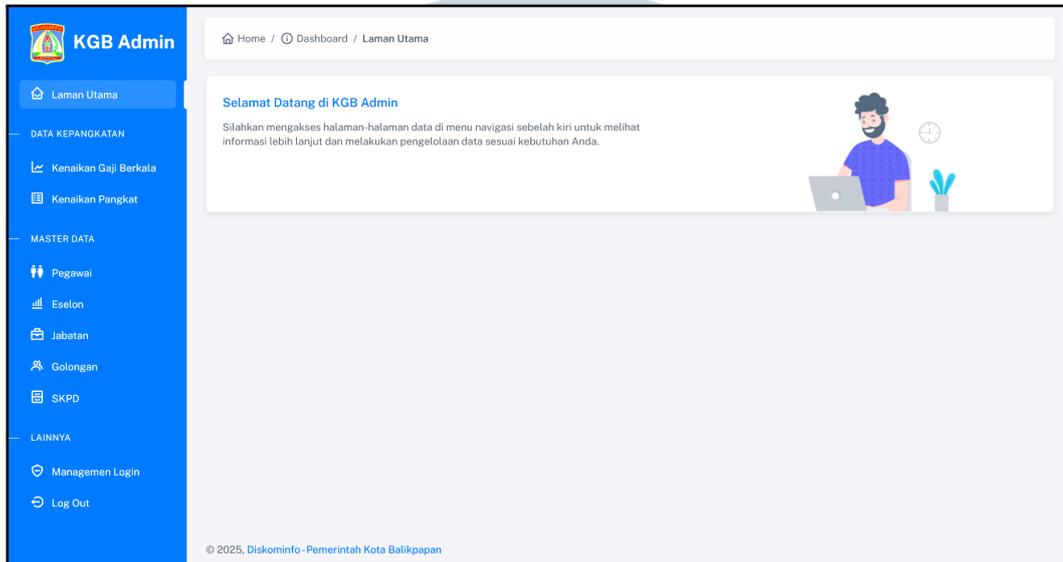
Gambar 3.17. Tampilan Login

#### **D Sistem *filtering* data berdasarkan SKPD login**

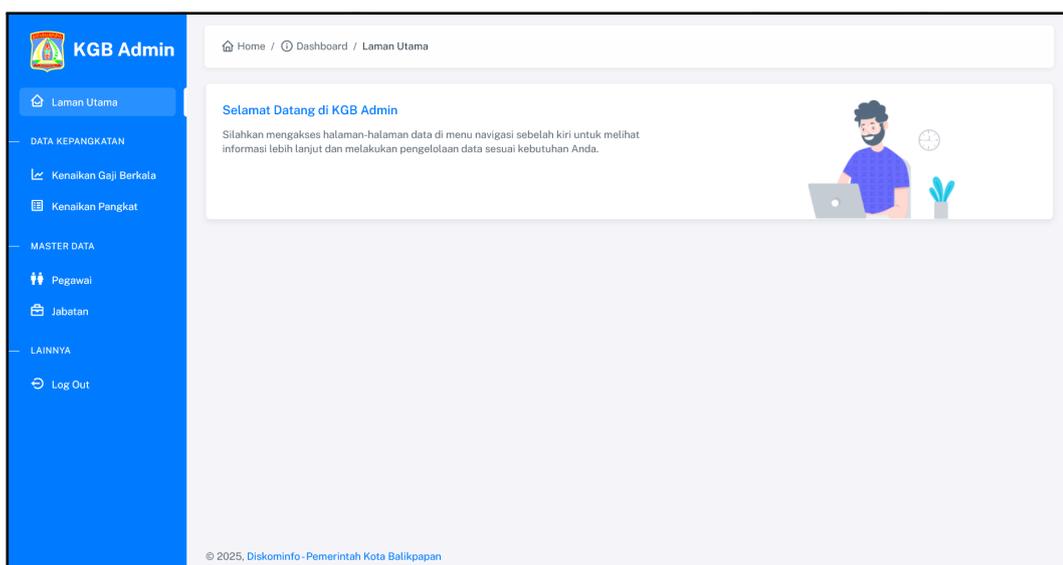
Sistem *filtering* berdasarkan SKPD (Satuan Kerja Perangkat Daerah) login dikembangkan untuk membatasi akses data sesuai dengan unit kerja pengguna. Mekanisme ini dirancang untuk memenuhi kebutuhan operasional dimana setiap pengguna memiliki lingkup wewenang yang berbeda dalam mengakses data pegawai.

Setiap akun login dalam sistem terhubung dengan data SKPD tertentu melalui field `id_skpd` pada tabel `users`. Terdapat dua skenario akses yang diimplementasikan:

- User dengan SKPD Tertentu
  - Hanya dapat melihat data pegawai dari SKPD yang sama



Gambar 3.18. Tampilan Superadmin



Gambar 3.19. Tampilan Admin

- Filtering otomatis pada tampilan data pegawai, KGB, dan KP
- Tidak memiliki akses ke data SKPD lain
- User Tanpa SKPD (null)
  - Akses penuh ke semua data dari seluruh SKPD
  - Tidak ada pembatasan dalam melihat data
  - Biasanya diberikan kepada superadmin atau admin pusat

Pengembangan sistem filtering memberikan manfaat penting:

- Menjaga kerahasiaan data antar SKPD
- Menerapkan prinsip *need-to-know*
- Mencegah akses tidak sah ke data sensitif
- Menyederhanakan tampilan data relevan

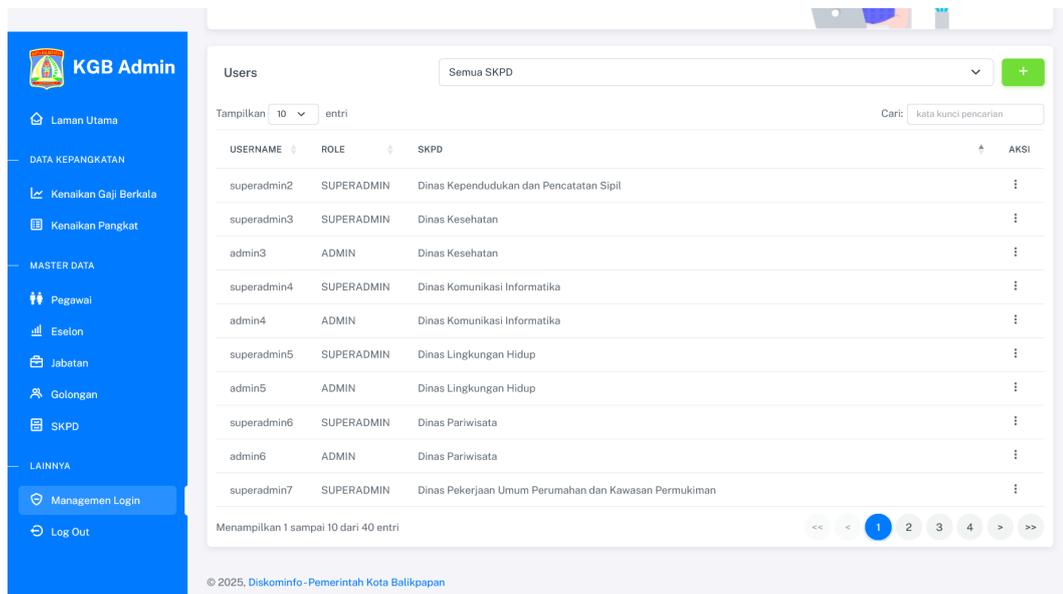
Dengan implementasi ini, sistem mampu memberikan pengalaman pengguna yang aman dan efisien sesuai struktur organisasi pemerintah daerah.

Berikut contohnya jika login dengan SKPD Dinas Kependudukan dan Pencatatan Sipil dan data SKPD pada setiap login:

The screenshot shows the 'KGB Admin' interface. The sidebar menu includes 'Laman Utama', 'DATA KEPANGKATAN', 'MASTER DATA', and 'LAINNYA'. The 'MASTER DATA' section is expanded to show 'Pegawai'. The main content area displays 'Manajemen Data Pegawai' with a sub-header 'Kelola data pegawai dalam organisasi.' Below this is a table titled 'Pegawai' with a search bar and a 'Tampilkan' dropdown set to '10' entries. The table contains the following data:

NIP	NAMA	JABATAN	SKPD	TANGGAL MASUK TUGAS	GOLONGAN	ESELON	KGB TERAKHIR	AKSI
0000000000000000017	Juston Douglas	Pembina Utama Muda	Dinas Kependudukan dan Pencatatan Sipil	17 November 2021	IV/e	IV.a	17 November 2021	⋮
0000000000000000038	Kari Walter	Penata	Dinas Kependudukan dan Pencatatan Sipil	31 Oktober 2021	IV/a	II.b	31 Oktober 2021	⋮
0000000000000000097	Milo Rau	Pembina Utama Muda	Dinas Kependudukan dan Pencatatan Sipil	23 Maret 2024	III/c	II.b	23 Maret 2024	⋮
0000000000000000103	Mrs. Caterina Stark MD	Pembina Utama Madya	Dinas Kependudukan dan Pencatatan Sipil	10 Februari 2025	IV/e	II.b	10 Februari 2025	⋮
0000000000000000116	Vito Boyle	Pembina Utama	Dinas Kependudukan dan Pencatatan Sipil	12 Juli 2024	III/a	III.b	12 Juli 2024	⋮
0000000000000000162	Joshuah Mueller	Penata Muda	Dinas Kependudukan dan Pencatatan Sipil	05 Juli 2023	III/b	Lb	05 Juli 2023	⋮

Gambar 3.20. Tampilan *Filtered Data* Sesuai SKPD Login



Gambar 3.21. Logins Dengan SKPD

Sistem mengimplementasikan filtering otomatis berdasarkan SKPD pengguna yang login untuk membatasi akses data.

### 3.5.3 Pengujian Website KGB

Pengujian Website Administrasi KGB dimulai dengan *unit testing backend* yang fokus pada validasi model *relationships* antara entitas User, SKPD, Pegawai, KGB, dan KP. Pengujian mencakup validasi *business logic* untuk perhitungan periode KGB dan KP, *testing middleware filtering SKPD*, serta pengujian *controller methods* untuk operasi CRUD. *Integration testing* dilakukan untuk memastikan integrasi *Laravel Eloquent* dengan *MySQL database*, validasi *Yajra DataTables* dengan *server-side processing*, dan pengujian konsistensi *filtering data SKPD* antar modul. Pengujian *frontend* mencakup *cross-browser compatibility testing* pada *Chrome, Firefox, Safari, dan Edge*, serta *responsive design testing* pada berbagai ukuran layar. *Performance testing* dilakukan dengan *load testing* menggunakan *100+ concurrent users*, optimisasi *query database*, dan *monitoring page load speed* menggunakan *Google Lighthouse*.

### 3.5.4 Deployment & Maintenance Website KGB

Deployment website administrasi KGB dilakukan oleh tim IT DATSI Diskominfo Balikpapan pada server produksi dengan domain resmi

<https://kgb.balikpapan.go.id/login>. Proses deployment menggunakan infrastruktur server pemerintah kota yang telah dikonfigurasi dengan standar keamanan tinggi dan sertifikat SSL untuk memastikan enkripsi data yang aman. Tim IT DATSI bertanggung jawab penuh dalam konfigurasi server, pengaturan database produksi, serta optimisasi performa aplikasi agar dapat menangani beban kerja operasional harian dengan baik.

Maintenance website KGB dilakukan secara berkala oleh tim IT DATSI dengan jadwal pemeliharaan rutin setiap minggu untuk memastikan sistem berjalan optimal. Kegiatan maintenance meliputi monitoring performa server, backup data reguler, update security patch, dan optimisasi database untuk menjaga kecepatan akses. Tim juga bertanggung jawab dalam menangani bug report dari pengguna, melakukan update fitur sesuai kebutuhan operasional, serta memberikan dukungan teknis kepada staff kepegawaian yang menggunakan sistem.

### **3.6 Frontend Aplikasi Mobile App Kota**

Aplikasi ini dikembangkan sebagai bagian awal dari pengembangan selanjutnya untuk aplikasi ini, maka ada banyak kustomisasi dari bentuk warna, perubahan elemen-elemen interaktif dan lain-lain. Aplikasi masyarakat ini adalah bagian dari inisiatif transformasi digital Pemerintah Kota Balikpapan untuk menyediakan akses mudah terhadap berbagai layanan publik. Tujuan utamanya adalah mempermudah masyarakat dalam mengakses informasi resmi, pengumuman penting, dan layanan administrasi kapan saja melalui smartphone. Aplikasi dirancang dengan antarmuka sederhana dan intuitif untuk menjangkau semua kalangan masyarakat, sekaligus mendukung efisiensi pelayanan pemerintah melalui kanal digital yang terintegrasi.

#### **3.6.1 Pembelajaran Flutter Mandiri**

Sebelum memulai pengembangan aplikasi mobile, dipelajari Flutter dari dasar karena sebelumnya hanya memiliki pengalaman menggunakan Kotlin dalam pembelajaran kuliah. Pembelajaran dilakukan secara mandiri selama satu minggu melalui seri video tutorial "*Fast Track Flutter Beginner*" oleh Kuldii Project [21], yang mencakup materi:

- Dasar pemrograman Dart: sintaks, tipe data, dan konsep asynchronous

- Pembangunan UI dengan Stateless dan Stateful Widget
- Manajemen state menggunakanGetX
- Integrasi API dan HTTP request
- Navigasi antar halaman

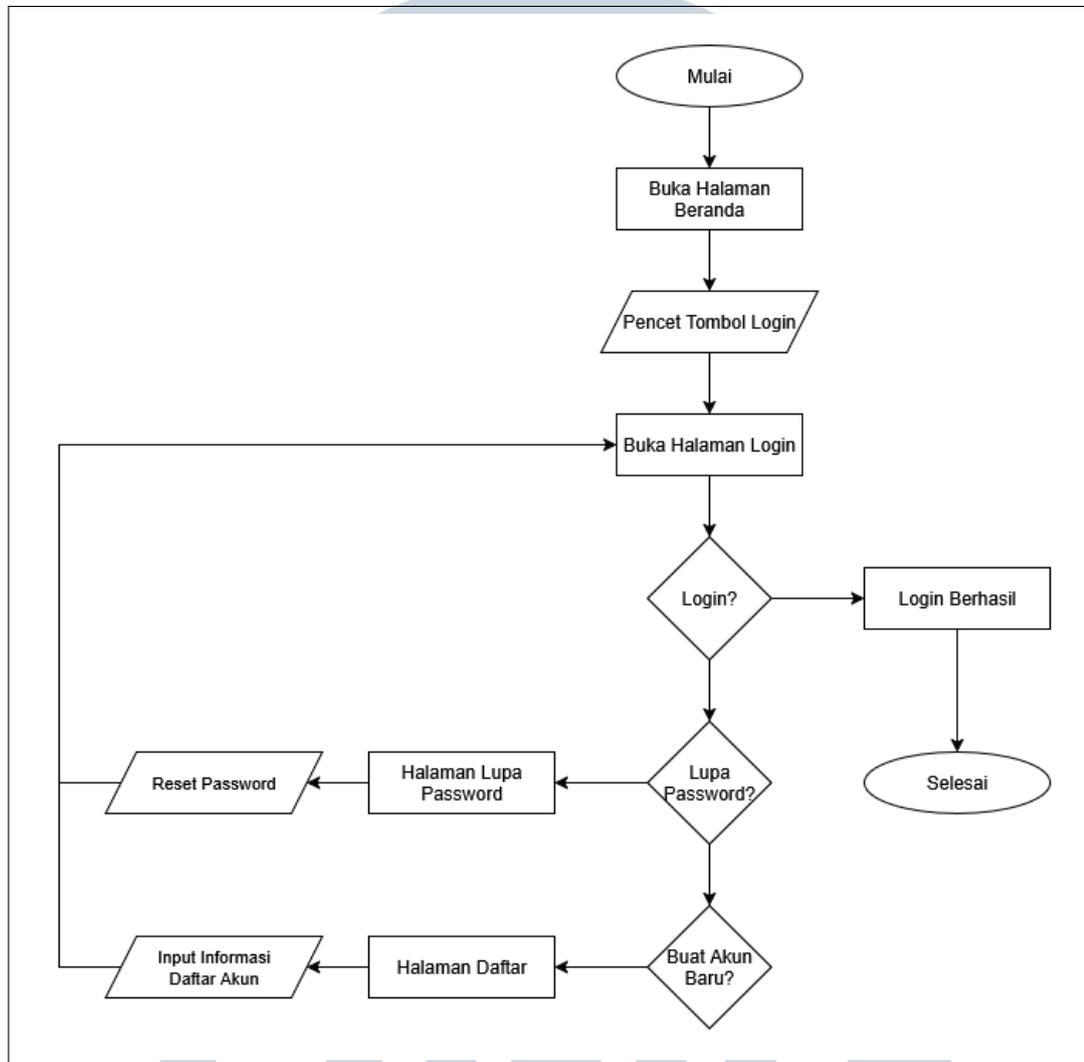
Setelah memahami konsep inti, pembelajaran langsung diterapkan dalam proyek dengan mengembangkan antarmuka aplikasi mobile menggunakan komponen seperti BottomNavigationBar, ListView, dan berbagai Custom Widgets. Pembelajaran intensif selama satu minggu ini memungkinkan untuk beradaptasi cepat dengan Flutter meskipun belum memiliki pengalaman sebelumnya dengan framework ini, dan hanya sebatas mengetahui pembelajaran Kotlin dalam pembelajaran materi kuliah.

### **3.6.2 Analisis Kebutuhan & Desain Sistem**

Analisis kebutuhan aplikasi mobile Kota Balikpapan dimulai dengan identifikasi tujuan utama sebagai platform digital yang menyediakan akses mudah terhadap berbagai layanan publik bagi masyarakat. Aplikasi dirancang untuk menjadi bagian dari inisiatif transformasi digital Pemerintah Kota Balikpapan yang memungkinkan masyarakat mengakses informasi resmi, pengumuman penting, dan layanan administrasi kapan saja melalui smartphone. Kebutuhan fungsional meliputi pengembangan antarmuka yang sederhana dan intuitif untuk menjangkau semua kalangan masyarakat, implementasi sistem navigasi yang user-friendly dengan BottomNavigationBar, serta integrasi komponen interaktif seperti carousel hero untuk menampilkan informasi utama, news cards untuk berita terkini, dan button rows untuk akses cepat ke layanan populer.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

## A Flowchart



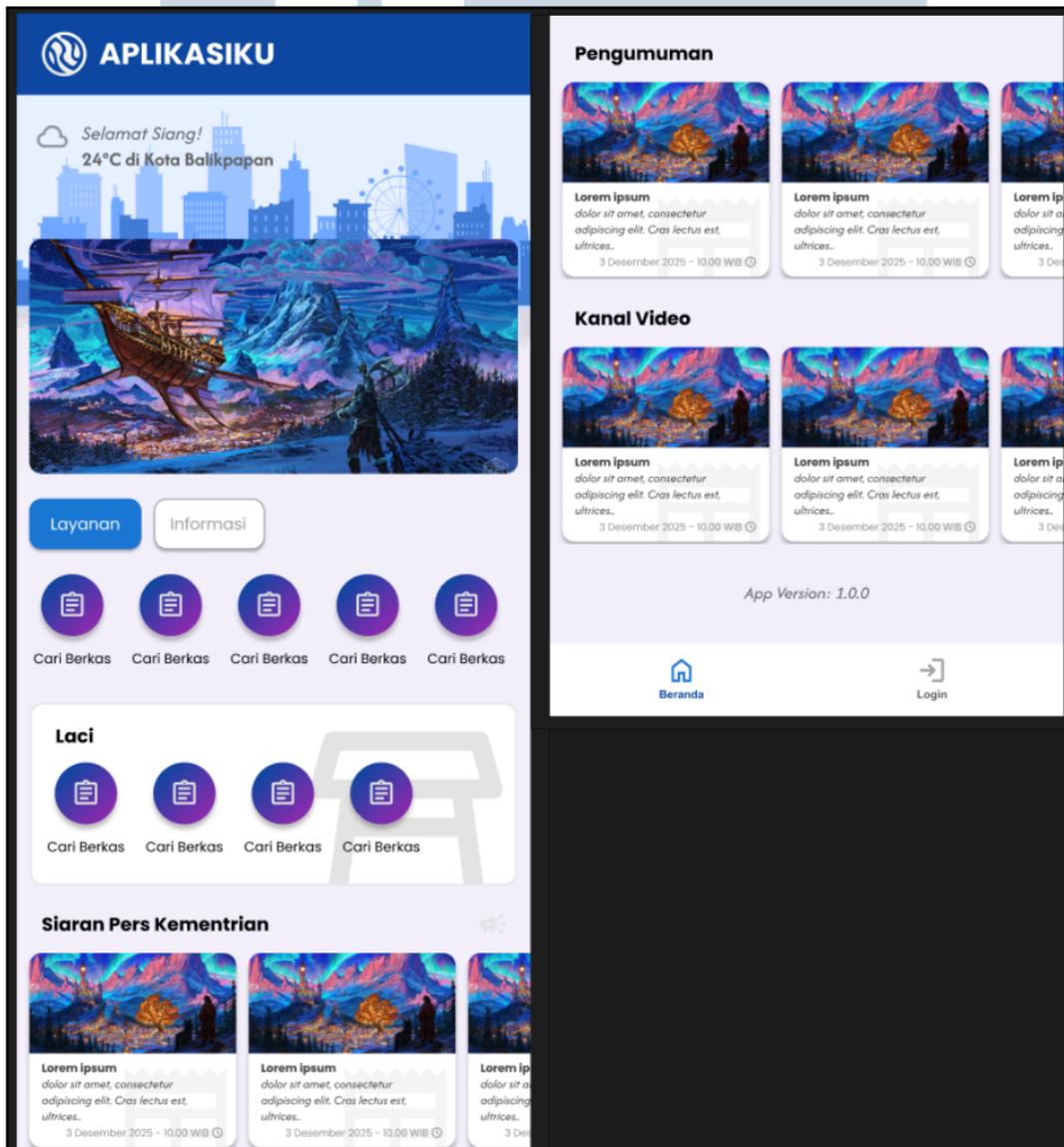
Gambar 3.22. Flowchart Navigasi Halaman Aplikasi Mobile Balikpapan

Berikut adalah alur navigasi aplikasi mobile Balikpapan yang dimulai dari pembukaan aplikasi, kemudian mengarahkan pengguna ke halaman beranda untuk memilih navigasi, dengan fokus utama pada proses autentikasi yang mencakup tiga jalur utama yaitu login langsung bagi pengguna yang sudah memiliki akun, pendaftaran akun baru bagi pengguna baru, dan fitur reset password bagi pengguna yang lupa kata sandi, dimana setiap jalur autentikasi menyediakan opsi untuk kembali ke tahap sebelumnya atau melanjutkan ke halaman beranda setelah berhasil login. Flowchart ini dirancang dengan pendekatan user-centered design yang memprioritaskan kemudahan navigasi dan fleksibilitas pengguna dalam mengakses

berbagai fitur autentikasi, serta menyediakan jalur yang jelas untuk setiap skenario yang mungkin dihadapi pengguna saat menggunakan aplikasi mobile Balikpapan.

## B Desain Aplikasi

Desain awal aplikasi dilakukan menggunakan Figma, yang memungkinkan pembuatan desain dengan mudah, dimana setiap elemen UI dirancang dengan mempertimbangkan prinsip-prinsip desain yang responsif dan intuitif. Karena membuat aplikasi kota, diambil banyak referensi dari aplikasi kota lain seperti Sapawarga[22] dan Jakarta Kini (JAKI)[23]. Desain ini mencakup:



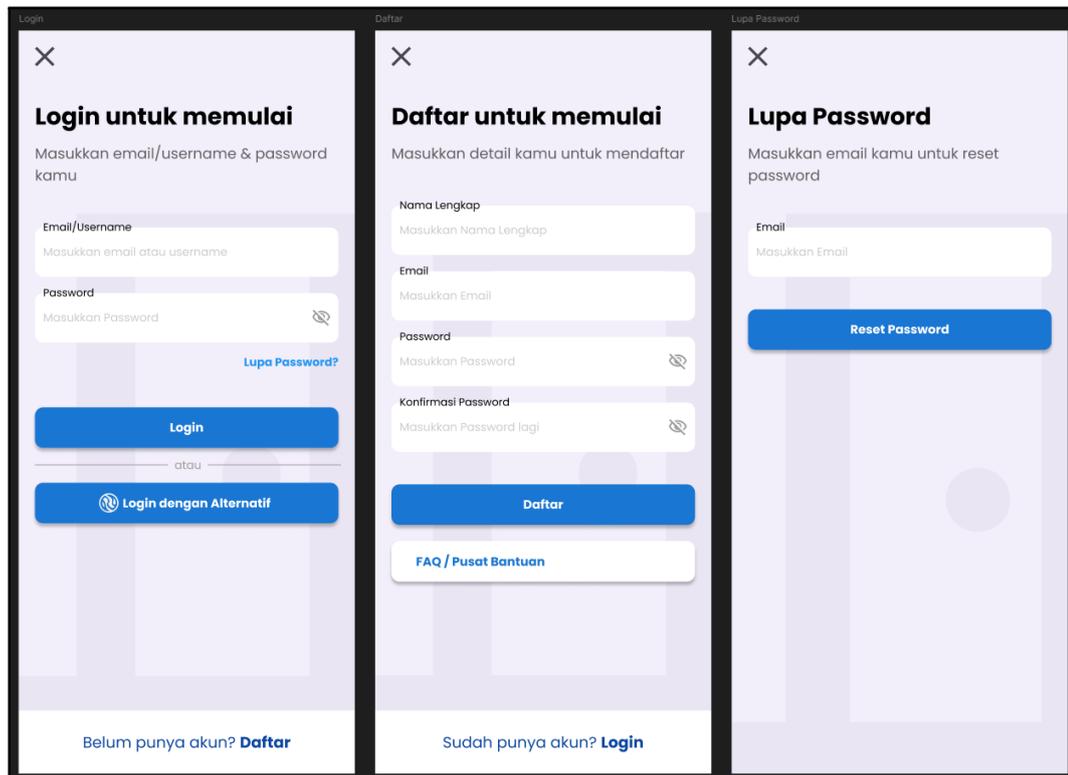
Gambar 3.23. Desain Figma Aplikasi Mobile Balikpapan - Bagian 1

Tampilan halaman depan aplikasi mobile Balikpapan ini bertujuan memberikan pengalaman pengguna yang informatif, interaktif, dan mudah dinavigasi. Pada bagian atas, pengguna langsung disambut dengan sapaan personal dan informasi cuaca terkini di Kota Balikpapan, lengkap dengan ilustrasi visual yang menyesuaikan nuansa cuaca, sehingga memberikan kesan kontekstual dan relevan. Tepat di bawahnya, terdapat carousel gambar yang akan berfungsi menampilkan berita ataupun informasi penting.

Desain kemudian dilanjutkan dengan tab Layanan dan Informasi, yang memudahkan pengguna untuk berpindah konteks antara dua jenis konten utama tanpa harus berpindah halaman. Ikon-ikon layanan seperti misalnya seperti “Cari Bidang,” “Cari Berkas,” hingga “Buku Tamu” disusun dalam bentuk grid horizontal agar mudah diakses dengan ibu jari pada perangkat mobile. Bagian bawah menampilkan konten dinamis berupa Siaran dan Pengumuman dalam bentuk kartu carousel horizontal, yang memungkinkan pengguna untuk menggulir dan melihat lebih banyak informasi dengan cepat tanpa perlu berpindah halaman. Terakhir, ada bagian Ikuti YouTube Kami yang menampilkan video unggulan dari kanal YouTube pemerintah atau mitra terkait. Bagian ini tidak hanya berfungsi sebagai kanal informasi tambahan, tetapi juga mendukung pendekatan multikanal untuk menjangkau masyarakat.

Berikut adalah desain-desain dalam Figma untuk bagian autentikasi dalam aplikasi mobile Balikpapan, yang terdiri dari tiga halaman, yaitu Login, Daftar dan Lupa Password. Halaman pertama adalah Login, yang menjadi pintu utama untuk mengakses aplikasi. Elemen penting seperti input *email/username* dan *password* ditampilkan secara jelas dengan indikator visual modern seperti ikon dan tombol *eye toggle* untuk melihat *password*, dan penggunaan font modern seperti “Poppins”. Terdapat juga opsi “Login dengan Alternatif” yang akan diganti dengan cara login alternatif seperti “Login dengan SSO” pada saat pengembangan selanjutnya. Lalu ada tautan “Lupa Password?” untuk mengarahkan pengguna yang mengalami kendala masuk.

Halaman kedua adalah Daftar. Formulir pendaftaran dibuat ringkas dengan hanya mencantumkan *field* penting seperti nama lengkap, *email*, *password*, dan konfirmasi *password*, untuk mengurangi beban kognitif saat mengisi. Tombol aksi utama “Daftar” diberi warna yang kontras agar mudah dikenali, sementara di bagian bawah juga tersedia tombol menuju *FAQ* atau pusat bantuan untuk mendukung pengguna yang membutuhkan informasi tambahan. Elemen “Sudah punya akun? Login” disematkan dengan gaya teks interaktif agar proses berpindah antarhalaman



Gambar 3.24. Desain Figma Aplikasi Mobile Balikpapan - Bagian 2

menjadi *seamless* dan familiar.

Halaman ketiga adalah *Lupa Password*, yang menyajikan antarmuka minimalis dengan satu kolom input *email* dan tombol "*Reset Password*". Fokus desain diarahkan pada satu tujuan utama: memudahkan pengguna dalam proses pemulihan akun. Tampilan sederhana ini tidak hanya mempercepat interaksi, tetapi juga menurunkan tingkat stres pengguna saat mengalami masalah *login*.

Ketiga halaman ini dirancang dengan prinsip aksesibilitas dan kejelasan visual. Konsistensi warna, tipografi tebal untuk judul, dan *whitespace* yang cukup membuat pengalaman pengguna terasa ringan dan profesional, sangat penting untuk aplikasi layanan publik seperti ini. Desainnya mengedepankan kemudahan akses bagi pengguna dari berbagai latar belakang usia dan tingkat literasi digital.

### 3.6.3 Implementasi Aplikasi Kota

Implementasi aplikasi ini dilakukan menggunakan Flutter, sebuah framework open-source yang memungkinkan pembuatan aplikasi mobile dengan satu basis kode untuk Android dan iOS. Proses pengembangan meliputi beberapa tahap utama:

## 1) Halaman Depan

Halaman depan aplikasi mobile Balikpapan dirancang sebagai pintu gerbang digital untuk mengakses layanan pemerintah kota. Dibangun menggunakan Flutter dengan `StatefulWidget`, halaman ini menggunakan `ListView` untuk menampung berbagai section informasi dan layanan dalam layout yang dapat di-scroll secara vertikal.

Bagian atas halaman terdiri dari `AppBar` dengan logo dan branding aplikasi, serta `BottomNavigationBar` dengan dua tab utama "Beranda" dan "Login". Section carousel menampilkan informasi cuaca real-time dengan greeting sesuai waktu dan slideshow banner menggunakan `CarouselSlider` dengan autoplay feature. Background menggunakan image asset dengan `ColorFiltered` overlay untuk efek visual yang menarik.

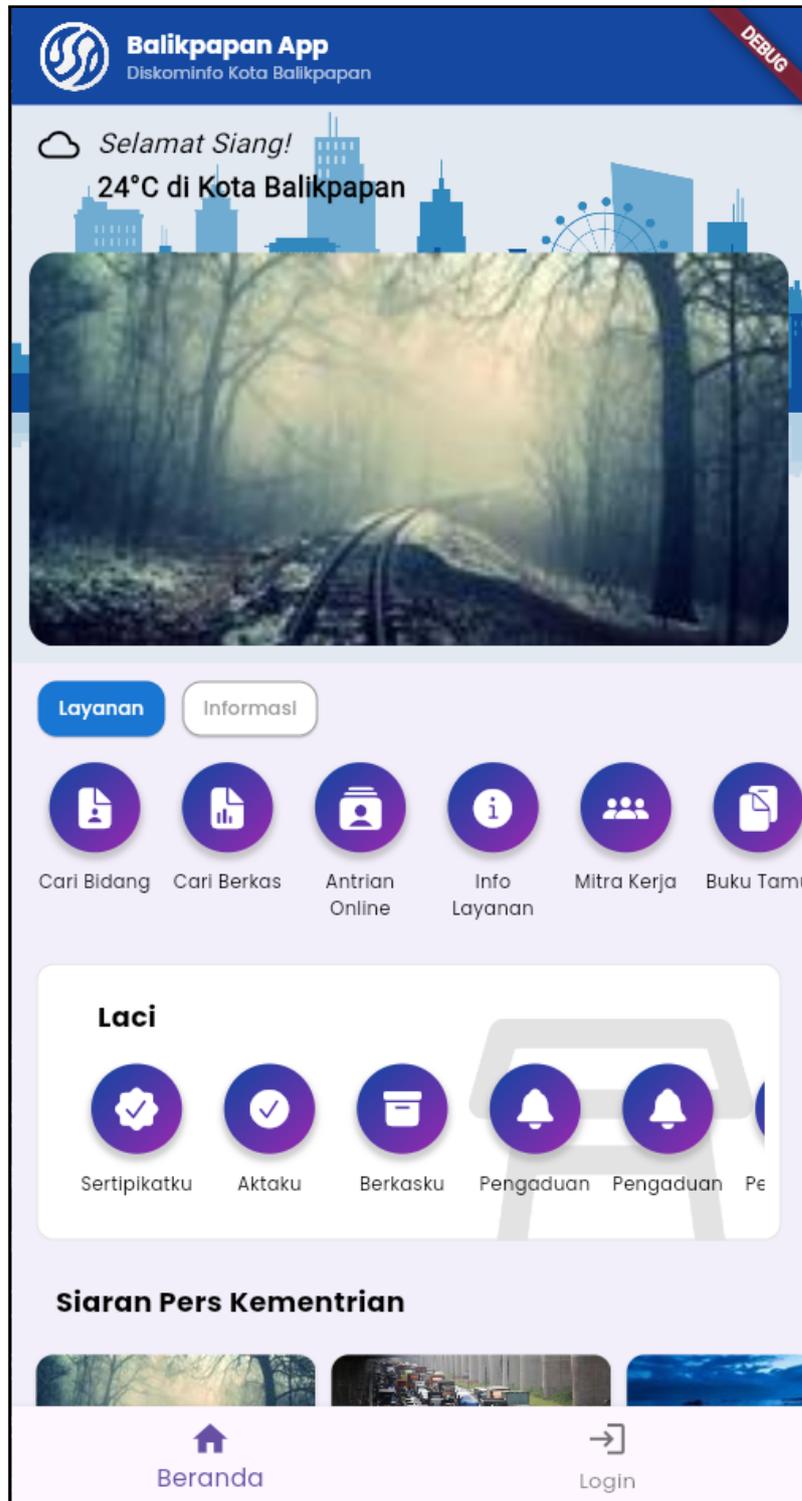
Sistem layanan mengimplementasikan komponen toggle yang memungkinkan pengguna beralih antara mode "Layanan" dan "Informasi". Mode Layanan menampilkan button dengan icon untuk layanan-layanan pemerintah kota, sedangkan mode Informasi berisi akses ke kontak kantor, bantuan, website resmi, dan informasi publik. Buttons container dibungkus dalam `GradientContainer` dengan efek gradient background dan transisi yang smooth menggunakan conditional rendering.

Section personal menampilkan layanan personal masyarakat dalam bentuk card dengan elevation dan border styling. Untuk konten dinamis, halaman mengimplementasikan horizontal scrollable cards untuk berita, pengumuman, dan multimedia menggunakan `ClipRRect`, `Image.network`, dan overflow handling. Section multimedia menggunakan thumbnail dengan overlay `Stack` dan integrasi `url_launcher` untuk redirect eksternal.

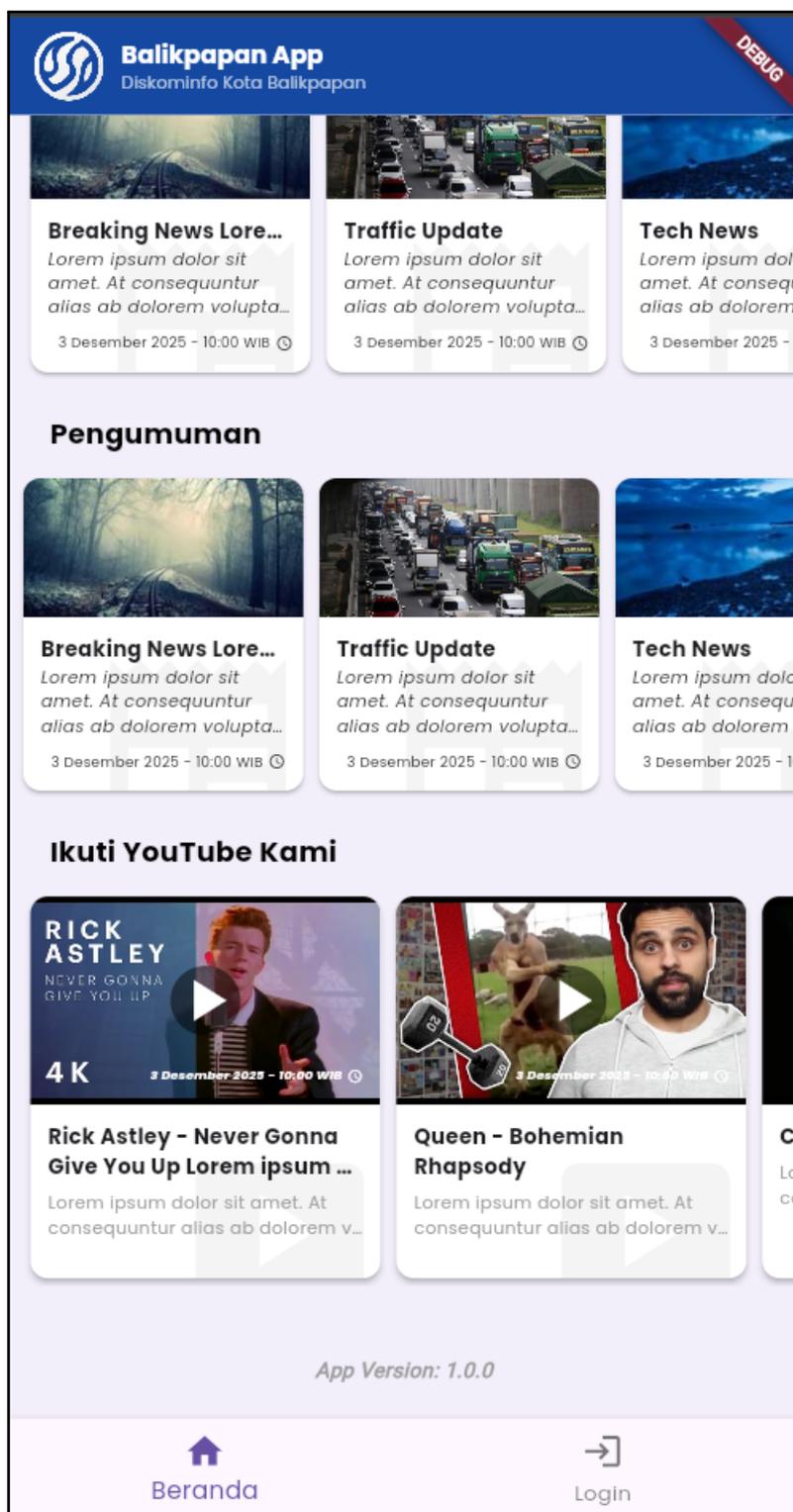
Implementasi teknis menggunakan `SingleChildScrollView` untuk scroll behavior yang smooth, state management dengan `setState`, dan responsive design dengan `MediaQuery`. Layout menggunakan percentage-based sizing untuk adaptasi berbagai ukuran layar dan `maxLines` properties untuk handling konten yang bervariasi, memastikan konsistensi UI di semua perangkat.

## 2) Halaman Login

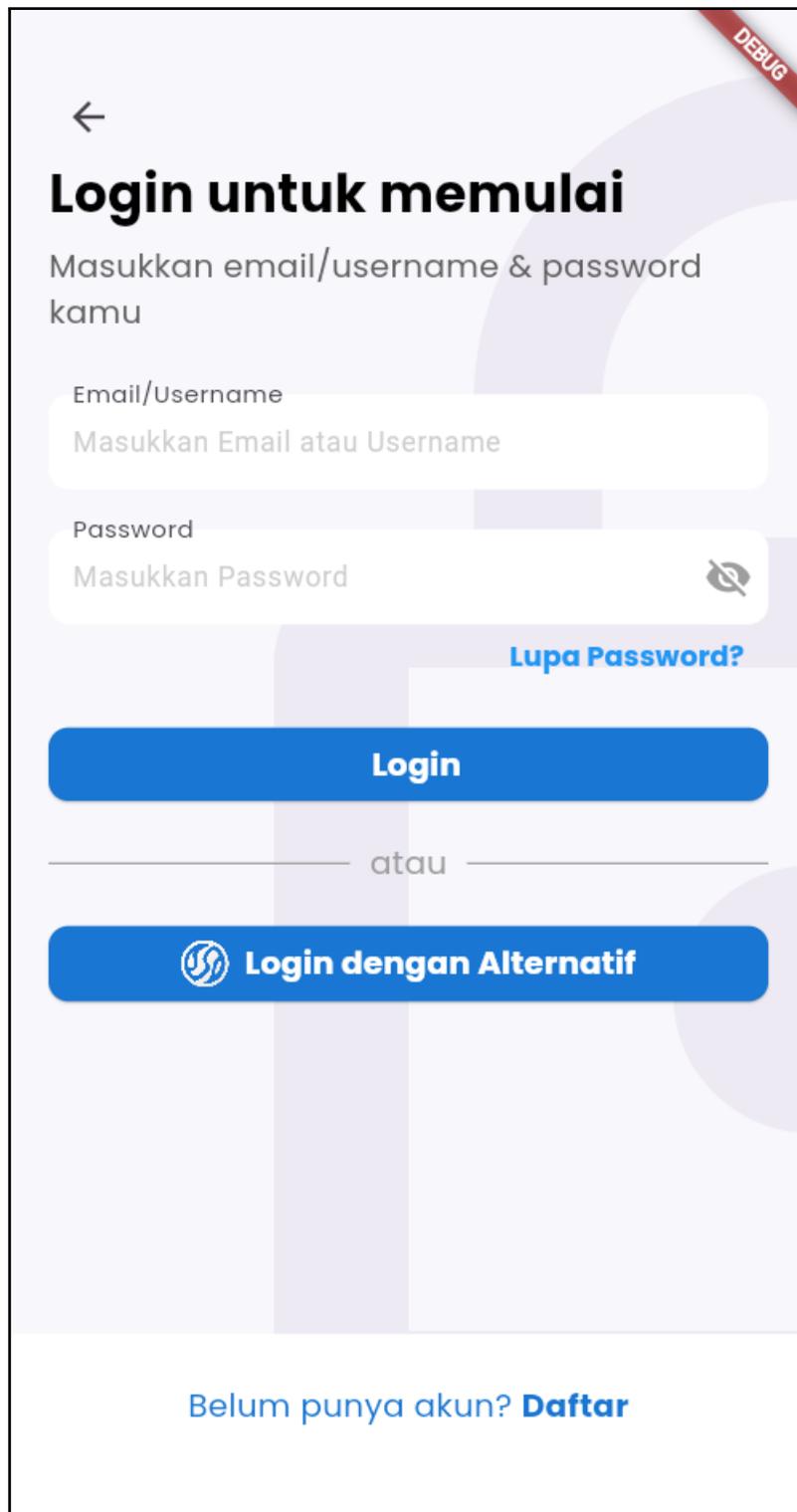
Halaman login aplikasi mobile Balikpapan merupakan komponen autentikasi utama yang diimplementasikan menggunakan `StatelessWidget`



Gambar 3.25. Halaman Depan Aplikasi Mobile Balikpapan - Bagian 1



Gambar 3.26. Halaman Depan Aplikasi Mobile Balikpapan - Bagian 2



Gambar 3.27. Halaman Login Aplikasi Mobile Balikpapan

dengan pendekatan *user-centered design*. Struktur halaman menggunakan arsitektur Stack dengan tiga layer: background color, vector background, dan main content layer untuk menciptakan visual depth yang optimal.

Halaman menggunakan `SingleChildScrollView` untuk menangani overflow content pada layar kecil, dengan header berisi `BackButton` dan `PageHeading` yang menampilkan teks "Login untuk memulai". Form section terdiri dari komponen `TextInput` untuk email/username dan `PasswordInput` dengan visibility toggle untuk keamanan input password.

Implementasi navigasi menggunakan GetX framework dengan `Get.to()` untuk transisi yang smooth ke halaman lain. Halaman menyediakan tiga aksi utama: `ForgotPasswordButton` untuk redirect ke halaman lupa password, `CustomElevatedButton` untuk proses login utama, dan tombol alternatif untuk metode autentikasi lainnya. Layout menggunakan `MediaQuery` untuk responsive sizing, memastikan tampilan optimal di berbagai ukuran layar mobile.

### 3) Halaman Daftar

Halaman daftar aplikasi mobile Balikpapan merupakan komponen penting untuk pendaftaran pengguna baru dalam sistem layanan digital pemerintah kota. Halaman ini dibangun menggunakan `StatelessWidget` dengan desain yang konsisten dengan halaman login untuk memberikan pengalaman pengguna yang familiar.

Struktur visual menggunakan arsitektur Stack dengan tiga lapisan utama: background berwarna, vector background dengan ikon kunci di bagian kanan bawah, dan konten utama. Header berisi tombol kembali dan judul "Daftar untuk memulai" beserta subjudul penjelasan.

Bagian form pendaftaran terdiri dari empat komponen input utama yang diimplementasikan dengan widget khusus Flutter untuk memastikan konsistensi dan validasi yang tepat:

- `TextInput` untuk Nama Lengkap: Field pertama menggunakan `TextFormField` dengan validator untuk memastikan nama tidak kosong dan memiliki panjang minimal tertentu. Input ini menggunakan `TextInputType.name` untuk mengoptimalkan keyboard sesuai konteks.
- `TextInput` untuk Email: Menggunakan `TextInputType.emailAddress` dengan validator regex untuk memverifikasi format email yang valid. Field

The image shows a mobile application registration screen. At the top right, there is a red diagonal banner with the word "DEBUG" in white. Below this is a back arrow icon. The main heading is "Daftar untuk memulai" in bold black text, followed by the subtitle "Masukkan detail kamu untuk mendaftar". There are four input fields: "Nama Lengkap" with placeholder "Masukkan Nama Lengkap", "Email" with placeholder "Masukkan Email", "Password" with placeholder "Masukkan Password" and a toggle icon, and "Konfirmasi Password" with placeholder "Masukkan Password lagi" and a toggle icon. Below the fields is a blue "Daftar" button, followed by a white button with blue text "FAQ / Pusat Bantuan". At the bottom, there is a white bar with the text "Sudah punya akun? Login" in blue.

Gambar 3.28. Halaman Daftar Aplikasi Mobile Balikpapan

ini dilengkapi dengan ikon email dan helper text untuk panduan pengguna.

- PasswordInput untuk Password: Komponen password khusus dengan fitur visibility toggle menggunakan IconButton dan obscureText property. Validator memastikan password memenuhi kriteria keamanan minimal seperti panjang karakter dan kombinasi huruf-angka.
- PasswordInput untuk Konfirmasi Password: Field terakhir menggunakan validator khusus yang membandingkan input dengan password sebelumnya untuk memastikan konsistensi. Menggunakan TextEditingController untuk mendapatkan nilai password pertama dan melakukan perbandingan real-time.

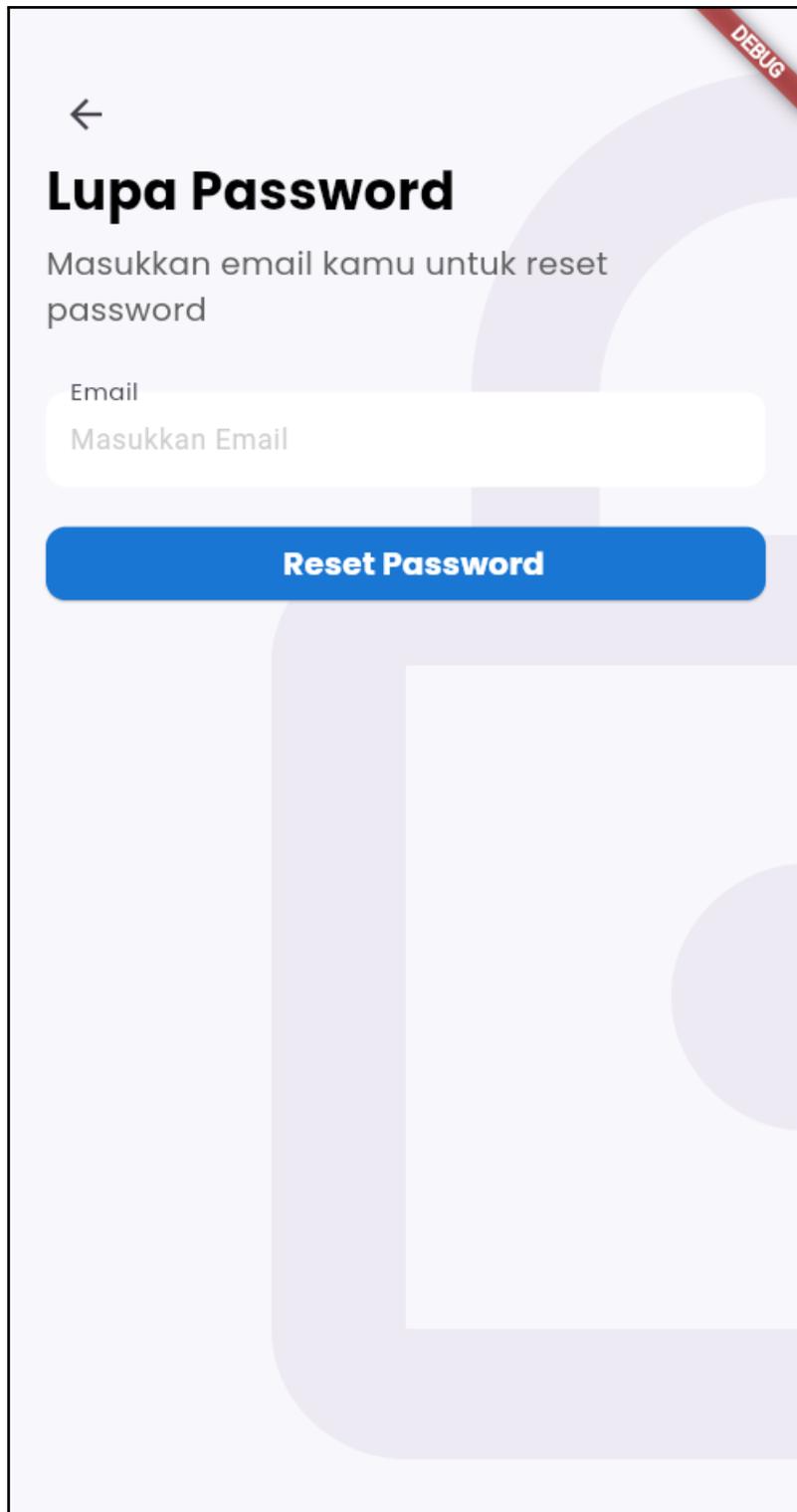
Setiap field input dibungkus dalam Container dengan margin dan padding yang konsisten, menggunakan InputDecoration dengan border styling yang seragam. Implementasi menggunakan SingleChildScrollView untuk menangani overflow konten di berbagai ukuran layar. Bagian bawah halaman menampilkan prompt "Sudah punya akun?" dengan navigasi ke halaman login menggunakan GetX framework. Desain responsif memastikan tampilan optimal di semua perangkat mobile dengan layout yang mudah digunakan dan accessible untuk semua pengguna.

#### **4) Halaman Lupa Password**

Halaman lupa password pada aplikasi mobile Balikpapan berfungsi sebagai fitur pemulihan akses bagi pengguna yang lupa kata sandi. Desain halaman ini sederhana dan konsisten dengan halaman login dan daftar, menggunakan satu kolom input email dan satu tombol aksi utama "Reset Password". Pengguna cukup memasukkan email yang terdaftar, lalu sistem akan mengirimkan instruksi reset password melalui email atau SMS. Tampilan halaman dibuat minimalis agar pengguna dapat fokus pada proses pemulihan tanpa gangguan visual, dengan navigasi jelas untuk kembali ke halaman login jika diperlukan.

#### **5) Kustomisasi Tampilan Mudah**

Karena pengembangan ini awal dari sebuah pengembangan lebih besar, maka dilakukan kustomisasi tampilan aplikasi mobile Balikpapan dengan konfigurasi terpusat. Konfigurasi ini memungkinkan perubahan tema dan identitas



Gambar 3.29. Halaman Lupa Password Aplikasi Mobile Balikpapan

aplikasi secara mudah melalui satu file konfigurasi, yaitu `settings.dart`. Dengan pendekatan ini, pengembang dapat mengubah warna dan elemen visual lainnya tanpa perlu mengedit banyak file, sehingga mempercepat proses pengembangan dan memudahkan maintenance. Berikut adalah filenya:

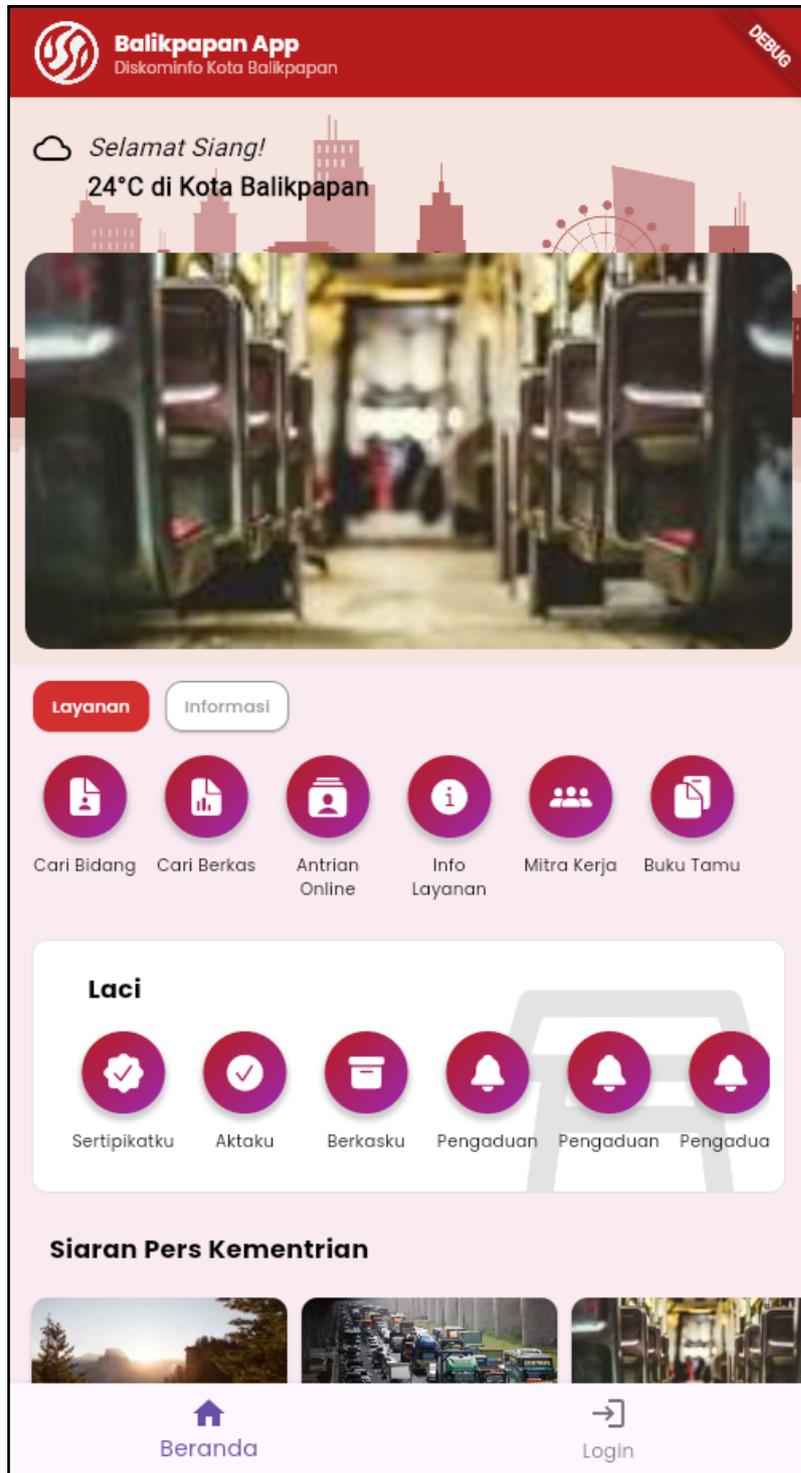
```
1 // Konfigurasi Identitas Aplikasi
2 final String appTitle = 'Balikpapan App';
3 final String appSubtitle = 'Diskominfo Kota Balikpapan';
4 final String appVersion = '1.0.0';
5
6 // Konfigurasi Tema Warna
7 final Color? primaryColor = Colors.red[900];
8 final Color? secondaryColor = Colors.red[700];
9 final Color? buttonHighlightColor = Colors.purple;
10 final Color? primaryTextColor = Colors.black;
11 final Color? secondaryTextColor = Colors.grey[700];
12 final Color? backgroundColor = Color(0xFFFF8F7FC);
```

Kode 3.1: Konfigurasi Aplikasi Mobile Balikpapan

Sistem konfigurasi ini memberikan fleksibilitas dalam mengubah tampilan aplikasi tanpa perlu memodifikasi kode di berbagai file. Dengan pendekatan ini, perubahan warna atau identitas aplikasi dapat dilakukan dengan mudah melalui satu file konfigurasi pusat. Berikut komponen konfigurasinya:

- `primaryColor`: Warna utama untuk AppBar dan elemen UI primer
- `secondaryColor`: Warna untuk tombol dan elemen UI sekunder
- `buttonHighlightColor`: Warna khusus untuk highlight tombol
- `primaryTextColor`: Warna teks utama
- `secondaryTextColor`: Warna teks sekunder
- `backgroundColor`: Warna latar belakang aplikasi

Implementasi ini memungkinkan pengembang atau administrator untuk dengan mudah mengubah skema warna aplikasi untuk berbagai keperluan, seperti penyesuaian dengan identitas visual instansi atau pembuatan variasi tema untuk kebutuhan khusus. Berikut adalah contoh kustomisasinya, dimana warna utama diubah menjadi merah:



Gambar 3.30. Contoh Kustomisasi Warna Aplikasi Mobile Balikpapan

### 3.6.4 Pengujian Aplikasi Kota

Pengujian aplikasi mobile Kota Balikpapan difokuskan pada aspek *frontend* dan *UI* yang telah dikembangkan, mengingat implementasi yang diselesaikan hanya mencakup antarmuka pengguna tanpa integrasi *backend*. Pengujian dilakukan pada berbagai perangkat *Android* dengan ukuran layar yang berbeda untuk memastikan responsivitas desain, *testing navigasi* antar halaman *aplikasi*, validasi *form input* pada halaman *login* dan *registrasi*, serta pengujian komponen interaktif seperti *carousel hero*, *news cards*, dan *button rows*. Pengujian juga mencakup verifikasi konsistensi *visual identity* dengan *branding* Pemerintah Kota Balikpapan dan *testing* kemudahan kustomisasi warna melalui *file konfigurasi settings.dart*.

### 3.6.5 Deployment & Maintenance Aplikasi Kota

Deployment aplikasi mobile Kota Balikpapan dilakukan dalam dua tahap, dimana pada tahap pertama yang telah diselesaikan hanya mencakup pengembangan UI dan frontend menggunakan Flutter framework. Aplikasi berhasil di-build menjadi file APK untuk platform Android dan dapat dijalankan pada berbagai perangkat dengan performa yang stabil. Tahap deployment awal ini meliputi kompilasi kode Flutter, pengujian pada multiple device, dan pembuatan installer APK yang siap untuk distribusi internal. Tim IT DATSI Diskominfo Balikpapan akan melanjutkan pengembangan pada tahap kedua dengan menambahkan backend, implementasi API, dan fitur-fitur lanjutan yang memerlukan koneksi dengan server pemerintah kota.

Maintenance aplikasi mobile akan sepenuhnya ditangani oleh tim IT DATSI setelah proses pengembangan backend selesai. Tim IT bertanggung jawab untuk mengintegrasikan frontend yang telah dikembangkan dengan sistem backend pemerintah kota, menghubungkan aplikasi dengan database dan API layanan publik, serta mengimplementasikan fitur-fitur tambahan sesuai kebutuhan operasional. Proses maintenance jangka panjang meliputi update konten aplikasi, perbaikan bug yang ditemukan, pengembangan fitur baru berdasarkan feedback pengguna, dan optimisasi performa aplikasi. Tim juga akan menangani proses submission ke Google Play Store dan App Store untuk distribusi publik, serta melakukan monitoring analytics dan user engagement untuk meningkatkan kualitas layanan digital pemerintah kota.

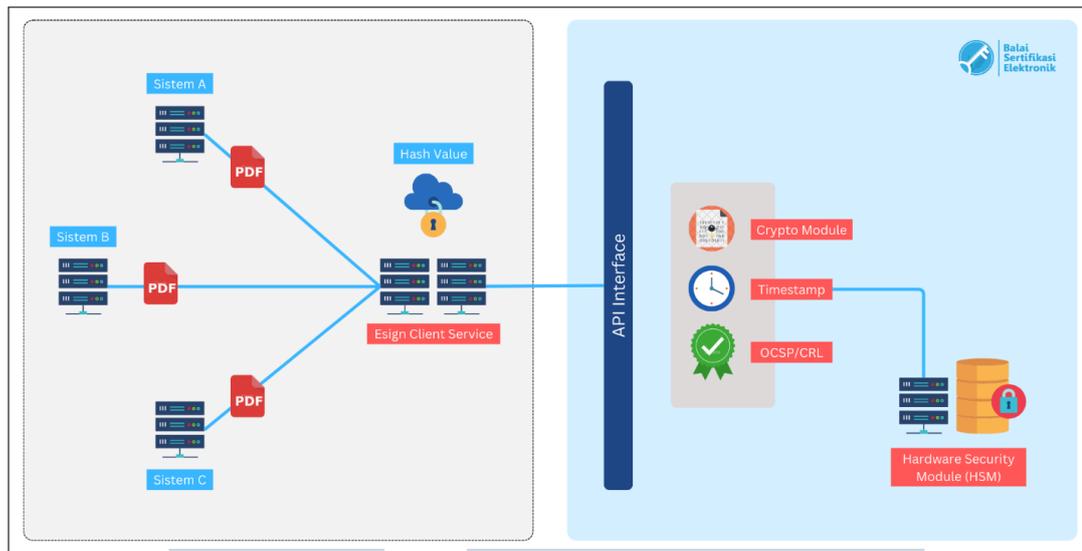
### 3.7 Website E-Signature Kota

Website E-Signature Kota adalah sebuah platform digital yang terintegrasi dengan API E-Sign Client Service dari Balai Besar Sertifikasi Elektronik (BSrE). Platform ini memungkinkan pengguna yang telah terdaftar dengan Nomor Induk Kependudukan (NIK) dan passphrase untuk membubuhkan tanda tangan elektronik dalam berbagai format, seperti tanda tangan digital, spesimen gambar, atau kode QR yang dihasilkan secara otomatis. E-Sign Client Service berfungsi sebagai penghubung antara sistem informasi milik pengguna dengan sistem penandatanganan elektronik berbasis cloud yang dikelola oleh BSrE, memfasilitasi pengembangan transaksi elektronik yang memerlukan validasi tanda tangan secara hukum. Selain itu, website ini dilengkapi dengan fitur verifikasi untuk memeriksa keaslian sertifikat digital pada dokumen PDF yang telah ditandatangani.

Website ini juga menyediakan antarmuka berbasis login dengan fitur-fitur tambahan seperti permintaan tanda tangan (*request signature*) dan penandatanganan massal (*bulk signing*). Fitur *bulk signing* sangat berguna untuk menandatangani sekumpulan dokumen PDF serupa secara efisien, seperti ijazah, sertifikat, atau dokumen administrasi lainnya, sehingga menghemat waktu dan meningkatkan produktivitas dalam proses administrasi digital. Dengan adanya website ini, instansi pemerintah atau pihak terkait dapat melakukan validasi dokumen secara lebih cepat, aman, dan sesuai dengan standar hukum yang berlaku.

#### 3.7.1 API E-Sign Client Service

Menurut dokumentasi *API E-Sign* oleh BSrE [24], *API E-Sign Client Service* adalah solusi penghubung antara sistem informasi milik pengguna (seperti *website* atau aplikasi) dengan sistem penandatanganan elektronik berbasis cloud milik Balai Sertifikasi Elektronik (BSrE). Layanan ini memungkinkan instansi pemerintah atau organisasi terdaftar untuk menerapkan tanda tangan digital yang sah secara hukum pada dokumen elektronik (PDF) dengan integrasi yang terstandarisasi.



Gambar 3.31. Skema API *E-Sign Client Service*

Sumber: [24]

Proses kerjanya API dimulai ketika sistem pengguna mengirim dokumen PDF yang akan ditandatangani ke server E-Sign Client yang terpasang secara lokal di instansi terkait seperti Diskominfo Balikpapan. Server E-Sign Client kemudian menghitung nilai hash dari dokumen tersebut dan mengirimkannya ke server BSRé melalui koneksi yang aman. Di sisi BSRé, nilai hash tersebut diolah menjadi tanda tangan digital yang sah secara hukum, kemudian dikirimkan kembali ke server E-Sign Client. Terakhir, tanda tangan digital tersebut disematkan ke dalam dokumen asli dan hasil akhir dokumen yang telah ditandatangani dikembalikan ke sistem pengguna.

Untuk menjalankan API E-Sign Client Service, diperlukan server dengan spesifikasi teknis tertentu yang memenuhi standar keamanan dan performa BSRé. Berikut rincian persyaratannya:

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

Tabel 3.6. Spesifikasi Server untuk API E-Sign Client Service

<b>Spesifikasi Minimum Server</b>	
<b>Komponen</b>	<b>Spesifikasi</b>
Sistem Operasi	CentOS 7 2009
Memori (RAM)	8 GB
Processor	4 Core
Penyimpanan (HDD)	50 GB
Jaringan	Koneksi Internet; Port 80, 443, 22 terbuka
<b>Whitelist Alamat yang Diperlukan</b>	
Production	<a href="https://api-bsre.bssn.go.id">https://api-bsre.bssn.go.id</a>
Development/Production	<a href="https://registry-bsre.bssn.go.id">https://registry-bsre.bssn.go.id</a>
Email	<a href="mailto:mail.bssn.go.id">mail.bssn.go.id</a>

Server untuk proyek ini telah dipersiapkan sepenuhnya oleh tim infrastruktur Diskominfo Balikpapan sebelum pelaksanaan magang dimulai. Dengan ini, pengembang hanya perlu fokus pada integrasi API dengan sistem website tanpa perlu mengurus infrastruktur dasar.

#### **A Proses Autorisasi dan Endpoint API E-Sign Client Service**

Proses implementasi API E-Sign Client Service melibatkan beberapa langkah penting yang harus diikuti untuk memastikan integrasi yang sukses. Berikut adalah penjelasan mengenai proses otorisasi dan endpoint API yang digunakan dalam sistem ini:

##### **1) Proses Authorization**

API E-Sign Client Service menggunakan *Basic Authentication* untuk otorisasi akses. Berikut mekanismenya:

- Kredensial (username dan password) dibuat melalui halaman admin Esign Client Service yang terpasang di server lokal
- Kredensial dikirim dalam header request dengan format:

Authorization: Basic base64(username:password)

- Sistem akan memverifikasi kredensial sebelum mengizinkan akses ke endpoint API

## 2) Endpoint Sign PDF (Versi 2)

Endpoint ini digunakan untuk membubuhkan tanda tangan elektronik pada dokumen PDF.

Tabel 3.7. Parameter Endpoint Sign PDF

Parameter	Tipe	Deskripsi
nik	String	Nomor Induk Kependudukan (16 digit)
email	String	Alternatif pengganti NIK
passphrase/otp	String	Password atau OTP untuk otentikasi
file	Base64 String	Dokumen PDF yang akan ditandatangani
signatureProperties	Array	Konfigurasi tampilan tanda tangan

Berikut detail parameter signatureProperties:

- tampilan: "VISIBLE" atau "INVISIBLE"
- imageBase64: Gambar tanda tangan dalam base64 (jika visible)
- page: Nomor halaman untuk tanda tangan
- originX/originY: Koordinat tanda tangan
- width/height: Ukuran tanda tangan dalam pixel
- location: Lokasi penandatanganan (opsional)
- reason: Alasan penandatanganan (opsional)

### Contoh Request:

```
1 POST /api/v2/sign/pdf
2 {
3   "nik": "3216785087543986",
4   "passphrase": "rahasia123",
5   "signatureProperties": [{
6     "tampilan": "VISIBLE",
7     "imageBase64": "iVBORw0KGgoAAA...",
8     "page": 1,
9     "originX": 0.5,
10    "originY": 0.8,
11    "width": 200,
```

```

12     "height": 100
13   }],
14   "file": ["JVBERi0xLjQK..."]
15 }

```

Kode 3.2: Contoh Request API Sign Endpoint

Berikut response Sukses:

- Status code: 200
- Body: Dokumen PDF yang telah ditandatangani dalam format base64
- Contoh:

```

1 {
2   "status": "success",
3   "signed_file": "JVBERi0xLjYK...",
4   "signature_info": {
5     "signer": "John Doe",
6     "timestamp": "2023-08-15T10:30:00Z"
7   }
8 }

```

Kode 3.3: Contoh Respons API Sign Endpoint

Berikut adalah berbagai contoh penggunaan API Sign Service dengan berbagai jenis visualisasi tanda tangan:

#### a) Visualisasi Invisible

Tipe tanda tangan invisible tidak menampilkan visualisasi tanda tangan pada dokumen PDF, namun dokumen tetap memiliki tanda tangan digital yang dapat diverifikasi secara elektronik.

```

1 POST /api/v2/sign/pdf
2 {
3   "email": "{{email}}",
4   "passphrase": "{{passphrase}}",
5   "signatureProperties": [{
6     "tampilan": "INVISIBLE"

```

```

7     }],
8     "file": ["{{fileBase64}}"]
9 }

```

Kode 3.4: Contoh Request Tanda Tangan Invisible

### b) Visualisasi Visible dengan Koordinat

Tipe tanda tangan visible dengan koordinat memungkinkan penempatan tanda tangan pada posisi spesifik di halaman dokumen menggunakan koordinat X dan Y.

```

1 POST /api/v2/sign/pdf
2 {
3     "nik": "{{nik}}",
4     "passphrase": "{{passphrase}}",
5     "signatureProperties": [{
6         "tampilan": "VISIBLE",
7         "imageBase64": "{{imageBase64}}",
8         "page": 1,
9         "originX": 10,
10        "originY": 10,
11        "width": 200,
12        "height": 100
13    }],
14    "file": ["{{fileBase64}}"]
15 }

```

Kode 3.5: Contoh Request Tanda Tangan Visible dengan Koordinat

### c) Visualisasi Visible dengan Tag Koordinat

Tipe tanda tangan visible dengan tag koordinat menggunakan placeholder berupa karakter khusus (seperti "\$") yang sudah ditempatkan dalam dokumen PDF sebagai penanda lokasi tanda tangan.

```

1 POST /api/v2/sign/pdf
2 {

```

```

3     "nik": "{{nik}}",
4     "otp": "{{otp}}",
5     "signatureProperties": [{
6         "tampilan": "VISIBLE",
7         "tag_koordinat": "$",
8         "imageBase64": "{{imageBase64}}",
9         "width": 200,
10        "height": 100
11    }],
12    "file": ["{{fileBase64}}"]
13 }

```

Kode 3.6: Contoh Request Tanda Tangan Visible dengan Tag Koordinat

#### d) Visualisasi Multiple Signatures

API mendukung penandatanganan dengan berbagai jenis visualisasi dalam satu request, memungkinkan kombinasi tanda tangan visible dan invisible pada satu atau beberapa dokumen sekaligus.

```

1 POST /api/v2/sign/pdf
2 {
3     "email": "{{email}}",
4     "otp": "{{otp}}",
5     "signatureProperties": [{
6         "tampilan": "VISIBLE",
7         "imageBase64": "{{imageBase64}}",
8         "page": 1,
9         "originX": 10,
10        "originY": 10,
11        "width": 200,
12        "height": 100,
13        "location": "Jakarta",
14        "reason": "Tanda tangan elektronik"
15    }, {
16        "tampilan": "INVISIBLE",
17        "location": null,

```

```

18     "reason": "Tanda tangan elektronik"
19   }, {
20     "tampilan": "VISIBLE",
21     "tag_koordinat": "$",
22     "imageBase64": "{{imageBase64}}",
23     "width": 120,
24     "height": 40,
25     "location": "Jakarta",
26     "reason": null
27   }],
28   "file": [ "{{fileBase64_1}}", "{{fileBase64_2}}", "
29   {{fileBase64_3}}" ]

```

Kode 3.7: Contoh Request Multiple Signatures

### 3) Endpoint Verify PDF (Versi 2)

Endpoint ini digunakan untuk memverifikasi keaslian dokumen yang telah ditandatangani.

Tabel 3.8. Parameter Endpoint Verify PDF

Parameter	Tipe	Deskripsi
file	Base64 String	Dokumen PDF yang akan diverifikasi
password	String (opsional)	Jika dokumen diproteksi password

#### Contoh Request:

```

1 POST /api/v2/verify/pdf
2 {
3   "file": "JVBERi0xLjYK...",
4   "password": "doc123" // opsional
5 }

```

Kode 3.8: Contoh Request API Verify Endpoint

#### Respons Sukses:

```

1 {
2   "conclusion": "VALID",

```

```

3   "description": "Dokumen valid dan utuh",
4   "signature_info": [{
5       "signer": "John Doe",
6       "signature_date": "2023-08-15T10:30:00Z",
7       "location": "Jakarta",
8       "certificate_valid": true,
9       "document_integrity": true
10  }],
11  "timestamp_info": {
12      "authority": "BSrE Timestamp Authority",
13      "timestamp_date": "2023-08-15T10:30:05Z"
14  }
15 }

```

Kode 3.9: Contoh Respons API Verify Endpoint

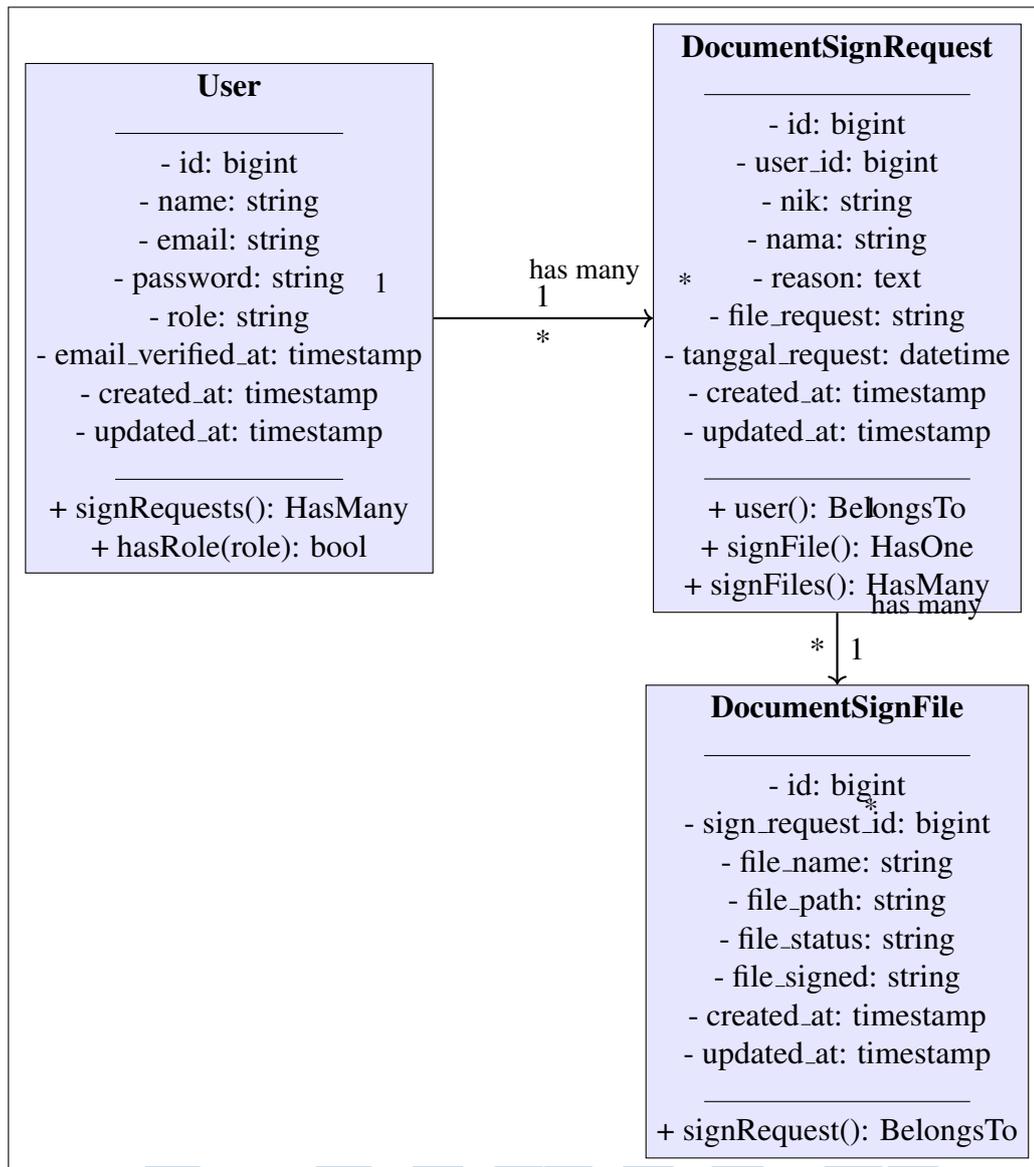
Berikut status verifikasi yang bisa di dapatkan dari response API:

- VALID: Tanda tangan dan dokumen sah
- INVALID: Tanda tangan tidak valid
- NO\_SIGNATURE: Dokumen tidak mengandung tanda tangan
- CERTIFICATE\_EXPIRED: Sertifikat tanda tangan kadaluarsa

### 3.7.2 Analisis Kebutuhan & Desain Sistem

Sistem Website E-Signature Kota memerlukan arsitektur yang terintegrasi dengan API E-Sign Client Service dari BSrE untuk mendukung penandatanganan elektronik yang sah secara hukum. Kebutuhan utama sistem meliputi autentikasi pengguna berbasis NIK dan passphrase, antarmuka web untuk upload dan signing dokumen PDF, serta fitur verifikasi dokumen yang telah ditandatangani. Website ini dirancang untuk digunakan oleh berbagai kalangan, termasuk sekolah-sekolah untuk penandatanganan ijazah dan sertifikat pendidikan, masyarakat umum untuk keperluan administrasi pribadi, serta instansi pemerintah dan swasta untuk kebutuhan dokumentasi resmi lainnya. Berikut penjelasannya:

#### A Class Diagram



Gambar 3.32. Class Diagram for E-Signature Application

Model *User* memiliki tiga metode utama: `documentSignRequests()` yang mengembalikan relasi *HasMany* untuk mengakses semua permintaan tanda tangan dokumen yang dibuat oleh *user*, `isAdmin()` yang mengembalikan nilai *boolean* untuk mengecek apakah *user* memiliki *role* admin, dan `isUser()` yang mengembalikan nilai *boolean* untuk memverifikasi *role user* biasa. Metode-metode ini mendukung sistem autentikasi dan otorisasi berbasis peran dalam aplikasi.

Model *DocumentSignRequest* mengimplementasikan tiga metode relasi *Eloquent*: `user()` yang mengembalikan relasi *BelongsTo* untuk mengakses data *user* pemilik *request*, `signFile()` yang mengembalikan relasi *HasOne*

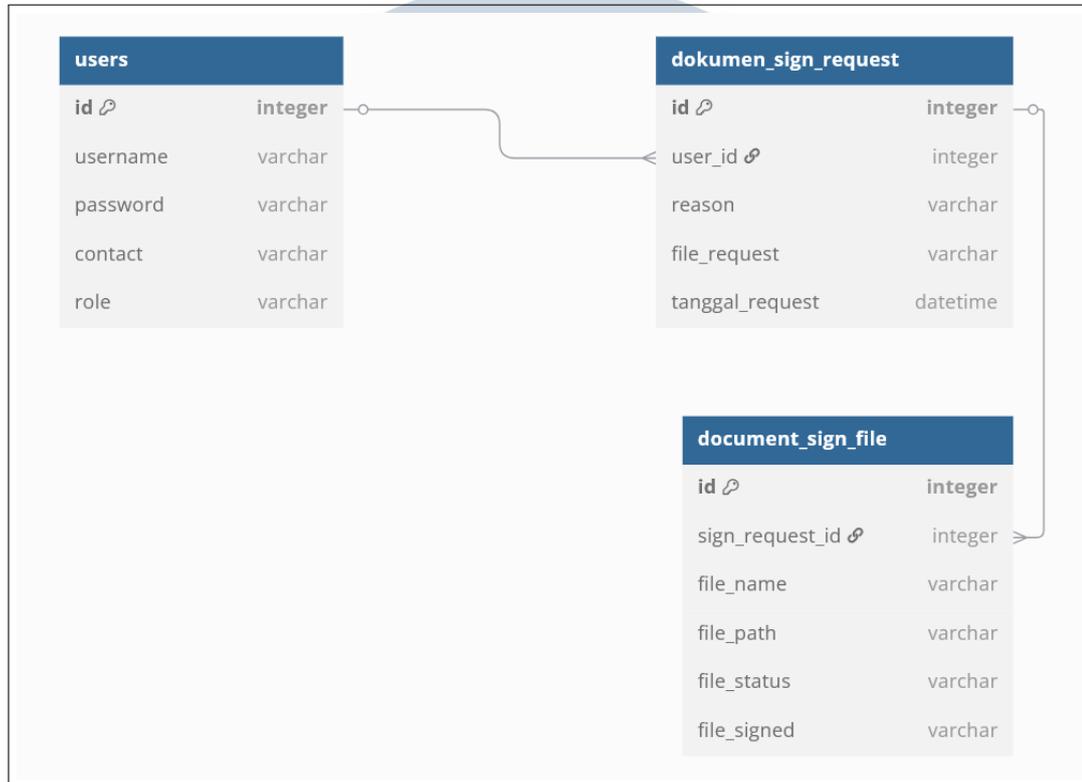
untuk mendapatkan satu *file* tanda tangan terkait, dan `signFiles()` yang mengembalikan relasi *HasMany* untuk mengakses semua *file* yang terkait dengan *request* tersebut. Metode-metode ini memfasilitasi navigasi data antar entitas dalam sistem manajemen dokumen.

Model *DocumentSignFile* memiliki satu metode relasi yaitu `signRequest()` yang mengembalikan relasi *BelongsTo* untuk mengakses data *request* induk dari *file* tersebut. Metode ini memungkinkan akses balik ke informasi *request* dan *user* yang memiliki *file*, mendukung integritas referensial dan kemudahan *query* data.

*Base Model* Laravel menyediakan metode-metode fundamental seperti `save()` untuk menyimpan perubahan data ke *database*, `delete()` untuk menghapus *record*, `find(id)` untuk mencari *record* berdasarkan *primary key*, `all()` untuk mengambil semua *record* dalam bentuk *Collection*, dan `create(array)` untuk membuat *record* baru. Metode-metode ini merupakan inti dari pola *Active Record* yang digunakan Laravel *Eloquent*. *HasFactory Trait* menyediakan metode `factory()` yang mengembalikan *instance Factory* untuk keperluan *seeding database* dan *testing*. *Trait* ini memungkinkan pembuatan data *dummy* secara otomatis dan konsisten untuk *development* dan *testing environment*, mendukung proses pengujian aplikasi yang lebih efektif.



## B Skema Database



Gambar 3.33. Desain Database E-Sign

Desain *database* sistem E-Signature dirancang untuk mendukung proses permintaan dan penandatanganan dokumen secara elektronik dengan arsitektur yang terstruktur dan efisien. *Database* ini terdiri dari tiga tabel utama yang saling berelasi untuk memastikan integritas data dan kemudahan pengelolaan sistem penandatanganan elektronik.

Tabel *users* berfungsi sebagai entitas pengguna sistem yang menyimpan informasi autentikasi dan otorisasi. Tabel ini memiliki lima *field* utama: *id* sebagai *primary key* yang memberikan *identifier* unik untuk setiap pengguna, *username* untuk nama pengguna yang digunakan dalam proses *login*, *password* yang menyimpan kata sandi terenkripsi untuk keamanan autentikasi, *contact* yang berisi informasi kontak seperti *email* atau nomor telepon pengguna, dan *role* yang menentukan peran pengguna dalam sistem (admin atau *user*) untuk mengatur hak akses yang berbeda sesuai tingkat kewenangan.

Tabel *document\_sign\_request* dirancang untuk mencatat semua permohonan tanda tangan dokumen yang diajukan melalui sistem. Tabel

ini memiliki enam *field* yang mencakup *user\_id* sebagai *foreign key* yang menghubungkan dengan tabel *users* untuk mengidentifikasi siapa yang mengajukan permohonan, *nik* yang menyimpan Nomor Induk Kependudukan orang yang akan melakukan penandatanganan, *nama* untuk menyimpan nama lengkap penandatanganan, *reason* yang berisi keterangan atau alasan pengajuan permohonan tanda tangan, *file\_request* yang menyimpan nama *file* dokumen yang diminta untuk ditandatangani dalam format *zip* atau *pdf*, dan *tanggal\_request* sebagai *timestamp* yang mencatat waktu pengajuan permohonan untuk keperluan *audit* dan *tracking*.

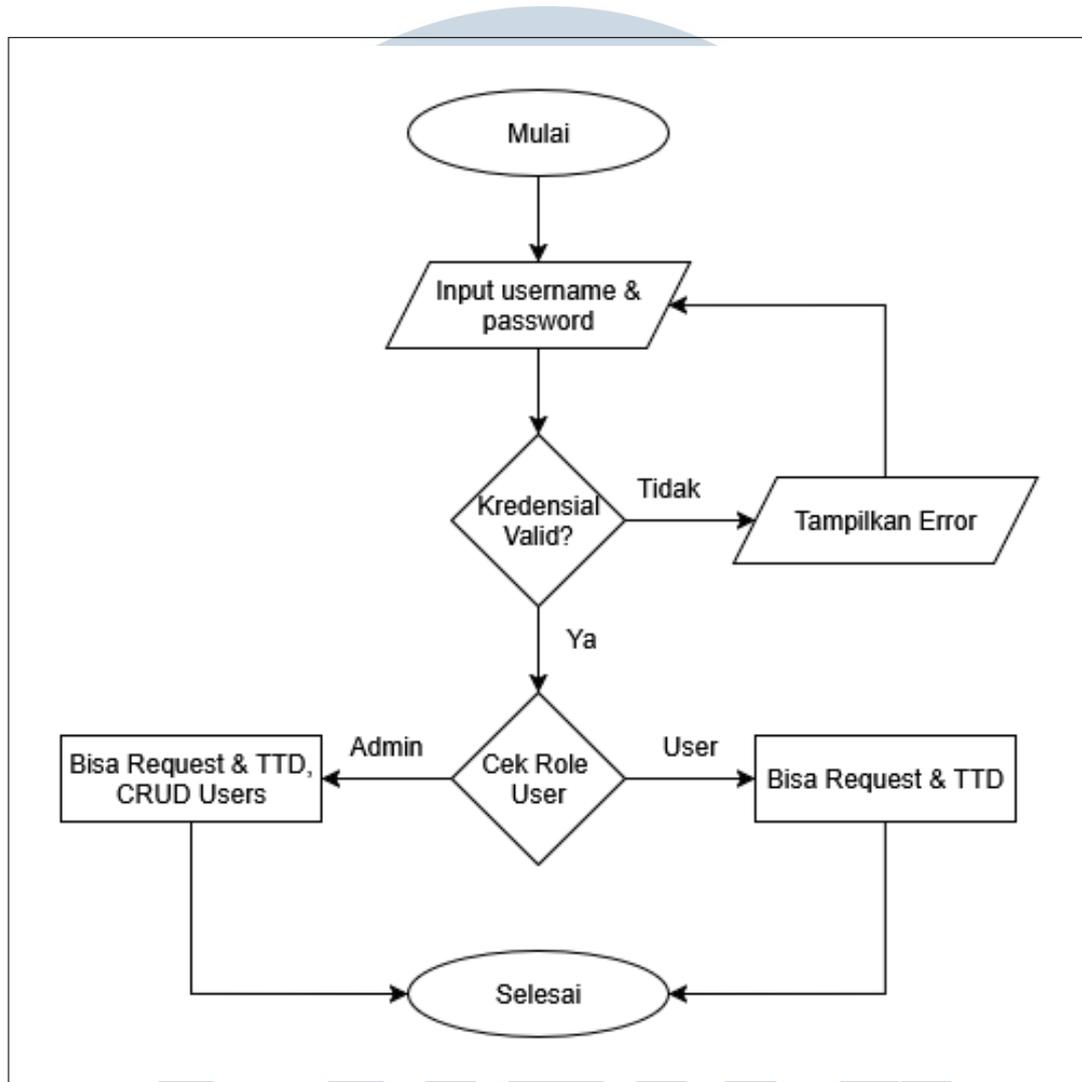
Tabel *document\_sign\_file* berfungsi untuk menyimpan detail setiap *file* individual yang akan atau sudah ditandatangani, terutama untuk menangani kasus dimana *file request* berupa *zip* yang berisi *multiple file* PDF. Tabel ini memiliki lima *field* yaitu *sign\_request\_id* sebagai *foreign key* yang menghubungkan dengan tabel *document\_sign\_request* untuk mengelompokkan *file-file* yang berasal dari satu permohonan yang sama, *file\_name* yang menyimpan nama *file* yang akan ditandatangani, *file\_path* yang berisi lokasi atau *path file* original sebelum proses penandatanganan, *file\_status* yang mencatat status proses penandatanganan dengan nilai seperti *pending* (menunggu), *processing* (sedang diproses), atau *completed* (selesai), dan *file\_signed* yang menyimpan *path* atau lokasi *file* hasil setelah berhasil ditandatangani secara elektronik.

Struktur *database* ini memungkinkan sistem untuk menangani proses *bulk signing* dimana satu permohonan dapat mencakup *multiple file* PDF yang akan ditandatangani secara bersamaan, sambil tetap mempertahankan *tracking* individual untuk setiap *file*. Relasi antar tabel dirancang untuk mendukung integritas referensial dan memudahkan *query* data untuk keperluan *reporting* dan *monitoring* status penandatanganan dokumen.

### C Flowchart

Flowchart sistem E-Signature Kota dirancang untuk menggambarkan alur kerja dari proses autentikasi pengguna, permintaan tanda tangan, hingga penandatanganan dokumen. Flowchart ini mencakup langkah-langkah utama seperti input kredensial, validasi, pengecekan role pengguna, upload dokumen, dan proses penandatanganan massal. Berikut adalah penjelasan dari flowchart tersebut:

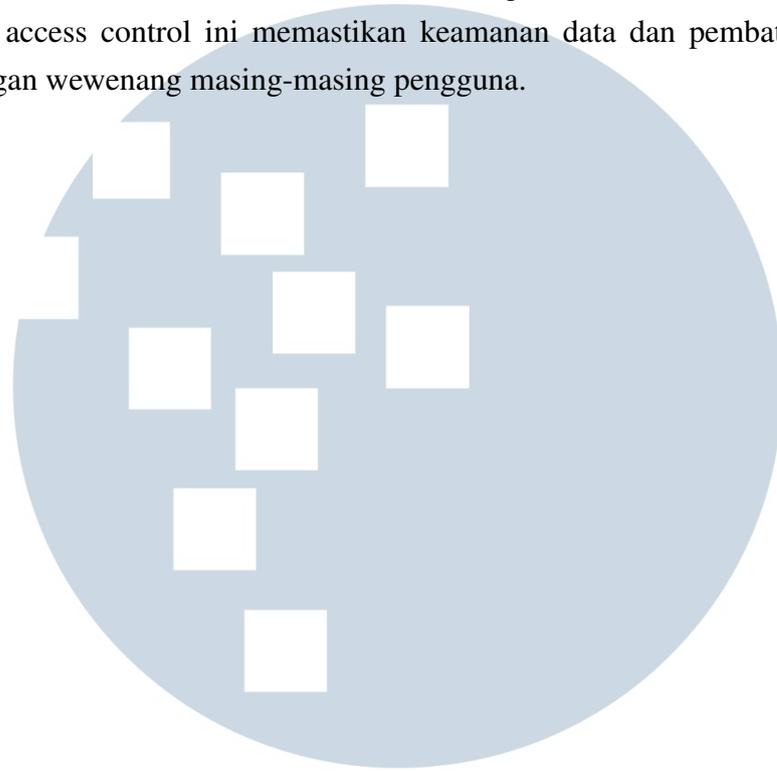
## 1) Sistem Autentikasi E-Sign BPN



Gambar 3.34. Flowchart Sistem Autentikasi E-Sign

Sistem autentikasi pada aplikasi e-sign dimulai dengan proses input kredensial berupa username dan password dari pengguna. Sistem kemudian melakukan validasi terhadap kredensial yang dimasukkan dengan mencocokkan data pada database. Jika kredensial tidak valid, sistem akan menampilkan pesan error dan mengarahkan pengguna kembali ke halaman login. Apabila validasi berhasil, sistem akan melakukan pengecekan role pengguna untuk menentukan dashboard yang sesuai. Pengguna dengan role "admin" akan diarahkan ke dashboard admin yang memiliki akses penuh untuk mengelola semua dokumen dan request tanda tangan, dan bisa manajemen Users, sedangkan pengguna dengan

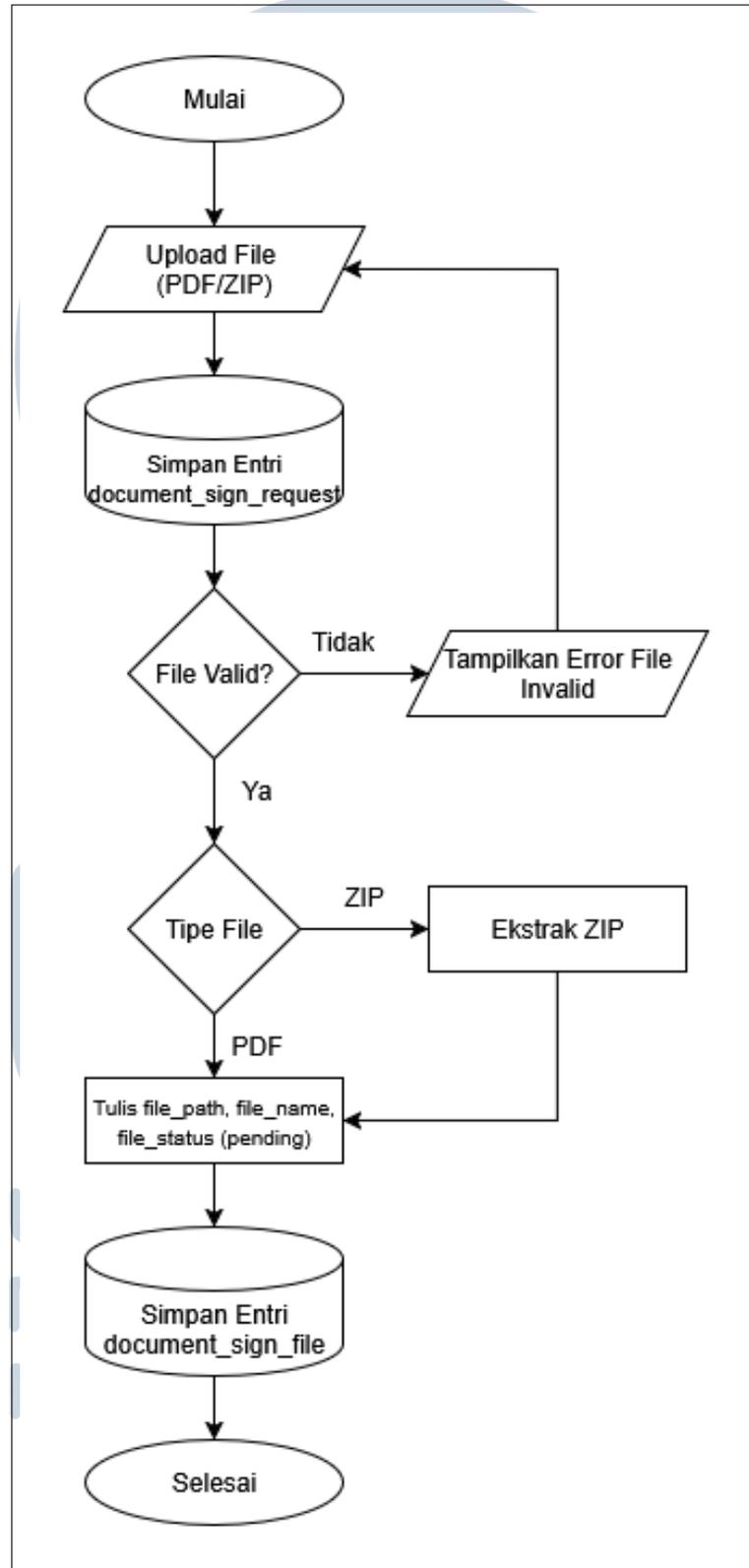
role "user" akan diarahkan ke dashboard user yang hanya dapat melihat, mengelola request mereka sendiri, dan melakukan tanda tangan untuk NIK mereka. Sistem role-based access control ini memastikan keamanan data dan pembatasan akses sesuai dengan wewenang masing-masing pengguna.



# UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

## 2) Proses Proses Upload dan Request TTD Dokumen

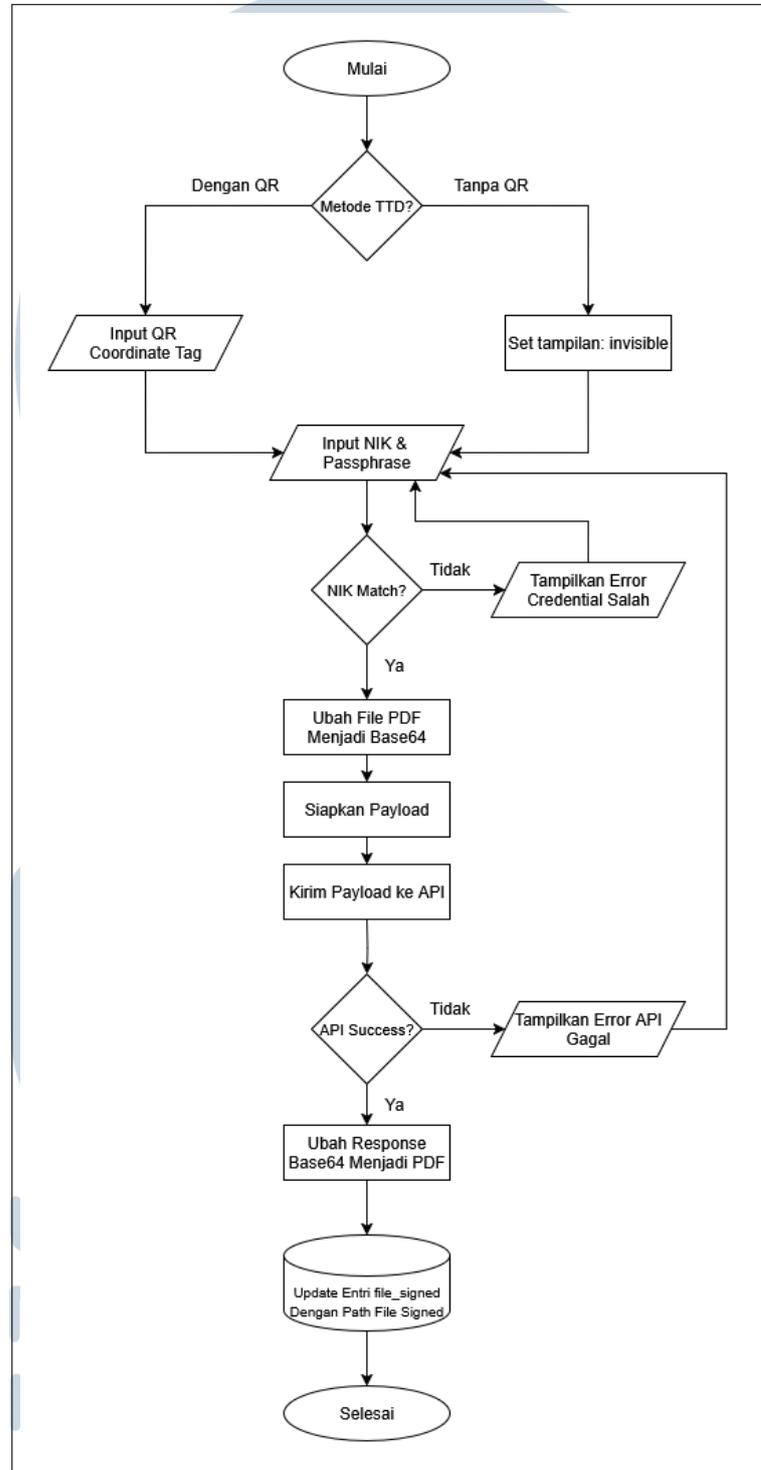


Gambar 3.35. Flowchart Proses Upload dan Request Dokumen

Proses pengajuan request dokumen dimulai ketika pengguna melakukan upload file melalui interface aplikasi. Sistem melakukan validasi terhadap file yang diupload dengan memeriksa format dan ukuran file yang diizinkan, yaitu PDF dan ZIP dengan maksimum ukuran 20MB. Jika file tidak memenuhi kriteria validasi, sistem akan menampilkan pesan error dan meminta pengguna untuk mengupload ulang file yang sesuai. Setelah validasi berhasil, sistem akan mengidentifikasi tipe file untuk menentukan metode pemrosesan yang tepat. Untuk file PDF tunggal, sistem akan langsung memproses dan menyimpan file tersebut. Sedangkan untuk file ZIP, sistem akan melakukan ekstraksi untuk mengambil semua file PDF yang terkandung di dalamnya. Setelah pemrosesan selesai, sistem akan menyimpan informasi file ke dalam database dengan status "pending" dan membuat record baru pada tabel `document_sign_request` dan `document_sign_file`.



### 3) Proses Tanda Tangan Digital dengan API

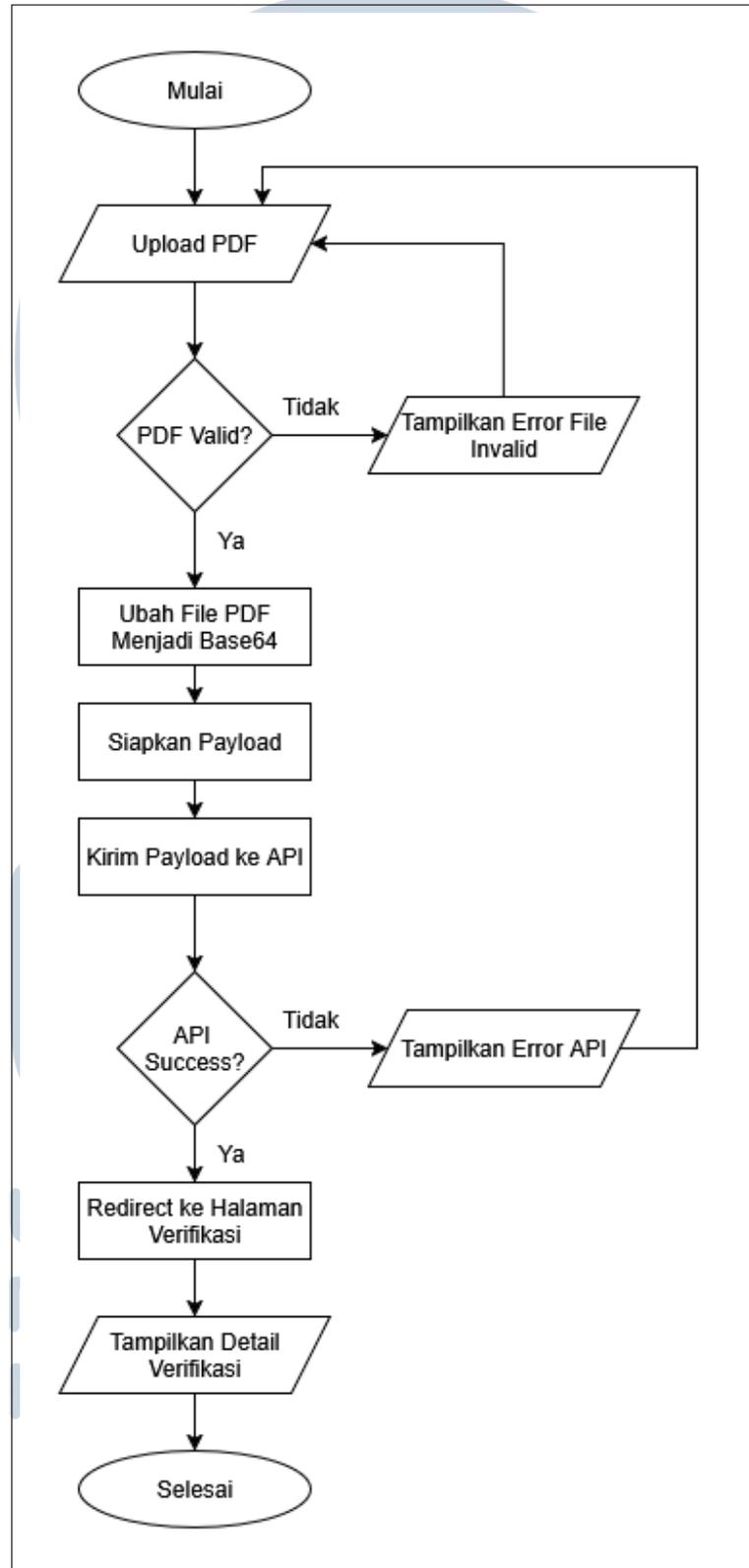


Gambar 3.36. Flowchart Proses Tanda Tangan Digital dengan API

Proses tanda tangan digital merupakan inti dari sistem e-sign yang melibatkan integrasi dengan API eksternal. Proses dimulai dengan pemilihan metode tanda tangan oleh pengguna, yaitu "Dengan QR Code" atau "Tanpa QR Code". Jika pengguna memilih metode dengan QR Code, sistem akan meminta input tag koordinat QR yang berfungsi sebagai penanda lokasi penempatan QR code pada dokumen. Selanjutnya, pengguna diminta untuk memasukkan NIK dan passphrase untuk autentikasi. Sistem melakukan validasi NIK dengan mencocokkan data pada request untuk memastikan kesesuaian identitas penandatangan. Setelah validasi berhasil, sistem akan menyiapkan payload API yang berisi file dalam format Base64, kredensial pengguna, dan properti tanda tangan. Payload tersebut kemudian dikirim ke E-Sign API v2 menggunakan metode HTTP POST dengan autentikasi Basic Auth. Jika API merespons dengan sukses, sistem akan memproses file yang telah ditandatangani dengan melakukan decode Base64 dan menyimpan file hasil ke storage server. Status dokumen akan diupdate menjadi "completed" pada database. Apabila terjadi kegagalan pada pemanggilan API atau validasi NIK, sistem akan menampilkan pesan error yang sesuai dan mengarahkan pengguna untuk mengulangi proses.



#### 4) Verifikasi Dokumen dengan API

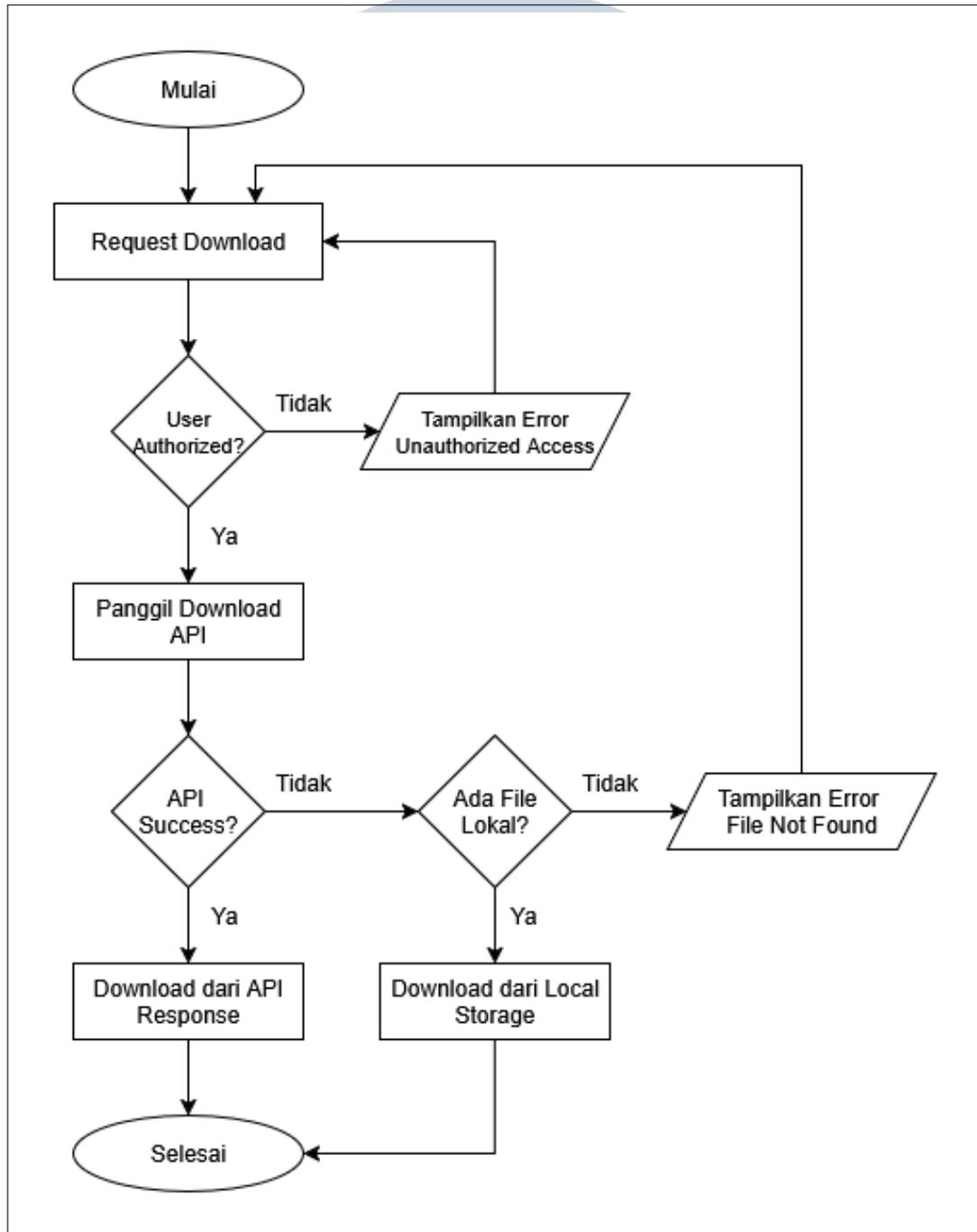


Gambar 3.37. Flowchart Verifikasi Dokumen dengan API

Proses verifikasi dokumen bertujuan untuk memvalidasi keaslian dan integritas tanda tangan digital pada file PDF. Proses dimulai dengan upload file PDF yang akan diverifikasi oleh pengguna melalui interface aplikasi. Sistem melakukan validasi format file untuk memastikan bahwa file yang diupload adalah PDF yang valid dengan ukuran maksimum 10MB. Jika validasi gagal, sistem akan menampilkan pesan error dan meminta pengguna mengupload file yang sesuai. Setelah validasi berhasil, sistem akan mengkonversi file PDF ke format Base64 untuk dikirim melalui API. File yang telah diekode kemudian dikirim ke Verify API v2 menggunakan metode HTTP POST. API akan melakukan analisis terhadap tanda tangan digital yang terdapat pada dokumen dan mengembalikan informasi detail mengenai sertifikat, timestamp, integritas dokumen, dan status validitas tanda tangan. Sistem akan mentransformasi response API ke format standar yang dapat ditampilkan pada interface pengguna. Hasil verifikasi mencakup informasi penandatanganan, detail sertifikat, informasi TSA (Time Stamping Authority), dan status integritas dokumen. Jika proses verifikasi gagal, baik karena masalah koneksi API atau file yang tidak valid, sistem akan menampilkan pesan error yang informatif kepada pengguna.



## 5) Proses Download Dokumen dengan API Fallback



Gambar 3.38. Flowchart Download Dokumen dengan API Fallback

Sistem download dokumen dirancang dengan mekanisme fallback untuk memastikan ketersediaan file bagi pengguna yang berwenang. Proses dimulai dengan permintaan download dari pengguna yang diikuti dengan pengecekan

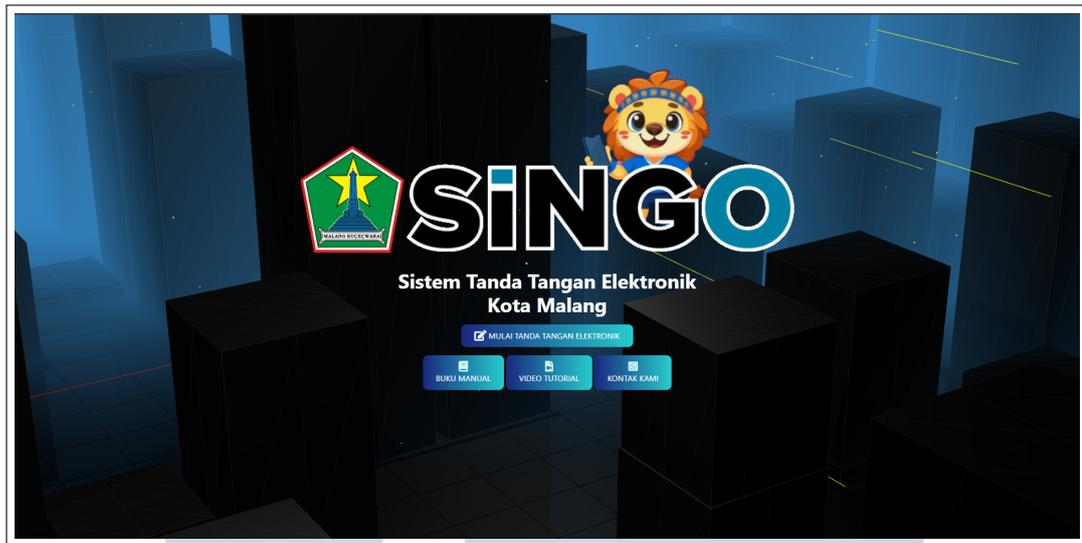
otorisasi untuk memastikan bahwa pengguna memiliki hak akses terhadap dokumen yang diminta. Sistem melakukan verifikasi bahwa pengguna hanya dapat mengakses dokumen milik sendiri (untuk role user) atau semua dokumen (untuk role admin). Setelah otorisasi berhasil, sistem akan memanggil Download API untuk mengambil file dari server eksternal. Jika API merespons dengan sukses dan mengembalikan file yang valid, sistem akan melayani download langsung dari response API tersebut. Namun, jika API gagal merespons atau mengembalikan error 404 (file tidak ditemukan), sistem akan mengaktifkan mekanisme fallback dengan mencari file di local storage server. Jika file ditemukan di local storage, sistem akan melayani download dari file lokal tersebut. Mekanisme fallback ini memastikan kontinuitas layanan meskipun terjadi gangguan pada server eksternal. Apabila file tidak ditemukan baik di API maupun local storage, sistem akan menampilkan pesan error "File Not Found" kepada pengguna. Seluruh proses download dilengkapi dengan header HTTP yang sesuai untuk memastikan file dapat diunduh dengan benar oleh browser pengguna.

#### **D Desain Website E-Signature**

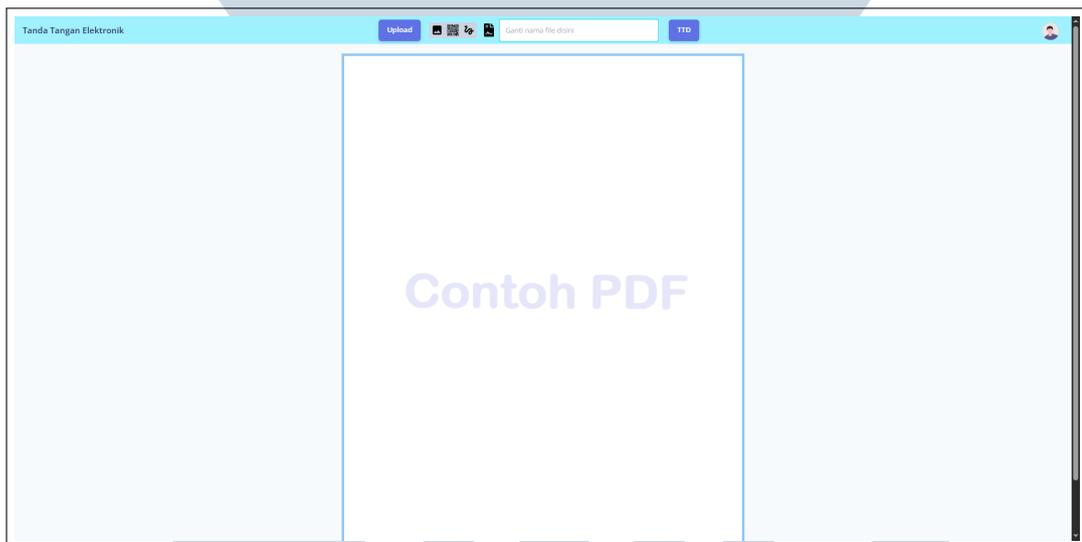
Desain website e-signature dibagi menjadi dua bagian, berikut penjelasannya:

##### **1) Desain Bagian Guest**

Desain sisi tidak login (Guest) pada sistem E-Signature ini dirancang untuk memberikan akses terbatas kepada pengguna yang belum melakukan autentikasi. Hal ini memungkinkan pengguna untuk mengakses beberapa fungsi dasar seperti menandatangani dokumen pribadi dan melakukan verifikasi, namun dengan keterbatasan tidak dapat menandatangani dokumen secara massal atau *request* tanda tangan.



Gambar 3.39. Desain Halaman Depan SINGO



Gambar 3.40. Desain Halaman Tanda Tangan SINGO

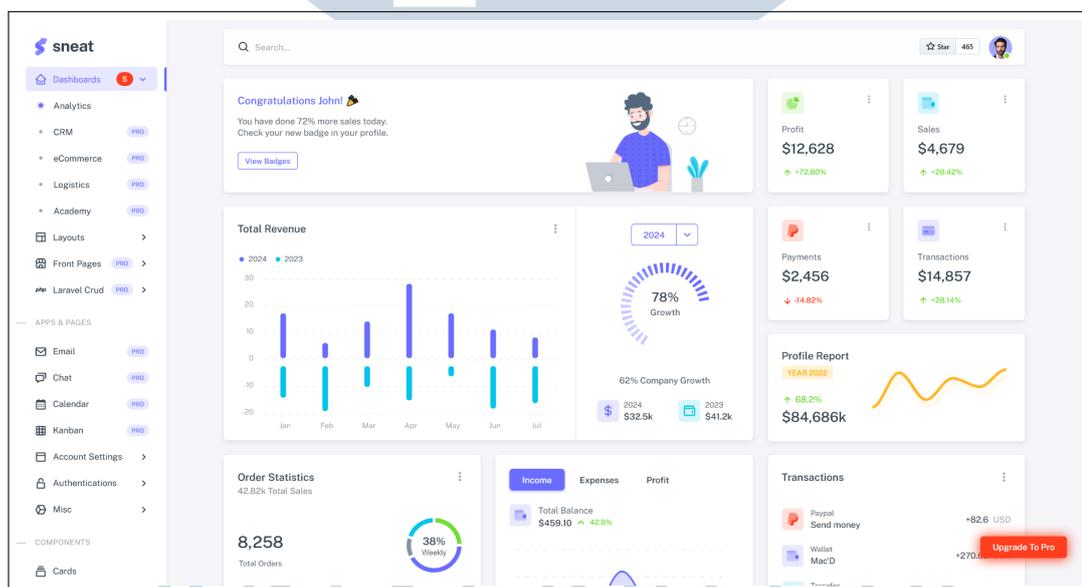
Desain website E-Signature Balikpapan terinspirasi dari website SINGO (Sistem Tanda Tangan Elektronik Kota Malang). Website SINGO menjadi inspirasi utama dalam pengembangan UI sistem E-Sign Balikpapan. SINGO menampilkan desain modern dengan background tiga dimensi menyerupai kota yang memberikan kesan futuristik dan profesional, sesuai dengan visi digitalisasi pelayanan publik. Layout halaman utama mengikuti pola SINGO dengan judul website berukuran besar sebagai focal point dan tombol call-to-action prominent di bawahnya. Pada E-Sign Balikpapan, ini diubah menjadi heading "ESIGN BALIKPAPAN" dengan font

ekstrabesar dan tombol "MULAI TANDA TANGAN", mengadopsi prinsip hierarki visual yang efektif.

Halaman tanda tangan mengambil inspirasi layout elemen SINGO dalam pengorganisasian fitur upload gambar, pembuatan tanda tangan digital, dan upload PDF. Setiap komponen menggunakan konsep modular interface yang terstruktur dengan visual hierarchy yang jelas, memungkinkan workflow yang intuitif dari upload hingga penyelesaian signing. Adopsi inspirasi SINGO memberikan konsistensi visual dengan standar platform digital pemerintah, menciptakan familiarity pengguna dan professional appearance yang credible untuk aplikasi tanda tangan elektronik yang menangani dokumen resmi.

## 2) Desain Bagian User & Admin

Desain sisi login dan admin ini mirip secara besar seperti website KGB, namun dengan beberapa penyesuaian untuk memenuhi kebutuhan sistem E-Signature. Berikut penjelasannya:



Gambar 3.41. Demo Laravel Sneat - Free Version

Sumber: [19]

Laravel Sneat merupakan template admin dashboard yang digunakan dalam pengembangan UI pengguna untuk sistem *e-signature* ini. Template ini dipilih karena konsistensinya dengan pengembangan website KGB sebelumnya, dimana Laravel Sneat juga telah terbukti efektif dalam menyediakan komponen UI yang

responsif dan modern.

Implementasi Laravel Sneat pada sistem *e-signature* mempertahankan sebagian besar elemen visual dan struktur layout yang sama dengan pengembangan website KGB. Kesamaan ini mencakup skema warna, tipografi, navigasi sidebar, dan komponen form yang telah familiar bagi pengguna. Konsistensi visual ini penting untuk mempertahankan *user experience* yang seragam across platform yang dikembangkan oleh tim.

Meskipun demikian, beberapa penyesuaian visual dilakukan untuk mengoptimalkan tampilan sesuai dengan kebutuhan spesifik sistem *e-signature*. Modifikasi yang dilakukan meliputi penyesuaian *spacing* antar elemen, penataan ulang beberapa komponen untuk meningkatkan keterbacaan, dan penambahan elemen visual khusus yang mendukung proses penandatanganan elektronik. Perubahan-perubahan ini tetap mempertahankan identitas visual Laravel Sneat sambil meningkatkan aspek fungsionalitas dan estetika yang lebih rapi.

### 3.7.3 Implementasi Website E-Signature

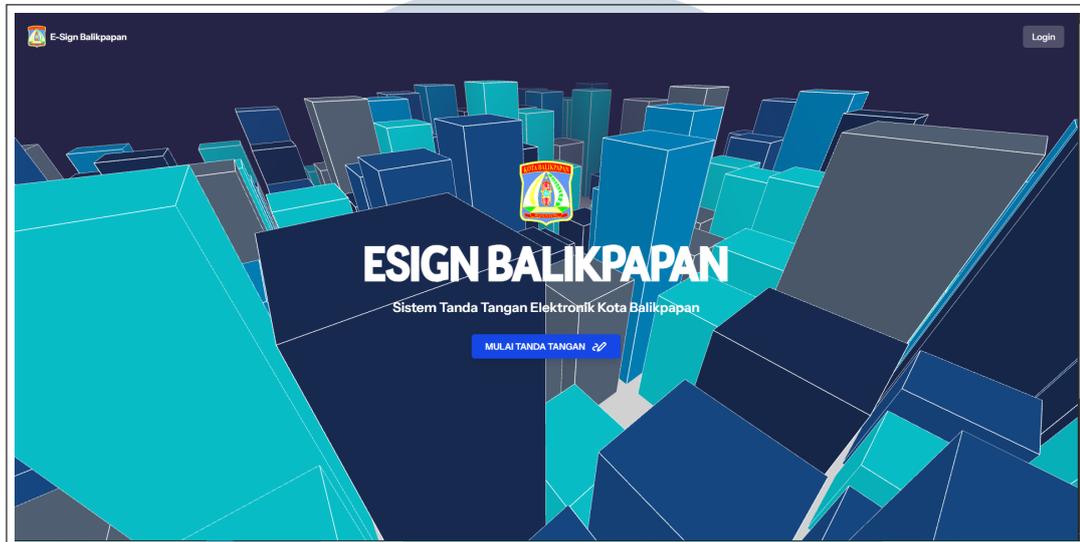
Implementasi website E-Signature Kota dilakukan menggunakan framework Laravel dengan tema Sneat yang responsif dan modern pada bagian User & Admin. Implementasi meliputi:

#### 1) Bagian Guest

Bagian guest pada website E-Signature Kota Balikpapan dirancang untuk memberikan akses terbatas kepada pengguna yang belum melakukan login. Pengguna dapat mengakses halaman depan, melakukan tanda tangan pada dokumen pribadi, dan melakukan verifikasi tanda tangan. Berikut penjelasannya:

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

## a) Halaman Depan



Gambar 3.42. Halaman Depan E-Signature Balikpapan

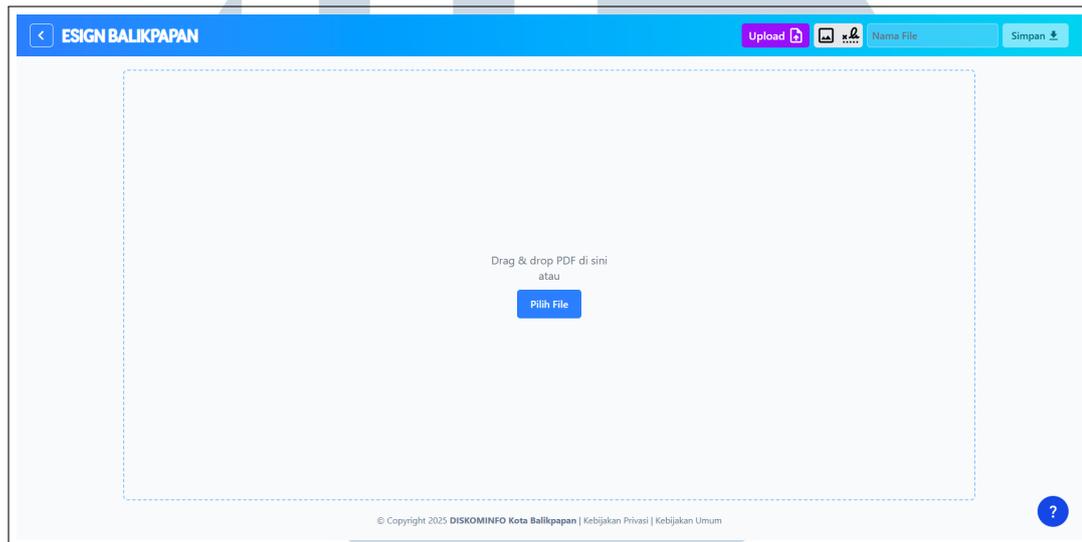
Halaman depan ini merupakan antarmuka utama yang pertama kali dilihat pengguna ketika mengakses sistem E-Sign Balikpapan. Halaman ini dirancang sebagai landing page yang menyediakan akses cepat dan intuitif ke seluruh fitur sistem tanda tangan elektronik. Implementasi halaman menggunakan teknologi modern dengan kombinasi Laravel Blade sebagai template engine, Tailwind CSS untuk styling responsif, dan Three.js untuk menciptakan background interaktif tiga dimensi.

Struktur navigasi pada halaman beranda mengimplementasikan sistem autentikasi dinamis yang menampilkan menu berbeda berdasarkan status login pengguna. Untuk pengguna yang belum login (guest), sistem menampilkan tombol "Login" yang mengarahkan ke halaman autentikasi. Sedangkan untuk pengguna yang sudah terautentikasi, navigasi akan menampilkan tombol "Dashboard" untuk mengakses panel kontrol sesuai role mereka dan tombol "Logout" untuk mengakhiri sesi.

Fungsi utama halaman beranda adalah menyediakan akses langsung ke fitur tanda tangan dokumen melalui tombol "MULAI TANDA TANGAN" yang mengarahkan ke endpoint `/sign`. Fitur ini memungkinkan dua skenario penggunaan: pertama, pengguna tanpa login dapat melakukan tanda tangan dokumen pribadi secara langsung; dan kedua, admin dan user dapat melakukan tanda tangan dokumen secara massal.

Halaman depan menampilkan latar belakang 3D menggunakan *Three.js* yang diintegrasikan langsung pada template Blade.

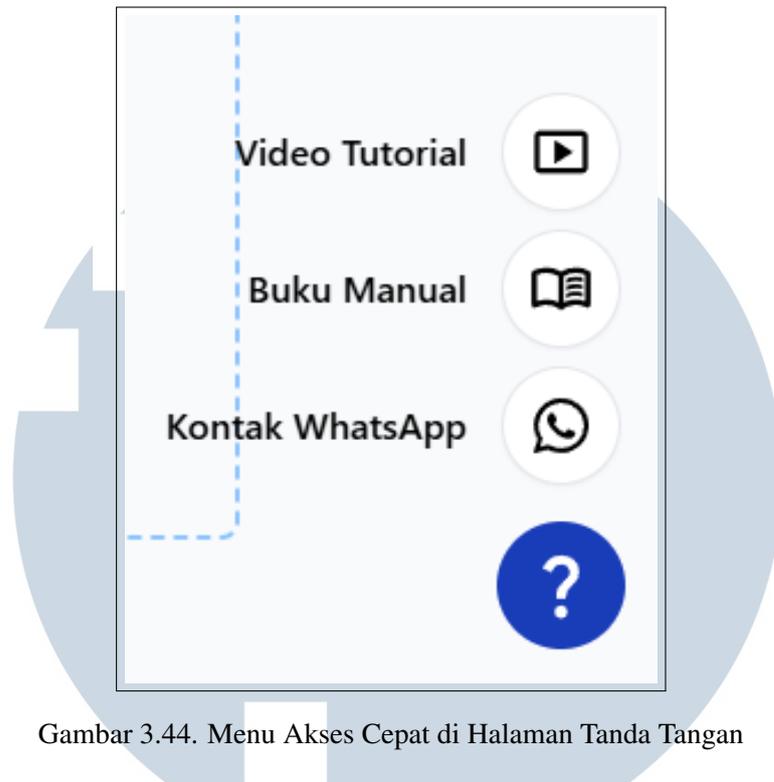
## b) Halaman Tanda Tangan Dokumen Sendiri



Gambar 3.43. Halaman Tanda Tangan E-Signature Balikpapan

Halaman tanda tangan digital merupakan inti dari sistem E-Sign Balikpapan yang menyediakan UI komprehensif untuk proses penandatanganan dokumen elektronik. Halaman ini dirancang dengan pendekatan *user-centered design* yang memungkinkan pengguna melakukan tanda tangan dokumen secara intuitif dan efisien. Antarmuka menggunakan kombinasi teknologi modern seperti *PDF.js* untuk merender dokumen, *Fabric.js* untuk manipulasi *canvas*, *Signature Pad* untuk pembuatan tanda tangan digital, dan *PDF-lib* untuk modifikasi dokumen PDF secara *real-time*.

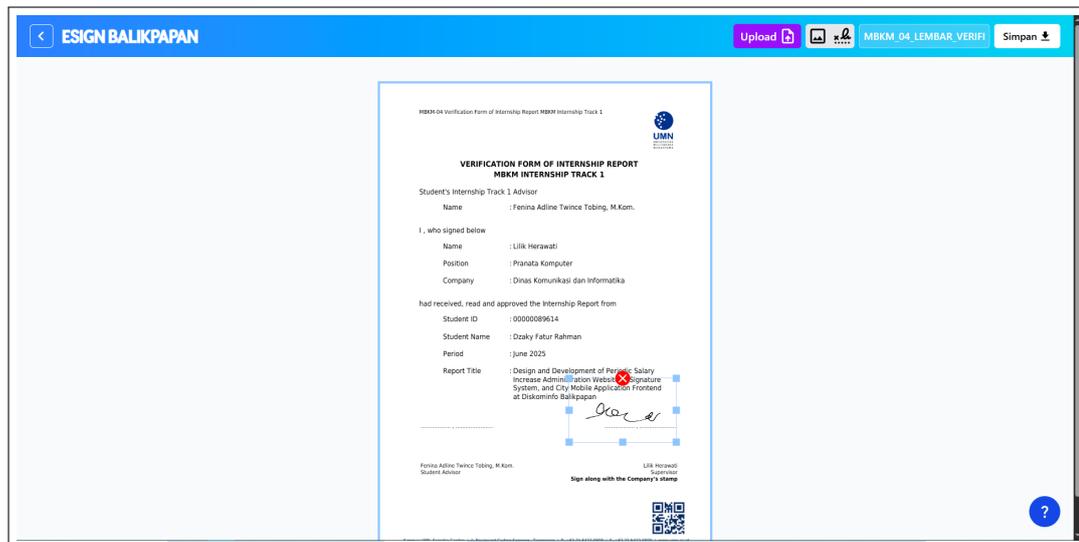
Struktur navigasi halaman menampilkan *navbar* dengan gradien biru yang konsisten dengan tema visual sistem, dilengkapi dengan tombol kembali untuk navigasi, merek "ESIGN BALIKPAPAN", dan *toolbar* fungsional. *Toolbar* utama terdiri dari tombol unggah untuk memuat dokumen PDF, alat untuk menambahkan gambar dan tanda tangan, *input field* untuk nama berkas, dan tombol simpan untuk finalisasi dokumen. Area kerja utama menampilkan *drop zone* yang responsif, memungkinkan pengguna melakukan *drag-and-drop* berkas PDF atau menggunakan tombol "Pilih File" untuk mengunggah dokumen.



Gambar 3.44. Menu Akses Cepat di Halaman Tanda Tangan

*Speed dial menu* yang terletak di kanan bawah halaman menyediakan akses langsung ke *resources* pendukung sistem. Menu ini mencakup tombol "Tutorial Video" untuk panduan visual penggunaan sistem, "Buku Manual" untuk dokumentasi lengkap, dan "Kontak WhatsApp" untuk dukungan langsung. Implementasi *speed dial* menggunakan *framework* Flowbite dengan animasi halus dan *positioning* yang optimal, memastikan *accessibility* tanpa mengganggu *workflow* utama pengguna.

UIMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.45. Contoh Tanda Tangan dengan PDF dan TTD

Fitur manipulasi dokumen memungkinkan pengguna menambahkan elemen visual ke dalam PDF melalui dua metode utama. Pertama, fitur "Tambah Gambar" yang memungkinkan unggah gambar dalam format JPG, JPEG, atau PNG untuk ditempatkan sebagai tanda tangan visual. Kedua, fitur "Tambah Tanda Tangan" yang membuka *modal signature pad* dengan *canvas* berukuran 500x300 piksel untuk membuat tanda tangan digital secara langsung menggunakan *mouse* atau *touch input*. Setiap elemen yang ditambahkan dapat dimanipulasi secara interaktif, termasuk mengubah posisi, ukuran, rotasi, dan penghapusan elemen sesuai kebutuhan pengguna.

U M N  
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

The image shows a modal window titled "Konfirmasi Simpan" (Save Confirmation) overlaid on a dark background. The modal contains the following elements:

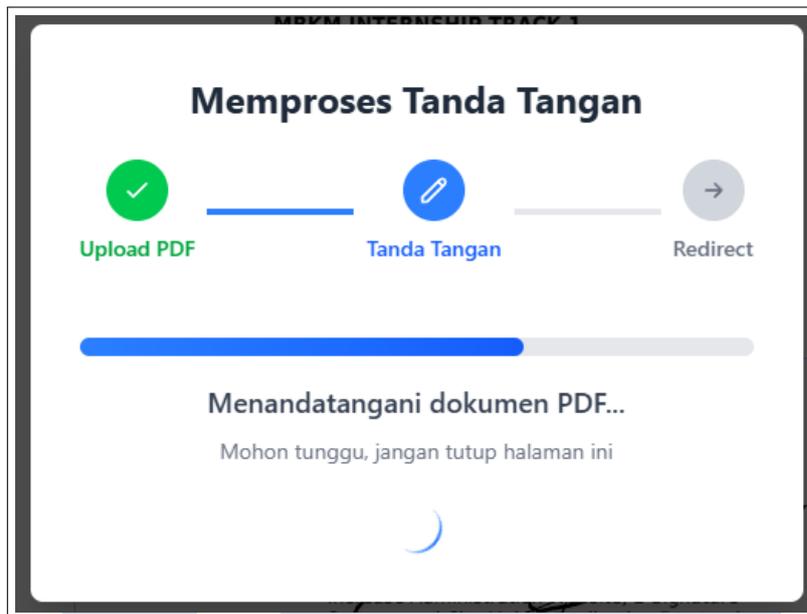
- NIK:** A text input field containing the number "647000007730006".
- Passphrase:** A text input field with masked characters (dots) and a toggle icon on the right.
- Alasan:** A dropdown menu with the selected option "Tanda Tangan Rekom Segel".
- Buttons:** Two buttons at the bottom: "Batal" (Cancel) and "Simpan" (Save).

The background of the page features a signature and a large, semi-transparent "UMN" watermark.

Gambar 3.46. Form Input Tanda Tangan - E-Signature Balikpapan

Proses tanda tangan elektronik difasilitasi melalui *modal form* yang muncul saat pengguna mengklik tombol simpan. Formulir ini mengumpulkan data penting berupa NIK dengan validasi numerik, *passphrase* dengan fitur *show/hide* untuk keamanan *input*, dan *dropdown* alasan tanda tangan dengan opsi yang telah ditentukan seperti "Tanda Tangan Rekomendasi Segel" dan "Tanda Tangan Dokumen". Validasi *client-side* memastikan semua *field required* terisi sebelum *submission* ke *backend API*.

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

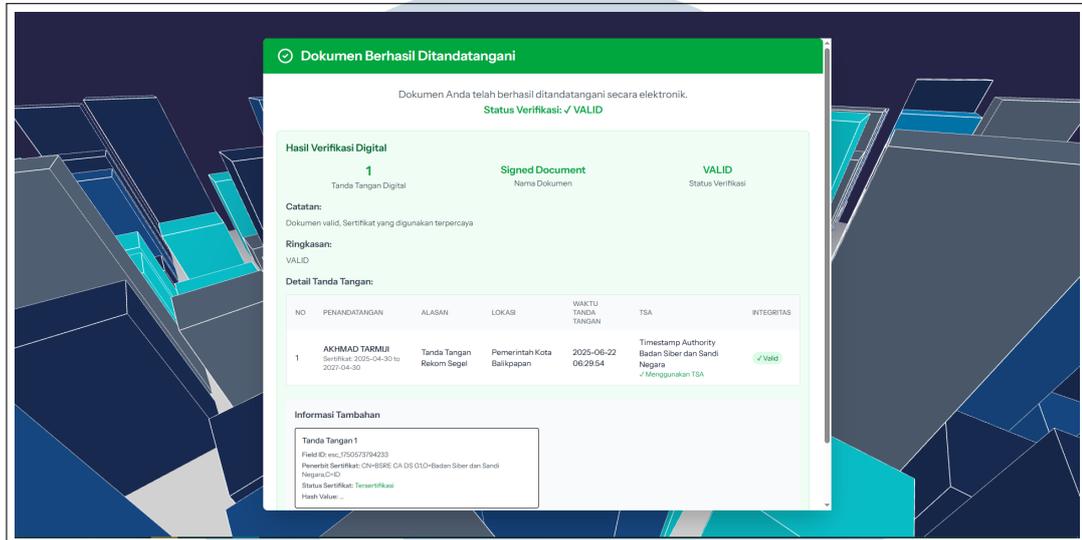


Gambar 3.47. Progress Bar - E-Signature Balikpapan

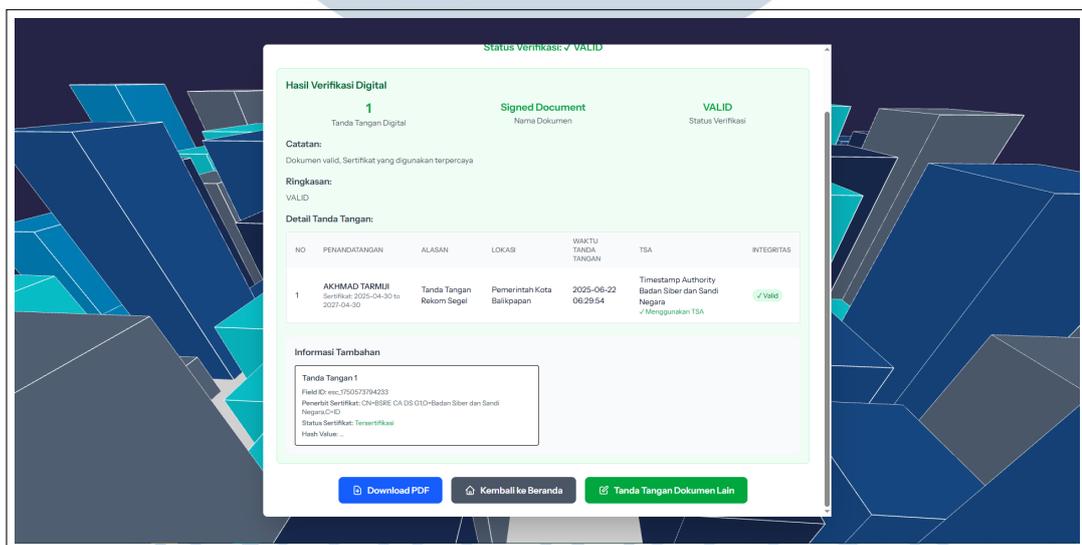
*Progress indicator* memberikan *feedback* visual secara *real-time* selama proses tanda tangan berlangsung. Sistem menampilkan tiga tahap progres: unggah PDF dengan indikator 33%, proses tanda tangan dengan indikator 66%, dan *redirect* hasil dengan indikator 100%. Setiap tahap dilengkapi dengan ikon visual, *progress bar* dengan animasi gradien, teks status yang informatif, dan *spinning loader* untuk menunjukkan aktivitas pemrosesan. *Modal progress* menggunakan *z-index* tinggi (60) dan *background overlay* untuk memastikan pengguna tidak melakukan aksi lain selama proses berlangsung.

UIN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

### c) Halaman Verifikasi Tanda Tangan



Gambar 3.48. Halaman Verifikasi Bagian 1 - E-Signature Balikpapan



Gambar 3.49. Halaman Verifikasi Bagian 2 - E-Signature Balikpapan

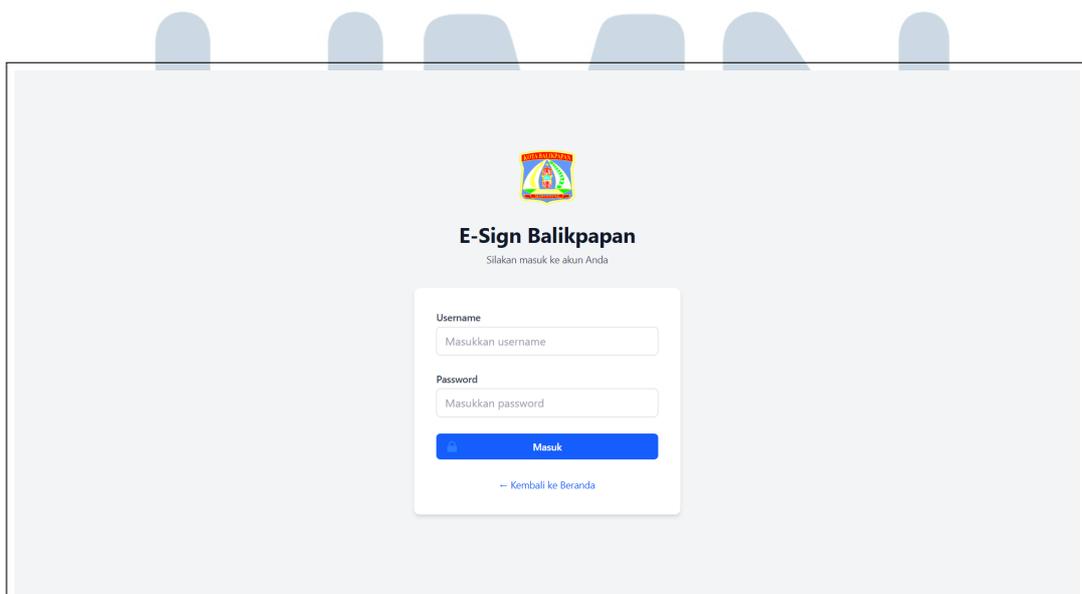
Halaman verifikasi merupakan halaman konfirmasi yang ditampilkan setelah proses tanda tangan dokumen berhasil diselesaikan, berfungsi sebagai *feedback* komprehensif terhadap status dan validitas dokumen yang telah ditandatangani. Halaman ini menggunakan *background* Three.js yang konsisten seperti halaman depan dengan desain sistem dan menampilkan *modal overlay* berukuran maksimal dengan informasi verifikasi lengkap.

Sistem verifikasi otomatis menganalisis dokumen yang telah ditandatangani. Informasi verifikasi mencakup jumlah tanda tangan digital, nama dokumen, dan status verifikasi yang ditampilkan dalam *grid layout* yang terstruktur dengan indikator visual berupa angka dan teks status.

Detail verifikasi komprehensif disajikan dalam bentuk tabel yang menampilkan informasi setiap tanda tangan secara individual, meliputi nama penandatanganan, alasan penandatanganan, lokasi, waktu tanda tangan, status *Time Stamp Authority* (TSA), dan integritas dokumen. Setiap entri tanda tangan juga menyertakan informasi sertifikat digital seperti validitas sertifikat, penerbit sertifikat, status sertifikasi pengguna, dan *hash value* untuk verifikasi kriptografis.

Fungsionalitas utama halaman mencakup tiga aksi: tombol *download* untuk mengunduh dokumen yang telah ditandatangani dengan sistem *one-time download* yang mencegah unduhan berulang, tombol kembali ke beranda untuk navigasi, dan tombol tanda tangan dokumen lain untuk melanjutkan proses signing. Sistem mengimplementasikan *auto-download* dengan parameter URL dan *local storage* untuk melacak status unduhan, memberikan peringatan jika dokumen sudah pernah diunduh sebelumnya serta menangani berbagai skenario error dengan pesan yang informatif.

## 2) Bagian Login



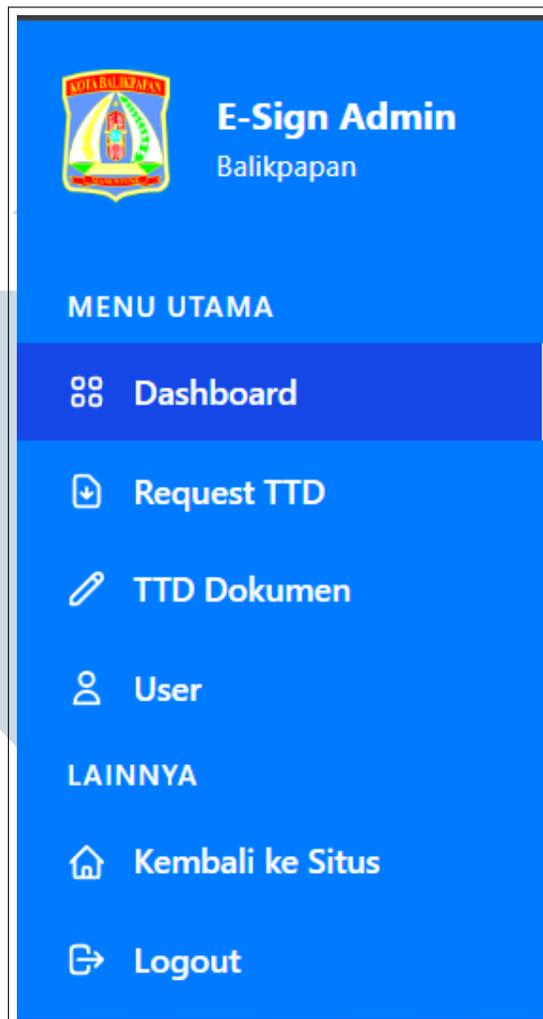
Gambar 3.50. Halaman Login E-Signature Balikpapan

Halaman *login* berfungsi sebagai gerbang autentikasi untuk mengakses fitur-fitur terbatas dalam sistem E-Sign Balikpapan, memungkinkan pengguna mengidentifikasi diri sesuai dengan *role* yang dimiliki. Halaman ini menerapkan desain minimalis dengan *background* abu-abu dan *card* putih terpusat yang menampilkan logo BPN, judul aplikasi, dan formulir *login* yang sederhana. Implementasi menggunakan Laravel Blade dengan validasi *server-side* dan *client-side*, dilengkapi *error handling* yang menampilkan pesan kesalahan secara spesifik untuk setiap *field input*.

Formulir autentikasi terdiri dari dua *field* utama yaitu *username* dan *password* dengan validasi *required* dan *styling* responsif menggunakan Tailwind CSS. Sistem menerapkan *focus states* dengan *ring* biru untuk meningkatkan *user experience* dan indikator visual *error* dengan *border* merah ketika terjadi kesalahan validasi. Tombol submit dilengkapi dengan ikon gembok dan efek *hover* yang halus, memberikan *feedback* visual yang jelas kepada pengguna saat melakukan interaksi.

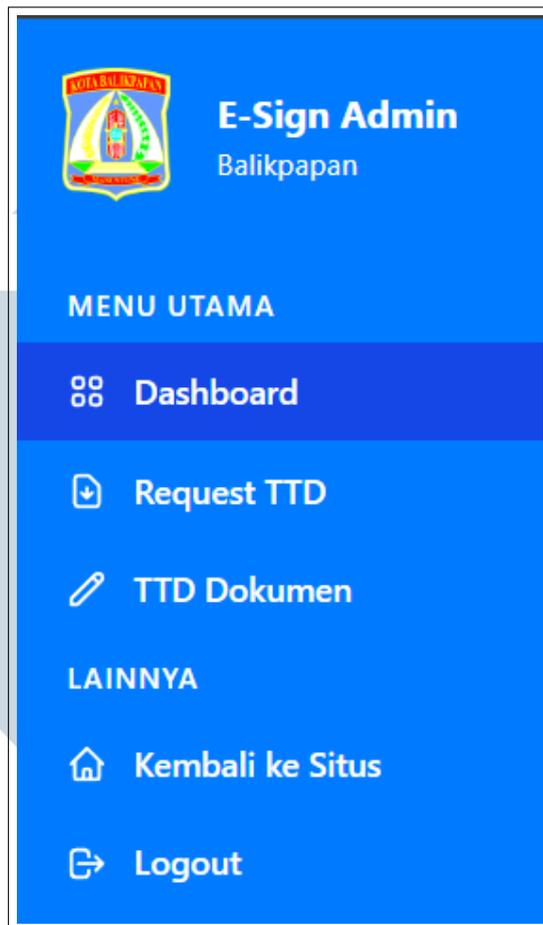
Proses autentikasi menggunakan Laravel's built-in authentication system dengan *middleware* yang mengarahkan pengguna ke *dashboard* sesuai *role* mereka setelah berhasil *login*. Sistem juga menyediakan navigasi kembali ke beranda dengan *link* "Kembali ke Beranda" untuk pengguna yang ingin mengakses fitur publik tanpa autentikasi. *Error handling* mencakup validasi *input*, penanganan kredensial yang salah, dan perlindungan CSRF melalui token Laravel untuk keamanan aplikasi.





Gambar 3.51. Sidebar Admin E-Signature Balikpapan

UIN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.52. Siderbar User E-Signature Balikpapan

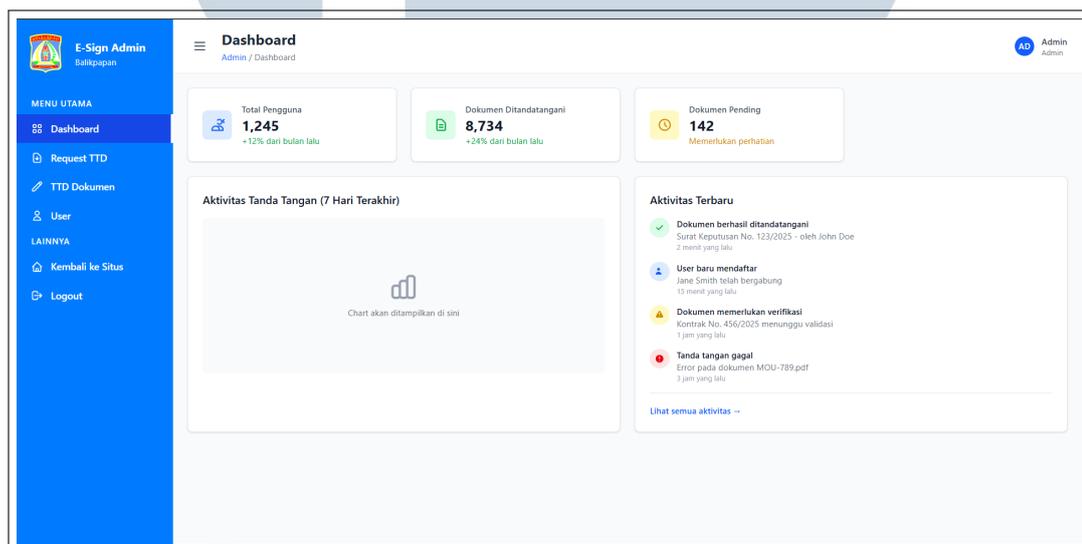
*Sidebar* admin dan user dalam sistem E-Sign Balikpapan menerapkan konsep *role-based access control* yang membedakan hak akses berdasarkan tingkat kewenangan pengguna. *Sidebar* admin menyediakan akses lengkap ke semua fitur sistem meliputi *Dashboard* untuk monitoring aktivitas, *Request TTD* untuk mengelola permintaan tanda tangan dari semua pengguna, *TTD Dokumen* untuk proses penandatanganan massal, dan menu *User* khusus yang hanya muncul untuk role admin untuk manajemen akun pengguna. Menu navigasi admin juga dilengkapi dengan fitur *collapsible sidebar* menggunakan Alpine.js yang memungkinkan ekspansi dan kontraksi untuk optimasi ruang layar.

Sebaliknya, *sidebar* user memiliki akses terbatas yang hanya mencakup fitur-fitur yang berkaitan dengan dokumen milik mereka sendiri. User dapat mengakses *Dashboard* personal untuk melihat statistik dokumen pribadi, *Request TTD* untuk membuat dan mengelola permintaan tanda tangan sendiri, dan *TTD Dokumen* untuk menandatangani dokumen yang mereka buat. Menu *User* untuk

manajemen akun tidak tersedia bagi role user, mencerminkan prinsip *least privilege* dalam keamanan sistem.

Implementasi kondisional menggunakan direktif Blade `@if(auth()->check() && auth()->user()->role === 'admin')` memastikan menu administratif hanya ditampilkan untuk pengguna dengan role admin, sementara kedua interface menggunakan *styling* visual yang konsisten dengan *background* biru, ikon SVG, dan *active state indicator* berupa *border* putih. Perbedaan utama terletak pada cakupan fungsionalitas dimana admin memiliki akses universal terhadap semua data dan operasi sistem, sedangkan user dibatasi pada *scope* data dan operasi yang berkaitan dengan akun mereka sendiri.

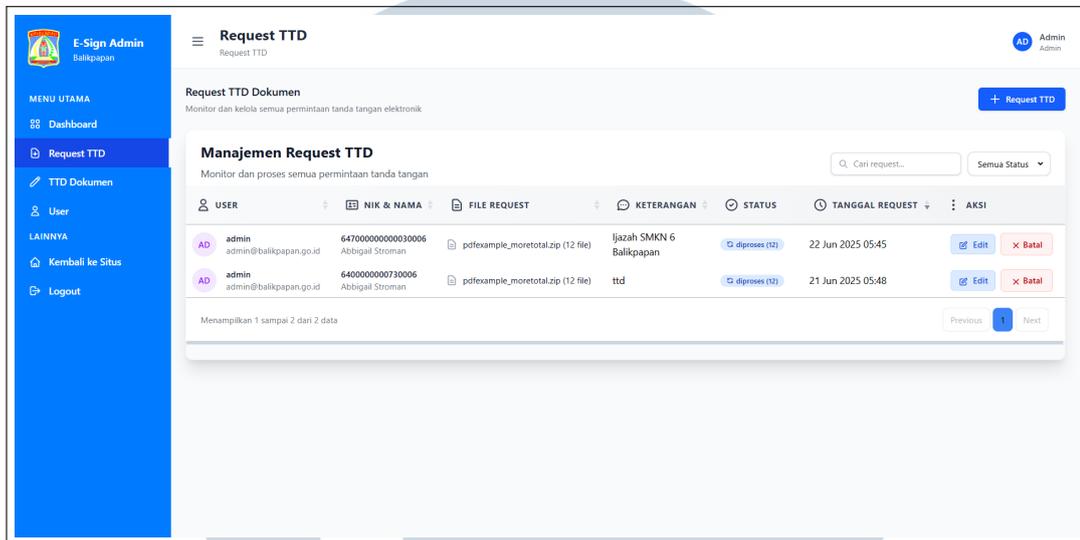
### a) Halaman Dashboard



Gambar 3.53. Halaman Dashboard E-Signature Balikpapan

*Dashboard* admin menyediakan tampilan ringkasan komprehensif aktivitas sistem E-Sign Balikpapan melalui *grid layout* responsif yang menampilkan statistik utama dalam bentuk *cards*. Komponen utama meliputi *stats cards* yang menampilkan total pengguna, dokumen ditandatangani, dan dokumen *pending* dengan indikator pertumbuhan, area *chart placeholder* untuk visualisasi aktivitas tanda tangan, dan panel aktivitas terbaru yang menampilkan *real-time updates*. Setiap elemen dilengkapi ikon SVG dan *color coding* untuk memberikan *visual feedback* yang intuitif bagi administrator dalam memantau performa dan aktivitas sistem secara keseluruhan.

## b) Halaman Request TTD



Gambar 3.54. Halaman Request E-Signature Balikpapan

Tambah Request TTD

NIK  
64700000000030006

Nama Lengkap  
Abbigail Stroman

Keterangan Request  
Ijazah SMKN 6 Balikpapan

Upload File  
Upload File atau drag and drop  
PDF atau ZIP (max 20MB)

pdfexample\_moretotal.zip (0.54 MB, ZIP Archive)

Batal Buat Request

Gambar 3.55. Add Request - E-Signature Balikpapan

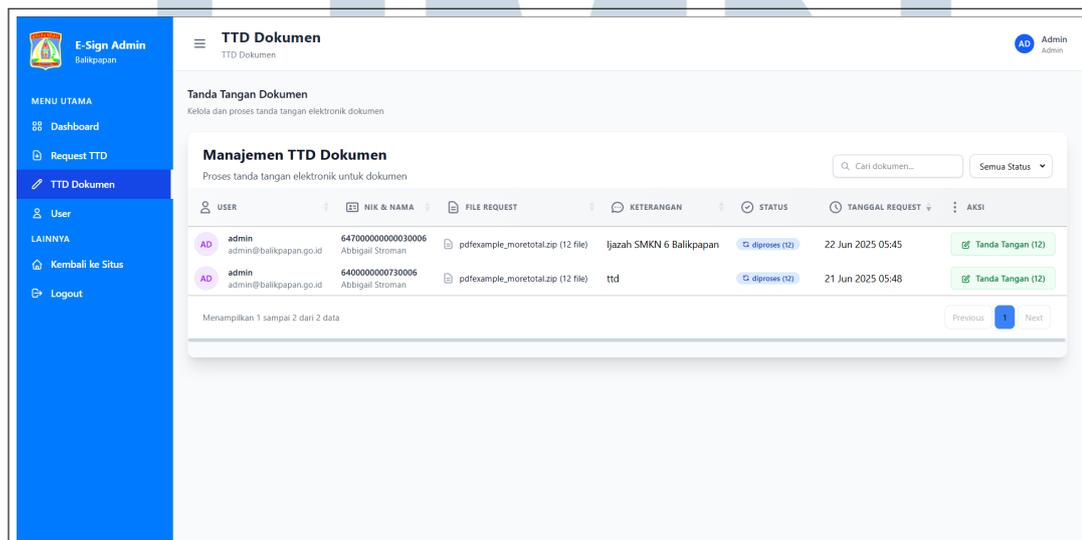
Halaman manajemen *request* TTD merupakan fitur sentral yang memungkinkan admin dan user untuk membuat, mengelola, dan memantau permintaan tanda tangan dokumen dalam sistem E-Sign Balikpapan. Halaman ini mengimplementasikan antarmuka berbasis tabel dengan fungsionalitas *CRUD* lengkap yang menampilkan informasi komprehensif setiap *request* meliputi data user, NIK dan nama pemohon, file *request*, keterangan, status, tanggal *request*, dan kolom aksi untuk operasi edit, hapus, atau lihat detail. Sistem menggunakan

DataTables jQuery dengan fitur *sorting*, *pagination*, dan pencarian *real-time* untuk optimasi pengelolaan data dalam jumlah besar.

Proses pembuatan *request* TTD dimulai dengan mengklik tombol "Request TTD" yang membuka *modal form* responsif dengan validasi *client-side* dan *server-side*. User mengisi formulir yang terdiri dari NIK dengan validasi numerik, nama lengkap pemohon, keterangan *request* dalam bentuk *textarea*, dan unggahan file dokumen yang mendukung format PDF untuk dokumen tunggal atau ZIP untuk *batch processing* multiple dokumen. *Modal* dilengkapi dengan *drag-and-drop* area untuk kemudahan upload file dan menampilkan informasi file yang dipilih sebelum *submission*.

Sistem menyediakan tiga status *request* yaitu "Menunggu" untuk *request* yang baru dibuat, "Diproses" untuk *request* yang sedang dalam tahap penandatanganan, dan "Selesai" untuk *request* yang telah ditandatangani dan siap diunduh. Admin dapat memfilter data berdasarkan status melalui *dropdown filter* dan melakukan pencarian global menggunakan *search box* yang terintegrasi dengan DataTables. Fitur edit memungkinkan modifikasi data *request* dengan opsi mengganti file dokumen yang bersifat opsional, sementara fitur hapus menyediakan konfirmasi untuk mencegah penghapusan yang tidak disengaja.

### c) Halaman TTD Dokumen



The screenshot displays the 'TTD Dokumen' page in the 'E-Sign Admin' system. The page title is 'TTD Dokumen' and the subtitle is 'Tanda Tangan Dokumen'. The main content area is titled 'Manajemen TTD Dokumen' and contains a table of document requests. The table has the following columns: USER, NIK & NAMA, FILE REQUEST, KETERANGAN, STATUS, and TANGGAL REQUEST. There are two rows of data, both with a status of 'Diproses (12)'. The first row shows a user named 'admin' with NIK '64700000000030006' and a file named 'pdfexample\_moretotal.zip (12 file)'. The second row shows a user named 'admin' with NIK '6400000000730006' and a file named 'pdfexample\_moretotal.zip (12 file)'. The page also features a search bar, a 'Semua Status' dropdown menu, and pagination controls at the bottom.

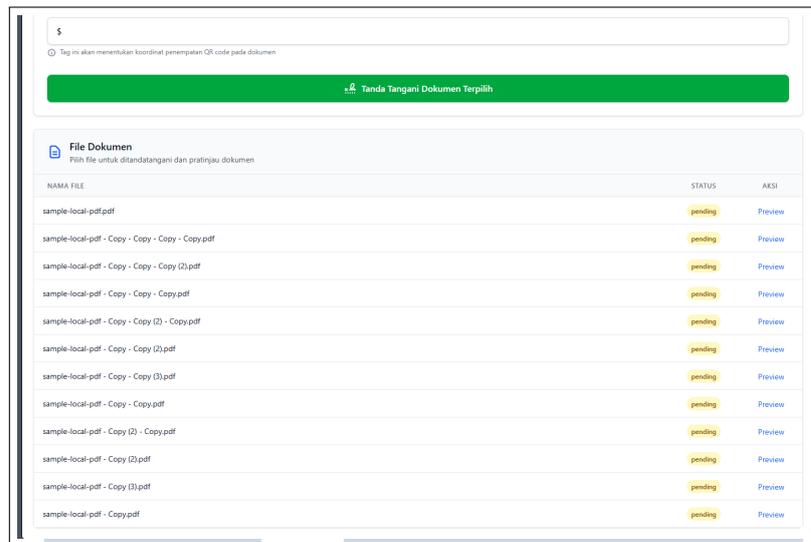
USER	NIK & NAMA	FILE REQUEST	KETERANGAN	STATUS	TANGGAL REQUEST	AKSI
admin admin@balikpapan.go.id	64700000000030006 Abbigail Stroman	pdfexample_moretotal.zip (12 file)	Ijazah SMKN 6 Balikpapan	Diproses (12)	22 Jun 2025 05:45	Tanda Tangan (12)
admin admin@balikpapan.go.id	6400000000730006 Abbigail Stroman	pdfexample_moretotal.zip (12 file)	ttd	Diproses (12)	21 Jun 2025 05:48	Tanda Tangan (12)

Gambar 3.56. Halaman TTD Dokumen E-Signature Balikpapan

Halaman TTD Dokumen merupakan fitur administratif tingkat lanjut yang memungkinkan admin dan user melakukan proses tanda tangan elektronik secara *batch* terhadap dokumen-dokumen yang telah di-*request* dalam sistem E-Sign Balikpapan. Halaman ini mengimplementasikan antarmuka berbasis tabel dengan DataTables jQuery yang menampilkan informasi lengkap setiap *request* meliputi data user, NIK dan nama pemohon, file *request*, keterangan, status proses, tanggal *request*, dan kolom aksi untuk tanda tangan *batch*. Sistem menyediakan fitur pencarian global dan *filter* status yang memungkinkan admin mencari dan menyaring dokumen berdasarkan status "Menunggu", "Diproses", atau "Selesai" dengan *loading skeleton* yang memberikan *feedback* visual selama proses pengambilan data.

Proses tanda tangan elektronik dimulai dengan mengklik tombol "Tanda Tangan" pada kolom aksi yang membuka *modal* komprehensif berukuran besar dengan beberapa komponen utama. *Modal* menampilkan statistik dokumen dalam bentuk *card* dengan visualisasi jumlah total file, file menunggu, diproses, dan selesai menggunakan warna-coded *badges*. Bagian informasi tanda tangan menyediakan formulir dengan *field* NIK yang terisi otomatis dan bersifat *readonly*, *field* passphrase dengan fitur *show/hide* untuk keamanan, dan *textarea* keterangan yang menampilkan deskripsi *request* secara *readonly*.

Gambar 3.57. TTD Dokumen Bagian 1 - E-Signature Balikpapan

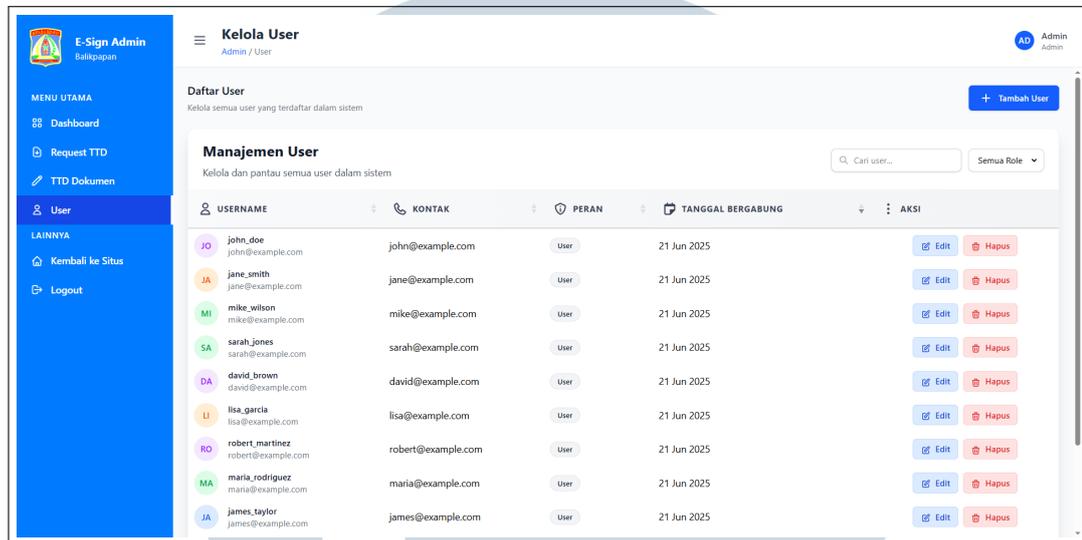


Gambar 3.58. TTD Dokumen Bagian 2 - E-Signature Balikpapan

Konfigurasi tanda tangan mencakup *dropdown* metode tanda tangan dengan dua opsi yaitu "Dengan QR Code" dan "Tanpa QR Code", dimana pemilihan "Dengan QR Code" akan menampilkan *field* tambahan "Tag Koordinat QR" yang dapat dikustomisasi untuk menentukan penempatan QR code pada dokumen. Sistem menggunakan JavaScript untuk mengontrol visibilitas *field* QR coordinate secara dinamis berdasarkan pilihan metode. Bagian bawah *modal* menampilkan tabel file dokumen yang dapat dipilih untuk ditandatangani dengan kolom nama file, status, dan aksi pratinjau atau unduh



## d) Halaman Manage Users



Gambar 3.59. Halaman Manage Users E-Signature Balikpapan

Halaman manajemen user merupakan fitur administratif khusus yang hanya dapat diakses oleh pengguna dengan *role* admin untuk mengelola seluruh akun pengguna dalam sistem E-Sign Balikpapan. Halaman ini mengimplementasikan konsep *CRUD* (Create, Read, Update, Delete) lengkap dengan antarmuka berbasis tabel yang menampilkan informasi komprehensif setiap user meliputi *username*, kontak, *role*, tanggal bergabung, dan kolom aksi untuk operasi edit atau hapus. Sistem menggunakan DataTables jQuery untuk menyediakan fitur *sorting*, *pagination*, dan pencarian *real-time* yang memudahkan admin dalam mengelola data user dalam jumlah besar.

Fungsionalitas pencarian dan *filtering* tersedia melalui *search box* global dan *dropdown filter role* yang memungkinkan admin mencari user spesifik berdasarkan nama atau menyaring berdasarkan *role* tertentu. Sistem menampilkan *loading skeleton* dengan animasi *pulse* selama proses pengambilan data untuk memberikan *feedback* visual yang baik kepada pengguna. Tombol "Tambah User" di bagian atas halaman membuka *modal form* yang memungkinkan admin membuat akun baru dengan *field username*, kontak, *password*, dan *role*.

Gambar 3.60. Add User - E-Signature Balikpapan

*Modal form* user menggunakan desain responsif dengan validasi *client-side* dan *server-side*, dilengkapi fitur *show/hide password* untuk kemudahan *input*. Proses *create* dan *update* user menggunakan *error handling* yang menampilkan pesan kesalahan spesifik untuk setiap *field*. Sistem juga menyediakan fungsi edit yang memuat data user ke dalam *modal* yang sama dengan *field password* opsional, memungkinkan admin mengubah informasi user tanpa harus mengganti *password* jika tidak diperlukan.

Keamanan halaman diimplementasikan melalui *middleware* Laravel yang memverifikasi *role* admin, *CSRF protection* pada semua *form submission*, dan validasi *input* yang ketat. Integrasi dengan *backend API* menggunakan *route* khusus untuk operasi data user dengan *response* JSON, mendukung operasi asinkron yang efisien. Halaman ini menjadi pusat kontrol utama bagi admin untuk mengelola *user access control* dan memantau aktivitas pengguna dalam ekosistem sistem E-Sign

Balikpapan.

#### **3.7.4 Pengujian Website E-Signature**

Pengujian Website E-Signature memerlukan pendekatan khusus karena integrasi dengan API eksternal BSR E. *API integration testing* mencakup pengujian integrasi dengan *BSR E-Sign API v2*, *mock testing* untuk skenario kegagalan API, testing autentikasi dengan *NIK* dan *passphrase*, serta validasi format berkas *PDF* dan *ZIP*. Pengujian *frontend* meliputi testing *PDF.js rendering* untuk berbagai jenis *PDF*, pengujian manipulasi *kanvas Fabric.js*, testing fungsionalitas *Signature Pad*, dan pengujian performa latar belakang *Three.js*. *Security testing* dilakukan untuk memvalidasi keamanan *file upload*, proteksi *CSRF*, testing *role-based access control*, dan enkripsi data untuk informasi sensitif. Pengujian *end-to-end* mencakup alur kerja penandatanganan lengkap dari unggah hingga verifikasi dokumen, testing *bulk signing* dengan multiple dokumen, pengujian verifikasi dokumen dengan *PDF* tertandatangani, dan testing mekanisme *fallback* untuk kegagalan API.

#### **3.7.5 Deployment & Maintenance Website E-Signature**

Deployment website E-Signature dilakukan oleh tim IT DATSI Diskominfo Balikpapan pada server produksi dengan domain resmi <https://edocument.balikpapan.go.id/>. Proses deployment menggunakan infrastruktur server pemerintah kota yang telah dikonfigurasi khusus untuk menangani integrasi dengan API E-Sign Client Service dari BSR E (Balai Besar Sertifikasi Elektronik). Tim IT DATSI bertanggung jawab penuh dalam konfigurasi server E-Sign Client Service, pengaturan database produksi, setup koneksi ke API BSR E, serta optimisasi performa aplikasi untuk menangani proses penandatanganan elektronik dan verifikasi dokumen PDF. Sistem dilengkapi dengan sertifikat SSL dan standar keamanan tinggi sesuai dengan kebutuhan penandatanganan dokumen digital yang sah secara hukum.

Maintenance website E-Signature dilakukan secara berkala dengan fokus khusus pada monitoring konektivitas API E-Sign Client Service dan performa integrasi dengan server BSR E. Kegiatan maintenance meliputi pemantauan status koneksi API, backup data dokumen signed secara reguler, monitoring kapasitas storage untuk file PDF, dan update security patch sistem. Tim IT juga bertanggung jawab dalam menangani troubleshooting masalah tanda tangan elektronik,

melakukan monitoring log API calls untuk debugging, serta memberikan dukungan teknis kepada pengguna yang mengalami kendala dalam proses signing dokumen.

### 3.8 Kendala dan Solusi yang Ditemukan

Selama proses pengembangan ketiga sistem digital di Diskominfo Kota Balikpapan, ditemukan beberapa kendala teknis yang signifikan. Setiap kendala yang dihadapi memerlukan analisis mendalam dan solusi yang tepat untuk memastikan keberhasilan implementasi sistem. Berikut adalah kendala-kendala utama beserta solusi yang diterapkan:

#### 1. Inkonsistensi Koordinat dan Ukuran pada API E-Sign v1

Penggunaan API E-Sign v1 mengalami masalah serius dalam hal peletakan koordinat dan ukuran gambar tanda tangan. Koordinat yang telah diatur pengguna pada antarmuka *kanvas* tidak sesuai dengan hasil tampilan pada dokumen PDF yang diproses API. Rasio ukuran antara pratinjau di peramban dengan hasil akhir API tidak konsisten, menyebabkan tanda tangan muncul di posisi yang salah atau dengan ukuran yang tidak sesuai ekspektasi. Masalah ini sangat mengganggu pengalaman pengguna karena hasil akhir dokumen tidak sesuai dengan yang diharapkan pengguna.

Solusinya adalah migrasi ke API E-Sign v2 yang menggunakan sistem koordinat ternormalisasi (0-1) dan pengaturan ukuran yang lebih konsisten. API v2 menggunakan sistem koordinat yang relatif terhadap ukuran halaman, sehingga mengeliminasi masalah perbedaan resolusi dan faktor skala. Implementasi lapisan transformasi koordinat yang mengkonversi koordinat piksel dari *kanvas* ke koordinat ternormalisasi untuk API v2, memastikan pengalaman WYSIWYG (*What You See is What You Get*) yang akurat.

#### 2. Kompleksitas Penyaringan Data Berdasarkan SKPD

Implementasi sistem penyaringan data berdasarkan SKPD pengguna yang masuk menghadapi kompleksitas dalam menangani berbagai skenario akses data. Sistem harus membedakan antara pengguna dengan SKPD tertentu yang hanya dapat mengakses data dari unit kerjanya dan pengguna tanpa SKPD yang memiliki akses ke semua data. Hal ini memerlukan logika penyaringan yang kompleks di setiap kueri basis data dan berpotensi menyebabkan inkonsistensi data jika tidak diimplementasikan dengan benar pada semua titik akses.

Solusinya adalah pengembangan perangkat lunak tengah *Laravel* khusus yang menangani penyaringan SKPD secara otomatis dan konsisten di seluruh aplikasi. Perangkat lunak tengah ini menganalisis peran dan SKPD pengguna yang masuk, kemudian secara otomatis menambahkan batasan ke kueri basis data. Implementasi cakupan global pada model *Eloquent* yang secara otomatis memfilter data berdasarkan SKPD pengguna, memastikan konsistensi penyaringan di semua operasi *CRUD* tanpa perlu menambahkan logika penyaringan manual di setiap pengontrol.

### 3. Responsivitas Antarmuka pada Berbagai Ukuran Layar Ponsel

Pengembangan antarmuka depan aplikasi ponsel menghadapi tantangan dalam memastikan tampilan yang optimal di berbagai ukuran layar dan resolusi perangkat *Android* dan *iOS*. Komponen antarmuka seperti korsel, kartu, dan tata letak tombol tidak konsisten ketika dijalankan di perangkat dengan kepadatan layar dan rasio aspek yang berbeda. Masalah ini terutama terlihat pada perangkat dengan layar sangat kecil atau sangat besar, dimana elemen antarmuka menjadi tidak proporsional atau sulit diakses.

Solusinya adalah implementasi sistem desain responsif menggunakan *MediaQuery* dan *LayoutBuilder Flutter* untuk pengaturan ukuran dinamis. Penggunaan dimensi berbasis persentase dan *widget* fleksibel seperti *Expanded* dan *Flexible* untuk memastikan adaptabilitas tata letak. Pengembangan kelas pembantu responsif khusus yang menyediakan titik putus standar dan fungsi utilitas untuk menghitung ukuran elemen secara proporsional. Pengujian ekstensif pada berbagai ukuran layar menggunakan emulator perangkat dan perangkat pengujian fisik untuk memastikan konsistensi antarmuka.

### 4. Penanganan Berkas PDF Berukuran Besar pada Sistem *E-Signature*

Sistem *E-Signature* menghadapi masalah kinerja saat menangani berkas *PDF* berukuran besar (lebih dari 10MB) yang memerlukan waktu lama untuk diproses oleh mesin tampil *PDF.js*. Proses unggah, tampilan, dan manipulasi *kanvas* menjadi lambat dan dapat menyebabkan peramban berhenti atau mogok. Hal ini terutama terjadi pada dokumen dengan banyak halaman atau resolusi tinggi yang memerlukan sumber daya komputasi yang besar untuk penampilan.

Solusinya adalah implementasi pemuatan malas dan penampilan progresif

untuk halaman *PDF* yang hanya menampilkan halaman yang sedang dilihat pengguna. Penggunaan *pekerja web* untuk memproses *PDF* di *utas* latar belakang sehingga tidak memblokir *utas* antarmuka utama. Implementasi teknik kompresi dan optimisasi untuk mengurangi ukuran berkas sebelum pemrosesan, serta penambahan indikator kemajuan yang informatif untuk memberikan umpan balik kepada pengguna selama proses berlangsung. Batasan ukuran berkas maksimum dan validasi di antarmuka depan untuk mencegah unggah berkas yang terlalu besar.

