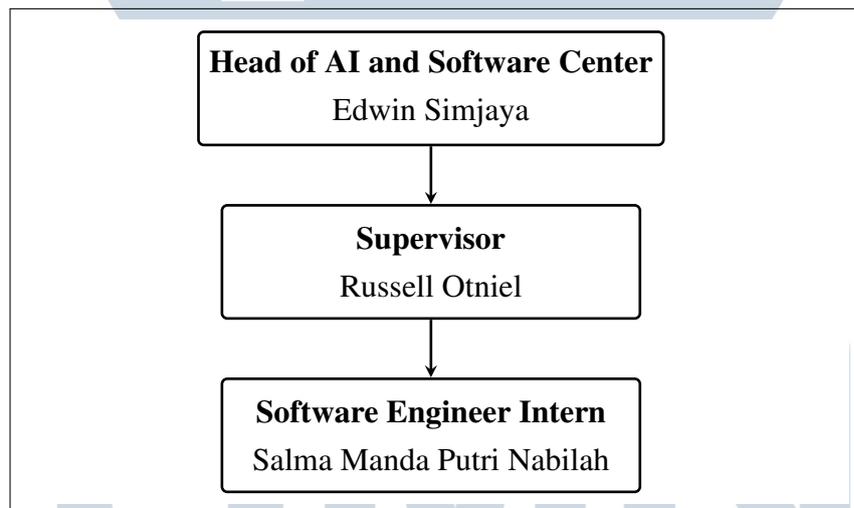


BAB 3 PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama masa pelaksanaan program magang di PT Kalbe Farma Tbk, peran yang dijalankan berada pada posisi Software Engineer Intern di bawah unit Software Center, bagian dari Departemen Corporate Digital Technology (CDT). Departemen CDT bertanggung jawab dalam pengelolaan dan pengembangan teknologi informasi perusahaan, terutama dalam pengembangan perangkat lunak. Seluruh kegiatan magang didampingi oleh seorang mentor yang menjabat sebagai Software Engineer Officer. Struktur kedudukan dan koordinasi selama program magang dapat dilihat pada Gambar 3.1.



Gambar 3.1. Struktur Kedudukan dan Koordinasi Magang

Fokus utama kegiatan magang adalah pembuatan *Proof of Concept* (PoC) untuk fitur-fitur pada sistem Regulatory Information Management (RIM). Sistem ini dirancang untuk mendukung pengelolaan dokumen regulasi di lingkungan PT Kalbe Farma Tbk. Sebelum penyusunan PoC, dilakukan kegiatan eksplorasi terhadap kebutuhan fungsional dan teknis. Eksplorasi dilakukan melalui *user requirements gathering*, diskusi lintas tim, dan pemetaan kebutuhan dari tim Regulatory Affairs (RA).

PoC disusun sebagai bentuk validasi awal terhadap pendekatan teknis dan kelayakan fitur. Proyek ini melibatkan kerja sama lintas tim yang terkoordinasi

secara aktif. Tim Software Engineer bertugas merancang dan mengembangkan purwarupa fitur sistem. Tim UI/UX menyediakan rancangan antarmuka pengguna sebagai acuan pengembangan visual. Tim Project Management memfasilitasi alur kerja dan komunikasi antar tim selama proses pengembangan. Rancangan simulasi proses dan dokumentasi kebutuhan juga dikelola oleh tim ini. Seluruh koordinasi dilakukan untuk memastikan fitur memenuhi kebutuhan pengguna dan regulator. Dengan demikian, kolaborasi antar tim berjalan efektif dan terarah.

Komunikasi dilakukan secara daring menggunakan WhatsApp, Microsoft Teams, dan Discord sebagai media utama. Dokumentasi proyek dan pelacakan perkembangan dikelola melalui GitHub, SharePoint, dan Notion. Rapat mingguan dilakukan secara rutin untuk mengevaluasi progres proyek dan mendiskusikan kendala teknis. Rencana tindak lanjut juga dirumuskan pada setiap sesi tersebut. Dokumentasi teknis disusun sebagai bagian dari laporan dan bahan evaluasi pekerjaan selama magang.

Selain berkontribusi dalam pengembangan PoC, dukungan juga diberikan kepada tim Project Management. Dukungan tersebut diberikan dalam bentuk perancangan alur kerja fitur berbasis kebutuhan pengguna. Upaya ini dilakukan agar rancangan solusi tetap realistis dan dapat diimplementasikan secara efektif. Peran ini turut memastikan kesesuaian antara kebutuhan bisnis dan solusi teknis yang dikembangkan.

3.2 Tugas yang Dilakukan

Selama pelaksanaan magang di PT Kalbe Farma Tbk sebagai Software Engineer Intern, tanggung jawab utama adalah mendukung proses eksplorasi dan perancangan *Proof of Concept* (PoC) untuk fitur pada sistem Regulatory Information Management (RIM). Sistem ini dirancang untuk membantu tim Regulatory Affairs (RA) dalam pengelolaan dokumen regulasi secara lebih efisien dan terstruktur. Salah satu fitur utama yang dirancang adalah pengelompokan dokumen ke dalam struktur bernama *dossier*, yang digunakan dalam proses pengajuan izin produk. Selain itu, sistem juga ditujukan untuk memfasilitasi kolaborasi lintas tim selama proses pengembangan produk.

Kegiatan magang difokuskan pada perancangan purwarupa aplikasi web yang berisi fitur-fitur penting sesuai kebutuhan pengguna. Sistem dibangun menggunakan Next.js sebagai kerangka kerja *frontend* dan Supabase sebagai basis layanan *backend* serta penyimpanan data. Proses pengembangan dilakukan

berdasarkan hasil eksplorasi terhadap kebutuhan fungsional dan teknis yang diperoleh melalui *requirement gathering*. Diskusi rutin dilakukan bersama tim Project Management dan tim Regulatory Affairs. PoC dimanfaatkan sebagai validasi awal untuk menilai kelayakan teknis sebelum sistem dikembangkan lebih lanjut.

Beberapa fitur utama yang dikembangkan meliputi unggah dokumen ke layanan penyimpanan *cloud*, pembuatan *template dossier*, pembuatan *dossier* berdasarkan *template*, serta fitur pencarian. Pengujian fungsional dan proses *debugging* dilakukan secara berkala guna memastikan stabilitas sistem. Kegiatan pengembangan dijalankan bersama tim UI/UX dan tim Project Management untuk memastikan kesesuaian antarmuka dan proses.

Secara keseluruhan, tanggung jawab selama magang mencakup seluruh fase awal dalam siklus pengembangan perangkat lunak, mulai dari analisis kebutuhan, perencanaan, perancangan, hingga pengujian awal. Pengalaman ini memberikan wawasan mendalam terkait praktik *software engineering* dalam industri yang sangat memperhatikan aspek regulasi dan kepatuhan. Selain pengembangan teknis, kemampuan komunikasi, kerja tim, dan penyelesaian masalah turut dikembangkan. Tabel 3.1 merangkum linimasa pekerjaan mingguan selama masa magang.

Tabel 3.1. Pekerjaan Mingguan Selama Magang

Minggu ke-	Pekerjaan yang Dilakukan
1	<i>Onboarding</i> perusahaan, pengenalan struktur organisasi, pengenalan proyek masa depan, proyek terdahulu, serta mempelajari TypeScript dasar.
2	Belajar TypeScript menggunakan Next.js dengan membuat proyek pencarian lokasi <i>campsite</i> menggunakan API.
3	Mempelajari penggunaan Supabase untuk manajemen basis data dan integrasinya dengan proyek Next.js.
4	Menerapkan TypeScript, Next.js, dan Supabase pada mini proyek <i>to-do list</i> dengan fitur kontrol akses berbasis peran.
5	Pengenalan awal terhadap proyek RIM dan eksplorasi dokumentasi fungsionalnya.

Minggu ke-	Pekerjaan yang Dilakukan
6	Melanjutkan eksplorasi proyek dan melakukan uji coba <i>cloud S</i> sebagai bagian dari arsitektur sistem.
7	Eksplorasi <i>cloud S</i> menggunakan Postman, mengikuti rapat mingguan, dan melakukan <i>requirement gathering</i> .
8	Penetapan <i>cloud S</i> sebagai penyimpanan dokumen proyek, serta melanjutkan <i>requirement gathering</i> dan rapat mingguan.
9	Proyek <i>handoff</i> dari <i>intern</i> sebelumnya, penyambungan sistem ke <i>cloud</i> penyimpanan, serta koordinasi proyek.
10	Pengembangan fitur unggah dokumen dan penggabungan beberapa dokumen PDF, serta diskusi kebutuhan fitur.
11	Penetapan Supabase sebagai sistem basis data untuk pemetaan dokumen dan integrasinya dalam proyek RIM.
12	Pengembangan fitur pembuatan <i>dossier</i> dan <i>template dossier</i> untuk pengelolaan dokumen regulasi.
13	Diskusi dengan <i>intern</i> sebelumnya terkait proyek <i>handoff</i> serta pengembangan fitur <i>edit dossier</i> .
14	Melanjutkan pengembangan dan penyempurnaan fitur <i>edit dossier</i> .
15	Pengembangan fitur pencarian dan penampil dokumen (<i>PDF viewer</i>) dalam sistem.
16	Migrasi penyimpanan dari <i>cloud S</i> ke <i>cloud SS</i> , serta pengembangan fitur pencarian dan <i>PDF viewer</i> .
17	Pengembangan fitur pencarian dokumen dan penerapan fitur terjemahan halaman (<i>page translate</i>).

3.3 Uraian Pelaksanaan Magang

Pelaksanaan kegiatan magang dilakukan dengan menyelesaikan berbagai tugas yang diberikan oleh mentor dan tim pengembang. Selama kurang lebih empat bulan, fokus utama adalah keterlibatan langsung dalam eksplorasi awal pengembangan sistem Regulatory Information Management (RIM). Kegiatan ini

mencakup proses identifikasi kebutuhan teknis serta perancangan solusi awal sistem. Peran yang dijalankan lebih ditekankan pada tahap konseptual dan validasi fitur melalui pembuatan *Proof of Concept* (PoC), yang digunakan sebagai dasar penilaian kelayakan teknis pengembangan sistem.

Beberapa tugas penting yang dilakukan antara lain mengumpulkan kebutuhan pengguna (*user requirement*), merancang alur fungsional, serta menguji kelayakan (*feasibility*) dari fitur-fitur melalui PoC. Validasi dilakukan untuk menilai apakah pendekatan teknis yang digunakan sudah sesuai dengan kebutuhan pengguna dan standar sistem. PoC juga berfungsi sebagai bahan komunikasi visual kepada pengguna untuk memperoleh umpan balik sebelum tahap pengembangan lanjutan. Seluruh langkah ini bertujuan untuk memastikan bahwa pengembangan sistem dilakukan secara terstruktur dan tepat sasaran sejak tahap awal perancangan.

3.3.1 Fitur Unggah Dokumen

A. Tujuan dan Kesesuaian Fitur

Fitur ini dirancang untuk mendukung proses pengunggahan dokumen sebagai bagian dari pengajuan regulasi. Dokumen yang telah diunggah akan digunakan dalam proses penyusunan dan evaluasi dokumen regulatori oleh tim Regulatory Affairs. Fitur ini menjawab kebutuhan pengguna dalam hal *Submission Management* dan *Document Storage*. Selain itu, fitur ini juga mendukung pengelolaan berkas secara terpusat, terdokumentasi, dan aman.

B. Proses Alur Fitur

Proses dimulai saat pengguna memilih dokumen dari perangkat lokal melalui antarmuka sistem. Sistem akan melakukan verifikasi terhadap ukuran dan jenis dokumen untuk memastikan kesesuaian dengan ketentuan yang berlaku. Jika dokumen valid, maka dokumen dikonversi ke format base64 untuk memudahkan pengiriman data ke penyimpanan. Dokumen yang telah dikonversi kemudian dikirimkan ke media penyimpanan eksternal melalui integrasi API, baik ke *cloud S* maupun *cloud SS*.

Umpan balik diberikan kepada pengguna berdasarkan status hasil unggahan yang dilakukan. Notifikasi visual akan muncul untuk menandakan apakah proses unggah berhasil atau gagal. Pendekatan ini memberikan pengalaman pengguna yang interaktif dan membantu dalam pemantauan status dokumen. Selain itu,

proses ini juga memastikan bahwa hanya dokumen valid yang diterima oleh sistem penyimpanan.

C. Penjelasan Teknis dan Flowchart

Pada tahap awal pengembangan fitur unggah dokumen, dilakukan eksplorasi dua pendekatan penyimpanan: *cloud S* dan *cloud SS*. *Cloud S* dipilih karena telah terintegrasi dengan sistem internal perusahaan dan mendukung keamanan data tingkat tinggi. *Cloud SS* digunakan untuk mengevaluasi fleksibilitas teknis serta kemudahan integrasi dengan teknologi modern seperti Next.js dan TypeScript. Kedua pendekatan diuji dan divalidasi melalui pembuatan PoC untuk menilai kelayakan penerapan fitur dalam konteks sistem akhir.

Fitur unggah dokumen memungkinkan pengguna memilih dokumen dari perangkat lokal. Sistem akan melakukan validasi terhadap ukuran dan format dokumen, kemudian mengonversi dokumen menjadi *base64*. Setelah itu, dokumen dikirim ke *cloud* penyimpanan menggunakan metode API yang relevan. Pengguna akan menerima notifikasi visual mengenai hasil proses unggah, baik sukses maupun gagal.

1. Proses Unggah Dokumen ke Cloud S

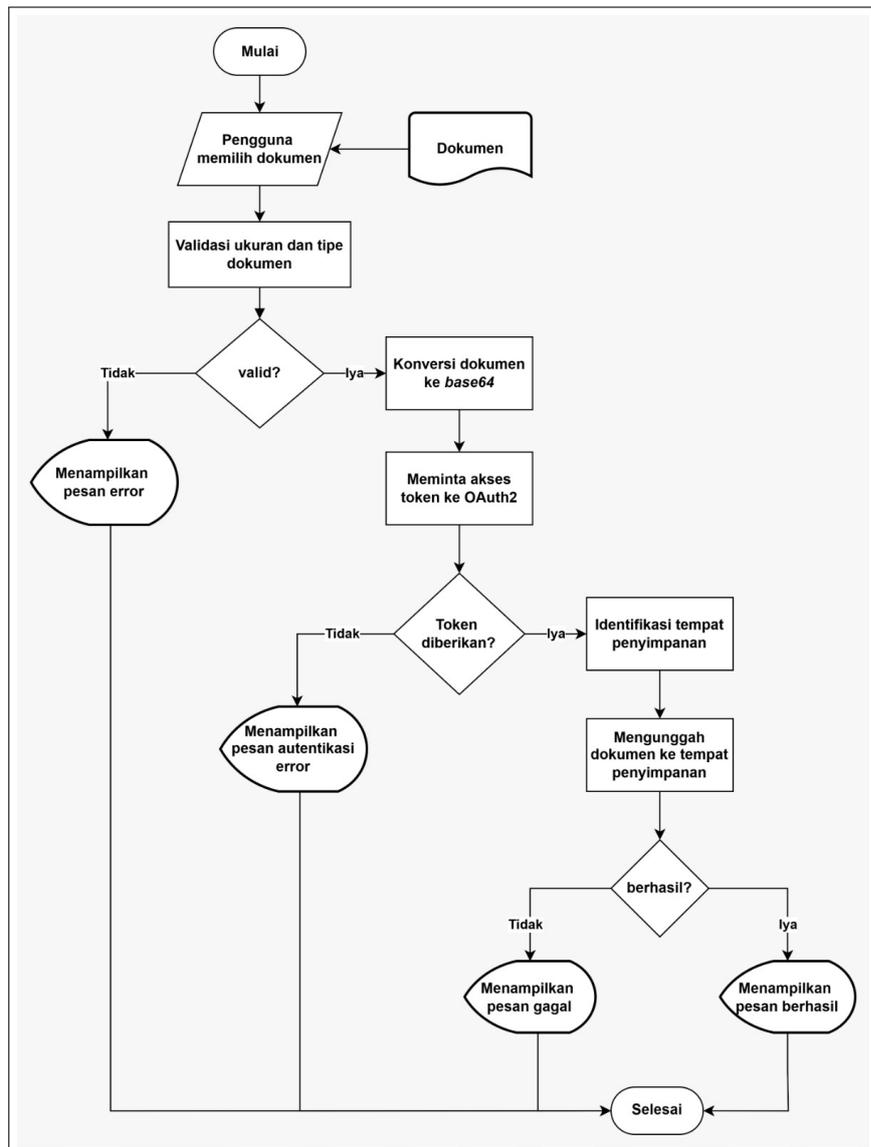
Proses unggah dokumen ke *cloud S* dilakukan melalui integrasi dengan layanan API milik pihak ketiga, yang memerlukan otentikasi menggunakan protokol OAuth 2.0. Proses dimulai ketika pengguna memilih dokumen dari perangkat lokal, kemudian sistem melakukan validasi terhadap ukuran dan jenis dokumen. Jika dokumen dinyatakan valid, dokumen tersebut akan dikonversi ke dalam format *base64* agar dapat dikirim dengan aman melalui jaringan. Tahapan ini penting untuk memastikan bahwa dokumen memenuhi standar sistem sebelum dikirimkan ke server penyimpanan eksternal.

Setelah proses konversi selesai, sistem menginisiasi otentikasi dengan mengirim permintaan token akses ke server otorisasi dari penyedia layanan pihak ketiga. Permintaan dilakukan melalui *endpoint* protokol OAuth dengan metode *POST*, menggunakan parameter autentikasi yang telah dikonfigurasi dalam sistem. Token akses yang diperoleh selanjutnya digunakan untuk mengakses *endpoint* layanan API dari *cloud* penyimpanan eksternal. Tahapan ini bertujuan untuk memastikan bahwa proses unggah hanya dapat dilakukan oleh pihak yang memiliki otorisasi resmi.

Setelah berhasil terotentikasi, sistem mengidentifikasi lokasi penyimpanan berdasarkan struktur direktori internal pada *cloud* penyimpanan eksternal. Setelah direktori target ditemukan, dokumen yang telah dikonversi diunggah menggunakan metode PUT. Jika seluruh proses berjalan dengan sukses, sistem akan menampilkan notifikasi keberhasilan kepada pengguna.

Apabila terjadi kegagalan pada salah satu tahapan, sistem akan memberikan notifikasi berisi informasi kesalahan yang spesifik. Hal ini bertujuan untuk membantu pengguna dalam mengenali titik kegagalan dan mengambil langkah korektif yang sesuai. Rangkaian proses unggah ini divisualisasikan dalam Gambar 3.2, yang menggambarkan alur dari pemilihan dokumen hingga proses unggah selesai.





Gambar 3.2. Flowchart Proses Unggah Dokumen ke Cloud S

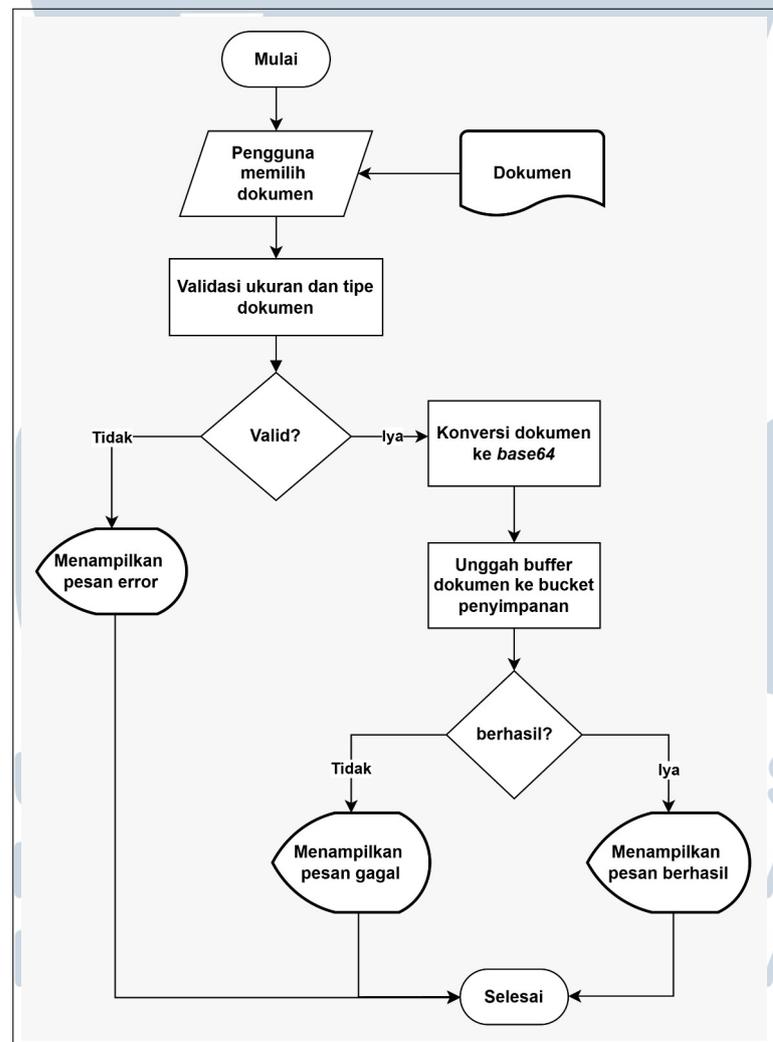
2. Proses Unggah Dokumen ke Cloud SS

Proses ini dikembangkan sebagai alternatif untuk mengeksplorasi fleksibilitas *cloud* SS sebagai media penyimpanan dokumen regulasi. Pengguna memulai dengan memilih dokumen dari perangkat lokal, yang kemudian diverifikasi ukurannya dan jenis formatnya. Jika dokumen tidak sesuai dengan kriteria yang ditentukan, sistem akan langsung menampilkan notifikasi kesalahan. Proses unggah akan dihentikan untuk mencegah masuknya dokumen yang tidak sah ke dalam sistem.

Jika dokumen dinyatakan valid, sistem akan mengonversinya ke dalam format

base64, kemudian membentuk `buffer` dan melakukan inialisasi koneksi ke `cloud SS`. Setelah koneksi terbentuk, dokumen diunggah ke dalam `bucket` penyimpanan yang telah ditentukan melalui pemanggilan API. Setelah proses unggah selesai, sistem memverifikasi status unggahan dan memberikan umpan balik kepada pengguna.

Apabila proses unggahan berhasil, sistem akan menampilkan notifikasi bahwa dokumen telah tersimpan di `cloud SS`. Sebaliknya, jika terjadi kegagalan teknis, sistem akan menampilkan pesan kesalahan yang sesuai. Pendekatan ini dirancang untuk memberikan pengalaman unggah yang aman, efisien, dan dapat diandalkan. Alur lengkap proses ini disajikan dalam Gambar 3.3 untuk mempermudah pemahaman secara visual.



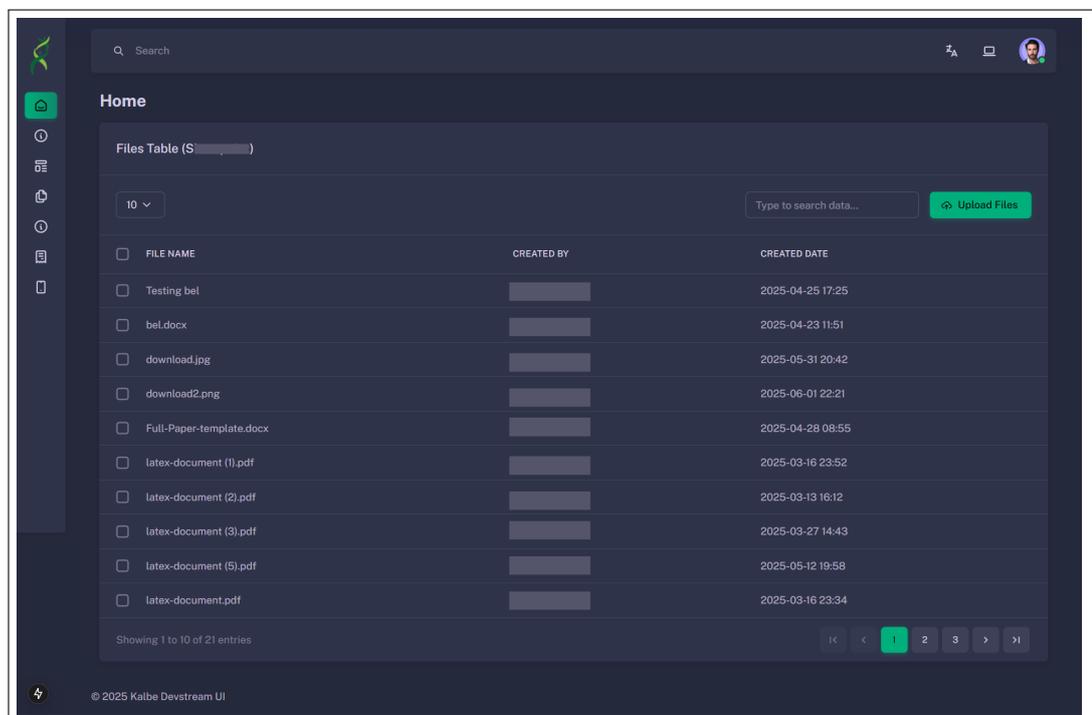
Gambar 3.3. Flowchart Proses Unggah Dokumen ke Cloud SS

D. Hasil dan Evaluasi PoC

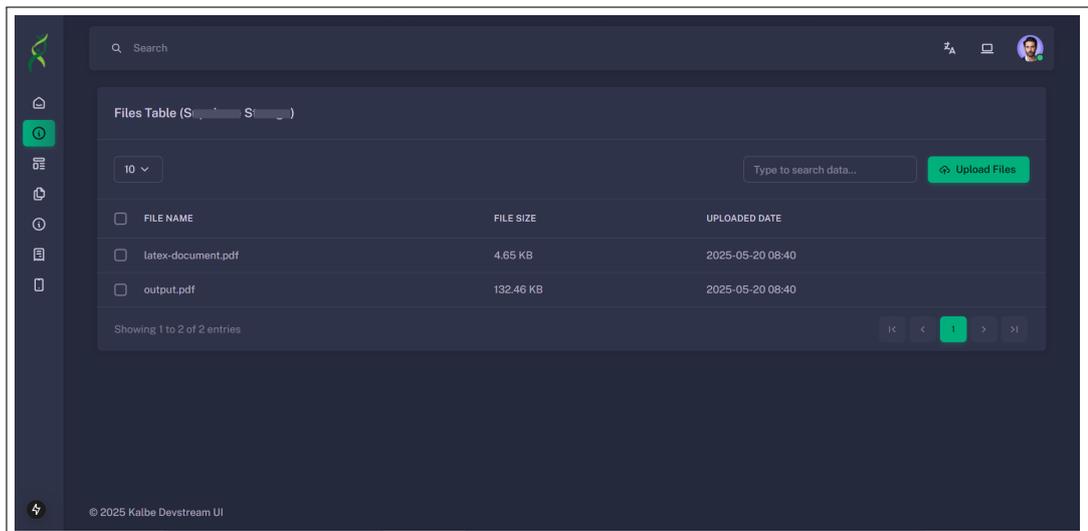
Hasil dari implementasi PoC menunjukkan bahwa kedua pendekatan penyimpanan, *cloud S* dan *cloud SS* berhasil digunakan untuk mengunggah dan menampilkan kembali dokumen. *Cloud S* dinilai sesuai untuk integrasi dengan sistem internal yang telah ada sebelumnya, khususnya dalam ekosistem perusahaan. Di sisi lain, *Cloud SS* memberikan keunggulan dalam hal dokumentasi yang terbuka, kemudahan penggunaan, dan fleksibilitas pengembangan sistem.

Evaluasi menunjukkan bahwa masing-masing pendekatan memiliki kelebihan yang dapat dioptimalkan sesuai dengan konteks kebutuhan organisasi. *Cloud S* dapat digunakan sebagai solusi utama apabila diperlukan keterhubungan erat dengan infrastruktur digital perusahaan yang telah berjalan. Sebaliknya, *cloud SS* lebih ideal untuk pengembangan sistem modern berbasis *web framework* yang mendukung integrasi API secara cepat dan efisien.

PoC ini memberikan dasar teknis yang kuat dalam pengambilan keputusan arsitektur penyimpanan dokumen pada tahap pengembangan lanjutan. Ilustrasi hasil unggahan dokumen dari masing-masing *cloud* ditunjukkan pada Gambar 3.4 dan Gambar 3.5.



Gambar 3.4. Tampilan Halaman Tabel Dokumen dari Cloud S



Gambar 3.5. Tampilan Halaman Tabel Dokumen dari Cloud SS

3.3.2 Fitur Template Dossier

A. Tujuan dan Kesesuaian Fitur

Fitur ini memfasilitasi pembuatan struktur pengelompokan dokumen regulasi dalam bentuk *dossier* yang terdiri dari beberapa *section* dan *subsection*. Tujuan utamanya adalah untuk menyusun struktur pengajuan dokumen regulasi secara modular dan fleksibel. Dengan pendekatan ini, pengguna dapat menyesuaikan susunan dokumen sesuai dengan kebutuhan pengajuan masing-masing negara atau lembaga regulator. Fitur ini juga memberikan konsistensi dalam dokumentasi dan mempermudah proses pelacakan serta pengelolaan dokumen regulasi.

B. Proses Alur Fitur

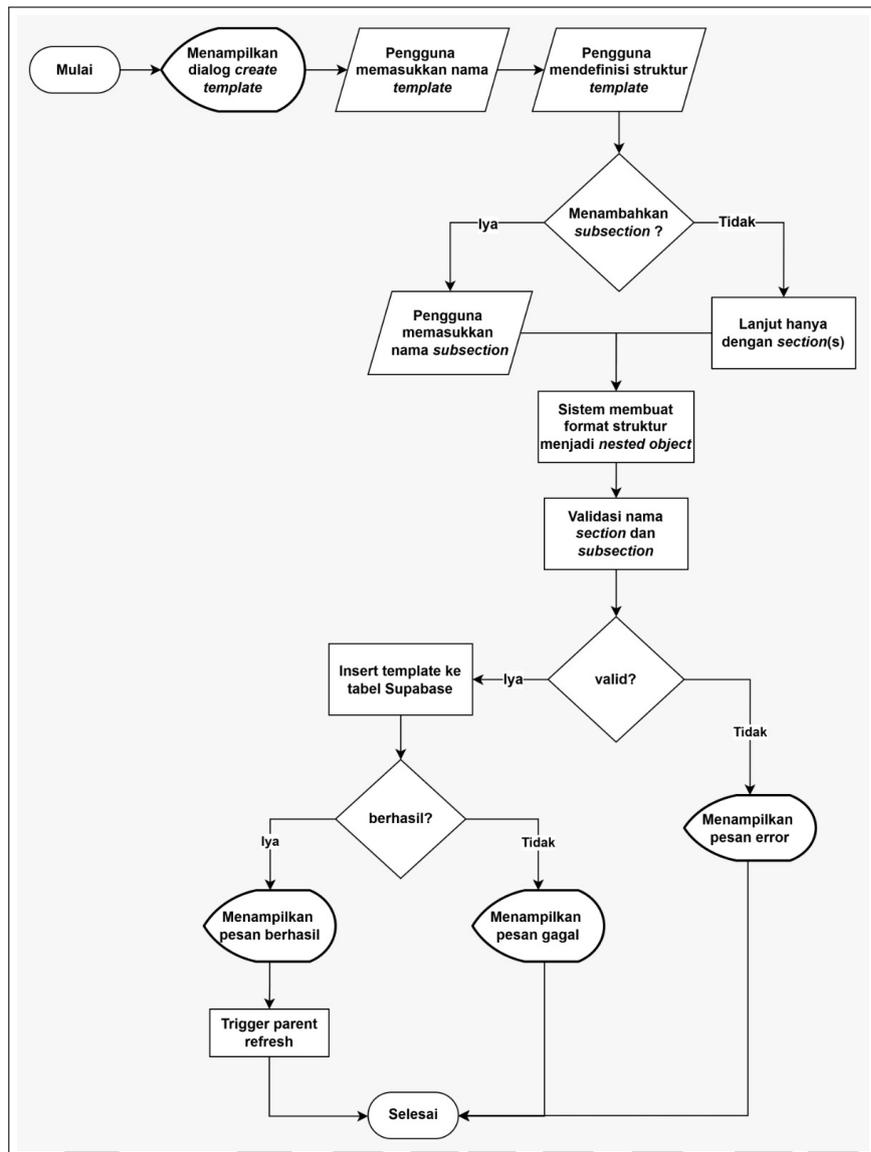
Proses dimulai saat pengguna menekan tombol *Create Template* pada antarmuka sistem. Setelah itu, pengguna mengisi nama *template* dan mulai mendefinisikan struktur *dossier* dengan menambahkan *section* serta *subsection*. Sistem akan memverifikasi apakah struktur yang dibentuk telah sesuai dan tidak kosong. Jika validasi berhasil, informasi struktur tersebut akan disimpan secara otomatis ke dalam basis data.

C. Penjelasan Teknis dan Flowchart

Struktur yang dibuat disimpan dalam format JSON dengan skema hierarkis, yaitu: *section* → *subsection* → *documents*. Data yang disimpan mencakup nama *template*, struktur *dossier* yang telah dibuat, metadata pembuat, serta waktu pembuatan. Proses pengiriman data ke basis data dilakukan menggunakan layanan Supabase yang telah terintegrasi dengan sistem. Jika proses penyimpanan berhasil, sistem akan menampilkan notifikasi keberhasilan kepada pengguna dan menutup jendela dialog isian *template*.

Sebaliknya, jika terjadi kesalahan selama proses penyimpanan data, sistem akan memberikan notifikasi berupa pesan kesalahan kepada pengguna. Hal ini membantu pengguna mengetahui penyebab kegagalan dan memperbaikinya. Dengan adanya validasi otomatis dan umpan balik visual, proses pembuatan *template* menjadi lebih andal dan efisien. Alur lengkap proses ini ditampilkan pada Gambar 3.6.



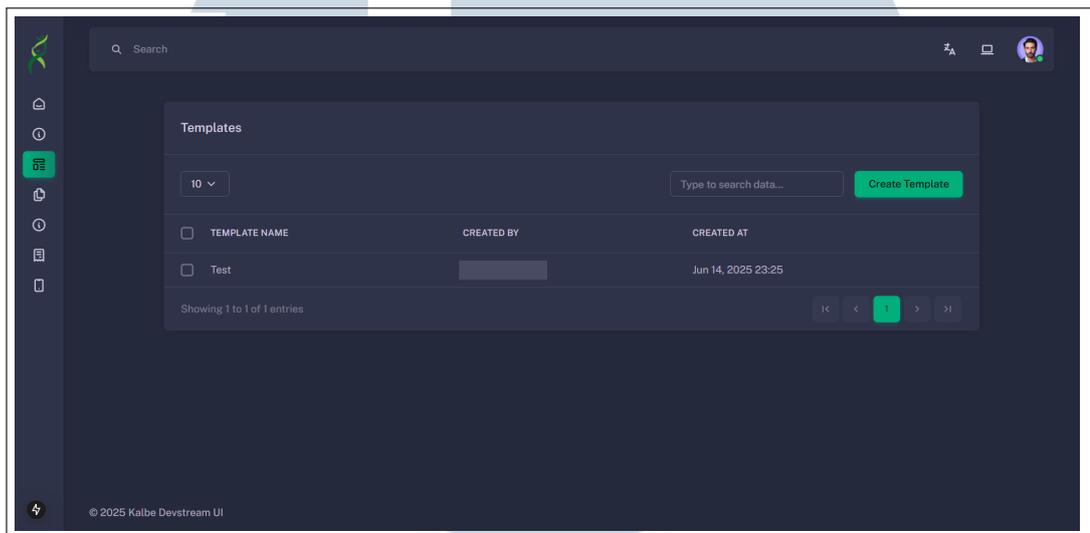


Gambar 3.6. Flowchart Proses Pembuatan Template Dossier

D. Hasil dan Evaluasi PoC

PoC pada fitur Template Dossier berhasil memvalidasi pendekatan penyusunan struktur dokumen regulasi secara dinamis dan fleksibel. Pengguna dapat menambahkan *section* dan *subsection* sesuai kebutuhan, kemudian menyimpannya dalam basis data dengan format JSON. Hasil evaluasi menunjukkan bahwa struktur yang dibuat bersifat *reusable* dan dapat digunakan kembali dalam pengelompokan dokumen untuk berbagai pengajuan. Hal ini memberikan efisiensi waktu dan konsistensi dalam proses dokumentasi regulasi.

Proses penyimpanan ke dalam basis data berhasil dilakukan melalui integrasi Supabase tanpa ditemukan kendala teknis yang berarti. Antarmuka pengguna yang interaktif juga turut mendukung pengalaman penggunaan yang intuitif dalam membentuk *template*. Fitur ini dinilai layak untuk dikembangkan lebih lanjut sebagai bagian inti dari sistem Regulatory Information Management. Tampilan hasil penyimpanan dan daftar *template* dapat dilihat pada Gambar 3.7.



Gambar 3.7. Tampilan Halaman Template Dossier

3.3.3 Fitur Pembuatan Dossier

A. Tujuan dan Kesesuaian Fitur

Fitur ini dirancang untuk memfasilitasi proses pengelompokan dokumen regulasi ke dalam satu kesatuan bernama *dossier*. Setiap *dossier* dibentuk berdasarkan struktur *template dossier* yang telah disusun sebelumnya oleh pengguna. Tujuan utama dari fitur ini adalah untuk menyatukan dokumen yang relevan dalam satu wadah yang terstruktur dan sesuai dengan standar regulator. Kesesuaian fitur ini berada pada aspek *Submission Management* yang dibutuhkan oleh tim Regulatory Affairs untuk efisiensi dan konsistensi dokumentasi.

B. Proses Alur Fitur

Proses dimulai ketika pengguna menekan tombol *Create Dossier* pada antarmuka aplikasi. Sistem akan mengambil data *template* yang telah tersedia dan

memuat daftar dokumen yang dapat digunakan. Selanjutnya, pengguna diminta mengisi nama *dossier* dan memilih *template* struktur yang ingin digunakan. Setelah *template* dipilih, sistem akan menampilkan struktur lengkap yang terdiri dari *section* dan *subsection*.

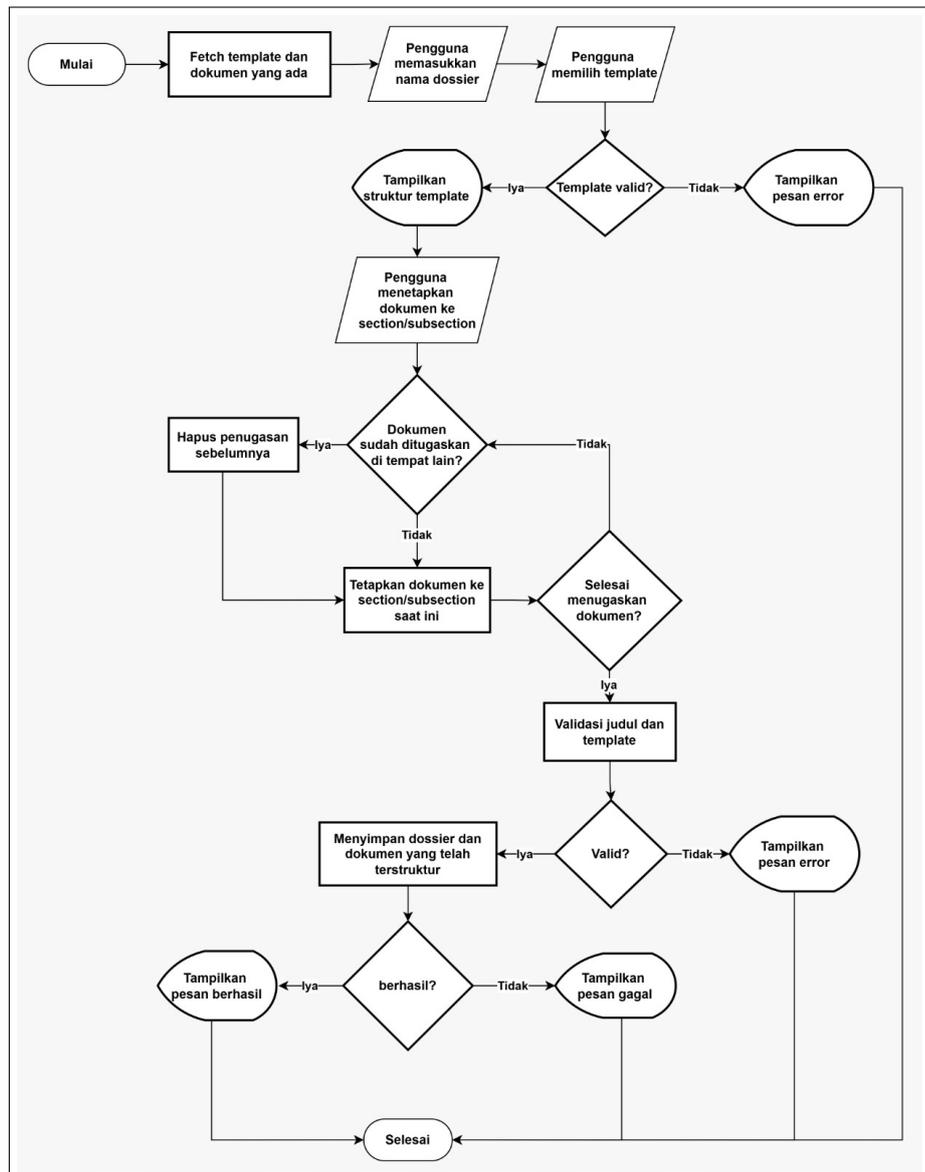
Setiap bagian dari struktur tersebut memungkinkan pengguna menetapkan dokumen yang sesuai dengan fungsinya. Sistem akan melakukan pengecekan otomatis apabila dokumen yang dipilih telah digunakan pada bagian lain dalam *template*. Jika ditemukan duplikasi, sistem akan membatalkan penugasan sebelumnya dan menggantinya dengan penugasan baru yang dipilih. Setelah semua dokumen ditetapkan, sistem akan melakukan validasi untuk memastikan nama *dossier* dan *template* telah diisi dengan benar sebelum menyimpan data.

C. Penjelasan Teknis dan Flowchart

Struktur *dossier* disimpan dalam format JSON bertingkat yang mengikuti hierarki dari *template* yang digunakan. Data yang dicatat meliputi nama *dossier*, susunan dokumen per bagian (*section* dan *subsection*), serta referensi langsung terhadap *template* yang dipilih. Informasi ini kemudian disimpan ke dalam tabel *dossiers* pada basis data Supabase. Dengan demikian, struktur dapat digunakan kembali dan dikelola dalam sistem secara efisien.

Selain itu, sistem menyediakan fitur pencarian dokumen yang berguna untuk membantu pengguna menemukan dokumen yang sesuai dalam proses penyusunan *dossier*. Validasi akhir dilakukan sebelum data dikirim untuk memastikan kelengkapan input dan kesesuaian struktur. Jika proses validasi gagal, sistem akan memberikan pesan kesalahan kepada pengguna. Namun, apabila berhasil, sistem akan menampilkan notifikasi sukses dan menyimpan data ke basis data seperti yang ditunjukkan pada Gambar 3.8.

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

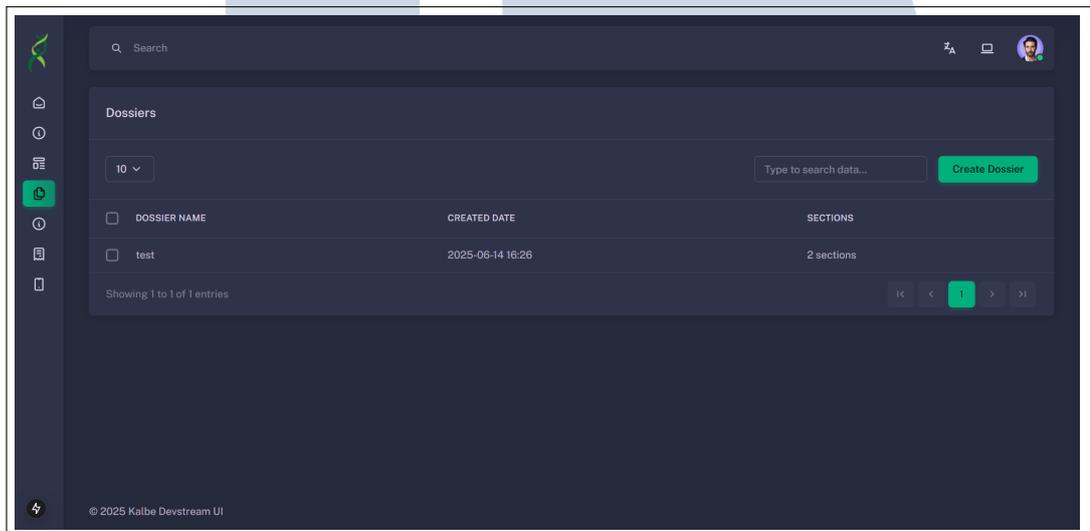


Gambar 3.8. Flowchart Proses Pembuatan Dossier

D. Hasil dan Evaluasi PoC

Hasil dari implementasi PoC menunjukkan bahwa pengguna dapat membentuk *dossier* dengan struktur yang fleksibel dan sesuai dengan kebutuhan pengajuan regulasi. Proses penyusunan dimulai dari pemilihan *template*, pengisian nama, penetapan dokumen ke masing-masing bagian, hingga penyimpanan ke dalam basis data tanpa hambatan berarti. Representasi data dalam bentuk JSON terbukti efektif dalam mendeskripsikan hierarki dan relasi antar dokumen. Format ini juga kompatibel dengan berbagai sistem yang mengelola data berbasis struktur.

Antarmuka pengguna yang dirancang interaktif dan disertai validasi input memberikan kemudahan dan kejelasan selama penggunaan fitur. Evaluasi juga menunjukkan bahwa pendekatan modular terhadap pengelompokan dokumen dapat diterapkan secara praktis di lingkungan pengembangan. Fitur ini memiliki potensi besar untuk dikembangkan lebih lanjut sebagai komponen utama dalam sistem Regulatory Information Management. Tampilan halaman daftar *dossier* ditunjukkan pada Gambar 3.9.



Gambar 3.9. Tampilan Halaman Tabel Dossier

3.3.4 Fitur Pencarian Dokumen dan Dossier

A. Tujuan dan Kesesuaian Fitur

Fitur ini dirancang untuk memudahkan pengguna dalam menemukan dokumen atau *dossier* melalui satu kolom pencarian terpadu. Proses pencarian dilakukan berdasarkan kata kunci yang dimasukkan oleh pengguna. Seluruh entitas, baik dokumen maupun *dossier*, dikumpulkan dalam satu sistem pencarian agar proses navigasi menjadi lebih cepat dan efisien. Fitur ini sesuai untuk mendukung efisiensi pengelolaan konten dan meningkatkan pengalaman pengguna.

B. Proses Alur Fitur

Proses pencarian dimulai ketika sistem melakukan *fetch* data dari dua sumber, yaitu daftar *dossier* dan dokumen-dokumen yang tersedia. Kedua sumber

tersebut kemudian digabungkan ke dalam satu daftar opsi pencarian. Saat pengguna mulai mengetik pada kolom pencarian, sistem akan secara otomatis memfilter daftar tersebut berdasarkan kecocokan kata kunci.

Setelah hasil pencarian muncul, pengguna dapat memilih salah satu item dari daftar. Sistem kemudian mendeteksi apakah item yang dipilih merupakan dokumen atau *dossier*. Jika item tersebut adalah dokumen, maka sistem akan mengambil URL dokumen dari Supabase Storage dan mencoba membuka dokumen tersebut di tab baru. Jika URL tidak valid atau dokumen tidak tersedia, sistem akan menampilkan notifikasi kesalahan.

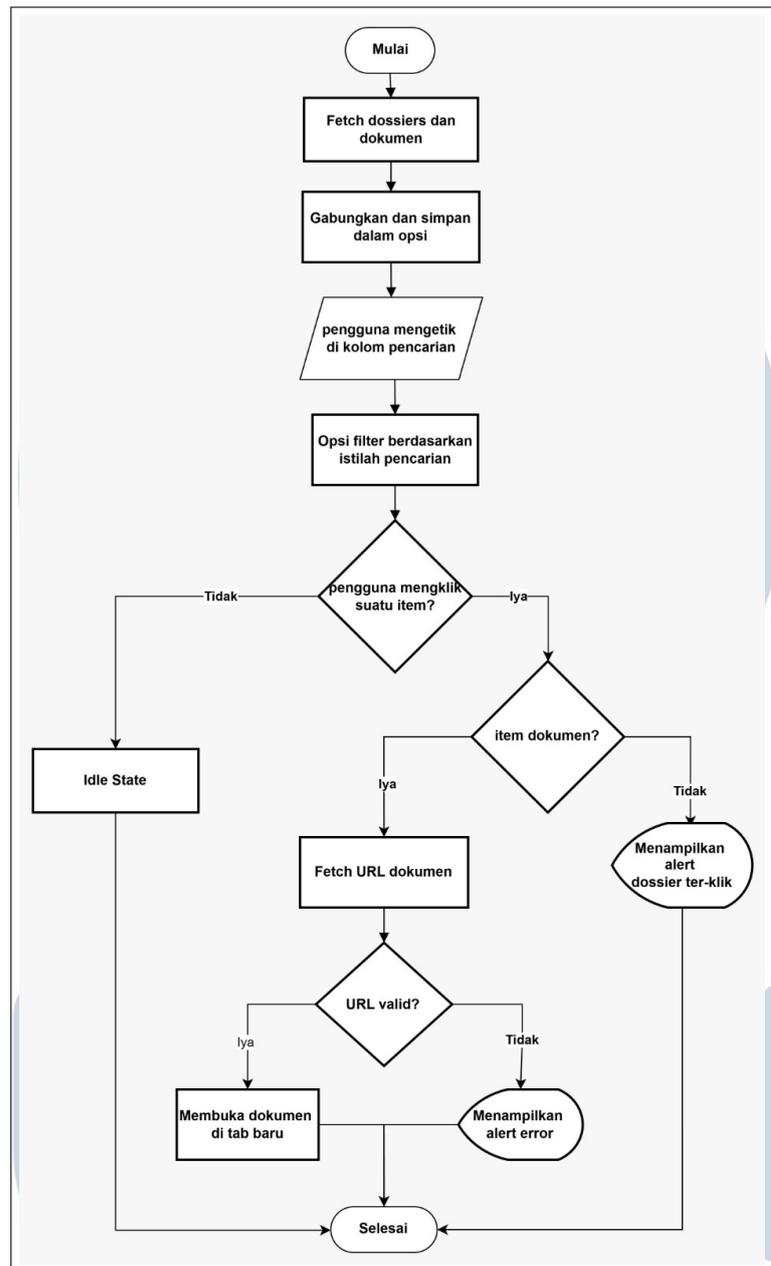
Sebaliknya, apabila item yang dipilih adalah sebuah *dossier*, maka sistem akan menampilkan notifikasi sederhana yang menunjukkan bahwa item telah dipilih. Meskipun fungsionalitas untuk membuka detail *dossier* belum tersedia pada tahap PoC, pendekatan ini membuka ruang pengembangan lebih lanjut. Dengan skema ini, pengguna dapat langsung mengetahui status dan jenis dari item yang dipilih.

C. Penjelasan Teknis dan Flowchart

Fitur ini dikembangkan dengan pendekatan gabungan antara pengambilan data statis dan filter dinamis berbasis input pengguna. Sistem pertama-tama mengambil data dari dua sumber utama, yaitu daftar *dossier* dan daftar dokumen yang tersimpan di Supabase. Data dari kedua sumber kemudian digabungkan menjadi satu struktur `array` opsi yang siap difilter sesuai input pengguna. Proses ini menjamin bahwa semua entitas dapat dicari melalui satu kolom pencarian secara efisien.

Setelah data dimuat, pengguna mengetikkan kata kunci pada kolom pencarian, dan sistem langsung memfilter opsi berdasarkan kecocokan `string`. Hasil pencarian ditampilkan secara terpisah dalam dua kategori: *dossier* dan dokumen. Ketika pengguna mengklik salah satu item, sistem akan memeriksa tipe entitas dari hasil tersebut. Jika dokumen, maka fungsi `getdokumenUrl()` akan digunakan untuk mengambil URL akses dokumen dari *cloud*.

Jika URL berhasil diperoleh dan dokumen tersedia, sistem akan membuka dokumen di *tab* baru browser secara otomatis. Jika tidak berhasil, maka notifikasi kesalahan akan ditampilkan untuk memberitahu pengguna. Untuk item bertipe *dossier*, sistem akan memberikan *alert* sederhana bahwa item telah dipilih. Alur lengkap proses ini divisualisasikan dalam Gambar 3.10.



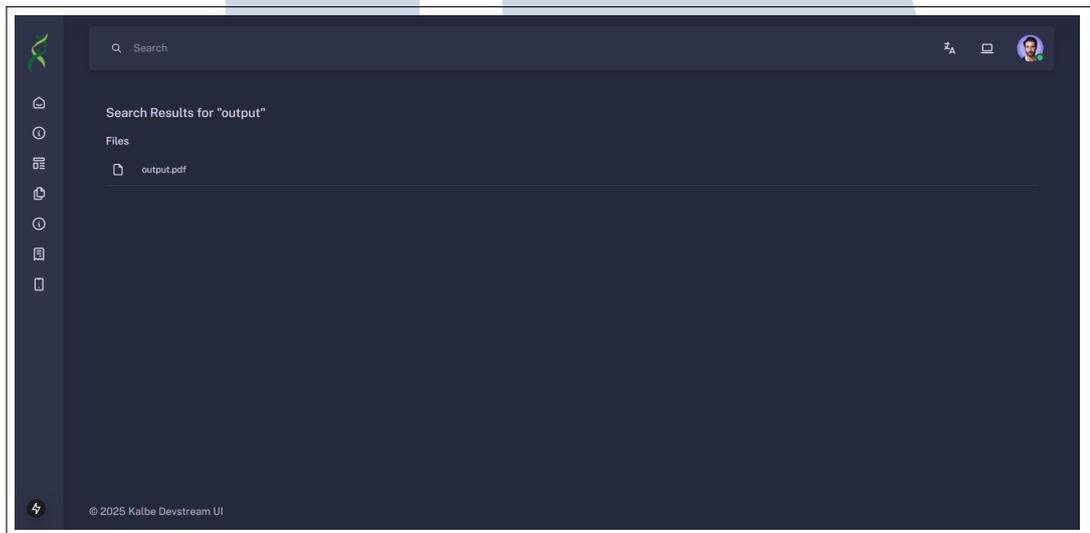
Gambar 3.10. Flowchart Proses Pencarian

D. Hasil dan Evaluasi PoC

PoC membuktikan bahwa pencarian lintas entitas dalam sistem dapat dilakukan dengan baik dan efisien. Proses pemfilteran berlangsung cepat, dan hasil ditampilkan dalam format yang terstruktur serta mudah dipahami pengguna. Untuk dokumen, dokumen dapat langsung dibuka melalui URL yang valid, sedangkan untuk *dossier*, sistem masih terbatas pada notifikasi sederhana. Namun demikian,

pendekatan ini membuka peluang pengembangan lebih lanjut menuju fitur pratinjau atau tampilan detail *dossier*.

Dari sisi teknis, proses integrasi data lintas entitas berjalan tanpa kendala berarti. Antarmuka pengguna yang responsif mendukung pencarian cepat bahkan pada jumlah data yang besar. Evaluasi menunjukkan bahwa fitur ini memberikan nilai tambah dalam hal navigasi dan pengaksesan konten. Tampilan hasil pencarian yang diuji pada PoC ditunjukkan pada Gambar 3.11.



Gambar 3.11. Tampilan Halaman Hasil Pencarian

3.4 Kendala dan Solusi yang Ditemukan

Kendala yang dihadapi selama pelaksanaan magang di PT Kalbe Farma Tbk beserta solusi yang diimplementasikan dirangkum sebagai berikut.

3.4.1 Kendala

1. Proses integrasi dengan *cloud* penyimpanan eksternal mengalami hambatan karena keterbatasan dokumentasi internal, khususnya dalam hal struktur penyimpanan dan parameter autentikasi akses.
2. Tidak semua dokumen berhasil diproses karena terdapat file yang melebihi batas ukuran atau memiliki format yang tidak sesuai, sehingga konversi dan pengunggahan tidak dapat dilanjutkan.

3. Koordinasi antar tim pendukung, seperti tim desain antarmuka dan pengguna sistem utama, terkadang terhambat karena perbedaan fokus dan jadwal, yang menyebabkan diskusi kebutuhan dan penyesuaian fitur menjadi kurang optimal.

3.4.2 Solusi

1. Eksplorasi mandiri dilakukan terhadap struktur layanan penyimpanan dengan bantuan alat bantu eksplorasi API, seperti Postman. Diskusi juga dilakukan bersama mentor dan tim pengembang untuk mengonfirmasi alur integrasi yang sesuai.
2. Validasi tambahan diterapkan untuk memeriksa ukuran dan jenis dokumen sebelum diproses. Informasi kesalahan juga disusun secara lebih jelas agar pengguna memahami penyebab kegagalan.
3. Koordinasi lintas fungsi dijadwalkan ulang secara berkala dan terdokumentasi dengan baik. Setiap sesi pertemuan direkam dan disimpan dalam sistem dokumentasi internal untuk memastikan kesinambungan informasi meskipun tidak semua pihak dapat hadir.

Dengan penanganan yang tepat, kendala-kendala tersebut tidak hanya dapat diatasi secara bertahap, tetapi juga memberikan pembelajaran penting terkait adaptasi sistem, ketelitian dalam pemrosesan data, serta pentingnya komunikasi yang efektif dalam lingkungan kerja profesional.

U M N
UNIVERSITAS
MULTIMEDIA
NUSANTARA