

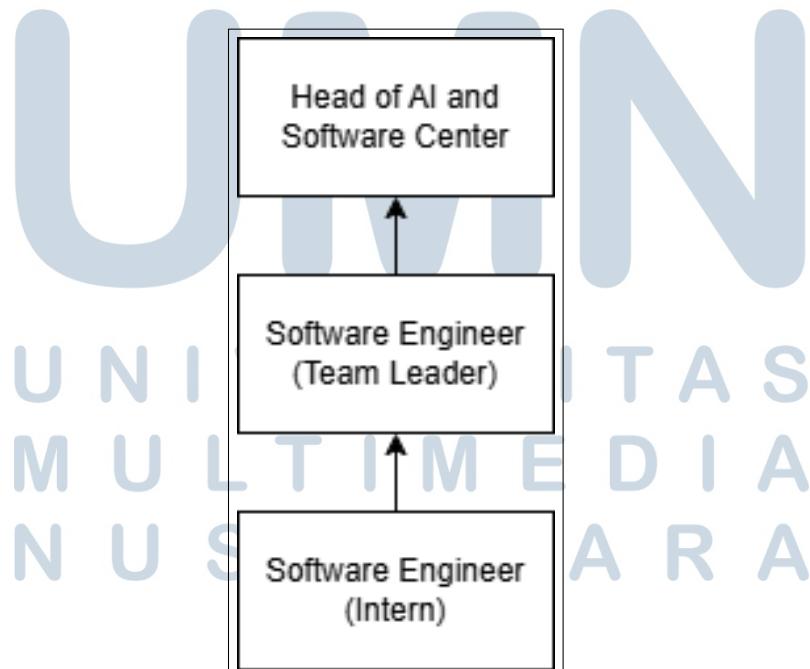
## BAB 3

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Koordinasi

Kegiatan magang dilaksanakan di bawah naungan Divisi *Corporate Digital Technology* PT Kalbe Farma Tbk, dengan kedudukan pada unit kerja yang berfokus pada pengembangan perangkat lunak dan penerapan teknologi digital perusahaan. Posisi ini berada dalam struktur tim pengembangan perangkat lunak (*software engineering*) yang berperan dalam merancang, membangun, dan memelihara berbagai sistem internal perusahaan.

Dalam pelaksanaan tugas, kegiatan kerja berada di bawah arahan seperti pada Gambar 3.1, kepada pemimpin tim (*team leader*) yang merupakan pegawai tetap perusahaan. Setiap tugas dan tanggung jawab diberikan melalui rapat koordinasi mingguan, serta komunikasi harian melalui grup percakapan daring. Mekanisme ini memungkinkan pemantauan kemajuan pekerjaan secara berkelanjutan, identifikasi kendala teknis maupun non-teknis, serta penyusunan rencana kerja untuk minggu berikutnya, sehingga setiap anggota tim memperoleh arahan yang jelas mengenai prioritas kegiatan, ekspektasi kinerja, serta langkah-langkah perbaikan yang perlu dilakukan.



Gambar 3.1. Struktur pelaporan dan pertanggungjawaban

Selain kegiatan koordinasi internal, dilakukan pula pertemuan mingguan bersama manajer proyek (*project manager*) dan pengguna akhir (*end user*) dari berbagai departemen terkait melalui Microsoft Teams. Pertemuan ini berfungsi sebagai forum pengumpulan kebutuhan sistem (*requirement gathering*), di mana proses diskusi dilakukan untuk menyelaraskan kebutuhan bisnis dengan rancangan teknis yang akan dikembangkan. Mekanisme koordinasi lintas departemen ini, menjadi komponen penting antara tim pengembang dan pengguna dalam menghasilkan solusi digital yang sesuai dengan kebutuhan organisasi.

Secara keseluruhan, pola koordinasi yang diterapkan bersifat kolaboratif dan adaptif, dengan menekankan transparansi komunikasi, pembagian tanggung jawab yang jelas, serta orientasi pada hasil kerja yang efektif dan terukur.

### 3.2 Tugas yang Dilakukan

Pelaksanaan kegiatan magang di PT Kalbe Farma Tbk melibatkan penugasan langsung dalam proses pengembangan sistem aplikasi internal perusahaan, dengan fokus pada aspek perancangan basis data, antarmuka pengguna, serta pengelolaan sesi pengguna. Setiap tugas dilaksanakan secara terkoordinasi dengan tim pengembang dan pembimbing teknis untuk memastikan kesesuaian dengan standar teknis dan kebutuhan bisnis yang telah ditetapkan.

#### 1. Pengembangan Antarmuka Pengguna untuk Penugasan

Tugas berikutnya adalah pengembangan antarmuka pengguna (*User Interface/UI*) untuk fitur penugasan pengguna (*user assignment*). Kegiatan ini tidak berfokus pada aspek perancangan visual, melainkan pada implementasi kode program yang merealisasikan logika dan fungsi interaktif dalam antarmuka. Komponen UI dikembangkan menggunakan *framework front-end* yang digunakan di lingkungan kerja perusahaan, dengan penerapan prinsip modularitas dan *component reusability*. Fitur ini memungkinkan pengelolaan peran, pembagian tanggung jawab, serta pelacakan status penugasan secara dinamis melalui sistem.

#### 2. Integrasi Alur Kerja Persetujuan

Tahapan selanjutnya melibatkan integrasi alur kerja persetujuan (*workflow integration*) ke dalam sistem aplikasi. Fitur ini mendukung berbagai mekanisme seperti persetujuan (*approval*) untuk menyetujui dan mengirimkan tugas ke

pengguna berikutnya berdasarkan hierarki yang telah dikonfigurasi, mekanisme penolakan (*rejection*) untuk mengembalikan dokumen ke pihak sebelumnya disertai alasan penolakan yang tercatat, promosi dan demosi (*promote/demote*) untuk memulai sirkulasi persetujuan sesuai dengan peran (*role*) yang sudah ditentukan sebelumnya, delegasi (*delegate*) untuk menugaskan pengguna pengganti jika pengguna sedang berhalangan

Penghentian sementara (*halt*) untuk menghentikan sementara alur persetujuan yang sedang berlangsung, penerusan kembali (*resume*) untuk melanjutkan kembali alur penugasan yang sedang berhenti, serta pembatalan alur (*abort*) untuk menghapus sebuah alur penugasan secara keseluruhan. Implementasi alur kerja ini bertujuan untuk memastikan proses pengambilan keputusan berjalan secara sistematis, terdokumentasi, dan transparan. Integrasi ini juga memperkuat efektivitas komunikasi antarunit kerja melalui otomasi dan pencatatan status dokumen secara *real-time*.

### 3. Memperbaiki Manajemen Sesi Pengguna

Tugas ini berkaitan dengan penyempurnaan sistem manajemen sesi pengguna (*session management*), khususnya pada pengaturan waktu tidak aktif (*idle timer*) yang sebelumnya tidak tersinkronisasi antar-tab pada peramban. Permasalahan ini mengakibatkan sesi pengguna berakhir secara tidak konsisten apabila lebih dari satu tab aktif digunakan secara bersamaan. Sedangkan, pembaruan yang dilakukan memungkinkan sistem melakukan penghentian sesi otomatis (*auto logout*) secara serempak di seluruh tab aktif, sehingga memperkuat keamanan data dan menjaga konsistensi status pengguna dalam sistem.

### 4. Implementasi Pelabelan Baris/Item

Tugas selanjutnya berkaitan dengan implementasi mekanisme pelabelan pada setiap item atau baris data, di mana label dapat berbeda untuk setiap pengguna. Fitur ini memungkinkan kategori atau status tertentu diterapkan sesuai preferensi atau peran masing-masing pengguna. Label yang dibuat oleh satu pengguna tidak terlihat atau mempengaruhi tampilan pengguna lain. Fitur ini meningkatkan kemampuan personalisasi dan organisasi kerja individual. Penerapan logika pelabelan dilakukan pada tingkat antarmuka dan backend, termasuk penyimpanan metadata label serta pemanggilan kembali label saat halaman dimuat.

## 5. Pembuatan Pusat Notifikasi

Tugas selanjutnya adalah implementasi pusat notifikasi berupa menu *dropdown* yang menampilkan pemberitahuan atau pengingat yang relevan bagi pengguna aktif. Fitur ini memuat notifikasi terkait tugas baru, perubahan status dokumen, pembaruan alur persetujuan, dan tindakan yang membutuhkan perhatian. Pengembangan dilakukan melalui integrasi logika backend untuk pemanggilan data notifikasi serta implementasi elemen antarmuka yang responsif dan mudah diakses.

## 6. Explorasi Penggunaan Tool C

Tugas ini dilakukan dengan melakukan kajian terhadap alat manajemen alur kerja dan otomasi proses bisnis, Tool C, untuk mengevaluasi potensi integrasinya dengan sistem yang sedang dikembangkan. Eksplorasi mencakup analisis terhadap diagram BPMN, kemampuan Tool tersebut dalam menangani proses bisnis, serta metode komunikasi dengan sistem yang sedang dikembangkan.

## 7. Refaktorisasi Kode

Aktivitas ini bertujuan meningkatkan keterbacaan dan pemeliharaan kode dengan melakukan identifikasi, menyatukan blok kode yang identik atau sangat mirip yang tersebar di beberapa file, melakukan *refactoring* dengan prinsip DRY (*Don't Repeat Yourself*) dengan mengekstrak logika yang sama ke dalam fungsi yang dapat dipanggil berulang, sehingga mengurangi redundansi dan memusatkan logika kode.

## 8. Penyelesaian Konflik Penggabungan Kode

Sebagai bagian dari tim *developer* yang menggunakan sistem kontrol versi terdistribusi (Git), tugas rutin yang dilakukan adalah mengidentifikasi dan menyelesaikan konflik penggabungan saat menggabungkan cabang kode (*branch*). Proses ini memerlukan analisis terhadap perubahan yang saling bertentangan dari beberapa sumber, koordinasi dengan *developer* lain, dan memastikan integritas fungsional aplikasi tetap terjaga setelah penggabungan.

### 3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang pada PT Kalbe Farma Tbk diuraikan seperti pada Tabel 3.1.

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

Minggu Ke -	Pekerjaan yang dilakukan
1	Refaktor struktur kode dan menghapus <i>comment</i> yang tidak diperlukan
2	Refaktor struktur kode dan memperbaiki konflik pada kode
3	Riset <i>library package</i> untuk <i>file preview</i>
4	Riset <i>library package</i> untuk <i>file preview</i> dan penyimpanan <i>cloud</i> pihak ketiga
5	Riset <i>library package</i> untuk database pihak ketiga dan implementasi penyimpanan <i>cloud</i> pihak ketiga pada aplikasi web
6	Dokumentasi implementasi penyimpanan <i>cloud</i> dan debugging upload API
7	Briefing, design dan meeting pembuatan database schema
8	Pembahasan dan penambahan database schema, serta menghubungkan aplikasi ke database
9	Perbaikan database schema, implementasi mekanisme <i>drag and drop</i>
10	Debugging mekanisme <i>state management</i> dan <i>drag and drop</i> beserta UI
11	Mempelajari QA proyek dan diskusi flow aplikasi
12	Mempelajari flow aplikasi dan <i>error handling</i>
13	Debugging UI dan <i>session management</i>
14	Debugging <i>session management</i> dan diskusi aplikasi
15	Mempelajari integrasi BPMN dengan aplikasi dan debugging <i>user assignment</i>
16	Implementasi <i>real data fetching</i> dan debugging <i>user assignment</i> form
17	Mempelajari integrasi BPMN dengan aplikasi
18	Implementasi workflow <i>user assignment</i> dan pembuatan RPC
19	Membuat alur <i>approval</i>
20	Membuat alur <i>approval</i> dan <i>promote</i>
21	Membuat alur <i>demote</i>
22	Update UI dan pembuatan alur <i>halt, abort, resume</i>

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

Minggu Ke -	Pekerjaan yang dilakukan
23	Membuat UI dialog untuk <i>action list</i>
24	Membuat alur delegasi tugas ke pengguna baru serta refactor kode
25	Memperbaiki alur pembuatan dan penambahan UI
26	Memperbaiki alur editing dan pembuatan alur labelling serta toggle label
27	Membuat UI dan alur notifikasi serta template dan pengingat
28	Debugging alur notifikasi dan update template notifikasi

### 3.3.1 Pengembangan Antarmuka Penugasan Pengguna

Salah satu kegiatan utama dalam pelaksanaan magang pada Divisi *Corporate Digital Technology* PT Kalbe Farma Tbk adalah pengembangan fitur *User Assignment* pada sistem aplikasi internal perusahaan. Fitur ini berfungsi untuk mengatur penugasan pengguna berdasarkan peran tertentu dalam alur kerja sistem, sehingga mendukung pengendalian akses dan distribusi tanggung jawab secara terstruktur.

#### A User Requirement Antarmuka Penugasan Pengguna

Tahap awal pengembangan fitur *User Assignment* diawali dengan proses analisis kebutuhan pengguna. Analisis ini dilakukan melalui diskusi dan pertemuan bersama manajer produk serta pengguna akhir dari departemen terkait. Dari hasil analisis tersebut, diperoleh beberapa kebutuhan utama yang harus dipenuhi oleh sistem, antara lain:

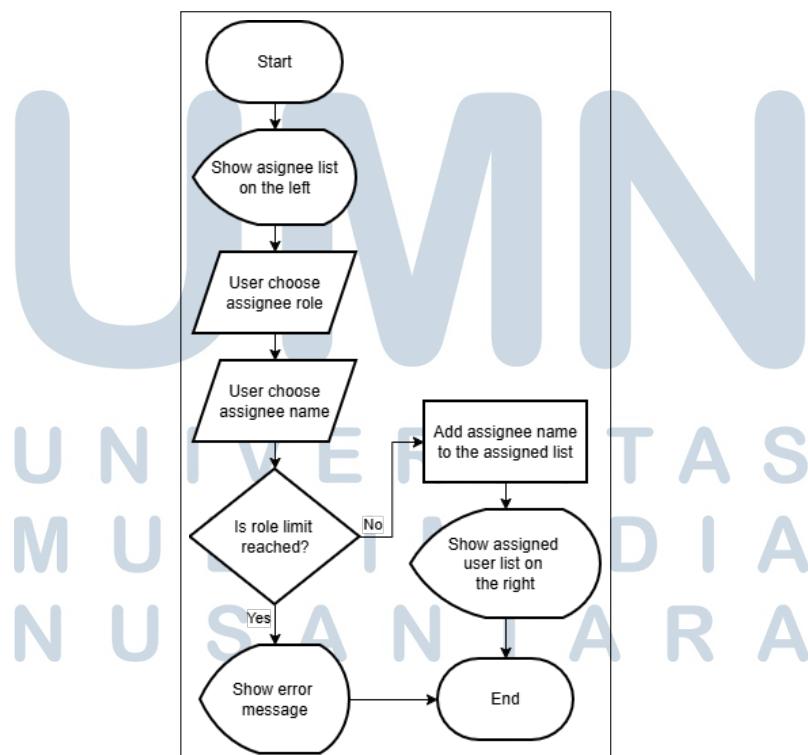
1. Sistem harus mampu mengelola beberapa peran pengguna (*user roles*) secara terpisah;
2. Sistem harus membatasi jumlah pengguna yang dapat ditugaskan pada setiap peran sesuai dengan kapasitas yang telah ditentukan;

3. Sistem harus menampilkan pesan kesalahan (*error message*) apabila pengguna mencoba menambahkan penugasan melebihi batas yang diperbolehkan;
4. Pada saat proses pengeditan data berlangsung, kolom atau komponen penugasan pengguna harus dinonaktifkan (*disabled*) untuk mencegah perubahan yang tidak diizinkan.

Kebutuhan tersebut menjadi acuan utama dalam proses perancangan dan implementasi fitur agar sesuai dengan standar operasional serta kebutuhan bisnis perusahaan.

## B Perancangan Antarmuka Penugasan Pengguna

Berdasarkan kebutuhan pengguna yang telah dianalisis, dilakukan perancangan alur kerja fitur *User Assignment* untuk memastikan proses penugasan berjalan secara sistematis dan terkontrol. Perancangan ini mencakup penentuan alur interaksi pengguna, validasi batas penugasan, serta mekanisme penyimpanan atau pembatalan perubahan data. Alur proses penugasan pengguna digambarkan dalam bentuk flowchart yang ditunjukkan pada Gambar 3.2.



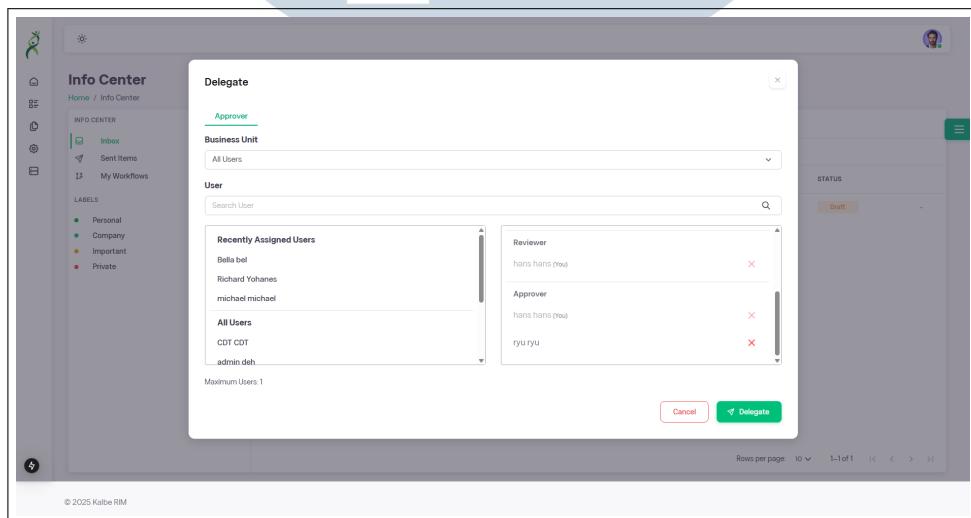
Gambar 3.2. Flowchart alur *User Assignment*

Perancangan ini bertujuan untuk memastikan bahwa setiap peran memiliki mekanisme pengelolaan pengguna yang terpisah, serta mencegah terjadinya kesalahan penugasan yang dapat berdampak pada integritas data dan alur kerja sistem.

## C Implementasi Antarmuka Penugasan Pengguna

Tahap implementasi dilakukan dengan mengembangkan antarmuka dan logika interaksi berdasarkan rancangan yang telah dibuat. Fitur *User Assignment* diimplementasikan menggunakan pendekatan navigasi berbasis tab, di mana setiap tab merepresentasikan satu jenis peran pengguna. Pengguna dapat berpindah antar-tab untuk melakukan penugasan sesuai dengan peran yang diinginkan.

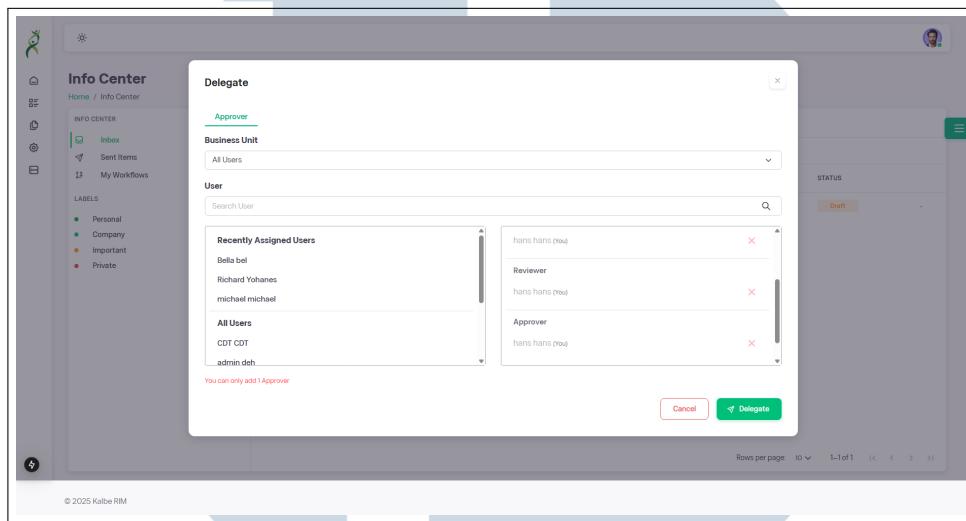
Tampilan keseluruhan antarmuka fitur *User Assignment* dapat dilihat pada Gambar 3.3. Pada tampilan ini, sistem menampilkan struktur utama halaman yang terdiri dari tab peran pengguna serta dua daftar utama, yaitu daftar pengguna dan daftar pengguna yang telah ditugaskan.



Gambar 3.3. Tampilan keseluruhan fitur *User Assignment*

Pada setiap tab, pengguna dapat memilih nama pengguna dari daftar di sisi kiri untuk kemudian ditambahkan ke daftar penugasan di sisi kanan sesuai dengan peran yang sedang aktif. Sistem secara otomatis melakukan validasi terhadap jumlah pengguna yang telah ditugaskan. Apabila jumlah penugasan melebihi batas yang ditentukan, sistem akan menolak penambahan dan menampilkan pesan kesalahan.

Contoh tampilan pesan kesalahan ketika batas jumlah penugasan terlampaui ditunjukkan pada Gambar 3.4. Pesan ini berfungsi sebagai umpan balik agar pengguna dapat segera memahami kesalahan yang terjadi dan melakukan penyesuaian yang diperlukan.



Gambar 3.4. Tampilan pesan kesalahan pada fitur *User Assignment*

Selain itu, sistem memungkinkan satu pengguna untuk memiliki lebih dari satu peran. Untuk menjaga konsistensi data, diterapkan mekanisme validasi tambahan yang memastikan tidak terjadi konflik penugasan antarperan. Selama proses pengeditan berlangsung, data penugasan yang telah tersimpan akan dinonaktifkan sehingga tidak dapat diubah secara langsung.

Setelah seluruh proses penugasan selesai, pengguna dapat menyimpan perubahan dengan menekan tombol *submit*, atau membatalkan perubahan melalui tombol *cancel* maupun ikon X. Implementasi ini dirancang untuk memberikan pengalaman penggunaan yang intuitif, aman, dan meminimalkan potensi kesalahan operasional.

### 3.3.2 Alur Kerja Persetujuan Dokumen

Salah satu kegiatan utama dalam pelaksanaan magang pada Divisi *Corporate Digital Technology* PT Kalbe Farma Tbk adalah pembuatan dan integrasi alur kerja persetujuan (*approval workflow*) ke dalam sistem aplikasi internal perusahaan. Fitur ini menjadi komponen esensial dalam sistem manajemen informasi yang digunakan untuk mendukung proses persetujuan dokumen sesuai dengan ketentuan dan standar regulasi yang berlaku.

## A User Requirement Alur Kerja Persetujuan

Tahapan awal pembuatan dilakukan melalui analisis kebutuhan sistem dan pengguna. Berdasarkan hasil diskusi dengan pembimbing teknis dan pihak pengguna, disimpulkan bahwa esensi utama sistem manajemen informasi yang dikembangkan adalah mekanisme persetujuan (*approval workflow*) yang berfungsi sebagai dasar pengendalian dokumen perusahaan. Dalam implementasinya, sistem persetujuan harus mampu:

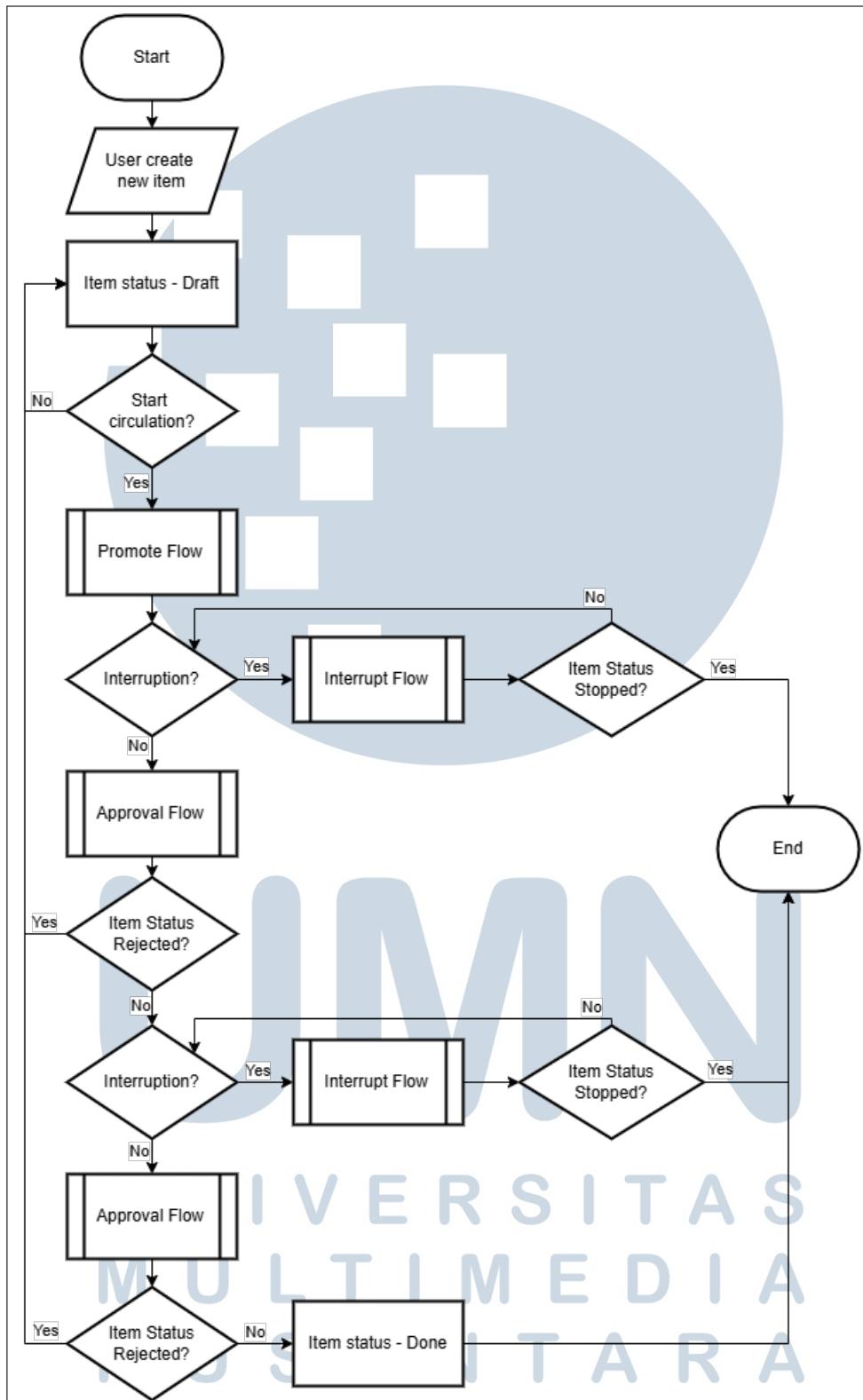
1. Mengelola proses persetujuan (*approval*) dan penolakan (*rejection*) dokumen sesuai dengan hierarki pengguna;
2. Mendukung proses sirkulasi dokumen untuk persetujuan (*promote*) dan penghentian sirkulasi untuk melakukan perubahan dokumen (*demote*);
3. Menyediakan fitur interupsi proses, seperti menghentikan sementara (*halt*), melanjutkan kembali (*resume*), dan membatalkan seluruh alur (*abort*);
4. Menjamin bahwa setiap dokumen yang telah disetujui dapat digunakan sebagai dokumen resmi untuk memenuhi kepatuhan terhadap regulasi yang berlaku.

Kebutuhan tersebut menunjukkan bahwa sistem tidak hanya berfungsi sebagai pelacak status dokumen, tetapi juga sebagai mekanisme pengendalian mutu dan kepatuhan terhadap prosedur operasional perusahaan.

## B Perancangan Alur Kerja Persetujuan

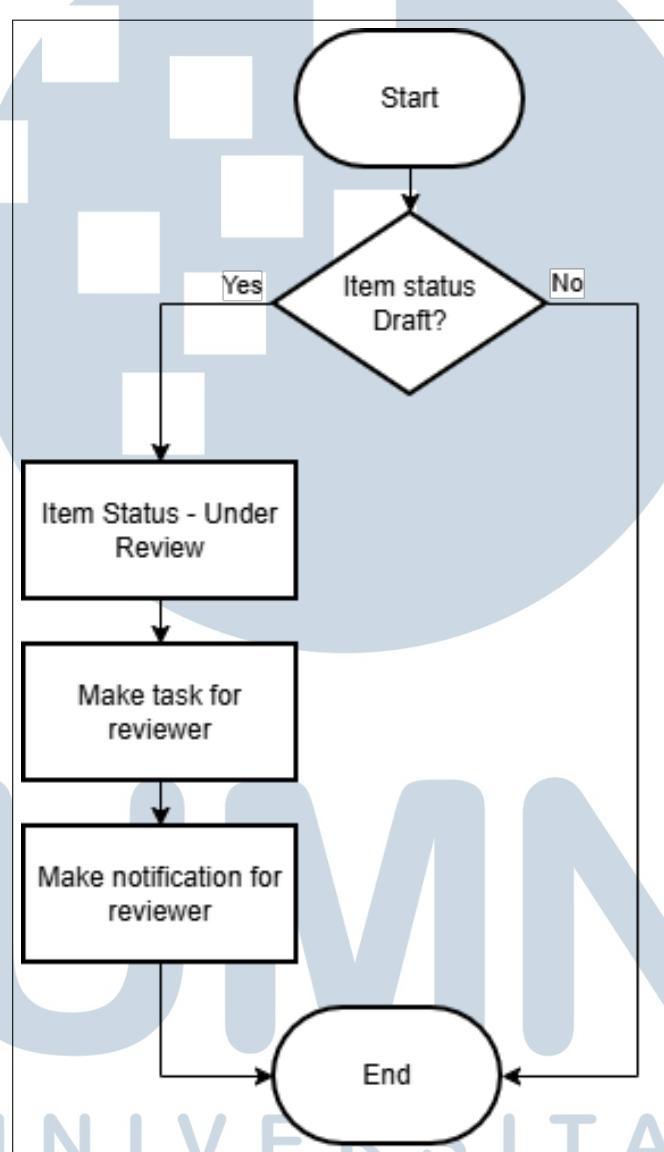
Perancangan sistem dilakukan dengan menyusun alur logika dan mekanisme persetujuan dokumen secara terstruktur. Tahapan ini bertujuan untuk menggambarkan hubungan antarperan pengguna, status dokumen, serta keputusan yang memengaruhi pergerakan dokumen dalam sistem.

Alur utama proses persetujuan dokumen dimulai ketika pengguna membuat sebuah item baru hingga dokumen mencapai status akhir. Rancangan alur utama tersebut ditunjukkan pada Gambar 3.5.



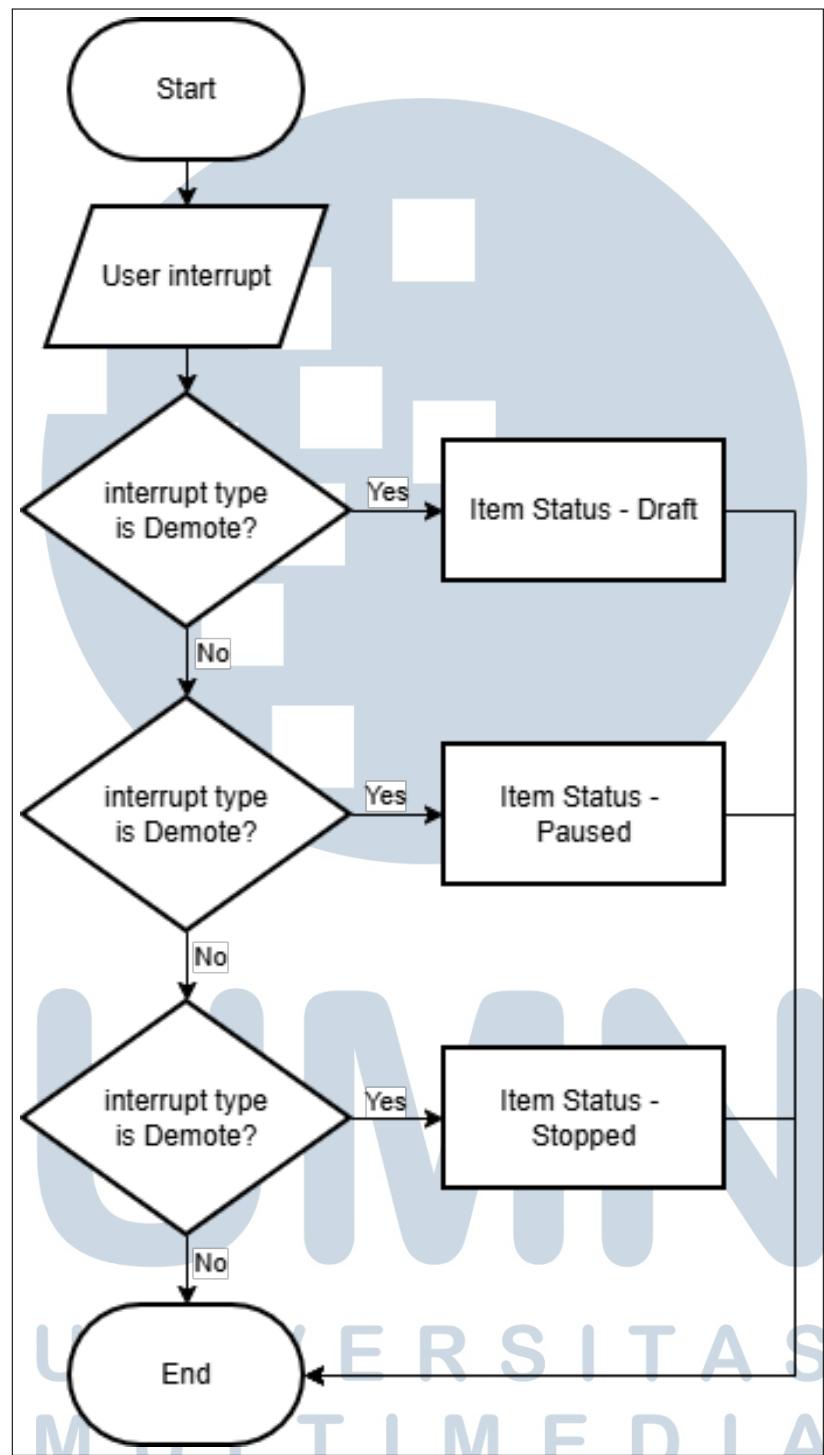
Gambar 3.5. Flowchart alur utama persetujuan dokumen

Setelah item dibuat dan berada pada status *Draft*, pengguna dapat memulai proses persetujuan dengan menjalankan aksi *promote*. Subproses ini mencakup validasi status item dan pembentukan tugas kepada *reviewer*, sebagaimana ditunjukkan pada Gambar 3.6.



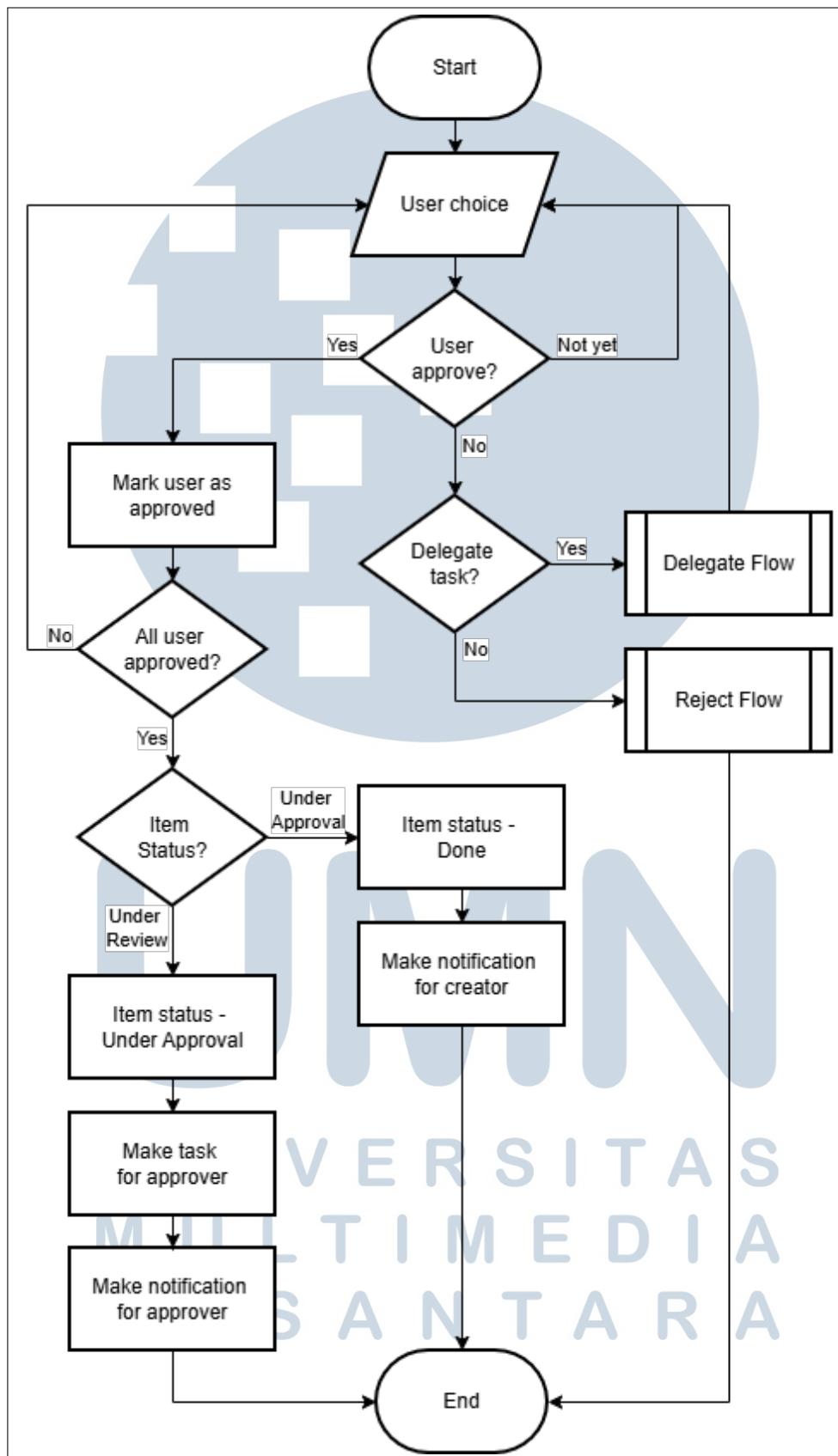
Gambar 3.6. Flowchart subproses promote

Dalam perancangannya, sistem juga menyediakan mekanisme interupsi untuk menangani kondisi tertentu, seperti kebutuhan revisi atau penghentian sementara proses. Hubungan antara alur utama dan mekanisme interupsi (*demote*, *halt*, dan *abort*) ditunjukkan pada Gambar 3.7.



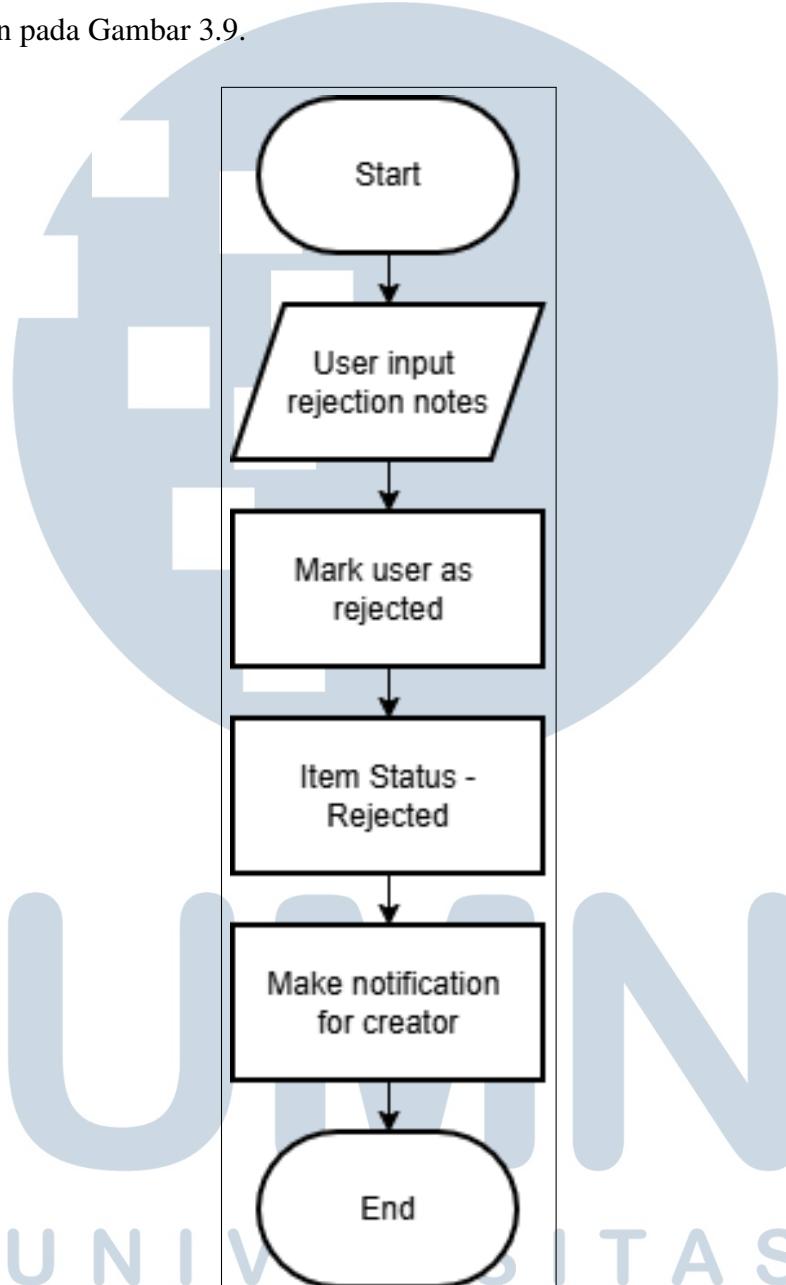
Gambar 3.7. Flowchart subproses interrupt

Setelah tahap evaluasi oleh *reviewer*, dokumen dapat dilanjutkan ke tahap persetujuan akhir oleh *approver*. Rancangan alur persetujuan hingga dokumen mencapai status *Done* ditunjukkan pada Gambar 3.8.



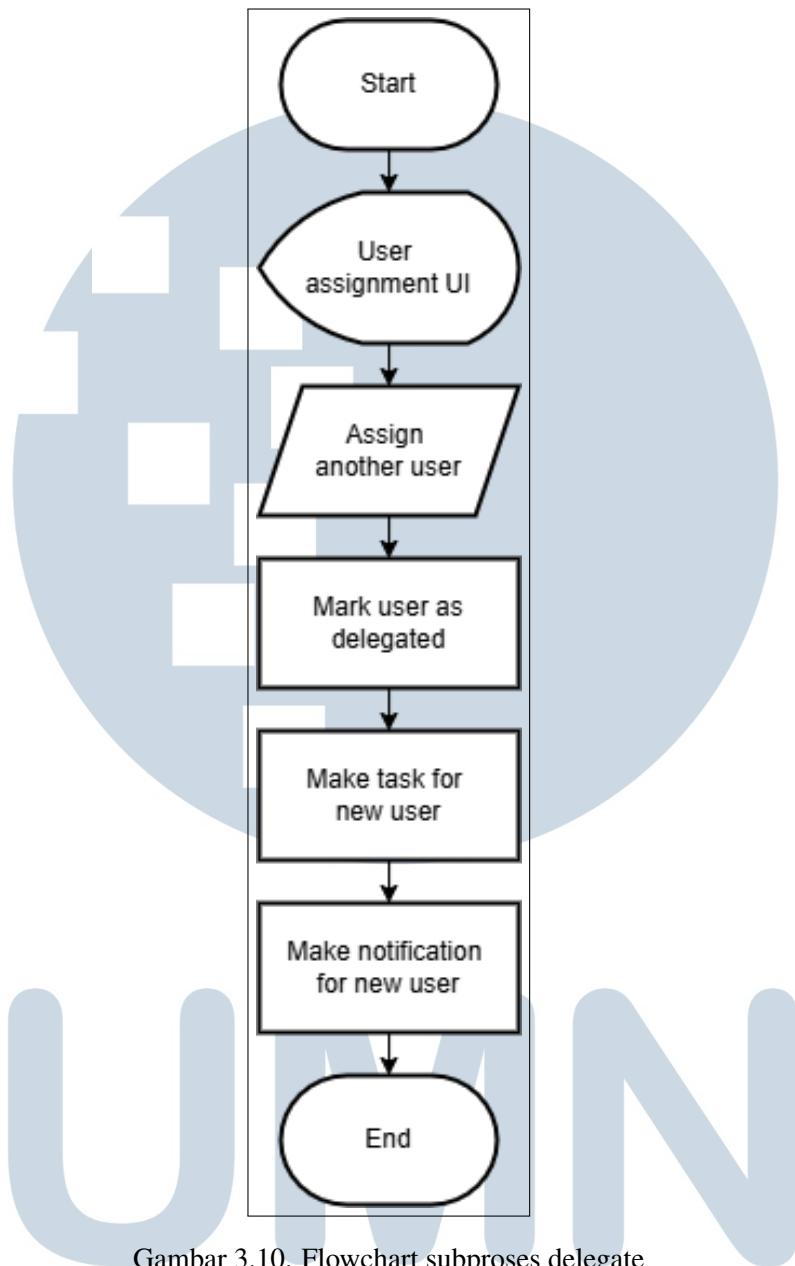
Gambar 3.8. Flowchart subproses approval

Selain persetujuan, sistem juga dirancang untuk menangani proses penolakan dokumen. Alur penolakan beserta pencatatan alasan penolakan ditunjukkan pada Gambar 3.9.



Gambar 3.9. Flowchart subproses reject

Sistem juga memungkinkan pendeklegasian tugas persetujuan kepada pengguna lain. Rancangan alur pendeklegasian tugas ditunjukkan pada Gambar 3.10.

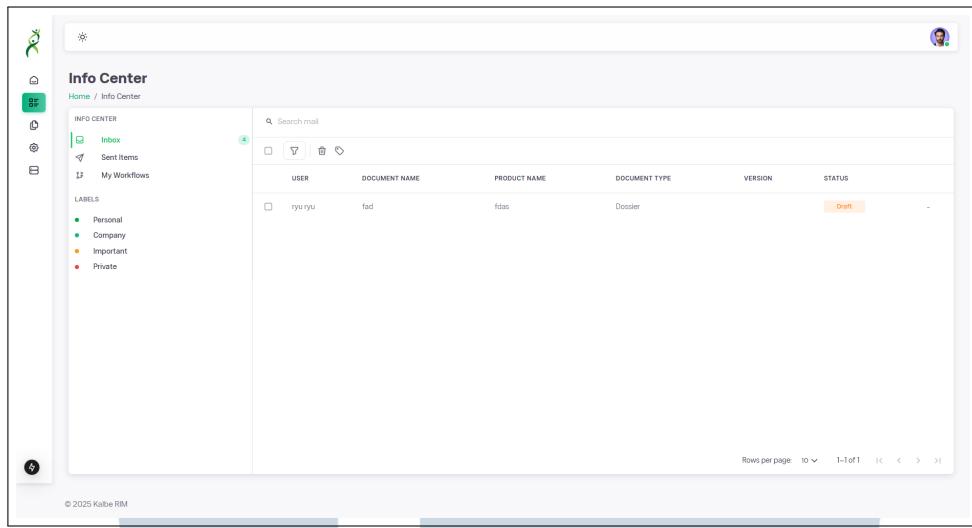


Gambar 3.10. Flowchart subproses delegate

### C Implementasi Alur Kerja Persetujuan

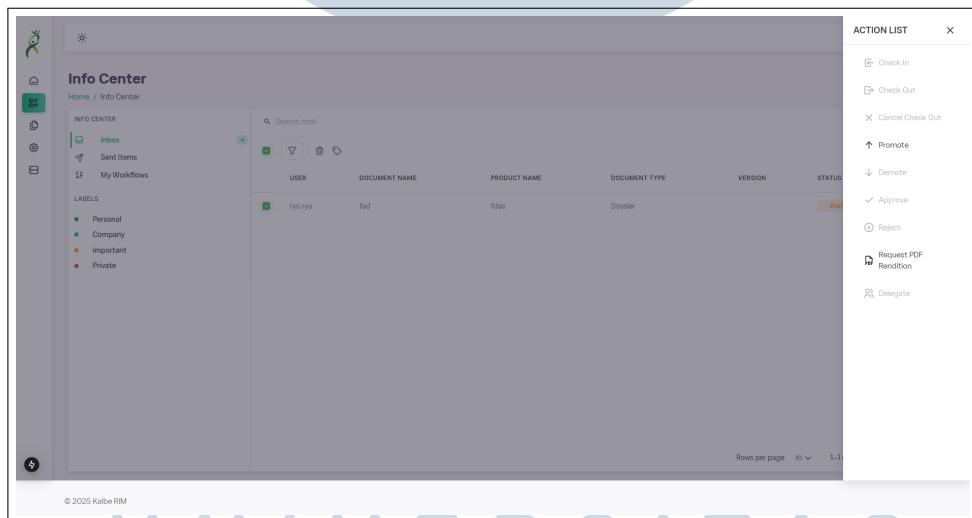
Tahap implementasi dilakukan dengan merealisasikan rancangan sistem ke dalam antarmuka aplikasi yang dapat digunakan oleh pengguna. Implementasi ini mencakup penyediaan tampilan informasi tugas, aksi persetujuan, serta detail status dokumen.

Tampilan awal fitur *approval workflow* disajikan dalam bentuk daftar tugas yang diterima oleh pengguna. Antarmuka tersebut ditunjukkan pada Gambar 3.11.



Gambar 3.11. Tampilan awal fitur approval workflow

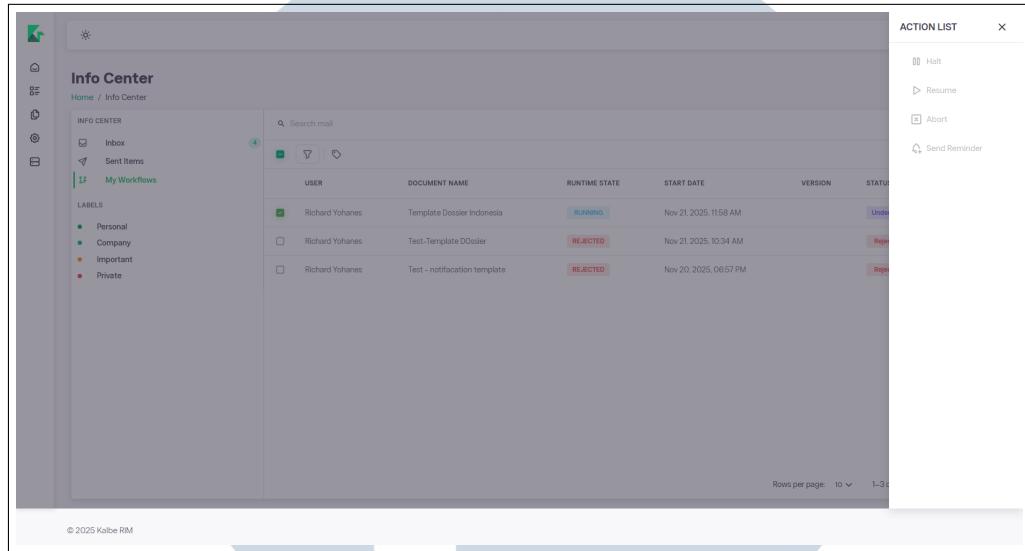
Pada halaman yang sama, sistem menyediakan berbagai aksi utama seperti *promote*, *approve*, *reject*, *demote*, dan *delegate*. Daftar aksi tersebut ditunjukkan pada Gambar 3.12.



Gambar 3.12. Daftar aksi pada alur persetujuan dokumen

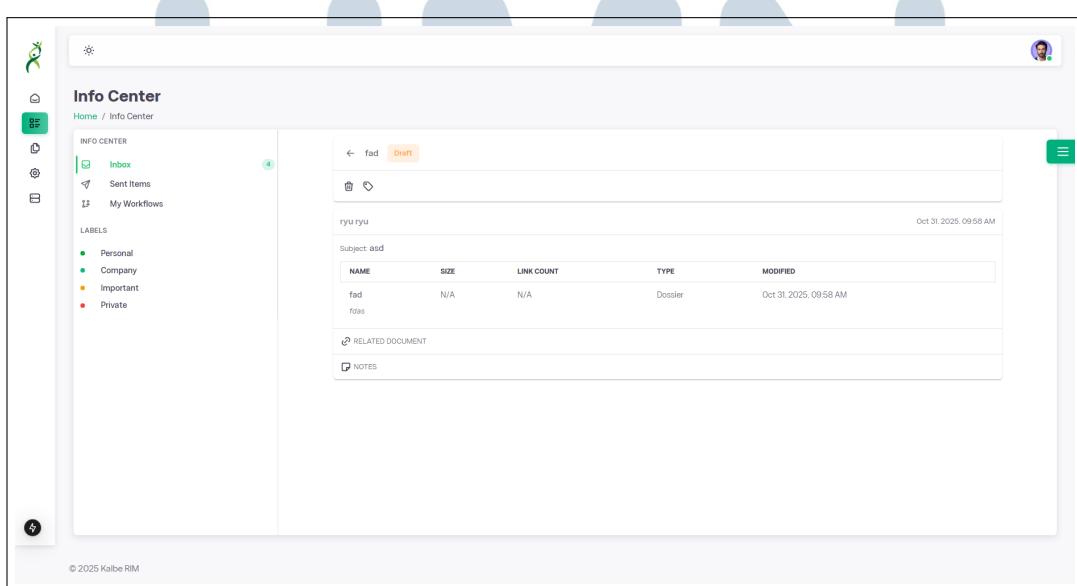
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

Selain aksi utama, sistem juga menyediakan aksi lanjutan untuk mengelola sirkulasi dokumen, seperti *halt*, *resume*, *abort*, dan *send reminder*. Tampilan aksi lanjutan tersebut diperlihatkan pada Gambar 3.13.



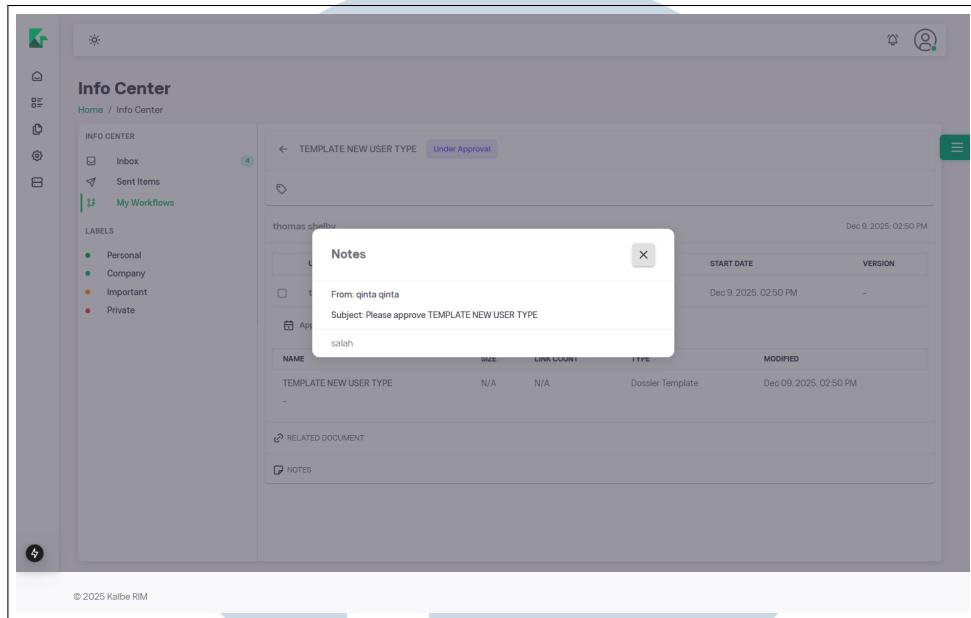
Gambar 3.13. Aksi lanjutan pada approval workflow

Sistem juga menampilkan detail status dokumen beserta riwayat proses persetujuan secara kronologis. Tampilan detail tersebut ditunjukkan pada Gambar 3.14.



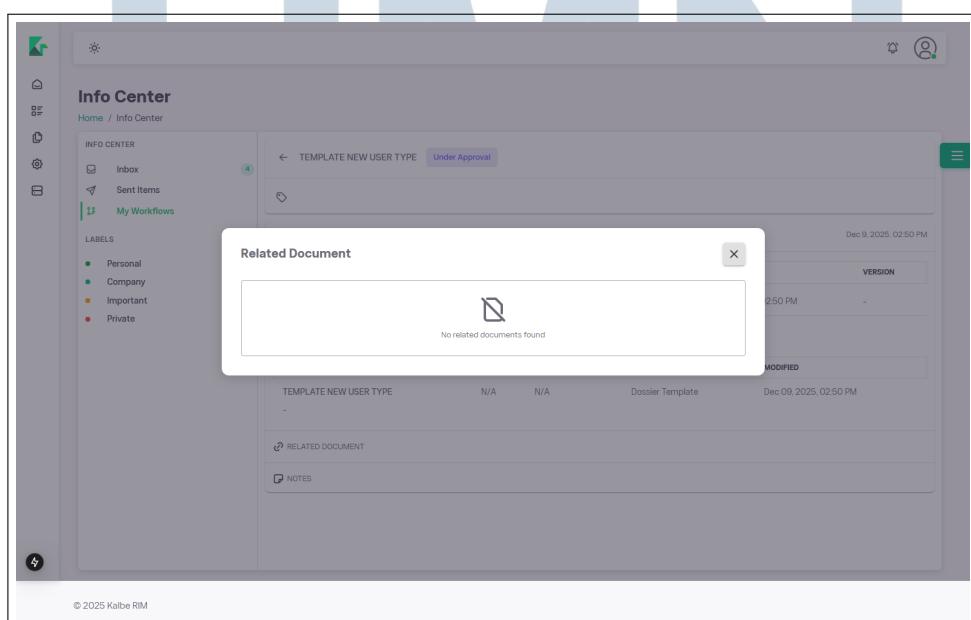
Gambar 3.14. Tampilan detail proses approval workflow

Pada halaman detail, pengguna dapat melihat catatan penolakan setelah menekan tombol *notes*, sebagaimana ditunjukkan pada Gambar 3.15.



Gambar 3.15. Tampilan catatan penolakan dokumen

Selain itu, pengguna juga dapat melihat dokumen pendukung setelah menekan tombol *related document*. Tampilan daftar dokumen terkait ditunjukkan pada Gambar 3.16.



Gambar 3.16. Tampilan daftar dokumen terkait

### **3.3.3 Perbaikan Manajemen Sesi Pengguna**

Salah satu kegiatan utama dalam pelaksanaan magang pada PT Kalbe Farma Tbk adalah pengembangan sistem manajemen sesi pengguna (*user session management*). Fitur ini berfungsi sebagai bagian dari mekanisme keamanan aplikasi dan pada tahap ini dilakukan perancangan ulang mekanisme pelacakan aktivitas pengguna agar sistem dapat mempertahankan sesi secara konsisten di seluruh tab peramban.

#### **A User Requirement Manajemen Sesi**

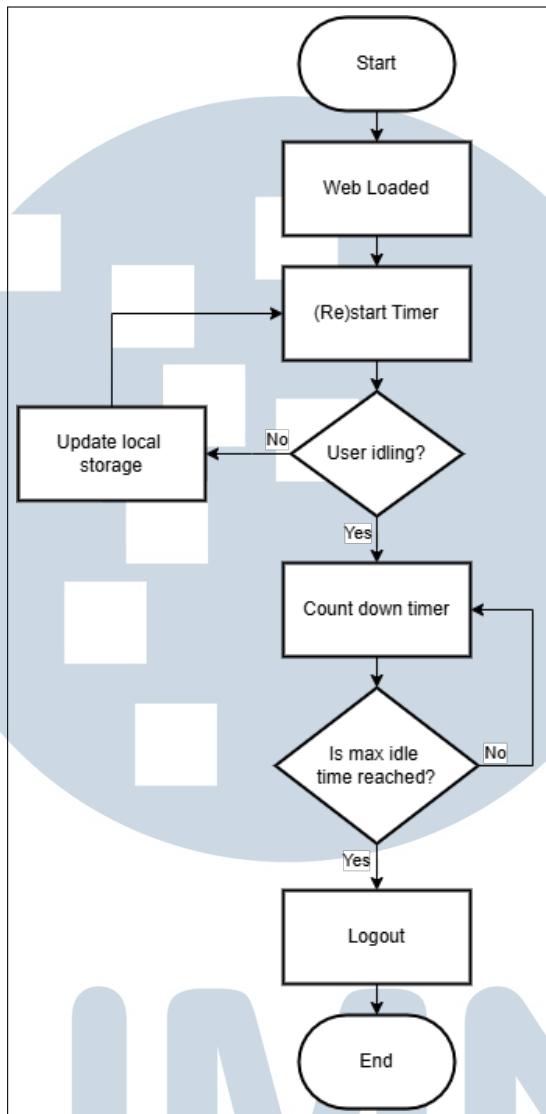
Analisis kebutuhan pengguna dilakukan berdasarkan evaluasi terhadap sistem manajemen sesi yang telah berjalan sebelumnya. Dari hasil analisis tersebut, ditemukan bahwa sistem belum mendukung sinkronisasi sesi antar-tab pada peramban (*browser*). Akibatnya, sesi pengguna dapat berakhir secara tidak konsisten meskipun pengguna masih aktif berinteraksi pada salah satu tab aplikasi.

Kondisi ini menimbulkan ketidaknyamanan karena sistem melakukan *logout* otomatis ketika tab tertentu dianggap tidak aktif, walaupun aktivitas masih berlangsung di tab lain. Oleh karena itu, dibutuhkan mekanisme baru yang mampu mendeteksi aktivitas pengguna secara menyeluruh di seluruh tab yang terbuka, sehingga sesi tetap aktif selama masih terdapat interaksi pengguna.

Kebutuhan utama pengguna dalam pengembangan ini adalah sistem yang mampu mempertahankan sesi secara konsisten antar-tab, mencegah *logout* yang tidak diinginkan, serta tetap menerapkan mekanisme keamanan berupa *logout* otomatis ketika tidak ada aktivitas sama sekali dalam jangka waktu tertentu.

#### **B Perancangan Manajemen Sesi Baru**

Berdasarkan kebutuhan pengguna yang telah diidentifikasi, dilakukan perancangan ulang mekanisme manajemen sesi dengan menambahkan fitur sinkronisasi aktivitas antar-tab. Perancangan ini berfokus pada alur pemantauan aktivitas pengguna dan pengelolaan waktu tidak aktif (*idle timer*) yang terintegrasi di seluruh tab aplikasi.



Gambar 3.17. Flowchart proses idle timer dan sinkronisasi antar-tab

Flowchart pada Gambar 3.17 menggambarkan alur utama sistem yang dirancang, dimulai dari pendekslan aktivitas pengguna, penyetelan ulang waktu tidak aktif, hingga keputusan sistem untuk melakukan logout otomatis. Diagram ini menunjukkan bagaimana setiap aktivitas pada salah satu tab akan memengaruhi status sesi secara global.

Perancangan ini juga mencakup penggunaan media komunikasi antar-tab, sehingga setiap perubahan status aktivitas dapat disebarluaskan secara real-time. Dengan pendekatan tersebut, seluruh tab memiliki pemahaman yang sama mengenai kondisi sesi pengguna dan dapat bertindak secara konsisten.

## C Implementasi Manajemen Sesi Baru

Tahap implementasi dilakukan dengan mengembangkan dan menyempurnakan logika sistem yang telah dirancang sebelumnya. Fokus utama implementasi adalah penerapan mekanisme pemantauan aktivitas pengguna dan sinkronisasi *idle timer* antar-tab agar sesuai dengan rancangan yang telah ditetapkan.

Setiap tab aplikasi diimplementasikan untuk mendeteksi berbagai jenis aktivitas pengguna, seperti klik, penekanan tombol, dan interaksi dengan komponen antarmuka. Ketika aktivitas terdeteksi, sistem akan mengatur ulang waktu tidak aktif dan menyinkronkan informasi tersebut ke tab lain yang sedang terbuka.

Proses sinkronisasi ini diimplementasikan dengan memanfaatkan *local storage* pada peramban sebagai media berbagi status antar-tab. Jika tidak terdapat aktivitas pada seluruh tab hingga melewati batas waktu yang ditentukan, sistem akan melakukan logout otomatis secara serentak untuk menjaga keamanan dan konsistensi sesi.

Implementasi ini tidak hanya meningkatkan pengalaman pengguna dengan mengurangi logout yang tidak diinginkan, tetapi juga memastikan sistem tetap memenuhi kebijakan keamanan informasi perusahaan. Antarmuka pengguna yang ditampilkan memberikan umpan balik yang jelas terkait status sesi, sehingga pengguna dapat memahami kondisi sesi mereka secara intuitif.

### 3.3.4 Pembuatan Item Labeling & Filtering

Fitur *item labeling* dan *filtering* dikembangkan sebagai bagian dari peningkatan antarmuka pada modul *approval workflow*. Fitur ini ditujukan untuk membantu pengguna dalam mengelola, mengelompokkan, dan menelusuri item persetujuan secara lebih efisien melalui mekanisme pelabelan visual dan penyaringan berdasarkan kategori tertentu.

## A User Requirement Item Labeling & Filtering

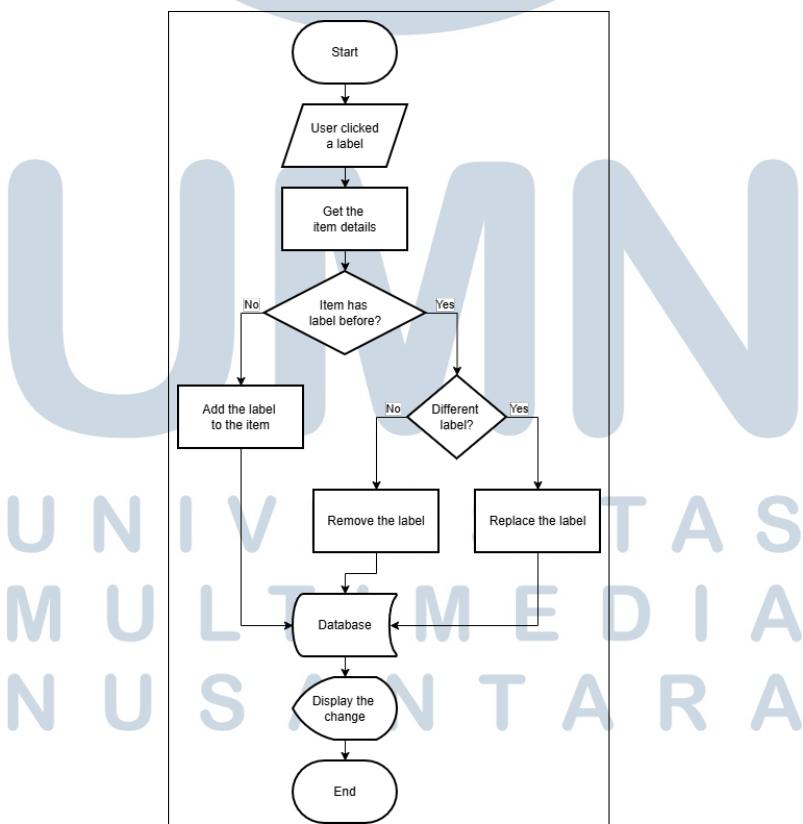
Dalam proses penggunaan sistem manajemen dokumen, pengguna menghadapi jumlah item persetujuan yang cukup banyak dan beragam. Kondisi ini menyulitkan pengguna dalam melakukan peninjauan dokumen secara cepat apabila tidak tersedia mekanisme pengelompokan yang fleksibel. Oleh

karena itu, pengguna menyampaikan kebutuhan akan fitur yang memungkinkan pengelompokan item berdasarkan kategori yang dapat ditentukan sendiri.

Pengguna menginginkan setiap item pada *approval workflow* dapat diberikan label kategori, yaitu *personal*, *company*, *important*, dan *private*. Label tersebut diharapkan berfungsi sebagai indikator visual yang mudah dikenali serta dapat digunakan sebagai dasar penyaringan tampilan. Dengan adanya fitur ini, pengguna dapat mengatur prioritas pekerjaan, mempercepat proses pencarian item, serta meningkatkan keteraturan dalam alur kerja persetujuan dokumen.

## B Perancangan Item Labeling & Filtering

Berdasarkan kebutuhan pengguna, dilakukan perancangan alur sistem untuk mendukung proses pemberian, pengubahan, dan penghapusan label pada setiap item. Alur proses dimulai dari pemilihan item oleh pengguna, dilanjutkan dengan pemilihan kategori label, hingga sistem memperbarui status label dan menampilkan perubahan secara visual. Rancangan alur logika tersebut divisualisasikan dalam bentuk flowchart yang ditunjukkan pada Gambar 3.18.



Gambar 3.18. Flowchart perubahan label

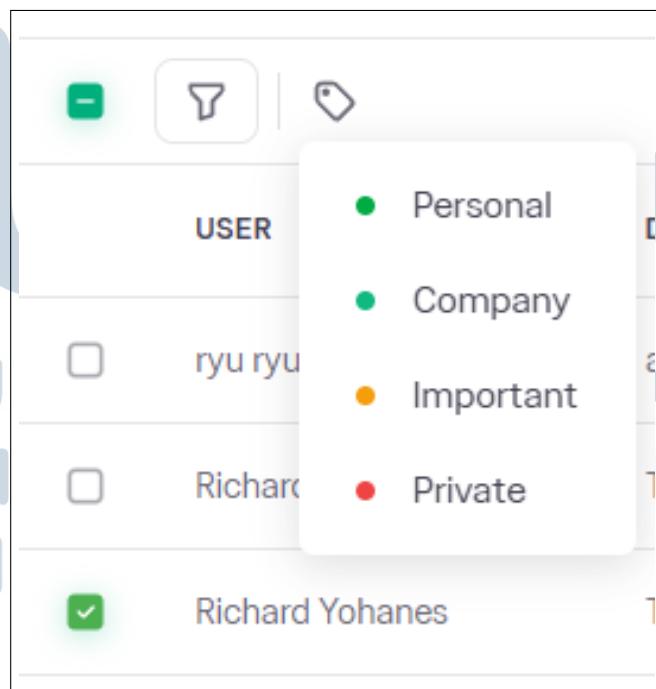
## C Implementasi Item Labeling & Filtering

Pada tahap implementasi, setiap item pada *approval workflow* secara default berada dalam kondisi tidak berlabel. Kondisi ini ditampilkan melalui indikator titik berwarna abu-abu pada kolom paling kanan tabel sebagai penanda status netral. Tampilan antarmuka awal tanpa label ditunjukkan pada Gambar 3.19.

USER	DOCUMENT NAME	RUNTIME STATE	START DATE	VERSION	STATUS
<input type="checkbox"/> ryu ryu	agrasg	RUNNING	Nov 21, 2025, 12:26 PM	-	•
<input type="checkbox"/> Richard Yohanes	Template Dossier Indonesia	RUNNING	Nov 21, 2025, 11:58 AM	Under Approval	•
<input type="checkbox"/> Richard Yohanes	Test-Template Dossier	RUNNING	Nov 21, 2025, 10:34 AM	Under Approval	•
<input type="checkbox"/> Richard Yohanes	Test - notification template	RUNNING	Nov 20, 2025, 06:57 PM	Under Approval	•

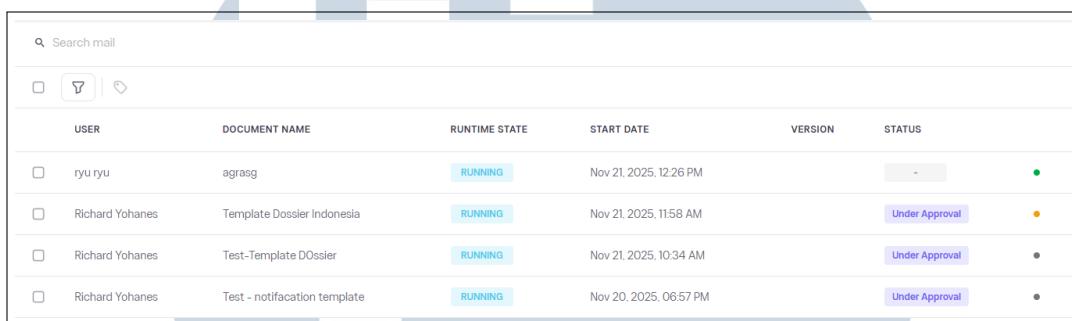
Gambar 3.19. Tampilan antarmuka tanpa label

Pengguna dapat memberikan label dengan terlebih dahulu memilih satu item menggunakan kotak centang yang tersedia di sisi kiri tabel. Setelah item dipilih, pengguna menekan ikon label pada bagian atas tabel untuk menampilkan daftar kategori label dalam bentuk *dropdown*. Tampilan menu pemilihan label pada item yang belum berlabel dapat dilihat pada Gambar 3.20.



Gambar 3.20. Tampilan pemilihan label saat item tidak berlabel

Setelah pengguna memilih kategori tertentu, sistem secara langsung menampilkan indikator warna sesuai label yang diterapkan tanpa perlu memuat ulang halaman. Warna hijau gelap digunakan untuk label *personal*, hijau terang untuk *company*, kuning untuk *important*, dan merah untuk *private*. Contoh tampilan item setelah diberikan label ditunjukkan pada Gambar 3.21.

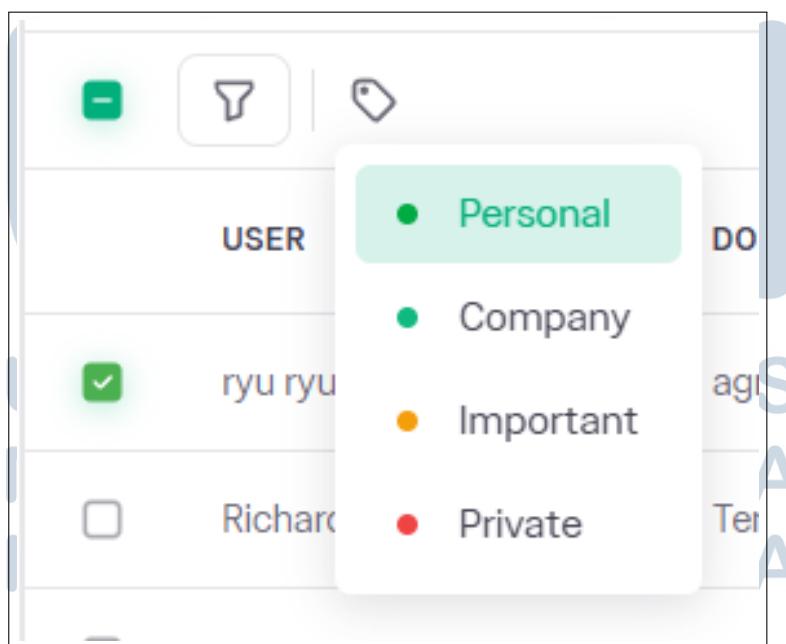


A screenshot of a user interface showing a list of items. The columns are labeled: USER, DOCUMENT NAME, RUNTIME STATE, START DATE, VERSION, and STATUS. The STATUS column contains colored dots representing labels: green (Personal), blue (Company), orange (Important), and red (Private). The items listed are: ryu ryu (Personal), Richard Yohanes (Company), Richard Yohanes (Important), and Richard Yohanes (Important).

USER	DOCUMENT NAME	RUNTIME STATE	START DATE	VERSION	STATUS
<input type="checkbox"/> ryu ryu	agrasg	RUNNING	Nov 21, 2025, 12:26 PM	-	●
<input type="checkbox"/> Richard Yohanes	Template Dossier Indonesia	RUNNING	Nov 21, 2025, 11:58 AM	Under Approval	○
<input type="checkbox"/> Richard Yohanes	Test-Template DOssier	RUNNING	Nov 21, 2025, 10:34 AM	Under Approval	●
<input type="checkbox"/> Richard Yohanes	Test - notification template	RUNNING	Nov 20, 2025, 06:57 PM	Under Approval	●

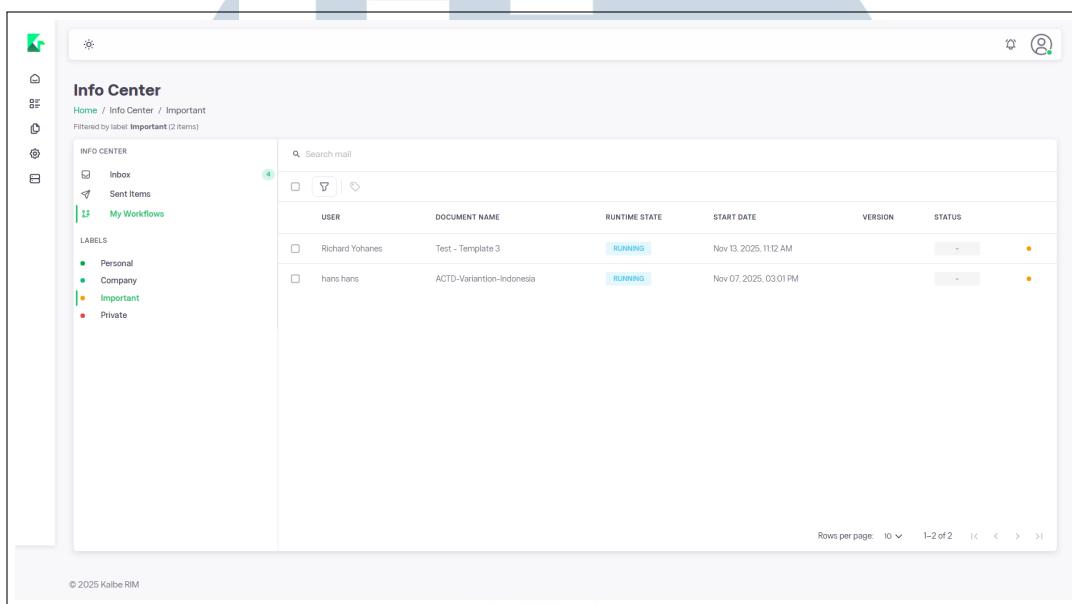
Gambar 3.21. Tampilan antarmuka setelah diberikan label

Sistem juga mendukung pengubahan dan penghapusan label. Ketika pengguna memilih item yang telah memiliki label dan kembali membuka menu *dropdown*, sistem akan menyoroti label yang sedang aktif. Dari menu ini, pengguna dapat mengganti label dengan kategori lain atau memilih label yang sama untuk menghapusnya. Tampilan kondisi ini diperlihatkan pada Gambar 3.22.



Gambar 3.22. Tampilan pemilihan label saat item memiliki label

Sebagai fitur pendukung, sistem menyediakan mekanisme *filtering* berdasarkan label. Panel daftar kategori ditempatkan secara permanen di sisi kiri antarmuka. Ketika pengguna memilih salah satu label, sistem akan memfilter daftar sehingga hanya item dengan kategori tersebut yang ditampilkan. Contoh tampilan saat filter aktif ditunjukkan pada Gambar 3.23.



Gambar 3.23. Tampilan memfilter aktif

Secara keseluruhan, implementasi fitur *item labeling* dan *filtering* mampu meningkatkan keteraturan dan efisiensi pengelolaan dokumen dalam modul *approval workflow*. Pengguna dapat dengan cepat mengidentifikasi, mengelompokkan, dan memprioritaskan item persetujuan sesuai dengan kebutuhan kerja masing-masing.

### 3.3.5 Pembuatan Pusat Notifikasi

Pengelolaan tugas dalam sistem alur persetujuan (*approval workflow*) yang kompleks memerlukan mekanisme pengingat yang efektif. Meskipun antarmuka utama aplikasi telah menyediakan daftar tugas, terdapat kemungkinan pengguna lupa atau tidak segera menyadari adanya tugas baru yang diberikan. Oleh karena itu, diperlukan sebuah subsistem yang secara aktif menginformasikan dan mengingatkan pengguna terkait aktivitas yang memerlukan perhatian. Implementasi sebuah pusat notifikasi (*notification center*) dirancang untuk

memenuhi kebutuhan ini, sehingga dapat meningkatkan kesadaran pengguna dan mempercepat respons terhadap setiap tugas atau permintaan.

### A User Requirement Pusat Notifikasi

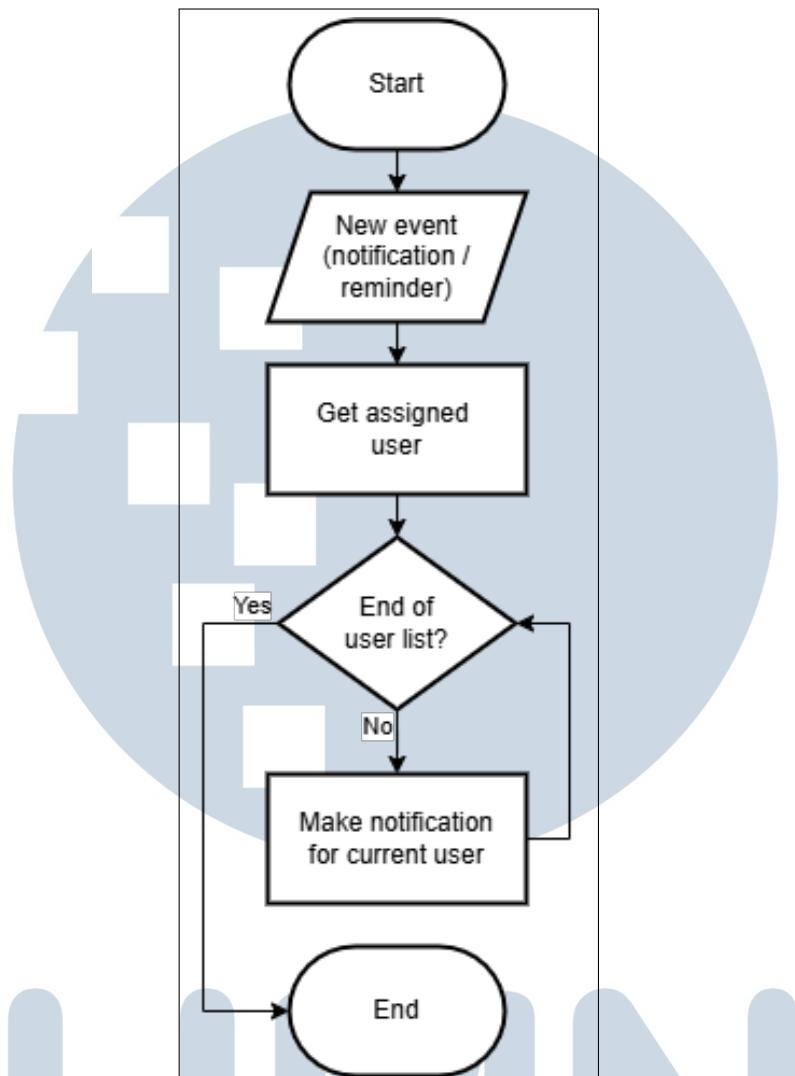
Kebutuhan pengguna terhadap fitur pusat notifikasi berangkat dari pengalaman penggunaan sistem sebelumnya yang telah memiliki mekanisme serupa dan dinilai penting untuk dihadirkan kembali pada sistem yang baru. Pengguna membutuhkan pemberitahuan yang muncul secara aktif dan tepat waktu ketika terdapat tugas baru, seperti pembuatan dokumen atau permintaan persetujuan yang dialamatkan kepada mereka. Selain itu, pengguna juga memerlukan pengingat berkelanjutan terhadap tugas-tugas yang belum diselesaikan agar tidak terlewat dalam alur kerja.

Berdasarkan kebutuhan tersebut, pusat notifikasi harus mampu menyampaikan informasi secara ringkas, mudah diakses, serta memberikan penanda yang jelas mengenai status setiap notifikasi. Dengan demikian, pengguna dapat segera mengidentifikasi tugas yang memerlukan perhatian dan meresponsnya tanpa harus menelusuri sistem secara manual.

### B Perancangan Pusat Notifikasi

Tahap perancangan pusat notifikasi diawali dengan penyusunan alur logika sistem yang menggambarkan bagaimana sebuah peristiwa bisnis (*event*) di dalam sistem dapat memicu pembuatan notifikasi. Perancangan ini mencakup identifikasi kondisi pemicu, penentuan pengguna tujuan notifikasi, serta mekanisme penyimpanan dan penyajian data notifikasi.

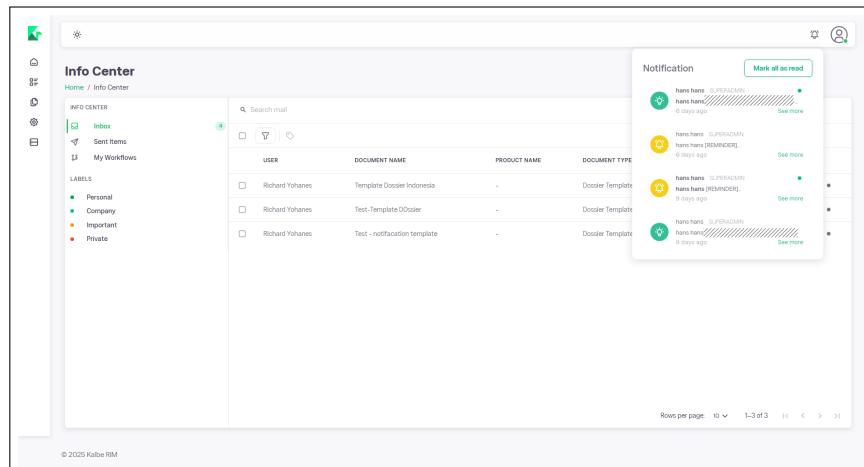
Alur proses pembuatan notifikasi divisualisasikan dalam bentuk flowchart yang ditunjukkan pada Gambar 3.24. Diagram ini memperlihatkan hubungan antara aktivitas sistem, proses pembuatan notifikasi, hingga notifikasi siap ditampilkan kepada pengguna. Flowchart tersebut menjadi acuan utama dalam pengembangan logika pada sisi *backend* dan integrasinya dengan antarmuka pengguna.



Gambar 3.24. Flowchart pembuatan notifikasi

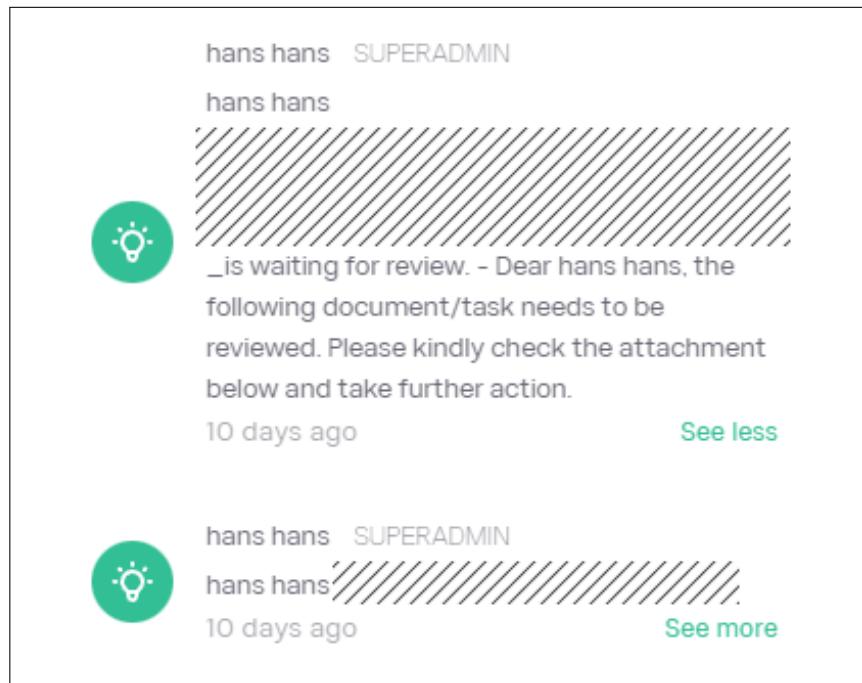
### C Implementasi Pusat Notifikasi

Implementasi pusat notifikasi dimulai dengan pengembangan antarmuka daftar notifikasi yang dapat diakses melalui ikon lonceng pada bagian atas halaman. Antarmuka ini disajikan dalam bentuk menu *dropdown* yang menampilkan ringkasan aktivitas terbaru, seperti penugasan baru, permintaan persetujuan, serta pengingat tugas yang masih tertunda. Tampilan awal pusat notifikasi ini dapat dilihat pada Gambar 3.25.



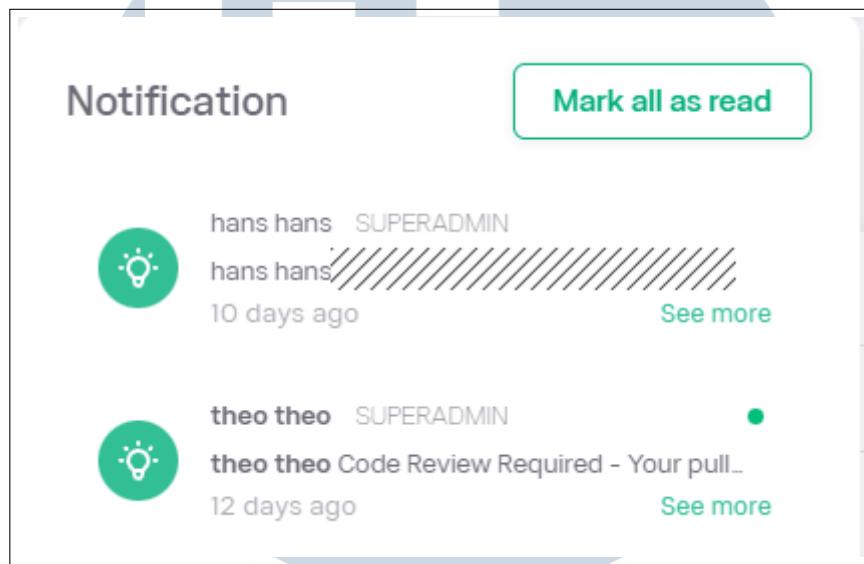
Gambar 3.25. Tampilan pusat notifikasi

Untuk mendukung keterbacaan informasi, setiap item notifikasi dirancang agar dapat diperluas (*expanded*) atau dipersingkat kembali (*collapsed*). Pengguna dapat melihat detail lengkap notifikasi dengan menekan tombol *see more*, dan mengembalikannya ke tampilan ringkas menggunakan tombol *see less*. Perbedaan antara tampilan notifikasi dalam kondisi *expanded* dan *collapsed* ditunjukkan pada Gambar 3.26. Pendekatan ini memungkinkan penyajian informasi yang fleksibel tanpa mengganggu tata letak antarmuka utama.



Gambar 3.26. Tampilan perbedaan notifikasi expanded dan collapsed

Selain itu, sistem menerapkan mekanisme penandaan status notifikasi secara otomatis. Ketika pengguna mengklik sebuah notifikasi, statusnya akan berubah menjadi *read*. Perbedaan visual antara notifikasi yang telah dibaca dan yang belum dibaca ditampilkan secara jelas, seperti yang diperlihatkan pada Gambar 3.27, untuk membantu pengguna memprioritaskan pesan yang masih memerlukan perhatian.



Gambar 3.27. Tampilan perbedaan notifikasi dibaca dan belum dibaca

Melalui implementasi antarmuka yang terintegrasi dengan mekanisme notifikasi berbasis *event-driven*, pusat notifikasi mampu menyampaikan informasi penting secara real-time. Keberadaan fitur-fitur yang ditunjukkan pada Gambar 3.25 hingga Gambar 3.27 memperkuat dukungan sistem terhadap alur persetujuan dokumen serta meningkatkan kesiapan pengguna dalam merespons dinamika proses bisnis yang berlangsung.

### 3.3.6 Eksplorasi Tools C

Eksplorasi Tool C dilakukan sebagai bagian dari kajian alternatif dalam merancang mekanisme penugasan pengguna pada sistem persetujuan dokumen internal. Tool C merupakan sebuah platform manajemen proses bisnis berbasis BPMN yang menyediakan pendekatan visual untuk memodelkan, mengotomatisasi, serta memantau alur kerja. Platform ini dipertimbangkan karena kemampuannya dalam merepresentasikan proses bisnis secara terstruktur dan terstandarisasi, sehingga diharapkan dapat meningkatkan keteraturan alur penugasan serta keterlacakkan proses persetujuan.

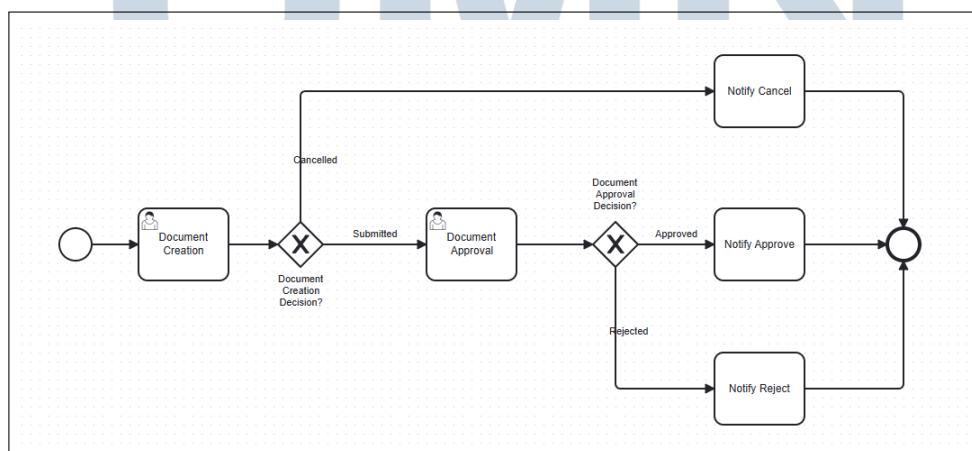
## A User Requirement Eksplorasi Tools C

Tahap awal pengembangan difokuskan pada identifikasi kebutuhan pengguna terkait mekanisme penugasan dalam alur persetujuan dokumen. Pengguna membutuhkan sistem yang mampu mendistribusikan tugas secara fleksibel, mendukung penetapan lebih dari satu pengguna untuk satu item persetujuan, serta mudah dikonfigurasi dan diintegrasikan dengan aplikasi yang telah berjalan. Selain itu, mekanisme penugasan diharapkan dapat beradaptasi dengan perubahan kebutuhan operasional tanpa memerlukan konfigurasi yang kompleks.

Berdasarkan kebutuhan tersebut, dilakukan eksplorasi terhadap Tool C untuk menilai kesesuaianya sebagai solusi pengelolaan penugasan berbasis proses bisnis. Fokus utama kajian diarahkan pada kemampuan Tool C dalam mengelola alur persetujuan, mendukung otomatisasi penugasan, serta fleksibilitas dalam pengaturan aktor atau pengguna yang terlibat dalam suatu proses.

## B Perancangan Eksplorasi Tools C

Pada tahap perancangan, Tool C digunakan untuk memodelkan alur persetujuan dokumen dalam bentuk diagram BPMN. Pemodelan ini bertujuan untuk mengevaluasi bagaimana proses penugasan pengguna direpresentasikan secara visual, mulai dari inisiasi dokumen, distribusi tugas kepada pengguna, hingga penyelesaian proses persetujuan. Diagram BPMN yang digunakan dalam proses kajian ditunjukkan pada Gambar 3.28.



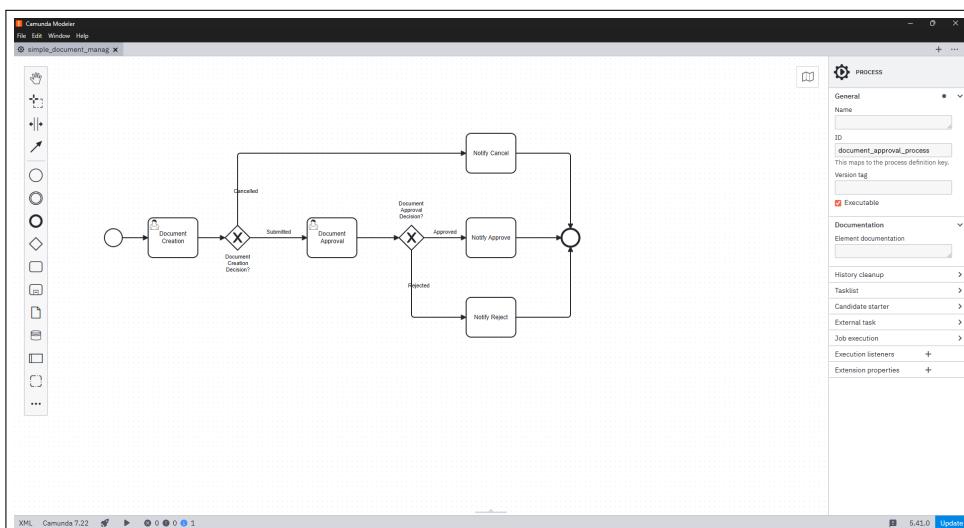
Gambar 3.28. Diagram BPMN approval workflow menggunakan Tool C

Melalui pemodelan tersebut, dilakukan analisis terhadap kompleksitas struktur proses yang dihasilkan. Setiap tahapan dalam alur persetujuan memerlukan definisi elemen BPMN yang cukup rinci, termasuk penentuan peran pengguna, kondisi transisi, serta mekanisme pemicu otomatis. Hasil perancangan ini kemudian dibandingkan dengan kebutuhan sistem untuk menilai kesesuaian antara kompleksitas perancangan dan manfaat yang diperoleh.

### C Implementasi Eksplorasi Tools C

Tahap implementasi mencakup proses instalasi dan konfigurasi awal Tool C berdasarkan dokumentasi resmi yang tersedia. Setelah instalasi berhasil dilakukan, langkah berikutnya meliputi pengaturan lingkungan kerja, penyesuaian struktur proyek, serta eksplorasi fitur-fitur utama yang berkaitan dengan pengelolaan tugas dan alur proses. Dokumentasi resmi Tool C dipelajari secara sistematis, khususnya pada bagian pengelolaan tugas dan integrasi proses bisnis.

Tampilan antarmuka Tool C setelah proses konfigurasi awal ditunjukkan pada Gambar 3.29. Gambar ini memperlihatkan lingkungan kerja yang digunakan untuk memodelkan proses dan mengelola tugas pengguna.



Gambar 3.29. Tampilan antarmuka Tool C

Hasil implementasi dan pengujian awal menunjukkan bahwa Tool C memiliki tingkat kompleksitas yang relatif tinggi dibandingkan dengan kebutuhan teknis sistem yang dikembangkan. Selain itu, ditemukan keterbatasan signifikan pada mekanisme penugasan, yaitu tidak adanya dukungan untuk menetapkan

beberapa pengguna secara dinamis pada satu tugas. Keterbatasan ini tidak sejalan dengan kebutuhan sistem persetujuan yang menuntut fleksibilitas dalam penugasan pengguna.

Berdasarkan hasil evaluasi dan diskusi bersama pembimbing teknis, diputuskan bahwa Tool C tidak dilanjutkan untuk diintegrasikan ke dalam sistem. Pengembangan sistem kemudian diarahkan kembali pada pendekatan berbasis *database* untuk mengelola penugasan pengguna. Pendekatan ini dinilai lebih sederhana, fleksibel, serta mudah diintegrasikan dengan arsitektur sistem yang telah ada, sekaligus memberikan ruang pengembangan yang lebih luas di masa mendatang.

### 3.3.7 Refaktor Kode

Tahap refaktor kode dilakukan sebagai bagian dari upaya peningkatan kualitas struktur perangkat lunak serta pemeliharaan jangka panjang pada sistem. Refaktor diperlukan untuk mengurangi kompleksitas, meningkatkan konsistensi, dan memastikan bahwa komponen aplikasi memiliki tingkat keterbacaan serta skalabilitas yang lebih baik. Kegiatan ini berfokus pada penyederhanaan struktur kode, pengurangan duplikasi, dan penerapan modularitas yang selaras dengan prinsip *clean code* serta praktik pengembangan perangkat lunak yang berkelanjutan.

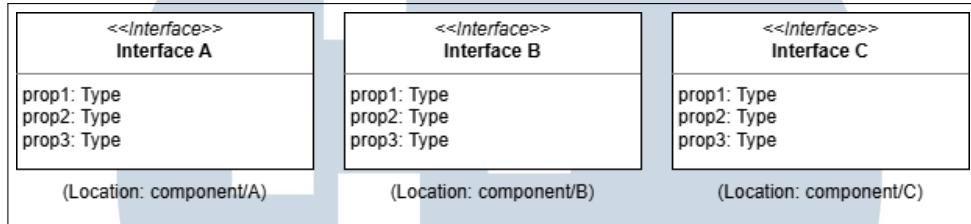
#### A User Requirement Refaktor Kode

Refaktor kode dilakukan untuk memenuhi kebutuhan pengguna dan pengembang terhadap sistem yang mudah dipelihara, konsisten, dan siap dikembangkan dalam jangka panjang. Dari sisi pengguna, sistem diharapkan tetap berfungsi dengan stabil tanpa perubahan perilaku fungsional. Sementara itu, dari sisi pengembang, dibutuhkan struktur kode yang lebih bersih, minim duplikasi, serta mudah dipahami agar proses pengembangan fitur baru dan perbaikan kesalahan dapat dilakukan secara efisien.

Kebutuhan ini mencakup penyatuan definisi tipe data agar tidak terjadi ketidaksinkronan, serta pengurangan pengulangan logika program yang berpotensi menimbulkan inkonsistensi implementasi. Dengan demikian, refaktor diarahkan untuk meningkatkan kualitas internal perangkat lunak tanpa memengaruhi keluaran yang dirasakan oleh pengguna akhir.

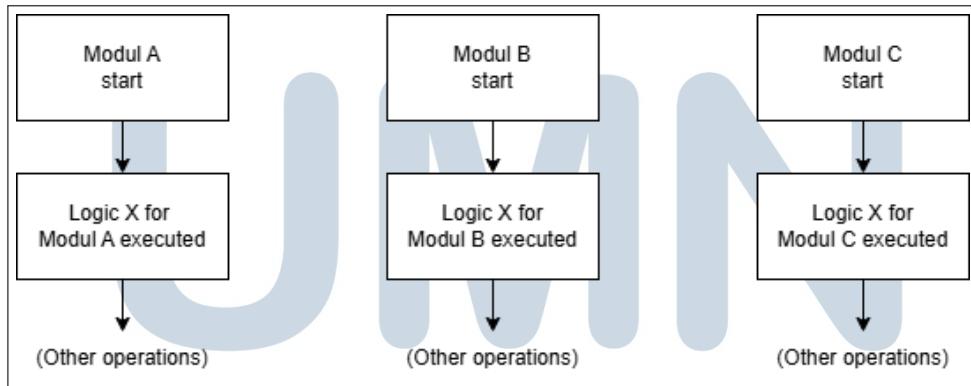
## B Perancangan Refaktor Kode

Tahap perancangan refaktor difokuskan pada identifikasi bagian kode yang memiliki struktur berulang dan logika yang sama. Salah satu temuan utama adalah tersebarnya *type interface* yang memiliki fungsi serupa namun didefinisikan secara terpisah di berbagai berkas. Kondisi awal ini digambarkan pada Gambar 3.30.



Gambar 3.30. Struktur interface sebelum konsolidasi

Selain duplikasi tipe data, ditemukan pula alur logika yang sama digunakan berulang kali pada beberapa fungsi. Pada tahap perancangan, alur ini dianalisis dan divisualisasikan dalam bentuk flowchart untuk menentukan bagian logika yang dapat diabstraksi ke dalam fungsi terpisah. Visualisasi alur sebelum perancangan ulang ditunjukkan pada Gambar 3.31.

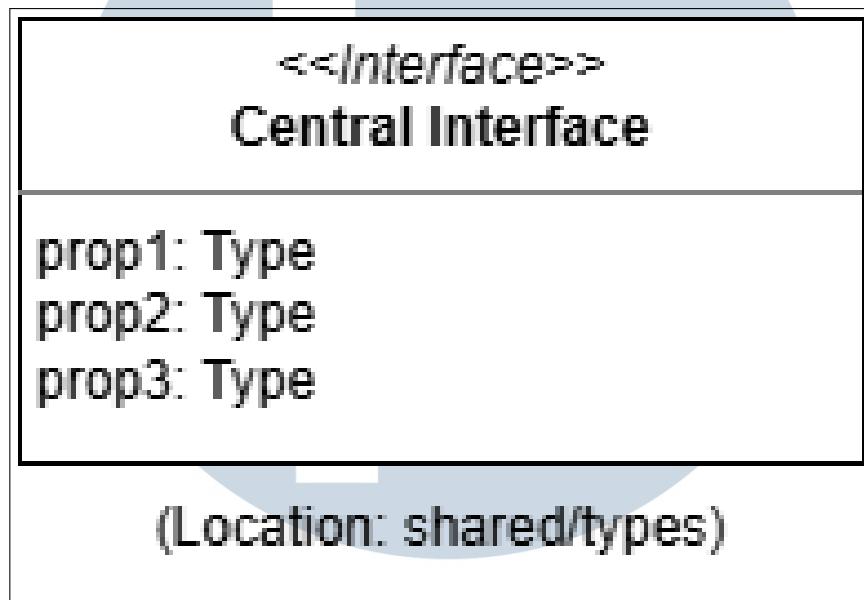


Gambar 3.31. Alur kode sebelum fungsi diekstraksi

Hasil perancangan menetapkan dua pendekatan utama, yaitu konsolidasi tipe data ke dalam satu lokasi terpusat sebagai *single source of truth* dan ekstraksi logika umum ke dalam fungsi modular. Pendekatan ini dipilih untuk menjaga konsistensi struktur sekaligus meningkatkan keterbacaan dan skalabilitas sistem.

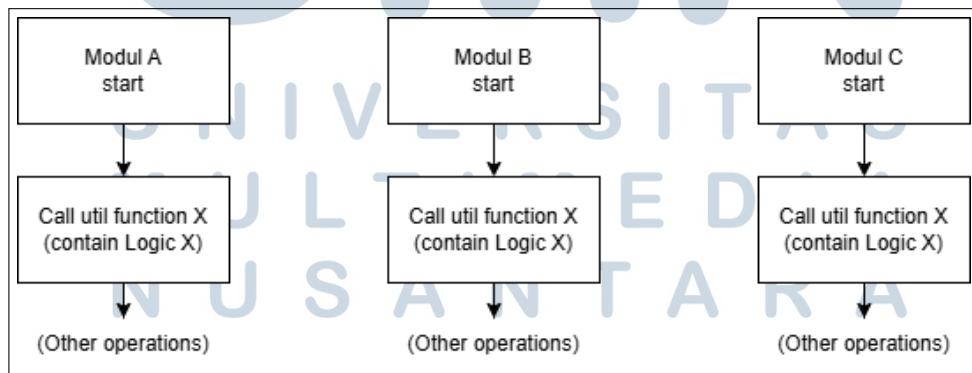
## C Implementasi Refaktor Kode

Implementasi refaktor dimulai dengan penghapusan kode duplikat pada definisi *type interface*. Seluruh tipe data yang memiliki kesamaan digabungkan ke dalam satu berkas terpusat. Hasil implementasi ini ditunjukkan pada Gambar 3.32, yang memperlihatkan struktur kode yang lebih ringkas dan konsisten.



Gambar 3.32. Struktur interface setelah konsolidasi

Selanjutnya, dilakukan ekstraksi fungsi dengan memindahkan logika yang berulang ke dalam satu fungsi terpisah yang dapat digunakan kembali. Implementasi setelah refaktor diperlihatkan pada Gambar 3.33, di mana alur kode menjadi lebih sederhana dan modular.



Gambar 3.33. Alur kode setelah fungsi diekstraksi

Seluruh proses implementasi refaktor dilakukan tanpa mengubah perilaku fungsional sistem. Hasilnya menunjukkan peningkatan keterbacaan kode, penurunan jumlah baris kode yang berulang, serta kemudahan dalam proses pemeliharaan. Struktur modular yang dihasilkan juga memberikan dasar yang lebih kuat untuk pengembangan lanjutan dan integrasi fitur baru di masa mendatang.

Dengan demikian, refaktor kode tidak hanya memenuhi kebutuhan teknis pengembang, tetapi juga mendukung keberlanjutan kualitas sistem sesuai dengan standar pengembangan perangkat lunak profesional.

### 3.3.8 Penyelesaian Merge Conflicts

#### A User Requirement Penyelesaian Merge Conflicts

Dalam pengembangan perangkat lunak berbasis kolaborasi, sistem pengelolaan versi harus mampu mengintegrasikan kontribusi dari beberapa pengembang tanpa mengganggu stabilitas aplikasi. Oleh karena itu, kebutuhan pengguna (*user requirement*) pada tahap ini adalah tersedianya mekanisme yang memungkinkan penggabungan kode dari berbagai kontributor secara konsisten, aman, dan mudah ditelusuri ketika terjadi *merge conflict*.

Pengguna sistem, dalam hal ini tim pengembang, membutuhkan proses identifikasi konflik yang jelas, kemampuan untuk memahami perbedaan perubahan kode, serta jaminan bahwa hasil akhir penggabungan tetap memenuhi kebutuhan fungsional aplikasi. Selain itu, sistem diharapkan mendukung kolaborasi melalui riwayat perubahan yang transparan serta meminimalkan risiko kesalahan akibat integrasi kode yang tidak terkontrol.

#### B Perancangan Penyelesaian Merge Conflicts

Tahap perancangan difokuskan pada penyusunan alur penyelesaian *merge conflict* yang sistematis. Proses dimulai dari penarikan perubahan terbaru (*pull*) dari repositori utama, dilanjutkan dengan identifikasi konflik menggunakan tampilan *diff*. Perancangan ini juga mencakup penggunaan alat bantu seperti *merge editor* pada Visual Studio Code dan riwayat komit GitHub untuk membantu pengembang memahami konteks perubahan. Dengan alur yang terstruktur, proses pengambilan keputusan menjadi lebih terarah dan mengurangi potensi kesalahan integrasi.

## C Implementasi Penyelesaian Merge Conflicts

Pada tahap implementasi, hasil perancangan diterapkan langsung pada basis kode. Penyelesaian konflik dilakukan dengan mengombinasikan perubahan yang relevan dari setiap kontributor, sekaligus melakukan refaktorisasi kode. Refaktorisasi ini mencakup penghapusan kode duplikat serta mengekstraksi bagian logika yang berulang menjadi fungsi atau metode terpisah agar kode lebih modular dan mudah dipelihara.

Setelah penggabungan dan refaktorisasi selesai, dilakukan pengujian untuk memastikan tidak terjadi regresi. Pengujian meliputi proses kompilasi, eksekusi aplikasi, serta verifikasi fungsi-fungsi yang terdampak konflik. Tahap ini memastikan bahwa hasil implementasi tidak hanya menyelesaikan konflik, tetapi juga meningkatkan kualitas struktur kode secara keseluruhan serta menjaga stabilitas sistem.

### 3.4 Kendala dan Solusi yang Ditemukan

Selama pelaksanaan kegiatan magang di PT Kalbe Farma Tbk, terdapat sejumlah kendala yang dihadapi baik dalam aspek teknis maupun nonteknis. Kendala-kendala tersebut menjadi bagian dari proses pembelajaran dalam memahami dinamika kerja di lingkungan industri berbasis teknologi informasi. Melalui pendekatan kolaboratif dan konsultasi rutin dengan pembimbing teknis, setiap kendala yang muncul berhasil ditangani secara bertahap dengan penerapan solusi yang tepat dan sistematis.

Salah satu tantangan yang muncul adalah ketidaksesuaian logika program antara modul yang dikembangkan lebih dahulu oleh anggota tim berbeda dengan permintaan pengguna, yang menyebabkan ketidakselarasan pada alur kerja aplikasi. Situasi ini berpotensi memengaruhi kestabilan sistem pada tahap pengujian awal. Untuk mengatasinya, dilakukan peninjauan ulang terhadap struktur logika dan mekanisme komunikasi antar-modul, diikuti dengan penerapan *code review* secara berkala. Pendekatan tersebut berhasil menyelaraskan integrasi sistem sekaligus meningkatkan konsistensi standar pengembangan di dalam tim.

Kendala teknis lain yang dihadapi berkaitan dengan sinkronisasi data pada fitur manajemen sesi pengguna, khususnya pada proses *idle timer* lintas tab peramban. Sistem pada versi awal belum mampu mempertahankan status aktivitas pengguna secara konsisten, sehingga sesi berakhir sebelum waktunya.

Solusi yang diterapkan meliputi penerapan sinkronisasi berbasis penyimpanan lokal (*local storage*) dan komunikasi antar-tab secara real-time, sehingga sistem dapat mendeteksi aktivitas pengguna dengan lebih akurat serta mempertahankan stabilitas sesi secara keseluruhan.

Selain itu, pemahaman terhadap alur bisnis dan kebutuhan pengguna dari sisi regulasi industri farmasi menjadi tantangan tersendiri. Kompleksitas dokumen dan persyaratan kepatuhan mengharuskan adaptasi yang cepat terhadap istilah teknis dan kebijakan perusahaan. Solusi yang dilakukan adalah melakukan konsultasi rutin dengan pembimbing dan pengguna terkait, serta meninjau dokumentasi proyek dan pedoman operasional perusahaan untuk memastikan kesesuaian rancangan sistem dengan kebutuhan aktual di lapangan.

Dalam proses pengembangan fitur secara paralel, kendala berupa konflik penggabungan (*merge conflicts*) juga sering kali muncul. Konflik sering terjadi pada file yang memiliki tingkat keterhubungan tinggi, seperti file utilitas bersama, konfigurasi global, atau komponen inti frontend, sehingga sistem kendali versi (Git) tidak dapat menentukan perubahan mana yang harus diprioritaskan. Untuk mengatasi kendala tersebut, dilakukan beberapa langkah sistematis seperti menggunakan pola *feature branching*, sehingga setiap fitur dikembangkan pada cabang terpisah dan hanya digabungkan setelah melalui proses validasi dan melakukan sesi penyelesaian bersama antara anggota tim dan pembimbing teknis. Pendekatan kolaboratif ini membantu memastikan bahwa keputusan penggabungan tidak menimbulkan regresi terhadap fitur yang sudah ada.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA