

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Pelaksanaan kegiatan magang di PT Kalbe Farma Tbk dilaksanakan pada departemen CDT dengan penempatan di Software Center pada posisi Software Engineer. Posisi ini memiliki tanggung jawab dalam merancang dan mengembangkan aplikasi yang digunakan untuk mendukung kebutuhan operasional internal perusahaan. Selama masa pelaksanaan magang, pengawasan dan bimbingan dilakukan oleh Bapak Russell Otniel selaku Software Engineer Officer di Software Center. Bimbingan tersebut bertujuan untuk memastikan seluruh aktivitas pengembangan aplikasi berjalan secara terarah, efektif, serta sesuai dengan standar dan prosedur yang telah ditetapkan oleh perusahaan.

Proses pembuatan aplikasi dilakukan melalui koordinasi dan diskusi bersama pihak pengguna dari departemen terkait untuk memastikan solusi yang dihasilkan sesuai dengan kebutuhan dan standar operasional yang berlaku. Kegiatan koordinasi dan pembahasan teknis proyek dilaksanakan secara daring melalui Microsoft Teams, sedangkan komunikasi sehari-hari dan penyampaian informasi singkat dilakukan melalui WhatsApp Group guna memperlancar proses kolaborasi. Selain itu, setiap hari Selasa pagi diadakan sesi *progress meeting* yang dihadiri oleh seluruh anggota tim pengembang. Pertemuan tersebut berfungsi untuk meninjau kemajuan proyek, mendiskusikan kendala yang dihadapi, serta menyampaikan pembaruan kepada pihak pengguna agar pengembangan RIMS tetap berjalan selaras dengan kebutuhan pengguna.

3.2 Tugas yang Dilakukan

Selama periode kedua kegiatan magang di PT Kalbe Farma Tbk, tanggung jawab yang dijalankan mencakup pembuatan dan integrasi aplikasi internal perusahaan yang mendukung kebutuhan operasional. Proses ini dimulai dengan riset fitur menggunakan pendekatan PoC dari periode pertama, kemudian dilanjutkan dengan pengembangan dan integrasi aplikasi, serta diakhiri dengan pengujian dan uji coba bersama pihak pengguna untuk memastikan fungsionalitas berjalan sesuai kebutuhan. Berikut merupakan rangkaian aktivitas yang dilaksanakan selama periode magang.

1. Membuat modul *Submission Management* pada RIMS

Modul ini berfungsi untuk memfasilitasi proses pembuatan dan pengelolaan *submission dossier* berupa *dossier* sesuai kebutuhan registrasi. Fitur-fitur utama yang dikembangkan mencakup menampilkan data *dossier*, penerapan *advanced filter* pada data *dossier*, serta pembuatan dan edit *dossier*.

2. Mengimplementasikan fitur *version history* pada modul *Document Editor*

Fitur ini dikembangkan untuk mencatat setiap perubahan yang terjadi pada dokumen secara otomatis dalam sistem. Setiap kali pengguna melakukan perubahan, sistem akan menyimpan versi baru dari dokumen beserta informasi pengguna, waktu perubahan, dan isi revisi.

3. Mengintegrasikan modul *Document Editor* pada RIMS

Fitur-fitur yang diintegrasikan dengan sistem pada bagian ini mencakup pengambilan dan pengolahan dokumen, *collaborative editing*, serta penyimpanan otomatis dokumen ke *database*.

3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

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

Minggu Ke -	Pekerjaan yang dilakukan
1	Melakukan PoC fitur <i>version history</i> pada <i>Document Editor</i> .
2	Melanjutkan PoC fitur <i>version history</i> pada <i>Document Editor</i> , serta <i>progress meeting</i> bersama internal tim pengembang dan pengguna.
3	Melanjutkan PoC fitur <i>version history</i> pada <i>Document Editor</i> .
4	Melanjutkan PoC fitur <i>version history</i> pada <i>Document Editor</i> , serta mengikuti <i>progress meeting</i> bersama internal tim pengembang dan pengguna.
5	Melanjutkan PoC fitur <i>version history</i> pada <i>Document Editor</i> , mengikuti <i>progress meeting</i> bersama tim pengembang internal dan pengguna, serta mempelajari dan melakukan eksplorasi terhadap <i>template starter project RIMS</i> .
Lanjut pada halaman berikutnya	

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

Minggu Ke -	Pekerjaan yang dilakukan
6	Mengerjakan antarmuka pada modul <i>Submission Management</i> , serta mengikuti <i>progress meeting</i> bersama internal tim pengembang dan pengguna.
7	Melanjutkan pengerjaan antarmuka pada modul <i>Submission Management</i> , membuat fitur penyimpanan otomatis untuk <i>draft</i> , serta mengikuti <i>progress meeting</i> bersama internal tim pengembang dan pengguna.
8	Melakukan revisi pada antarmuka modul <i>Submission Management</i> , mengimplementasikan fitur <i>reset form</i> pada formulir pembuatan <i>dossier</i> , menambahkan fitur <i>filter user</i> dan <i>business unit</i> pada formulir pembuatan <i>dossier</i> , serta mengikuti <i>progress meeting</i> bersama tim pengembang internal dan pengguna.
9	Mengerjakan fitur pemilihan dokumen yang berkaitan pada formulir pembuatan <i>dossier</i> , serta mengikuti <i>progress meeting</i> bersama tim pengembang internal dan pengguna.
10	Melakukan perbaikan <i>bug</i> baik antarmuka maupun logika pada modul <i>Submission Management</i> , serta mengikuti <i>progress meeting</i> bersama tim pengembang internal dan pengguna.
11	Melakukan perbaikan <i>bug</i> baik antarmuka maupun logika pada modul <i>Submission Management</i> , serta mengikuti <i>progress meeting</i> bersama tim pengembang internal.
12	Memperbaiki <i>bug</i> pada logika formulir pembuatan <i>dossier</i> , menambahkan validasi terhadap formulir <i>dossier</i> , serta mengikuti <i>progress meeting</i> bersama tim pengembang internal dan pengguna.
13	Memperbaiki <i>bug</i> pada formulir pembuatan <i>dossier</i> , <i>refactor</i> CSS pada modul <i>Submission Management</i> , serta mengikuti <i>progress meeting</i> bersama tim pengembang internal dan pengguna.
Lanjut pada halaman berikutnya	

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

Minggu Ke -	Pekerjaan yang dilakukan
14	Menambahkan validasi pada formulir pembuatan <i>dossier</i> , memperbaiki <i>bug</i> pada modul <i>Submission Management</i> , serta mengikuti <i>progress meeting</i> bersama tim pengembang internal.
15	Mengerjakan fitur <i>advanced filter</i> pada data <i>dossier</i> , serta mengikuti <i>progress meeting</i> bersama tim pengembang internal dan pengguna secara luring.
16	Melakukan revisi terhadap antarmuka <i>advanced filter</i> , serta mengikuti <i>progress meeting</i> bersama tim pengembang internal dan pengguna.
17	Mengerjakan logika fitur <i>advanced filter</i> , serta mengikuti <i>progress meeting</i> bersama tim pengembang internal.
18	Memperbaiki <i>error</i> pada formulir pembuatan <i>dossier</i> , serta mengikuti <i>progress meeting</i> bersama tim pengembang internal dan pengguna.
19	Mengimplementasikan hasil PoC <i>Document Editor</i> ke RIMS, serta mengikuti <i>progress meeting</i> bersama tim pengembang internal dan pengguna.

3.3.1 Pembuatan Modul Submission Management

A User Requirement

Modul *Submission Management* berfungsi sebagai pusat pengelolaan proses pengajuan dokumen regulasi. Modul ini dirancang untuk memfasilitasi penyusunan, pengelolaan, serta pelacakan dokumen registrasi produk secara terstruktur dan sesuai dengan standar regulator. Melalui modul ini, pengguna dapat membuat *submission dossier* berupa *dossier* utama berdasarkan kebutuhan registrasi produk. Berikut adalah beberapa fitur yang terdapat pada modul *Submission Management*.

A.1 Menampilkan Data Dossier

Fitur ini berfungsi untuk menampilkan seluruh daftar *dossier* utama yang telah dibuat dalam bentuk tabel. Setiap entri menampilkan informasi dasar yang diperlukan untuk memudahkan identifikasi dan pemantauan. Tampilan daftar ini

memudahkan pengguna untuk melihat dan mengelola *dossier* yang telah dibuat secara keseluruhan.

A.2 Advanced Filter

Fitur ini memungkinkan pengguna untuk melakukan *filter* data *dossier* utama berdasarkan kriteria tertentu secara lebih spesifik. Pengguna dapat mem-*filter* daftar *dossier* menggunakan parameter yang telah disesuaikan. Dengan adanya *advanced filter* ini, pengguna dapat menemukan data yang relevan dengan cepat tanpa harus menelusuri seluruh daftar *dossier* secara manual.

Fitur *advanced filter* juga mendukung pengaturan logika *filter* antar kolom dengan menggunakan operator *AND* atau *OR*. Melalui mekanisme ini, pengguna dapat mengombinasikan berbagai kolom dalam satu proses *filter*, seperti mencari data yang memenuhi seluruh kondisi yang ditentukan (*AND*), atau data yang sesuai dengan salah satu dari beberapa kondisi yang ada (*OR*). Pengguna dapat menyesuaikan operator logika sesuai kebutuhan untuk mendapatkan hasil pencarian yang lebih presisi.

Sistem juga menyediakan kemampuan untuk menyimpan konfigurasi *advanced filter* yang telah dibuat agar dapat digunakan kembali di kemudian hari. Dengan adanya fitur *advanced filter*, pengguna dapat mempercepat proses pencarian, serta meningkatkan akurasi hasil pencarian data *dossier*.

A.3 Pembuatan Dossier

Fitur ini digunakan untuk membuat *dossier* baru melalui formulir yang dirancang secara terstruktur untuk memandu pengguna dalam mengisi informasi yang diperlukan. Proses pengisian dilakukan secara berurutan, dengan setiap bagian dilengkapi mekanisme validasi agar data yang dimasukkan tetap konsisten dan lengkap. Informasi yang dikumpulkan mencakup aspek-aspek inti yang dibutuhkan untuk membentuk sebuah *dossier*, mulai dari data administratif, informasi terkait dokumen pendukung, hingga penetapan pengguna yang berperan dalam proses pengelolaan dan persetujuan dokumen. Sistem juga menyediakan fitur penyimpanan *draft*, sehingga pengguna dapat menghentikan sementara proses pembuatan dan melanjutkannya kembali tanpa mengulang dari awal. Setelah seluruh data dinyatakan lengkap, sistem akan membuat *dossier* baru dan ditampilkan pada daftar *dossier*.

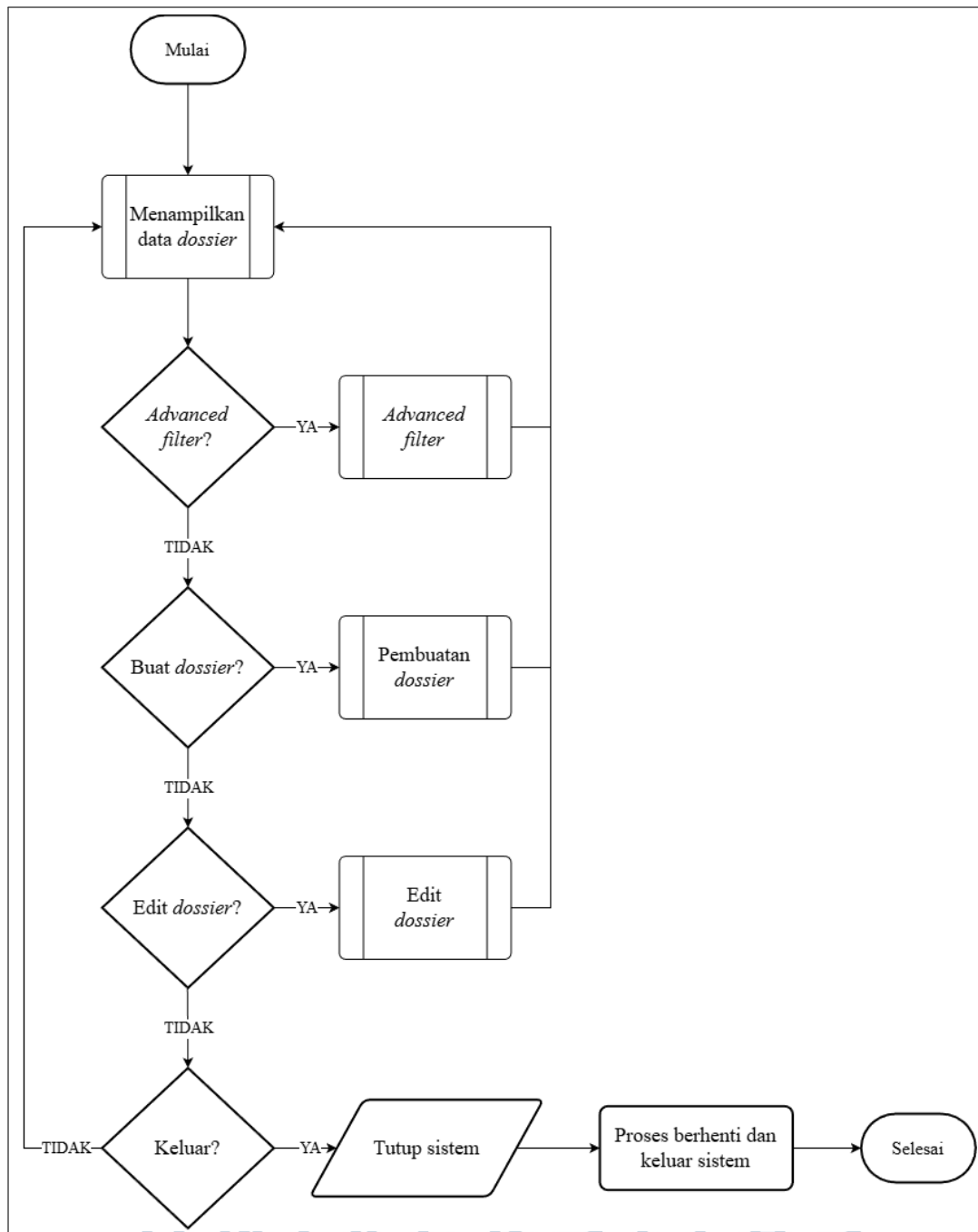
A.4 Edit Dossier

Fitur ini memungkinkan pengguna untuk memperbarui informasi yang telah dimasukkan pada tahap pembuatan *dossier*. Proses pengeditan hanya dapat dilakukan pada *dossier* dengan status *Draft*, sehingga perubahan tidak dapat dilakukan pada data yang telah difinalisasi. Fitur ini dirancang untuk memberikan fleksibilitas kepada pengguna dalam melakukan revisi atau penyesuaian terhadap data dan dokumen sebelum *dossier* di-submit secara resmi.

B Flowchart Modul Submission Management

Pada modul *Submission Management*, alur kerja keseluruhan dapat dilihat pada Gambar 3.1, yang memperlihatkan urutan proses utama dalam pengelolaan *dossier*. Proses dimulai dengan sub-proses fitur menampilkan data *dossier* yang digambarkan pada Gambar 3.2. Setelah data ditampilkan, pengguna dapat menentukan tindakan lanjutan sesuai kebutuhan. Apabila pengguna ingin memberikan *filter* terhadap data yang muncul, sistem menyediakan sub-proses fitur *advanced filter* yang diperlihatkan pada Gambar 3.3. Selanjutnya, pengguna dapat membuat *dossier* baru melalui sub-proses fitur pembuatan *dossier* yang ditunjukkan pada Gambar 3.5. Apabila diperlukan pembaruan terhadap *dossier* yang telah ada, pengguna dapat memanfaatkan sub-proses fitur edit *dossier* sebagaimana divisualisasikan pada Gambar 3.8. Apabila pengguna memilih untuk keluar dari sistem, proses akan dilanjutkan dengan penutupan sistem, kemudian sistem akan menghentikan seluruh aktivitas yang sedang berjalan sebelum keluar dari modul. Secara keseluruhan, alur tersebut menggambarkan hubungan antarproses utama dalam modul *Submission Management* yang mendukung penyusunan, pembaruan, dan pengelolaan *dossier* registrasi produk secara terstruktur.

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

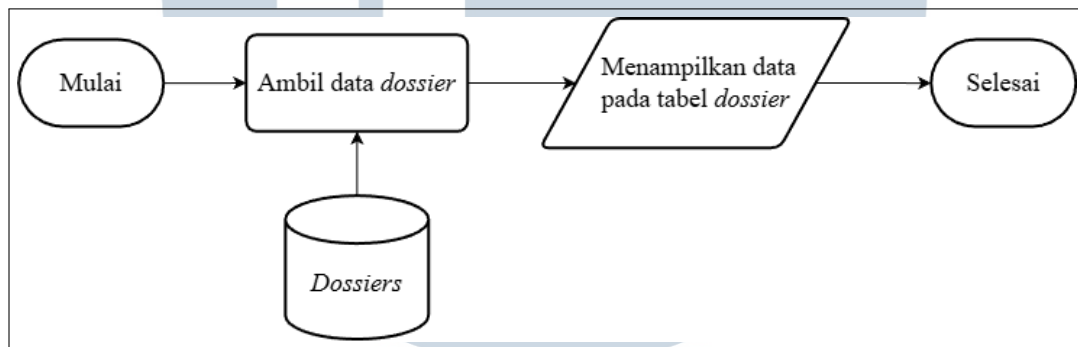


Gambar 3.1. *Flowchart modul Submission Management*

Berikut merupakan penjelasan sub-proses dari setiap fitur yang terdapat pada modul *Submission Management*.

B.1 Menampilkan Data Dossier

Sub-proses fitur menampilkan data *dossier* berfungsi untuk menampilkan daftar *dossier* yang tersimpan di dalam *database* dan menyajikannya dalam bentuk tabel pada antarmuka pengguna. Gambar 3.2 memperlihatkan alur proses dari fitur ini, yang dimulai dengan pengambilan data *dossier* dari tabel *Dossiers* pada *database*. Setelah data berhasil diambil, sistem akan memproses informasi tersebut agar sesuai dengan format tampilan tabel yang ditampilkan kepada pengguna. Melalui tampilan ini, pengguna dapat dengan mudah memantau, meninjau, dan mengelola seluruh data *dossier* yang tersedia di dalam sistem.



Gambar 3.2. Flowchart sub-proses fitur menampilkan data *dossier*

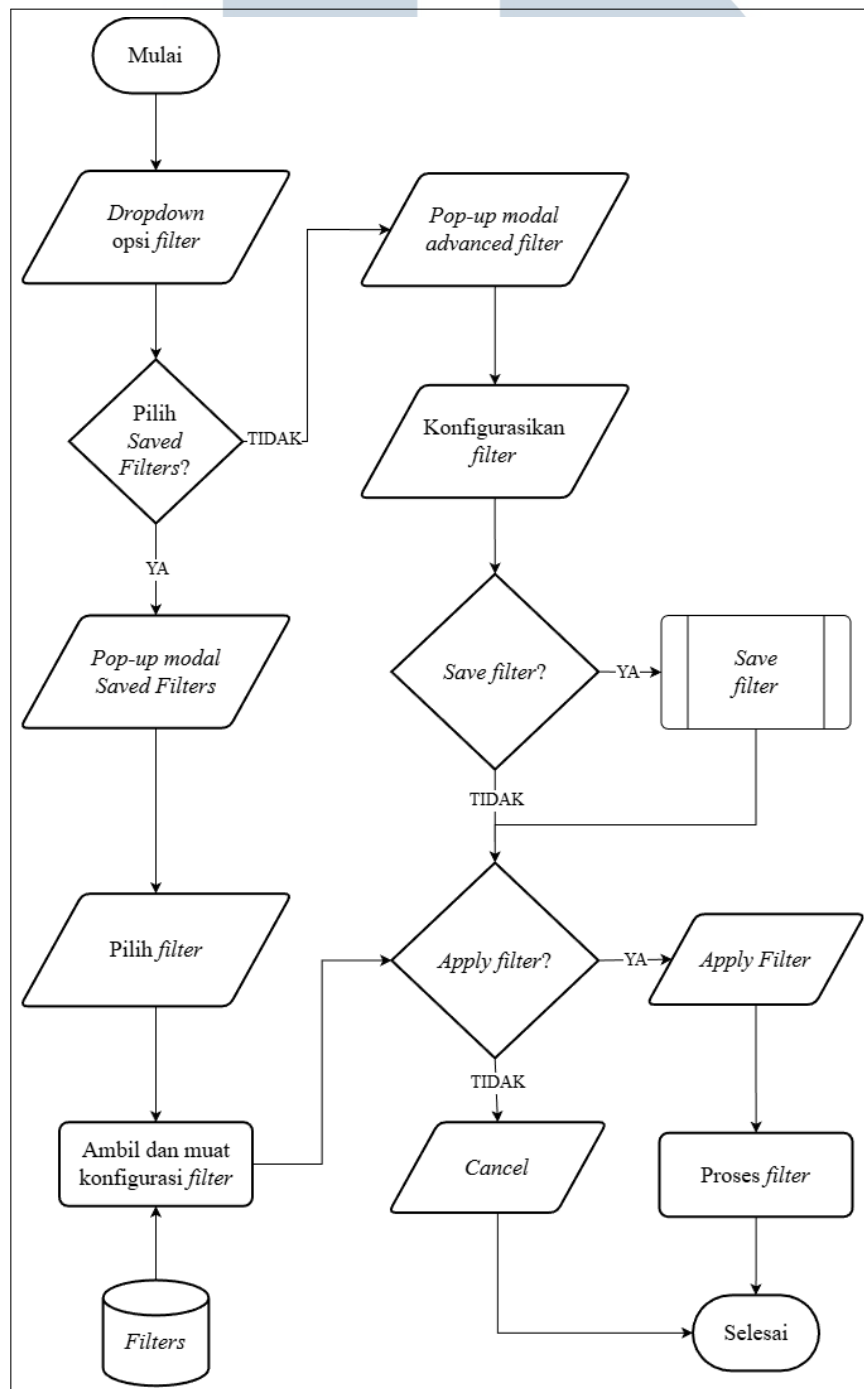
B.2 Advanced Filter

Sub-proses fitur *advanced filter* digunakan untuk melakukan *filter* data *dossier* berdasarkan kriteria tertentu yang ditentukan oleh pengguna. Gambar 3.3 memperlihatkan alur proses dari fitur ini, yang mencakup penggunaan *filter* yang telah disimpan sebelumnya atau pembuatan *filter* baru melalui formulir *advanced filter*. Proses dimulai ketika sistem menampilkan opsi *filter* dalam bentuk *dropdown*. Setelah itu, pengguna dapat memilih untuk menggunakan *Saved Filters* yang sudah tersedia atau membuat konfigurasi *filter* baru.

Jika pengguna memilih opsi *Saved Filters*, sistem akan menampilkan *pop-up modal* berisi daftar *filter* yang sebelumnya telah disimpan. Setelah salah satu *filter* dipilih, sistem akan memuat konfigurasi tersebut dari tabel *Filters* pada *database*. Pengguna dapat menekan *Apply Filter* untuk diproses *filter*-nya atau memilih *Cancel* apabila tidak ingin melanjutkan.

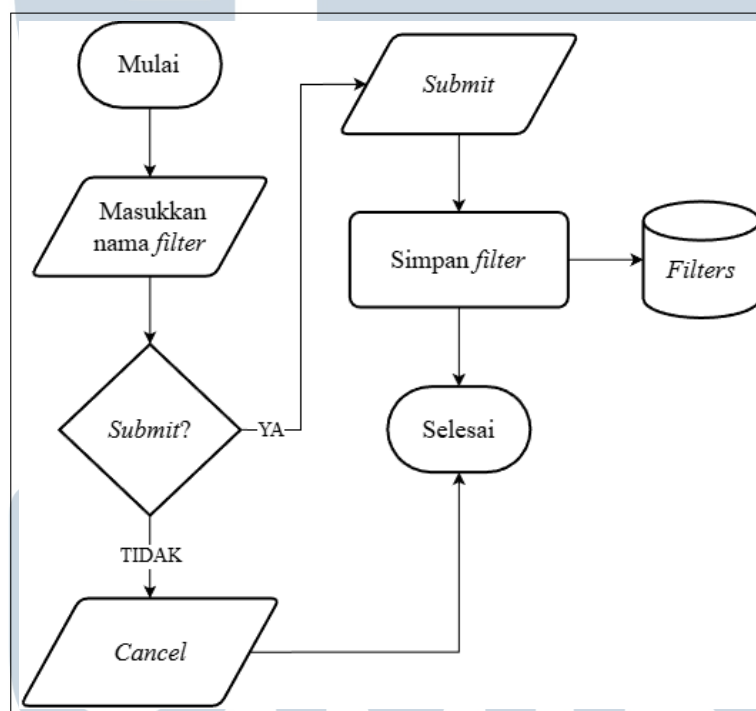
Sementara itu, apabila pengguna memilih untuk membuat *filter* baru, sistem akan menampilkan *pop-up modal advanced filter* yang memungkinkan pengguna

mengonfigurasi *filter* sesuai kebutuhan. Setelah pengaturan selesai dilakukan, pengguna dapat menyimpan konfigurasi *filter* atau menekan *Apply Filter* untuk memproses *filter* data pada tabel *dossier*, atau memilih *Cancel* jika tidak ingin menerapkannya.



Gambar 3.3. Flowchart sub-proses fitur *advanced filter*

Apabila konfigurasi *filter* yang telah dibuat ingin disimpan untuk digunakan kembali di kemudian hari, pengguna dapat memanfaatkan sub-proses *Save Filter* sebagaimana diperlihatkan pada Gambar 3.4. Sub-proses ini berfungsi untuk menyimpan konfigurasi *filter* yang telah disusun oleh pengguna agar dapat dimuat ulang tanpa perlu membuat konfigurasi baru secara manual. Proses dimulai dari pengguna memasukkan nama konfigurasi *filter*. Jika pengguna ingin menyimpannya, pengguna dapat menekan *Submit* untuk mengonfirmasi. Sistem kemudian memproses dan menyimpan konfigurasi tersebut ke dalam tabel *Filters* pada *database*. Sebaliknya, apabila pengguna tidak ingin menyimpan konfigurasi tersebut, pengguna dapat menekan tombol *Cancel*.

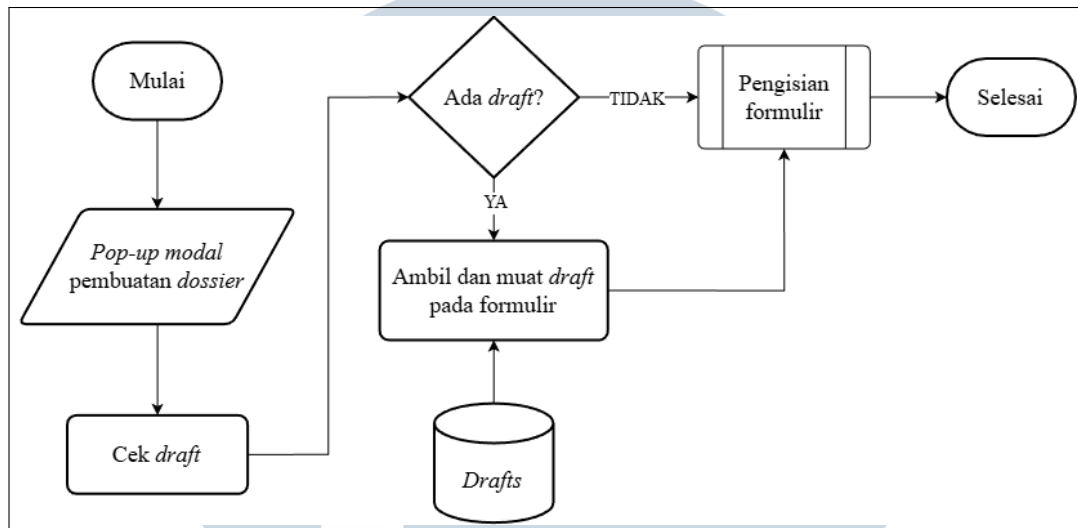


Gambar 3.4. Flowchart sub-proses fitur save filter

B.3 Pembuatan Dossier

Sub-proses fitur pembuatan *dossier* digunakan untuk membuat dan menyimpan data *dossier* baru ke dalam sistem. Gambar 3.5 memperlihatkan alur tahap awal proses ini, yaitu pengambilan *draft* yang tersimpan di dalam tabel *Drafts* pada *database*. Proses dimulai dari sistem menampilkan *pop-up modal* pembuatan formulir dan memeriksa ketersediaan *draft* yang tersimpan. Jika ditemukan *draft*, sistem akan mengambil dan memuatnya agar pengguna dapat

melanjutkan pengisian data yang tertunda. Jika tidak, proses dilanjutkan ke tahap pengisian formulir baru.

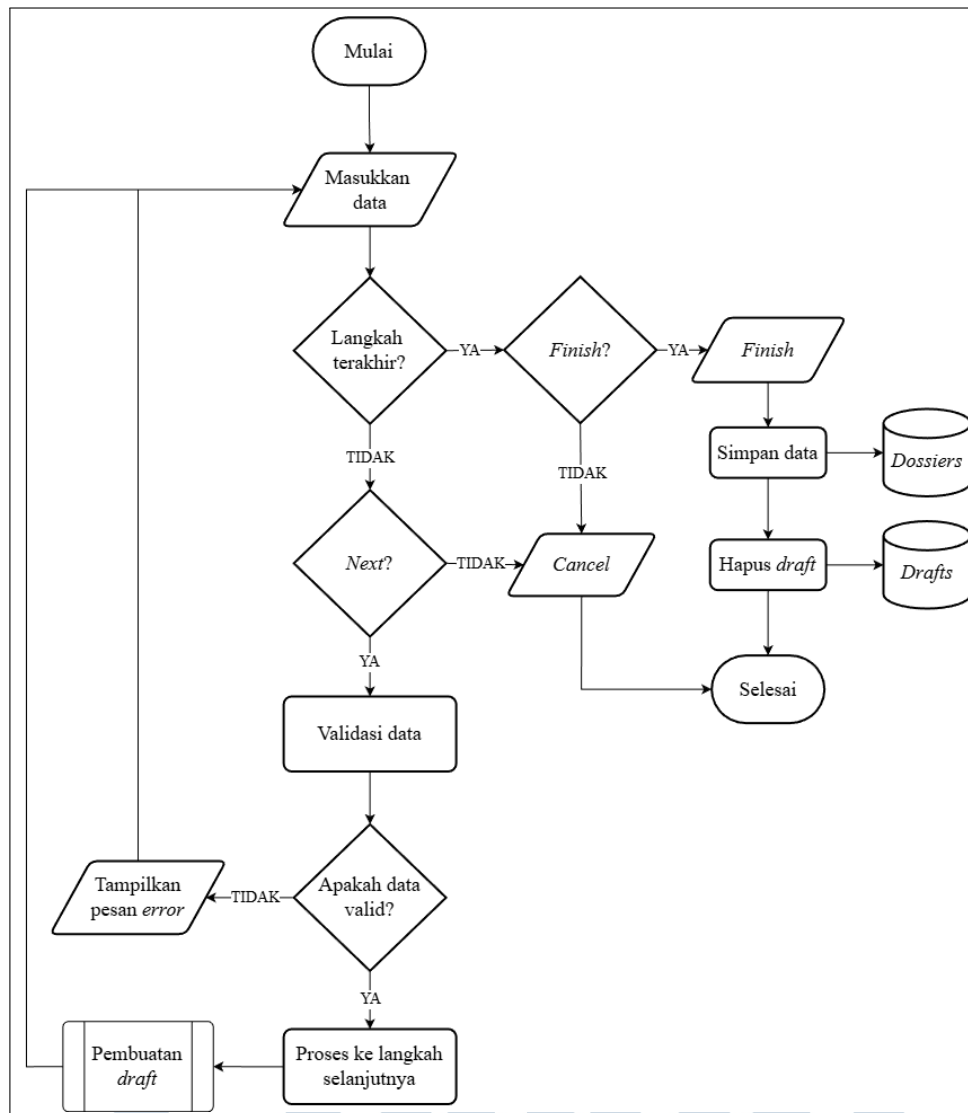


Gambar 3.5. Flowchart sub-proses fitur pengambilan draft

Pada tahap pengisian formulir, sebagaimana ditunjukkan pada Gambar 3.6, proses dimulai ketika pengguna memasukkan data ke dalam formulir. Selanjutnya, sistem akan memeriksa apakah pengguna berada pada langkah terakhir. Jika belum, pengguna dapat menekan *Next* untuk melanjutkan. Sistem kemudian melakukan validasi terhadap data yang diisi. Apabila data tidak valid, sistem akan menampilkan pesan *error*, sedangkan jika valid, sistem akan melanjutkan ke langkah berikutnya dan menyimpan data sementara sebagai *draft*.

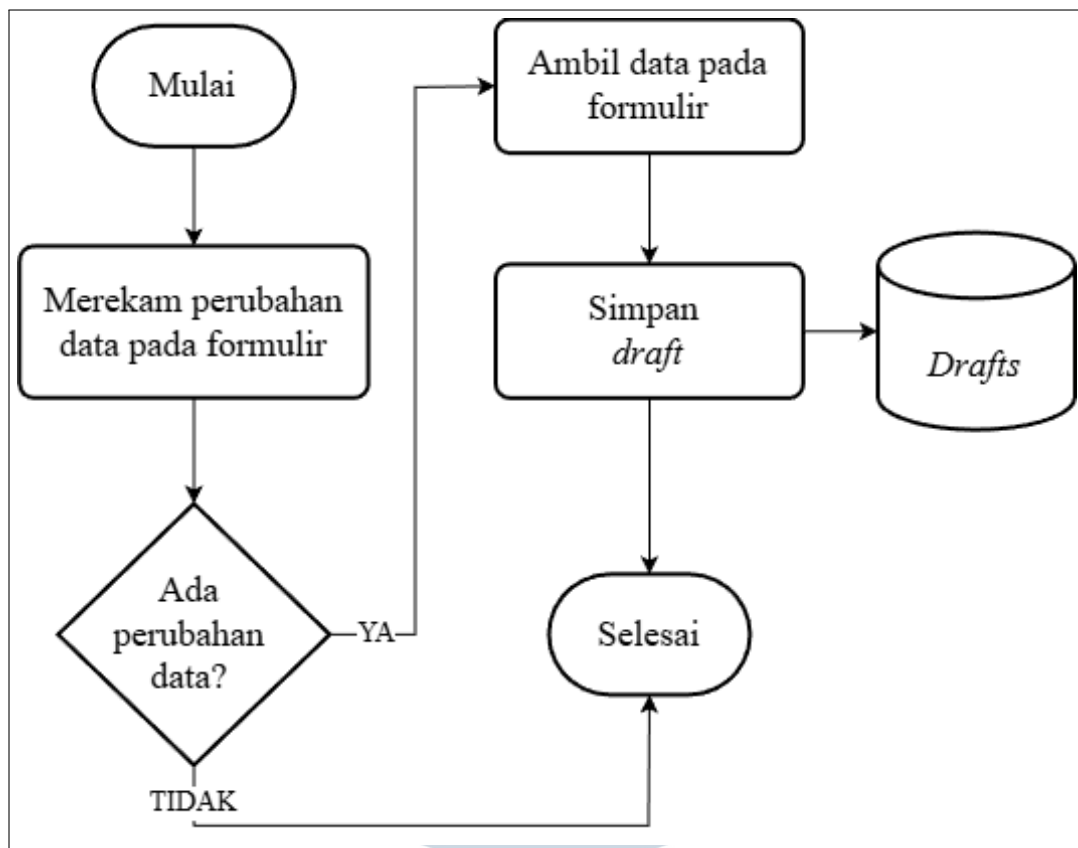
Apabila pengguna berada pada langkah terakhir dan memilih untuk menyelesaikan proses pembuatan, pengguna dapat menekan *Finish*. Sistem selanjutnya akan menyimpan seluruh data ke dalam tabel *Dossiers* serta menghapus seluruh *draft* yang tersisa pada tabel *Drafts* di dalam *database*. Sementara itu, apabila pengguna memutuskan untuk tidak melanjutkan proses, pengguna dapat menekan *Cancel* untuk membatalkan tahapan yang sedang berlangsung.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.6. Flowchart sub-proses fitur pengisian formulir

Selanjutnya, Gambar 3.7 menggambarkan sub-proses pembuatan *draft* yang dilakukan secara otomatis oleh sistem selama pengisian formulir berlangsung. Sistem akan merekam perubahan data pada formulir. Jika terdeteksi perubahan, sistem mengambil seluruh data dari formulir dan menyimpan *draft* ke dalam tabel *Drafts* pada *database*. Namun, jika tidak ada perubahan, sistem tidak melakukan penyimpanan ulang dan mempertahankan kondisi terakhir dari *draft* yang ada.



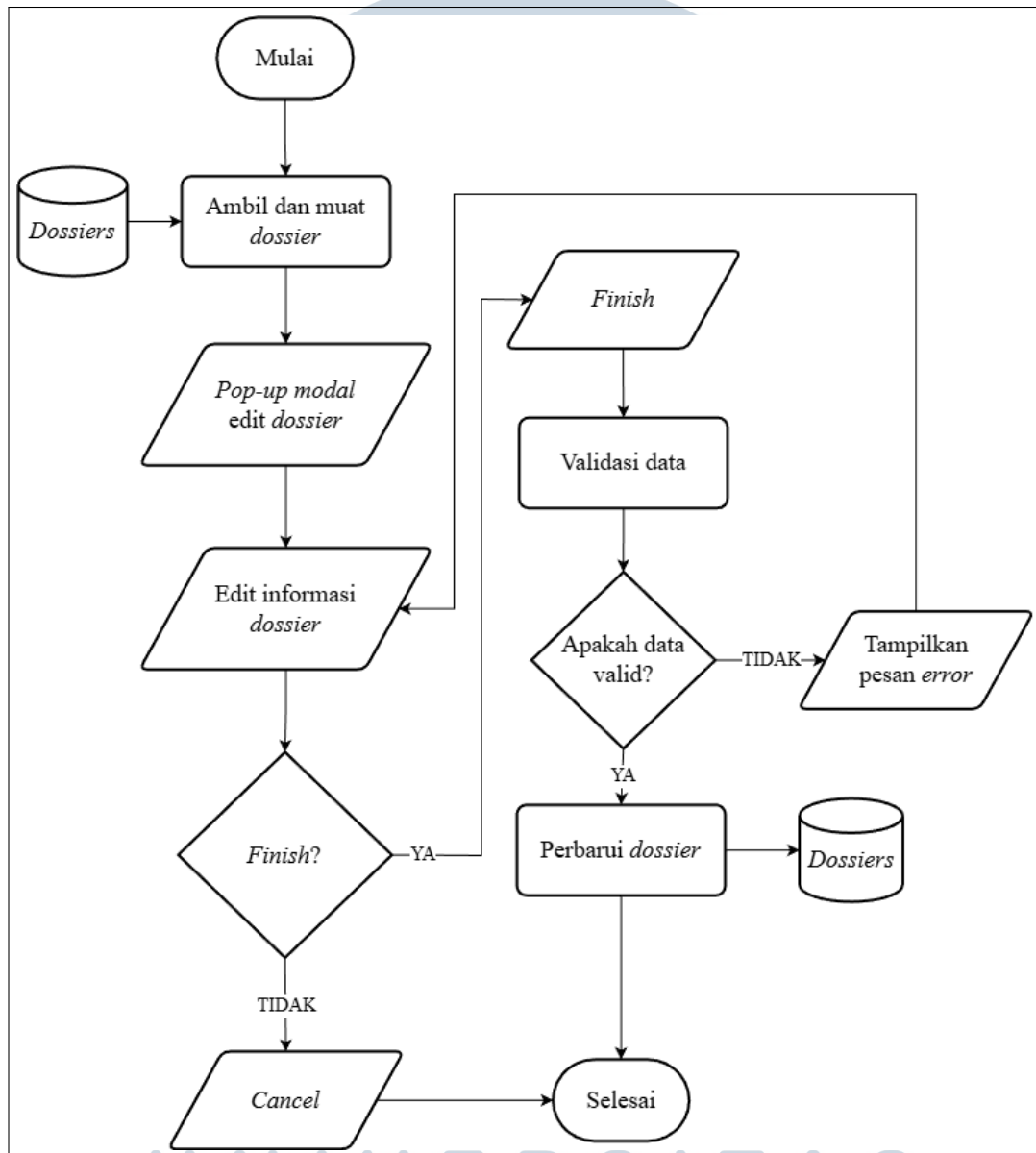
Gambar 3.7. Flowchart sub-proses fitur pembuatan *draft*

B.4 Edit Dossier

Sub-proses fitur pengeditan *dossier* digunakan untuk memperbarui informasi pada *dossier* yang telah dibuat sebelumnya. Gambar 3.8 memperlihatkan alur proses dari fitur ini, yang memungkinkan pengguna melakukan perubahan data pada *dossier* dengan status *Draft*. Proses dimulai ketika sistem mengambil dan memuat *dossier* yang akan diedit. Setelah itu, sistem menampilkan *pop-up modal* berisi formulir pengeditan yang telah terisi dengan data sebelumnya. Pada tahap ini, pengguna dapat melakukan penyesuaian informasi sesuai kebutuhan melalui kolom yang telah disediakan.

Setelah pengeditan selesai dan pengguna memilih untuk memperbarui *dossier*, pengguna dapat menekan tombol *Finish*. Sistem kemudian akan melakukan proses validasi terhadap data yang telah diperbarui. Jika ditemukan data yang tidak valid, sistem menampilkan pesan *error* untuk meminta pengguna melakukan perbaikan. Sebaliknya, apabila seluruh data dinyatakan valid, sistem akan memperbarui informasi *dossier* pada tabel *Dossiers* di dalam *database* dan

menyimpan perubahan tersebut. Jika pengguna memutuskan untuk membatalkan proses pengeditan, pengguna dapat menekan tombol *Cancel*.



Gambar 3.8. Flowchart sub-proses fitur edit dossier

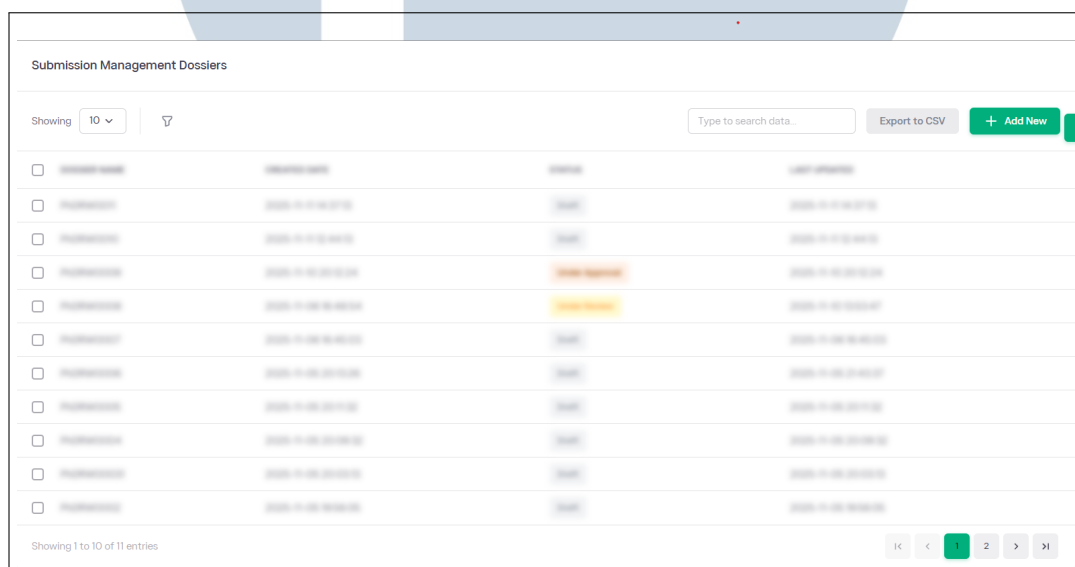
C Hasil Pembuatan Modul Submission Management

C.1 Menampilkan Data Dossier

Fitur menampilkan data *dossier* berfungsi untuk menampilkan seluruh daftar *dossier* yang telah dibuat oleh pengguna dalam bentuk tabel yang terstruktur.

Antarmuka pengguna dari fitur ini ditunjukkan pada Gambar 3.9. Pada gambar tersebut, setiap baris dalam tabel merepresentasikan satu entitas *dossier* yang memuat informasi penting. Tampilan ini membantu pengguna dalam memantau dan mengelola seluruh *dossier* yang berada di dalam sistem dengan mudah dan efisien.

Selain itu, antarmuka dilengkapi dengan berbagai elemen pendukung untuk meningkatkan fungsionalitas dan kemudahan penggunaan. Beberapa di antaranya meliputi fitur *pagination* yang mempermudah navigasi data dalam jumlah besar, kolom pencarian sederhana untuk *filter* data berdasarkan kata kunci tertentu, serta tombol *Add New* untuk menambahkan *dossier* baru ke dalam sistem. Terdapat juga tombol *Export to CSV* yang memungkinkan pengguna mengekspor data *dossier* dalam format berkas CSV. Keseluruhan rancangan ini bertujuan untuk mendukung proses pengelolaan *dossier* agar lebih efisien, terorganisir, dan mudah diakses.

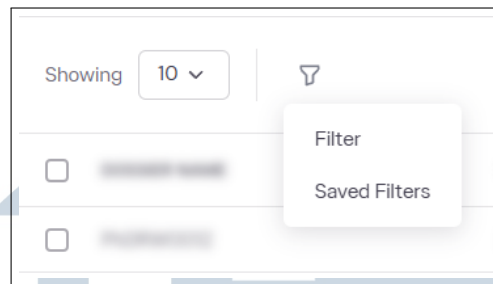


The screenshot shows a web interface titled "Submission Management Dossiers". At the top, there is a search bar with the placeholder text "Type to search data...", an "Export to CSV" button, and a green "+ Add New" button. Below the search bar, there is a table with four columns. The first column contains checkboxes for each row. The table displays 11 rows of data, with the first row highlighted in blue. At the bottom of the table, there is a pagination bar that says "Showing 1 to 10 of 11 entries" and a set of navigation buttons including "<", "<<", "1", "2", ">>", and ">".

Gambar 3.9. Antarmuka fitur menampilkan data *dossier*

C.2 Advanced Filter

Fitur *advanced filter* berfungsi untuk melakukan *filter* data *dossier* berdasarkan berbagai kriteria yang ditentukan oleh pengguna. Tahapan awal dari fitur ini dimulai melalui pemilihan opsi *filter* dalam bentuk *dropdown*, sebagaimana ditunjukkan pada Gambar 3.10. Pada tahap ini, pengguna menekan ikon *filter* yang terdapat pada tabel *dossier*, kemudian memilih salah satu dari dua opsi yang tersedia, yaitu *Filter* untuk membuat konfigurasi penyaringan baru atau *Saved Filters* untuk menggunakan konfigurasi yang telah disimpan sebelumnya.

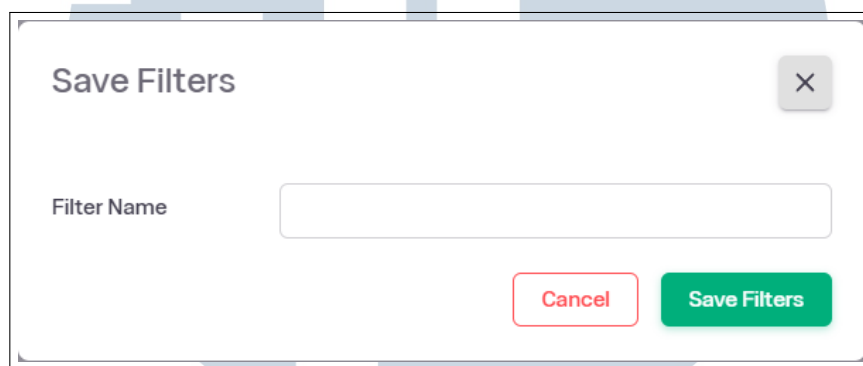


Gambar 3.10. Antarmuka pemilihan opsi *filter* pada fitur *advanced filter*

Apabila pengguna memilih menu *Filter*, maka sistem akan menampilkan antarmuka yang diperlihatkan pada Gambar 3.11. Pada tahap ini, pengguna dapat mengatur konfigurasi *filter* dengan menentukan nilai pada kolom yang diinginkan, menambahkan kondisi baru dengan ikon “+”, atau menghapus kondisi dengan ikon “-”. Di sisi kiri setiap kolom, terdapat *checkbox* yang dapat digunakan untuk mengatur visibilitas kolom tersebut pada tabel *dossier*, sehingga pengguna dapat menampilkan atau menyembunyikan kolom sesuai kebutuhan. Pengguna juga dapat memilih logika operator *AND* atau *OR* untuk menggabungkan beberapa kondisi, serta mengaktifkan opsi *case sensitive* jika diperlukan. Tersedia pula tombol *Reset All Filters* untuk mengatur ulang seluruh konfigurasi yang telah dimasukkan. Setelah konfigurasi selesai, pengguna menekan tombol *Apply Filters* untuk menerapkan *filter* pada tabel *dossier*.

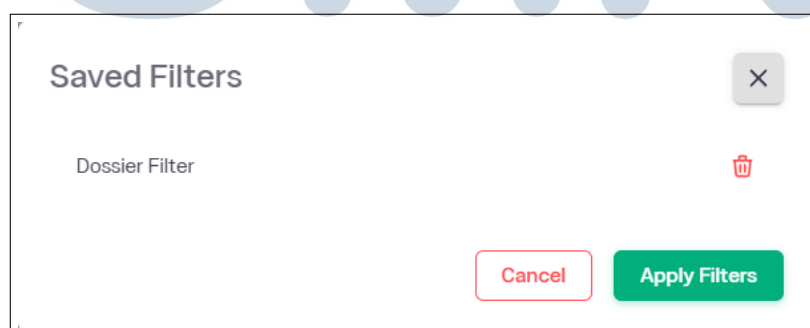
Gambar 3.11. Antarmuka pengaturan konfigurasi pada fitur *advanced filter*

Tahapan berikutnya ditunjukkan pada Gambar 3.12, yang menampilkan proses penyimpanan konfigurasi *filter* melalui fitur *save filter*. Fitur ini memungkinkan pengguna untuk menyimpan konfigurasi *filter* yang telah dibuat agar dapat digunakan kembali tanpa perlu menyusun ulang parameter. Setelah menekan tombol *Save Filters* pada Gambar 3.11, pengguna diminta untuk memberikan nama konfigurasi *filter* sebelum menekan tombol *Save Filters* untuk menyimpannya ke dalam sistem. Fitur ini meningkatkan efisiensi kerja pengguna dalam mengelola *filter* yang sering digunakan.



Gambar 3.12. Antarmuka *Save Filter* pada fitur *advanced filter*

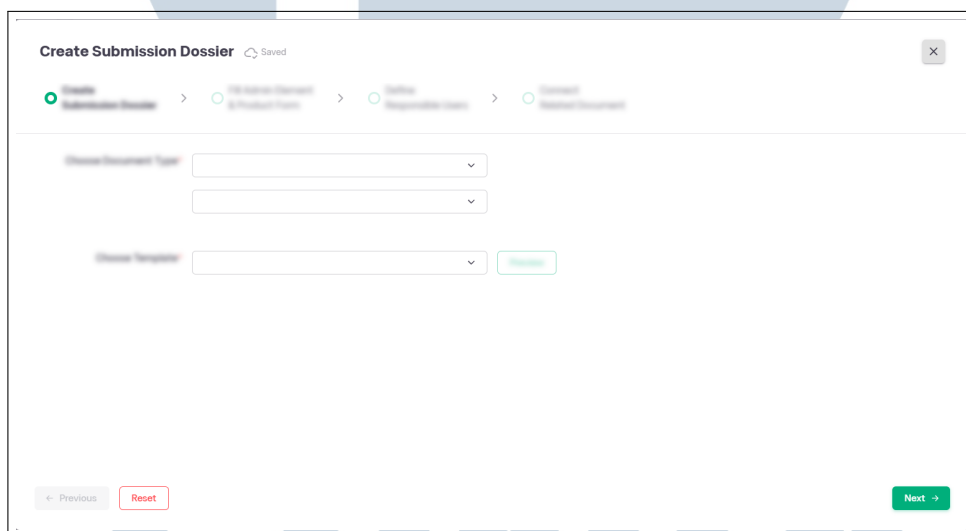
Selanjutnya, apabila pengguna memilih menu *Saved Filters* pada tahap awal yang ditunjukkan pada Gambar 3.10, sistem akan menampilkan antarmuka seperti pada Gambar 3.13. Tampilan ini berisi daftar *filter* yang telah disimpan, di mana pengguna dapat memilih *filter* yang diinginkan untuk diterapkan kembali. Selain itu, terdapat tombol hapus untuk menghapus *filter* yang tidak lagi diperlukan. Setelah memilih *filter* yang sesuai, pengguna menekan tombol *Apply Filters* untuk menerapkannya pada tabel *dossier*.



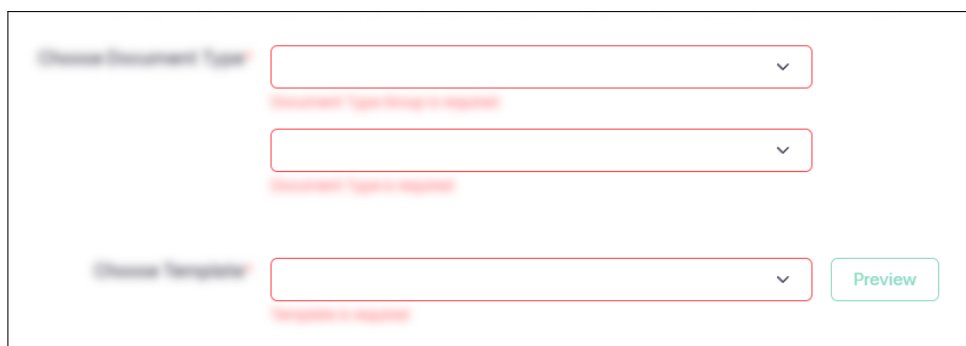
Gambar 3.13. Antarmuka *Saved Filter* pada fitur *advanced filter*

C.3 Pembuatan Dossier

Fitur pembuatan *dossier* terdiri atas beberapa tahapan yang dilakukan secara bertahap melalui sebuah formulir. Untuk membuka formulir tersebut, pengguna perlu menekan tombol *Add New* sebagaimana ditunjukkan pada Gambar 3.9. Antarmuka pada langkah pertama formulir pembuatan *dossier* ditampilkan pada Gambar 3.14. Pada tahap ini, pengguna diminta untuk mengisi data yang diperlukan sebelum melanjutkan ke langkah berikutnya. Setelah pengguna menekan tombol *Next*, sistem akan melakukan proses validasi terhadap data yang dimasukkan. Apabila ditemukan ketidaksesuaian atau ketidaklengkapan, sistem akan menampilkan pesan *error* berwarna merah, sebagaimana terlihat pada Gambar 3.15. Sebaliknya, apabila seluruh data dinyatakan valid, sistem membuat *draft* dan pengguna akan diarahkan ke tahap selanjutnya.



Gambar 3.14. Antarmuka fitur pembuatan *dossier* pada langkah pertama



Gambar 3.15. Antarmuka pesan *error* pada langkah pertama pembuatan *dossier*

Setelah tahap pertama berhasil dilalui, sistem akan menampilkan antarmuka langkah kedua sebagaimana ditunjukkan pada Gambar 3.16. Pada langkah ini, pengguna diminta untuk melengkapi informasi lanjutan sesuai dengan kebutuhan formulir. Sama seperti tahap sebelumnya, sistem akan melakukan proses validasi setelah pengguna menekan tombol *Next*. Jika ditemukan data yang belum sesuai, sistem akan menampilkan pesan *error* seperti yang ditunjukkan pada Gambar 3.17. Apabila data valid, pengguna akan diarahkan ke langkah berikutnya

Gambar 3.16. Antarmuka fitur pembuatan *dossier* pada langkah kedua

Gambar 3.17. Antarmuka pesan *error* pada langkah kedua pembuatan *dossier*

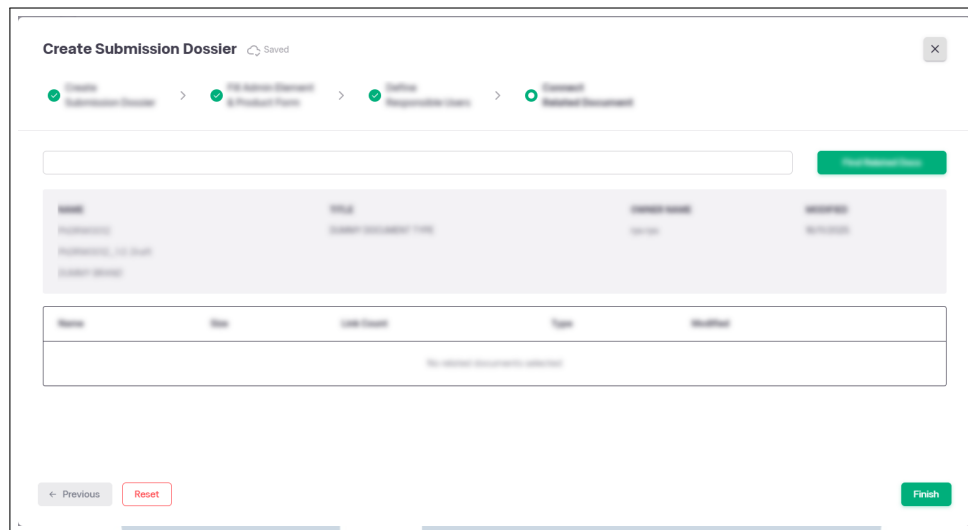
Pada langkah ketiga, antarmuka formulir ditampilkan sebagaimana terlihat pada Gambar 3.18. Tahap ini digunakan untuk mengisi informasi tambahan yang diperlukan *dossier*. Pengguna tetap mengikuti prosedur yang sama seperti tahap sebelumnya dan sistem akan melakukan proses validasi. Kesalahan dalam pengisian

akan ditandai dengan pesan *error* yang ditampilkan pada Gambar 3.19, sedangkan data yang valid memungkinkan pengguna untuk melanjutkan ke langkah terakhir.

Gambar 3.18. Antarmuka fitur pembuatan *dossier* pada langkah ketiga

Gambar 3.19. Antarmuka pesan *error* pada langkah ketiga pembuatan *dossier*

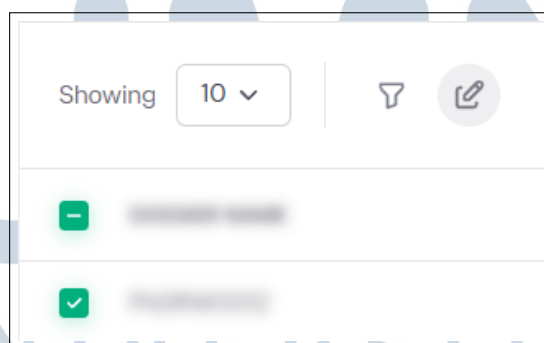
Antarmuka pada langkah terakhir ini diperlihatkan pada Gambar 3.20. Pada tahap ini, pengguna dapat meninjau data yang telah dimasukkan dan melakukan penyesuaian apabila diperlukan. Setelah seluruh informasi dianggap benar, pengguna dapat menekan *Finish* untuk menyelesaikan proses. Sistem kemudian akan menyimpan seluruh data ke dalam tabel *Dossiers*, serta menghapus *draft* yang tersisa. Setelah itu, sistem akan menampilkan data tersebut pada tabel daftar *dossier*.



Gambar 3.20. Antarmuka fitur pembuatan *dossier* pada langkah keempat

C.4 Edit Dossier

Untuk melakukan proses pengeditan, pengguna perlu memilih salah satu *dossier* yang berstatus *Draft* pada tabel daftar *dossier*, kemudian menekan ikon edit seperti yang diperlihatkan pada Gambar 3.21. Hanya *dossier* dengan status *Draft* yang dapat diperbarui, karena status ini menandakan bahwa dokumen masih dalam tahap penyusunan dan belum difinalisasi.



Gambar 3.21. Antarmuka ikon edit *dossier*

Setelah pengguna menekan ikon edit, sistem akan memuat seluruh data yang tersimpan pada *dossier* tersebut dan menampilkan formulir pengeditan seperti yang terlihat pada Gambar 3.22. Pada tahap ini, pengguna dapat memperbarui informasi yang diperlukan sesuai kebutuhan. Setelah proses pengeditan selesai, pengguna menekan tombol *Finish* untuk menyimpan perubahan. Sistem kemudian akan

melakukan validasi terhadap data yang dimasukkan, dan apabila data dinyatakan valid, sistem akan memperbarui informasi *dossier* di dalam *database*.

Gambar 3.22. Antarmuka fitur edit *dossier*

3.3.2 Implementasi Fitur Version History pada Modul Document Editor

A User Requirement

Fitur *Version History* pada modul *Document Editor* berfungsi untuk mencatat setiap perubahan pada dokumen, termasuk identitas pengguna yang melakukan perubahan, waktu modifikasi, dan versi dokumen yang dihasilkan. Riwayat ini memungkinkan pengguna melakukan *preview*, membandingkan, atau memulihkan versi sebelumnya apabila diperlukan. Tujuan utamanya adalah menjaga keterlacakan dan keutuhan dokumen selama proses kolaborasi, serta memastikan transparansi dalam pengelolaan dokumen. Berikut merupakan kebutuhan pengguna yang dirancang untuk mendukung implementasi fitur ini.

A.1 Pembuatan Versi Baru Dokumen

Sistem harus secara otomatis membuat versi baru setiap kali dokumen mengalami perubahan yang disimpan. Setiap versi yang tersimpan wajib memiliki penanda unik yang mencakup informasi versi, tanggal, dan waktu penyimpanan, serta identitas pengguna yang melakukan perubahan.

A.2 Preview Dokumen Versi Sebelumnya

Sistem menyediakan antarmuka yang menampilkan daftar seluruh versi dokumen yang pernah disimpan. Pengguna dapat memilih versi sebelumnya untuk dibandingkan guna mengidentifikasi perbedaan isi dokumen secara visual.

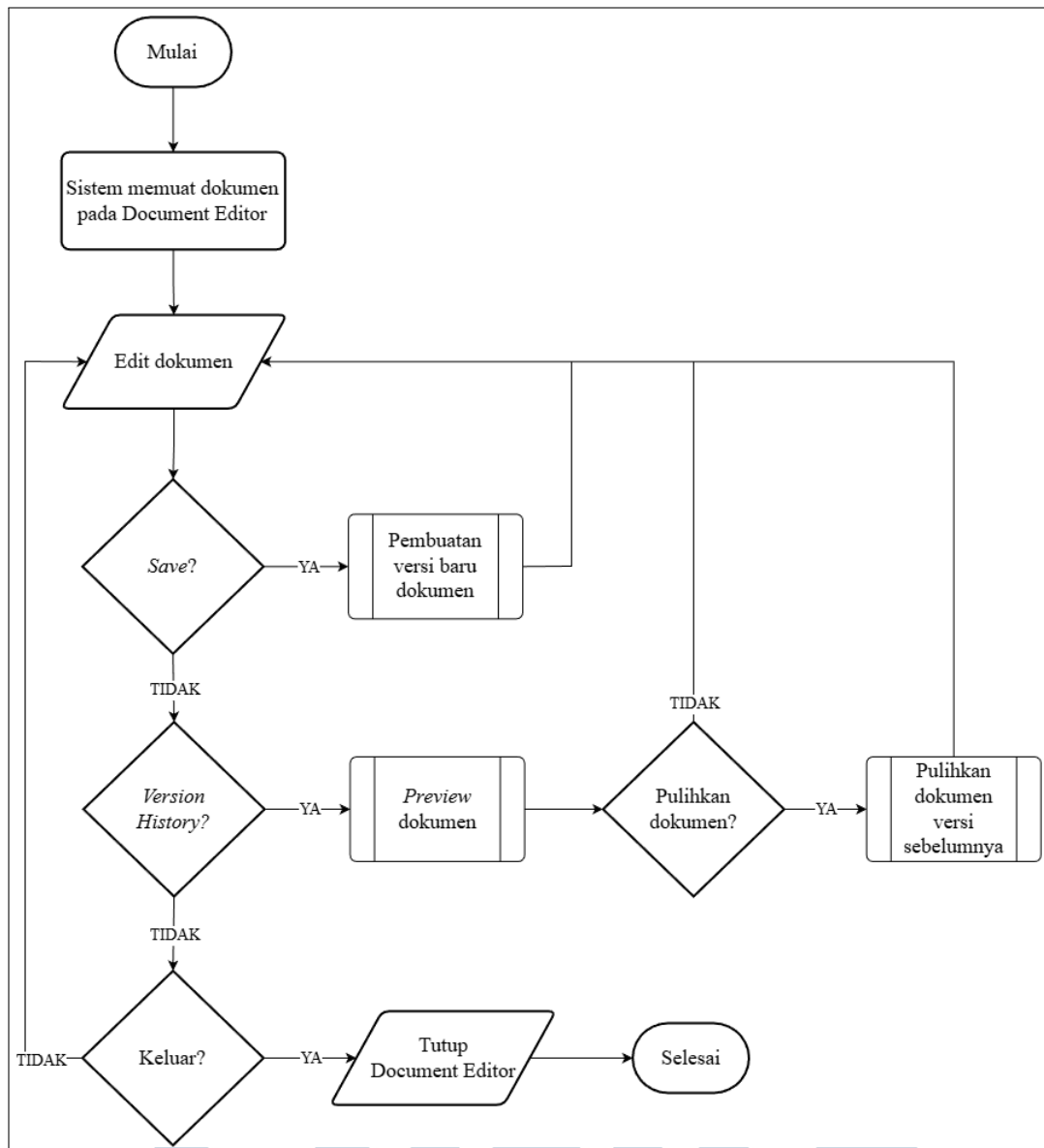
A.3 Pemulihan Dokumen Versi Sebelumnya

Fitur *restore* disediakan untuk memulihkan dokumen ke versi tertentu yang dipilih pengguna. Setelah proses pemulihan dilakukan, sistem akan mencatat tindakan tersebut sebagai versi baru guna mempertahankan integritas riwayat dokumen.

B Flowchart Fitur Version History

Alur proses dari fitur *version history* dapat dilihat pada Gambar 3.23. Proses dimulai ketika sistem memuat dokumen pada *Document Editor*, kemudian pengguna dapat melakukan pengeditan sesuai kebutuhan. Setelah itu, pengguna dapat memilih tindakan lanjutan yang tersedia. Apabila pengguna ingin *save* dokumen, sistem akan menjalankan sub-proses untuk pembuatan versi baru dokumen. Selain itu, ketika pengguna membuka menu *Version History*, pengguna juga dapat melakukan *preview* terhadap dokumen versi sebelumnya melalui sub-proses *preview* dokumen. Setelah melakukan *preview*, pengguna dapat memilih untuk memulihkan versi tersebut apabila diperlukan, yang kemudian akan dilanjutkan ke sub-proses pemulihan dokumen versi sebelumnya. Apabila seluruh kegiatan telah selesai dan pengguna ingin keluar, pengguna dapat menutup *Document Editor*, sehingga proses dianggap selesai.

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



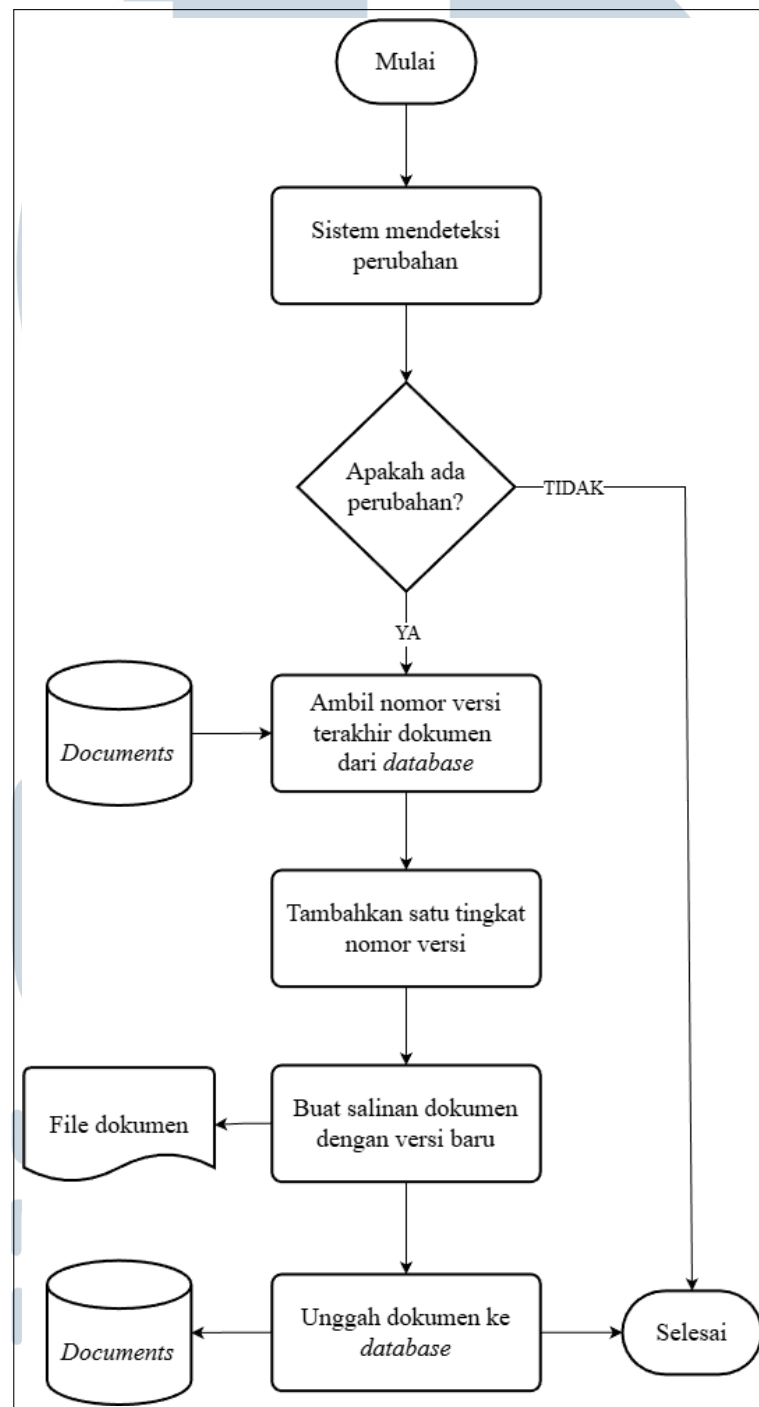
Gambar 3.23. Flowchart fitur *Version History*

Berikut merupakan penjelasan alur yang mendukung implementasi fitur ini.

B.1 Pembuatan Versi Baru Dokumen

Proses pembuatan versi baru dokumen ditunjukkan pada Gambar 3.24. Proses dimulai ketika sistem mendeteksi adanya perubahan pada isi dokumen. Jika perubahan terdeteksi, sistem akan mengambil nomor versi terakhir dari dokumen tersebut melalui tabel *Documents* pada *database*. Nomor versi tersebut kemudian ditambahkan satu tingkat untuk menghasilkan nomor versi baru. Setelah nomor

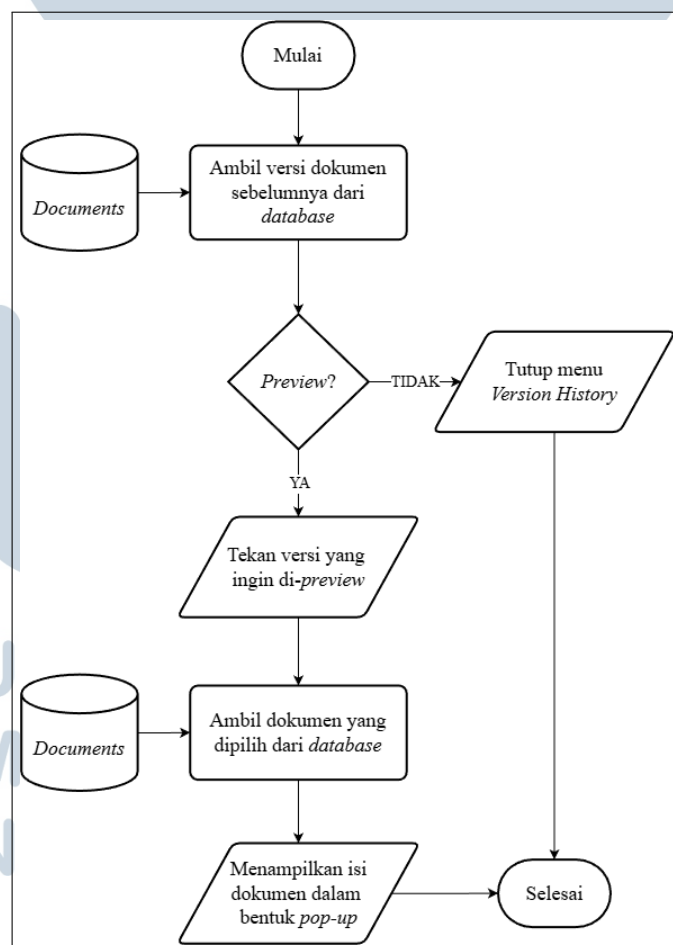
versi ditentukan, sistem akan membuat salinan dokumen dengan versi terbaru dan menambahkan penanda versi pada nama file. Selanjutnya, dokumen versi baru tersebut diunggah dan disimpan kembali ke tabel *Documents* sebagai entitas versi terbaru.



Gambar 3.24. Flowchart mekanisme pembuatan versi baru dokumen

B.2 Preview Dokumen Versi Sebelumnya

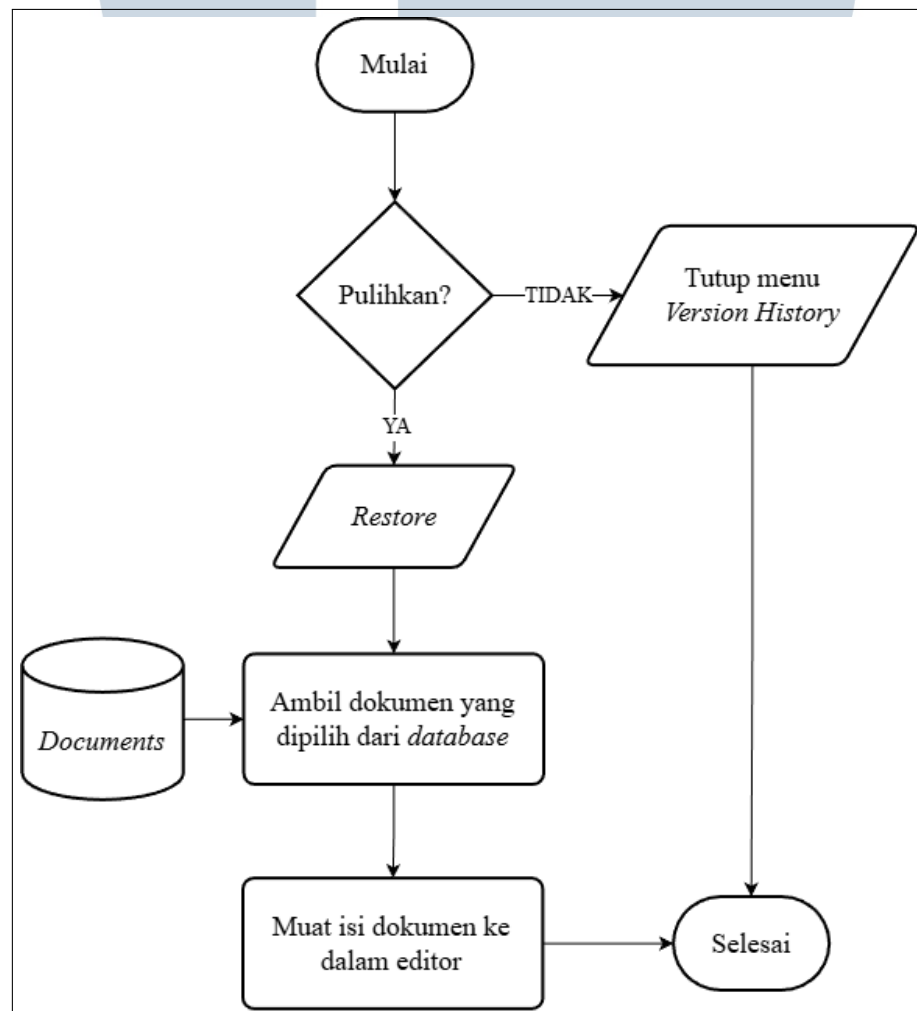
Proses *preview* dokumen versi sebelumnya ditunjukkan pada Gambar 3.25. Alur dimulai ketika sistem mengambil daftar seluruh versi dokumen yang tersimpan pada tabel *Documents* di *database*. Daftar tersebut kemudian dimuat dan ditampilkan kepada pengguna di menu *Version History*. Jika pengguna ingin *preview* dokumen, pengguna dapat memilih salah satu versi dokumen yang ingin di-*preview*. Ketika sebuah versi dipilih, sistem akan melakukan pengambilan data dokumen sesuai versi tersebut dari *database*, termasuk lokasi file dokumen yang terkait. File dokumen kemudian diproses dan ditampilkan kepada pengguna dalam bentuk *pop-up* sebagai tampilan *preview*. Apabila pengguna tidak ingin melakukan *preview*, pengguna dapat menutup menu *Version History*. Mekanisme ini memungkinkan pengguna meninjau konten dokumen versi tertentu secara langsung tanpa harus mengganti dokumen utama yang sedang dibuka.



Gambar 3.25. Flowchart mekanisme *preview* dokumen versi sebelumnya

B.3 Pemulihan Dokumen Versi Sebelumnya

Proses pemulihan dokumen ke versi sebelumnya dilanjutkan dari sub-proses *preview* dokumen versi sebelumnya. Proses ini ditunjukkan pada Gambar 3.26. Alur dimulai ketika pengguna ingin melanjutkan pemulihan dokumen. Apabila pengguna ingin melanjutkan, pengguna dapat menekan *Restore* pada versi yang ingin dipulihkan. Saat tindakan ini dilakukan, sistem mengambil file dokumen sesuai versi yang dipilih dari *database*. File dokumen tersebut kemudian dimuat kembali ke dalam *Document Editor* sebagai isi dokumen aktif. Apabila pengguna tidak ingin lanjut pemulihan dokumen, pengguna dapat menutup menu *Version History*.

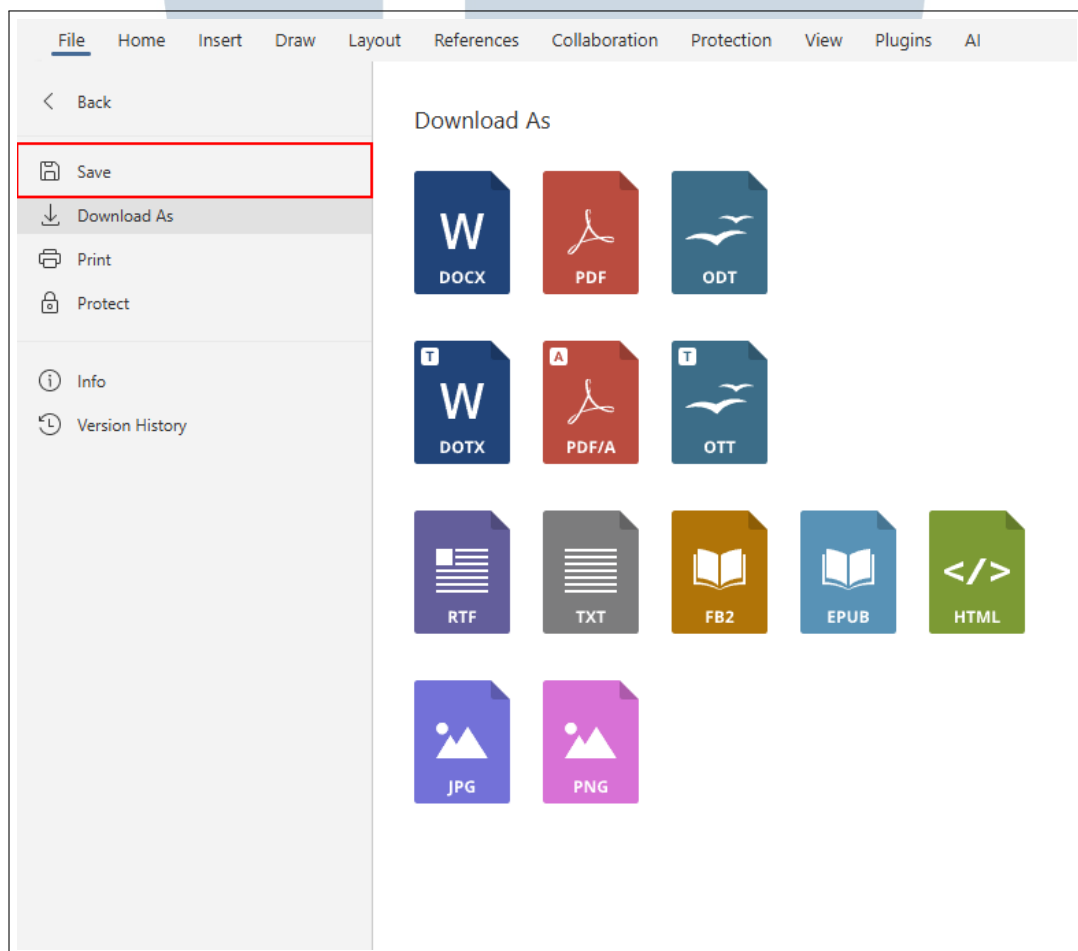


Gambar 3.26. Flowchart mekanisme pemulihan dokumen versi sebelumnya

C Hasil Implementasi Fitur Version History

C.1 Pembuatan Versi Baru Dokumen

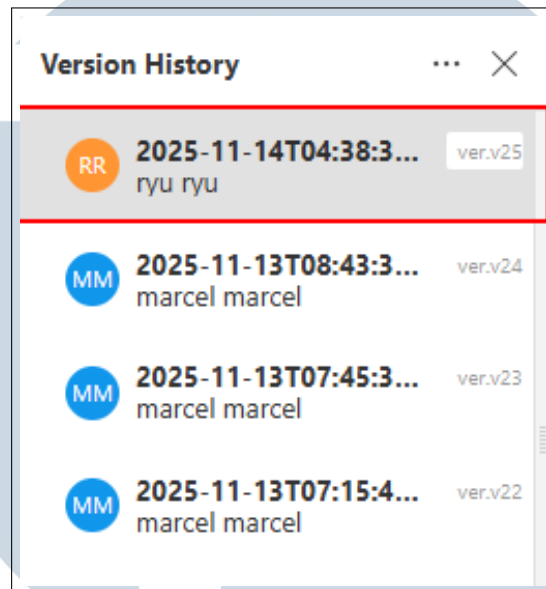
Gambar 3.27 menunjukkan antarmuka tombol *Save* yang dapat diakses melalui menu *File* pada *Document Editor*. Ketika pengguna memilih opsi ini, sistem akan melakukan penyimpanan terhadap dokumen yang sedang dikerjakan. Tindakan penyimpanan ini menjadi pemicu utama pembuatan versi baru, terutama apabila terdapat perubahan pada isi dokumen sejak versi terakhir disimpan. Mekanisme ini memastikan bahwa setiap pembaruan dapat direkam secara tepat waktu dan tersimpan sebagai bagian dari sejarah pengeditan dokumen.



Gambar 3.27. Antarmuka tombol *Save* yang digunakan untuk menyimpan dokumen

Setelah proses penyimpanan berhasil dilakukan, versi baru dokumen akan otomatis ditambahkan ke dalam *version history*. Gambar 3.28 memperlihatkan antarmuka *Version History* setelah versi baru terbentuk. Pada antarmuka tersebut,

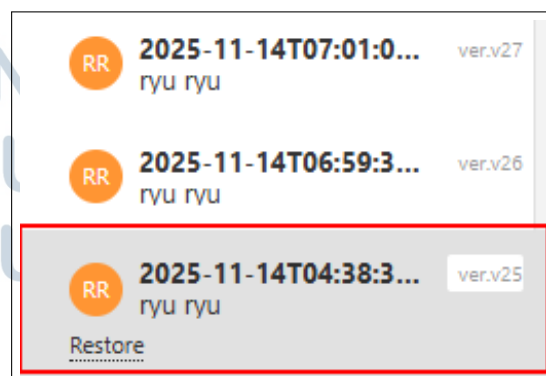
versi terbaru ditampilkan pada urutan teratas, dilengkapi dengan informasi waktu pembuatan versi, identitas pengguna yang melakukan penyimpanan, serta nomor versi yang dihasilkan oleh sistem.



Gambar 3.28. Antarmuka versi dokumen yang baru dibuat

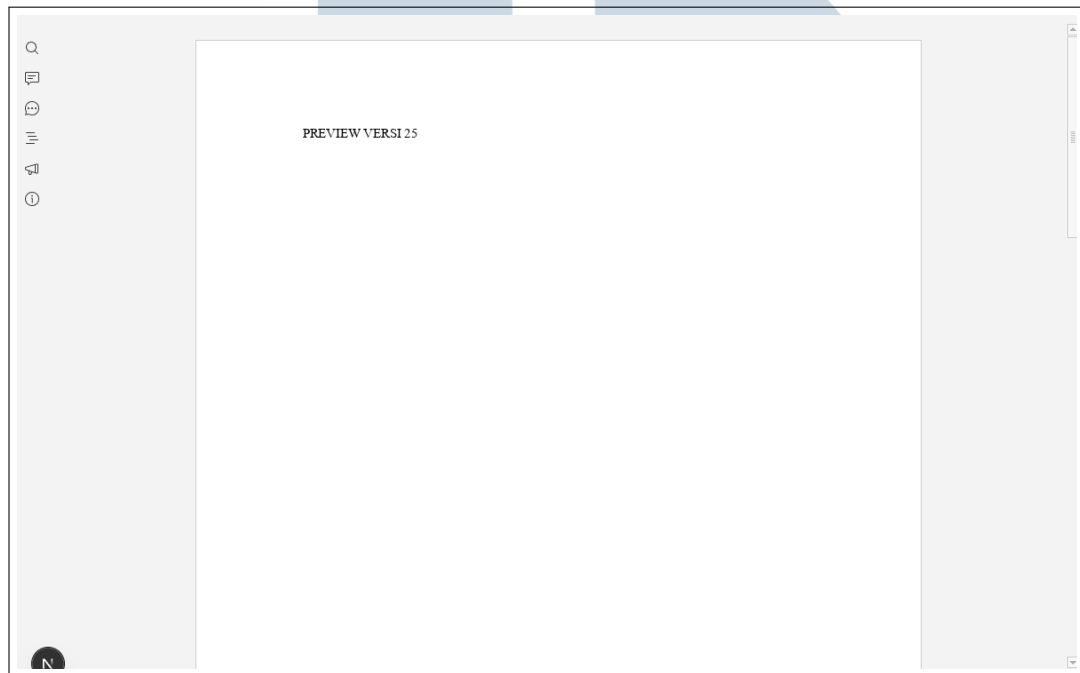
C.2 Preview Dokumen Versi Sebelumnya

Untuk melakukan *preview* terhadap dokumen pada versi tertentu, pengguna dapat membuka menu *Version History* dan memilih salah satu versi yang tersedia. Gambar 3.29 memperlihatkan antarmuka ketika pengguna menekan versi dokumen yang ingin ditinjau. Setiap versi ditampilkan beserta informasi waktu pembuatan dan identitas pengguna yang melakukan perubahan, sehingga memudahkan proses pemilihan versi yang relevan.



Gambar 3.29. Antarmuka pemilihan versi untuk melakukan *preview*

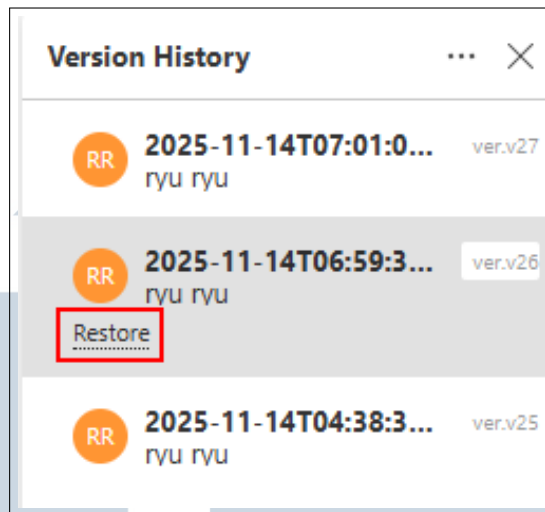
Setelah pengguna memilih versi yang diinginkan, sistem akan memuat isi dokumen dari versi tersebut dan menampilkannya dalam bentuk *pop-up* seperti yang ditunjukkan pada Gambar 3.30. Tampilan ini bersifat *read-only*, sehingga pengguna dapat meninjau isi dokumen tanpa risiko mengubah konten aslinya.



Gambar 3.30. Antarmuka *preview* dokumen versi sebelumnya dalam tampilan *pop-up*

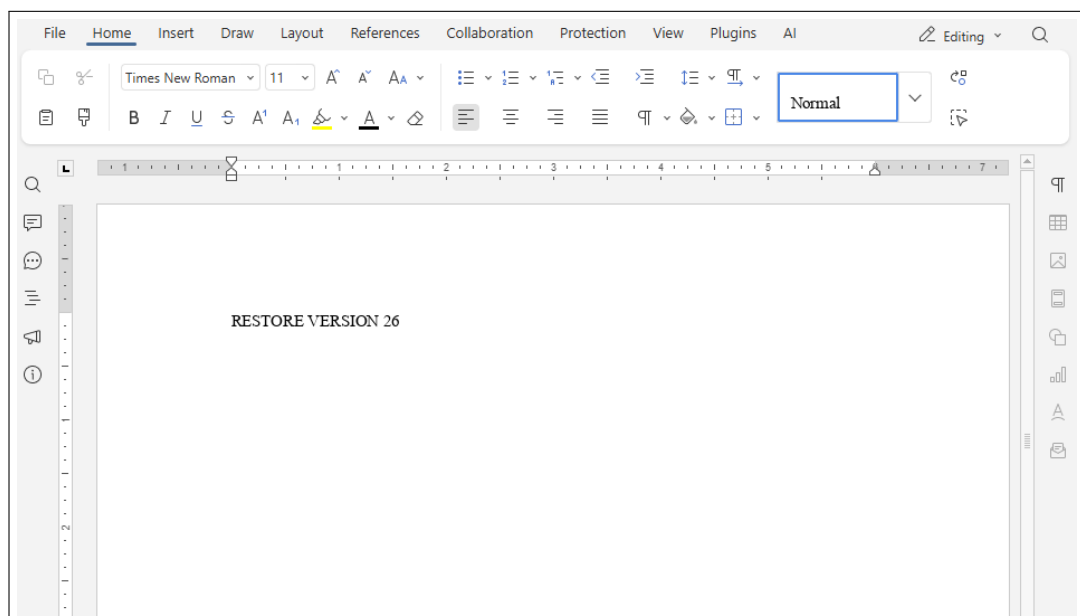
C.3 Pemulihan Dokumen Versi Sebelumnya

Untuk melakukan pemulihan terhadap dokumen pada versi tertentu, pengguna dapat membuka menu *Version History* dan memilih salah satu versi yang tersedia pada daftar riwayat. Gambar 3.31 menunjukkan antarmuka ketika pengguna menekan tombol *Restore* pada versi dokumen yang ingin dipulihkan. Tombol ini memungkinkan pengguna mengembalikan isi dokumen ke kondisi sebagaimana versi tersebut dibuat.



Gambar 3.31. Antarmuka tombol *Restore* pada daftar versi dokumen

Setelah pengguna menekan tombol *Restore*, sistem akan mengambil file dokumen sesuai versi yang dipilih dari *database* dan memuatnya kembali ke dalam *Document Editor*. Hasil pemulihan ditampilkan sebagai isi dokumen aktif, seperti terlihat pada Gambar 3.32. Pada tahap ini, pengguna dapat langsung melanjutkan proses pengeditan apabila diperlukan.



Gambar 3.32. Antarmuka isi dokumen yang baru dipulihkan

3.3.3 Integrasi Modul Document Editor

A User Requirement

Integrasi modul *Document Editor* dengan RIMS bertujuan untuk memusatkan proses pengeditan dokumen dalam satu sistem, sehingga seluruh aktivitas pengolahan dokumen dapat dilakukan tanpa berpindah sistem. Dengan adanya integrasi ini, konsistensi, keterlacakan, dan efisiensi dalam pengelolaan dokumen dapat terjaga. Berikut merupakan beberapa fitur yang dibuat di dalam modul ini.

A.1 Pengambilan dan Pengolahan Dokumen

Sistem harus mampu mengakses dan memproses dokumen yang tersimpan di *database* secara langsung. Pengguna dapat membuka serta mengedit dokumen tanpa perlu mengunduh berkas secara manual. Integrasi ini memastikan seluruh aktivitas pengolahan dilakukan secara terpusat, sehingga alur kerja menjadi lebih efisien dan terdokumentasi dengan baik.

A.2 Collaborative Editing

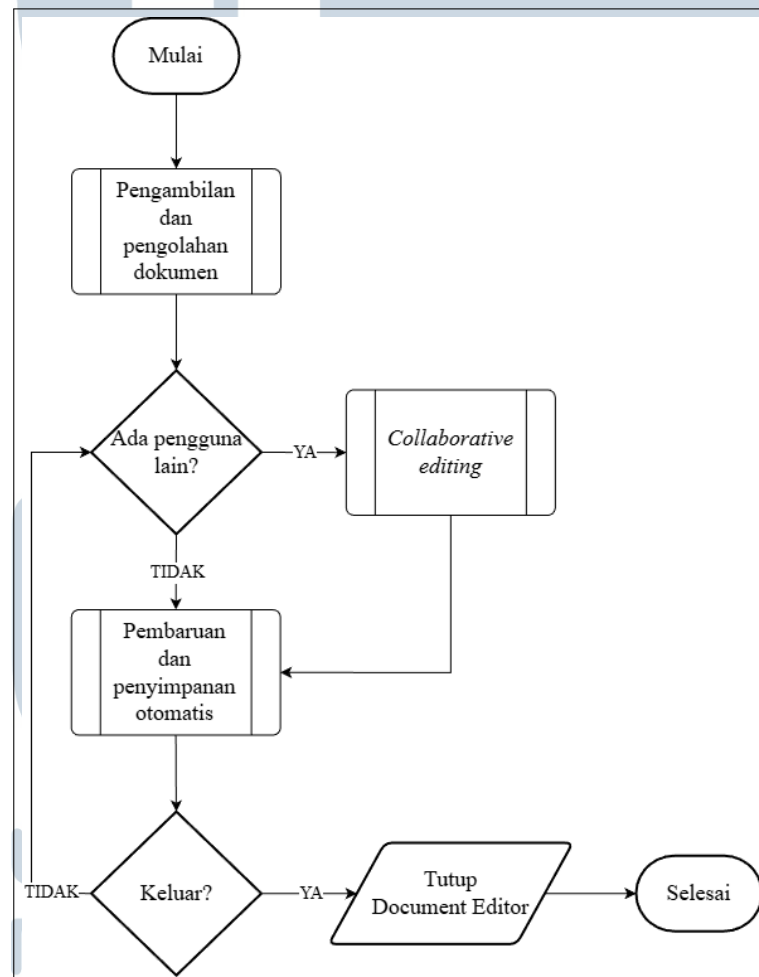
Fitur ini memungkinkan beberapa pengguna untuk melakukan pengeditan dokumen secara bersamaan dan *real-time*. Perubahan yang dilakukan oleh masing-masing pengguna akan langsung terlihat oleh pengguna lain. Sistem juga dilengkapi dengan mekanisme penguncian atau penanda area yang sedang diedit untuk mencegah konflik selama proses pengeditan bersama.

A.3 Pembaruan dan Penyimpanan Otomatis

Sistem harus mendukung penyimpanan otomatis terhadap dokumen yang telah diperbarui. Setiap kali pengguna melakukan penyimpanan melalui *Document Editor*, perubahan akan langsung tersinkronisasi dengan dokumen di *database*. Mekanisme ini mencegah terjadinya duplikasi versi dan menjamin bahwa dokumen yang diakses selalu merupakan versi terbaru.

B Flowchart Modul Document Editor

Alur dari modul *Document Editor* ditunjukkan pada Gambar 3.33. Proses diawali dengan sub-proses pengambilan dan pengolahan dokumen. Setelah itu, sistem akan memeriksa apakah terdapat pengguna lain yang mengakses dokumen yang sama. Jika ada lebih dari satu pengguna, sistem menjalankan sub-proses *collaborative editing* yang kemudian diteruskan dengan mekanisme pembaruan dan penyimpanan otomatis. Jika hanya terdapat satu pengguna, proses langsung diarahkan menuju tahap pembaruan dan penyimpanan otomatis. Ketika pengguna sudah selesai, proses ditutup dengan menutup *Document Editor*.

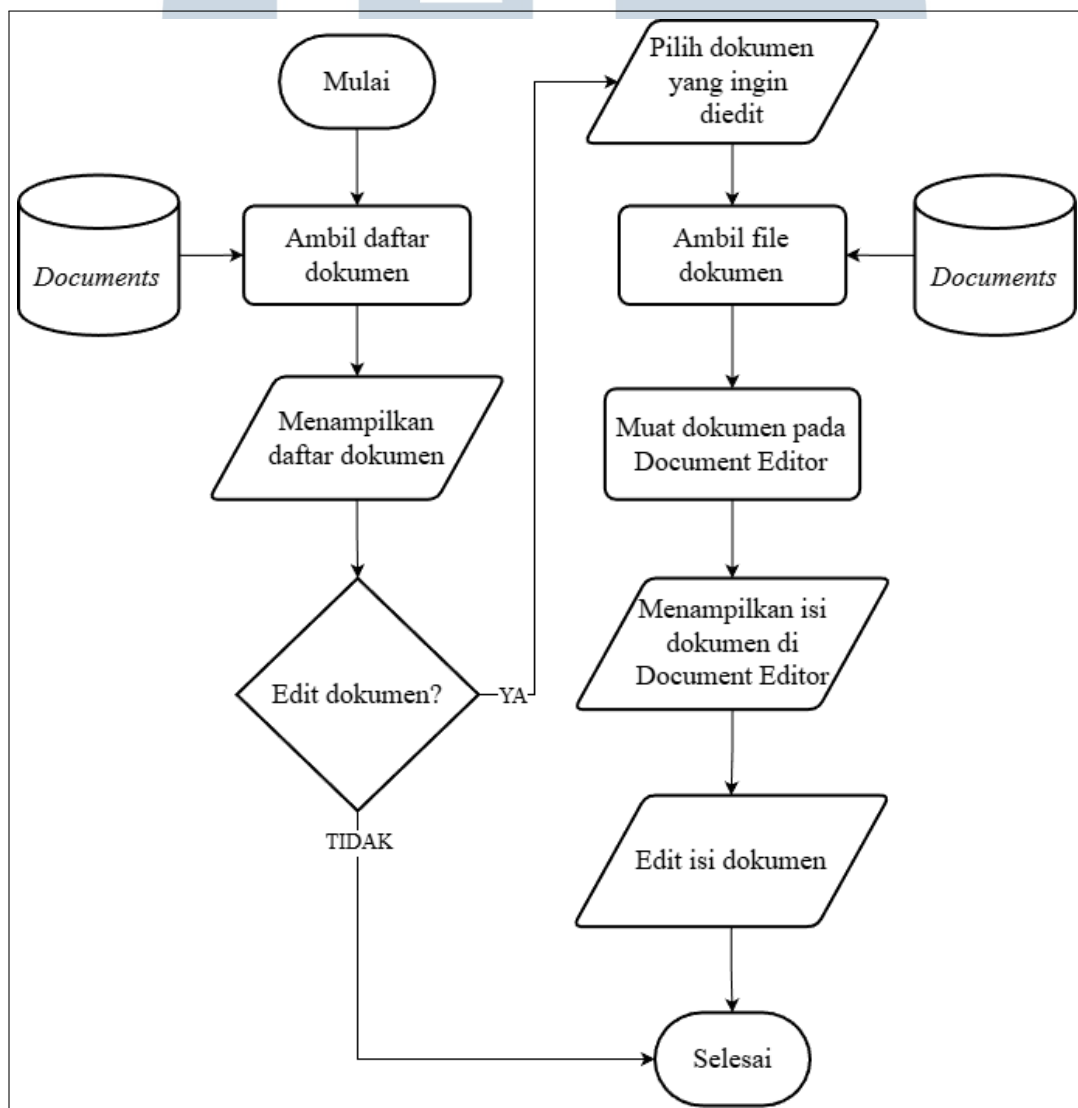


Gambar 3.33. Flowchart modul *Document Editor*

Berikut merupakan penjelasan alur dari fitur yang dibuat di dalam modul *Document Editor*.

B.1 Pengambilan dan Pengolahan Dokumen

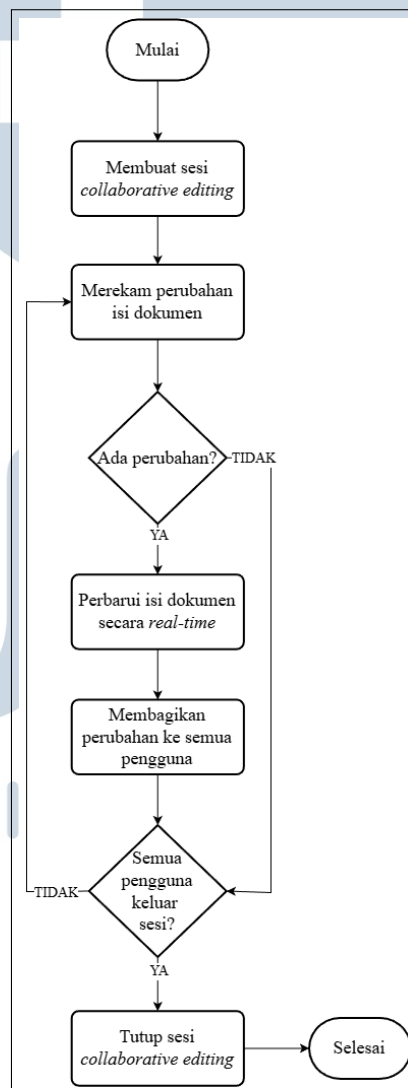
Alur proses untuk fitur pengambilan dan pengolahan dokumen ditunjukkan pada Gambar 3.34. Proses dimulai dengan pengambilan daftar dokumen dari tabel *Documents* pada *database* untuk kemudian ditampilkan kepada pengguna. Apabila pengguna ingin mengedit dokumen tertentu, pengguna dapat memilih dokumen yang diinginkan, dan sistem akan mengambil file dokumen tersebut dari *database*. Setelah file berhasil dimuat, sistem akan menampilkan isi dokumen pada *Document Editor* sehingga pengguna dapat langsung melakukan pengeditan.



Gambar 3.34. Flowchart fitur pengambilan dan pengolahan dokumen

B.2 Collaborative Editing

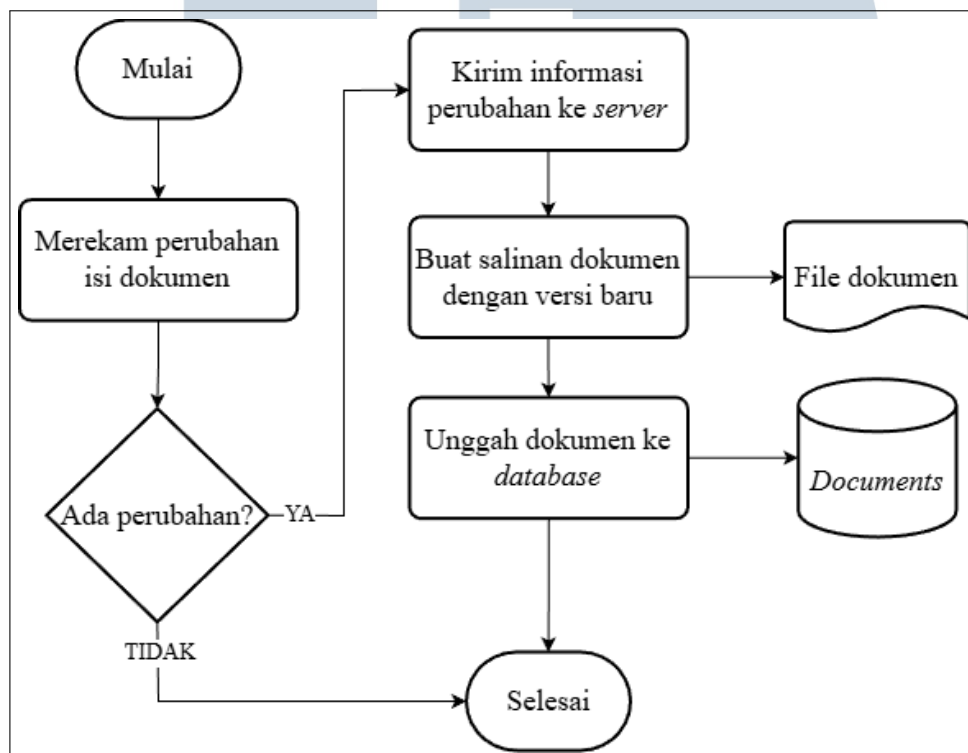
Alur proses untuk fitur *collaborative editing* ditunjukkan pada Gambar 3.35. Proses dimulai ketika sistem membentuk sebuah sesi *collaborative editing*. Setelah sesi berhasil dibuat, sistem akan memantau dan merekam setiap perubahan pada isi dokumen. Apabila terdeteksi perubahan, sistem akan memperbarui isi dokumen secara *real-time* dan mendistribusikan pembaruan tersebut kepada seluruh pengguna yang sedang mengakses dokumen yang sama. Selanjutnya, apabila seluruh pengguna telah keluar dari sesi *collaborative editing*, sistem akan menutup sesi tersebut. Namun, jika masih terdapat pengguna yang terhubung, sistem akan terus melanjutkan proses pemantauan dan perekaman perubahan dokumen.



Gambar 3.35. Flowchart fitur *collaborative editing*

B.3 Pembaruan dan Penyimpanan Otomatis

Alur proses untuk fitur pembaruan dan penyimpanan otomatis ditunjukkan pada Gambar 3.36. Pada proses ini, sistem akan memantau dan merekam setiap perubahan pada isi dokumen. Apabila ada perubahan, sistem akan mengirimkan informasi tersebut ke *server* untuk kemudian membuat salinan dokumen dalam versi terbaru. Setelah versi baru berhasil dibuat, sistem akan mengunggah dokumen tersebut ke dalam *database*.



Gambar 3.36. Flowchart fitur pembaruan dan penyimpanan otomatis

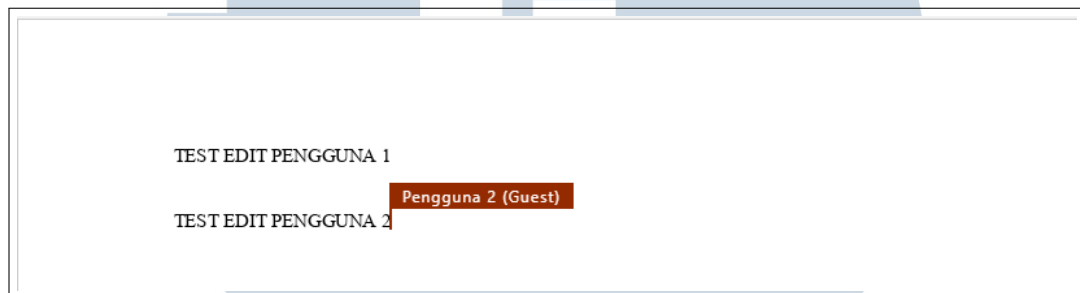
C Hasil Integrasi Modul Document Editor

C.1 Pengambilan dan Pengolahan Dokumen

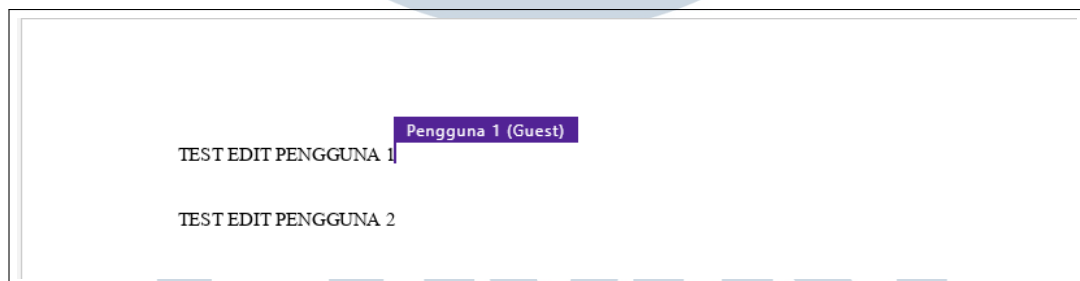
Gambar 3.37 menunjukkan tampilan daftar dokumen yang diperoleh melalui fitur pengambilan dokumen. Seluruh dokumen ditampilkan dalam bentuk tabel sehingga memudahkan pengguna dalam memilih dokumen yang akan diolah. Untuk mulai mengolah dokumen, pengguna dapat menekan nama dokumen yang diinginkan.

C.2 Collaborative Editing

Fitur *collaborative editing* akan aktif ketika lebih dari satu pengguna membuka dan mengolah dokumen yang sama. Gambar 3.39 menampilkan antarmuka *collaborative editing* dari sudut pandang Pengguna 1. Pada tampilan tersebut, Pengguna 1 dapat melihat nama pengguna lain yang sedang mengakses dokumen beserta posisi *cursor* mereka saat melakukan pengeditan. Selanjutnya, Gambar 3.40 menunjukkan tampilan yang sama dari sudut pandang Pengguna 2.



Gambar 3.39. Antarmuka *collaborative editing* dari sudut pandang Pengguna 1



Gambar 3.40. Antarmuka *collaborative editing* dari sudut pandang Pengguna 2

C.3 Pembaruan dan Penyimpanan Otomatis

Fitur pembaruan dan penyimpanan otomatis dapat diaktifkan melalui pengaturan pada sisi server dan *Document Server*. Pada pengaturan tersebut, diperlukan penambahan logika yang memungkinkan sistem mendeteksi perubahan dokumen, membuat salinan versi terbaru, serta menyimpannya secara berkala tanpa perlu tindakan manual dari pengguna.

3.4 Kendala dan Solusi yang Ditemukan

Selama mengikuti kegiatan magang di PT Kalbe Farma Tbk, terdapat beberapa kendala yang ditemukan dalam pengerjaan. Berikut merupakan beberapa kendala yang dialami.

1. Kesulitan dalam mengimplementasikan alur logika fitur *version history* pada *Document Editor*

Kendala ini muncul karena terbatasnya dokumentasi resmi yang tersedia mengenai mekanisme internal *Document Editor*. Hal tersebut menyebabkan proses pengembangan terhambat dan memerlukan waktu lebih lama untuk memahami alur kerja fitur tersebut.

Solusi: Melakukan eksplorasi mandiri secara bertahap, termasuk melakukan pengujian berulang, serta berdiskusi dengan tim terkait untuk memastikan pemahaman alur fitur sudah sesuai kebutuhan sistem.

2. *User requirement* yang sering berubah

Selama proses pengembangan, *user requirement* mengalami beberapa kali perubahan. Kondisi ini menyebabkan alur proses yang telah disusun harus direvisi berulang kali, sehingga berdampak pada waktu penyelesaian yang lebih lama dan menuntut penyesuaian ulang pada beberapa bagian perancangan.

Solusi: Melakukan komunikasi rutin dengan pihak terkait untuk memastikan kebutuhan sudah berada dalam kondisi final sebelum masuk ke tahap implementasi, serta menyiapkan rancangan alur yang lebih fleksibel agar perubahan dapat diakomodasi tanpa mengubah keseluruhan struktur.

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