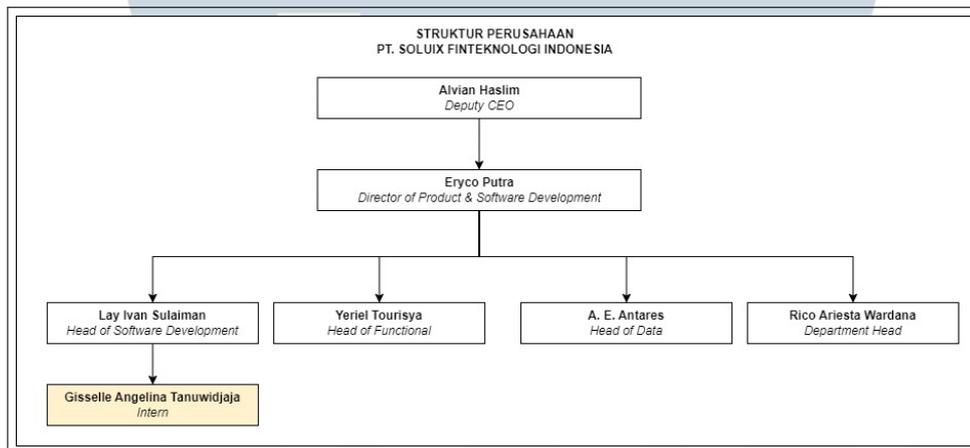


BAB 3 PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama program kerja magang di PT. Soluix Finteknologi Indonesia, posisi yang diberikan adalah sebagai anggota tim *Software Development*. Pada Gambar 3.1, tanggung jawab yang diemban mencakup pengembangan aplikasi internal perusahaan serta pemberian dukungan apabila terjadi permasalahan pada proyek-proyek perusahaan, sehingga seluruh proses dapat berjalan dengan lancar dan mendukung kelancaran operasional perusahaan.



Gambar 3.1. Struktur organisasi perusahaan PT Soluix Finteknologi Indonesia

Sumber: [9]

Dalam proses magang, berbagai media digunakan untuk mendukung kegiatan dan mempermudah komunikasi, yaitu:

1. WhatsApp → digunakan sebagai platform komunikasi utama untuk koordinasi harian.
2. Microsoft Teams → digunakan sebagai platform komunikasi utama oleh tim untuk menyampaikan pembaruan dan melakukan koordinasi selama proses pengembangan.

3.2 Tugas yang Dilakukan

Tugas utama yang dilaksanakan selama masa magang adalah merancang dan membangun sebuah program internal yang digunakan untuk keperluan operasional perusahaan. Proses pengembangan aplikasi ini dimulai dengan kegiatan perancangan dokumen *Functional Specification Document* yang mencakup elemen-elemen penting seperti *Entity Relationship Diagram* dan *flowchart* alur sistem secara menyeluruh. FSD ini disusun berdasarkan kebutuhan dari pihak perusahaan, dan selanjutnya melalui proses revisi untuk memastikan bahwa rancangan sistem telah sesuai dengan harapan dan kebutuhan bisnis.

Setelah FSD disepakati, tahapan berikutnya adalah perancangan dan implementasi struktur basis data, yaitu *database design*, dari awal, termasuk penentuan skema, tabel, relasi, serta aturan validasi data yang sesuai dengan model yang telah disusun sebelumnya. Barulah setelah struktur basis data selesai, proses pengembangan aplikasi dimulai, mencakup pembuatan *backend service*, *frontend interface*, serta fitur-fitur utama lainnya sesuai spesifikasi yang telah ditetapkan.

Alasan utama pengembangan program internal adalah karena keterbatasan Power Automate dalam memenuhi kebutuhan perusahaan. Oleh karena itu, perusahaan memutuskan untuk mengembangkan sistem internal yang lebih fleksibel, efisien, dan mudah disesuaikan. Sistem ini dirancang khusus untuk mendukung kebutuhan internal dengan alur kerja yang selaras dengan struktur organisasi dan regulasi yang berlaku. Sebagaimana ditunjukkan pada Tabel 3.1, kegiatan magang dilaksanakan secara bertahap setiap minggunya.

Tabel 3.1. Ringkasan Tugas Mingguan (Februari - Mei 2024)

Minggu Ke-	Ringkasan Tugas
1	Memperdalam pengetahuan mengenai Java serta mulai mempelajari <i>framework</i> pendukung seperti Spring Boot, Thymeleaf, dan MyBatis sebagai fondasi utama dalam pengembangan proyek.
2	Mempelajari proyek serupa dan sistem yang telah ada untuk memahami pola pengembangan serta struktur sistem yang digunakan dalam proyek.
Lanjut pada halaman berikutnya	

Minggu Ke-	Ringkasan Tugas
3	Mengerjakan tugas dokumentasi yang diberikan oleh perusahaan menggunakan draw.io dan Docusaurus, sebagai bagian dari pemahaman alur sistem.
4	Menyusun dokumen <i>Functional Specification Design</i> berdasarkan analisis kebutuhan dan hasil studi sistem terdahulu.
5	Melakukan revisi dan penyempurnaan dokumen <i>Functional Specification Design</i> agar lebih sesuai dengan kebutuhan sistem serta standar dokumentasi perusahaan.
6	Merancang struktur basis data menggunakan MySQL dengan bantuan aplikasi DBeaver, berdasarkan kebutuhan sistem yang telah dirumuskan.
7	Melakukan revisi terhadap struktur database, termasuk pengaturan relasi antar tabel, penambahan kolom, dan pengoptimalan tipe data.
8	Melakukan <i>initial commit</i> proyek dan membangun halaman login sederhana menggunakan Thymeleaf, serta menghubungkannya ke <i>backend</i> Spring Boot.
9	Mengimplementasikan fitur autentikasi pengguna menggunakan <i>JSON Web Token</i> , termasuk pengaturan akses dinamis berdasarkan peran pengguna.
10	Menyempurnakan mekanisme autentikasi dengan validasi <i>JSON Web Token</i> , <i>idle timeout</i> , serta mengamankan penyimpanan password menggunakan SHA-256.
11	Membangun halaman <i>Admin</i> untuk manajemen pengguna dan proyek, termasuk fitur tambah, edit, dan hapus, dengan tampilan berbasis Thymeleaf dan data dari MyBatis.
12	Mengembangkan fitur pengajuan klaim dengan form dinamis, validasi data, dan penyimpanan ke basis data yang mendukung beberapa jenis klaim (<i>Personal, Project, Company</i>).
Lanjut pada halaman berikutnya	

Minggu Ke-	Ringkasan Tugas
13	Melanjutkan pengembangan fitur klaim, termasuk pengaturan <i>field</i> yang fleksibel berdasarkan tipe klaim, serta dukungan upload file dengan penyimpanan ke MinIO.
14	Membangun fitur <i>submission</i> , termasuk tampilan pembuatan <i>submission</i> , relasi antar klaim, pengecekan <i>draft</i> , dan logika penyimpanan yang aman.
15	Mengimplementasikan alur <i>approval</i> multi-level berdasarkan konfigurasi struktur organisasi, dengan pengecekan peran pengguna dan pengiriman notifikasi otomatis.
16	Melakukan uji coba fungsionalitas sistem secara menyeluruh, melakukan <i>debugging</i> , perbaikan antarmuka, serta finalisasi seluruh fitur.

3.3 Uraian Pelaksanaan Magang

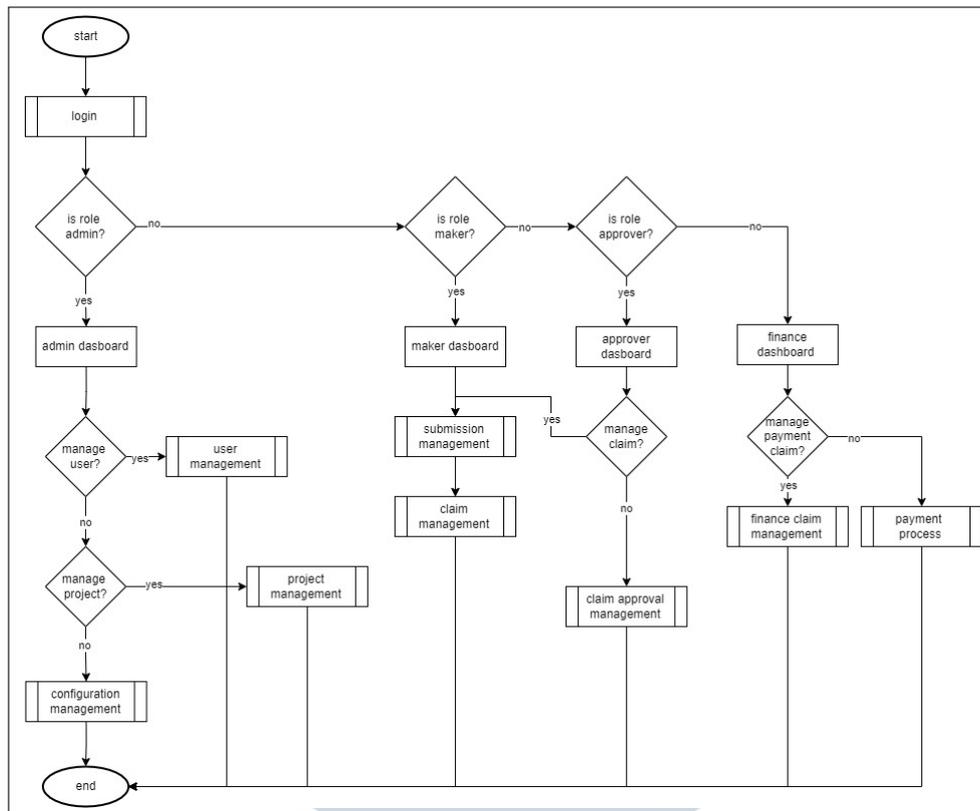
3.3.1 Perancangan Alur Proses (Flowchart)

Pada tahap ini, dilakukan perancangan alur sistem bisnis dalam bentuk *flowchart* untuk memberikan gambaran visual mengenai tahapan-tahapan yang dilakukan oleh pengguna dan sistem selama proses berlangsung.

Perancangan *flowchart* ini mengacu pada kebutuhan fungsional dan non-fungsional yang telah dirumuskan dan disepakati sebelumnya dalam dokumen *Functional Specification Document*, sehingga menjadi dasar utama dalam proses pengembangan sistem ke tahap selanjutnya.

A Flowchart Keseluruhan Sistem

Flowchart ini menjelaskan secara menyeluruh mengenai alur proses yang dapat dilakukan oleh masing-masing pengguna sesuai dengan peran yang dimilikinya dalam sistem. Seluruh proses ini dibagi ke dalam beberapa modul sesuai dengan fungsionalitas masing-masing, dan akan dijelaskan secara lebih rinci pada bagian-bagian berikutnya.



Gambar 3.2. *Flowchart* Keseluruhan Sistem

Pada Gambar 3.2, pengguna pertama-tama akan diarahkan ke modul *login*. Setelah proses autentikasi berhasil, sistem akan melakukan pengecekan peran pengguna.

Pengguna dengan peran *Admin* akan diarahkan menuju *Admin dashboard* yang pada bagian *sidebar* tersedia menu navigasi utama seperti "User Management", "Project Management", dan "Configuration Management". Masing-masing menu ini merepresentasikan modul yang berbeda. *Admin* juga memiliki fleksibilitas untuk memilih menu yang tersedia di *sidebar*, sehingga dapat dengan mudah berpindah antar modul sesuai kebutuhan. Hal ini memungkinkan *Admin* untuk mengelola beberapa modul dalam satu sesi kerja tanpa perlu melakukan proses *login* ulang atau *logout*.

Sementara itu, pengguna dengan peran *Maker* akan diarahkan menuju *Maker dashboard*, yang juga dilengkapi dengan *sidebar* sebagai navigasi utama. Pada bagian ini, tersedia menu seperti "Claim List" dan "Request Claim". Untuk dapat membuat klaim baru maupun menampilkan daftar klaim yang telah diajukan, *Maker* harus terlebih dahulu masuk ke dalam modul *Submission Management*. Proses ini bersifat wajib karena setiap klaim dalam sistem harus berada dalam satu

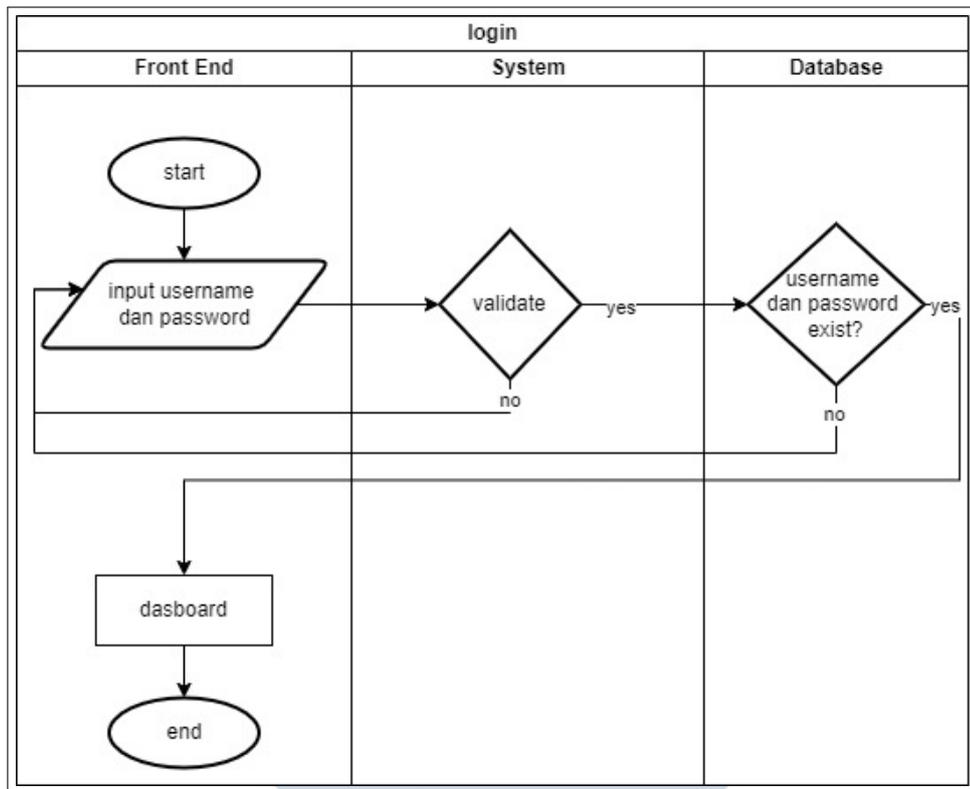
submission tertentu. Dalam sistem ini, satu *submission* dapat berisi satu atau lebih klaim, sehingga seluruh klaim selalu terkait dengan *submission*.

Sementara itu, pengguna dengan peran *Approver* akan diarahkan menuju *Approver dashboard*, yang pada bagian *sidebar* juga dilengkapi dengan menu navigasi seperti "Claim List", "Request Claim", "Approve Claim", dan "Approval Payment". Dalam sistem ini, seorang *Approver* bisa melakukan proses persetujuan klaim sekaligus berperan sebagai *Maker* dan mengajukan klaim sendiri. Namun, seorang *Maker* hanya bisa mengajukan klaim dan tidak bisa melakukan proses persetujuan klaim. Jika seorang *Approver* ingin mengajukan klaim, alur yang diikuti akan sama seperti pengguna dengan peran *Maker*, yaitu harus melalui modul *Submission Management* terlebih dahulu, kemudian dilanjutkan ke modul *Claim Management*. Sementara itu, apabila *Approver* ingin melakukan proses persetujuan terhadap klaim yang diajukan oleh pengguna lain, maka *Approver* akan diarahkan ke modul *Approval Claim Management*.

Terakhir, pengguna dengan peran *finance* akan diarahkan menuju *finance dashboard*, yang pada bagian *sidebar* menyediakan menu navigasi seperti "Claim List" dan "Request Payment". Jika *finance* ingin mengelola atau membuat klaim, maka alur yang diikuti adalah masuk ke modul *Finance Claim Management*. Sementara itu, untuk melakukan pengelolaan pembayaran atas klaim yang telah disetujui, *finance* akan diarahkan ke modul *Payment Process*.

B Flowchart Login

Proses *login* merupakan tahapan awal dan modul pertama yang harus dilalui oleh setiap pengguna sebelum dapat mengakses fitur-fitur yang tersedia sesuai dengan peran masing-masing. Tahapan ini berfungsi sebagai mekanisme autentikasi untuk memastikan bahwa hanya pengguna yang terdaftar dan memiliki kredensial yang valid yang dapat masuk ke dalam sistem. Setelah proses *login* berhasil, sistem akan mengarahkan pengguna ke *dashboard* yang sesuai dengan peran masing-masing.



Gambar 3.3. Flowchart Login Untuk Seluruh Pengguna

Pada Gambar 3.3, pengguna pertama-tama harus memasukkan informasi berupa *username* dan *password* yang sebelumnya telah didaftarkan. Sistem kemudian akan melakukan validasi data tersebut dengan mencocokkannya ke dalam basis data. Jika data yang dimasukkan sesuai, maka sistem akan memberikan akses ke halaman *dashboard* sesuai dengan peran pengguna. Namun, jika terjadi kesalahan, seperti *password* yang tidak sesuai atau *username* yang tidak ditemukan, maka sistem akan menampilkan *error message* dan pengguna akan diminta untuk mengulangi proses *login* dengan memasukkan informasi yang benar.

Proses *login* juga diintegrasikan dengan mekanisme keamanan tambahan guna meningkatkan perlindungan terhadap data pengguna. Salah satu langkah yang diterapkan adalah penggunaan algoritma enkripsi *Secure Hash Algorithm 256-bit* (SHA-256) untuk mengamankan *password* pengguna. Dengan metode ini, *password* yang dimasukkan oleh pengguna tidak disimpan dalam bentuk *plain text*, melainkan dalam bentuk hash yang tidak dapat dikembalikan ke bentuk semula. Hal ini bertujuan untuk menghindari potensi kebocoran informasi sensitif jika terjadi pelanggaran keamanan pada basis data.

Selain itu, sistem juga menggunakan token sesi berbasis *JSON Web Token*

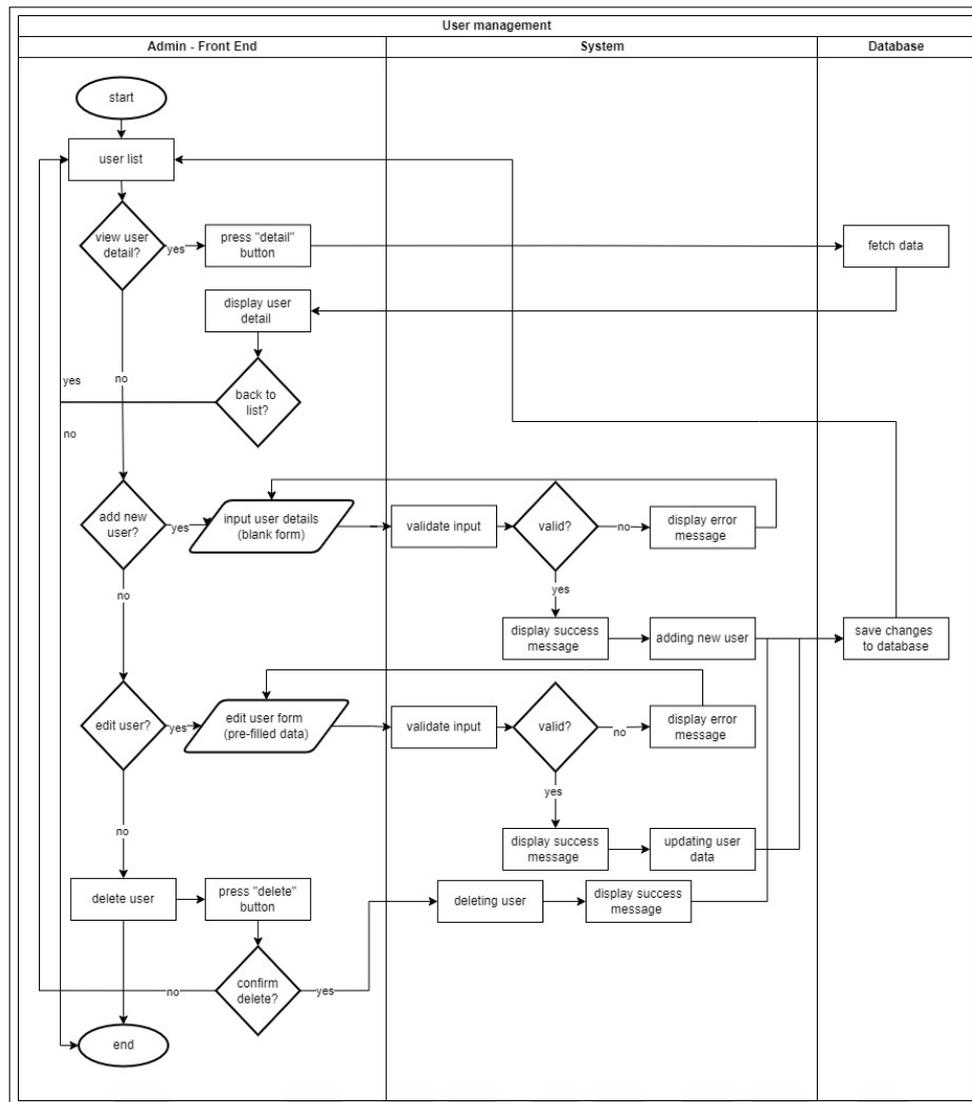
sebagai metode manajemen sesi autentikasi. Setelah pengguna berhasil *login*, sistem akan menghasilkan token yang dienkripsi dan dikirimkan kepada pengguna untuk disimpan di *local storage*.

Penggunaan *JSON Web Token* dipilih karena beberapa keunggulan, di antaranya adalah kemampuannya untuk menyimpan informasi penting seperti identitas pengguna secara aman di token. Dengan menggunakan *JSON Web Token*, sistem dapat mempertahankan sesi autentikasi selama token masih valid, sehingga pengguna tidak perlu melakukan *login* ulang setiap kali melakukan navigasi antar halaman atau modul, selama token belum kedaluwarsa atau dihapus secara manual.

C Flowchart User Management

Flowchart ini menggambarkan secara menyeluruh alur proses yang terjadi dalam modul *User Management* pada sistem. Modul ini memungkinkan *Admin* untuk melakukan pengelolaan penuh terhadap data pengguna yang terdaftar, mencakup aktivitas seperti menambahkan pengguna baru, melihat detail informasi pengguna, memperbarui data pengguna yang sudah ada, hingga menghapus pengguna yang sudah tidak aktif atau tidak diperlukan.





Gambar 3.4. Flowchart User Management

Dari Gambar 3.4, ketika Admin memilih menu *User Management*, sistem akan menampilkan daftar pengguna (*user list*). Apabila Admin ingin melihat informasi detail dari salah satu pengguna, cukup dengan menekan tombol "Detail" pada baris pengguna yang diinginkan. Sistem kemudian akan mengambil data pengguna tersebut dari basis data dan menampilkannya secara lengkap. Setelah itu, Admin dapat menekan tombol "Back" untuk kembali ke daftar pengguna untuk melihat atau mengelola data pengguna lainnya.

Apabila Admin ingin menambahkan pengguna baru, Admin dapat menekan tombol "Add User" yang akan mengarahkan ke halaman formulir input. Pada halaman ini, Admin harus mengisi seluruh *field* yang diperlukan, seperti nama,

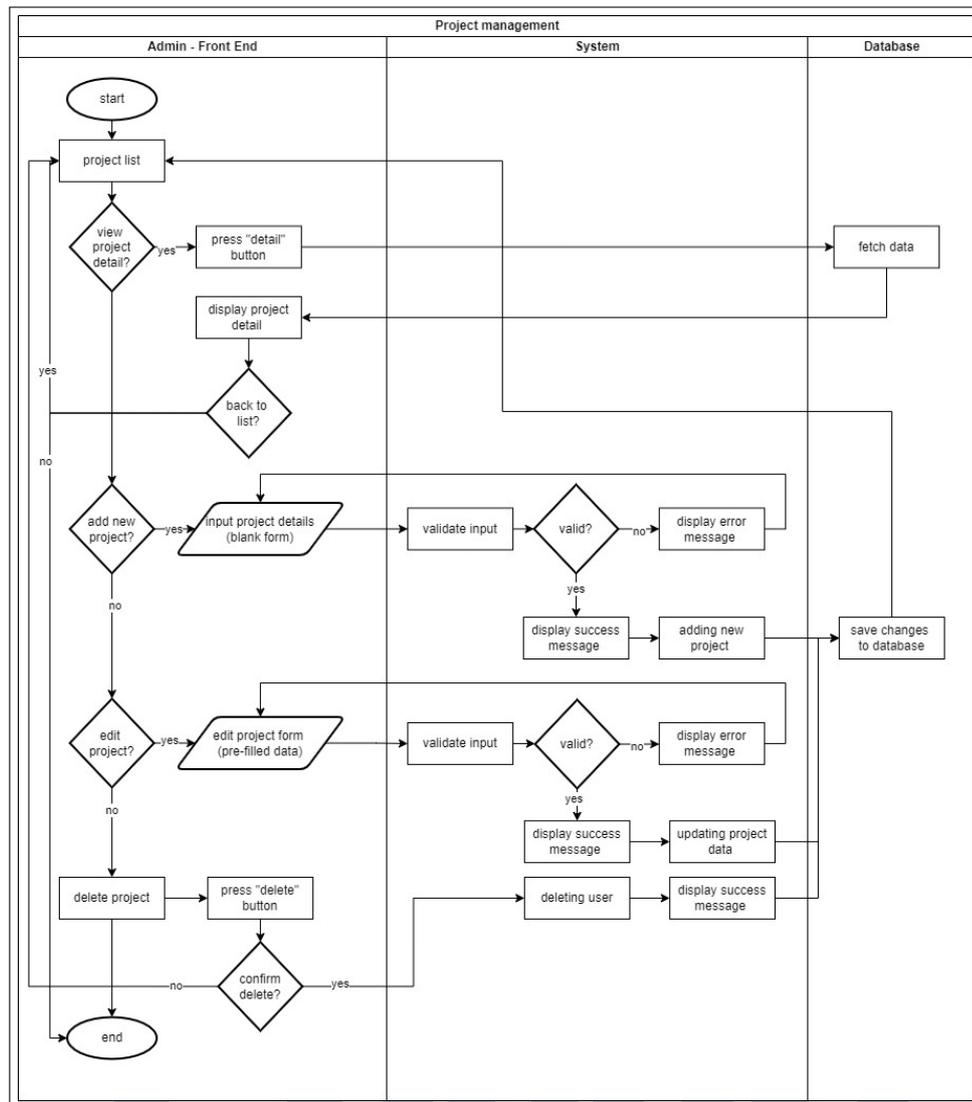
username, *password*, peran, dan informasi lain yang relevan. Setelah data diisi dan tombol simpan ditekan, sistem akan melakukan proses validasi untuk memastikan semua data wajib telah terisi dengan benar dan sesuai format. Jika terdapat kesalahan, sistem akan menampilkan *error message* yang menunjukkan bagian mana yang perlu diperbaiki. Sebaliknya, jika seluruh data valid, maka data pengguna akan disimpan ke dalam basis data dan sistem akan menampilkan *success message*. Setelah proses penyimpanan selesai, Admin akan secara otomatis diarahkan kembali ke halaman *user list*.

Jika *Admin* ingin melakukan perubahan pada data pengguna yang sudah ada, Admin dapat menekan tombol "Edit User" dan mengubah informasi yang diperlukan. Namun, terdapat beberapa data yang tidak dapat diubah karena memiliki dampak besar terhadap keseluruhan sistem. Setelah *Admin* menyelesaikan proses pengeditan, sistem akan melakukan validasi untuk memastikan bahwa data yang dimasukkan sudah benar dan lengkap. Jika validasi berhasil, sistem akan memperbarui data pengguna di basis data dan menampilkan *success message*. Sebaliknya, jika terdapat kesalahan seperti data tidak lengkap atau format tidak sesuai, sistem akan menampilkan *error message* yang sesuai. Setelah proses selesai, *Admin* akan diarahkan kembali ke halaman *user list*.

Terakhir, jika *Admin* ingin menghapus pengguna yang ada, *Admin* dapat menekan tombol "Delete", kemudian sistem akan menampilkan konfirmasi penghapusan. Jika *Admin* berubah pikiran, tombol "Cancel" dapat ditekan untuk membatalkan proses dan kembali ke halaman *user list*. Namun, jika *Admin* melanjutkan proses penghapusan, sistem akan menghapus data pengguna yang dipilih dari basis data, kemudian menampilkan pesan notifikasi berhasil setelah proses selesai.

D Flowchart Project Management

Flowchart ini menggambarkan secara menyeluruh alur proses yang terjadi dalam modul *Project Management* pada sistem. Modul ini memungkinkan *Admin* untuk melakukan pengelolaan terhadap data proyek yang terdaftar pada sistem. Pengelolaan tersebut mencakup berbagai aktivitas seperti menambahkan proyek baru, melihat detail informasi proyek, mengubah data proyek yang sudah ada, serta menghapus proyek yang tidak lagi diperlukan.



Gambar 3.5. Flowchart Project Management

Pada Gambar 3.5 dapat terlihat proses pengelolaan data pada modul *Project Management* pada dasarnya memiliki alur dan struktur yang serupa dengan modul *User Management* seperti yang telah dijelaskan pada Gambar 3.4. *Admin* dapat menekan tombol "Add Project" untuk menambahkan data proyek baru, kemudian mengisi seluruh *field* yang diperlukan. Setelah itu, sistem akan melakukan validasi untuk memastikan seluruh data sudah benar dan lengkap. Jika validasi berhasil, data akan disimpan ke dalam basis data dan sistem akan menampilkan *success message* serta mengarahkan kembali ke halaman *project list*. Sebaliknya, jika terjadi kesalahan atau ada data yang tidak valid, sistem akan menampilkan *error message* yang sesuai.

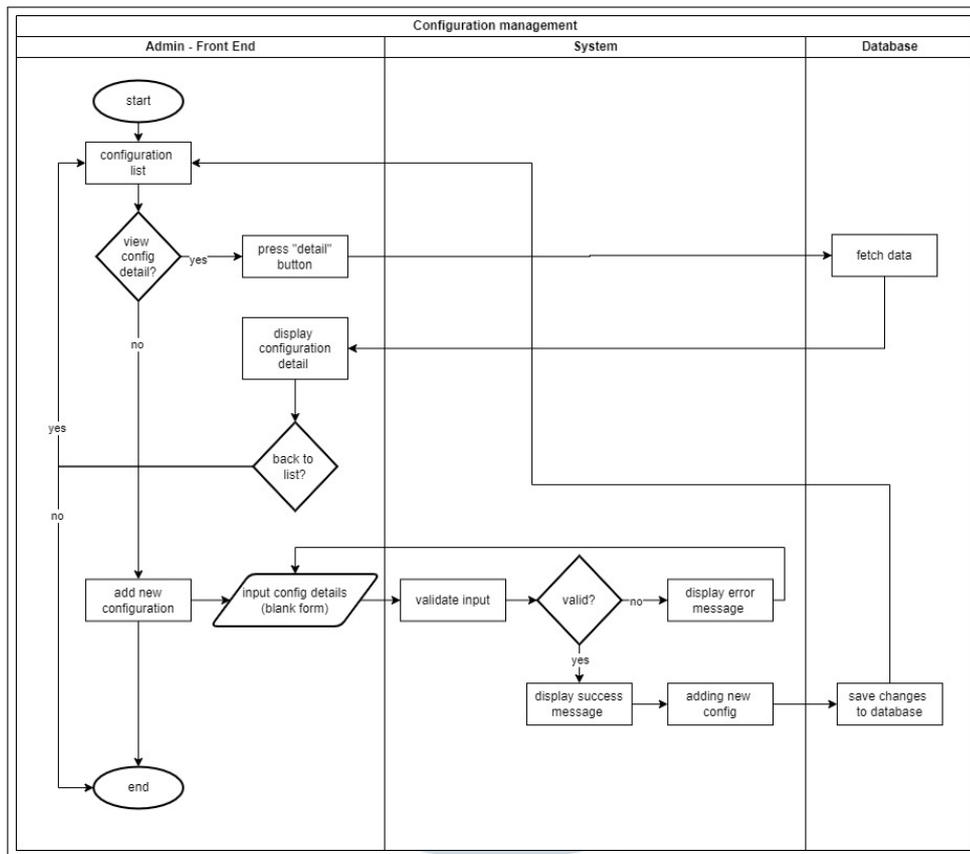
Untuk melihat informasi lebih detail, *Admin* cukup menekan tombol "Detail", sistem kemudian akan mengambil dan menampilkan informasi dari basis data. Jika diperlukan perubahan data, tombol "Edit Project" dapat digunakan untuk memodifikasi informasi yang sudah ada. Sistem akan kembali melakukan validasi sebelum menyimpan perubahan tersebut. Apabila berhasil, sistem akan menampilkan *success message* dan mengarahkan kembali ke halaman *project list*. Jika terdapat kesalahan validasi, sistem akan menampilkan *error message*.

Terakhir, apabila data proyek ingin dihapus, *Admin* dapat menekan tombol "Delete" dan melakukan konfirmasi penghapusan. Jika konfirmasi dilanjutkan, sistem akan menghapus data proyek secara permanen dari basis data dan menampilkan *success message*, kemudian mengarahkan *Admin* kembali ke halaman *project list*.

E Flowchart Configuration Management

Flowchart ini menggambarkan secara menyeluruh alur proses yang terjadi pada modul *Configuration Management* dalam sistem. Modul ini berfungsi untuk mengatur alur persetujuan berdasarkan tipe klaim yang berlaku, sehingga sistem dapat menyesuaikan proses *approval* secara dinamis sesuai kebutuhan perusahaan.





Gambar 3.6. Flowchart Configuration Management

Seperti terlihat pada Gambar 3.6, alur proses pada modul *Configuration Management* memiliki sedikit perbedaan dibandingkan dengan proses yang terdapat pada modul *User Management* dan *Project Management*. Meskipun ketiganya sama-sama diatur oleh *Admin*, namun pada modul "Configuration Management", hak akses yang diberikan bersifat lebih terbatas.

Admin tidak dapat secara langsung mengubah atau menghapus data konfigurasi melalui antarmuka sistem. Jika diperlukan perubahan terhadap data konfigurasi yang telah ada, maka proses tersebut harus dilakukan secara manual melalui basis data oleh pihak yang berwenang. Pembatasan ini diberlakukan untuk menjaga stabilitas dan konsistensi sistem secara keseluruhan, karena data konfigurasi bersifat sensitif dan berdampak langsung terhadap cara sistem berjalan.

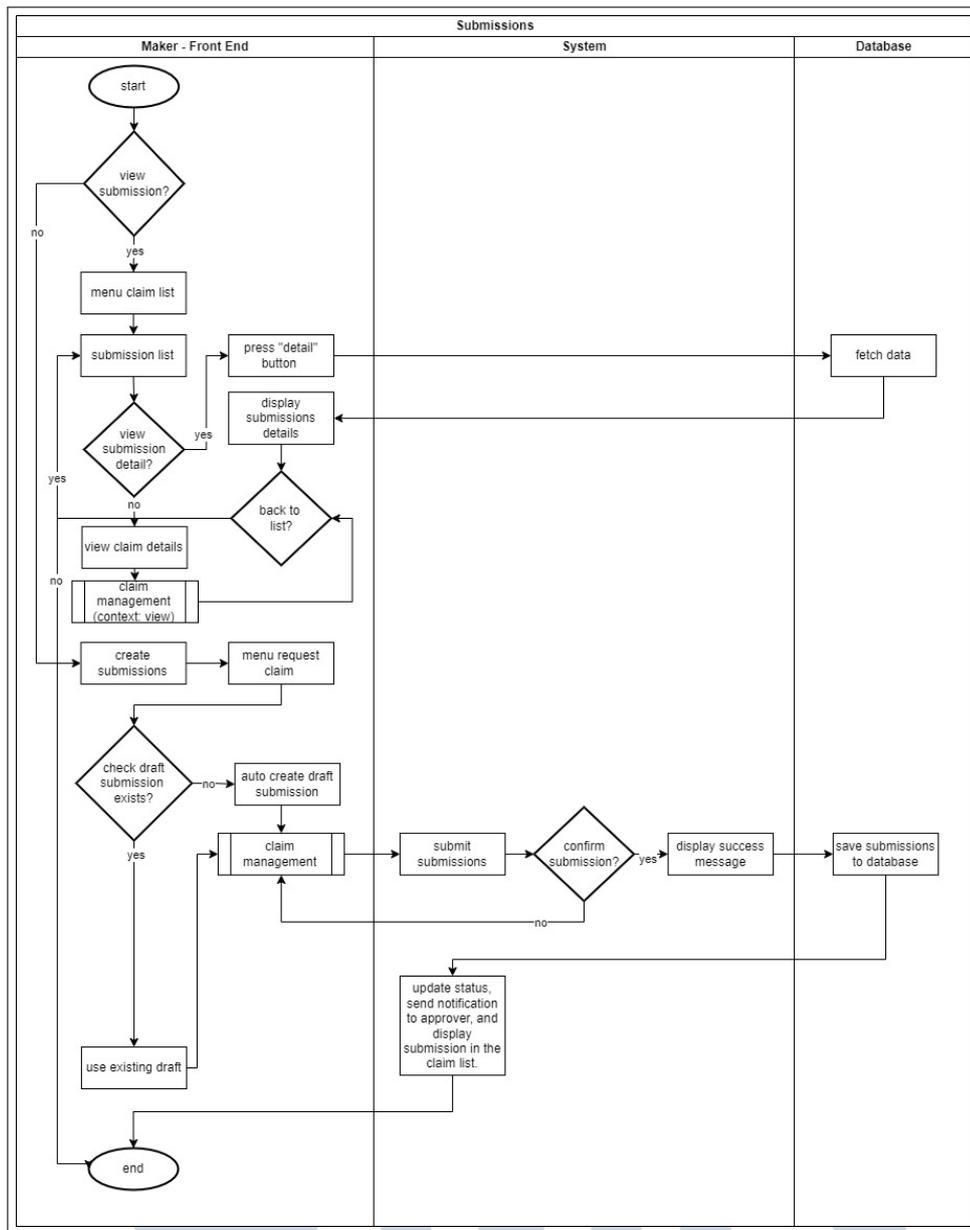
Namun demikian, jika *Admin* ingin menambahkan data konfigurasi baru atau melihat informasi konfigurasi yang sudah ada, langkah-langkahnya serupa seperti pada modul lainnya. *Admin* cukup menekan tombol "Add Configuration" untuk menambahkan data baru dan mengisi seluruh *field* yang diperlukan, kemudian sistem akan melakukan proses validasi sebelum menyimpan data ke dalam basis

data. Setelah berhasil, sistem akan menampilkan *success message* dan mengarahkan pengguna kembali ke daftar konfigurasi. Jika *Admin* hanya ingin melihat detail konfigurasi, cukup tekan tombol "Detail", dan sistem akan menampilkan informasi konfigurasi dari basis data.

F Flowchart Submission Management

Flowchart ini menggambarkan secara menyeluruh alur proses yang terjadi di modul *Submission Management* pada sistem. Alur ini mencakup berbagai langkah yang dilakukan oleh *Maker* saat berinteraksi dengan sistem, mulai dari pembuatan hingga pengiriman *submission*. Dengan adanya *flowchart* ini, *Maker* dapat memahami secara visual bagaimana proses *submission* dijalankan.





Gambar 3.7. Flowchart Pembuatan Submission

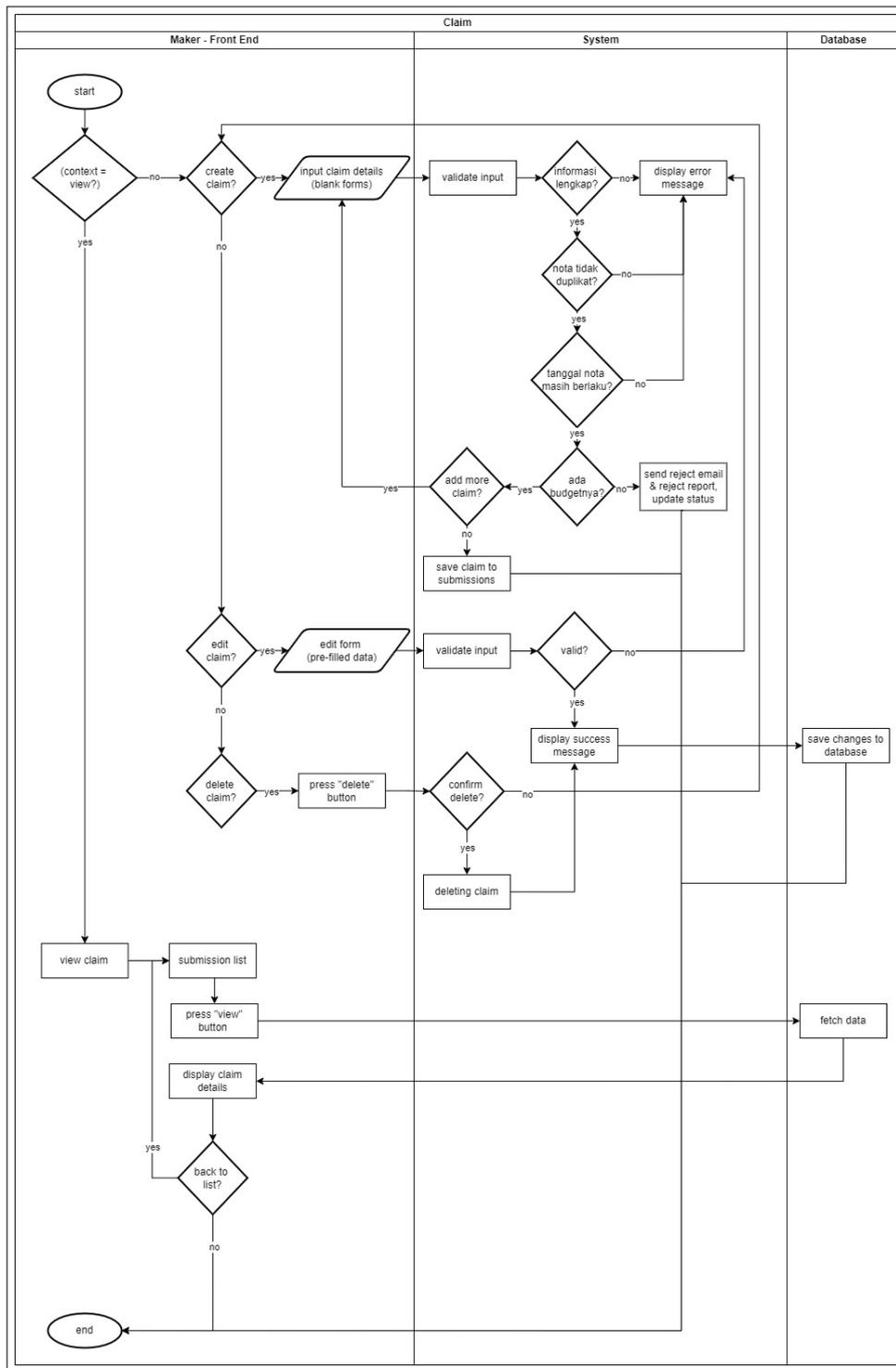
Pada Gambar 3.7, Jika *Maker* ingin melihat informasi lebih detail dari salah satu *submission* yang tersedia, *Maker* bisa memilih menu "Claim List". Sistem akan menampilkan *submission list* yang pernah dibuat. Pada *submission list* tersedia 2 tombol, yaitu "View" dan "Detail". Tombol "View" digunakan untuk menampilkan informasi detail dari setiap klaim yang terdapat dalam *submission* tersebut. Sedangkan tombol "Detail" digunakan untuk melihat detail *submission* yang pernah diajukan. Jika *Maker* menekan tombol "View", sistem akan mengarahkan *Maker* ke modul *Claim Management* dalam konteks *view-only*.

Maker juga dapat membuat sebuah *submission* melalui menu "Request Claim". Saat menu ini dibuka, sistem terlebih dahulu akan memeriksa apakah terdapat *submission* dengan status *Draft* di basis data. Status ini disediakan untuk menghindari risiko kehilangan data, seperti ketika *Maker* menutup halaman secara tidak sengaja. Jika *submission draft* ditemukan, sistem akan membuka kembali data tersebut agar *Maker* dapat melanjutkan pengisian. Namun, jika tidak ditemukan, sistem akan secara otomatis membuat *submission* baru. Setelah *submission* tersedia, *Maker* akan diarahkan ke modul *Claim Management* untuk proses pembuatan klaim. Saat proses pengisian telah selesai, *Maker* dapat menekan tombol "Submit". Sebelum data dikirim, sistem akan menampilkan pesan konfirmasi. Jika *Maker* memilih untuk membatalkan proses ini, maka sistem akan mengarahkan kembali ke modul *Claim Management*. Namun, jika konfirmasi dilanjutkan, maka sistem akan menjalankan proses validasi, menampilkan *success message* apabila data berhasil disimpan, dan memperbarui status *submission* di basis data. Selanjutnya, notifikasi akan dikirim secara otomatis kepada *Approver* yang ditentukan, dan *submission* tersebut akan muncul dalam menu "Claim List" pengguna.

Submission tidak dapat diubah maupun dihapus setelah proses *submit* dilakukan. Apabila terjadi kesalahan dalam pengisian atau pengajuan *submission*, maka *Maker* harus menghubungi *Approver* untuk menolak *submission* tersebut. Setelah *submission* ditolak, *Maker* dapat membuat *submission* baru yang sesuai dengan data yang benar.

G Flowchart Claim Management oleh Maker

Flowchart ini menggambarkan secara menyeluruh alur proses yang terjadi di modul *Claim Management* pada sistem. Modul ini menjadi bagian penting dalam pengelolaan klaim yang dilakukan oleh *Maker*, mulai dari pembuatan klaim baru, validasi, hingga penyimpanan data. Dengan adanya *flowchart* ini, pengguna dapat memahami tahapan yang harus dilalui dan sistem dapat memastikan bahwa proses berjalan sesuai aturan yang telah ditentukan.



Gambar 3.8. Flowchart Pembuatan Klaim

Pada Gambar 3.8, sistem terlebih dahulu akan memeriksa apakah *Maker* mengakses modul ini dari *Submission Management* dengan konteks *view-only*. Jika ya, maka *Maker* cukup menekan tombol "View", dan sistem akan mengambil serta menampilkan data klaim yang sesuai dari basis data. Setelah selesai, *Maker* dapat kembali ke *submission list* untuk melihat detail klaim dari *submission* lainnya.

Maker juga bisa membuat klaim baru. Pertama-tama *Maker* bisa mengakses menu "Request Claim" kemudian menekan tombol "Create Claim". Selanjutnya, *Maker* mengisi detail klaim sesuai *field* yang tersedia. Terdapat tiga jenis klaim yang dapat dibuat oleh *Maker*, yaitu *Personal Claim*, *Project Claim*, dan *Company Claim*. Dalam satu *submission*, *Maker* hanya diperbolehkan membuat klaim dari satu jenis yang sama, sehingga tidak dimungkinkan untuk mencampur klaim dari jenis yang berbeda dalam satu *submission*. Sistem kemudian akan melakukan proses validasi. Proses validasi ini mencakup beberapa tahap, antara lain: pengecekan kelengkapan informasi, pemeriksaan apakah nota yang diunggah merupakan duplikat, validitas tanggal nota, serta ketersediaan anggaran untuk kategori klaim yang dipilih. Jika sistem gagal memvalidasi pada salah satu tahap validasi tersebut, maka sistem akan menampilkan *error message*. Apabila kegagalan terjadi pada tahap validasi akhir, sistem juga akan mengirimkan *reject report* dan memperbarui status klaim di basis data. Jika semua tahapan validasi berhasil dilewati, maka klaim akan disimpan sementara dalam *submission*. *Maker* dapat menambahkan klaim tambahan jika diperlukan, dengan mengulangi proses yang sama. Setelah seluruh klaim berhasil dibuat, *Maker* akan diarahkan kembali ke modul *Submission Management* untuk melakukan proses submit.

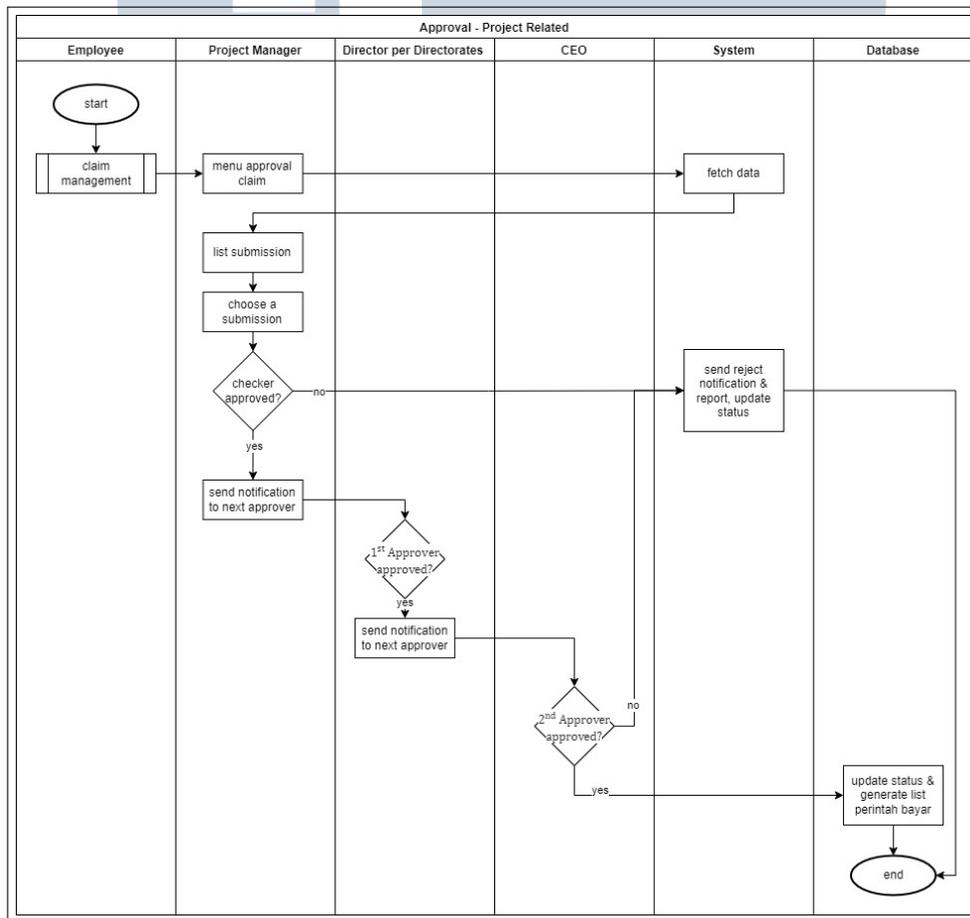
Pada Gambar 3.8 juga dijelaskan tahapan yang dilakukan untuk mengubah data klaim yang telah dibuat. Perlu dicatat bahwa klaim hanya dapat diubah dan dihapus apabila *submission* belum di-submit. Jika *submission* sudah dikirim dan data telah disimpan ke dalam basis data, maka perubahan data klaim tidak dapat dilakukan lagi. Pertama-tama, *Maker* dapat menekan tombol "Edit" pada klaim yang ingin diubah, kemudian melakukan perubahan terhadap data yang diperlukan. Perlu dicatat bahwa terdapat beberapa *field* yang tidak dapat diubah karena dapat memengaruhi keseluruhan sistem. Setelah *Maker* menyelesaikan pengeditan, sistem akan melakukan proses validasi, dan apabila validasi berhasil, sistem akan memperbarui data klaim serta menyimpan perubahan tersebut ke dalam basis data.

Selain mengedit, *Maker* juga memiliki opsi untuk menghapus klaim yang telah dibuat. Penghapusan dilakukan dengan menekan tombol "Delete", kemudian

melakukan konfirmasi penghapusan. Setelah dikonfirmasi, sistem akan menghapus klaim dari basis data.

H Flowchart Claim Approval untuk Klaim yang Terkait Proyek

Flowchart ini menggambarkan secara menyeluruh alur proses *approval* untuk klaim yang berkaitan dengan proyek. Setelah *Maker* mengirimkan *submission* yang berisi satu atau beberapa klaim, sistem akan mengirimkan notifikasi kepada pihak terkait untuk melakukan *approval* sesuai urutan yang ditentukan.



Gambar 3.9. *Flowchart* Proses *Approval* untuk Klaim yang Terkait Proyek

Pada klaim yang berkaitan dengan proyek, proses *approval* akan mengikuti struktur hierarki yang telah ditetapkan dalam proyek tersebut. Pada Gambar 3.9, dapat dilihat proses persetujuan dimulai dari pengguna dengan peran sebagai Project Manager yang bertindak sebagai *Checker*. Project Manager dapat

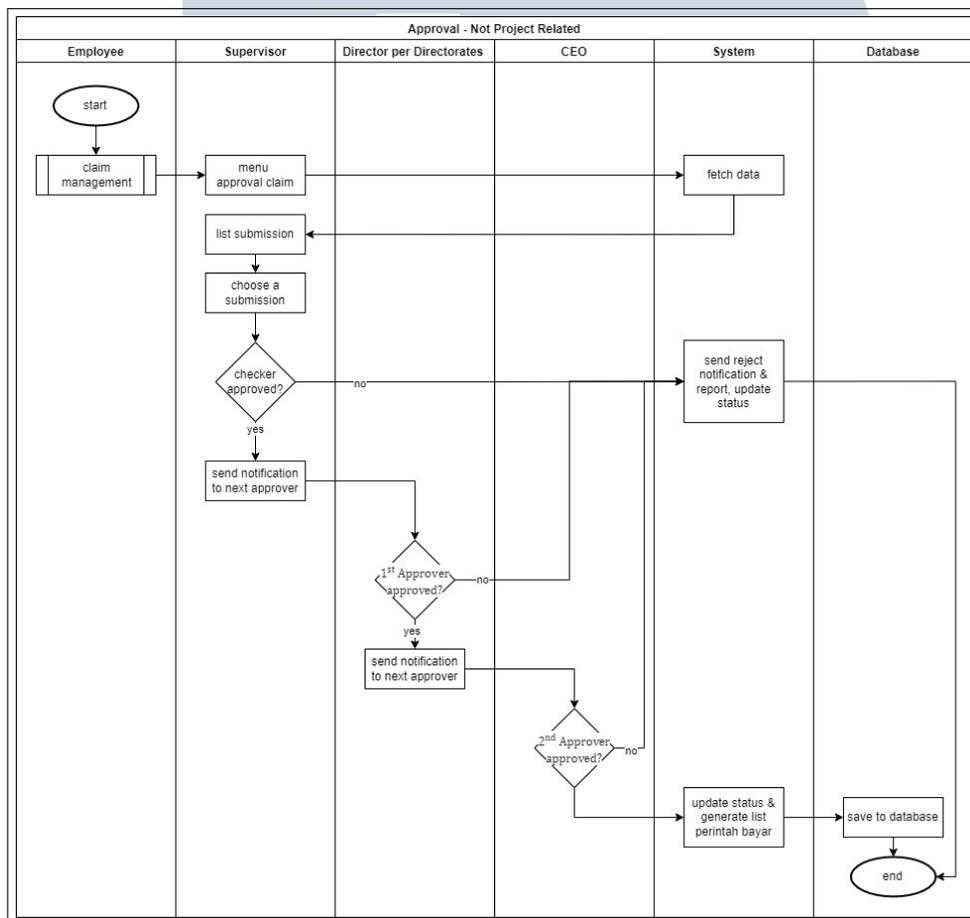
mengakses menu "Approval Claim", kemudian sistem akan menampilkan daftar *submission* yang membutuhkan persetujuan. Selanjutnya, Project Manager dapat memilih salah satu *submission* dan memberikan keputusan untuk menyetujui atau menolak klaim tersebut.

Jika klaim ditolak, sistem akan mengirimkan notifikasi penolakan beserta alasan penolakan kepada pembuat klaim (*Maker*). Status klaim akan diperbarui menjadi **Rejected** dan proses persetujuan untuk klaim tersebut akan dihentikan. Apabila *Maker* ingin mengajukan kembali klaim yang telah ditolak, maka *Maker* perlu membuat dan mengirimkan *submission* baru. Namun, jika klaim disetujui, sistem akan mengirimkan notifikasi kepada *Approver* berikutnya, dan proses persetujuan akan diteruskan sesuai dengan urutan yang telah ditentukan. Proses persetujuan kemudian dilanjutkan ke *Approver* pertama, yaitu Director. Sama seperti sebelumnya, pengguna dengan peran sebagai Director dapat mengakses menu "Approval Claim" untuk melihat daftar *submission* yang membutuhkan persetujuan. Director dapat memilih salah satu *submission* dan memutuskan untuk menyetujui atau menolak klaim tersebut. Jika klaim ditolak, sistem akan mengirimkan notifikasi penolakan beserta laporan alasan penolakan kepada pembuat klaim (*Maker*). Status klaim akan diperbarui menjadi **Rejected**, dan proses persetujuan akan dihentikan. Apabila *Maker* ingin mengajukan kembali klaim yang telah ditolak, maka ia harus membuat *submission* baru. Namun, jika klaim disetujui, sistem akan mengirimkan notifikasi kepada *Approver* berikutnya, dan proses persetujuan akan dilanjutkan sesuai dengan urutan yang telah ditentukan.

Setelah disetujui oleh Director, proses persetujuan dilanjutkan ke *Approver* terakhir, yaitu CEO. Pengguna dengan peran sebagai CEO akan menerima notifikasi terkait klaim yang telah melewati tahap persetujuan sebelumnya. Sama seperti tahap-tahap sebelumnya, CEO dapat mengakses menu "Approval Claim", meninjau detail *submission* yang diajukan, lalu memberikan keputusan akhir untuk menyetujui atau menolak klaim. Jika CEO menolak klaim, sistem akan mengirimkan notifikasi penolakan dan laporan alasan penolakan kepada pembuat klaim, status klaim akan diperbarui menjadi **Rejected**, dan proses persetujuan dihentikan. Jika CEO menyetujui klaim, maka proses persetujuan dianggap selesai, dan status klaim akan diperbarui menjadi **Approved**, dan sistem secara otomatis akan menghasilkan daftar perintah bayar berdasarkan klaim yang telah disetujui. Data klaim yang telah disetujui akan disimpan di basis data, dan *Maker* dapat melihat status akhir klaim melalui menu "Claim List".

I Flowchart Claim Approval dengan Maker dari Karyawan (Employee)

Flowchart ini menjelaskan alur proses persetujuan untuk klaim yang diajukan oleh pengguna dengan peran sebagai Employee. Perbedaan utama dibandingkan dengan persetujuan klaim yang terkait dengan proyek terletak pada urutan persetujuannya. Proses persetujuan klaim non-proyek mengikuti struktur hierarki perusahaan secara umum, bukan struktur proyek.



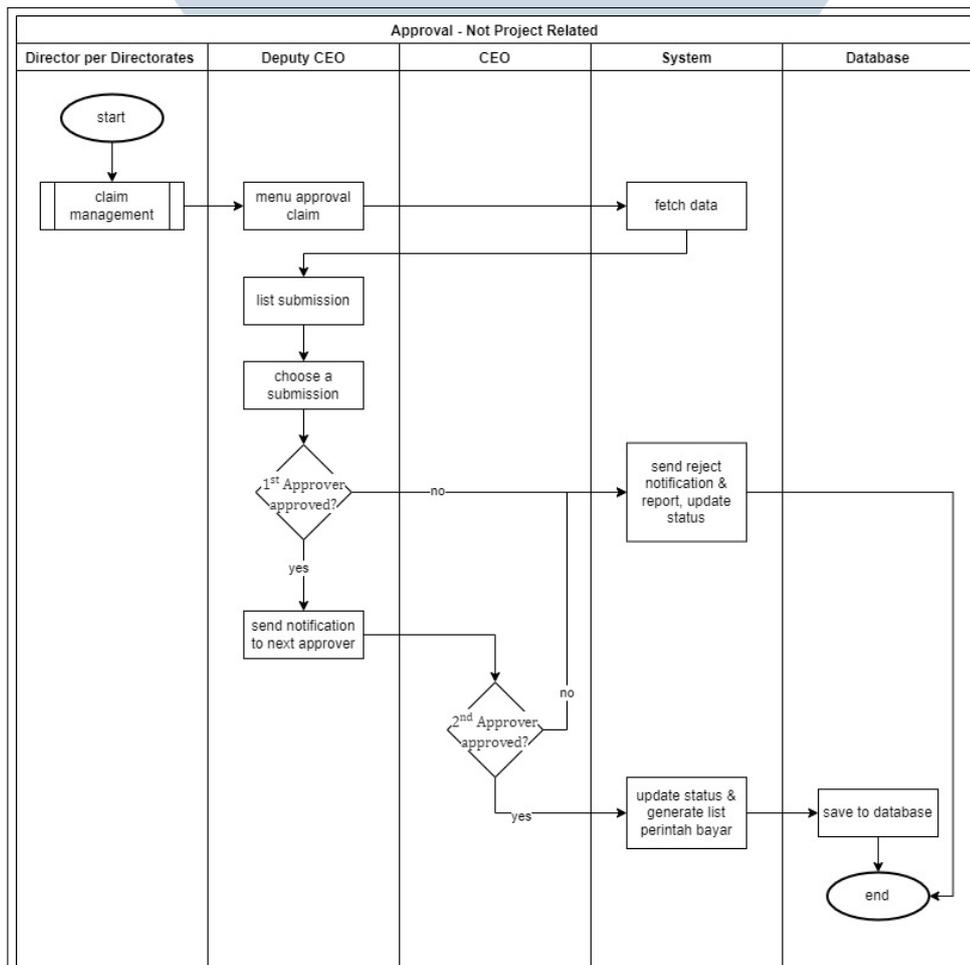
Gambar 3.10. *Flowchart* Proses *Approval* dengan *Maker* dari Karyawan

Pada Gambar 3.10, dapat dilihat bahwa proses persetujuan dimulai dari pengguna dengan peran sebagai Supervisor yang bertindak sebagai *Checker*. Selanjutnya, proses dilanjutkan oleh Director sebagai *First Approver*, dan diakhiri oleh CEO sebagai *Second Approver*. Sama seperti alur persetujuan sebelumnya, apabila klaim ditolak pada salah satu tahap persetujuan, maka sistem akan secara otomatis mengirimkan notifikasi penolakan beserta laporan alasan penolakan (*reject report*) kepada pembuat klaim (*Maker*), serta melakukan pembaruan status klaim

menjadi **Rejected**. Setelah itu, proses persetujuan akan dihentikan. Apabila *Maker* ingin mengajukan kembali klaim yang telah ditolak, maka ia harus membuat *submission* baru. Namun, apabila klaim disetujui pada seluruh tahap, maka status klaim akan diperbarui menjadi **Approved**, dan sistem akan secara otomatis menghasilkan daftar perintah pembayaran untuk diproses lebih lanjut.

J Flowchart Claim Approval dengan Maker dari Direktur (Director)

Flowchart ini menjelaskan secara rinci proses persetujuan klaim yang diajukan oleh pengguna dengan peran sebagai Direktur. Dalam konteks ini, Direktur bertindak sebagai *Maker*, yang berarti ia merupakan pihak yang mengajukan klaim ke dalam sistem. *Flowchart* ini dirancang untuk memberikan pemahaman yang jelas mengenai tahapan persetujuan yang berlaku khusus untuk kasus tersebut.



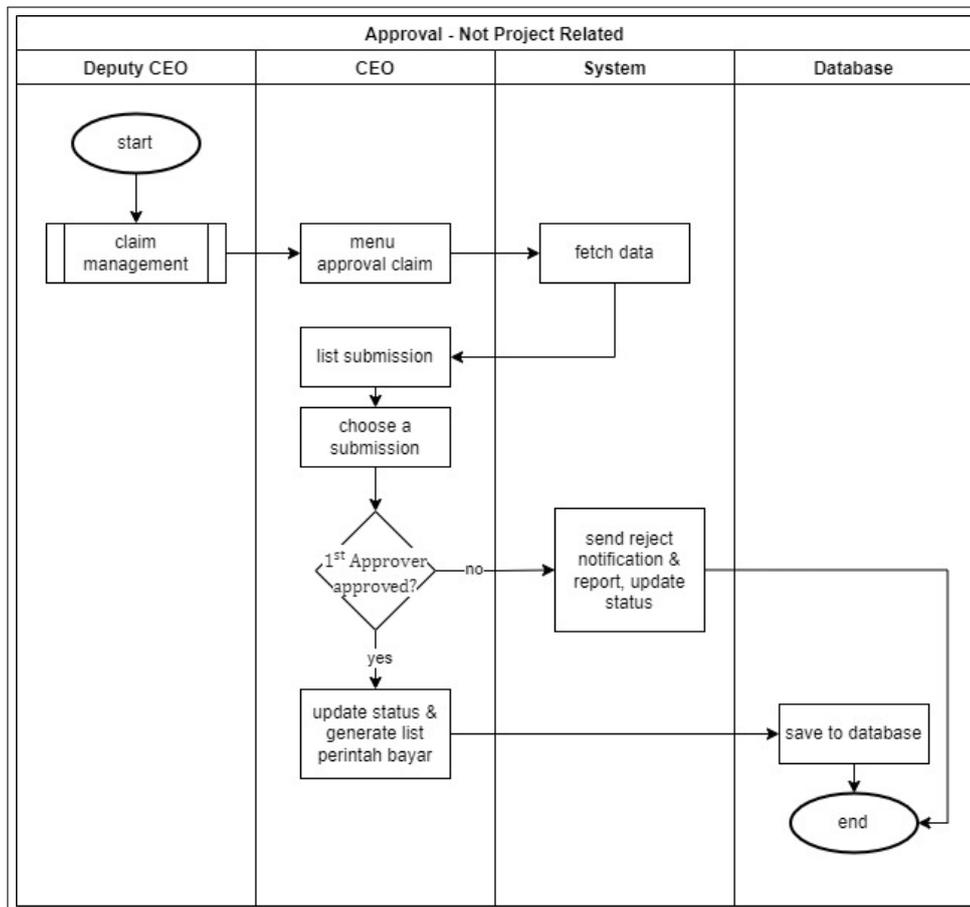
Gambar 3.11. *Flowchart* Proses Persetujuan dengan Maker dari Direktur

Pada Gambar 3.11, dapat dilihat bahwa tidak terdapat peran *Checker* dalam alur persetujuan ini. Proses persetujuan langsung dimulai dari Deputy CEO sebagai *First Approver*, kemudian dilanjutkan oleh CEO sebagai *Second Approver*. Sama seperti alur persetujuan sebelumnya, apabila klaim ditolak pada salah satu tahap persetujuan, maka sistem akan secara otomatis mengirimkan notifikasi penolakan beserta laporan alasan penolakan (*reject report*) kepada pembuat klaim (*Maker*), serta melakukan pembaruan status klaim menjadi **Rejected**. Setelah itu, proses persetujuan akan dihentikan. Apabila *Maker* ingin mengajukan kembali klaim yang telah ditolak, maka ia harus membuat *submission* baru. Namun, apabila klaim disetujui pada seluruh tahap, maka status klaim akan diperbarui menjadi **Approved**, dan sistem akan secara otomatis menghasilkan daftar perintah pembayaran untuk diproses lebih lanjut.

K Flowchart Claim Approval dengan Maker dari Wakil CEO (Deputy CEO)

Flowchart ini menjelaskan secara rinci proses persetujuan klaim yang diajukan oleh pengguna dengan peran sebagai Wakil CEO (Deputy CEO). Dalam situasi ini, Wakil CEO bertindak sebagai pembuat klaim (*Maker*), sehingga sistem perlu mengikuti alur persetujuan khusus yang menyesuaikan dengan struktur organisasi dan kewenangan yang berlaku.





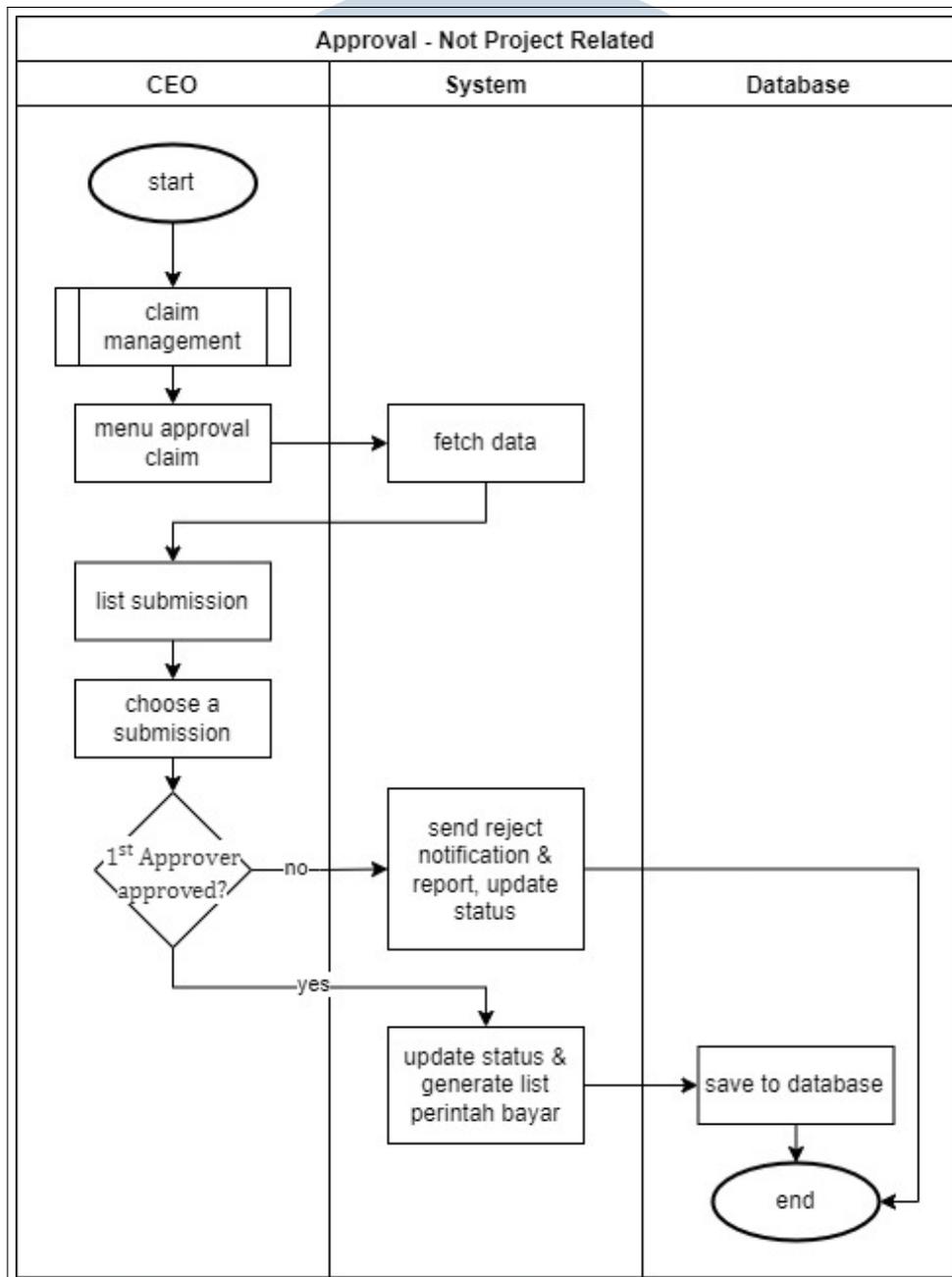
Gambar 3.12. Flowchart Proses Persetujuan dengan *Maker* dari Wakil CEO

Pada Gambar 3.12, dapat dilihat bahwa tidak terdapat peran *Checker* dalam alur persetujuan ini. Proses persetujuan hanya melibatkan satu *Approver*, yaitu CEO. Jika CEO menolak klaim, maka sistem akan secara otomatis mengirimkan notifikasi penolakan beserta laporan alasan penolakan (*reject report*) kepada pembuat klaim (*Maker*), serta memperbarui status klaim menjadi **Rejected**. Setelah itu, proses persetujuan akan dihentikan. Apabila *Maker* ingin mengajukan kembali klaim yang telah ditolak, maka ia harus membuat *submission* baru. Namun, apabila klaim disetujui pada seluruh tahap persetujuan, maka status klaim akan diperbarui menjadi **Approved**, dan sistem akan secara otomatis menghasilkan daftar perintah pembayaran (*payment instruction list*) untuk diproses lebih lanjut.

L Flowchart Claim Approval dengan *Maker* dari CEO

Flowchart ini menjelaskan alur persetujuan klaim apabila klaim tersebut diajukan langsung oleh CEO. Karena posisi CEO merupakan otoritas tertinggi

dalam struktur organisasi, sistem tidak lagi melibatkan peran lain dalam proses persetujuan klaim.



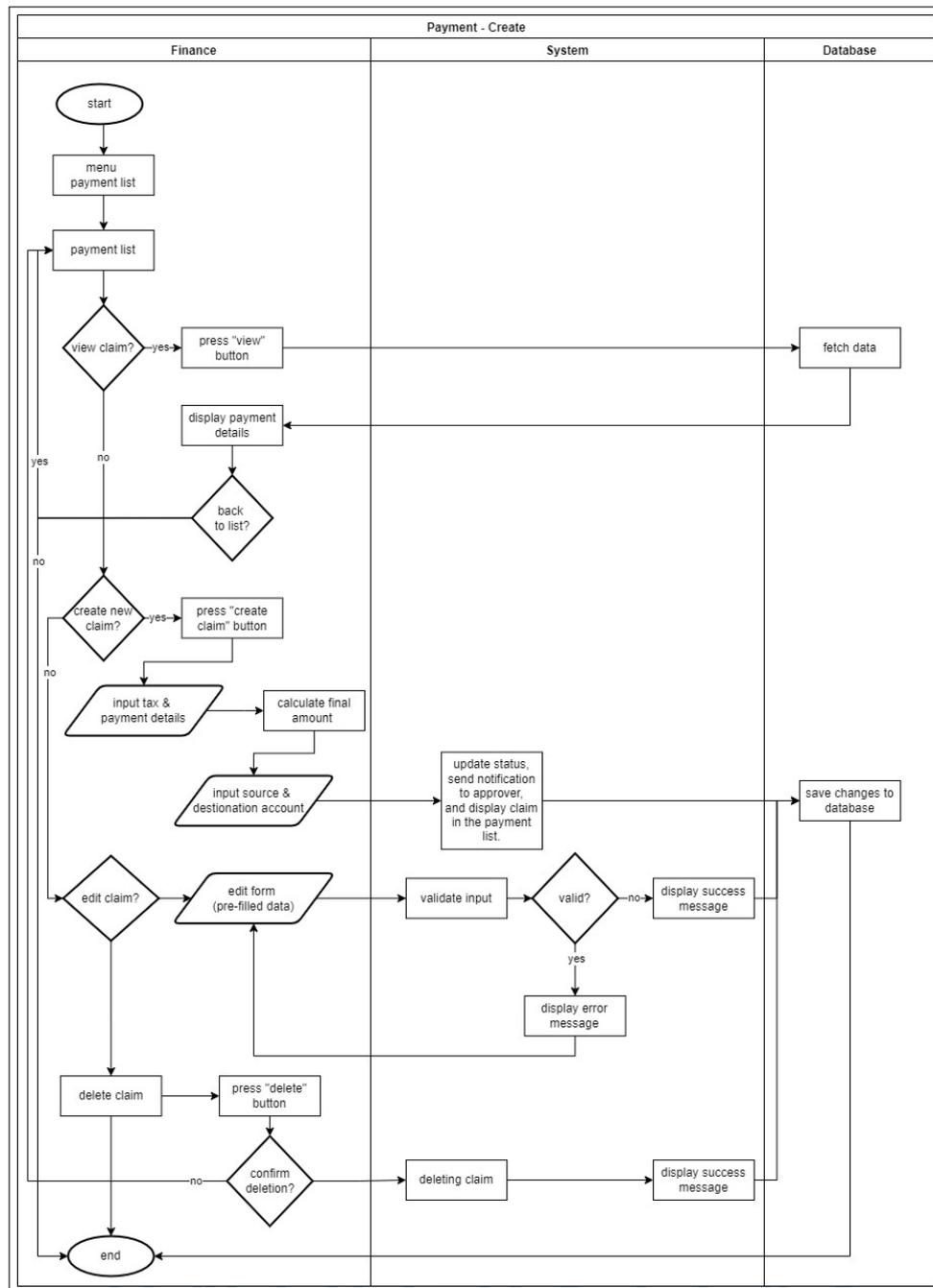
Gambar 3.13. Flowchart Proses Persetujuan dengan *Maker* dari CEO

Pada Gambar 3.13, dapat dilihat bahwa alur persetujuan ini tidak melibatkan peran lain selain CEO itu sendiri. Dalam hal ini, CEO bertindak sekaligus sebagai pembuat (*Maker*) dan pemberi persetujuan (*Approver*).

M Flowchart Finance Claim Management

Flowchart ini menjelaskan alur pembuatan klaim oleh pengguna dengan peran sebagai *Finance*. Setelah klaim yang diajukan oleh *Maker* disetujui oleh seluruh *Approver* sesuai alur yang telah ditentukan, maka sistem akan secara otomatis menghasilkan daftar perintah pembayaran (*payment instruction list*). Daftar ini akan muncul pada menu khusus yang hanya dapat diakses oleh pengguna dengan peran sebagai *Finance*. Selanjutnya, pengguna dengan peran sebagai *Finance* akan membuat klaim pembayaran berdasarkan daftar perintah bayar yang telah dihasilkan oleh sistem.





Gambar 3.14. Flowchart Pembuatan Klaim oleh Finance

Pada Gambar 3.14, dijelaskan tahapan pembuatan klaim pembayaran oleh pengguna dengan peran sebagai Finance. Proses dimulai dengan membuka menu "Request Payment", kemudian menekan tombol "Create Claim". Setelah itu, Finance akan mengisi informasi terkait pajak dan detail pembayaran sesuai dengan field yang tersedia pada formulir. Sistem secara otomatis akan menghitung jumlah

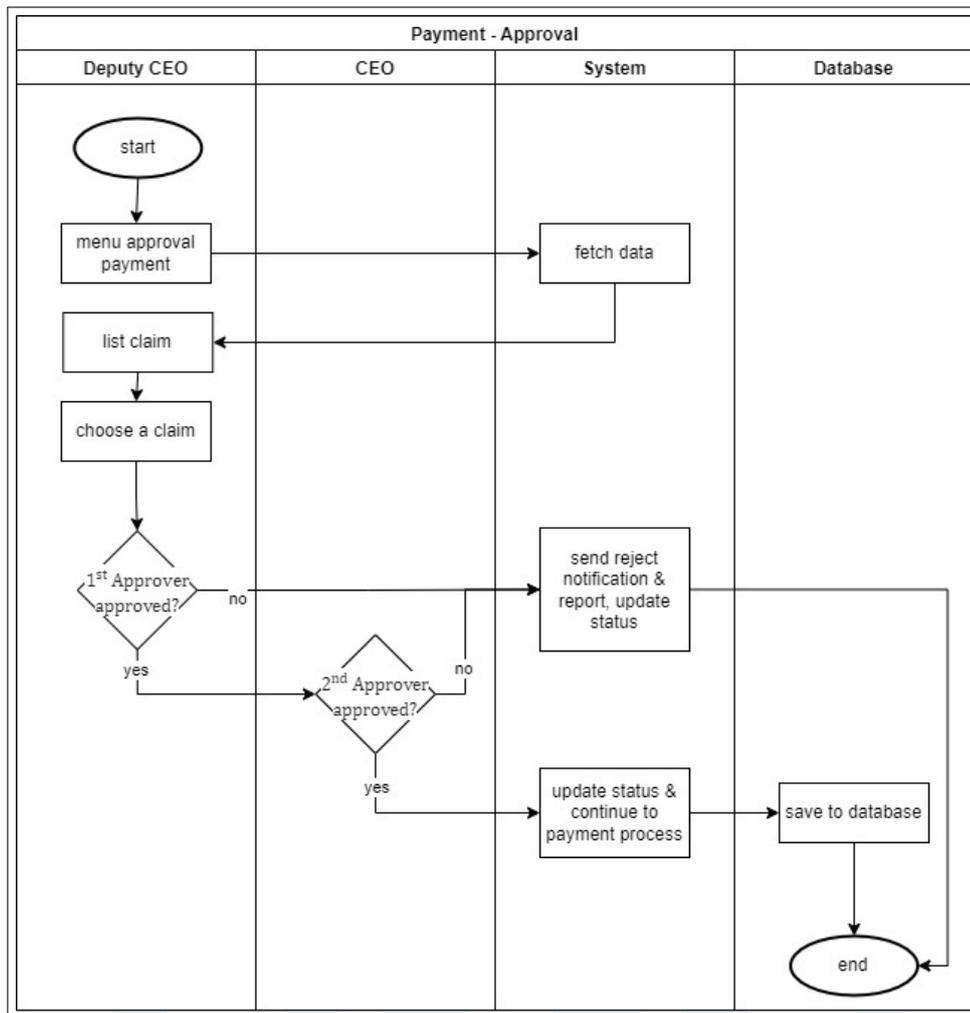
akhir yang perlu dibayarkan berdasarkan data yang telah diinput. Selanjutnya, *Finance* mengisi informasi rekening, yaitu *source account* dan *destination account* untuk pembayaran tersebut. Setelah seluruh data dinyatakan valid, sistem akan memperbarui status klaim, mengirimkan notifikasi kepada *Approver* yang ditentukan, menyimpan data klaim ke dalam basis data, dan menampilkan klaim tersebut pada menu "Payment List". Jika *Finance* ingin melihat daftar klaim yang telah dibuat, *Finance* dapat mengakses menu "List Approval" dan menekan tombol "View" pada salah satu klaim. Sistem kemudian akan mengambil data klaim dari basis data dan menampilkannya.

Pada Gambar 3.14, dapat dilihat bahwa klaim yang telah dibuat oleh pengguna dengan peran sebagai *Finance* juga dapat diedit. Proses pengeditan dilakukan dengan mengubah data pada formulir klaim yang telah disediakan. Setelah dilakukan perubahan, sistem akan melakukan validasi terhadap data yang diubah dan kemudian memperbarui data tersebut di basis data. Perlu diperhatikan bahwa terdapat beberapa *field* yang tidak dapat diubah karena berpengaruh terhadap keseluruhan proses sistem. Selain itu, *Finance* juga memiliki opsi untuk menghapus klaim yang telah dibuat. Proses penghapusan cukup dengan menekan tombol "Delete" dan melakukan konfirmasi penghapusan, setelah itu sistem akan memperbarui data di basis data dan menghapus klaim yang bersangkutan.

N Flowchart Claim Approval untuk Klaim oleh Finance

Flowchart ini menjelaskan alur proses persetujuan untuk klaim yang diajukan oleh pengguna dengan peran sebagai *Finance*. Alur persetujuannya secara umum mirip dengan proses sebelumnya, klaim yang telah dibuat dan disimpan ke dalam sistem akan diteruskan kepada pihak-pihak yang berwenang untuk melakukan persetujuan sesuai dengan urutan yang telah ditentukan.

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

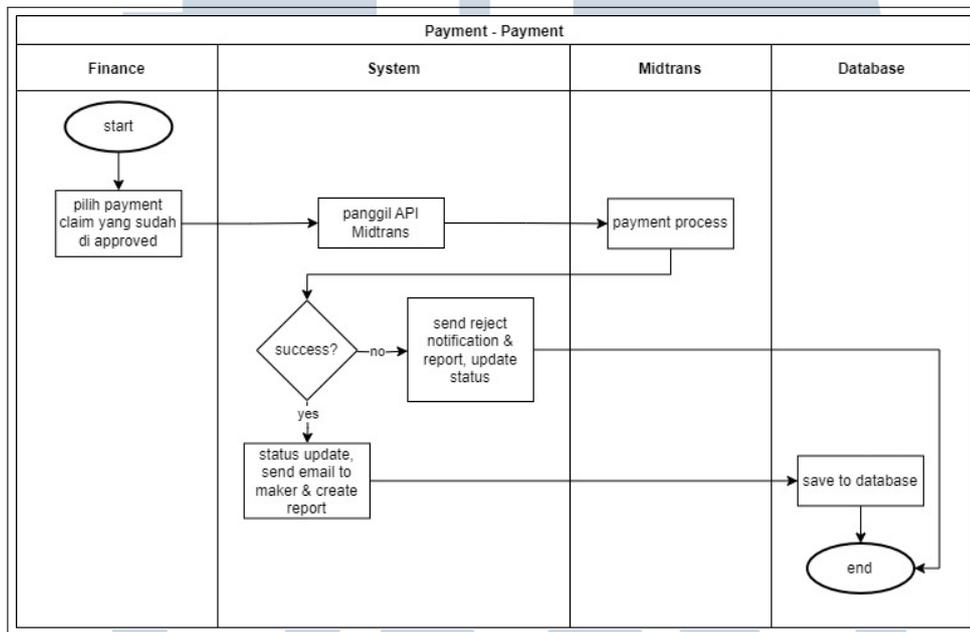


Gambar 3.15. Flowchart Proses Persetujuan untuk Klaim oleh Finance

Pada Gambar 3.15, diperlihatkan bahwa alur persetujuan untuk klaim yang diajukan oleh Finance dimulai dari Deputy CEO sebagai *First Approver*, kemudian dilanjutkan ke CEO sebagai *Second Approver*. Proses ini mengikuti pola persetujuan berjenjang seperti alur-alur sebelumnya, setiap *Approver* memiliki wewenang untuk menyetujui atau menolak klaim. Jika Deputy CEO menolak klaim, maka sistem akan secara otomatis mengirimkan notifikasi penolakan beserta laporan alasan penolakan kepada Finance sebagai pembuat klaim, memperbarui status klaim menjadi **Rejected**, dan menghentikan proses persetujuan. Namun, jika klaim disetujui oleh Deputy CEO, maka klaim akan diteruskan ke CEO untuk tahap persetujuan akhir. Setelah CEO menyetujui klaim tersebut, maka status klaim akan diperbarui menjadi **Approved** dan sistem akan melanjutkan ke tahap berikutnya yaitu proses pembayaran.

O Flowchart Proses Pembayaran (Payment)

Flowchart ini menjelaskan secara rinci alur proses pembayaran yang terjadi setelah seluruh tahap persetujuan klaim selesai dilaksanakan. Proses ini melibatkan pihak ketiga, yaitu Midtrans, sebagai *payment gateway* yang digunakan untuk memfasilitasi transaksi keuangan. Divisi *Finance* menjadi pihak utama yang menangani proses pembayaran ini, mulai dari pemilihan klaim yang telah disetujui hingga konfirmasi transaksi.



Gambar 3.16. *Flowchart* Proses Pembayaran

Pada Gambar 3.16, setelah klaim pembayaran disetujui oleh seluruh *Approver*, proses dilanjutkan ke tahap pembayaran yang dilakukan oleh divisi *Finance*. *Finance* akan memilih daftar klaim yang telah di *approve* untuk diproses pembayarannya. Selanjutnya, sistem akan mengirim permintaan pembayaran ke Midtrans melalui API. Midtrans akan memproses transaksi tersebut dan mengirimkan *callback* ke sistem untuk menginformasikan status pembayaran, apakah berhasil atau gagal. Jika pembayaran berhasil, sistem akan melakukan pembaruan status klaim, mengirimkan notifikasi email kepada pembuat klaim (*Maker*), serta membuat laporan pembayaran. Data hasil transaksi ini kemudian disimpan ke dalam basis data sebagai pencatatan resmi. Sebaliknya, jika transaksi gagal, sistem akan memperbarui status klaim sebagai gagal dan mengirimkan notifikasi penolakan serta laporan kepada pihak terkait. Informasi kegagalan ini

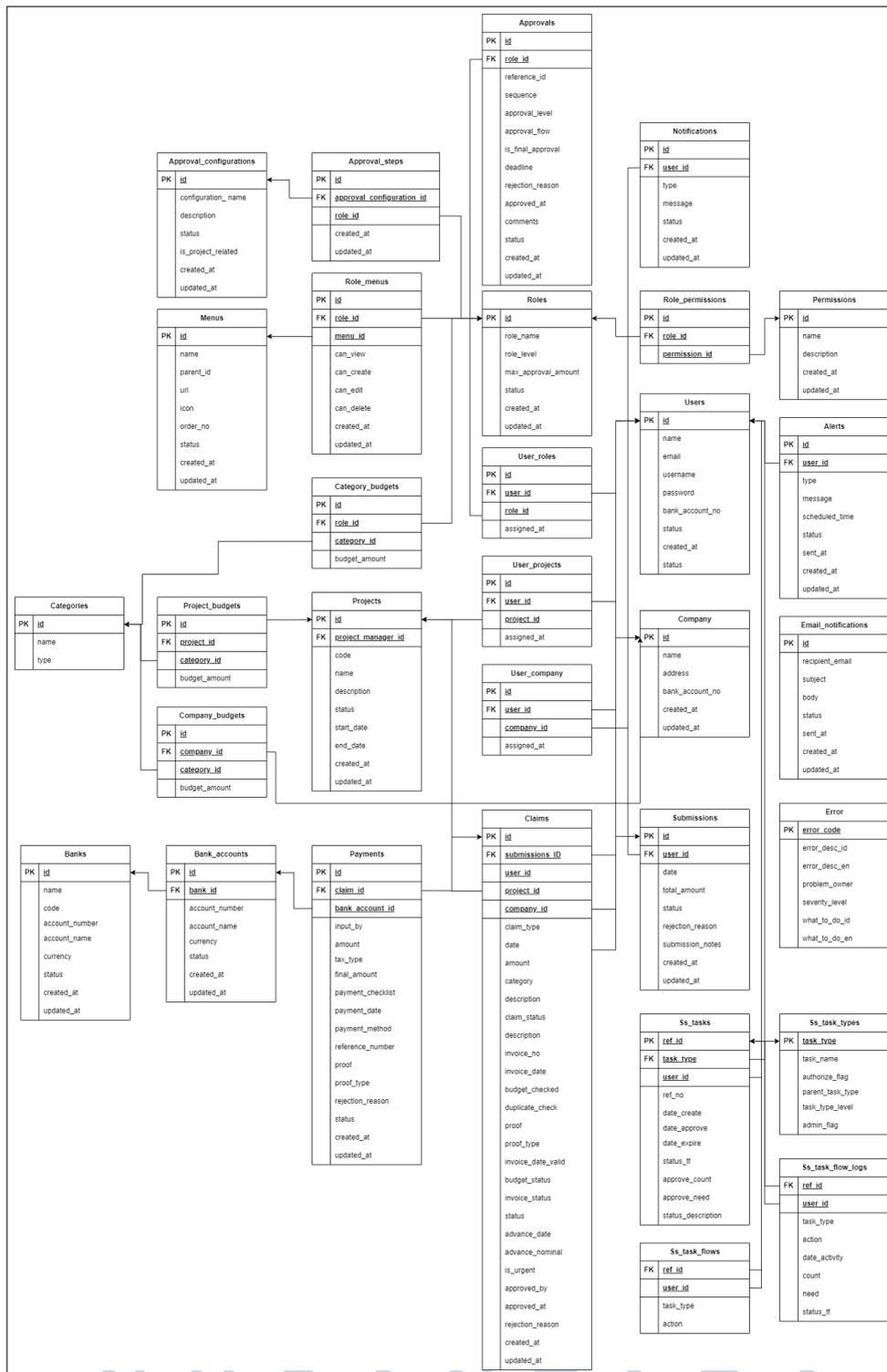
juga disimpan dalam basis data.

3.3.2 Perancangan Basis Data (Database Design)

Setelah perancangan *flowchart* selesai, tahap selanjutnya adalah merancang struktur basis data yang akan digunakan dalam sistem. Perancangan ini bertujuan untuk memastikan bahwa data yang dibutuhkan oleh sistem dapat tersimpan dengan baik, serta dapat diakses dan dikelola secara efisien.

Desain *database* dilakukan dengan menggunakan ERD (*Entity Relationship Diagram*) sebagai alat bantu visualisasi struktur data. Melalui ERD, hubungan antar entitas, atribut yang dimiliki, serta relasi antar tabel dapat digambarkan secara jelas dan sistematis. Hasil dari tahap ini akan menjadi acuan dalam pembuatan skema database yang akan diimplementasikan dalam sistem.





Gambar 3.17. Database Design

Pada Gambar 3.17, ditampilkan struktur basis data yang mencakup sejumlah entitas utama yang merepresentasikan komponen penting dalam sistem, antara

lain *users*, *role*, *menu*, *claims*, *submissions*, dan *approvals*. Entitas *users* menyimpan informasi pengguna sistem seperti nama, email, dan kredensial lainnya. Sementara itu, entitas *role* digunakan untuk mengelompokkan pengguna berdasarkan hak akses dan tanggung jawab masing-masing, seperti *Maker*, *Checker*, dan *Approver*.

Tabel *menu* menyimpan data tentang fitur-fitur atau halaman dalam aplikasi yang dapat diakses oleh pengguna, dan biasanya terhubung dengan entitas *role* melalui aturan hak akses. Untuk mengelola relasi banyak ke banyak antara pengguna dan peran, digunakan tabel penghubung bernama *user_role*, setiap entri dalam tabel ini mengaitkan seorang pengguna ke satu atau lebih peran tertentu. Ini memungkinkan fleksibilitas dalam manajemen hak akses pengguna.

Entitas *claims* merepresentasikan data klaim yang diajukan oleh pengguna, sedangkan entitas *submissions* berfungsi sebagai wadah pengelompokan klaim yang dikirim sekaligus dalam satu proses pengajuan. Setiap *claim* memiliki relasi ke *submission*, dan selanjutnya, proses persetujuan dari klaim-klaim tersebut direkam dalam entitas *approvals*. Tabel *approvals* menyimpan informasi tentang siapa yang menyetujui atau menolak klaim, termasuk urutan dan statusnya.

Untuk mendukung proses pembayaran, terdapat tabel *payment*, yang menyimpan informasi terkait permintaan pembayaran setelah klaim disetujui. Informasi ini mencakup akun sumber dan tujuan, nilai akhir pembayaran (*final amount*), status pembayaran, serta catatan jika terjadi kegagalan transaksi. Tabel ini juga terhubung ke entitas *claims* dan digunakan oleh pengguna dengan peran *Finance* dalam proses.

3.3.3 Struktur Database

Setelah dilakukan pembuatan *Entity Relationship Diagram* (ERD), struktur tabel disusun secara rinci untuk setiap entitas yang terlibat dalam sistem. Penyusunan ini mencakup penjabaran nama kolom, tipe data, *primary key*, *foreign key*, serta atribut lain yang mendukung kebutuhan sistem.

Tahapan ini bertujuan untuk memastikan bahwa seluruh data yang dibutuhkan sistem dapat direpresentasikan dengan tepat dalam bentuk tabel, serta dapat saling terhubung dengan baik antar entitas melalui relasi yang telah ditentukan sebelumnya pada ERD. Selain itu, penentuan tipe data yang sesuai juga menjadi bagian penting dalam proses ini untuk menjaga efisiensi penyimpanan, validasi input, serta performa query pada saat sistem dijalankan.

A Struktur Tabel Users

Tabel *Users* merupakan salah satu tabel utama dalam sistem yang berfungsi untuk menyimpan data seluruh pengguna aplikasi. Tabel ini digunakan untuk keperluan autentikasi, otorisasi, serta manajemen akun pengguna berdasarkan peran masing-masing. Struktur lengkap dari tabel *Users* dapat dilihat pada Gambar 3.18.

Field Name	Field Type	Max Length	Data Type	M / O	Keterangan
Id	Primary Key	11	Integer	M	Id unik untuk tabel users
Role_id	Foreign Key	11	Integer	M	FK untuk id di tabel role
Name	-	100	Varchar	M	Nama lengkap pengguna
Email	-	100	Varchar	M	Email pengguna
Username	-	100	Varchar	M	Username pengguna
Password	-	255	Varchar	M	Password
Bank_account_no	-	50	Varchar	M	Nomor rekening pengguna
Status	-	10	Enum	M	Status pengguna ('ACTIVE', 'INACTIVE')
Created_at	-	-	Timestamp	M	Waktu pembuatan
Updated_at	-	-	Timestamp	M	Waktu terakhir diperbarui

Gambar 3.18. Struktur Tabel *Users*

B Struktur Tabel User_Company

Tabel *User_Company* merupakan tabel penghubung (*junction table*) yang berfungsi untuk mengelola hubungan *one-to-many* antara pengguna dan perusahaan dalam sistem. Struktur lengkap dari tabel *User_Company* dapat dilihat pada Gambar 3.19. Tabel ini memungkinkan satu pengguna memiliki satu perusahaan, dan sebaliknya, satu perusahaan dapat memiliki beberapa pengguna.

Field Name	Field Type	Max Length	Data Type	M / O	Keterangan
Id	Primary Key	11	Integer	M	Id unik untuk tabel user projects
User_id	Foreign Key	11	Integer	M	FK untuk id di tabel users
Company_id	Foreign Key	11	Integer	M	FK untuk id di tabel company
Created_at	-	-	Timestamp	M	Waktu pembuatan
Updated_at	-	-	Timestamp	M	Waktu terakhir diperbarui

Gambar 3.19. Struktur Tabel *User_Company*

C Struktur Tabel User Projects

Tabel *User_Projects* merupakan tabel penghubung (*junction table*) yang berfungsi untuk mengelola hubungan *many-to-many* antara pengguna dan proyek dalam sistem. Berdasarkan struktur pada Gambar 3.20, tabel ini memungkinkan satu pengguna dapat ditugaskan ke beberapa proyek, dan sebaliknya, satu proyek dapat memiliki beberapa pengguna yang terlibat di dalamnya. Tabel ini berperan penting dalam sistem manajemen klaim karena memungkinkan pengguna untuk mengajukan klaim yang terkait dengan proyek tertentu sesuai dengan penugasan mereka.

Field Name	Field Type	Max Length	Data Type	M / O	Keterangan
Id	Primary Key	11	Integer	M	Id unik untuk tabel user projects
User_id	Foreign Key	11	Integer	M	FK untuk id di tabel users
Project_id	Foreign Key	11	Integer	M	FK untuk id di tabel projects
Assigned_at	-	-	Timestamp	M	Waktu user di assign ke project
Created_at	-	-	Timestamp	M	Waktu pembuatan
Updated_at	-	-	Timestamp	M	Waktu terakhir diperbarui

Gambar 3.20. Struktur Tabel *User_Projects*

D Struktur Tabel Roles

Tabel *Roles* berfungsi untuk mengelola peran dan tingkat akses pengguna dalam sistem. Tabel ini menyimpan informasi tentang berbagai peran yang dapat dimiliki pengguna, beserta level hierarki dan batas maksimal approval yang dapat dilakukan. Struktur lengkap dari tabel *Roles* dapat dilihat pada Gambar 3.21.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Field Name	Field Type	Max Length	Data Type	M / O	Keterangan
Id	Primary Key	11	Integer	M	Id unik untuk tabel roles
Name	-	100	Varchar	M	Nama role
Level	-	11	Integer	M	Tingkatan peran pada sistem
Max_approval_amount	-	20	Decimal	M	Jumlah maksimal approval yang bisa dilakukan dalam sehari
Status	-	20	Enum	M	Status proyek ('ACTIVE', 'INACTIVE')
Created_at	-	-	Timestamp	M	Waktu pembuatan
Updated_at	-	-	Timestamp	M	Waktu terakhir diperbarui

Gambar 3.21. Struktur Tabel *Roles*

E Struktur Tabel Menus

Tabel *Menus* berfungsi untuk mengelola struktur navigasi dan menu dalam sistem aplikasi. Tabel ini menyimpan informasi tentang hierarki menu, ikon, serta urutan tampilan menu. Dengan adanya kolom *parent_id*, tabel ini mendukung struktur menu bertingkat yang memungkinkan pengorganisasian navigasi yang lebih terstruktur. Struktur lengkap dari tabel *Menus* dapat dilihat pada Gambar 3.22.

Field Name	Field Type	Max Length	Data Type	M / O	Keterangan
Id	Primary Key	11	Integer	M	Id unik untuk tabel menu
Name	-	100	Varchar	M	Nama menu
Parent_id	-	11	Integer	O	Menu utama
Url	-	255	Varchar	M	Alamat yang dituju
Icon	-	-	-	O	Icon
Order_no	-	11	Integer	M	Urutan posisi menu
Status	-	10	Enum	M	Status menu ('ACTIVE', 'INACTIVE')
Created_at	-	-	Timestamp	M	Waktu pembuatan
Updated_at	-	-	Timestamp	M	Waktu terakhir diperbarui

Gambar 3.22. Struktur Tabel *Menus*

F Struktur Tabel *Role_Menus*

Tabel *Role_Menus* merupakan tabel penghubung (*junction table*) yang berfungsi untuk mengelola hubungan *many-to-many* antara peran pengguna dan menu dalam sistem. Tabel ini mengatur hak akses setiap peran terhadap menu tertentu, termasuk izin untuk melihat, membuat, mengedit, dan menghapus data

pada menu tersebut. Struktur lengkap dari tabel *Role_Menu* dapat dilihat pada Gambar 3.23.

Field Name	Field Type	Max Length	Data Type	M / O	Keterangan
Id	Primary Key	11	Integer	M	Id unik untuk tabel role menu
Role_id	Foreign Key	11	Integer	M	FK untuk id di tabel roles
Menu_id	Foreign Key	11	Integer	M	FK untuk id di tabel menus
Can_view	-	1	Tinyint	M	1 = role terkait bisa mengakses menu 0 = role terkait tidak bisa mengakses menu
Can_create	-	1	Tinyint	M	1 = role terkait bisa melakukan create di menu 0 = role terkait tidak bisa melakukan create di menu
Can_edit	-	1	Tinyint	M	1 = role terkait bisa melakukan edit di menu 0 = role terkait tidak bisa melakukan edit di menu
Can_delete	-	1	Tinyint	M	1 = role terkait bisa melakukan delete di menu 0 = role terkait tidak bisa melakukan delete di menu
Created_at	-	-	Timestamp	M	Waktu pembuatan
Updated_at	-	-	Timestamp	M	Waktu terakhir diperbarui

Gambar 3.23. Struktur Tabel *Role_Menu*

G Struktur Tabel *Company*

Tabel *Company* berfungsi untuk menyimpan informasi lengkap tentang perusahaan-perusahaan yang terdaftar dalam sistem. Tabel ini mencakup data dasar perusahaan seperti nama, alamat, dan nomor rekening bank yang digunakan untuk keperluan transaksi dan pembayaran klaim. Struktur lengkap dari tabel *Company* dapat dilihat pada Gambar 3.24.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Field Name	Field Type	Max Length	Data Type	M / O	Keterangan
Id	Primary Key	11	Integer	M	Id unik untuk tabel company
Name	-	255	Varchar	M	Nama perusahaan
Address	-	-	Text	M	Alamat Perusahaan
Bank_account_no	-	50	Varchar	M	Nomor rekening perusahaan
Created_at	-	-	Timestamp	M	Waktu pembuatan
Updated_at	-	-	Timestamp	M	Waktu terakhir diperbarui

Gambar 3.24. Struktur Tabel *Company*

H Struktur Tabel *Projects*

Tabel *Projects* berfungsi untuk menyimpan informasi lengkap tentang proyek-proyek yang berjalan dalam sistem. Tabel ini mencakup data proyek seperti kode identifikasi, nama, deskripsi, periode waktu pelaksanaan, serta status proyek. Setiap proyek memiliki Project Manager yang bertanggung jawab, dan informasi ini penting untuk mengatur klaim yang terkait dengan proyek tertentu. Struktur lengkap dari tabel *Projects* dapat dilihat pada Gambar 3.25.

Field Name	Field Type	Max Length	Data Type	M / O	Keterangan
Id	Primary Key	11	Integer	M	Id unik untuk tabel projects
Project_manager_id	Foreign Key	11	Integer	M	FK dengan id tabel user
Code	-	50	Varchar	M	Kode proyek
Name	-	255	Varchar	M	Nama proyek
Description	-	-	Text	O	Deskripsi proyek
Start_date	-	-	Date	M	Tanggal mulai proyek
End_date	-	-	Date	M	Tanggal berakhirnya proyek
Status	-	20	Enum	M	Status proyek ('IN_PROGRESS', 'ON_HOLD', 'COMPLETED', 'CANCELED')
Created_at	-	-	Timestamp	M	Waktu pembuatan
Updated_at	-	-	Timestamp	M	Waktu terakhir diperbarui

Gambar 3.25. Struktur Tabel *Projects*

I Struktur Tabel *Categories*

Tabel *Categories* berfungsi untuk menyimpan informasi kategori-kategori klaim yang tersedia dalam sistem. Tabel ini mengklasifikasikan jenis klaim

berdasarkan tipenya, seperti klaim yang terkait dengan proyek (*PROJECT*), keperluan personal (*PERSONAL*), atau keperluan perusahaan (*COMPANY*). Kategorisasi ini membantu dalam pengelolaan dan pengaturan anggaran serta proses *approval* yang berbeda untuk setiap jenis klaim. Struktur lengkap dari tabel *Categories* dapat dilihat pada Gambar 3.26.

Field Name	Field Type	Max Length	Data Type	M / O	Keterangan
Id	Primary Key	11	Integer	M	Id unik untuk tabel kategori
Name	-	100	Varchar	M	Nama kategori
Type	-	10	Enum	M	Tipe kategori klaim ('PROJECT', 'PERSONAL', 'COMPANY')
Created_at	-	-	Timestamp	M	Waktu pembuatan
Updated_at	-	-	Timestamp	M	Waktu terakhir diperbarui

Gambar 3.26. Struktur Tabel *Categories*

J Struktur Tabel *Submissions*

Tabel *Submissions* berfungsi untuk menyimpan informasi pengajuan klaim yang dibuat oleh pengguna. Tabel ini merupakan wadah yang mengelompokkan beberapa klaim individual menjadi satu pengajuan, sehingga pengguna dapat mengajukan beberapa klaim sekaligus dalam satu *submission*. Tabel ini menyimpan *total amount* dari seluruh klaim dalam *submission* tersebut, status pengajuan, serta informasi tambahan seperti catatan dan alasan penolakan jika diperlukan. Struktur lengkap dari tabel *Submissions* dapat dilihat pada Gambar 3.27.

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

Field Name	Field Type	Max Length	Data Type	M / O	Keterangan
Id	Primary Key	11	Integer	M	Id unik untuk tabel submissions
User_id	Foreign Key	11	Integer	M	FK untuk id di tabel user
Date	-	-	Timestamp	M	Waktu pembuatan submissions
Total_amount	-	20	Decimal	M	Jumlah amount dari seluruh klaim didalam submissions
Status	-	20	Enum	M	Status submissions ('DRAFT', 'PENDING', 'APPROVED', 'REJECTED', 'PARTIALLY_APPROVED')
Rejection_reason	-	-	Text	O	Alasan penolakan (ketika status 'REJECTED')
Submission_notes	-	-	Text	O	Informasi tambahan terkait submissions
Created_at	-	-	Timestamp	M	Waktu pembuatan
Updated_at	-	-	Timestamp	M	Waktu terakhir diperbarui

Gambar 3.27. Struktur Tabel *Submissions*

K Struktur Tabel *Claims*

Tabel *Claims* merupakan tabel inti dalam sistem yang berfungsi untuk menyimpan detail setiap klaim yang diajukan oleh pengguna. Tabel ini mencakup informasi lengkap klaim seperti jumlah dana, deskripsi, bukti pembayaran, serta berbagai validasi otomatis seperti pengecekan budget, validasi tanggal *invoice*, dan deteksi duplikasi. Setiap klaim terhubung dengan *submission*, *Maker*, kategori, dan dapat dikaitkan dengan proyek atau perusahaan tertentu. Struktur lengkap dari tabel *Claims* dapat dilihat pada Gambar 3.28.

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

Field Name	Field Type	Max Length	Data Type	M / O	Keterangan
id	Primary Key	11	Integer	M	Id unik untuk tabel klaim
Submissions_id	Foreign Key	11	Integer	M	FK ke id tabel submissions
User_id	Foreign Key	11	Integer	M	FK ke id tabel users
Project_id	Foreign Key	11	Integer	M	FK ke id tabel projects
Company_id	Foreign Key	11	Integer	M	FK ke id tabel company
Category_id	Foreign Key	11	Integer	M	FK ke id tabel categories
Claim_type	-	20	Varchar	M	Tipe klaim
Date	-	-	Date	M	Tanggal pembuatan klaim
Amount	-	20	Decimal	M	Jumlah dana klaim
Description	-	-	Text	O	Deskripsi klaim
Invoice_no	-	255	Varchar	M	Nomor invoice
Invoice_date	-	-	Date	M	Tanggal penerbitan invoice
Invoice_date_valid	-	1	Tinyint	M	1 = Invoice berada dalam periode yang dapat diklaim 0 = Tanggal invoice sudah lewat
Budget_status	-	20	Enum	M	Status budget ('AVAILABLE', 'NOT_AVAILABLE', 'NO_BUDGET_DEFINED')
Budget_checked	-	1	Tinyint	M	1 = budget tersedia, 0 = budget tidak tersedia
Duplicate_check	-	1	Tinyint	M	1 = klaim duplikat 0 = klaim unique
Proof	-	-	Text	M	Link storage
Proof_type	-	10	Enum	M	Tipe bukti ('PDF', 'IMAGE')
Status	-	20	Enum	M	Status klaim ('PENDING', 'APPROVED', 'REJECTED')
Advance_date	-	-	Date	O	Tanggal uang muka diberikan
Advance_nominal	-	20	Decimal	O	Nominal uang muka yang diberikan
Is_urgent	-	1	Tinyint	O	0 = klaim tidak bersifat urgent 1 = klaim bersifat urgent
Approved_by	-	11	Integer	M	Id user yang menyetujui
Approved_at	-	-	Timestamp	M	Waktu persetujuan
Rejection_reason	-	-	Text	O	Alasan penolakan klaim
Created_at	-	-	Timestamp	M	Waktu pembuatan
Updated_at	-	-	Timestamp	M	Waktu terakhir diperbarui

Gambar 3.28. Struktur Tabel *Claims*

L Struktur Tabel *Company_Budgets*

Tabel *Company_Budgets* berfungsi untuk mengelola anggaran yang dialokasikan untuk setiap perusahaan berdasarkan kategori klaim tertentu. Tabel ini menyimpan informasi tentang total anggaran yang ditetapkan dan jumlah anggaran yang telah digunakan untuk setiap kombinasi perusahaan dan kategori. Sistem ini memungkinkan kontrol pengeluaran yang lebih efektif dengan memantau penggunaan budget secara *real-time*. Struktur lengkap dari tabel *Company_Budgets* dapat dilihat pada Gambar 3.29.

Field Name	Field Type	Max Length	Data Type	M / O	Keterangan
Id	Primary Key	11	Integer	M	Id unik untuk tabel budget company
Company_id	Foreign Key	11	Integer	M	FK ke id tabel company
Category_id	Foreign Key	11	Integer	M	FK ke id tabel categories
Budget_amount	-	30	Decimal	M	Total anggaran yang telah ditetapkan
Used_budget	-	20	Decimal	M	Total anggaran yang telah digunakan
Created_at	-	-	Timestamp	M	Waktu pembuatan
Updated_at	-	-	Timestamp	M	Waktu terakhir diperbarui

Gambar 3.29. Struktur Tabel *Company_Budgets*

M Struktur Tabel *Project_Budgets*

Tabel *Project_Budgets* berfungsi untuk mengelola anggaran yang dialokasikan untuk setiap proyek berdasarkan kategori klaim tertentu. Tabel ini menyimpan informasi tentang total anggaran yang ditetapkan dan jumlah anggaran yang telah digunakan untuk setiap kombinasi proyek dan kategori. Sistem ini memungkinkan Project Manager untuk memantau penggunaan *budget* proyek secara *real-time* dan memastikan pengeluaran tidak melebihi anggaran yang telah ditetapkan. Struktur lengkap dari tabel *Project_Budgets* dapat dilihat pada Gambar 3.30.

Field Name	Field Type	Max Length	Data Type	M / O	Keterangan
Id	Primary Key	11	Integer	M	Id unik untuk tabel budget projects
Project_id	Foreign Key	11	Integer	M	FK ke id di tabel projects
Category_id	Foreign Key	11	Integer	M	FK ke id di tabel categories
Budget_amount	-	30	Decimal	M	Total anggaran yang telah ditetapkan
Used_budget	-	20	Decimal	M	Total anggaran yang telah digunakan
Created_at	-	-	Timestamp	M	Waktu pembuatan
Updated_at	-	-	Timestamp	M	Waktu terakhir diperbarui

Gambar 3.30. Struktur Tabel *Project_Budgets*

N Struktur Tabel *Category_Budgets*

Tabel *Category_Budgets* berfungsi untuk mengelola anggaran yang dialokasikan untuk setiap peran pengguna berdasarkan kategori klaim tertentu. Tabel ini menyimpan informasi tentang total anggaran yang ditetapkan dan jumlah anggaran yang telah digunakan untuk setiap kombinasi peran dan kategori. Sistem ini memungkinkan kontrol pengeluaran berdasarkan level peran pengguna, sehingga setiap *role* memiliki batasan *budget* yang berbeda untuk kategori klaim yang sama. Struktur lengkap dari tabel *Category_Budgets* dapat dilihat pada Gambar 3.31.

Field Name	Field Type	Max Length	Data Type	M / O	Keterangan
Id	Primary Key	11	Integer	M	Id unik untuk tabel budget kategori
Role_id	Foreign Key	100	Varchar	M	FK ke id tabel role
Category_id	Foreign Key	10	Enum	M	FK ke id tabel categories
Budget_amount	-	20	Decimal	M	Total anggaran yang telah ditetapkan
Used_budget	-	20	Decimal	M	Total anggaran yang telah digunakan
Created_at	-	-	Timestamp	M	Waktu pembuatan
Updated_at	-	-	Timestamp	M	Waktu terakhir diperbarui

Gambar 3.31. Struktur Tabel *Category_Budgets*

O Struktur Tabel Payments

Tabel *Payments* berfungsi untuk menyimpan informasi detail pembayaran yang dilakukan untuk setiap klaim yang telah disetujui. Tabel ini mencakup informasi lengkap pembayaran seperti jumlah yang dibayarkan, jenis pajak yang dikenakan, jumlah akhir setelah pajak, metode pembayaran, serta bukti pembayaran. Tabel ini juga menyimpan status pembayaran dan memungkinkan *tracking* proses pembayaran dari *pending* hingga *completed*. Struktur lengkap dari tabel *Payments* dapat dilihat pada Gambar 3.32.

Field Name	Field Type	Max Length	Data Type	M / O	Keterangan
Id	Primary Key	11	Integer	M	Id unik untuk tabel user projects
Claim_id	Foreign Key	11	Integer	M	FK untuk id di tabel claims
Input_by	Foreign Key	11	Integer	M	FK untuk id di tabel users
Bank_account_id	-	11	Integer	M	Nomor rekening
Amount	-	20	Decimal	M	Jumlah yang dibayarkan
Tax_type	-	10	Enum	M	Tipe pajak ('PPN', 'PPH 23', 'PPH 21', 'PPH 4(2)', 'NONE')
Final_amount	-	20	Decimal	M	Jumlah akhir setelah ditambahkan pajak
Payment_checklist	-	1	Tinyint	M	1 = sudah diproses 0 = belum diproses
Payment_date	-	-	Date	M	Tanggal pembayaran
Payment_method	-	20	Enum	M	Metode pembayaran ('TRANSFER', 'CASH', 'CHEQUE', 'OTHER')
Reference_number	-	100	Varchar	M	Nomor referensi
Proof	-	255	Varchar	M	Link file storage bukti pembayaran
Proof_type	-	20	Enum	M	Tipe bukti ('PDF', 'IMAGE')
Rejection_reason	-	-	Text	O	Alasan penolakan
Status	-	20	Enum	M	Status pembayaran ('PENDING', 'PROCESSING', 'COMPLETED', 'FAILED')
Created_at	-	-	Timestamp	M	Waktu pembuatan
Updated_at	-	-	Timestamp	M	Waktu terakhir diperbarui

Gambar 3.32. Struktur Tabel *Payments*

P Struktur Tabel Approval_Configuration

Tabel *Approval_Configuration* berfungsi untuk menyimpan konfigurasi proses persetujuan dalam sistem klaim. Tabel ini menyimpan informasi tentang berbagai konfigurasi *approval* yang dapat digunakan dalam sistem, seperti nama

konfigurasi, deskripsi, status aktif/tidak aktif, dan apakah konfigurasi tersebut terkait dengan proyek atau tidak. Setiap konfigurasi dapat memiliki langkah-langkah *approval* yang berbeda sesuai dengan kebutuhan bisnis. Struktur lengkap dari tabel *Approval_Configuration* dapat dilihat pada Gambar 3.33.

Field Name	Field Type	Max Length	Data Type	M / O	Keterangan
Id	Primary Key	11	Integer	M	Id unik untuk konfigurasi
Configuration_name	-	100	Varchar	M	Nama konfigurasi
Description	-	255	Varchar	O	Deskripsi konfigurasi
Status	-	10	Enum	M	Status konfigurasi ('ACTIVE', 'INACTIVE')
Is_project_related	-	1	Tinyint	M	1 = melibatkan proyek, 0 = tidak melibatkan proyek
Created_at	-	-	Timestamp	M	Waktu pembuatan
Updated_at	-	-	Timestamp	M	Waktu terakhir diperbarui

Gambar 3.33. Struktur Tabel *Approval_Configuration*

Q Struktur Tabel *Approval_Steps*

Tabel *Approval_Steps* berfungsi untuk menyimpan langkah-langkah spesifik dalam proses persetujuan yang telah dikonfigurasi. Tabel ini memiliki relasi dengan tabel *Approval_Configuration* untuk menentukan konfigurasi mana yang digunakan, serta dengan tabel *Roles* untuk menentukan peran yang bertanggung jawab pada setiap langkah *approval*. Setiap langkah dalam proses persetujuan akan memiliki urutan dan peran yang jelas, memungkinkan sistem untuk menjalankan alur *approval* secara terstruktur sesuai dengan hierarki organisasi. Struktur lengkap dari tabel *Approval_Steps* dapat dilihat pada Gambar 3.34.

Field Name	Field Type	Max Length	Data Type	M / O	Keterangan
Id	Primary Key	11	Integer	M	Id unik untuk langkah konfigurasi
Approval_configuration_id	Foreign Key	11	Integer	M	FK ke id tabel <i>approval_configuration</i>
Role_id	Foreign Key	11	Integer	M	FK ke id tabel <i>role</i>
Created_at	-	-	Timestamp	M	Waktu pembuatan
Updated_at	-	-	Timestamp	M	Waktu terakhir diperbarui

Gambar 3.34. Struktur Tabel *Approval_Steps*

R Struktur Tabel Ss.Tasks

Tabel *Ss.Tasks* berfungsi untuk melacak dan mengelola tugas-tugas *approval* yang perlu diselesaikan dalam sistem. Tabel ini menyimpan informasi mengenai progress *approval* untuk setiap tugas, termasuk jumlah *approval* yang sudah dilakukan dan yang masih dibutuhkan. Dengan adanya *field approve_count* dan *approve_need*, sistem dapat menentukan apakah suatu tugas sudah mendapat persetujuan yang cukup atau masih memerlukan *approval* tambahan. Tabel ini juga mencatat waktu pembuatan tugas dan waktu *approval* untuk keperluan audit dan *tracking*. Struktur lengkap dari tabel *Ss.Tasks* dapat dilihat pada Gambar 3.35.

Field Name	Field Type	Max Length	Data Type	M / O	Keterangan
Ref_id	Primary Key	11	Integer	M	Id unik untuk tabel <i>ss_tasks</i>
User_id	Foreign Key	11	Integer	M	FK untuk id di tabel user
Date_create	-	-	Datetime	M	Waktu pembuatan klaim
Date_approve	-	-	Datetim	M	Waktu klaim di approve
Approve_count	-	5	Decimal	M	Jumlah approval yang sudah dilakukan
Approve_need	-	5	Decimal	M	Jumlah approval yang dibutuhkan
Created_at	-	-	Timestamp	M	Waktu pembuatan
Updated_at	-	-	Timestamp	M	Waktu terakhir diperbarui

Gambar 3.35. Struktur Tabel *ss.tasks*

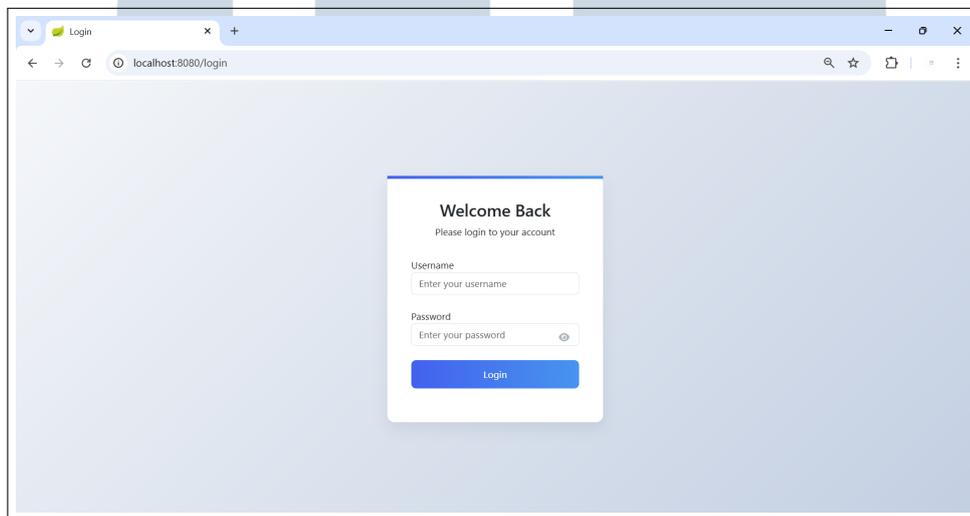
3.3.4 Perancangan Antarmuka Pengguna (Frontend Design)

Setelah proses perancangan basis data selesai dilakukan, tahap selanjutnya adalah merancang antarmuka pengguna (*user interface*). Desain antarmuka ini dibangun menggunakan teknologi Thymeleaf, yang terintegrasi langsung dengan kerangka kerja Spring Boot untuk menghasilkan tampilan web yang dinamis dan responsif.

A Tampilan Login

Bagian ini memberikan penjelasan mengenai tampilan halaman *login* yang digunakan oleh seluruh pengguna untuk mengakses sistem. Halaman ini merupakan gerbang utama dalam proses autentikasi, di mana pengguna diminta untuk memasukkan kredensial yang valid agar dapat melanjutkan ke dalam sistem.

Desain antarmuka *login* dibuat sesederhana mungkin untuk memastikan kemudahan penggunaan. Dari sisi keamanan, sistem menggunakan token *JSON Web Token* sebagai metode autentikasi. Setelah pengguna berhasil *login*, token ini akan disimpan di sisi klien dan digunakan untuk mengakses berbagai fitur dalam sistem secara aman. Selain itu, sistem juga menerapkan mekanisme *idle timeout* selama lima menit, di mana pengguna akan otomatis logout apabila tidak melakukan aktivitas apa pun dalam kurun waktu tersebut. Mekanisme ini dirancang untuk mencegah penyalahgunaan akses oleh pihak yang tidak berwenang, terutama saat pengguna meninggalkan perangkat dalam keadaan *login*.

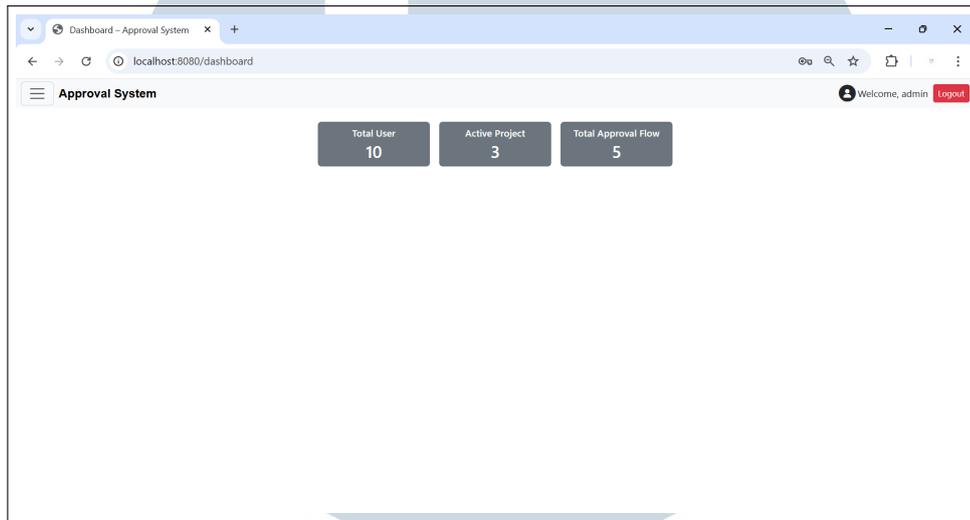


Gambar 3.36. Tampilan *Login*

Pada Gambar 3.36, halaman *login* menampilkan formulir dengan dua input utama, yaitu *Username* dan *Password*, disertai tombol *Login* di bagian bawah. Untuk meningkatkan kenyamanan pengguna, terdapat pula ikon berbentuk mata di sebelah kanan *input password* yang memungkinkan pengguna melihat isi *password* yang baru saja dimasukkan. Fitur ini berguna terutama ketika pengguna ingin memastikan bahwa *password* yang diketik sudah benar, mengingat karakter *password* secara *default* ditampilkan dalam bentuk simbol bintang (***) . Jika berhasil, pengguna akan diarahkan ke *dashboard* sesuai dengan peran mereka. Jika gagal, sistem akan menampilkan notifikasi kesalahan "*Username atau Password salah*".

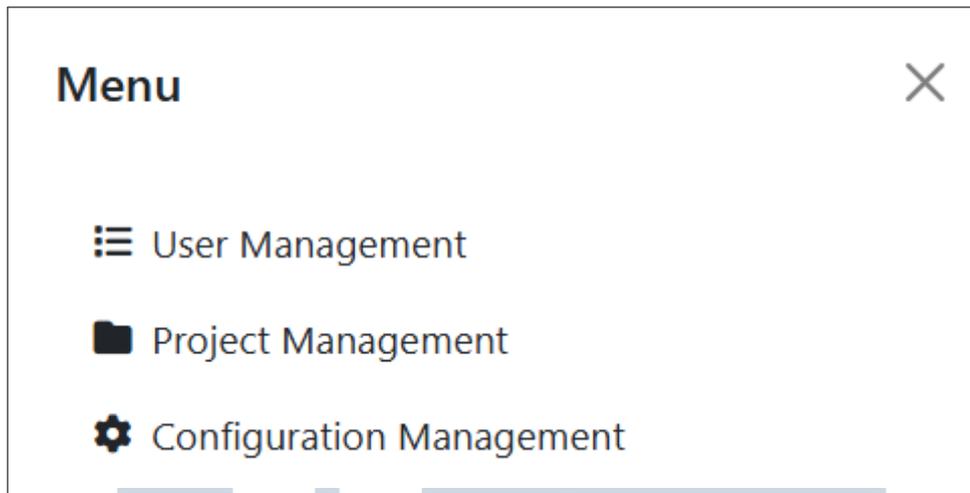
B Tampilan Dashboard Admin

Bagian ini memberikan penjelasan mengenai tampilan halaman *dashboard* yang digunakan oleh pengguna dengan peran *Admin*. Melalui tampilan ini, *Admin* dapat melihat ringkasan data, termasuk jumlah pengguna, jumlah proyek aktif, dan jumlah konfigurasi alur persetujuan yang tersedia.



Gambar 3.37. Tampilan *Dashboard* untuk *Admin*

Gambar 3.37 menampilkan tampilan *dashboard* utama yang dilihat oleh pengguna dengan peran *Admin*. Pada bagian atas halaman terdapat *navbar* (*navigation bar*) yang berfungsi sebagai elemen navigasi utama dalam sistem. Di sebelah kiri atas *navbar* terdapat ikon *hamburger* (tiga garis horizontal) yang dapat ditekan untuk membuka atau menutup menu *sidebar*. Jika pengguna menekan tulisan “Approval System” yang juga berada di *navbar*, sistem akan mengarahkan kembali ke halaman *dashboard* utama. Sementara itu, di bagian kanan atas *navbar* terdapat tombol “Logout” yang digunakan untuk keluar dari sistem. Pada halaman *dashboard*, terdapat tiga kotak informasi yang menampilkan data penting bagi *Admin*, seperti *total user*, *active project*, dan *total approval flow*.



Gambar 3.38. Tampilan *Sidebar* untuk *Admin*

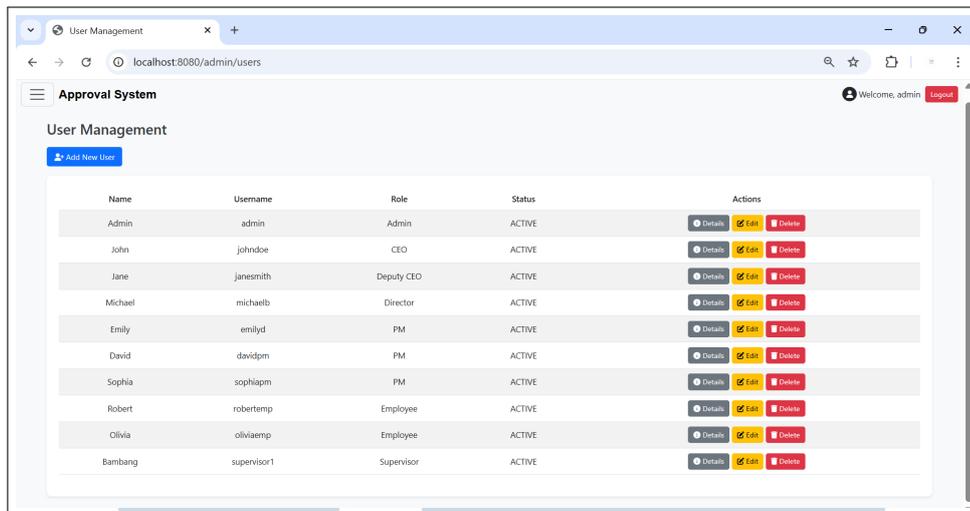
Gambar 3.38 menunjukkan tampilan *sidebar* atau menu navigasi untuk *Admin*. Pada *sidebar* ini, ditampilkan menu-menu utama yang relevan dengan peran *Admin*, seperti "User Management", "Project Management", dan "Configuration Management".

C User Management

Bagian ini memberikan penjelasan mengenai tampilan dari menu "User Management" yang digunakan oleh pengguna dengan peran *Admin*. Menu ini merupakan salah satu fitur penting dalam sistem yang memungkinkan *Admin* untuk mengelola data pengguna secara menyeluruh.

C.1 Tampilan List User

Bagian ini memberikan penjelasan mengenai tampilan daftar pengguna (*user list*) yang ditampilkan pada menu "User Management". Tampilan ini menyajikan informasi penting mengenai seluruh pengguna yang terdaftar dalam sistem, seperti nama, *username*, email, peran, dan status akun. *Admin* dapat menggunakan daftar ini untuk memantau aktivitas pengguna serta melakukan tindakan administratif, seperti mengedit data pengguna, mengatur ulang *password*, atau menonaktifkan akun.

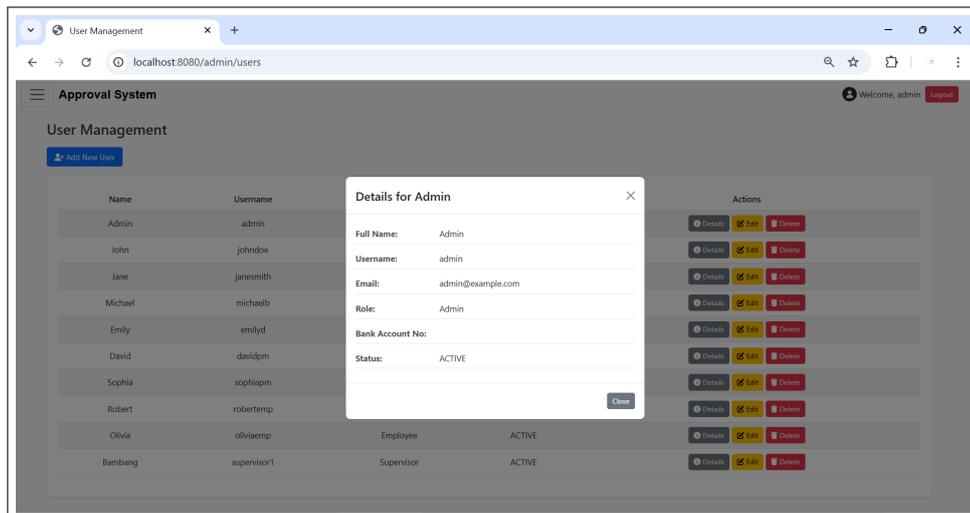


Gambar 3.39. Tampilan *List* Pengguna di "User Management"

Gambar 3.39 menunjukkan tampilan daftar pengguna yang terdapat pada menu "User Management". Halaman ini menampilkan semua pengguna yang telah terdaftar di sistem, lengkap dengan informasi seperti *username*, *role*, status, dan opsi aksi seperti "View", "Edit", dan "Delete". Selain itu, di bagian kiri atas halaman juga tersedia tombol "Add New User" yang digunakan untuk menambahkan pengguna baru ke dalam sistem.

C.2 Menampilkan Detail User

Gambar 3.40 di bawah menunjukkan tampilan detail informasi pengguna yang muncul ketika *Admin* menekan tombol "View" pada daftar pengguna di halaman menu "User Management". Pada tampilan ini, sistem menyajikan informasi lengkap terkait data pengguna yang dipilih, seperti nama lengkap, *username*, peran, dan status aktif akun. Fitur ini memudahkan *Admin* untuk melakukan pemeriksaan data pengguna secara lebih mendalam sebelum mengambil tindakan lebih lanjut, seperti melakukan perubahan data atau menonaktifkan akun.

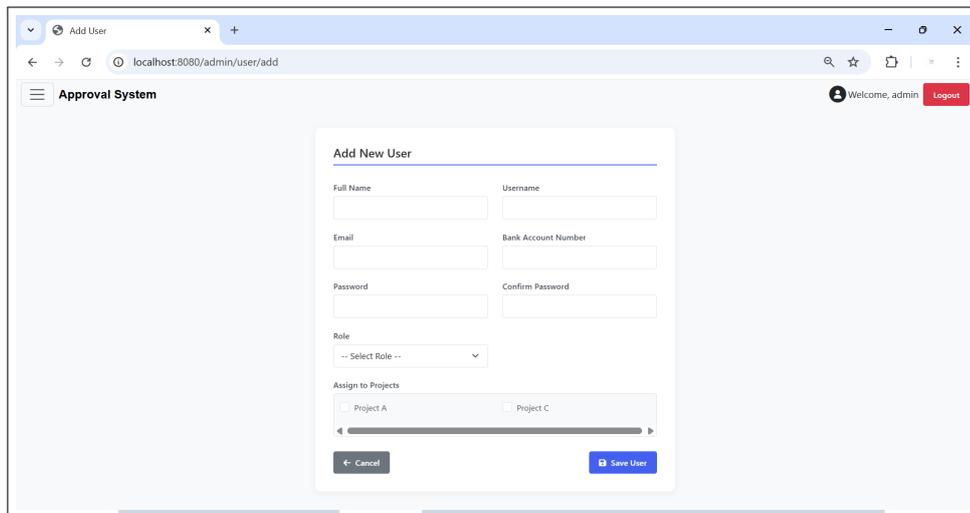


Gambar 3.40. Tampilan Detail Pengguna di "User Management"

Tampilan detail ini ditampilkan dalam bentuk *pop-up* atau *modal window* untuk menjaga kenyamanan dan konsistensi tampilan antarmuka. Penggunaan *pop-up* dipilih agar tampilan utama tidak perlu berpindah halaman, serta untuk mengantisipasi kemungkinan penambahan *field* informasi pengguna di masa mendatang. Dengan desain ini, sistem dapat dengan mudah menampilkan informasi tambahan tanpa mengubah struktur halaman utama. Pengguna dapat menekan tombol "Close" pada *pop-up window* untuk menutup tampilan detail dan kembali ke halaman sebelumnya tanpa melakukan *refresh* atau berpindah halaman.

C.3 Membuat User Baru

Gambar 3.41 di bawah menunjukkan tampilan saat *Admin* ingin menambahkan pengguna baru ke dalam sistem. Tampilan ini akan muncul ketika *Admin* menekan tombol "Add New User" yang terdapat di halaman menu "User Management". Pada halaman ini, *Admin* diminta untuk mengisi sejumlah informasi penting seperti nama lengkap, *username*, alamat email, *password* awal, serta memilih peran yang sesuai bagi pengguna tersebut. Selain itu, jika peran yang dipilih adalah Project Manager, maka *Admin* juga dapat langsung menetapkan proyek tertentu kepada *user* tersebut. Formulir ini dirancang agar mudah digunakan dan memastikan bahwa seluruh data yang dibutuhkan untuk registrasi pengguna dapat diinput dengan lengkap dan benar.



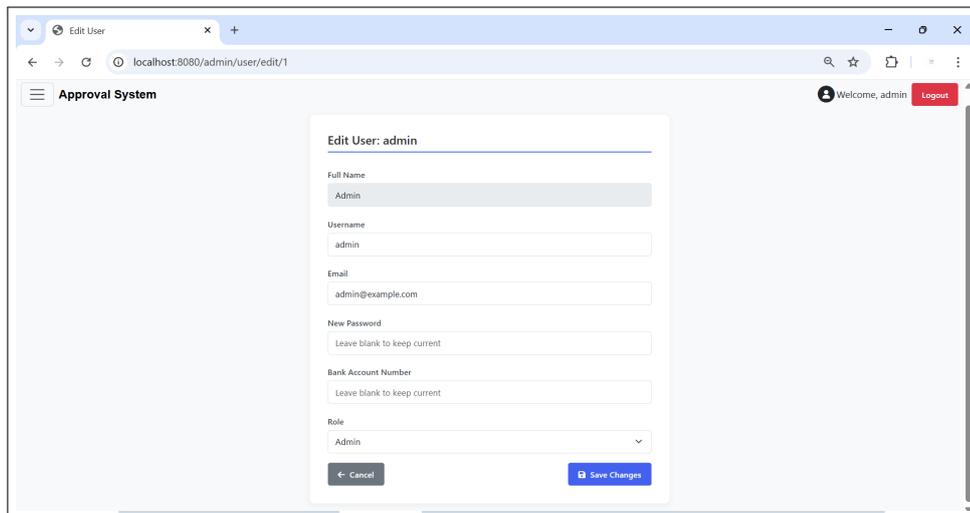
Gambar 3.41. Tampilan Membuat *User* Baru di "User Management"

Setelah seluruh data diisi dengan benar, *Admin* dapat menekan tombol "Save User" yang berada di bagian kanan bawah untuk menyimpan data pengguna ke dalam basis data. Sebaliknya, apabila *Admin* memutuskan untuk membatalkan proses penambahan, tombol "Cancel" dapat digunakan untuk kembali ke halaman sebelumnya tanpa menyimpan perubahan.

C.4 Mengedit Data User

Gambar 3.42 di bawah menunjukkan tampilan saat *Admin* ingin menambahkan pengguna baru ke dalam sistem. Tampilan ini akan muncul ketika *Admin* menekan tombol "Add New User" yang terdapat di halaman menu "User Management".

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

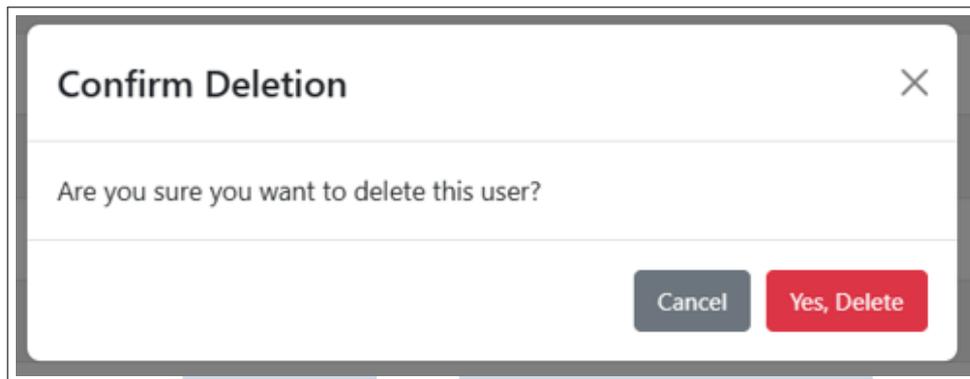


Gambar 3.42. Tampilan Mengedit Data User di "User Management"

Pada halaman ini, data pengguna yang telah diinput sebelumnya akan ditampilkan dalam form untuk mempermudah proses pengeditan. Perlu dicatat bahwa *field Full Name* tidak dapat diedit. Jika *Admin* ingin mengubah *Full Name* pengguna maka harus membuat pengguna baru. Selain itu, *field Password* dan *Bank Account* bersifat opsional. Jika tidak ada perubahan pada informasi tersebut, maka *field* dapat dibiarkan kosong. Setelah selesai melakukan perubahan, *Admin* dapat menyimpan data dengan menekan tombol "Save Changes". Jika *Admin* memutuskan untuk membatalkan proses pengeditan dan ingin kembali ke halaman sebelumnya, maka *Admin* dapat menekan tombol "Cancel".

C.5 Menghapus User

Gambar 3.43 di bawah menunjukkan tampilan konfirmasi yang muncul ketika *Admin* ingin menghapus salah satu pengguna dari sistem. Fitur ini tersedia di halaman menu "User Management" dan dirancang untuk memberikan keamanan tambahan sebelum tindakan penghapusan dilakukan.



Gambar 3.43. Tampilan Menghapus Data *User* di "User Management"

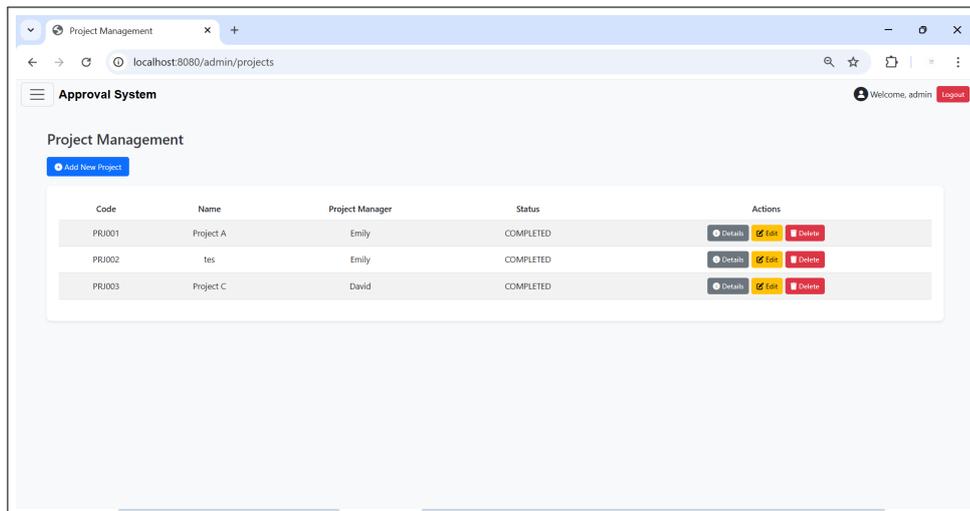
Untuk menghapus pengguna, *Admin* dapat menekan tombol "Delete" yang terdapat pada daftar pengguna. Setelah tombol ditekan, sistem akan menampilkan jendela konfirmasi seperti yang ditunjukkan pada Gambar 3.43. Halaman konfirmasi ini berfungsi untuk memastikan bahwa *Admin* benar-benar ingin menghapus pengguna tersebut, sehingga dapat mencegah penghapusan yang tidak disengaja. Jika *Admin* memilih untuk melanjutkan, maka data pengguna akan dihapus secara permanen dari basis data.

D Project Management

Bagian ini memberikan penjelasan mengenai tampilan dari menu "Project Management" yang digunakan oleh *Admin*. Menu ini berfungsi untuk mengelola data proyek yang terlibat dalam proses pengajuan dan persetujuan klaim. *Admin* dapat menambahkan, memperbarui, atau menghapus data proyek, serta mengatur keterkaitan proyek dengan pengguna dengan peran sebagai Project Manager.

D.1 Tampilan List Project

Bagian ini memberikan penjelasan mengenai tampilan daftar proyek yang tersedia pada menu "Project Management". Melalui tampilan ini, *Admin* dapat memantau seluruh proyek yang telah terdaftar dan melakukan tindakan administratif yang diperlukan terhadap setiap entri proyek.



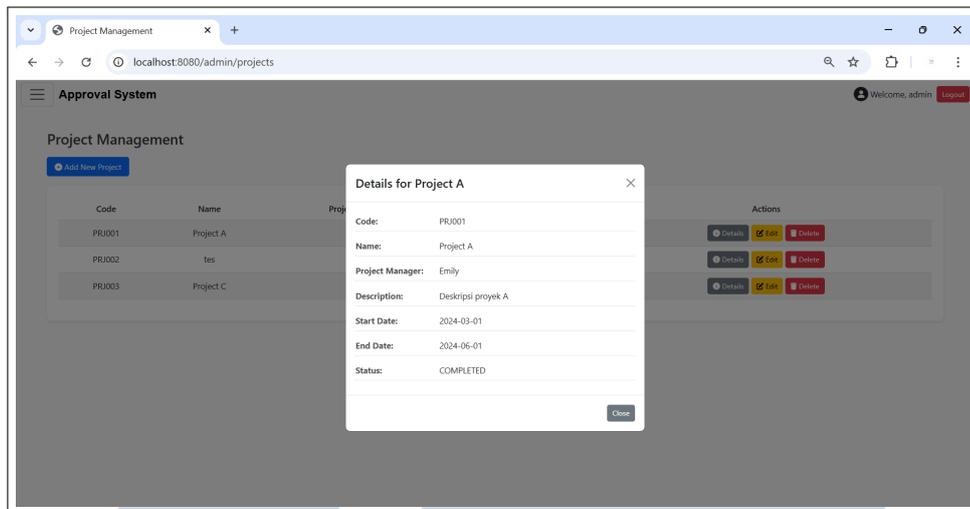
Gambar 3.44. Tampilan *List* Proyek di "Project Management"

Gambar 3.44 menunjukkan tampilan daftar proyek dalam sistem, yang memiliki struktur antarmuka serupa dengan halaman menu "User Management". Pada halaman ini, setiap entri proyek ditampilkan dalam bentuk tabel beserta informasi penting seperti nama proyek dan deskripsi singkat. *Admin* juga dapat menggunakan tombol aksi seperti "View", "Edit", dan "Delete" yang tersedia untuk setiap proyek guna melakukan manajemen data dengan cepat dan efisien.

D.2 Menampilkan Detail Project

Gambar 3.45 di bawah menunjukkan tampilan halaman yang muncul ketika *Admin* ingin melihat detail dari salah satu proyek yang terdaftar dalam sistem. Tampilan ini dapat diakses dengan menekan tombol "View" pada entri proyek tertentu di halaman daftar proyek (*Project List*).

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

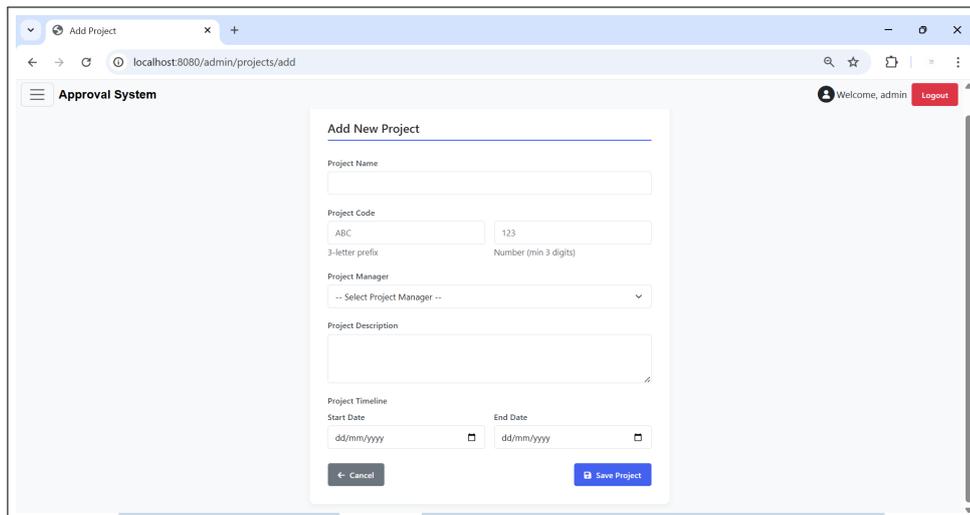


Gambar 3.45. Tampilan Detail Proyek di "Project Management"

Tampilan detail ini menyajikan informasi lengkap mengenai proyek yang dipilih, seperti nama proyek, manajer proyek, tanggal mulai, serta deskripsi atau informasi tambahan terkait proyek tersebut. Tujuan dari halaman ini adalah untuk memberikan akses cepat bagi *Admin* dalam melakukan pengecekan data proyek secara rinci, tanpa perlu masuk ke mode pengeditan. Struktur dan antarmuka halaman ini juga dirancang secara konsisten dengan halaman detail pengguna, sehingga memudahkan *Admin* dalam navigasi dan penggunaan sistem secara keseluruhan.

D.3 Membuat Project Baru

Gambar 3.46 di bawah menunjukkan tampilan antarmuka saat *Admin* ingin menambahkan proyek baru melalui halaman menu "Project Management". Fitur ini digunakan untuk mencatat proyek yang akan terlibat dalam proses klaim, sehingga penting bagi *Admin* untuk mengisi data secara lengkap dan akurat.

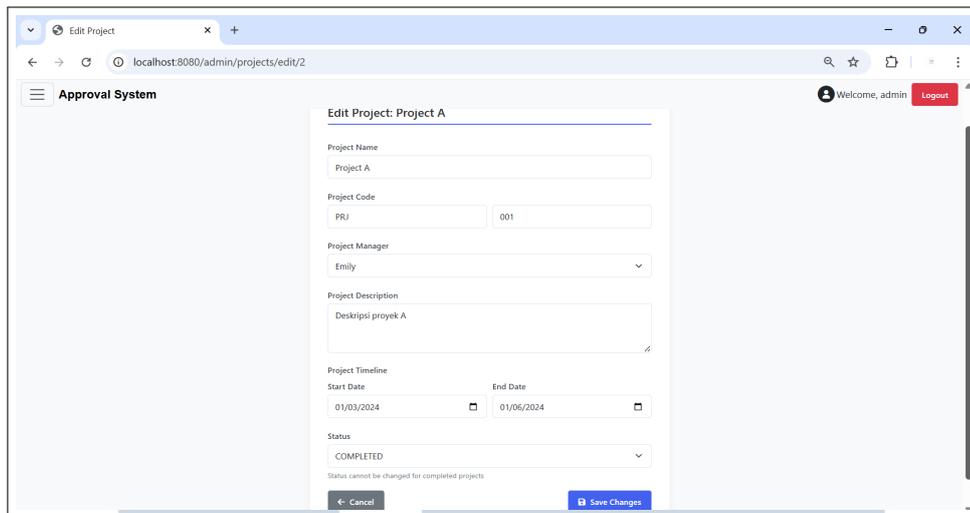


Gambar 3.46. Tampilan Membuat Proyek Baru di "Project Management"

Pada tampilan ini, *Admin* dapat mengisi sejumlah informasi penting seperti nama proyek, manajer proyek yang ditugaskan, tanggal mulai, estimasi tanggal selesai, dan deskripsi proyek. Formulir ini dirancang untuk memastikan bahwa seluruh informasi yang dibutuhkan tersedia sejak awal proses pendaftaran proyek. Setelah data dilengkapi, *Admin* dapat menekan tombol "Save Project" untuk menyimpan proyek ke dalam sistem. Sebaliknya, jika *Admin* ingin membatalkan proses tersebut dan kembali ke halaman sebelumnya, maka tombol "Cancel" dapat digunakan untuk keluar tanpa menyimpan data apa pun.

D.4 Mengedit Data Project

Gambar 3.47 di bawah memperlihatkan tampilan saat *Admin* ingin melakukan perubahan data pada proyek yang telah terdaftar di halaman menu "Project Management". Untuk mengakses halaman ini, *Admin* cukup menekan tombol "Edit" pada baris proyek yang ingin diperbarui.



Gambar 3.47. Tampilan Mengedit Data Proyek di "Project Management"

Sistem kemudian akan menampilkan formulir yang telah terisi dengan data proyek sebelumnya. Pada tampilan ini, *Admin* dapat mengubah seluruh informasi proyek seperti nama proyek, kode proyek, tanggal mulai dan berakhir, deskripsi, serta informasi lainnya sesuai kebutuhan. Perlu dicatat bahwa status proyek akan berubah secara otomatis menjadi **Completed** apabila tanggal berakhir proyek sudah terlewati. Jika status proyek sudah berubah menjadi **Completed**, maka status tersebut tidak dapat diubah kembali menjadi **On Progress**. Setelah seluruh pembaruan selesai dilakukan, *Admin* dapat menekan tombol "Save Changes" untuk menyimpan perubahan.

D.5 Menghapus Project

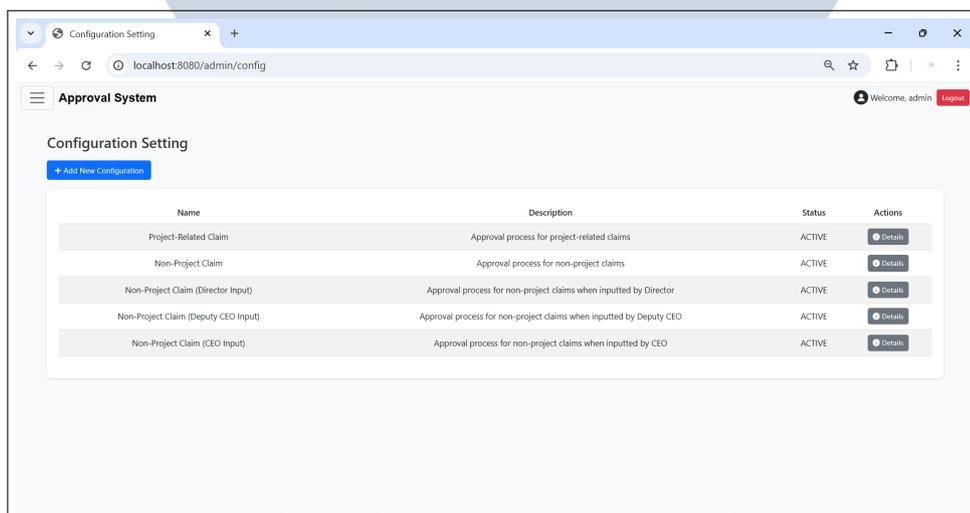
Admin dapat menghapus data proyek yang sudah tidak diperlukan melalui halaman menu "Project Management". Untuk melakukan penghapusan, *Admin* cukup menekan tombol "Delete" yang tersedia di baris proyek yang ingin dihapus. Setelah tombol tersebut ditekan, sistem akan menampilkan dialog konfirmasi untuk memastikan bahwa tindakan penghapusan benar-benar diinginkan. Hal ini bertujuan untuk mencegah terjadinya penghapusan data secara tidak sengaja. Jika *Admin* menekan tombol "Yes, Delete", sistem akan langsung menghapus data proyek dari basis data secara permanen. Namun, apabila *Admin* memilih untuk membatalkan penghapusan dengan menekan tombol "Cancel", maka tidak akan ada perubahan yang dilakukan terhadap data proyek.

E Configuration Management

Bagian ini memberikan penjelasan mengenai tampilan dari menu "Configuration Management" untuk *Admin*. Seperti yang sudah disebutkan sebelumnya, menu "Configuration Management" sendiri hanya memperbolehkan *Admin* untuk melihat dan menambahkan konfigurasi.

E.1 Tampilan List Configuration

Bagian ini memberikan penjelasan mengenai tampilan *list* konfigurasi dari menu Configuration Management. Pada halaman ini, *admin* dapat melihat seluruh entri konfigurasi yang telah terdaftar dalam sistem, lengkap dengan informasi penting seperti nama konfigurasi, deskripsi singkat, serta urutan persetujuan yang terkait.



The screenshot shows a web browser window with the URL localhost:8080/admin/config. The page title is "Approval System" and the user is logged in as "admin". The main content area is titled "Configuration Setting" and contains a table with the following data:

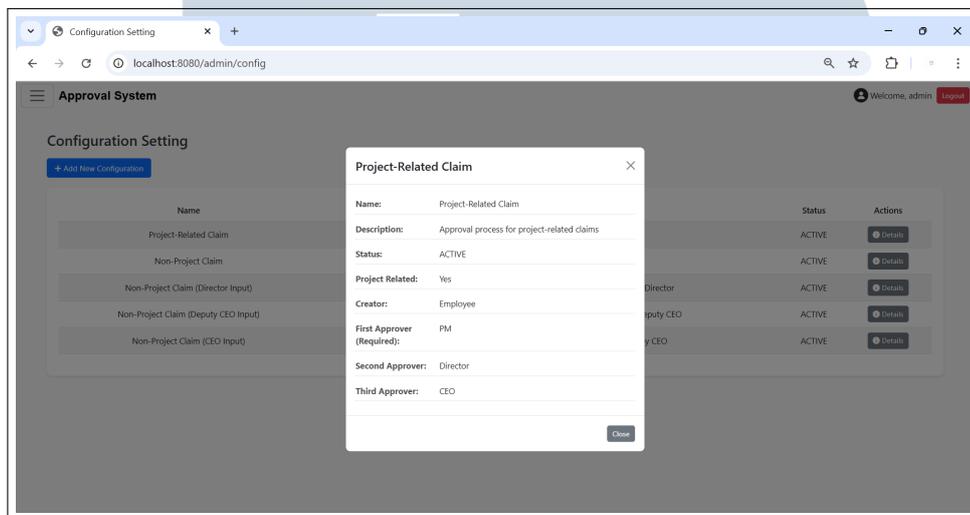
Name	Description	Status	Actions
Project-Related Claim	Approval process for project-related claims	ACTIVE	Details
Non-Project Claim	Approval process for non-project claims	ACTIVE	Details
Non-Project Claim (Director Input)	Approval process for non-project claims when inputted by Director	ACTIVE	Details
Non-Project Claim (Deputy CEO Input)	Approval process for non-project claims when inputted by Deputy CEO	ACTIVE	Details
Non-Project Claim (CEO Input)	Approval process for non-project claims when inputted by CEO	ACTIVE	Details

Gambar 3.48. Tampilan *List* Konfigurasi di "Configuration Management"

Gambar 3.48 menunjukkan tampilan daftar konfigurasi yang tersedia pada menu "Configuration Management". Tampilan ini memiliki struktur yang serupa dengan halaman menu "User Management" dan "Project Management", di mana *Admin* dapat melihat daftar seluruh konfigurasi yang telah terdaftar dalam sistem dalam bentuk tabel yang rapi dan terstruktur. Selain itu, dalam halaman ini juga disediakan tombol aksi yaitu tombol "Detail" pada setiap entri konfigurasi. Fitur ini memungkinkan *Admin* untuk melihat informasi lebih lanjut mengenai konfigurasi yang bersangkutan.

E.2 Menampilkan Detail Configuration

Gambar 3.49 di bawah menunjukkan tampilan detail informasi konfigurasi yang muncul ketika *Admin* menekan tombol "View" pada salah satu entri di halaman menu "Configuration Management". Halaman ini dirancang untuk membantu *Admin* meninjau setiap konfigurasi yang telah ditetapkan dalam sistem, terutama yang berkaitan dengan alur persetujuan klaim.

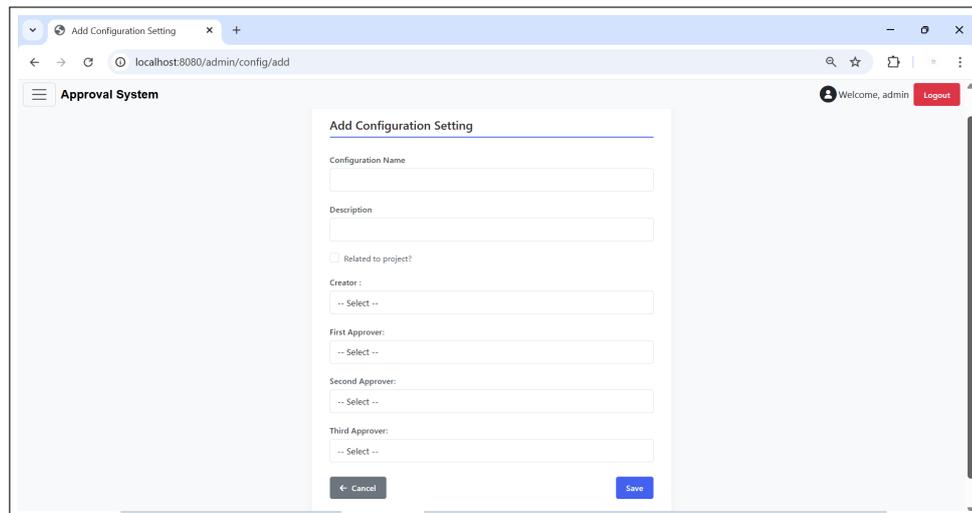


Gambar 3.49. Tampilan Detail Konfigurasi di "Configuration Management"

Tampilan ini menyajikan informasi lengkap mengenai konfigurasi yang dipilih, seperti nama konfigurasi, deskripsi, status aktif/tidak aktif, serta urutan persetujuan yang berlaku. Informasi ini penting bagi *Admin* untuk memastikan bahwa aturan dan struktur persetujuan klaim telah diatur dengan benar sesuai dengan kebutuhan organisasi. Melalui halaman ini, *Admin* dapat melakukan validasi terhadap konfigurasi yang sedang berjalan tanpa perlu masuk ke mode pengeditan.

E.3 Membuat Configuration Baru

Gambar 3.50 di bawah menunjukkan tampilan antarmuka saat *Admin* ingin menambahkan konfigurasi baru pada halaman menu "Configuration Management". Konfigurasi ini berfungsi untuk menetapkan aturan atau parameter sistem, seperti alur persetujuan klaim berdasarkan jabatan atau kategori tertentu.



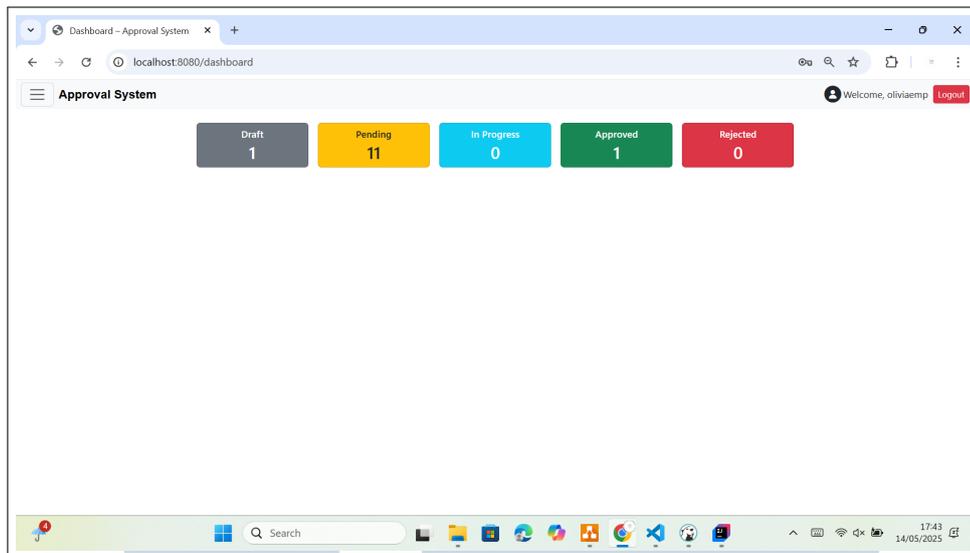
Gambar 3.50. Tampilan Membuat Konfigurasi Baru di "Configuration Management"

Pada halaman ini, *Admin* dapat mengisi formulir yang tersedia dengan informasi penting seperti nama konfigurasi, deskripsi, status aktif, serta urutan alur persetujuan yang diinginkan. Setelah seluruh data diisi dengan benar, *Admin* dapat menekan tombol "Save" untuk menyimpan konfigurasi tersebut ke dalam sistem. Jika *Admin* memutuskan untuk membatalkan proses penambahan dan kembali ke halaman sebelumnya, tombol "Cancel" dapat digunakan tanpa menyimpan perubahan.

F Dashboard Maker

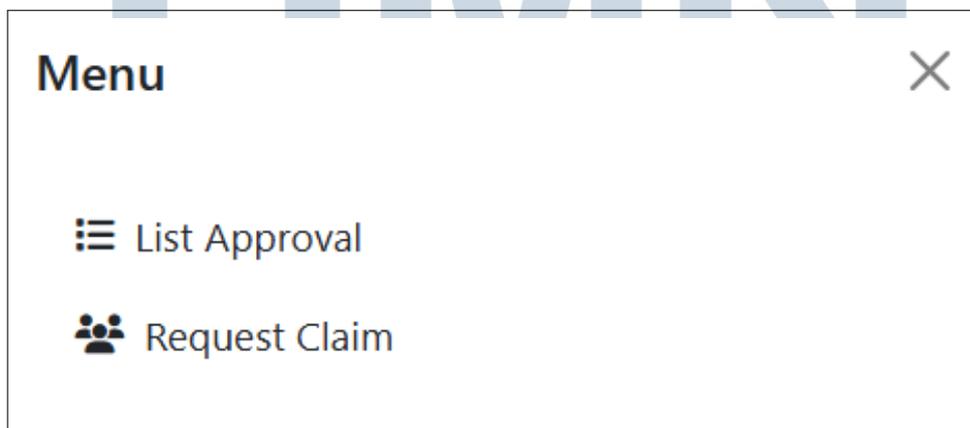
Bagian ini memberikan penjelasan mengenai tampilan halaman *dashboard* yang digunakan oleh pengguna dengan peran *Maker*. *Dashboard* ini merupakan halaman utama yang ditampilkan setelah pengguna berhasil masuk ke dalam sistem, dan menyajikan ringkasan aktivitas yang dapat dilakukan oleh *Maker*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.51. Tampilan *Dashboard* untuk *Maker*

Gambar 3.51 menampilkan tampilan utama dari *dashboard Maker*. Pada halaman ini, pengguna dapat melihat informasi ringkas mengenai klaim yang pernah diajukan, termasuk status terkini dari setiap *submission*. Di bagian atas terdapat *navbar* utama yang dilengkapi dengan tombol *hamburger* di kiri atas untuk membuka atau menutup *sidebar* menu, teks “Approval System” yang dapat diklik untuk kembali ke *dashboard*, serta tombol “Logout” di kanan atas yang digunakan untuk keluar dari sistem. Tampilan dan fungsionalitas *navbar* ini serupa dengan *dashboard* untuk *Admin*, namun konten dan menu di dalamnya telah disesuaikan dengan kebutuhan peran *Maker*.



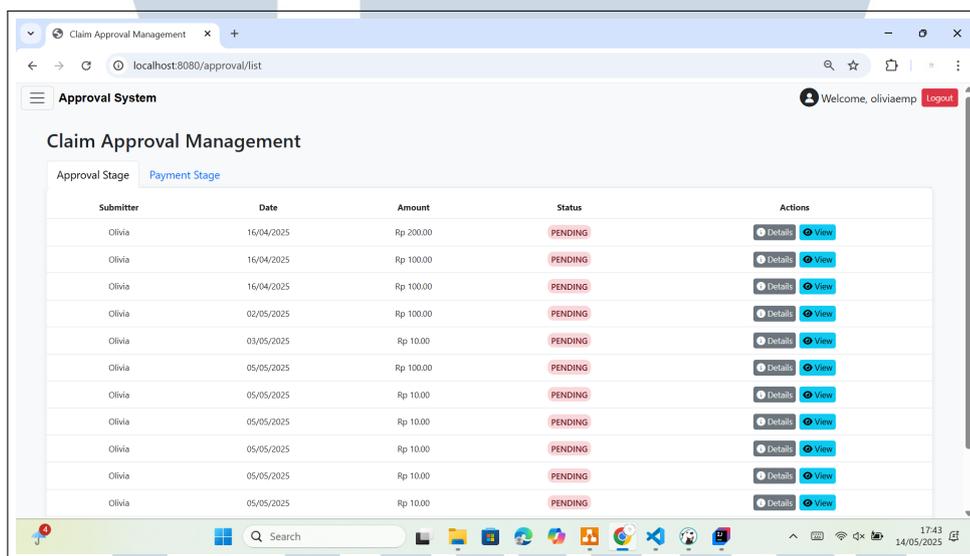
Gambar 3.52. Tampilan *Sidebar* untuk *Maker*

Gambar 3.52 menunjukkan tampilan *sidebar* menu yang digunakan oleh

Maker. *Sidebar* ini berisi beberapa menu utama yang berkaitan langsung dengan tugas dan fungsinya, seperti "List Approval" untuk melihat status *submission* yang telah diajukan, dan "Request Claim" untuk membuat *submission* baru beserta klaim-klaim di dalamnya. Tampilan *sidebar* ini disusun agar mudah diakses dan intuitif, sehingga dapat mempercepat proses kerja pengguna dalam mengelola klaim.

G List Approval

Bagian ini memberikan penjelasan mengenai tampilan menu "List Approval" yang digunakan oleh pengguna dengan peran *Maker*. Menu ini digunakan untuk memantau status klaim yang telah diajukan, baik yang masih menunggu persetujuan maupun yang sudah ditindaklanjuti oleh *Approver*.

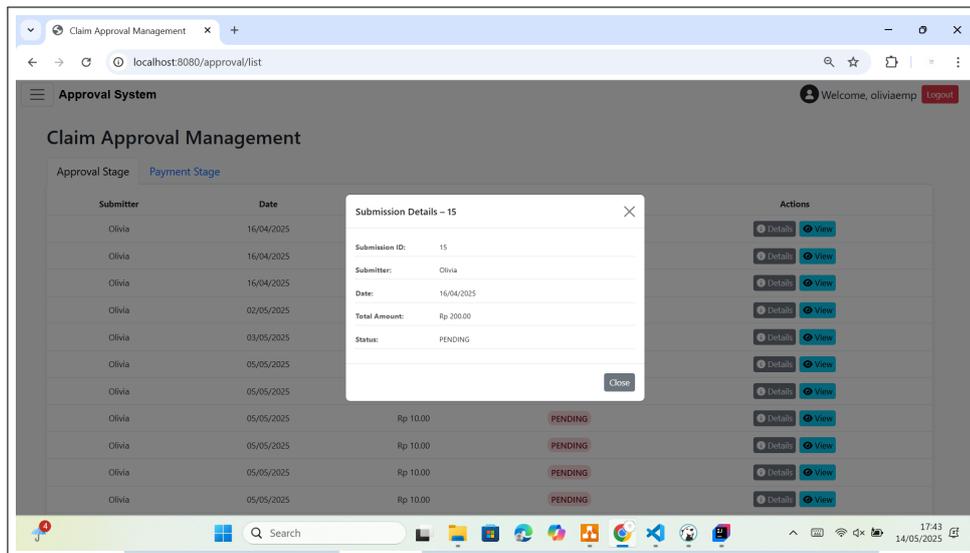


The screenshot shows a web browser window with the URL localhost:8080/approval/list. The page title is "Approval System" and the user is logged in as "oliviaemp". The main heading is "Claim Approval Management". There are two tabs: "Approval Stage" (selected) and "Payment Stage". Below the tabs is a table with the following columns: Submitter, Date, Amount, Status, and Actions. The table contains 12 rows of data, all with a status of "PENDING".

Submitter	Date	Amount	Status	Actions
Olivia	16/04/2025	Rp 200.00	PENDING	Details View
Olivia	16/04/2025	Rp 100.00	PENDING	Details View
Olivia	16/04/2025	Rp 100.00	PENDING	Details View
Olivia	02/05/2025	Rp 100.00	PENDING	Details View
Olivia	03/05/2025	Rp 10.00	PENDING	Details View
Olivia	05/05/2025	Rp 100.00	PENDING	Details View
Olivia	05/05/2025	Rp 10.00	PENDING	Details View
Olivia	05/05/2025	Rp 10.00	PENDING	Details View
Olivia	05/05/2025	Rp 10.00	PENDING	Details View
Olivia	05/05/2025	Rp 10.00	PENDING	Details View
Olivia	05/05/2025	Rp 10.00	PENDING	Details View

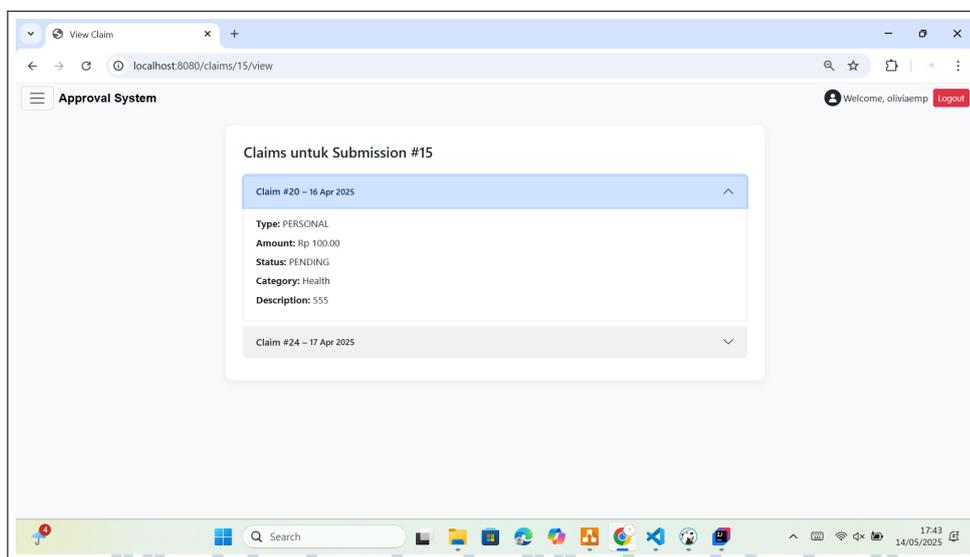
Gambar 3.53. Tampilan *List Approval*

Gambar 3.53 memperlihatkan halaman daftar klaim dan *submission* yang telah dibuat oleh *Maker*, lengkap dengan informasi seperti tanggal pengajuan, nominal klaim, status persetujuan, serta nama *Approver* terkait. Setiap baris dalam tabel dilengkapi dengan dua tombol aksi, yaitu "View" dan "Detail". Tombol "View" digunakan untuk melihat informasi detail dari masing-masing klaim, sedangkan tombol "Detail" digunakan untuk menampilkan informasi lengkap dari satu *submission* beserta seluruh klaim di dalamnya.



Gambar 3.54. Tampilan Detail untuk *List Approval* (Bagian 1)

Gambar 3.54 menampilkan informasi awal dari suatu *submission*, seperti *submission ID*, *submitter*, tanggal, *total amount*, dan status. Jika statusnya *Rejected* maka akan muncul juga kolom alasan penolakan. Halaman ini ditampilkan setelah pengguna menekan tombol *Detail* pada daftar *submission* di halaman sebelumnya.



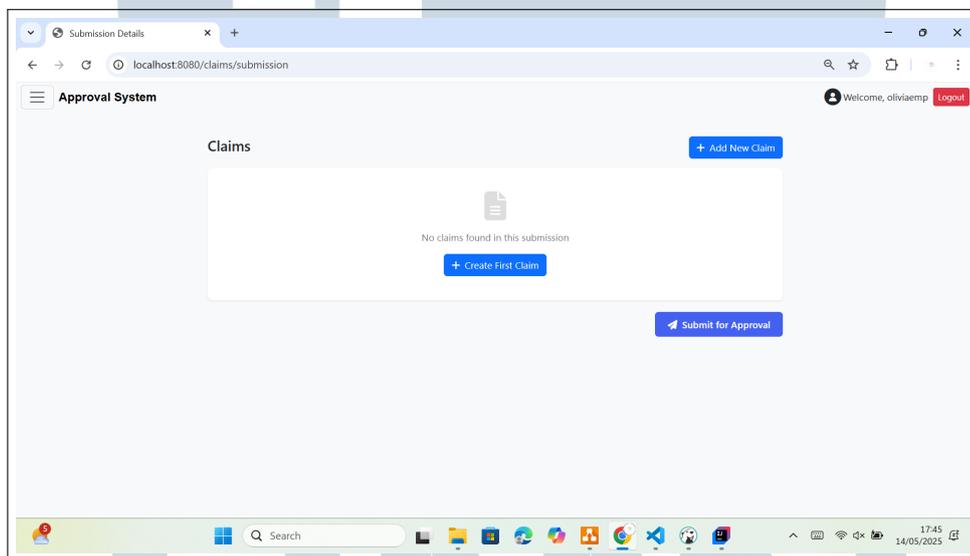
Gambar 3.55. Tampilan Detail untuk *List Approval* (Bagian 2)

Gambar 3.55 melanjutkan tampilan sebelumnya dengan menyajikan daftar klaim individual yang terdapat dalam *submission* tersebut. Setiap klaim ditampilkan dengan informasi seperti kategori, nominal, tanggal nota, status validasi, serta tombol untuk melihat detail klaim. Halaman ini membantu *Maker* untuk melakukan

pelacakan secara menyeluruh terhadap semua klaim yang telah diajukannya dalam satu *submission*.

H Request Claim

Bagian ini memberikan penjelasan mengenai tampilan menu "Request Claim", yaitu fitur yang digunakan oleh pengguna dengan peran *Maker* untuk mengajukan klaim secara mandiri. Fitur ini mencakup proses pembuatan, pengeditan, pengunggahan dokumen, hingga pengiriman klaim untuk proses persetujuan.

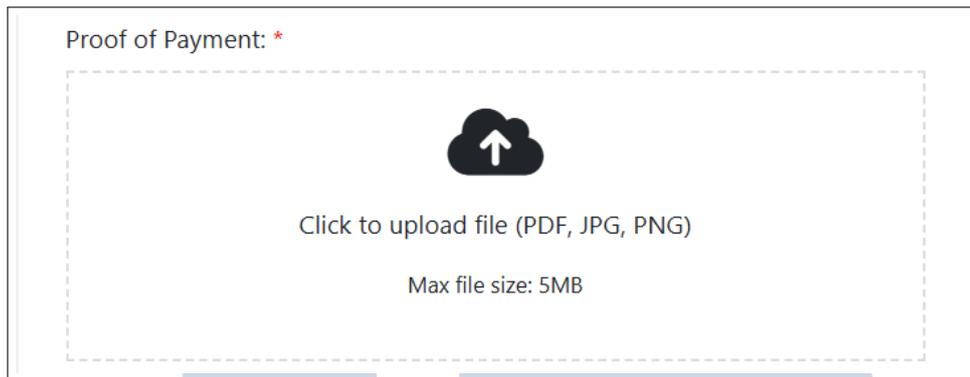


Gambar 3.56. Tampilan Menu "Request Claim"

Gambar 3.56 menunjukkan tampilan awal ketika pengguna dengan peran *Maker* ingin membuat klaim. Pada halaman ini, terdapat dua tombol utama, yaitu "Add New Claim" dan "Create First Claim", yang keduanya akan mengarahkan pengguna ke halaman pembuatan klaim baru. Tombol "Create First Claim" hanya akan muncul apabila pengguna belum pernah membuat klaim sebelumnya, sedangkan tombol "Add New Claim" akan selalu tersedia, memungkinkan pengguna untuk membuat klaim baru kapan pun diperlukan, tanpa batasan jumlah. Dengan demikian, pengguna dapat mengajukan klaim sebanyak yang dibutuhkan.

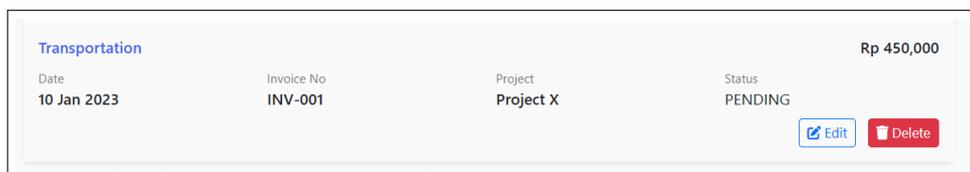
Gambar 3.57. Tampilan *Add Claim*

Gambar 3.57 menunjukkan tampilan halaman ketika pengguna dengan peran *Maker* sedang membuat klaim baru. Pada halaman ini, pengguna terlebih dahulu diminta untuk memilih jenis klaim yang ingin diajukan, yaitu *Personal*, *Project*, atau *Company*. Jika pengguna memilih klaim *Personal*, maka form yang ditampilkan memungkinkan pengguna untuk mengajukan klaim yang bersifat umum seperti kesehatan atau kebutuhan seperti alat olahraga. Sementara itu, jika memilih klaim *Company*, maka form yang muncul hanya ditujukan untuk keperluan perusahaan seperti peralatan kantor. Jika memiliki klaim *Project*, maka pengguna bisa mengajukan klaim yang berkaitan dengan proyek seperti transportasi. Setiap jenis klaim memiliki struktur *field* yang berbeda, menyesuaikan kebutuhan dari masing-masing kategori klaim. Setelah pengguna mengisi data dengan lengkap dan sesuai, pengguna dapat menekan tombol "Submit" untuk melanjutkan proses pengajuan klaim. Namun, jika pengguna memutuskan untuk membatalkan proses pembuatan klaim, tersedia juga tombol "Cancel" yang dapat digunakan untuk kembali ke halaman sebelumnya tanpa menyimpan perubahan apa pun. Seperti telah dijelaskan sebelumnya, *Maker* dapat menambahkan klaim sebanyak yang dibutuhkan, namun seluruh klaim dalam satu pengajuan harus memiliki tipe yang sama. Ketentuan ini diterapkan untuk menjaga konsistensi data serta mempermudah proses verifikasi dan persetujuan.



Gambar 3.58. Tampilan *Upload File*

Gambar 3.58 menunjukkan tampilan di dalam form klaim di mana pengguna dapat mengunggah *file* pendukung. Fitur unggah *file* ini disediakan di bagian bawah form pembuatan klaim, baik untuk klaim *Personal*, *Project*, maupun *Company*. Pengguna dapat mengunggah dokumen seperti nota, atau bukti pendukung lainnya yang relevan dengan klaim yang diajukan. Sistem ini menggunakan penyimpanan berbasis MinIO, yaitu layanan penyimpanan objek yang di *host* secara lokal, yang memungkinkan *file* yang diunggah disimpan dengan aman dan efisien. Setelah *file* berhasil diunggah, sistem akan menyimpan tautan *file* tersebut dan mengaitkannya langsung dengan klaim yang bersangkutan.



Gambar 3.59. Tampilan *Claim*

Gambar 3.59 menunjukkan tampilan daftar klaim yang telah dibuat oleh pengguna dengan peran *Maker*. Pada tampilan ini, pengguna dapat melakukan beberapa aksi terhadap klaim yang masih dalam status *draft*, seperti menambahkan entri baru, mengedit klaim yang sudah ada, atau menghapus klaim jika diperlukan. Setelah seluruh klaim telah diisi dengan lengkap dan sesuai, *Maker* dapat menekan tombol "Submit" untuk mengirimkan klaim tersebut ke tahap proses selanjutnya, yaitu proses *approval*.

Gambar 3.60. Tampilan *Edit Claim*

Gambar 3.60 menunjukkan tampilan saat pengguna dengan peran *Maker* ingin mengedit klaim yang telah dibuat. Selama klaim belum dikirim ke tahap *approval*, pengguna masih dapat mengubah isi klaim sesuai kebutuhan. Namun, perlu dicatat bahwa tipe klaim (*personal*, *project*, atau *company*) yang telah dipilih saat pembuatan awal tidak dapat diubah pada tahap *edit* ini. *Field* lainnya akan ditampilkan sesuai dengan tipe klaim yang telah dipilih sebelumnya.

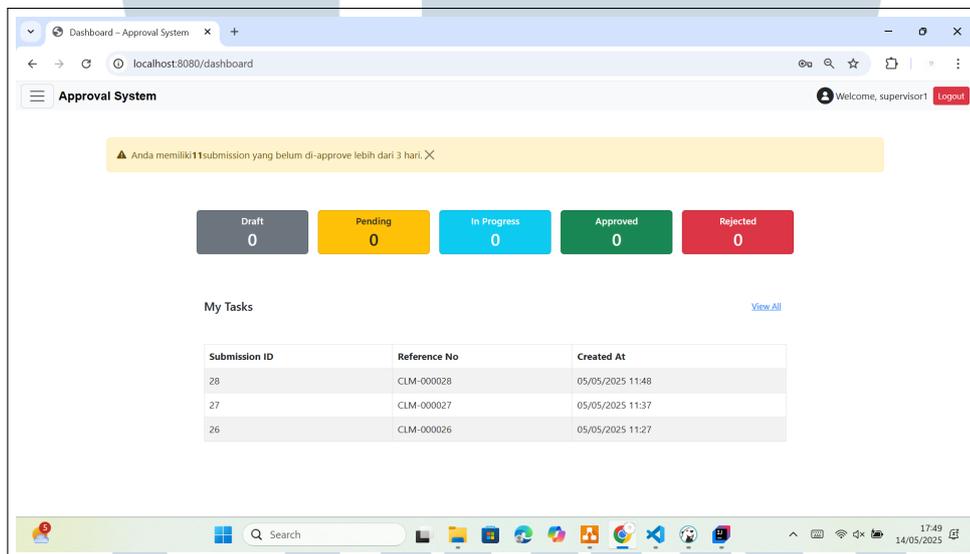
Gambar 3.61. Tampilan *Submit Claim*

Gambar 3.61 menunjukkan tampilan yang muncul ketika pengguna dengan peran *Maker* menekan tombol "Submit" pada halaman klaim. Setelah tombol ditekan, sistem akan menampilkan *pop-up* konfirmasi sebagai langkah untuk

memastikan bahwa pengguna benar-benar ingin mengirim klaim ke proses *approval*. Pada *pop-up* ini, tersedia dua pilihan, yaitu "Cancel" untuk membatalkan pengiriman dan kembali ke halaman sebelumnya, serta "Confirm" untuk melanjutkan dan mengirim klaim ke tahapan berikutnya.

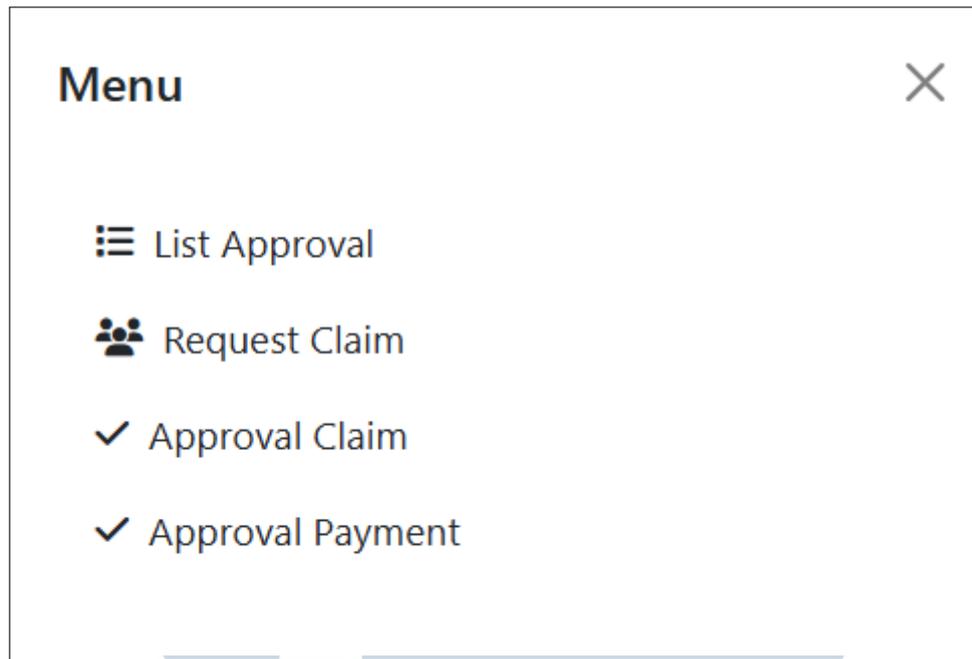
I Dashboard Approver

Bagian ini memberikan penjelasan mengenai tampilan halaman *dashboard* yang digunakan oleh pengguna dengan peran *Approver*. *Dashboard* ini dirancang untuk memudahkan pengguna dalam memantau klaim yang diajukan dan klaim yang perlu diproses.



Gambar 3.62. Tampilan *Dashboard* untuk *Approver*

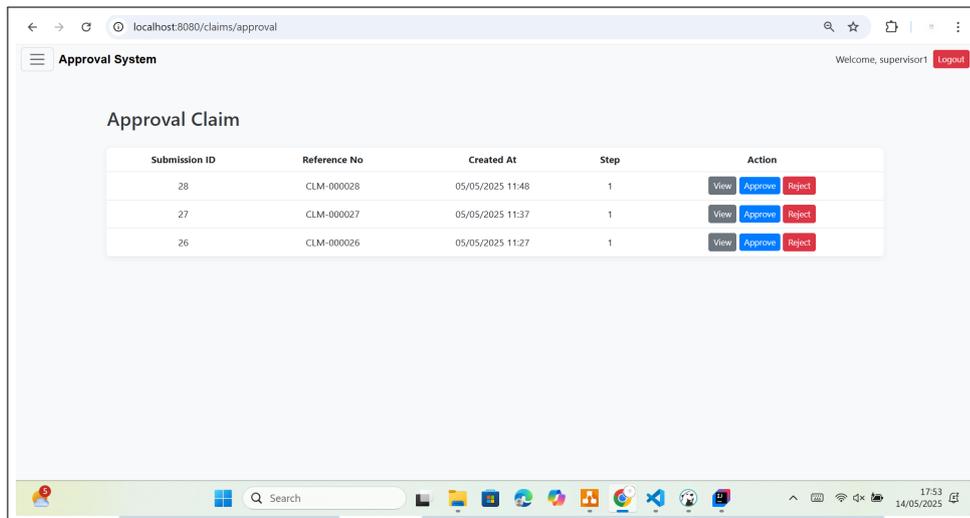
Gambar 3.62 menampilkan tampilan *dashboard* utama yang dilihat oleh pengguna dengan peran *Approver*. Perlu diketahui bahwa seorang *Approver* bisa juga berperan sebagai *Maker*, namun *Maker* tidak memiliki peran sebagai *Approver*. Pada halaman ini, pengguna dapat dengan mudah mengakses ringkasan informasi terkait klaim-klaim yang telah diajukan oleh dirinya sendiri. Selain itu, *Approver* juga dapat melihat daftar *submission* klaim yang sedang menunggu persetujuan, termasuk pemberitahuan khusus untuk klaim yang terlambat diproses atau sudah menumpuk lebih dari 3 hari agar dapat segera ditindaklanjuti. Tampilan *dashboard* ini dilengkapi dengan *navigation bar* (*navbar*) yang sama seperti yang digunakan oleh peran *Maker* dan *Admin*.



Gambar 3.63. Tampilan *Sidebar* untuk *Approver*

Gambar 3.63 menampilkan *sidebar* yang digunakan oleh pengguna dengan peran *Approver*. Pada *sidebar* ini, pengguna dapat dengan mudah mengakses beberapa menu utama, yaitu fitur untuk mengajukan klaim (*request claim*), menyetujui klaim dari pengguna lain (*approve other claim*), serta menyetujui pembayaran (*approve payment*). Karena tampilan menu "List Approval" dan "Request Claim" pada pengguna *Approver* memiliki kesamaan dengan tampilan milik *Maker*, maka bagian ini tidak akan dijelaskan ulang. Penjelasan akan langsung difokuskan pada proses persetujuan (*approval*) klaim yang dilakukan oleh *Approver*, termasuk bagaimana alur kerja, tampilan form persetujuan, serta interaksi yang dilakukan pengguna dalam menyetujui atau menolak klaim yang diajukan.

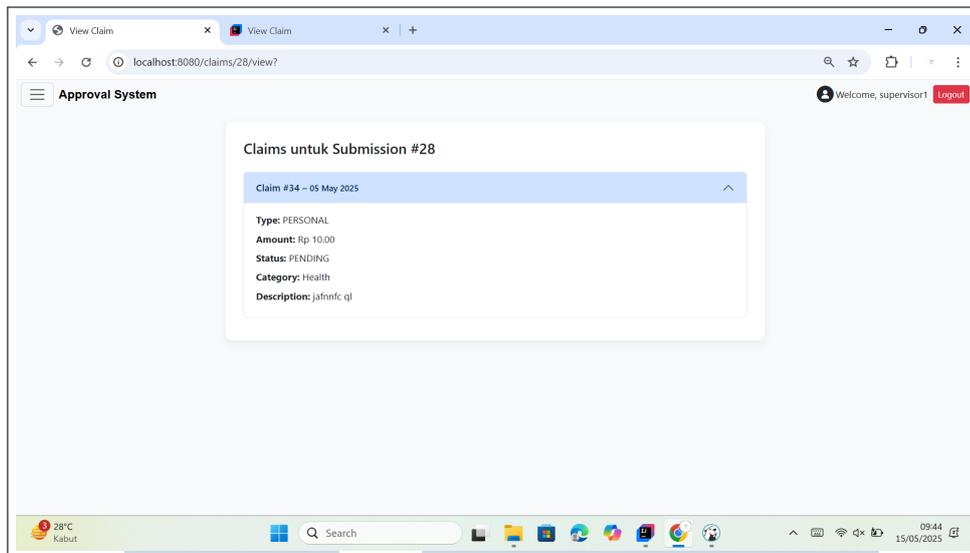
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.64. Tampilan *Approval* untuk *Approver*

Gambar 3.64 menampilkan tampilan saat pengguna dengan peran *Approver* akan melakukan proses persetujuan klaim. Pada halaman ini, *Approver* diberikan dua opsi utama, yaitu langsung menyetujui (*Approve*) atau menolak (*Reject*) klaim yang diajukan. Sebelum mengambil keputusan, *Approver* juga memiliki opsi untuk melihat detail klaim terlebih dahulu melalui tombol "View", sehingga dapat memastikan informasi yang diberikan telah lengkap dan sesuai. Setelah keputusan diambil, sistem akan memproses aksi tersebut, serta mengirimkan notifikasi kepada pihak terkait. Jika klaim telah disetujui oleh seluruh *Approver* sesuai alur persetujuan yang ditentukan, maka status klaim akan diperbarui secara otomatis menjadi **Approved** oleh sistem.

U M N
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.65. Tampilan *View Approval*

3.4 Kendala dan Solusi yang Ditemukan

Selama menjalani magang di PT Soluix Finteknologi Indonesia, terdapat beberapa kendala yang dihadapi, terutama terkait keterbatasan pengetahuan dan pengalaman teknis pada tahap awal.

Beberapa tantangan yang muncul selama proses implementasi sistem antara lain:

1. Kendala teknis dalam merancang alur persetujuan yang cukup rumit karena adanya berbagai tipe klaim dengan proses persetujuan yang berbeda-beda serta melibatkan banyak pihak terkait.
2. Kendala dalam pengelolaan token dan sesi pengguna, terutama masalah pada penyimpanan dan validasi token yang menyebabkan token tidak tersimpan dengan baik.

Guna menjawab tantangan yang muncul selama proses magang, dilakukan beberapa penyesuaian dan strategi pemecahan masalah sebagai berikut:

1. Meningkatkan pemahaman melalui pembelajaran mandiri menggunakan sumber terpercaya seperti YouTube.
2. Berkonsultasi secara rutin dengan supervisor untuk mendapatkan arahan dan solusi.