



PERANCANGAN SISTEM DIGITAL UNTUK MANAJEMEN DAN PENYELESAIAN KELUHAN TRANSAKSI KEUANGAN DI PT GEMA MERAKI SUKSES

Bertalia Naul¹, Farhan Hidinillah², Lutfi Nurwasfiah³, Ari Syaripudin⁴

¹⁻⁴fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Pamulang, Tangerang Selatan Banten Indonesia

Email: 1Bertalianaul51@gmail.com, 2fiproject714@gmail.com, 3wasfiah35@gmail.com,
4dosen00671@unpam.ac.id

A B S T R A K

Sistem manajemen keluhan merupakan komponen krusial dalam mendukung kualitas layanan dan kepuasan pelanggan di industri jasa keuangan. Namun, proses penanganan keluhan transaksi remitansi yang masih diterapkan secara manual di PT Gema Meraki Sukses menimbulkan berbagai kendala, seperti keterlambatan penanganan, risiko kehilangan data, dan kurangnya transparansi bagi klien. Penelitian ini bertujuan untuk merancang dan mengembangkan sistem manajemen keluhan berbasis *web* dengan alur kerja yang terstruktur guna meningkatkan efisiensi dan akurasi proses penyelesaian keluhan di PT Gema Meraki Sukses. Metode pengembangan sistem yang digunakan adalah model *Waterfall*, yang mencakup tahapan analisis kebutuhan, perancangan, implementasi, dan pengujian. Hasil penelitian menunjukkan bahwa sistem yang dibangun dapat mempermudah klien dalam mengajukan dan melacak keluhan secara *real-time*, memungkinkan supervisor untuk menindaklanjuti keluhan secara sistematis, serta memberikan admin kemampuan untuk mengelola seluruh proses dan pengguna. Sistem ini dikembangkan menggunakan HTML, CSS, dan JavaScript untuk *frontend*, serta Node.js dengan framework Express.js untuk *backend*. Keamanan otentikasi dijamin dengan JSON Web Token (JWT) melalui arsitektur REST API. Dengan implementasi sistem ini, proses administrasi dan penyelesaian keluhan di PT Gema Meraki Sukses menjadi lebih efektif, efisien, dan terintegrasi.

Kata kunci : sistem manajemen keluhan, *web-based*, transaksi keuangan, PT Gema Meraki Sukses, *Waterfall*.

A B S T R A C T

Management system is a crucial component in supporting service quality and customer satisfaction in the financial services industry. However, the manual handling process for remittance transaction complaints still implemented at PT Gema Meraki Sukses creates various obstacles, such as handling delays, the risk of data loss, and a lack of transparency for clients. This study aims to design and develop a web-based complaint management system with a structured workflow to improve the efficiency and accuracy of the complaint resolution process at PT Gema Meraki Sukses. The system development method used is the Waterfall model, which includes the stages of requirements analysis, design, implementation, and testing. The results show that the developed system facilitates clients in

Article History

Received: Juni 2025

Reviewed: Juni 2025

Published: Juni 2025

Plagirism Checker No
234

Prefix DOI : Prefix DOI :
10.8734/Kohesi.v1i2.365

Copyright : Author

Publish by : Kohesi



This work is licensed
under a [Creative
Commons Attribution-
NonCommercial 4.0
International License](https://creativecommons.org/licenses/by-nc/4.0/)



submitting and tracking complaints in real-time, enables supervisors to follow up on complaints systematically, and provides administrators with the ability to manage the entire process and users. The system was developed using HTML, CSS, and JavaScript for the frontend, and Node.js with the Express.js framework for the backend. Authentication security is ensured with JSON Web Token (JWT) through a REST API architecture. With the implementation of this system, the complaint administration and resolution process at PT Gema Meraki Sukses becomes more effective, efficient, and integrated.

Keywords: *complaint management system, web-based, financial transaction, PT Gema Meraki Sukses, Waterfall.*

1. PENDAHULUAN

Manajemen keluhan pelanggan adalah komponen penting dalam operasional perusahaan jasa keuangan untuk menjaga kepercayaan dan loyalitas. Di PT Gema Meraki Sukses, sebuah perusahaan yang fokus pada layanan remitansi, proses penanganan keluhan terkait transaksi masih umum dilakukan secara manual melalui email, telepon, dan pencatatan terpisah. Hal ini sering menimbulkan masalah seperti keterlambatan respons, data yang tidak terpusat, dan kesulitan bagi klien untuk melacak status penyelesaian keluhan mereka. Dalam lingkungan bisnis yang kompetitif, waktu dan transparansi menjadi faktor krusial yang memengaruhi kepuasan pelanggan.

Dengan perkembangan teknologi informasi, sistem manajemen keluhan berbasis *web* menjadi solusi praktis yang memungkinkan pencatatan terpusat, alur kerja yang jelas, laporan otomatis, dan akses mudah bagi semua pihak terkait. Beberapa studi sebelumnya telah menunjukkan bahwa implementasi sistem digital untuk layanan pelanggan dapat meningkatkan efisiensi operasional dan akurasi data secara signifikan. Namun, di PT Gema Meraki Sukses, belum ada sistem komprehensif yang mengintegrasikan kebutuhan klien, supervisor, dan admin dalam satu *platform* terpadu. Maka, penelitian ini bertujuan merancang dan mengimplementasikan Sistem Digital untuk Manajemen dan Penyelesaian Keluhan Transaksi Keuangan guna meningkatkan efisiensi, transparansi, dan akuntabilitas dalam penanganan keluhan di PT Gema Meraki Sukses.

2. METODE PENELITIAN

2.1 Pendekatan Penelitian

Penelitian ini menggunakan pendekatan metode campuran (*mixed methods*) yang menggabungkan metode kualitatif dan kuantitatif untuk mendapatkan gambaran yang komprehensif mengenai proses penanganan keluhan transaksi keuangan di PT Gema Meraki Sukses.

Secara kualitatif, data dikumpulkan melalui observasi langsung terhadap alur kerja pengelolaan keluhan dan wawancara mendalam dengan beberapa pihak terkait, seperti staf administrasi, supervisor, dan klien. Salah satunya adalah wawancara dengan staf admin untuk memahami kendala operasional serta kebutuhan pengguna terhadap sistem informasi pengelolaan keluhan. Sementara secara kuantitatif, penelitian ini juga memanfaatkan data statistik, seperti jumlah keluhan per hari/bulan, waktu rata-rata penyelesaian, dan jenis keluhan, yang digunakan untuk analisis kebutuhan sistem serta pembuatan grafik dan *dashboard* monitoring. Data ini berfungsi sebagai dasar evaluasi efektivitas sistem yang akan dikembangkan.

Dengan pendekatan campuran ini, hasil penelitian diharapkan mampu memberikan pemahaman yang lebih menyeluruh terhadap permasalahan dan kebutuhan di lapangan, sehingga dapat mendukung perancangan sistem yang efisien, akurat, dan sesuai dengan kebutuhan pengguna.



2.2 Metode Pengembangan Sistem

Metode yang digunakan dalam pengembangan sistem ini adalah metode *Waterfall*. Model ini merupakan salah satu metode pengembangan perangkat lunak klasik yang sistematis, di mana setiap fase harus diselesaikan sebelum melanjutkan ke fase berikutnya. Model *Waterfall* cocok diterapkan karena ruang lingkup dan kebutuhan sistem telah terdefinisi dengan baik sejak awal. Adapun tahapan-tahapan dalam model *Waterfall* yang digunakan pada penelitian ini adalah sebagai berikut:

1. **Analisis Kebutuhan (*Requirement Analysis*)**
Pada tahap ini dilakukan identifikasi kebutuhan sistem melalui observasi dan wawancara. Kebutuhan fungsional (seperti pengajuan keluhan oleh klien, validasi oleh supervisor, manajemen pengguna oleh admin) dan kebutuhan non-fungsional (keamanan, kemudahan penggunaan) dirumuskan sebagai dasar pengembangan.
2. **Perancangan Sistem (*System Design*)** Setelah kebutuhan dikumpulkan, dilakukan perancangan sistem yang meliputi perancangan basis data (ERD), arsitektur REST API, serta diagram alur proses (*Use Case* dan *Activity Diagram*). Desain antarmuka (UI) juga dibuat menggunakan pendekatan
3. **Implementasi (*Implementation*)**
Tahap ini adalah proses pembangunan aplikasi menggunakan teknologi yang telah ditetapkan: HTML, CSS, dan JavaScript untuk *frontend*, serta Node.js dengan framework Express.js untuk backend. Keamanan ditangani dengan JWT.
4. **Pengujian (*Testing*)**
Pengujian dilakukan menggunakan metode *black-box testing*, di mana setiap fungsi sistem diuji berdasarkan input dan output-nya untuk memastikan kesesuaian dengan kebutuhan fungsional. Selain itu, dilakukan uji coba langsung oleh calon pengguna untuk menilai kemudahan penggunaan.
5. **Pemeliharaan (*Maintenance*)**
Tahap ini mencakup perbaikan *bug* yang ditemukan saat implementasi awal serta pemberian dokumentasi teknis untuk keperluan pengembangan di masa mendatang.

2.3 Lokasi dan Waktu Penelitian

Penelitian ini dilaksanakan di PT Gema Meraki Sukses yang berlokasi di Menara Rajawali Lantai 8, Jl. Dr. Ide Anak Agung Gde Agung Lot 5.1, Kawasan Mega Kuningan, Jakarta 12950 - Indonesia. Kegiatan penelitian dan pengembangan sistem dilaksanakan mulai bulan April hingga Juni 2025.

3. ANALISA DAN PEMBAHASAN

Berdasarkan hasil observasi yang dilakukan di PT Gema Meraki Sukses, sistem manajemen keluhan yang digunakan sebelumnya masih bersifat manual dan belum terintegrasi. Keluhan dari klien diterima melalui berbagai kanal komunikasi seperti email, WhatsApp, dan Telegram, kemudian dicatat oleh staf administrasi ke dalam master data berbasis spreadsheet (seperti *Microsoft Excel* atau *Google Sheets*). Informasi tersebut selanjutnya diteruskan secara lisan atau dikirimkan kembali melalui email kepada supervisor untuk ditindaklanjuti.

Beberapa permasalahan utama yang ditemukan dalam sistem manual ini antara lain:

1. Tidak adanya pusat data yang terintegrasi
Data keluhan tersimpan di berbagai media dan file spreadsheet yang terpisah, sehingga meningkatkan risiko kehilangan, duplikasi, atau inkonsistensi data. Sistem pencatatan ini juga menyulitkan pencarian data historis secara cepat dan akurat.
2. Rekapitulasi dan pelaporan memakan waktu lama
Karena tidak adanya sistem otomatis, staf harus mengolah data secara manual dari spreadsheet untuk menyusun laporan mingguan/bulanan, yang rentan terhadap kesalahan dan membutuhkan waktu lama.
3. Minimnya transparansi bagi klien
Klien tidak memiliki akses untuk memantau status keluhan yang telah diajukan. Seluruh



informasi perkembangan biasanya disampaikan secara terpisah melalui WhatsApp atau email, yang kerap kali tidak terdokumentasi secara sistematis.

4. Alur eskalasi tidak terdokumentasi dengan baik Karena proses penanganan dilakukan secara informal (melalui *Telegram/WhatsApp/email*), tidak ada catatan resmi terkait siapa yang bertanggung jawab dalam setiap tahapan penyelesaian keluhan. Hal ini membuat proses tindak lanjut menjadi tidak terkontrol dan rawan terlewat.

3.1 PERANCANGAN SISTEM

Sistem dikembangkan berbasis *web* dengan menerapkan arsitektur *client-server*. Pada sisi *frontend*, sistem menggunakan HTML, CSS, dan JavaScript untuk membangun antarmuka yang responsif dan ramah pengguna.

Sedangkan pada sisi *backend*, sistem dikembangkan menggunakan *Node.js* dengan *framework Express.js*, yang memungkinkan pembuatan server dan pengelolaan rute API secara efisien. Komunikasi antara *frontend* dan *backend* dilakukan melalui REST API (Application Programming Interface), yang memungkinkan pertukaran data secara real-time dan terstruktur. Untuk mendukung keamanan proses autentikasi, digunakan JWT (JSON Web Token) sebagai metode otorisasi pengguna. Selama tahap pengembangan dan debugging, digunakan *nodemon* untuk melakukan *live-reload* secara otomatis saat ada perubahan kode backend.

Sistem ini dirancang dengan tiga tingkat hak akses pengguna, yaitu:

- Admin
Bertanggung jawab untuk mengelola data master (klien dan supervisor), menginput keluhan, serta melihat dan memantau seluruh data keluhan yang masuk ke sistem.
- Supervisor (SPV)
Memiliki peran sebagai pengawas alur kerja penanganan keluhan. SPV dapat melihat data keluhan, melakukan *monitoring* progres, serta mengunduh laporan atau dokumen keluhan yang telah ditindaklanjuti.
- Klien
Diberikan akses untuk Melihat dan mengajukan keluhan, mengunggah bukti pendukung, serta memantau status dan riwayat keluhan yang telah dikirimkan.

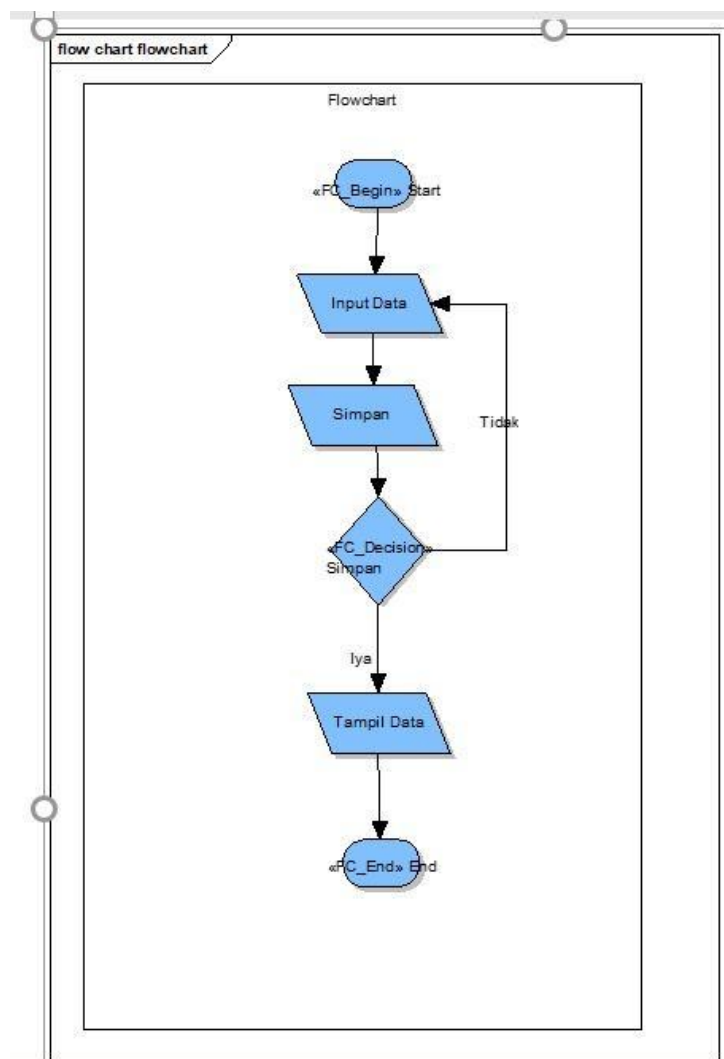
3.1.1 Sistem berjalan

Analisis dilakukan terhadap sistem manajemen keluhan transaksi di PT Gema Meraki Sukses untuk mengetahui alur proses keluhan yang berjalan secara terinci. Analisis ini menggunakan *document Flowchart* sebagai alat bantu visualisasi.

Document Flowchart dibuat untuk menggambarkan alur proses penginputan, penyimpanan, dan penampilan data keluhan dari pengguna. Tujuan utama dari alur ini adalah untuk menjelaskan proses awal hingga akhir ketika klien mengajukan keluhan, mulai dari input data, penyimpanan, hingga data tampil kembali sebagai konfirmasi.

Melalui analisis ini, dapat diketahui bahwa sistem yang berjalan masih memiliki beberapa keterbatasan, seperti belum adanya pengecekan validasi otomatis saat input data, tidak ada mekanisme eskalasi ke supervisor, serta keterbatasan transparansi terhadap status keluhan bagi klien. *Flowchart* juga menunjukkan bahwa jika proses penyimpanan gagal, pengguna harus kembali mengisi data secara manual, yang berpotensi menyebabkan pengulangan dan keterlambatan proses.

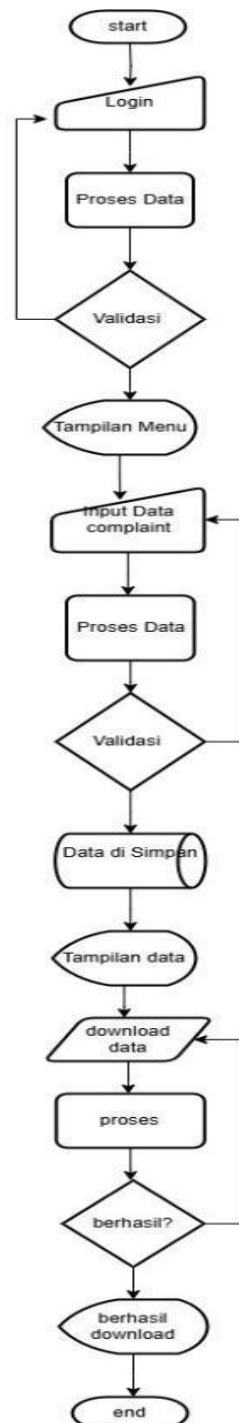
Analisis ini menjadi dasar penting dalam pengembangan sistem digital yang lebih terstruktur dan efisien untuk pengelolaan keluhan transaksi di perusahaan.



Gambar 1 Sistem berjalan



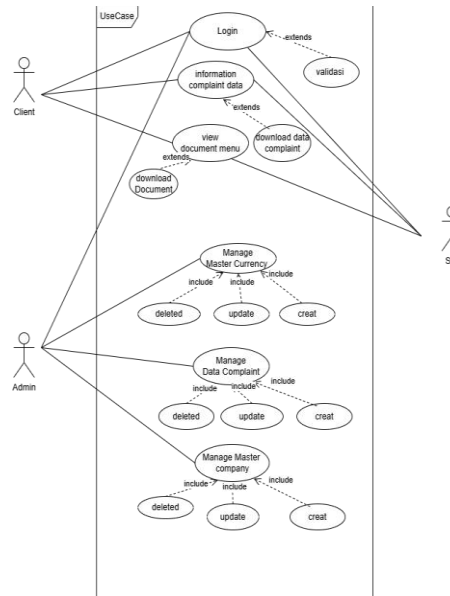
3.1.2 Sitem usulan



Gambar 2 Sistem Usulan

1. Use Case

Use Case adalah deskripsi fungsionalitas sistem yang menggambarkan interaksi antara aktor (pengguna atau sistem eksternal) dengan sistem untuk mencapai suatu tujuan tertentu. Use case menentukan serangkaian tindakan yang dilakukan oleh sistem, yang menghasilkan keluaran yang dapat diamati dan bernilai bagi seorang aktor. (Menurut UML Specification (OMG, 2023)

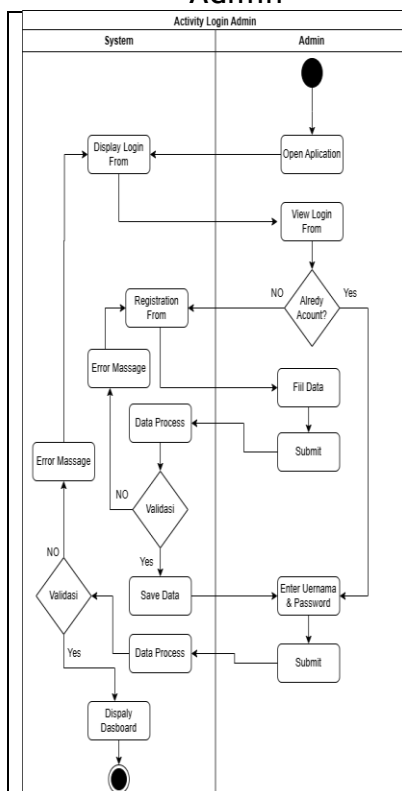


Gambar 3 Use Case

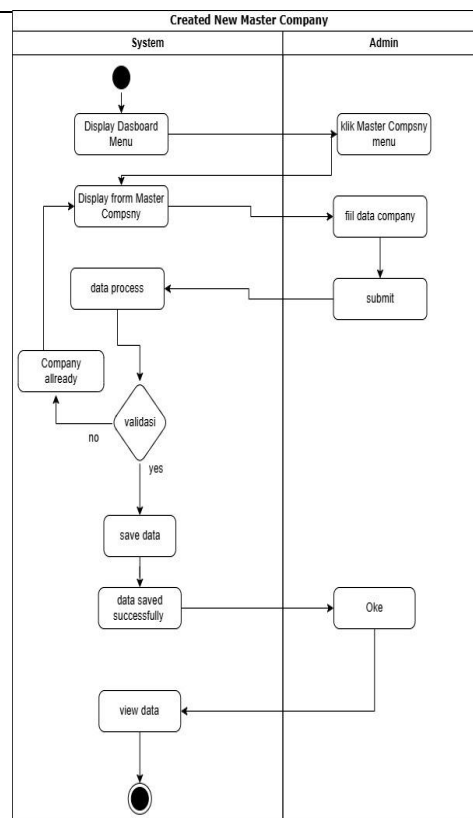
2. Activity Diagram

Activity diagram adalah diagram yang menggambarkan aliran aktivitas atau alur kerja dalam suatu sistem secara visual, mulai dari awal hingga akhir proses. Diagram ini menunjukkan urutan aktivitas, percabangan, paralelisme, dan aliran data antar aktivitas dalam sistem tersebut. *Activity diagram* merupakan bagian dari *Unified Modeling Language* (UML) yang digunakan untuk memodelkan aspek dinamis dari sistem, termasuk proses bisnis dan logika prosedural, sehingga mempermudah pemahaman proses secara keseluruhan.

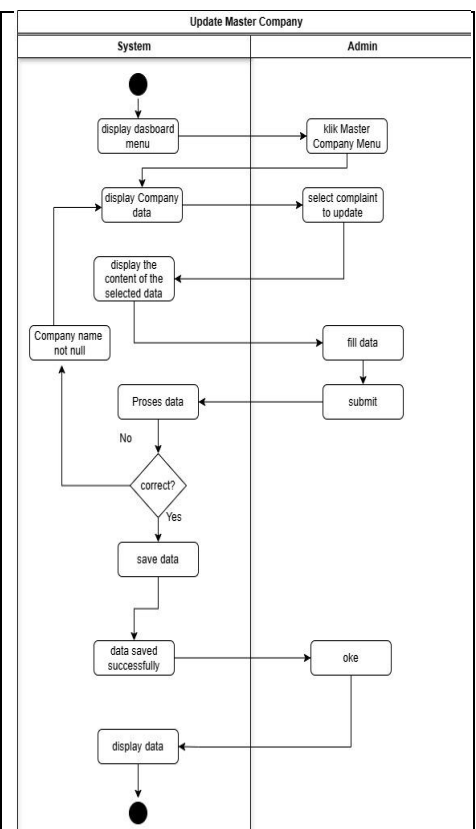
• Admin



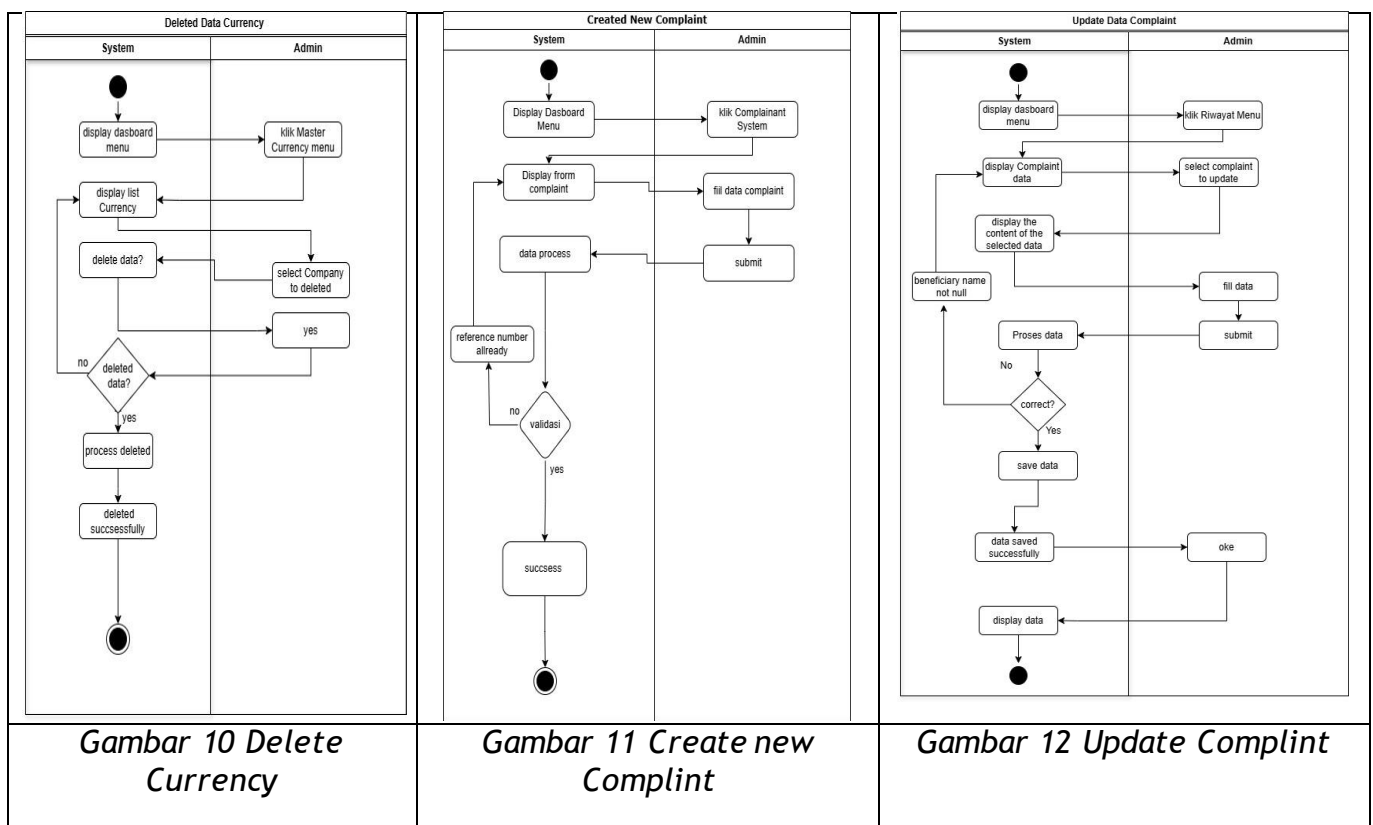
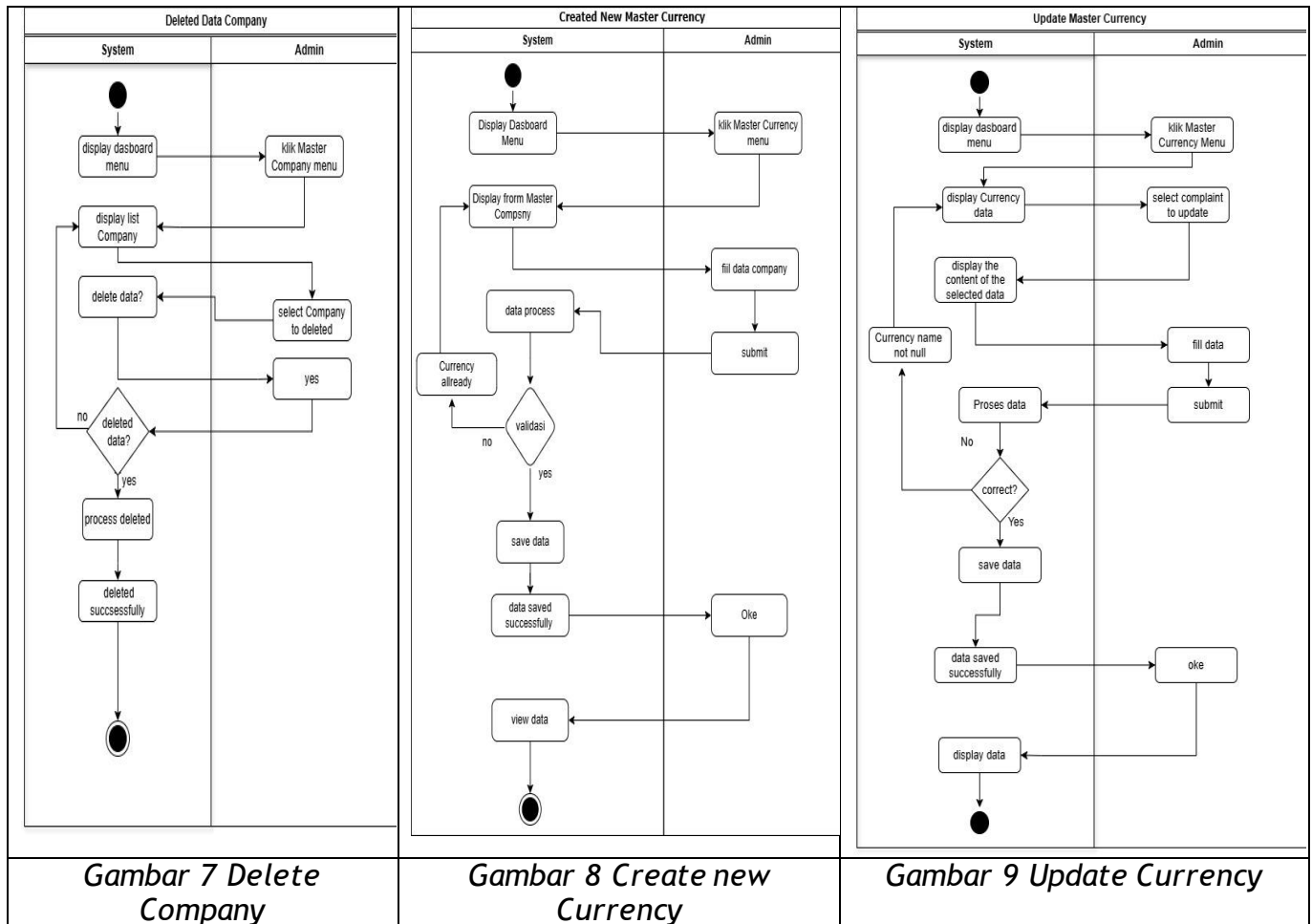
Gambar 4 Activity Login Admin

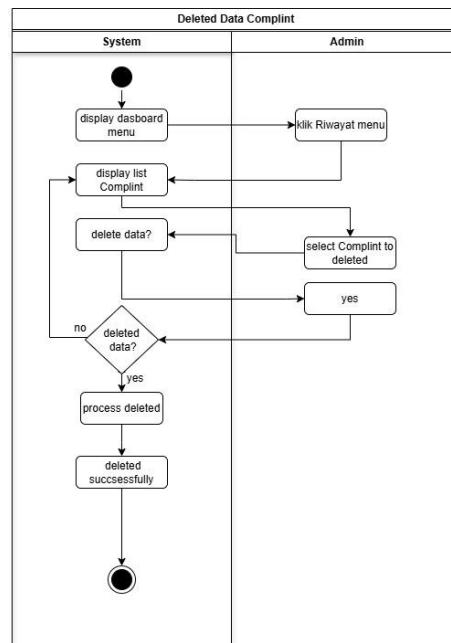


Gambar 5 Create new company



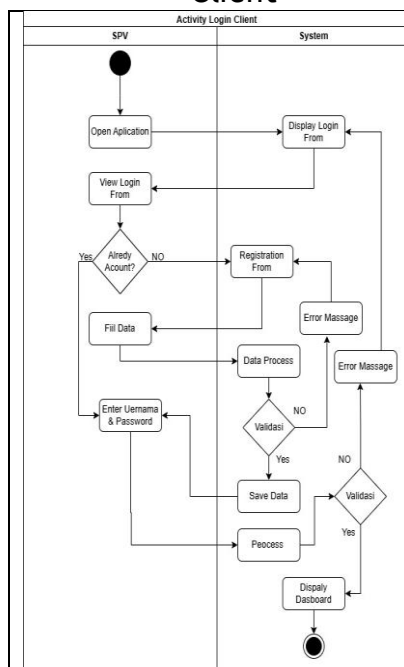
Gambar 6 Update Company



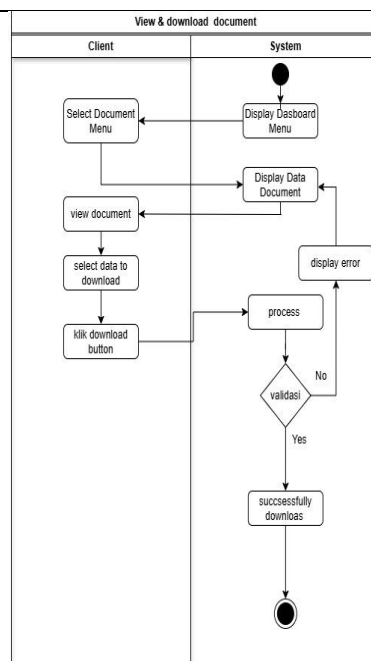


Gambar 13 Delete Complint

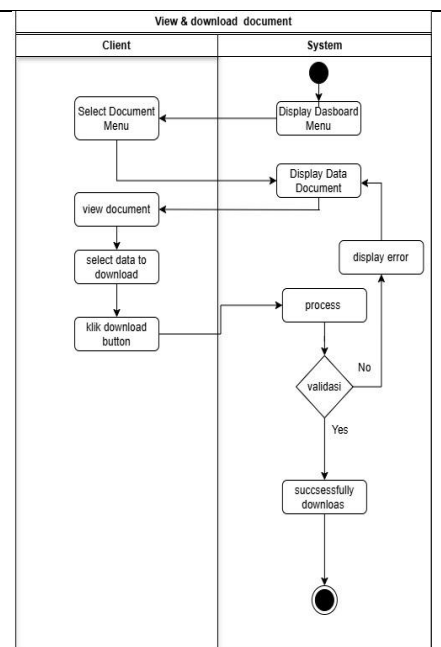
• Client



Gambar 14 Login client



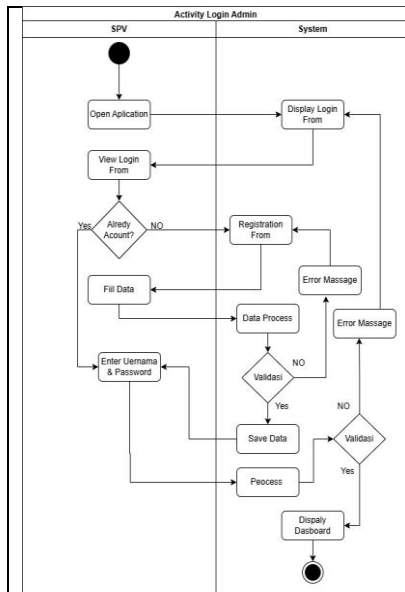
Gambar 15 View & download document



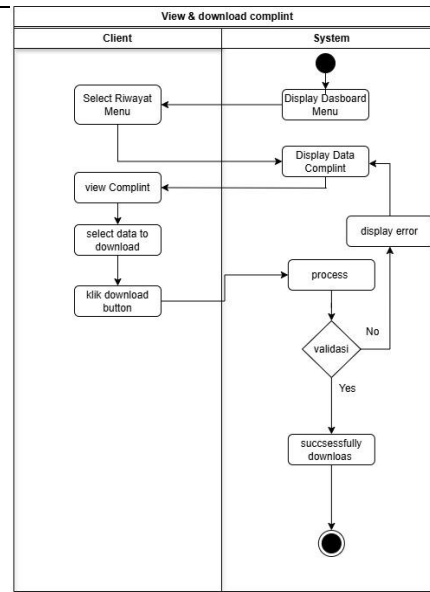
Gambar 16 View & download data complint



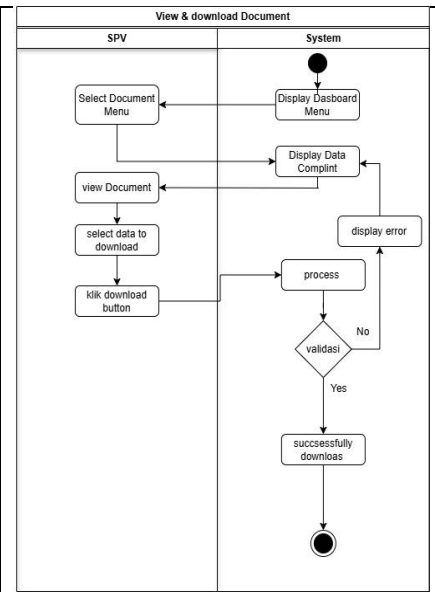
• SPV



Gambar 17 Login spv



Gambar 18 View & download Complaint



Gambar 19 View & download Document

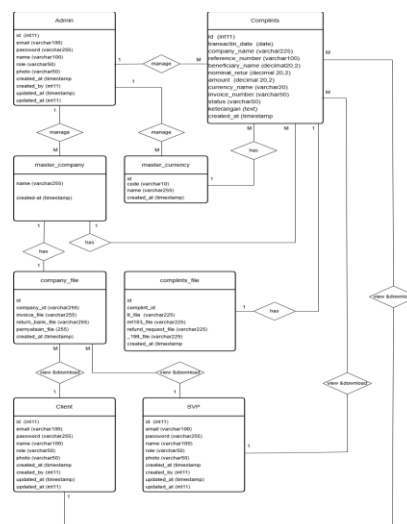
3. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah diagram yang digunakan untuk merancang dan memodelkan struktur basis data dengan menggambarkan entitas-entitas utama dalam sistem, atribut-atributnya, serta hubungan (relasi) antar entitas tersebut secara detail dan terstruktur.

ERD membantu memvisualisasikan bagaimana data disimpan dan saling terkait dalam sebuah database, sehingga memudahkan pengembang dan analis dalam memahami, merancang, dan mengelola *database* dengan lebih efisien.

ERD biasanya dibuat pada tahap perancangan konseptual *database* untuk menjadi panduan sebelum implementasi fisik database.

Singkatnya, ERD adalah alat visual yang menggambarkan struktur data dan hubungan antar data dalam sistem basis data secara sistematis dan mudah dipahami.

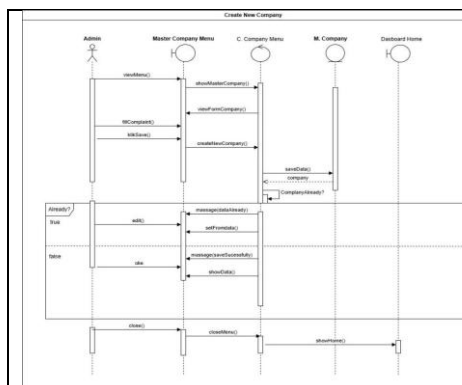


Gambar 20 ERD

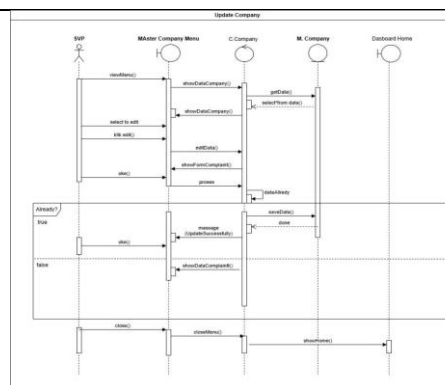
4. Sequence diagram

Sequence diagram adalah diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk memvisualisasikan interaksi antar objek atau komponen dalam suatu sistem secara berurutan berdasarkan waktu. Diagram ini menampilkan objek-objek yang terlibat secara horizontal dan urutan pesan atau perintah yang dikirim antar objek tersebut secara vertikal, menggambarkan aliran komunikasi dan waktu pelaksanaan setiap langkah dalam suatu proses atau skenario tertentu. *Sequence diagram* membantu memahami, merancang, dan mendokumentasikan perilaku dinamis sistem serta interaksi antar komponen secara jelas dan terstruktur.

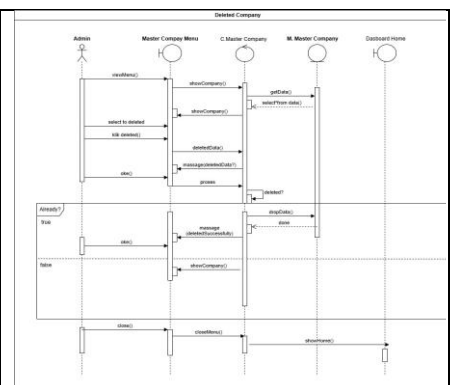
- Admin



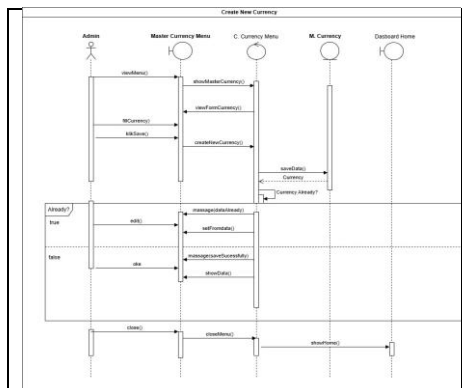
Gambar 21 Create new Company



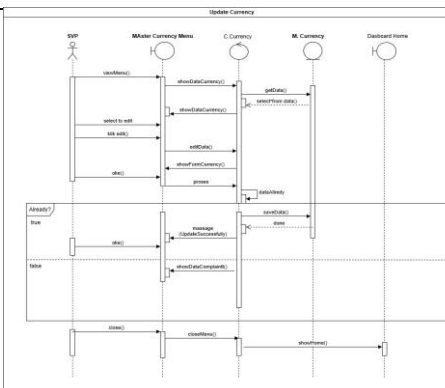
Gambar 22 Update Company



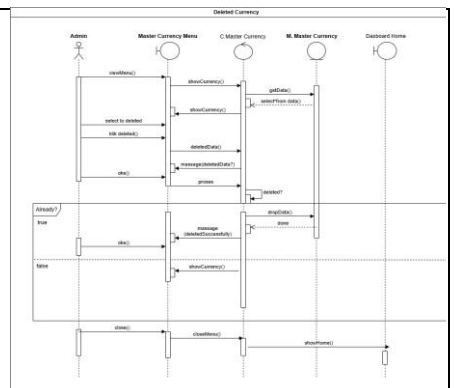
Gambar 23 Delete Company



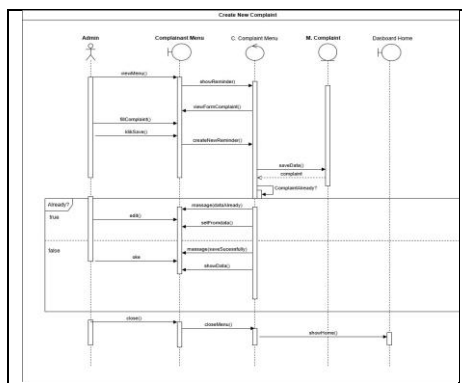
Gambar 24 Create new Currency



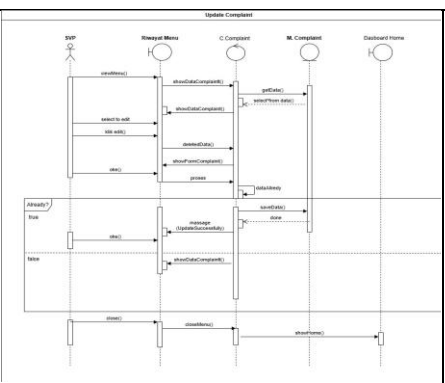
Gambar 25 Update Master Currency



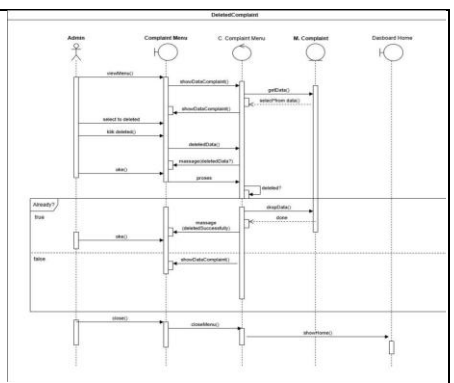
Gambar 26 Delete Currency



Gambar 27 Create new Complint

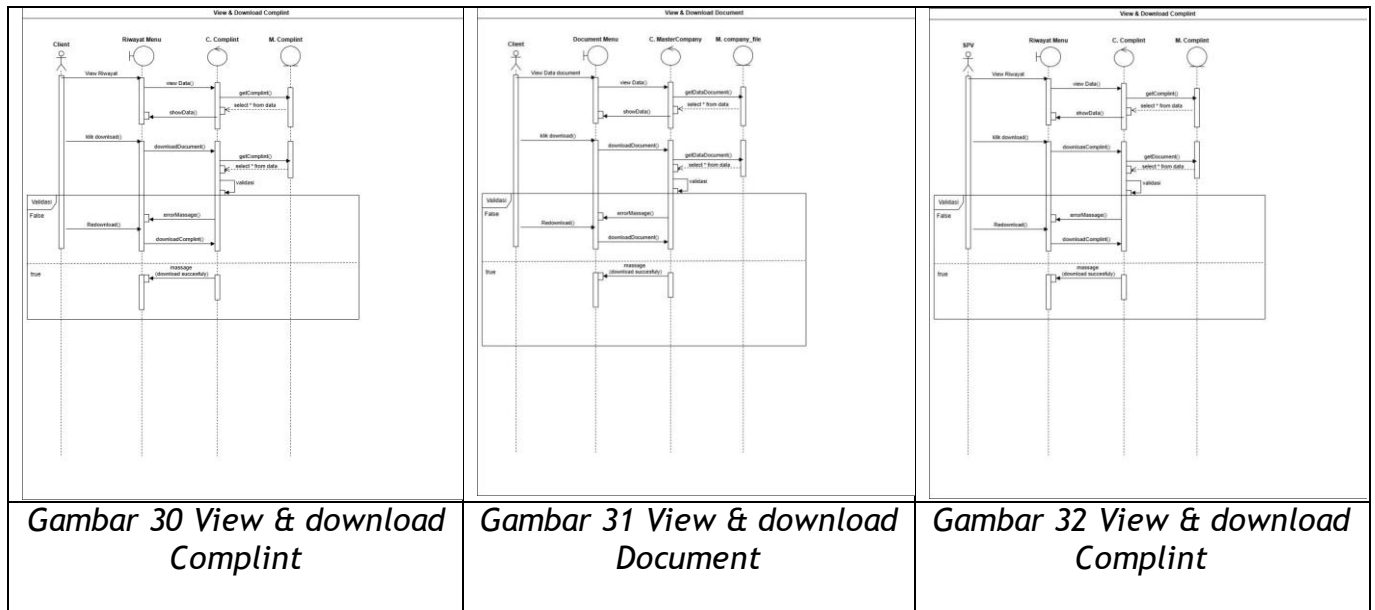


Gambar 28 Update Complint

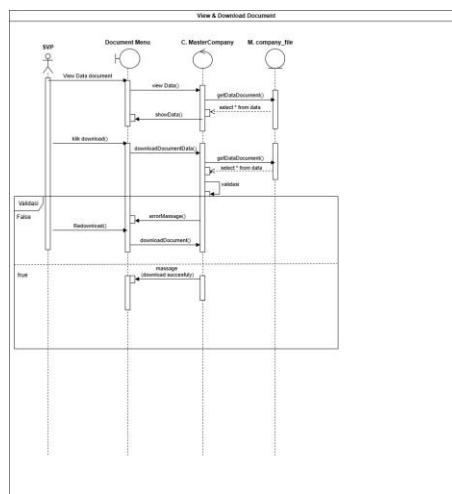


Gambar 29 Delete Complint

- Client



- SPV



Gambar 33 View & download Document

5. Normalisasi

Normalisasi adalah teknik dalam desain *database relasional* yang digunakan untuk memecah tabel besar menjadi tabel-tabel yang lebih kecil dan saling berelasi, dengan tujuan agar data menjadi lebih efisien, konsisten, dan mudah dikelola.



complainant_db complaints id : int(11) transaction_date : date company_name : varchar(255) reference_number : varchar(100) beneficiary_name : varchar(255) nominal_retur : decimal(20,2) amount : decimal(20,2) currency_name : varchar(20) invoice_number : varchar(50) tt_file : varchar(255) mt103_file : varchar(255) refund_request_file : varchar(255) _199_file : varchar(255) status : varchar(50) keterangan : text created_at : timestamp	complainant_db master_currency id : int(11) code : varchar(20) name : varchar(255) created_at : timestamp complainant_db master_company id : int(11) name : varchar(255) created_at : timestamp description : varchar(255) invoice_file : varchar(255) return_bank_file : varchar(255) pernyataan_file : varchar(255)	complainant_db users id : int(11) email : varchar(100) password : varchar(255) name : varchar(100) role : varchar(50) photo : varchar(255) created_at : timestamp created_by : int(11) updated_at : timestamp updated_by : int(11)
--	---	---

Gambar 34 Normalisasi table

Complaints id (int11) PK transaction_date (date) company_name (varchar225) reference_number (varchar100) beneficiary_name (decimal20,2) nominal_retur (decimal 20,2) amount (decimal 20,2) currency_name (varchar20) invoice_number (varchar50) tt_file (varchar225) mt103_file (varchar225) refund_request_file (varchar225) _199_file (varchar225) status (varchar50) keterangan (text) created_at (timestamp)	master_company name (varchar255) PK created_at (timestamp)	company_file id PK company_id (varchar255) FK invoice_file (varchar255) return_bank_file (varchar255) pernyataan_file (255) created_at (timestamp)
master_currency id (int11) PK code (varchar20) name (varchar255) created_at (timestamp)	users id (int11) PK email (varchar100) password (varchar255) name (varchar100) role (varchar50) photo (varchar50) created_at (timestamp) created_by (int11) updated_at (timestamp) updated_by (int11)	

Gambar 35 Table 1NF

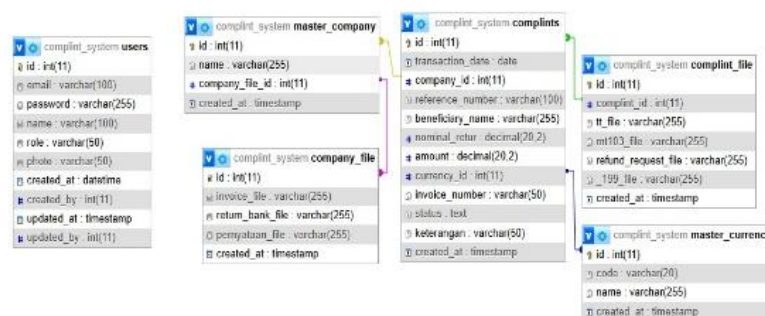
Complaints id (int11) PK transaction_date (date) company_name (varchar225) reference_number (varchar100) beneficiary_name (decimal20,2) nominal_retur (decimal 20,2) amount (decimal 20,2) currency_name (varchar20) invoice_number (varchar50) status (varchar50) keterangan (text) created_at (timestamp)	master_company name (varchar255) PK created_at (timestamp)	company_file id PK company_id (varchar255) FK invoice_file (varchar255) return_bank_file (varchar255) pernyataan_file (255) created_at (timestamp)
master_currency id (int11) PK code (varchar20) name (varchar255) created_at (timestamp)	users id (int11) PK email (varchar100) password (varchar255) name (varchar100) role (varchar50) photo (varchar50) created_at (timestamp) created_by (int11) updated_at (timestamp) updated_by (int11)	complaint_file id complaint_id tt_file (varchar225) mt103_file (varchar225) refund_request_file (varchar225) _199_file (varchar225) created_at (timestamp)

Gambar 36 Table 2NF

6. Relasi Tabel

Relasi tabel adalah hubungan antar tabel dalam basis data (database) yang menunjukkan bagaimana data di satu tabel terhubung dengan data di tabel lainnya.

Relasi ini sangat penting dalam perancangan basis data relasional, karena membantu menjaga integritas data dan memudahkan *query* (pengambilan data).



Gambar 37 Relasi tabel



7. User Interface (UI)

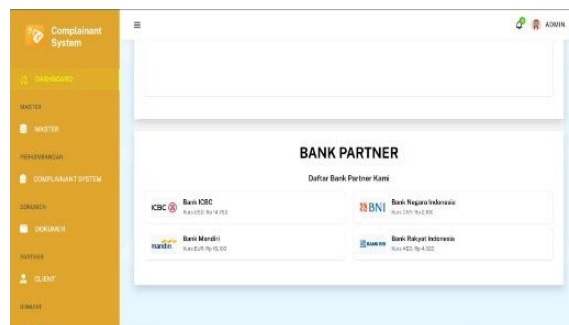
User Interface (UI) adalah tampilan visual dan elemen-elemen yang memungkinkan pengguna berinteraksi dengan sebuah sistem, aplikasi, atau perangkat digital secara langsung. UI mencakup berbagai komponen seperti tombol, menu, ikon, teks, warna, tata letak, dan grafik yang disusun sedemikian rupa agar pengguna merasa nyaman, mudah, dan intuitif saat menggunakan produk tersebut.

Secara lebih spesifik, UI berfungsi sebagai jembatan antara pengguna dan sistem, sehingga pengguna dapat memberikan input dan menerima output dengan cara yang efektif dan efisien. Desain UI yang baik akan meningkatkan pengalaman pengguna (*user experience*) dengan memastikan tampilan yang menarik, responsif, dan mudah dipahami.

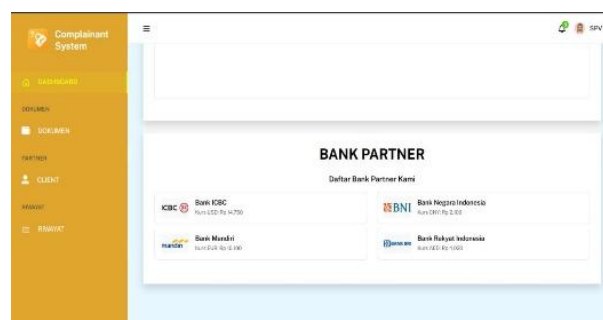
<i>Gambar 38 UI Dashboard Admin</i>	<i>Gambar 39 UI Dashboard SPV</i>	<i>Gambar 40 UI Dashboard Client</i>

8. Implementasi Program

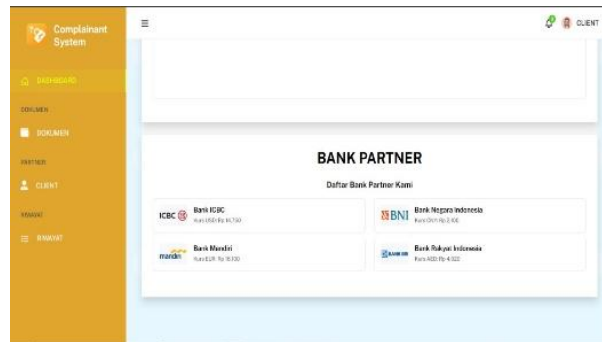
A. Tampilan *Dashboard Admin*



Gambar 41 Dashboard Admin



Gambar 42 Dashboard SPV



Gambar 42 Dashboard Client

4. KESIMPULAN

Pengembangan sistem digital untuk manajemen keluhan di PT Gema Meraki Sukses merupakan solusi strategis untuk modernisasi layanan pelanggan. Berdasarkan analisis kebutuhan yang dilakukan, sistem berbasis *web* ini dirancang untuk mengatasi kelemahan proses manual, seperti keterlambatan, risiko kehilangan data, dan kurangnya transparansi. Dengan menerapkan model pengembangan *Waterfall*, sistem berhasil dirancang dengan fitur-fitur yang terdefinisi dengan baik, mencakup pengajuan keluhan online, pelacakan status *real-time*, alur kerja investigasi untuk supervisor, dan *dashboard monitoring* untuk admin. Implementasi teknologi seperti Node.js, Express.js, dan JWT memastikan sistem ini tidak hanya fungsional tetapi juga aman dan dapat diandalkan. Dengan diterapkannya sistem ini, diharapkan PT Gema Meraki Sukses dapat meningkatkan efisiensi operasional, memperkuat akuntabilitas dalam penanganan masalah, dan pada akhirnya meningkatkan kepuasan serta kepercayaan klien.



DAFTAR PUSTAKA

- Aulia, F., & Hidayat, T. (2022). Pengembangan Sistem Informasi Manajemen Keluhan Pelanggan Berbasis Web Studi Kasus: Layanan Pelanggan PDAM. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 9(2), 243-250. <https://doi.org/10.25126/jtiik.20229255>
- Auth0. (2024). *JWT Handbook: JSON Web Token for Secure Authentication*. Diakses dari <https://auth0.com/learn/json-web-tokens>
- Budi, S. (2023). *Rekayasa Perangkat Lunak: Pendekatan Praktis untuk Pengembangan Sistem Informasi*. Yogyakarta: Deepublish.
- Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures* (Disertasi Doktorat, University of California, Irvine).
- GitHub, Inc. (2024). *Express.js Documentation*. Diakses dari <https://expressjs.com>
- Hidayatullah, M. S., & Rasyid, A. (2023). Penerapan JWT untuk Autentikasi Pengguna dalam Sistem Informasi Berbasis Web. *Jurnal Teknologi dan Sistem Informasi*, 11(1), 101-110.
- Mozilla Developer Network (MDN). (2024). *HTML, CSS, and JavaScript Guides*. Diakses dari <https://developer.mozilla.org>
- Nugroho, A. (2022). *Analisis dan Perancangan Sistem Informasi: Pendekatan Terstruktur dan Berorientasi Objek*. Yogyakarta: Andi Publisher.
- Object Management Group (OMG). (2023). *Unified Modeling Language (UML) Specification v2.5.1*. Diakses dari <https://www.omg.org/spec/UML>
- Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach* (Edisi ke-9). New York: McGraw-Hill Education.
- Raj, R. (2023). *Mastering Node.js: Build Scalable Network Applications Using JavaScript*. Birmingham: Packt Publishing.
- Sommerville, I. (2016). *Software Engineering* (Edisi ke-10). London: Pearson Education.