

# **PENGEMBANGAN SISTEM INFORMASI KOPERASI BERBASIS MOBILE DAN WEB**

## **Studi Kasus : Sistem Koperasi**



## **LAPORAN PROYEK AKHIR**

Dibuat untuk memenuhi syarat menyelesaikan matakuliah Bahasa Indonesia, Rekayasa Perangkat Lunak, Pemograman Berbasis Fremwork, dan Pemograman Mobile

Oleh:

1. AAN NAWANGSIH (23416255201270)
2. NELSEN HERLAN SAPUTRA (23416255201260)
3. FAJAR RIZQY RABBAANIISYAH (23416255201250)
4. GALIH ADIA RAMADHAN (23416255201252)
5. ABDUL HAFIDZ AZHAR (23416255201240)

**TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS BUANA PERJUANGAN KARAWANG**

**2024**

## KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan yang berjudul “Pengembangan Sistem Informasi Berbasis Mobile dan Web” ini dengan baik dan tepat waktu.

Laporan ini disusun sebagai bentuk dokumentasi dari proses perancangan dan implementasi sistem informasi koperasi berbasis mobile dan web, yang bertujuan untuk meningkatkan efisiensi pengelolaan simpanan dan pinjaman dalam lingkungan koperasi. Penulis berharap laporan ini dapat memberikan gambaran yang jelas mengenai tahapan pengembangan sistem, mulai dari analisis kebutuhan, perancangan, hingga implementasi.

Dalam penyusunan laporan ini, penulis menyadari sepenuhnya bahwa tanpa bantuan, bimbingan, dan dukungan dari berbagai pihak, laporan ini tidak akan terselesaikan dengan baik. Oleh karena itu, penulis mengucapkan terima kasih kepada semua pihak yang telah membantu, khususnya kepada:

1. Tim pengembangan sistem, yang telah berkontribusi dalam perancangan, pengujian, dan implementasi sistem Koperasi berbasis Mobile dan Web.
2. Sistem Koperasi, yang telah berkontribusi dalam memberikan masukan, data, dan kesempatan untuk mengimplementasikan sistem ini.
3. Seluruh pihak terkait, yang telah memberikan dukungan moral, material, serta ide yang sangat membantu selama proses penyusunan laporan ini.

Kami menyadari bahwa laporan ini masih memiliki kekurangan. Oleh karena itu kami sangat terbuka terhadap kritik dan saran yang membangun demi kesempurnaan laporan ini dimasa mendatang. Akhir kata, kami berharap laporan ini dapat memberikan manfaat bagi pembaca.

Karawang, tanggal bulan 2025

Penyusun

## DAFTAR ISI

## **DAFTAR GAMBAR**

## DAFTAR TABLE

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Sistem Koperasi merupakan sebuah aplikasi digital yang dirancang untuk mendukung proses pengelolaan simpan pinjam di lingkungan koperasi secara efisien dan terintegrasi. Aplikasi ini menyediakan fitur pencatatan simpanan, pengajuan dan pelunasan pinjaman, pembayaran angsuran, serta penyusunan laporan keuangan koperasi. Melalui sistem ini, anggota koperasi dapat memantau secara real-time status simpanan dan pinjaman mereka, sementara kolektor dan pengurus koperasi dapat mencatat, mengelola, dan memantau seluruh aktivitas keuangan dengan lebih mudah dan akurat.

Aplikasi dikembangkan menggunakan *Ionic* dan *Angular* sebagai antarmuka mobile untuk anggota dan kolektor, sedangkan bagian backend dibangun dengan *Laravel* atau *Django Rest Framework (DRF)*, yang dihubungkan melalui *RESTful API*. Seluruh data disimpan menggunakan sistem basis data *MySQL*. Semua komponen dirancang untuk berfungsi secara nyata dan terintegrasi penuh demi mendukung operasional koperasi secara menyeluruh.

Fungsi utama dari sistem ini adalah untuk memberikan kemudahan bagi anggota koperasi dalam mengakses informasi saldo simpanan, status pinjaman, dan riwayat transaksi mereka secara mandiri. Selain itu, sistem ini juga memfasilitasi kolektor dalam mencatat pembayaran angsuran dari anggota, baik melalui koneksi online maupun secara offline. Di sisi lain, admin koperasi memiliki kontrol penuh untuk mengelola data anggota, kolektor, transaksi, dan pinjaman, serta menyusun laporan keuangan dengan lengkap dan akurat.

### 1.2 Rumusan Masalah

1. Bagaimana cara sistem koperasi meningkatkan efisiensi pengelolaan simpan pinjam?
2. Fitur apa yang harus ada untuk memudahkan anggota dalam mengakses informasi simpanan dan pinjaman?
3. Bagaimana sistem mendukung kolektor dalam mencatat pembayaran angsuran secara efektif?
4. Apa saja kendala yang dihadapi admin dalam mengelola data anggota dan transaksi?
5. Bagaimana integrasi antara *frontend* dan *backend* dapat meningkatkan operasional koperasi?

### 1.3 Tujuan Proyek

1. Meningkatkan efisiensi dalam pengelolaan simpan pinjam melalui sistem digital yang terintegrasi, sehingga proses transaksi menjadi lebih cepat dan minim kesalahan.
2. Menyediakan akses informasi simpanan dan pinjaman bagi anggota secara real-time melalui aplikasi mobile yang *user-friendly*.
3. Memfasilitasi kolektor dalam melakukan pencatatan pembayaran angsuran dengan antarmuka yang sederhana dan mendukung input data lengkap.
4. Mengidentifikasi permasalahan yang dihadapi oleh admin koperasi dan memberikan solusi sistematis untuk meningkatkan akurasi dan kecepatan penyusunan laporan keuangan.
5. Menjamin integrasi yang optimal antara frontend dan backend melalui penggunaan *RESTful API*, guna mendukung fungsionalitas sistem secara menyeluruh.

## **1.4 Manfaat Proyek**

Pengembangan sistem aplikasi koperasi ini diharapkan dapat memberikan berbagai manfaat nyata bagi seluruh pihak yang terlibat, baik secara operasional maupun strategis, antara lain:

### **1. Bagi Anggota Koperasi:**

Mempermudah akses terhadap informasi finansial pribadi seperti saldo simpanan, status pinjaman, dan riwayat transaksi secara real-time melalui aplikasi mobile. Hal ini turut meningkatkan transparansi serta membangun kepercayaan anggota terhadap tata kelola koperasi.

### **2. Bagi Kolektor:**

Menyediakan sarana pencatatan pembayaran angsuran yang praktis, efisien, dan dapat digunakan dalam kondisi online maupun offline. Dengan demikian, proses pengumpulan data di lapangan menjadi lebih mudah, cepat, dan minim risiko kehilangan informasi.

### **3. Bagi Pengurus atau Admin Koperasi:**

Memfasilitasi pengelolaan data anggota, pencatatan transaksi, serta penyusunan laporan keuangan secara lebih terstruktur dan akurat. Sistem ini juga membantu meminimalisasi kesalahan akibat pencatatan manual dan meningkatkan kecepatan dalam proses administrasi.

### **4. Bagi Sistem Operasional Koperasi secara Keseluruhan:**

Meningkatkan efisiensi dan keterpaduan antar bagian dalam organisasi koperasi melalui integrasi sistem berbasis RESTful API. Integrasi ini memungkinkan pertukaran data yang cepat, aman, dan konsisten antar komponen sistem, sehingga mendukung pengambilan keputusan yang lebih baik.

### **5. Bagi Pengembangan Teknologi Digital di Lingkungan Koperasi:**

Menjadi model implementasi teknologi informasi yang dapat dijadikan rujukan bagi koperasi lain dalam rangka digitalisasi proses bisnis. Proyek ini turut mendorong transformasi digital di sektor koperasi secara lebih luas dan berkelanjutan.

## BAB II

### ANALISIS SISTEM

#### 2.1 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk mengidentifikasi fungsi-fungsi utama yang harus dimiliki oleh sistem, serta karakteristik teknis dan non-teknis yang mendukung performa dan kualitas sistem. Kebutuhan sistem dibagi menjadi dua kategori, yaitu kebutuhan fungsional dan non-fungsional.

#### 2.2 Fungsional Ssitem

Kebutuhan fungsional menjelaskan layanan-layanan utama yang harus disediakan oleh sistem agar dapat memenuhi tujuan penggunaannya. Adapun kebutuhan fungsional dari sistem koperasi ini meliputi:

1. Sistem harus dapat mencatat data anggota koperasi secara lengkap.
2. Sistem memungkinkan anggota untuk melihat saldo simpanan, status pinjaman, dan riwayat transaksi secara *real-time*.
3. Kolektor dapat mencatat pembayaran angsuran dari anggota, baik secara online maupun offline.
4. Admin dapat mengelola data kolektor, transaksi simpanan, pinjaman, serta angsuran anggota.
5. Sistem menyediakan fitur laporan keuangan koperasi berdasarkan transaksi yang tercatat.
6. Sistem mendukung proses login dan autentikasi pengguna berdasarkan peran (admin, kolektor, anggota).
7. Sistem memungkinkan integrasi antara *frontend mobile* dan *backend API* melalui *RESTful API*.

#### 2.3 Non-Fungsional Sistem

Kebutuhan non-fungsional menggambarkan kualitas sistem dan batasan teknis yang harus dipenuhi untuk memastikan sistem berjalan dengan optimal. Adapun kebutuhan non-fungsional dari sistem ini antara lain:

1. Ketersediaan (*Availability*): Sistem harus dapat diakses selama 24 jam dengan downtime seminimal mungkin.
2. Responsif dan Mudah Digunakan: Antarmuka aplikasi mobile harus dirancang agar intuitif dan ramah pengguna, baik bagi anggota maupun kolektor.
3. Keamanan Data: Sistem harus menyediakan mekanisme keamanan untuk menjaga kerahasiaan data pengguna, termasuk enkripsi data dan otorisasi akses.
4. Kinerja (*Performance*): Sistem harus mampu menangani transaksi dan permintaan data dalam waktu respons yang cepat, bahkan saat digunakan secara bersamaan oleh banyak pengguna.

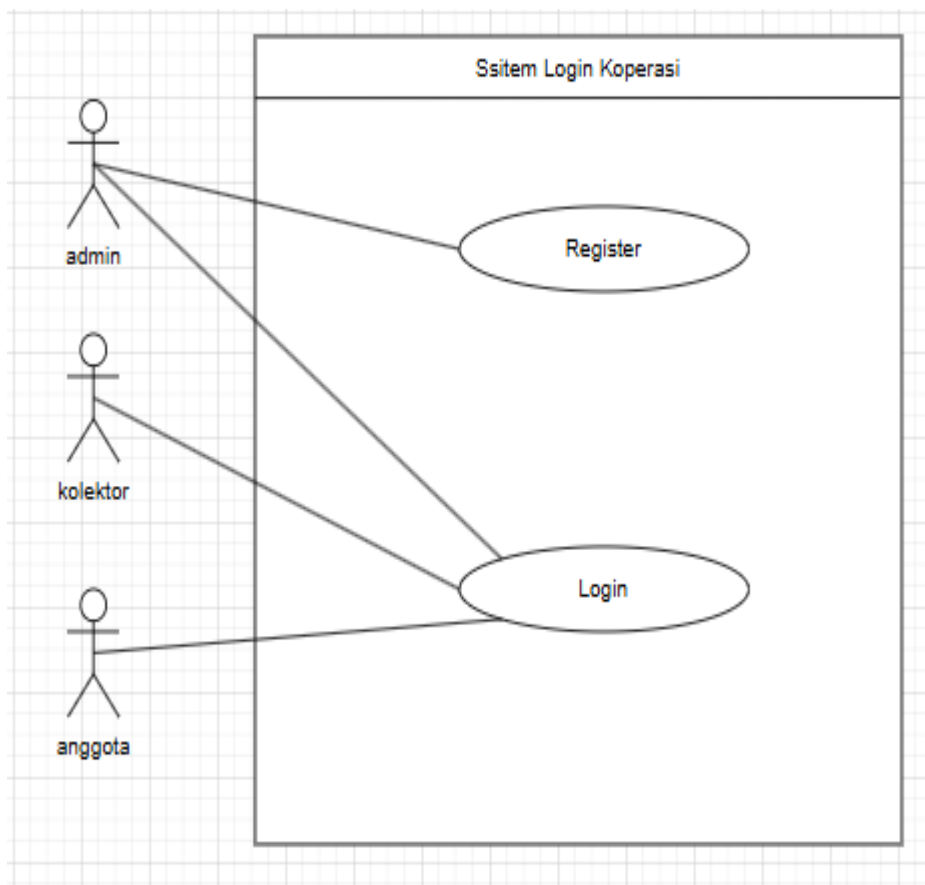


5. Portabilitas: Aplikasi *mobile* harus dapat berjalan pada berbagai perangkat *Android* dengan spesifikasi minimum yang ditentukan.
6. Skalabilitas: Sistem harus dirancang untuk memungkinkan pengembangan lebih lanjut, baik dari sisi penambahan fitur maupun jumlah pengguna

## 2.4 Diagram Use Case

*Usecase* diagram adalah diagram yang digunakan untuk menggambarkan serangkaian Tindakan (*usecase*) yang beberapa sistem atau sistem (subjek) dapat lakukan dalam kolaborasi dengan satu atau lebih pengguna eksternal sistem (aktor). *Usecase* bertujuan untuk mengumpulkan kebutuhan dari sebuah sistem, mendapatkan pandangan dari luar sistem, mengidentifikasi faktor yang mempengaruhi sistem, dan menunjukkan interaksi antara aktor dengan sistem. Diagram *use case* terdiri dari tiga elemen utama, yaitu:

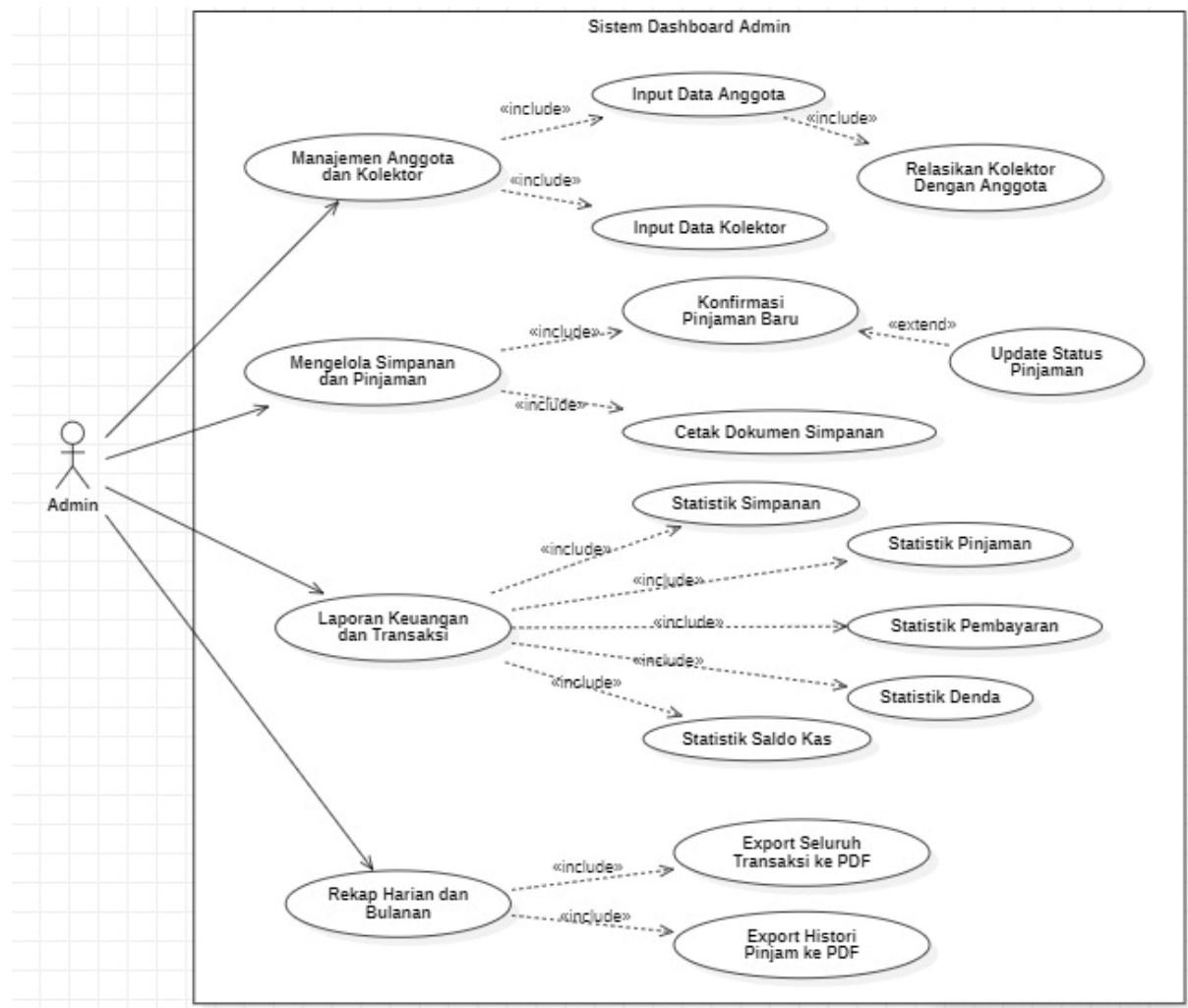
1. Aktor mewakili orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat.
2. *Use case* berisi perilaku apa saja yang dapat dilakukan oleh aktor.
3. Relasi menunjukkan bagaimana interaksi antara aktor dan *use case*.



Use case diagram di atas menggambarkan sistem login pada koperasi yang melibatkan tiga aktor utama, yaitu Admin, Kolektor, dan Anggota. Dalam sistem ini, Admin memiliki peran penting karena dapat melakukan proses *Register* untuk mendaftarkan pengguna baru, baik itu kolektor maupun anggota. Admin juga memiliki hak untuk melakukan *Login* ke dalam sistem. Sementara itu, Kolektor dan Anggota tidak memiliki akses untuk melakukan pendaftaran secara mandiri. Mereka hanya dapat melakukan *Login* ke dalam sistem setelah sebelumnya didaftarkan oleh Admin. Dengan demikian, use case *Register* hanya terhubung dengan aktor Admin, sedangkan use case *Login* terhubung dengan ketiga aktor, yaitu Admin, Kolektor, dan Anggota. Diagram ini menunjukkan dengan jelas pembagian wewenang dan alur interaksi masing-masing aktor dalam sistem login koperasi, yang bertujuan untuk menjaga keamanan dan ketertiban proses akses pengguna ke dalam sistem.

Komponen	Deskripsi
<b>Nama Sistem</b>	Sistem Login Koperasi
<b>Aktor</b>	- Admin: Dapat melakukan Register dan Login. - Kolektor: Hanya dapat Login. - Anggota: Hanya dapat Login.
<b>Use Case</b>	- Register: Digunakan oleh Admin untuk mendaftarkan pengguna baru ke sistem. - Login: Digunakan oleh semua aktor (Admin, Kolektor, Anggota) untuk masuk ke dalam sistem.
<b>Hak Akses</b>	- Admin memiliki akses penuh (Register dan Login). - Kolektor hanya dapat login ke sistem. - Anggota hanya dapat login setelah didaftarkan oleh Admin.
<b>Fungsi Register</b>	Memungkinkan Admin untuk memasukkan data pengguna baru dan menentukan peran serta hak akses dalam sistem.
<b>Fungsi Login</b>	Memberikan akses masuk ke sistem sesuai dengan peran masing-masing pengguna. Setelah login, pengguna akan diarahkan ke menu atau dashboard sesuai levelnya.

### 2.4.1 Use Case Admin

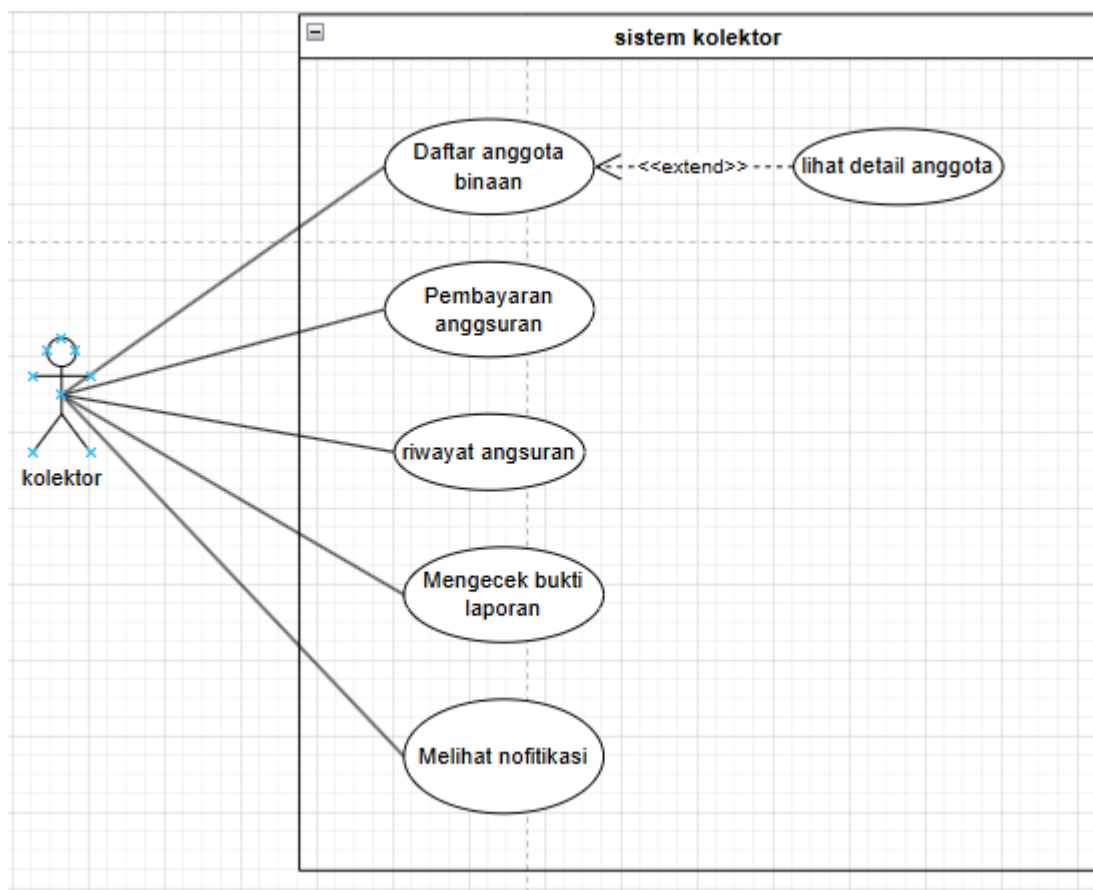


Use case diagram yang menggambarkan fitur-fitur utama yang dapat diakses oleh aktor Admin dalam Sistem Dashboard Admin. Admin memiliki empat aktivitas utama, yaitu: Manajemen Anggota dan Kolektor, Mengelola Simpanan dan Pinjaman, Laporan Keuangan dan Transaksi, serta Rekap Harian dan Bulanan.

1. Dalam Manajemen Anggota dan Kolektor, admin dapat melakukan input data anggota, input data kolektor, dan melakukan relasi antara kolektor dengan anggota.
2. Pada bagian Mengelola Simpanan dan Pinjaman, admin bisa melakukan konfirmasi pinjaman baru, mencetak dokumen simpanan, serta melakukan update status pinjaman (ditandai dengan relasi «extend»).
3. Selanjutnya, pada menu Laporan Keuangan dan Transaksi, admin dapat melihat statistik keuangan, seperti statistik simpanan, pinjaman, pembayaran, denda, dan saldo kas (dihubungkan dengan relasi «include»).
4. Terakhir, pada bagian Rekap Harian dan Bulanan, admin dapat mengekspor seluruh transaksi dan histori pinjaman dalam format PDF untuk kebutuhan pelaporan.

No	Use Case Utama	Use Case Turunan	Relasi
1	Manajemen Anggota dan Kolektor	Input Data Anggota, Input Data Kolektor, Relasikan Kolektor dengan Anggota	«include»
2	Mengelola Simpanan dan Pinjaman	Konfirmasi Pinjaman Baru, Cetak Dokumen Simpanan, Update Status Pinjaman	«include», «extend»
3	Laporan Keuangan dan Transaksi	Statistik Simpanan, Statistik Pinjaman, Statistik Pembayaran, Statistik Denda, Statistik Saldo Kas	«include»
4	Rekap Harian dan Bulanan	Export Seluruh Transaksi ke PDF, Export Histori Pinjam ke PDF	«include»

### 2.4.2 Use Case Kolektor



Gambar yang ditampilkan adalah sebuah diagram use case dari sistem yang diberi nama "sistem kolektor". Diagram ini menjelaskan interaksi antara aktor dan use case dalam sistem. Aktor yang berperan di dalam sistem ini adalah kolektor, yang digambarkan dengan ikon manusia di sebelah kiri. Kolektor dapat melakukan beberapa aktivitas (use case) yang berkaitan dengan pekerjaannya dalam sistem. Aktivitas-aktivitas tersebut meliputi:

1. Daftar anggota binaan – Kolektor dapat melihat daftar anggota binaan yang menjadi tanggung jawabnya.

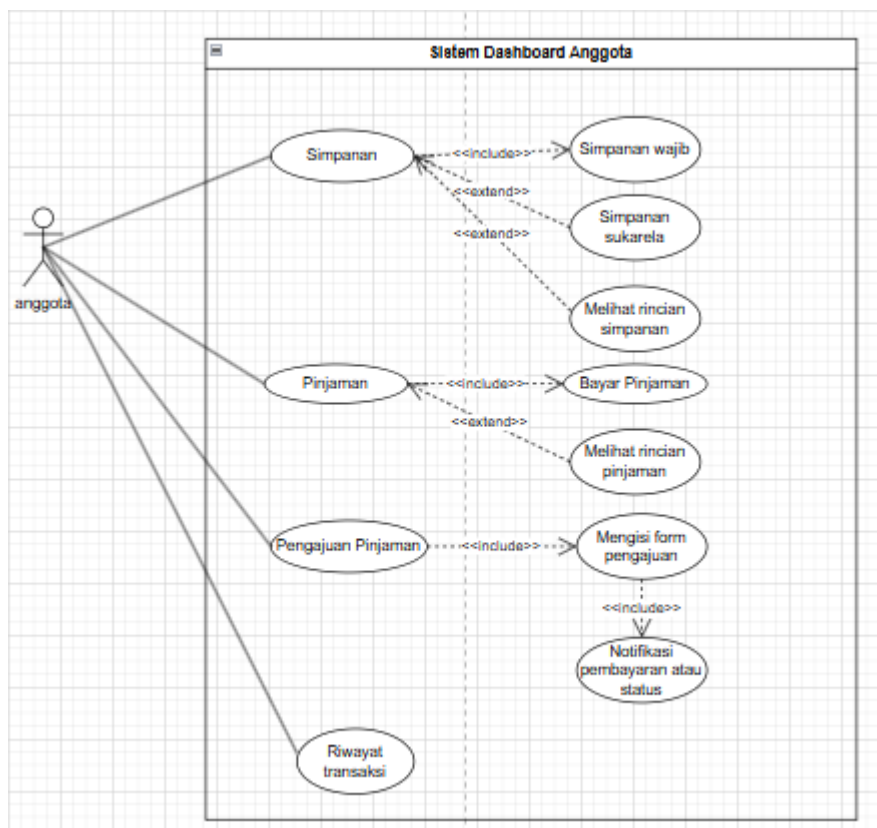
2. Pembayaran angsuran Kolektor dapat melakukan pencatatan atau pemrosesan pembayaran angsuran dari anggota.
3. Riwayat angsuran Kolektor dapat mengakses dan melihat riwayat pembayaran angsuran dari anggota binaan.
4. Mengecek bukti laporan Kolektor dapat memeriksa atau mengakses laporan terkait kegiatan atau transaksi.
5. Melihat notifikasi Kolektor juga dapat menerima dan melihat notifikasi dari sistem, kemungkinan berkaitan dengan informasi penting atau pengingat tugas.

Selain itu, terdapat relasi antara use case "Daftar anggota binaan" dengan "lihat detail anggota". Ini berarti bahwa use case *lihat detail anggota* merupakan ekstensi opsional yang dapat terjadi selama proses menampilkan daftar anggota binaan — jika kolektor ingin melihat lebih dalam informasi mengenai anggota tertentu.

Diagram ini secara keseluruhan menggambarkan fungsi-fungsi utama yang dapat dilakukan oleh kolektor dalam sistem dan bagaimana sistem mendukung aktivitas mereka dalam mengelola informasi anggota dan pembayaran.

No	Nama Use Case	Aktor	Deskripsi Singkat	Relasi
1	Daftar anggota binaan	Kolektor	Menampilkan daftar anggota binaan yang berada di bawah pengawasan kolektor.	Extend ke "Lihat detail anggota"
2	Lihat detail anggota	Kolektor	Melihat informasi detail dari anggota yang dipilih dalam daftar anggota binaan.	dari "Daftar anggota binaan"
3	Pembayaran angsuran	Kolektor	Melakukan pencatatan atau proses pembayaran angsuran dari anggota.	-
4	Riwayat angsuran	Kolektor	Melihat catatan atau riwayat pembayaran angsuran anggota.	-
5	Mengecek bukti laporan	Kolektor	Memverifikasi atau melihat bukti laporan kegiatan atau transaksi.	-
6	Melihat notifikasi	Kolektor	Menampilkan notifikasi atau informasi penting dari sistem kepada kolektor.	-

### 2.4.3 Use Case Anggota



Use case diagram yang menggambarkan fungsionalitas yang dapat diakses oleh aktor *anggota* dalam sistem dashboard anggota koperasi. Terdapat lima use case utama, yaitu **Simpanan, Pinjaman, Pengajuan Pinjaman, Riwayat Transaksi**, dan beberapa use case tambahan yang di-include di dalamnya. Pada menu *Simpanan*, anggota dapat mengakses *Simpanan Wajib*, *Simpanan Sukarela*, dan melihat rincian simpanan. Pada menu *Pinjaman*, anggota bisa melihat rincian pinjaman serta melakukan pembayaran pinjaman. Fitur *Pengajuan Pinjaman* memungkinkan anggota mengisi form pengajuan dan menerima notifikasi status atau pembayaran. Terakhir, anggota juga dapat melihat *Riwayat Transaksi* yang pernah dilakukan. Diagram ini membantu menggambarkan hubungan antara aktor dengan fitur-fitur utama dalam sistem secara ringkas dan terstruktur.

No	Use Case Utama	Use Case Turunan (<>)	Keterangan
1	Simpanan	- Simpanan Wajib- Simpanan Sukarela- Melihat rincian simpanan	Fitur untuk melihat dan mengelola jenis simpanan anggota
2	Pinjaman	- Bayar Pinjaman- Melihat rincian pinjaman	Fitur pengelolaan pinjaman oleh anggota

3	Pengajuan Pinjaman	- Mengisi form pengajuan- Notifikasi (pembayaran atau status)	Proses pengajuan pinjaman baru dan notifikasinya
4	Riwayat Transaksi	-	Melihat riwayat semua transaksi anggota

## 2.5 Activity Diagram

*Activity* diagram adalah diagram yang digunakan untuk menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. *Activity* diagram memiliki struktur yang mirip dengan flowchart. AD bermanfaat untuk membantu memahami proses secara keseluruhan. AD dibuat berdasarkan sebuah atau beberapa *use case* pada *use case* diagram.

Elemen utama dari *activity* diagram adalah:

1. Aktivitas atau tindakan yang dilakukan dalam sistem.
2. Status atau kondisi sistem pada suatu titik waktu tertentu.
3. Keputusan atau titik di mana sistem harus membuat pilihan.
4. Kondisi atau pernyataan yang dapat bernilai benar atau salah.
5. Aliran yang menunjukkan urutan proses aktivitas dalam sistem.

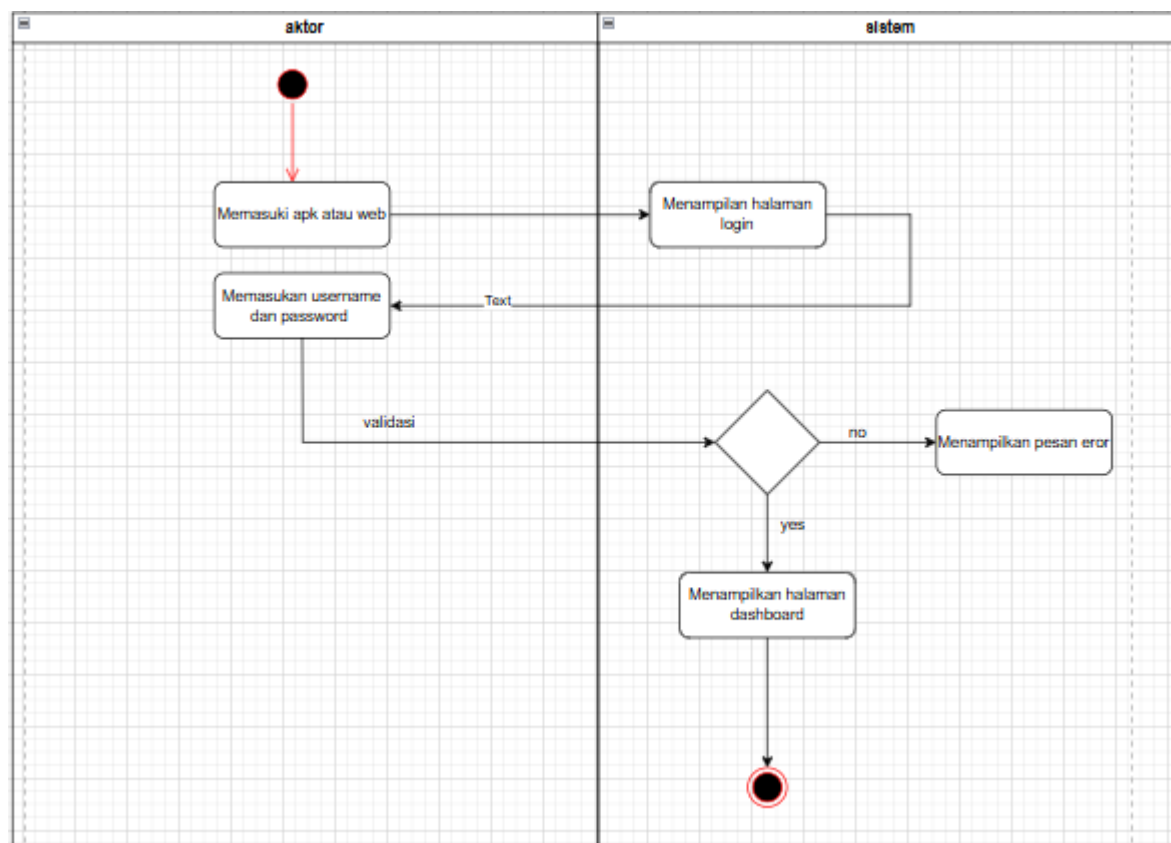
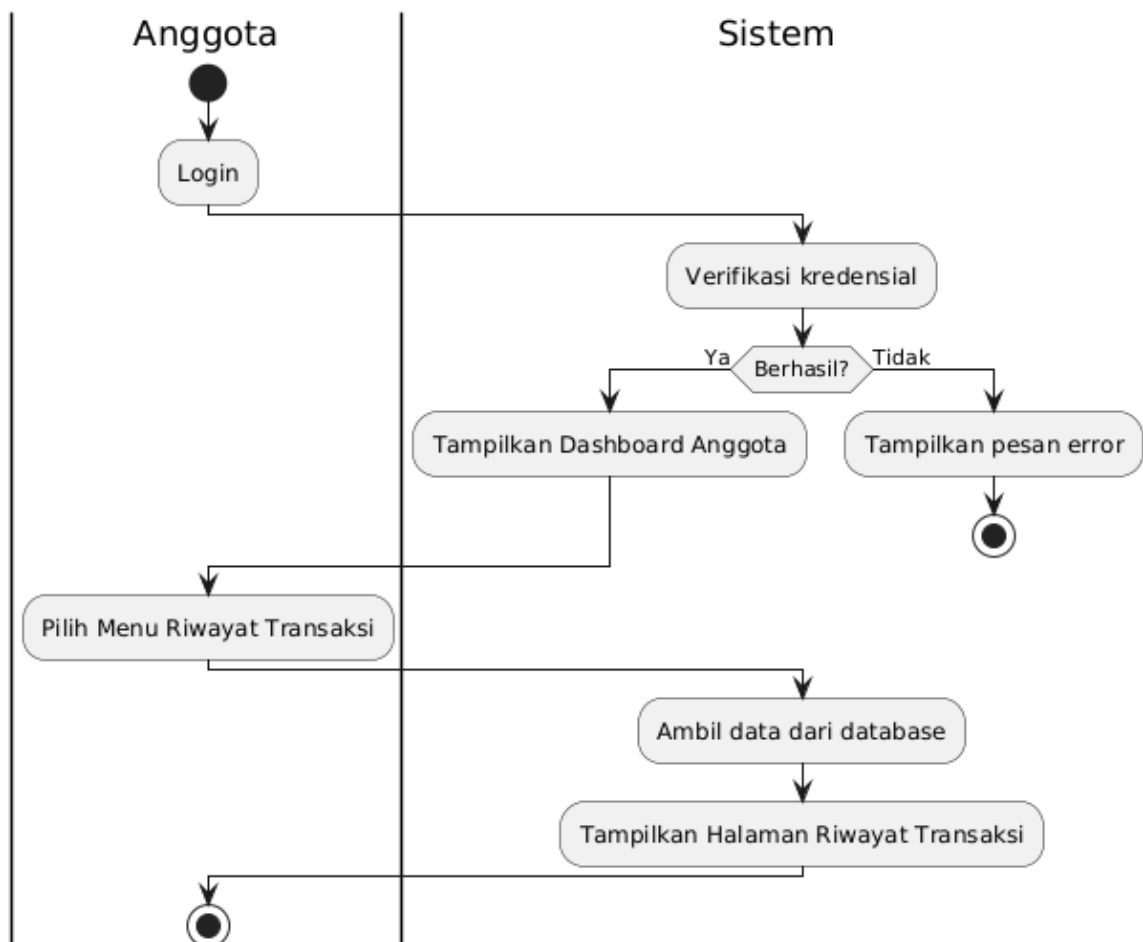


Diagram aktivitas yang ditampilkan merupakan alur proses autentikasi pengguna dalam sebuah sistem aplikasi atau situs web. Diagram ini dibagi menjadi dua swimlane utama, yaitu *Aktor* (pengguna) dan *Sistem*, yang menggambarkan interaksi antara pengguna dan sistem selama proses login.

Proses diawali ketika aktor atau pengguna membuka aplikasi atau situs web. Setelah itu, sistem akan menampilkan halaman login. Pengguna kemudian memasukkan *username* dan *password*, lalu sistem melakukan proses validasi terhadap data yang dimasukkan. Hasil dari proses validasi ini menentukan jalur berikutnya: apabila data yang dimasukkan benar (*Yes*), sistem akan menampilkan halaman dashboard sebagai bukti bahwa login berhasil. Sebaliknya, jika data tidak valid (*No*), sistem akan menampilkan pesan kesalahan (error) dan kembali meminta pengguna untuk menginput ulang data login.

Diagram ini memberikan representasi visual yang jelas mengenai alur login standar, dan sangat bermanfaat dalam proses analisis dan perancangan sistem informasi, khususnya dalam menjamin keamanan serta kemudahan akses pengguna terhadap sistem.

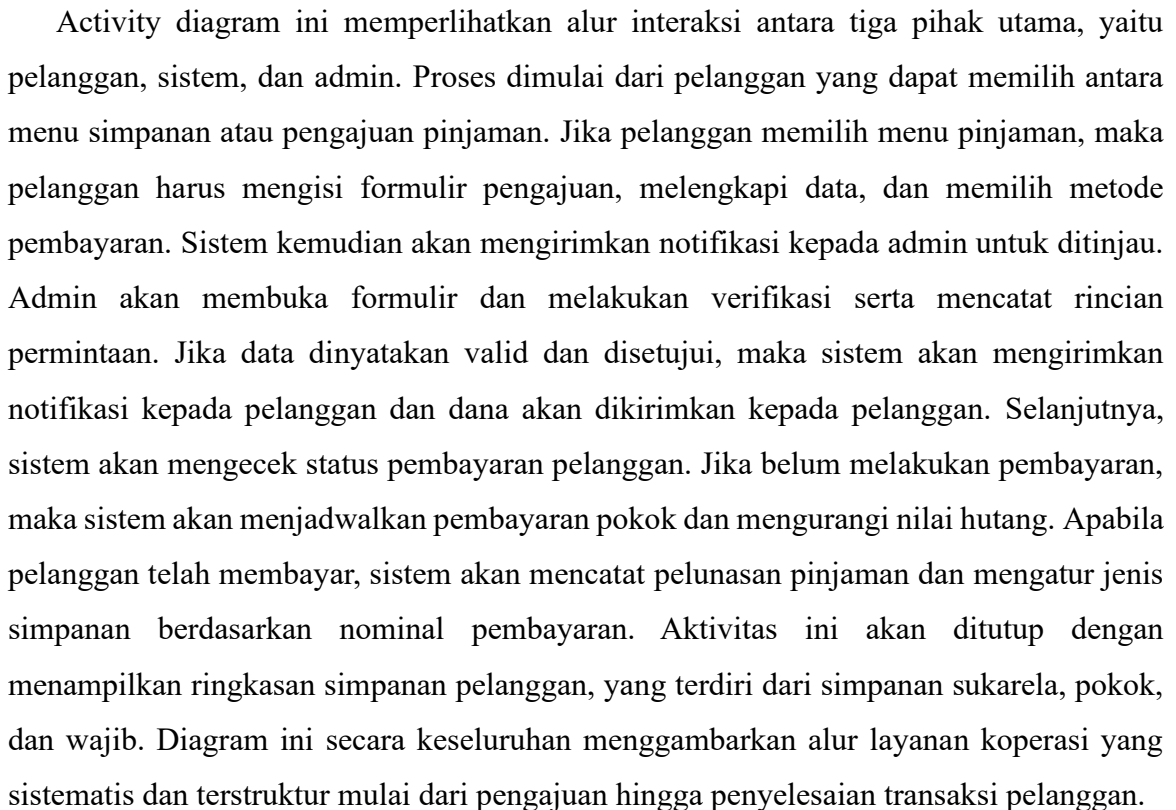


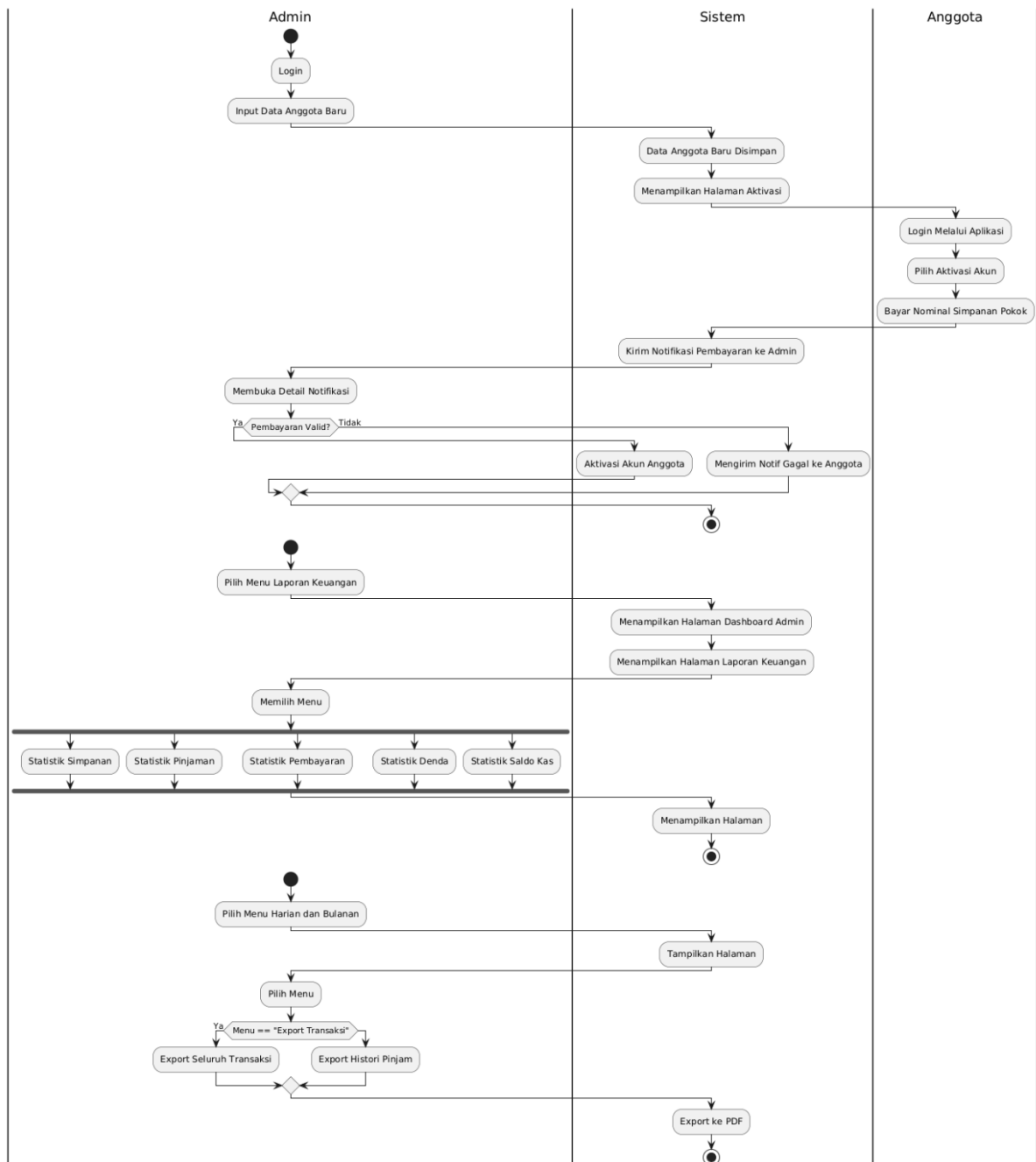


Activity diagram ini menggambarkan alur aktivitas yang dilakukan oleh seorang Anggota dalam sistem, dimulai dari proses login. Pertama-tama, pengguna (Anggota) melakukan login dengan memasukkan kredensial mereka, seperti username dan password. Selanjutnya, sistem melakukan proses *verifikasi kredensial* untuk memastikan bahwa data yang dimasukkan valid. Jika *kredensial* benar, sistem akan menampilkan Dashboard Anggota, yang merupakan halaman utama berisi menu-menu yang bisa diakses oleh anggota. Namun, jika login gagal, sistem akan menampilkan pesan error dan proses dihentikan.

Setelah berhasil login dan melihat dashboard, Anggota kemudian memilih menu Riwayat Transaksi untuk melihat catatan transaksi sebelumnya. Pilihan ini dikirimkan ke sistem, dan sistem merespons dengan melakukan pengambilan data riwayat transaksi dari database. Setelah data berhasil diambil, sistem akan menampilkan halaman Riwayat Transaksi kepada anggota. Proses ini kemudian berakhir setelah halaman berhasil ditampilkan dan tidak ada aktivitas lanjutan lainnya.

Diagram ini juga menggunakan swimlane untuk membedakan antara aktivitas yang dilakukan oleh Anggota dan aktivitas yang dilakukan oleh Sistem. Hal ini membantu memperjelas tanggung jawab masing-masing pihak dalam proses yang digambarkan. Dengan penambahan verifikasi login, kondisi berhasil/gagal, dan proses pengambilan data oleh sistem, diagram ini menjadi lebih lengkap, realistis, dan siap digunakan sebagai dokumentasi dalam proyek atau laporan.





Pada diagram aktivitas ini, terdapat tiga peran utama yaitu Admin, Sistem, dan Anggota. Proses dimulai saat Admin melakukan login dan menginput data anggota baru. Data tersebut kemudian disimpan oleh sistem dan ditampilkan dalam bentuk halaman aktivasi.

Anggota dapat login melalui aplikasi, memilih menu aktivasi akun, dan melakukan pembayaran simpanan pokok. Sistem kemudian akan mengirim notifikasi pembayaran ke Admin. Admin akan membuka detail notifikasi dan memverifikasi pembayaran. Jika pembayaran valid, maka sistem akan mengaktifkan akun anggota. Jika tidak valid, sistem akan mengirim notifikasi kegagalan aktivasi kepada anggota.

Admin dapat melanjutkan ke proses berikutnya, yaitu mengakses menu laporan keuangan. Sistem akan menampilkan dashboard dan halaman laporan keuangan. Di sini, admin dapat memilih berbagai menu laporan keuangan seperti statistik simpanan, pinjaman, pembayaran, denda, dan saldo kas. Pada bagian terakhir, admin memilih menu laporan harian dan bulanan. Sistem akan menampilkan halaman sesuai menu tersebut. Admin bisa memilih apakah ingin mengekspor seluruh transaksi atau hanya histori pinjam, dan sistem akan mengeluarkan hasil dalam bentuk file PDF.

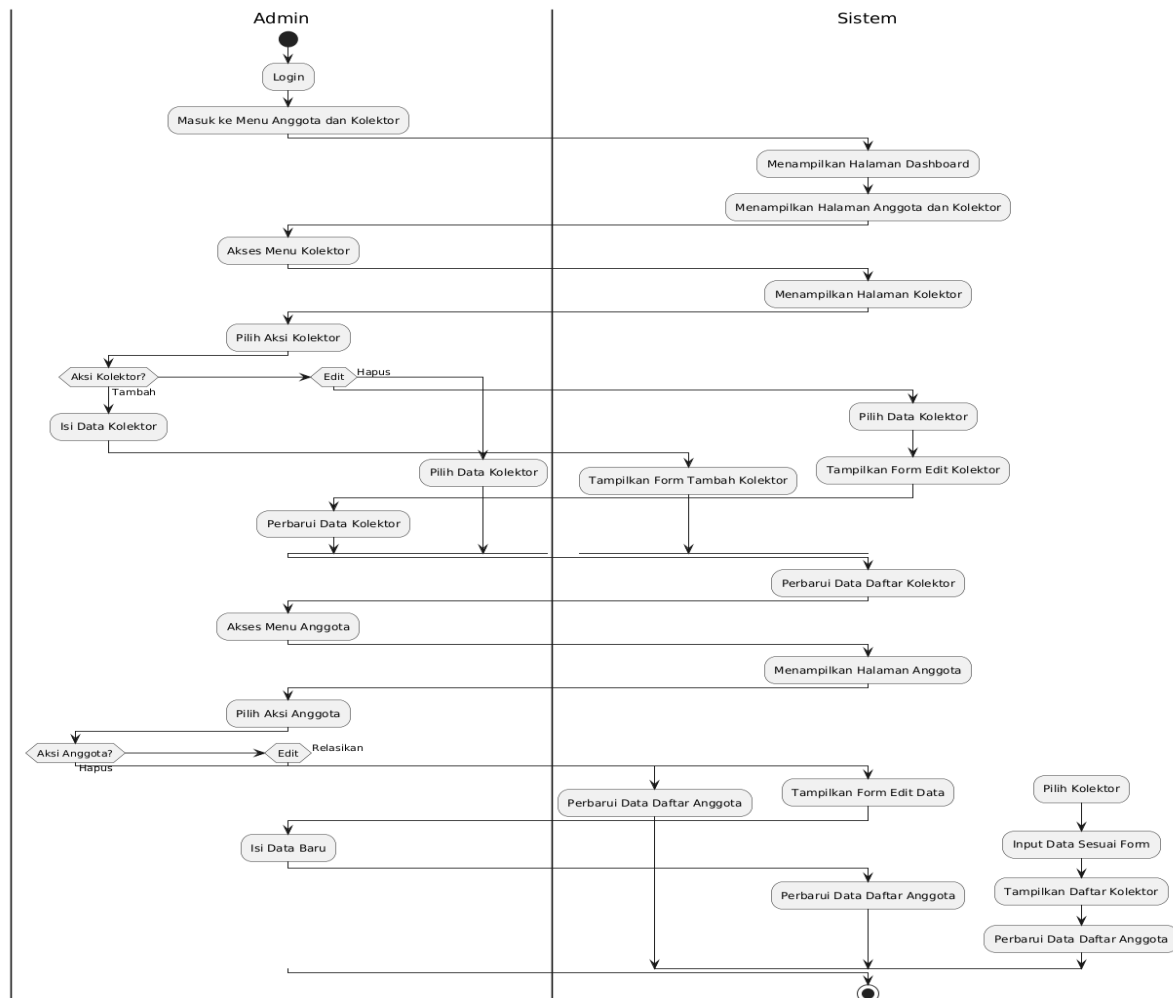
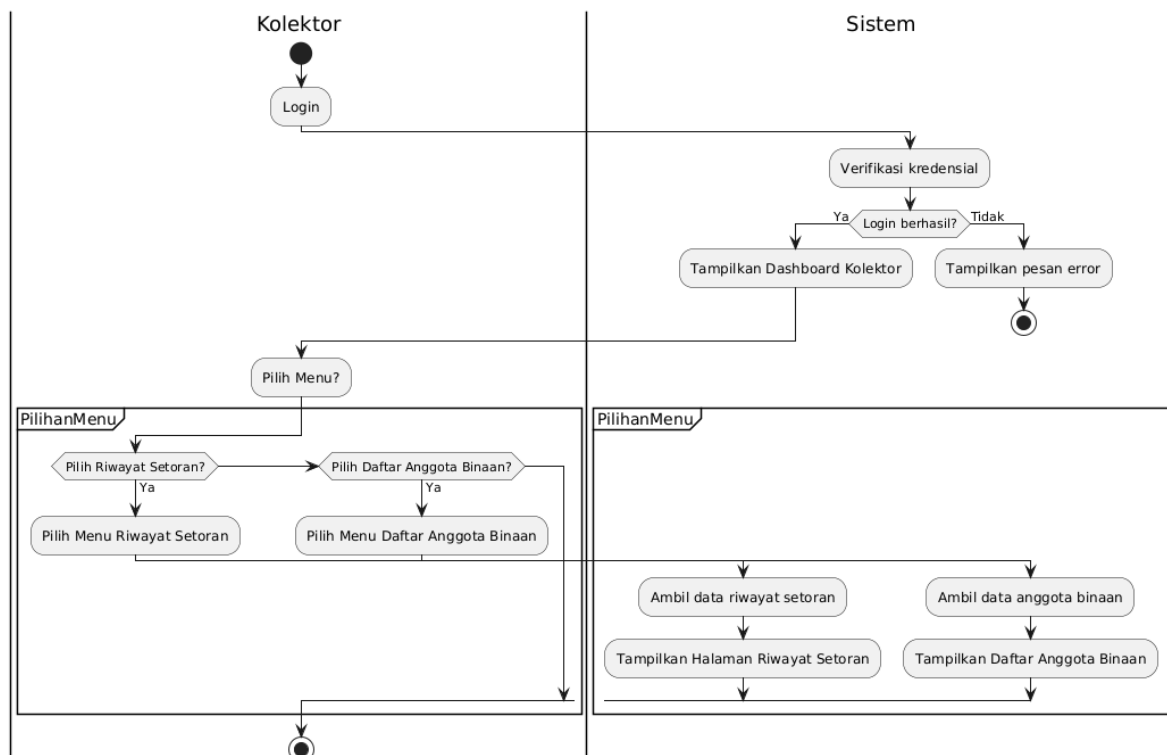


Diagram aktivitas ini menggambarkan proses kerja Admin dalam mengelola data kolektor dan anggota melalui sistem. Proses dimulai saat Admin *login* dan sistem menampilkan dashboard. Admin memilih menu "Anggota dan Kolektor", lalu sistem menampilkan halaman terkait. Ketika Admin memilih menu Kolektor, sistem menampilkan halaman kolektor, di mana Admin bisa menambah, mengedit, atau menghapus data kolektor. Untuk penambahan dan pengeditan, sistem menampilkan *form*, dan setelah data diisi, sistem memperbarui daftar kolektor. Jika memilih hapus, sistem langsung menghapus data tanpa *form*.

Untuk pengelolaan anggota, setelah memilih menu Anggota, Admin dapat menghapus, mengedit, atau merealisasikan anggota ke kolektor. Jika menghapus, sistem langsung memperbarui data. Jika mengedit, sistem menampilkan form, dan setelah perubahan, data diperbarui otomatis. Untuk merealisasikan, sistem menampilkan daftar kolektor, Admin memilih kolektor, mengisi form, dan sistem memperbarui data anggota. Secara keseluruhan, diagram ini menunjukkan proses pengelolaan data anggota dan kolektor yang efisien dan terstruktur, mendukung kebutuhan administrasi internal maupun relasi lapangan.



Activity diagram untuk peran Kolektor menggambarkan dua proses utama: melihat Riwayat Setoran dan melihat Daftar Anggota Binaan. Setiap diagram diawali dengan aktivitas login. Setelah Kolektor memasukkan kredensial, sistem akan melakukan verifikasi. Jika verifikasi berhasil, maka sistem menampilkan Dashboard Kolektor. Namun, jika gagal, sistem akan memberikan pesan kesalahan dan proses dihentikan. Setelah login berhasil, Kolektor dapat memilih menu Riwayat Setoran, dan sistem akan mengambil data dari basis data untuk kemudian menampilkan halaman riwayat tersebut. Pada diagram lainnya, Kolektor dapat memilih menu Daftar Anggota Binaan, yang kemudian akan diproses oleh sistem dengan mengambil data anggota binaan dan menampilkannya di halaman yang sesuai.

Penambahan proses verifikasi dan pengambilan data membuat diagram menjadi lebih realistis dan menggambarkan alur kerja sistem secara utuh. Swimlane digunakan untuk memisahkan peran antara Kolektor dan Sistem, sehingga alur tanggung jawab terlihat jelas.

## **2.6 ERD**

ERD atau *Entity-Relationship Diagram* adalah alat yang digunakan untuk menggambarkan entitas, atribut dan hubungan antara entitas. ERD dapat membantu memvisualisasikan struktur basis data dengan merinci.

## **2.7 Desain Antar Muka**

## **BAB III**

### **IMPLEMENTASI SISTEM**

#### **3.1 Antarmuka Pengguna (UI)**

##### **3.1.1 Admin**

##### **3.1.2 Kolektor**

##### **3.1.3 Anggota**

#### **3.2**

##### **3.2.1**

##### **3.2.2**

##### **3.2.3**

#### **3.3**

#### **3.4**

## **BAB IV**

### **PENGUJIAN SISTEM**

#### **4.1 Metode Pengujian**

#### **4.2 Hasil Pengujian**

#### **4.3 Evaluasi Sistem**

##### **4.3.1 Keuntungan**

##### **4.3.2 Kekurangan**



## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

#### **5.2 Saran**

## **DAFTAR PUSTAKA**

## **LAMPIRAN**