# BAB IV

# ANALISIS DAN PERANCANGAN SISTEM

## 4.1. Analisis Sistem Yang Sedang Berjalan

Penguraian dari suatu sistem yang berjalan dengan maksud untuk merumuskan dan mengevaluasi permasalahan-permasalahan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan suatu perbaikan.

Tahap analisis ini sangant penting, karena apabila terjadi kesalahan, akan mengakibatkan kesalahan pada sistem berikutnya. Untuk itu diperlukan ketelitian dan kecermatan yang tinggi untuk mendapatkan kualitas kerja sistem yang baik.

### 4.1.1. Analisis Prosedur yang sedang Berjalan

Sebelum membuat suatu sistem, hendaknya melakukan analisis terlebih dahulu terhadap kebutuhan-kebutuhan apa saja yang diperlukan dengan menggunakan metode-metode yang telah ada. Dalam hal ini penulis menggunakan metodologi berorientasi objek, yaitu *use case* diagram dan a*ctivity* diagram.

**A. Prosedur pendaftaran**

Prosedur ini merupakan prosedur pendaftaran anggota baru di Koperasi Pedagang Mitra Abadi Pangalengan :

1. Calon anggota mengajukan permohonan anggota baru dan persyaratan kepada petugas koperasi, kemudian petugas koperasi akan memeriksa kelengkapan persyaratan.

2. Apabila semua persyaratan lengkap maka ketua akan memverifikasi pengajuan permohonan anggota baru. Apabila tidak maka formulir akan diserahkan kembali kepada calon anggota.
3. Petugas koperasi akan membuatkan buku anggota, lalu diserahkan kepada anggota sebagai bukti telah menjadi anggota baru.

   (syarat menjadi anggota adalah pedagang pasar pangalengan)

**B. Prosedur Simpanan**

Simpanan Anggota terdiri dari :

A. Simpanan Pokok Rp. 25.000 (dibayar satu kali disaat mendaftar)
B. Simpanan wajib Rp. 10.000,-/ minggu
C. Simpanan sukarela merupakan simpanan pribadi anggota yang bersifat tidak wajib bagi anggota.

Untuk pengambilan simpanan pokok dan wajib hanya bisa diambil ketika anggota keluar dari keanggotaan koperasi, dan simpanan sukarela bisa diambil kapan saja.

Prosedur ini merupakan prosedur simpanan di Koperasi Pedagang Mitra Abadi Pangalengan :

1. Anggota menyerahkan setoran simpanan kepada petugas koperasi
2. Petugas koperasi menerima setoran kemudian memperbaharui buku anggota.
3. Setelah buku anggota diperbaharui maka buku tersebut akan diserahkan kembali kepada anggota.

4. Petugas koperasi akan membuat laporan simpanan dan diserahkan kepada ketua.

**C. Prosedur Pengambilan Simpanan**

Prosedur ini merupakan prosedur pengambilan simpanan di Koperasi Pedagang Mitra Abadi Pangalengan :

1. Anggota menyerahkan formulir pengambilan simpanan kepada petugas koperasi.
2. Petugas koperasi akan jumlah simpanan.
3. Petugas membuat laporan pengambilan simpanan.

**D. Prosedur Peminjaman**

Terdapat ketentuan dalam peminjaman yaitu peminjam adalah pedagang di pasar pangalengan, lama keanggotan selama 3 bulan, dan jumlah pinjaman maksimal dua kali lebih besar dari jumlah simpanan, jumlah bunga atau uang jasa sebesar 0,5%/minggu dan 2%/bulan, dan apabila anggota masih mempunyai pinjaman maka anggota tidak bisa meminjam. Adapun perhitungan angsuran yaitu:

Angsuran pokok : (pokok pinjaman/jangka waktu) jangka waktu disesuaikan dengan kesepakatan terlebih dahulu.

Bunga perminggu/bulan : pokok pinjaman awal x bunga perminggu/bulan

Angsuran perminggu/bulan : angsuran pokok + bunga.

Prosedur ini merupakan prosedur peminjaman di Koperasi Pedagang Mitra Abadi Pangalengan adalah sebagai berikut:

1. Anggota menyerahkan formulir pengajuan pinjaman kepada petugas koperasi.
2. Petugas koperasi akan mengecek persyaratan pinjaman.
3. Petugas akan membuat perjanjian pinjaman yang akan di verifikasi oleh ketua dan anggota.
4. Petugas koperasi akan mempersiapkan dana pinjaman dan diserahkan kepada anggota.
5. Petugas koperasi membuat laporan pinjaman

**E. Prosedur Angsuran**

Prosedur ini merupakan prosedur Angsuran di Koperasi Pedagang Mitra Abadi Pangalengan :

1. Anggota akan menyerahkan angsuran kepada petugas koperasi.
2. Petugas koperasi akan mengecek apakah ada denda angsuran atau tidak.
3. Petugas membuat laporan angsuran.

**Adapun perhitungan denda yaitu:**

Denda sebesar 0,5% / Minggu dari sisa angsuran, denda mingguan berlaku untuk angsuran perminggu atau perbulan.

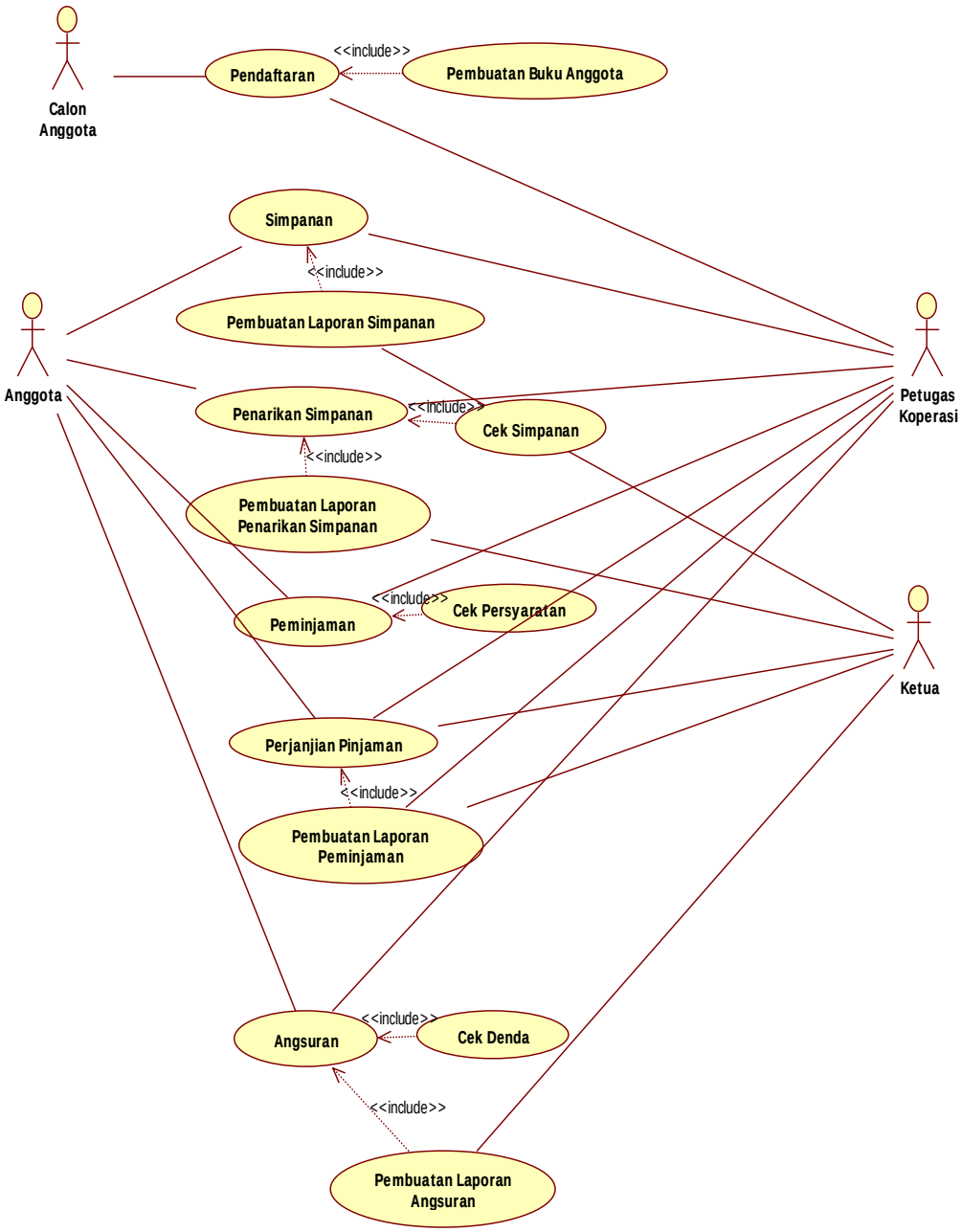
Denda : ( 0,5% x sisa angsuran)

Jumlah denda : Denda x jumlah minggu telat bayar

#### 4.1.1.1. *Use Case* Diagram

*Use case* diagram menggambarkan hubungan antara *use case* dan aktor, aktor merepresentasikan seorang *user* atau subsistem lain yang akan berinteraksi dengan sistem. Sedangkan *use case* merupakan urutan kejadian yang

menggambarkan interaksi antara *user* dengan sistem. Fungsionalitas sistem didefinisikan ke dalam *use case* dari sudut eksternal sistem yang berguna untuk uji kelayakan sistem.




Gambar 4.1 *Use Case* Diagram Simpan Pinjam pada Koperasi pedagang Mitra Abadi yang Sedang Berjalan.

#### 4.1.1.2. Skeneario *Use Case*

Skenario *use case* digunakan untuk memudahkan dalam menganalisa skenario yang akan kita gunakan pada fase-fase selanjutnya dengan melakukan penilaian terhadap skenario tersebut. Adapun tahapan-tahapan sekenario *use case* Simpan Pinjam pada koperasi pedagang Mitra Abadi yang sedang berjalan adalah sebagai berikut :

1. Nama *Use Case* : Pendaftaran

   Aktor : Calon Anggota, Petugas koperasi

   Tujuan : Calon anggota datang untuk mendaftar di koperasi

**Tabel 4.1. Skenario *Use case* Pendaftaran anggota baru**

| Aktor | Sistem |
|---|---|
| 1. Datang kekantor untuk menyerahkan formulir dan persyaratan. | |
| | 2. Menerima formulir pendaftaran. |
| | 3. Memeriksa kelengkapan persyaratan pendaftaran. |
| | 4. Menyetujui pendaftaran. |

2. Nama *Use Case* : Pembuatan Buku Anggota

   Aktor : Petugas, Anggota

   Tujuan : Melakukan pembuatan buku anggota

**Tabel 4.2. Skenario *Use case* Pembuatan buku anggota**

| Aktor | Sistem |
|---|---|
| | 1. Menyiapkan Buku Anggota. |
| 2. Menerima Buku Anggota. | |

3. Nama *Use Case* : Simpanan

   Aktor : Anggota, Petugas Koperasi

   Tujuan : Melakukan penyetoran simpanan

**Tabel 4.3. Skenario *Use case* Simpanan**

| Aktor | Sistem |
|---|---|
| 1. Menyerahkan setoran simpanan. | |
| | 2. Menerima setoran simpanan. |
| | 3. Mencatat data simpanan. |

4. Nama *Use Case* : Pembuatan Laporan Simpanan

   Aktor : Petugas Koperasi, Ketua

   Tujuan : Melakukan pembuatan laporan simpanan

**Tabel 4.4. Skenario *Use case* pembuatan laporan simpanan**

| Aktor | Sistem |
|---|---|
| | 1. Membuat laporan simpanan. |
| 2. Menerima laporan simpanan. | |

5. Nama *Use Case* : Penarikan Simpanan

   Aktor : Anggota, Petugas koperasi

   Tujuan : Melakukan penarikan simpanan

**Tabel 4.5. Skenario *Use case* penarikan simpanan**

| Aktor | Sistem |
|---|---|
| 1. Mengisi form penarikan simpanan. | |
| 2. Menyerahkan form penarikan simpanan. | |
| | 3. Mengecek jumlah simpanan. |
| | 4. Mencatat data penarikan simpanan. |

6. Nama *Use Case* : Pembuatan Laporan Penarikan Simpanan

   Aktor : Petugas Koperasi, Ketua

   Tujuan : Melakukan penarikan simpanan

**Tabel 4.6. Skenario *Use case* pembuatan laporan penarikan simpanan**

| Aktor | Sistem |
|---|---|
| | 1. Membuat laporan penarikan simpanan. |
| 2. Menerima Laporan penarikan simpanan. | |

7. Nama *Use Case* : Pinjaman

   Aktor : Anggota, Petugas koperasi

   Tujuan : Melakukan pinjaman

**Tabel 4.7. Skenario *Use case* Pinjaman**

| Aktor | Sistem |
|---|---|
| 1. Mengajukan permohonan pinjaman. | |
| | 2. Mengecek persyaratan pinjaman. |
| | 3. Menghitung angsuran pinjam dan bunga pinjaman. |

8. Nama *Use Case* : Perjanjian Pinjaman

   Aktor : Anggota, Ketua, Petugas Koperasi

   Tujuan : Membuat perjanjian pinjaman

**Tabel 4.8. Skenario *Use case* Perjanjian Pinjaman**

| Aktor | Sistem |
|---|---|
| | 1. Menyiapkan perjanjian pinjaman. |
| 2. Memverifikasi perjanjian pinjaman. | |
| 3. Menerima perjanjian pinjaman. | |
| | 4. Mencatat data pinjaman. |

9. Nama *Use Case* : Pembuatan Laporan Pinjaman

   Aktor : Petugas Koperasi, Ketua

   Tujuan : Membuat laporan pinjaman

**Tabel 4.9. Skenario *Use case* Pembuatan Laporan Pinjaman**

| Aktor | Sistem |
|---|---|
| | 1. Membuat laporan pinjaman. |
| 2. Menerima laporan pinjaman. | |

10. Nama *Use Case* : Bayar Angsuran

    Aktor : Anggota, Petugas koperasi

    Tujuan : Melakukan pembayaran angsuran

**Tabel 4.10. Skenario *Use case* Bayar Angsuran**

| Aktor | Sistem |
|---|---|
| 1. Menyerahkan angsuran. | |
| | 2. Mencatat data angsuran. |

11. Nama *Use Case* : Bayar Denda

    Aktor : Anggota, Petugas koperasi

    Tujuan : Membayar denda pinjaman

**Tabel 4.11. Skenario *Use case* Bayar Denda**

| Aktor | Sistem |
|---|---|
| 1. Membayar denda. | |
| | 2. Mencatat data denda. |

12. Nama *Use Case* : Pembuatan Laporan Angsuran

Aktor : Petugas koperasi, Ketua

Tujuan : Membuat laporan angsuran

**Tabel 4.12. Skenario *Use case* Pembuatan Laporan Angsuran**

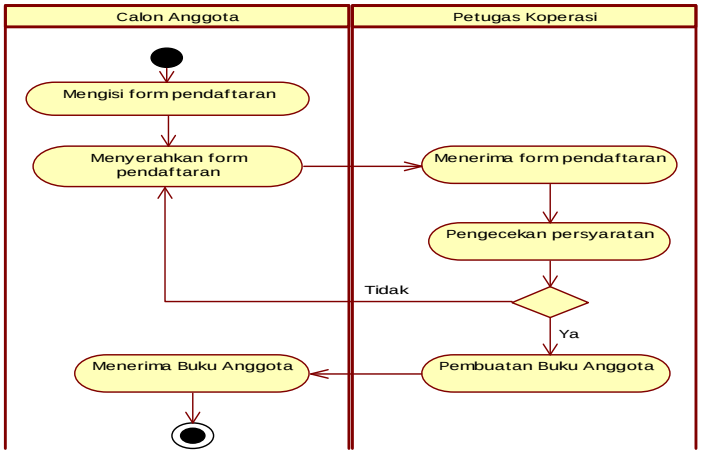
| Aktor | Sistem |
|---|---|
| | 1. Membuat laporan angsuran. |
| 2. Menerima laporan angsuran. | |

### 4.1.1.3. *Activity* Diagram

Pada bagian ini akan digambarkan dokumentasi alur kerja pada sistem yang sedang berjalan yang bertujuan untuk melihat urutan alur proses sistem yang sedang berjalan.

#### 4.1.1.3.1. *Activity* Diagram Pendaftaran Anggota

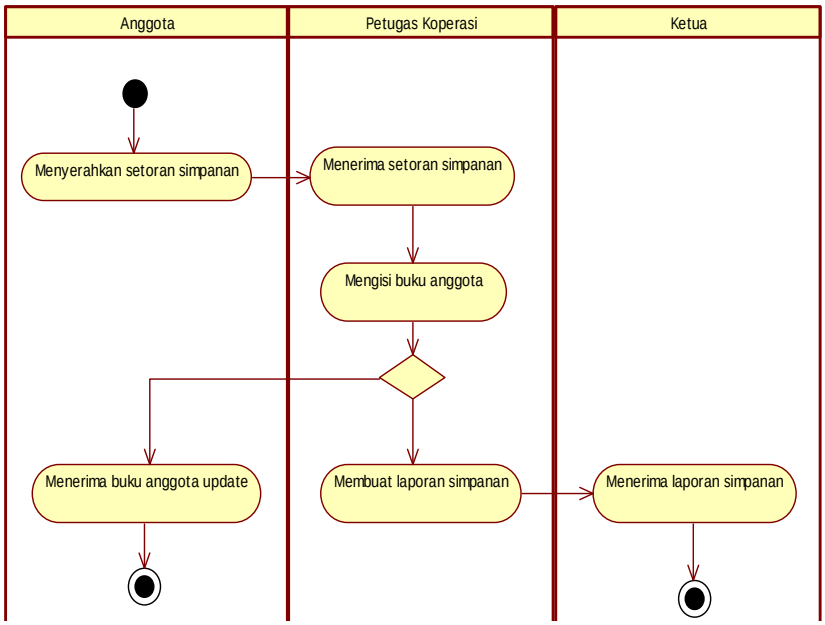
Berikut ini penggambaran *activity* diagram pendaftaran anggota yang sedang berjalan pada Koperasi Pedagang Mitra Abadi Pangalengan adalah sebagai berikut:




Gambar 4.2. *Activity* Diagram Pendaftaran anggota yang sedang berjalan

#### 4.1.1.3.2. *Activity* Diagram Simpanan Anggota

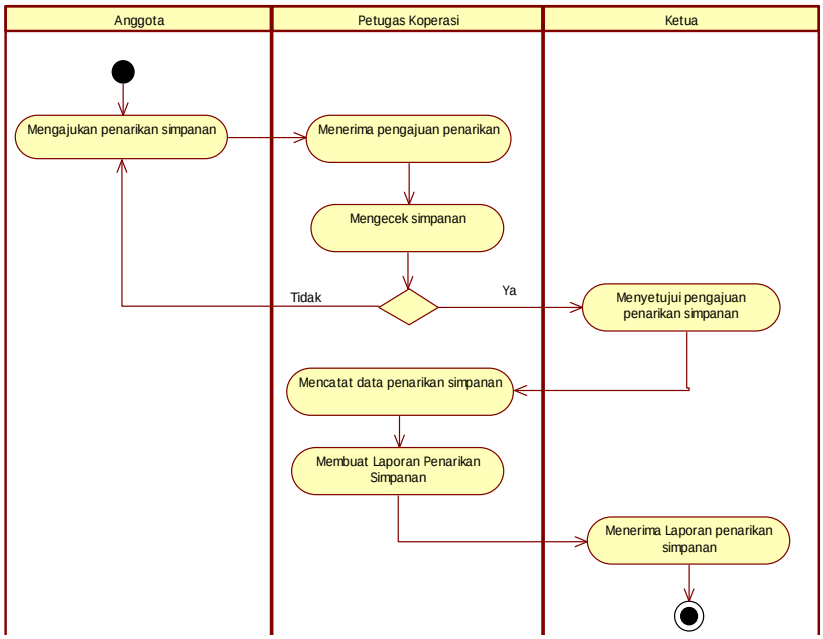
Berikut ini penggambaran *activity* diagram simpanan anggota yang sedang berjalan pada Koperasi Pedagang Mitra Abadi Pangalengan adalah sebagai berikut:




Gambar 4.3. *Activity* Diagram Simpanan Anggota yang sedang berjalan

#### 4.1.1.3.3. *Activity* Diagram Pengambilan Simpanan Anggota

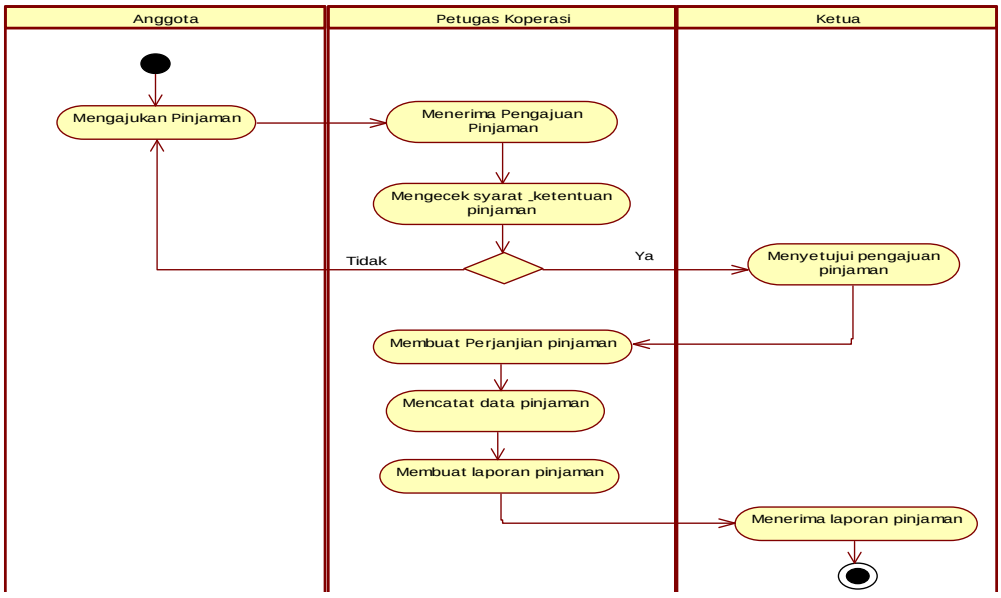
Berikut ini penggambaran *activity* diagram pengambilan simpanan anggota yang sedang berjalan pada Koperasi Pedagang Mitra Abadi Pangalengan adalah sebagai berikut:




Gambar 4.4. *Activity* Diagram Pengambilan Simpanan yang sedang berjalan

#### 4.1.1.3.4. *Activity* Diagram Pinjaman

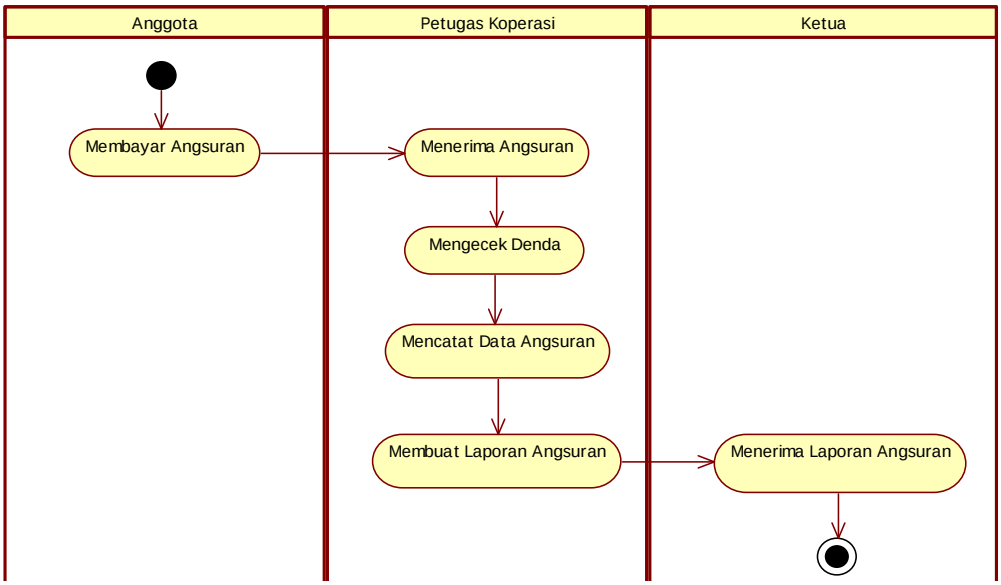
Berikut ini penggambaran *activity* diagram pinjaman yang sedang berjalan pada Koperasi Pedagang Mitra Abadi Pangalengan adalah sebagai berikut:




Gambar 4.5. *Activity* Diagram Pinjaman yang sedang berjalan

#### 4.1.1.3.5. *Activity* Diagram Angsuran

Berikut ini penggambaran *activity* diagram angsuran yang sedang berjalan pada Koperasi Pedagang Mitra Abadi Pangalengan adalah sebagai berikut:




Gambar 4.6. *Activity* Diagram Angsuran yang sedang berjalan

### 4.1.2 Evaluasi Sistem Yang Berjalan

Setelah melakukan analisis terhadap sistem yang sedang berjalan, baik analisis terhadap prosedur yang ada, penulis menemukan beberapa permasalahan baik itu secara langsung maupun tidak langsung yang dapat mempengaruhi kinerja terhadap sistem yang ada. Masalah-masalah ini apabila ditinjau lebih jauh lagi akan berpengaruh terhadap pengembangan sistem nantinya. Setelah melakukan identifikasi terhadap penyebab permasalahan yang ada, penulis juga melakukan pengamatan dengan maksud untuk mengumpulkan fakta atau data yang menjadi permasalahan pada sistem ini. Adapun masalah-masalahnya adalah sebagai berikut:

**Tabel 4.13. Hasil Evaluasi Sistem Informasi yang Berjalan**

| Permasalahan | Rencana Solusi | Bagian |
|---|---|---|
| 1. Kesulitan dalam pencarian data ketika data atau arsip semakin banyak yang disebabkan oleh pencatatan transaksi simpan pinjam masih dituliskan dalam buku simpan pinjam dan buku kas. | Menerapkan *database* untuk mengelola dan menyimpan transaksi simpan pinjam yang juga dapat mempersingkat waktu dalam melakukan pencarian data anggota. | Petugas Koperasi |
| 2. Pembuatan laporan memerlukan waktu yang cukup lama, karena pembuatan laporan dilakukan dengan merekap data atau arsip simpan dan pinjaman. | Program aplikasi yang dirancang dapat mempercepat dalam pembuatan laporan seluruh transaksi simpan pinjam, sehingga tidak ada lagi keterlambatan dalam pelaporan seluruh transaksi simpan pinjam. | Petugas Koperasi |

**Tabel 4.13. Hasil Evaluasi Sistem Informasi yang Berjalan (Lanj)**

| Permasalahan | Rencana Solusi | Bagian |
|---|---|---|
| 3. No anggota yang sekarang diterapkan hanya berupa berupa urutan no urut. | Merancang kodefikasi No anggota agar bisa bersifat unik. | Petugas Koperasi |
| 4. Belum adanya aplikasi untuk membantu proses perhitungan transaksi pinjaman sehingga proses pinjaman menjadi lambat. | Membuat *software* aplikasi yang dapat membantu mempercepat proses pinjaman. | Petugas Koperasi |

## 4.2. Perancangan Sistem

Perancangan sistem atau desain sistem dilakukan apabila tahap dari analisis sistem telah selesai dilakukan. Berdasarkan pada hasil analisis sistem yang sedang berjalan pada bab sebelumnya, maka diusulkan perancangan sistem baru, dimana kinerja dari suatu sistem yang baru diharapkan dapat mengatasi beberapa permasalahan yang ada sebelumnya.

### 4.2.1. Tujuan Perancangan Sistem yang Diusulkan

Tujuan dari perancangan sistem adalah sebagai gambaran umum pengembangan sistem usulan yang dapat memberikan spesifikasi sistem usulan perangkat lunak kepada pengguna dengan menganalisis kelemahan-kelemahan

dalam sistem yang sudah ada yang mengembangkan sistem yang telah ada demi kepuasan, kenyamanan dan kemudahan. Selain itu perancangan sistem bertujuan untuk mempermudah petugas koperasi dalam melakukan pelayanan simpan pinjam.

### 4.2.2. Gambaran Umum Sistem yang Diusulkan

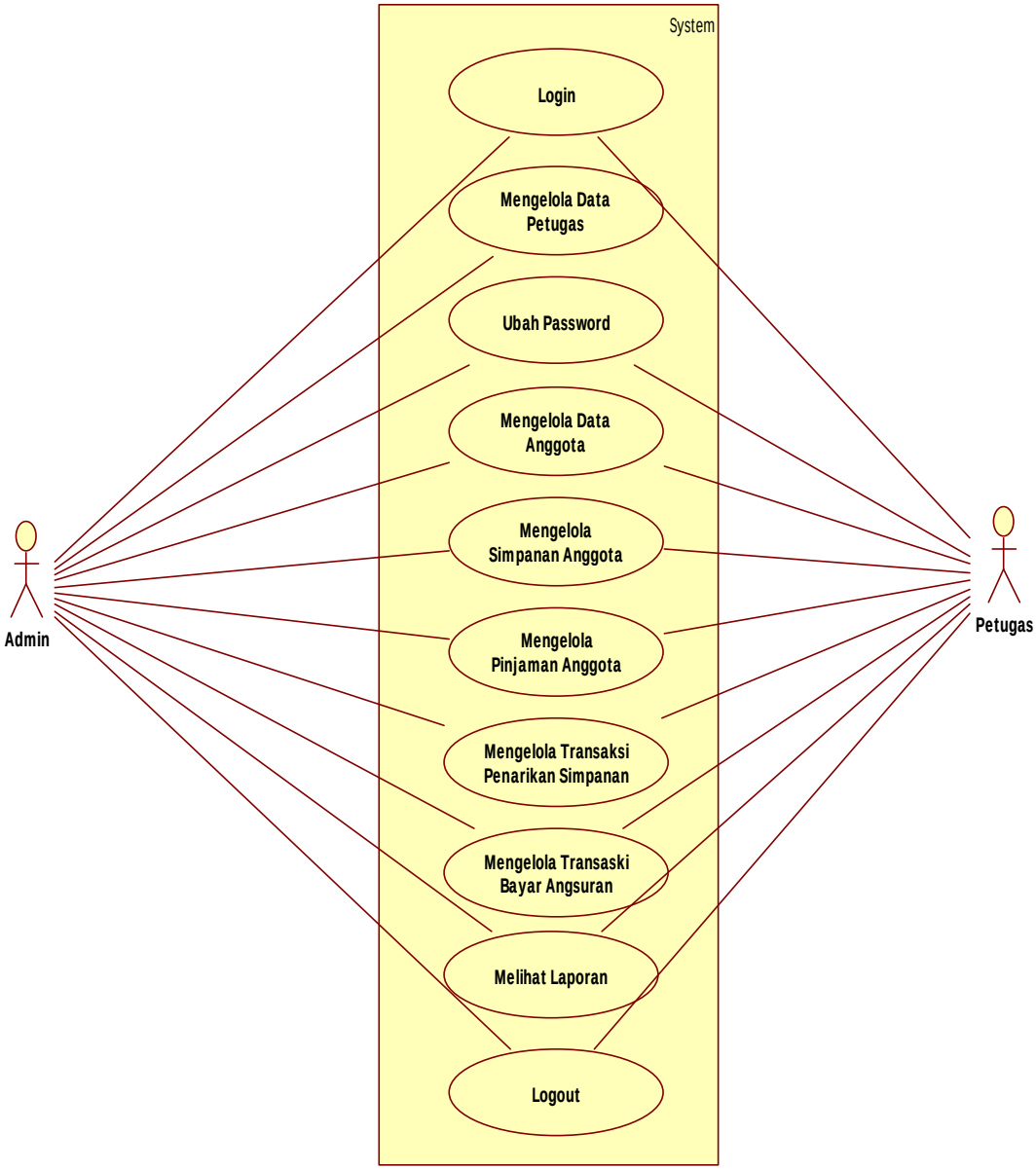
Gambaran sistem yang diusulkan merupakan berbentuk aplikasi sistem informasi koperasi simpan pinjam untuk membantu petugas dalam melakukan pengolahan data. Dengan dibangunnya aplikasi tersebut diharapkan bisa membantu petugas koperasi dalam kegiatan simpan pinjam yang meliputi mengolah data simpanan, penarikan simpanan, mengelola pinjaman, mengelola angsuran, perhitungan denda dan cicilan secara otomatis, dan pembuatan laporan.

### 4.2.3. Perancangan Prosedur yang Diusulkan

Didalam perancangan ini akan menjelaskan tentang bagaimana pihak koperasi melakukan aktivitas simpan pinjam dengan sistem yang diusulkan.

#### 4.2.3.1. Diagram *Use case* yang diusulkan

Diagram *use case* memperlihatkan pada kita hubungan-hubungan yang terjadi antara aktor-aktor dengan *use case-use case* dalam sistem.




Gambar 4.7 *Use case* Diagram Simpan Pinjam yang Diusulkan

#### 4.2.3.2 Skenario *Use Case*

Skenario *Use Case* digunakan untuk memudahkan dalam menganalisa skenario yang akan kita gunakan pada fase-fase selanjutnya dengan melakukan penilaian terhadap skenario tersebut.

1. Nama *Use Case* : Login

   Aktor : Petugas koperasi, Admin

   Tujuan : *Use case* ini menggambarkan kejadian pada saat pertama kali aktor masuk kedalam sistem. Aktor harus melakukan login dengan memasukan *username* serta *password*

**Tabel 4.14. Skenario *Use case* Login**

| Aktor | Respon Sistem |
|---|---|
| 1. Aktor masuk ke halaman login sistem. | |
| 2. Aktor mengisi form login dengan *username* serta *password.* | |
| | 3. Sistem memverifikasi data *username* serta *password* yang dimasukan oleh aktor. |
| | 4. Sistem mencocokan *username* serta *password* apakah ada di dalam database. |
| | 5. Jika cocok, maka sistem akan berpindah kehalaman utama. |

2. Nama *Use Case* : Mengelola Data Petugas

   Aktor : Admin

   Tujuan :*Use case* ini menggambarkan kejadian saat aktor mengelola data petugas, yaitu menambah, mengedit dan menghapus data petugas.

a. Skenario mengelola tambah data petugas

**Tabel 4.15. Skenario *Use case* Mengelola Tambah Data Petugas**

| Aktor | Respon Sistem |
|---|---|
| 1. Aktor memilih menu untuk mengelola data petugas. | |
| | 2. Sistem menampilkan halaman untuk mengelola data petugas. |
| 3. Aktor memilih pilihan tambah data petugas. | |
| 4. Aktor menginputkan data petugas. | |
| | 5. Sistem menyimpan data petugas pada *database.* |

b. Skenario mengelola edit data petugas

**Tabel 4.16. Skenario *Use case* Mengelola Edit Data Petugas**

| Aktor | Respon Sistem |
|---|---|
| 1. Aktor memilih menu untuk mengelola data petugas. | |
| | 2. Sistem menampilkan halaman untuk mengelola data petugas. |
| 3. Aktor memilih pilihan edit data petugas. | |
| | 4. Sistem menampilkan data petugas yang akan diedit. |
| 5. Aktor memilih data petugas yang akan diedit. | |
| 6. Aktor mengubah data petugas yang akan diedit. | |
| | 7. Sistem menyimpan perubahan data petugas pada *database.* |

c. Skenario mengelola hapus data petugas

**Tabel 4.17. Skenario *Use case* Mengelola Hapus Data Petugas**

| Aktor | Respon Sistem |
|---|---|
| 1. Aktor memilih menu untuk mengelola data petugas. | |
| | 2. Sistem menampilkan halaman untuk mengelola data petugas. |
| 3. Aktor memilih pilihan hapus data petugas. | |
| | 4. Sistem menampilkan data petugas yang akan dihapus. |
| 5. Aktor memilih data petugas yang akan dihapus. | |
| | 6. Sistem menghapus data petugas pada *database.* |

3. Nama *Use Case* : Ubah *Password*

   Aktor : Petugas koperasi, Admin

   Tujuan :*Use case* ini menggambarkan kejadian saat aktor mengubah *password*.

**Tabel 4.18. Skenario *Use case* Ubah *Password***

| Aktor | Respon Sistem |
|---|---|
| 1. Aktor memilih menu untuk ubah *password.* | |
| | 2. Sistem menampilkan halaman untuk ubah *password.* |
| 3. Aktor menginput *password* lama, *password* baru dan ulangi *password.* | |
| | *4.* Sistem membaca inputan *password* baru. |
| | *5. Password* baru tersimpan pada *database.* |

4. Nama *Use Case* : Mengelola Data Anggota

   Aktor : Petugas koperasi, Admin

   Tujuan :*Use Case* ini menggambarkan kejadian saat aktor mengelola data anggota, yaitu menambah, mengedit dan menghapus data anggota.

a. Skenario mengelola tambah data anggota

**Tabel 4.19. Skenario *Use case* Mengelola Tambah Data Anggota**

| Aktor | Respon Sistem |
|---|---|
| 1. Aktor memilih menu untuk mengelola data anggota. | |
| | 2. Sistem menampilkan halaman untuk mengelola data anggota. |
| 3. Aktor memilih pilihan tambah data anggota. | |
| 4. Aktor menginputkan data anggota baru. | |
| | 5. Sistem menyimpan data anggota pada *database.* |

b. Skenario mengelola edit data anggota

**Tabel 4.20. Skenario *Use case* Mengelola Edit Data Anggota**

| Aktor | Respon Sistem |
|---|---|
| 1. Aktor memilih menu untuk mengelola data anggota. | |
| | 2. Sistem menampilkan halaman untuk mengelola data anggota. |
| 3. Aktor memilih pilihan edit data anggota. | |
| | 4. Sistem menampilkan data anggota yang akan dirubah. |
| 5. Aktor memilih data anggota yang akan diedit. | |
| 6. Aktor merubah data anggota yang akan diedit. | |
| | 7. Sistem menyimpan perubahan data anggota pada *database.* |

c. Skenario mengelola hapus data anggota

**Tabel 4.21. Skenario *Use case* Mengelola Hapus Data Anggota**

| Aktor | Respon Sistem |
|---|---|
| 1. Aktor memilih menu untuk mengelola data anggota. | |
| | 2. Sistem menampilkan halaman untuk mengelola data anggota. |
| 3. Aktor memilih pilihan hapus data anggota. | |
| | 4. Sistem menampilkan data yang akan dihapus. |
| 5. Aktor memilih data anggota yang akan dihapus. | |
| | 6. Sistem menghapus data anggota pada *database.* |

5. Nama *Use Case* : Mengelola Simpanan Anggota

   Aktor : Petugas koperasi, Admin

   Tujuan :*Use Case* ini menggambarkan kejadian saat aktor mengelola input dan edit simpanan anggota.

a. Skenario mengelola input data simpanan

**Tabel 4.22. Skenario *Use case* Mengelola Input Simpanan Anggota**

| Aktor | Respon Sistem |
|---|---|
| 1. Aktor memilih menu untuk mengelola simpanan anggota. | |
| | 2. Sistem menampilkan halaman untuk mengelola simpanan anggota. |
| 3. Aktor memilih data anggota yang akan melakukan simpanan. | |
| 4. Aktor menginput simpanan anggota. | |
| | 5. Sistem membaca inputan *simpanan* anggota. |
| | 6. Menyimpan simpanan anggota pada *database.* |
| 7. Aktor memilih pilihan cetak. | |
| | 8. Mencetak struk simpanan. |

b. Skenario mengelola edit data simpanan

**Tabel 4.23. Skenario *Use case* Mengelola Edit Data Simpanan**

| Aktor | Respon Sistem |
|---|---|
| 1. Aktor memilih menu untuk mengelola data anggota. | |
| | 2. Sistem menampilkan halaman untuk mengelola data simpanan. |
| 3. Aktor memilih pilihan edit simpanan. | |
| | 4. Sistem menampilkan data simpanan yang akan diedit. |
| 5. Aktor memilih data simpanan yang akan diedit. | |
| 6. Aktor merubah data simpanan. | |
| | 7. Sistem menyimpan data perubahan simpanan pada *database.* |

6. Nama *Use Case* : Mengelola Pinjaman Anggota

   Aktor : Petugas koperasi, Admin

   Tujuan :*Use Case* ini menggambarkan kejadian saat aktor mengelola input dan edit pinjaman anggota.

a. Skenario mengelola input pinjaman

**Tabel 4.24. Skenario *use case* Mengelola Pinjaman Anggota**

| Aktor | Respon Sistem |
|---|---|
| 1. Aktor memilih menu untuk mengelola pinjaman anggota. | |
| | 2. Sistem menampilkan halaman untuk mengelola pinjaman. |
| 3. Aktor memilih data anggota yang akan melakukan pinjaman. | |
| 4. Aktor menginput pinjaman. | |
| | 5. Sistem membaca inputan pinjaman anggota. |
| | 6. Sistem mengecek persyaratan pinjaman dengan data dalam *database*. |
| | 7. Jika ketentuan sudah terpenuhi maka data akan tersimpan pada *database* dan jika tidak maka penyimpanan dibatalkan. Dan menampilkan pesan kesalahan. |
| 8. Aktor memilih pilihan cetak. | |
| | 9. Mencetak bukti pinjaman. |

b. Skenario mengelola edit data pinjaman

**Tabel 4.25. Skenario *Use case* Mengelola Edit Data Pinjaman**

| Aktor | Respon Sistem |
|---|---|
| 1. Aktor memilih menu untuk mengelola data pinjaman. | |
| | 2. Sistem menampilkan halaman untuk mengelola data pinjaman. |
| 3. Aktor memilih pilihan edit pinjaman. | |
| | 4. Sistem akan menampilkan data pinjaman yang akan diedit. |
| 5. Aktor memilih data pinjaman yang akan di edit. | |
| 6. Aktor merubah data pinjaman. | |
| | 7. Sistem menyimpan data perubahan pinjaman pada *database.* |

7. Nama *Use Case* : Mengelola Transaksi Penarikan Simpanan

   Aktor : Petugas koperasi, Admin

   Tujuan :*Use Case* ini menggambarkan kejadian saat aktor mengelola transaksi penarikan simpanan dan edit penarikan simpanan.

a. Skenario mengelola transaksi penarikan

**Tabel 4.26. Skenario *Use case* Mengelola transaksi penarikan simpanan**

| Aktor | Respon Sistem |
|---|---|
| 1. Aktor memilih menu untuk mengelola penarikan simpanan. | |
| | 2. Menampilkan halaman untuk mengelola penarikan simpanan. |
| 3. Aktor memilih data anggota yang akan melakukan penarikan. | |
| 4. Aktor menginput jumlah penarikan simpanan. | |
| | 5. Sistem membaca inputan penarikan simpanan. |
| | 6. Sistem mengecek data simpanan pada *database*. |
| | 7. Menyimpan data penarikan simpanan pada *database*. |
| 8. Aktor memilih pilihan cetak. | |
| | 9. Mencetak struk penarikan simpanan. |

b. Skenario mengelola edit data penarikan

**Tabel 4.27. Skenario *Use case* Mengelola Edit Data Penarikan**

| Aktor | Respon Sistem |
|---|---|
| 1. Aktor memilih menu untuk mengelola transaksi penarikan. | |
| | 2. Sistem menampilkan halaman untuk mengelola transaksi penarikan. |
| 3. Aktor memilih pilihan edit penarikan. | |
| | 4. Sistem menampilkan data penarikan yang akan diedit. |
| 5. Aktor memilih data penarikan yang akan diedit. | |
| 6. Aktor merubah data penarikan. | |
| | 7. Sistem menyimpan data perubahan penarikan pada *database.* |

8. Nama *Use Case* : Mengelola Transaksi Bayar Angsuran

   Aktor : Petugas koperasi, Admin

   Tujuan :*Use Case* ini menggambarkan kejadian saat aktor mengelola transaksi bayar angsuran anggota.

**Tabel 4.28. Skenario *Use case* Mengelola transaksi bayar angsuran**

| Aktor | Respon Sistem |
|---|---|
| 1. Aktor memilih menu untuk transaksi mengelola bayar angsuran. | |
| | 2. Sistem menampilkan halaman untuk mengelola transaksi bayar angsuran. |
| 3. Aktor mencari data pinjaman yang akan melakukan angsuran. | |
| | 4. Sistem menampilkan data pinjaman anggota. |
| 5. Aktor memilih data pinjaman yang akan melakukan angsuran. | |
| | 6. Sistem mengelola transaksi bayar angsuran. |
| | 7. Menyimpan data angsuran pada *database*. |
| 8. Aktor memilih pilihan cetak. | |
| | 9. Mencetak struk angsuran. |

9. Nama *Use Case* : Melihat Laporan

   Aktor : Petugas koperasi, Admin

   Tujuan : *Use Case* ini menggambarkan kejadian saat aktor melihat laporan yaitu laporan simpanan, laporan pinjaman dan laporan angsuran

**Tabel 4.29. Skenario *Use case* Melihat Laporan**

| Aktor | Respon Sistem |
|---|---|
| 1. Aktor memilih menu untuk melihat laporan. | |
| | 2. Sistem menampilkan halaman untuk melihat laporan. |
| 3. Aktor melihat laporan. | |
| 4. Aktor memilih pilihan cetak laporan. | |
| | 5. Proses mencetak laporan. |

10. Nama *Use Case* : Logout

    Aktor : Petugas koperasi, Admin

    Tujuan : *Use Case* ini menggambarkan kejadian saat aktor keluar dari sistem.

**Tabel 4.30. Skenario *Use case* Logout**

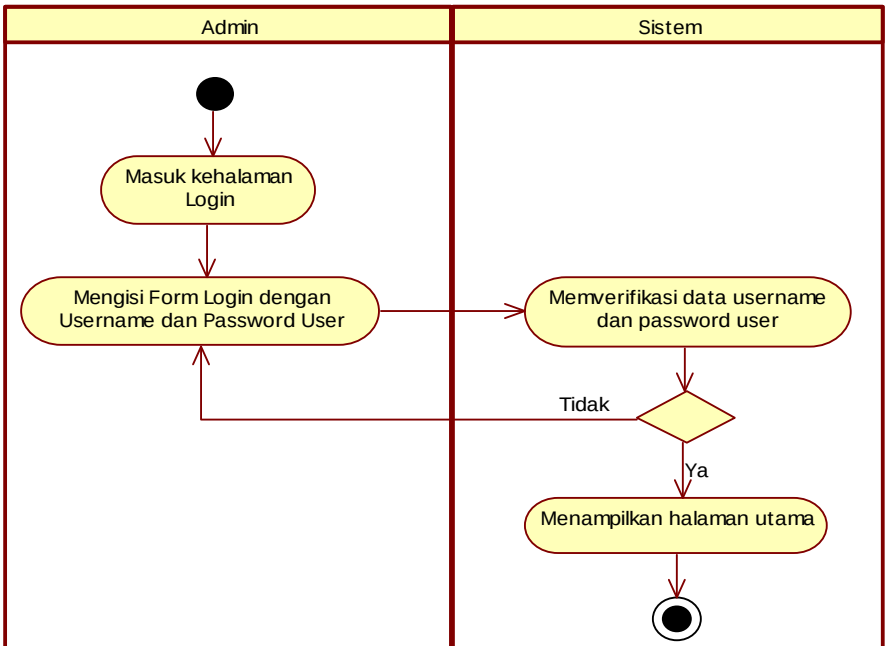
| Aktor | Respon Sistem |
|---|---|
| 1. Aktor memilih pilihan untuk logout. | |
| | 2. Sistem menutup halaman utama dan menampilkan halaman untuk login kembali. |

#### 4.2.3.3. *Activity* Diagram yang Diusulkan

*Activity* diagram (diagram aktivitas) adalah diagram yang menggambarkan aliran fungsionalitas dari sistem. Pada tahap pemodelan bisnis, diagram aktivitas dapat digunakan untuk menunjukkan aliran kerja bisnis (*business work flow*).

##### 4.2.3.3.1. *Activity* Diagram Login

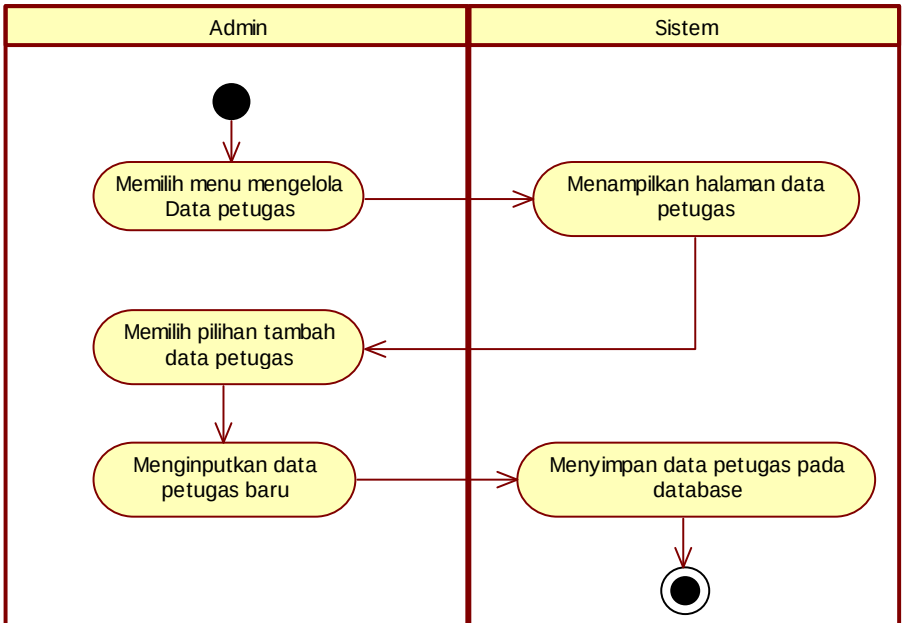
Petugas melakukan login, lalu sistem akan memverifikasi *username* dan *password* yang dimasukan oleh *user* (Admin dan petugas). Berikut ini adalah diagram aktivitas Login yang diusulkan:




Gambar 4.8. *Activity* Diagram Login petugas yang diusulkan

#### 4.2.3.3.2. *Activity* Diagram Mengelola Tambah Data Petugas

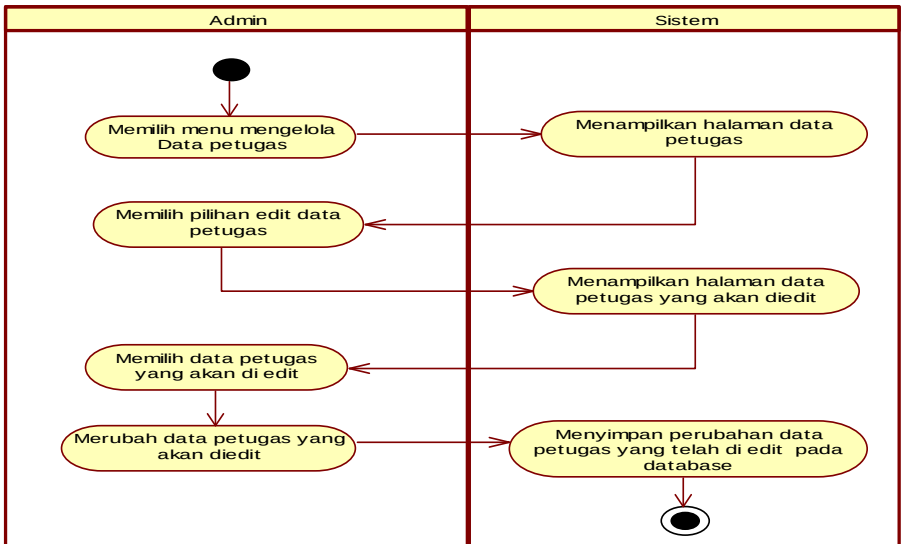
Admin dapat melakukan pengelolaan tambah data petugas seperti. Berikut ini adalah diagram aktivitas mengelola tambah data petugas yang diusulkan:




Gambar 4.9. *Activity* Diagram Mengelola Tambah data petugas

#### 4.2.3.3.3. *Activity* Diagram Mengelola Edit Data Petugas

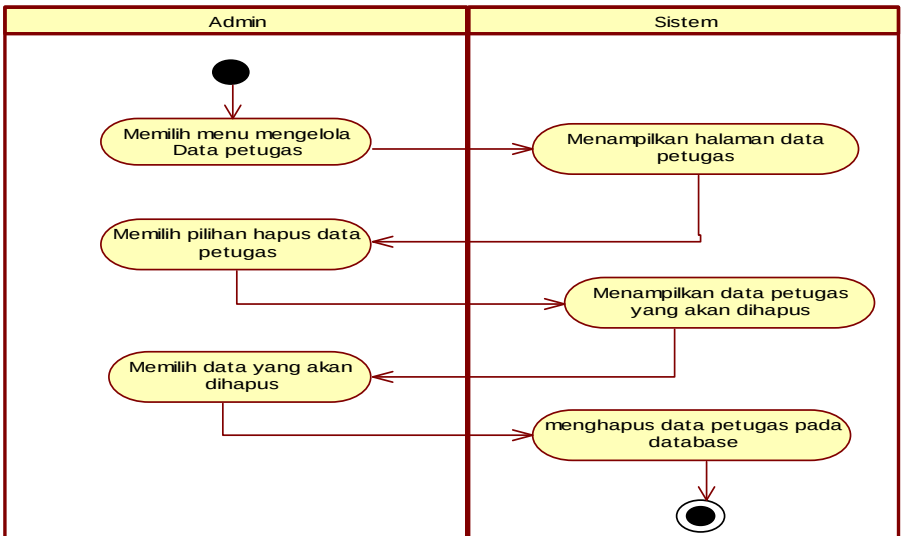
Admin melakukan pengelolaan edit data petugas. Berikut ini adalah diagram aktivitas mengelola edit data petugas yang diusulkan:




Gambar 4.10. *Activity* Diagram Mengelola Edit Data Petugas yang diusulkan

#### 4.2.3.3.4. *Activity* Diagram Mengelola Hapus Data Petugas

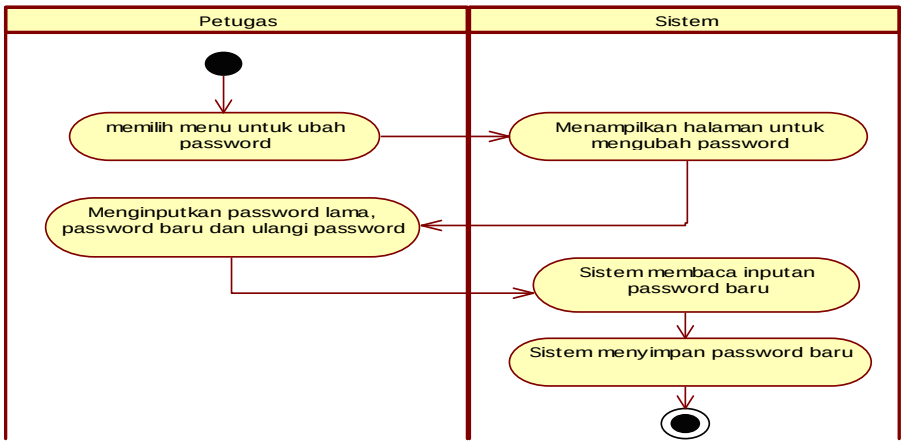
Admin melakukan pengelolaan hapus data petugas. Berikut ini adalah diagram aktivitas mengelola hapus data petugas yang diusulkan:




Gambar 4.11. *Activity* Diagram Mengelola Hapus Data Petugas yang diusulkan

#### 4.2.3.3.5. *Activity* Diagram Ubah *Password*

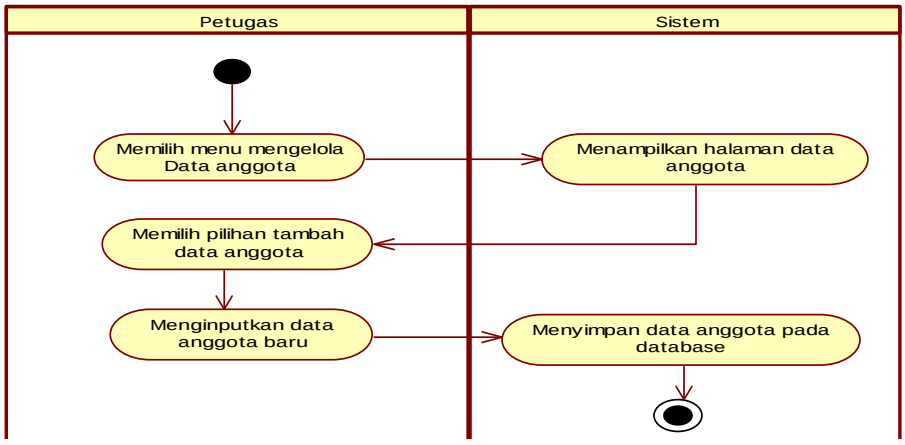
Petugas koperasi dan admin dapat melakukan pengubahan *password* pada sistem. Berikut ini adalah diagram aktivitas ubah *password* yang diusulkan:




Gambar 4.12. *Activity* Diagram Ubah *Password* yang diusulkan

#### 4.2.3.3.6. *Activity* Diagram Mengelola Tambah Data Anggota

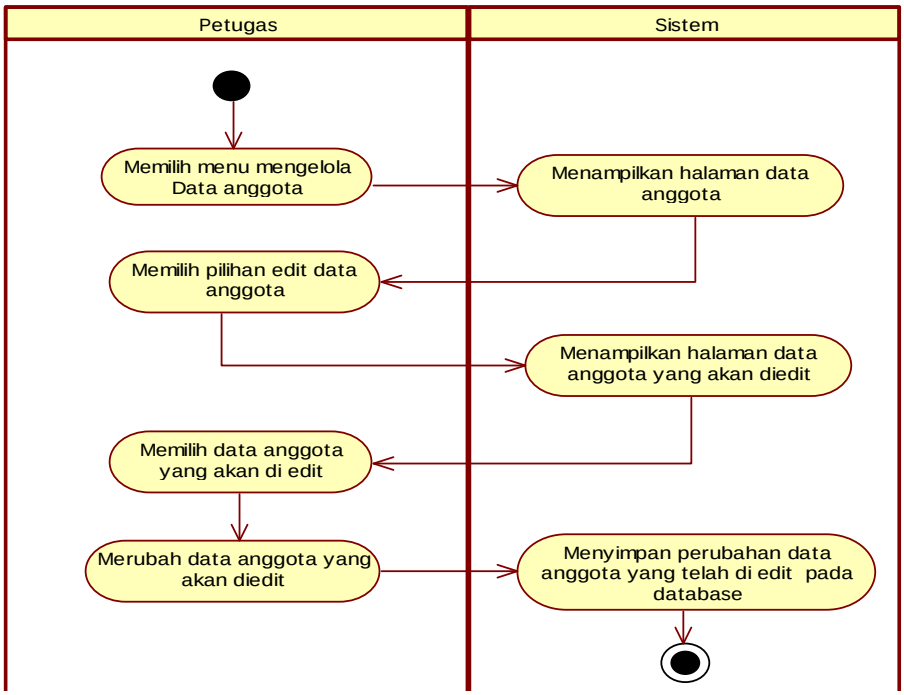
Petugas koperasi dan admin dapat melakukan pengelolaan data anggota seperti menambah data anggota baru pada saat ada anggota baru, mengedit dan menghapus data anggota. Berikut ini adalah diagram aktivitas mengelola tambah data anggota yang diusulkan:




Gambar 4.13. *Activity* Diagram Mengelola Tambah Data Anggota yang diusulkan

#### 4.2.3.3.7. *Activity* Diagram Mengelola Edit Data Anggota

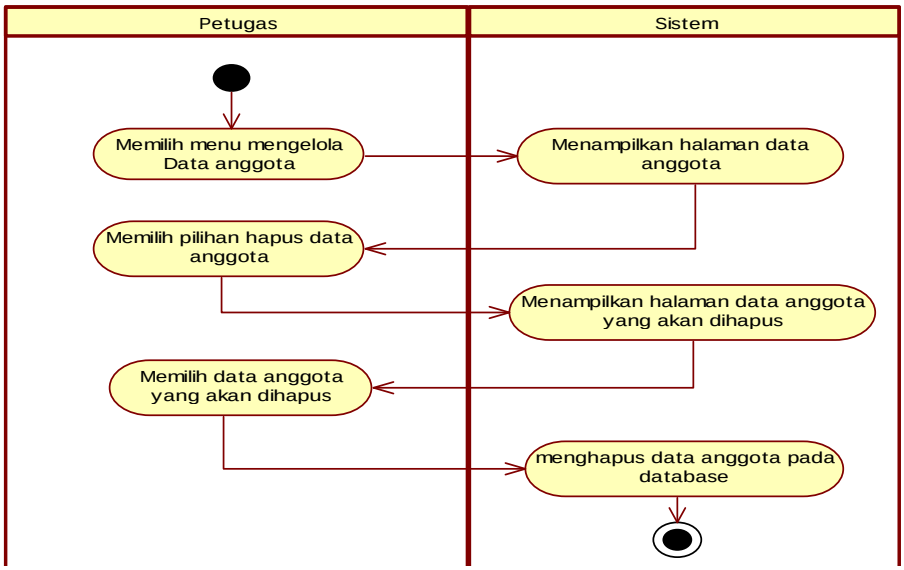
Petugas koperasi dan admin dapat melakukan pengelolaan edit data anggota. Berikut ini adalah diagram aktivitas mengelola edit data anggota yang diusulkan:




Gambar 4.14. *Activity* Diagram Mengelola Edit Data Anggota yang diusulkan

#### 4.2.3.3.8. *Activity* Diagram Mengelola Hapus Data Anggota

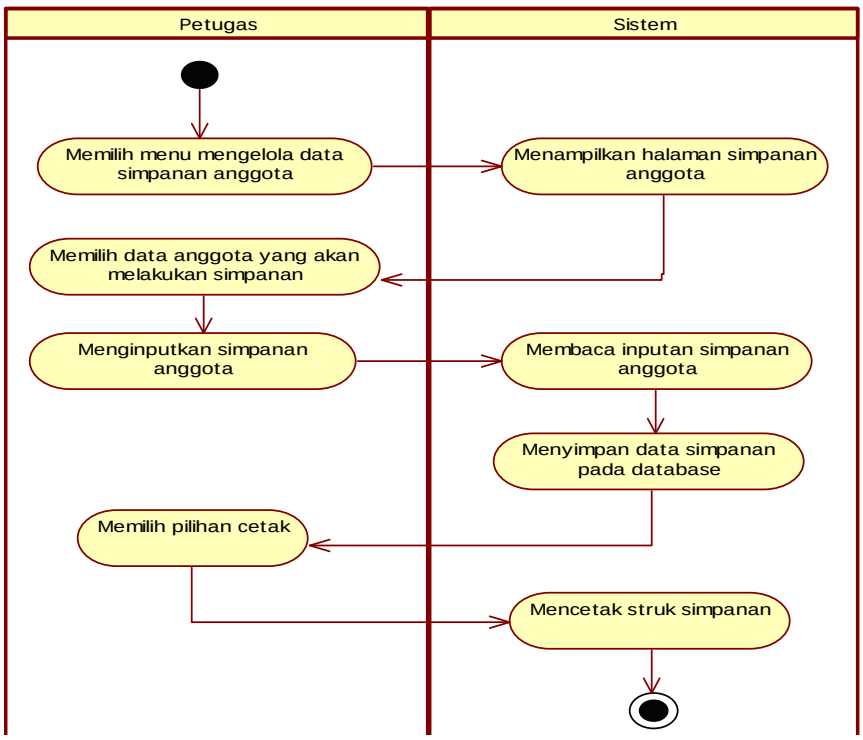
Petugas koperasi dan admin dapat melakukan pengelolaan hapus data anggota. Berikut ini adalah diagram aktivitas mengelola hapus data anggota yang diusulkan:




Gambar 4.15. *Activity* Diagram Mengelola Hapus Data Anggota yang diusulkan

#### 4.2.3.3.9. *Activity* Diagram Mengelola Simpanan Anggota

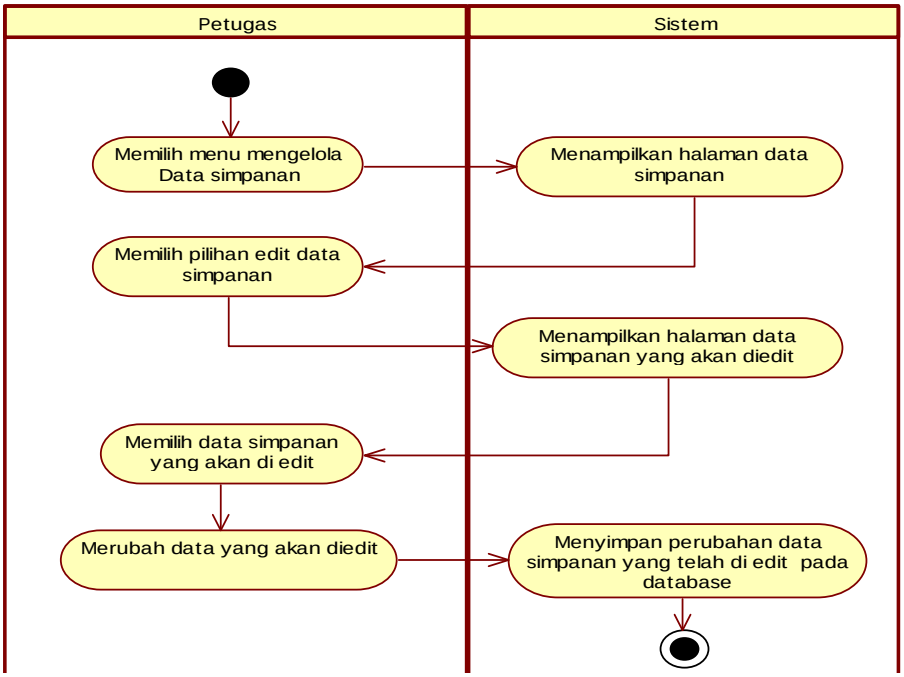
Petugas koperasi dapat melakukan pengelolaan simpanan anggota. Berikut ini adalah diagram aktivitas mengelola simpanan anggota yang diusulkan:




Gambar 4.16. *Activity* Diagram Mengelola Simpanan Anggota yang diusulkan

#### 4.2.3.3.10. *Activity* Diagram Mengelola Edit Simpanan Anggota

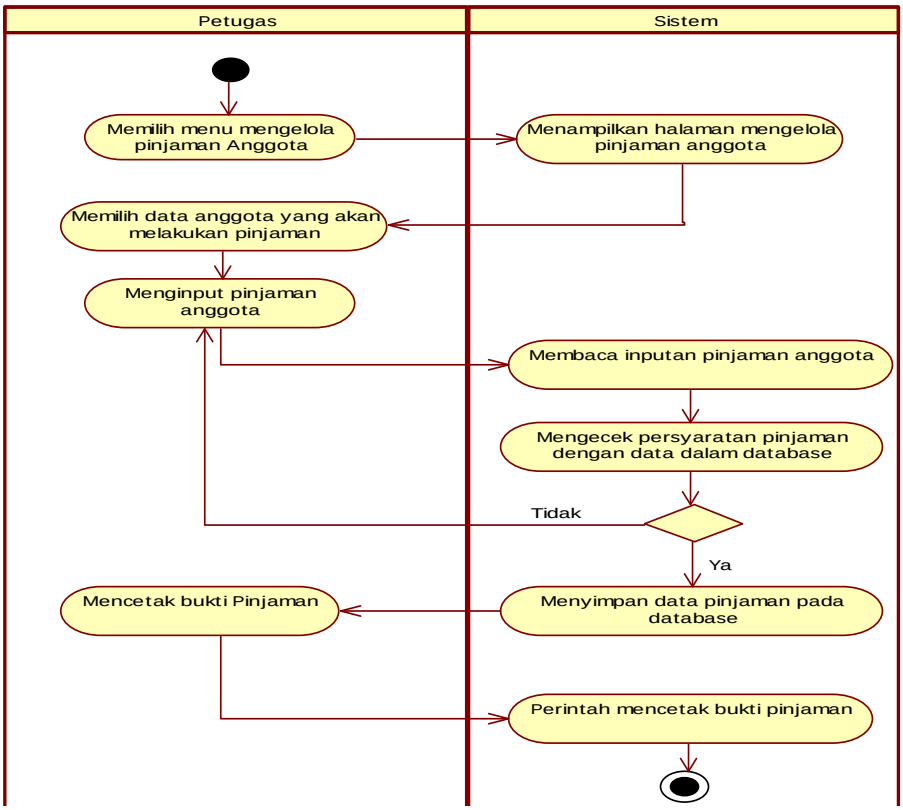
Petugas koperasi dapat melakukan pengelolaan edit simpanan anggota. Berikut ini adalah diagram aktivitas mengelola edit simpanan anggota yang diusulkan:




Gambar 4.17. *Activity* Diagram Mengelola Edit Data simpanan yang diusulkan

#### 4.2.3.3.11. *Activity* Diagram Mengelola Pinjaman Anggota

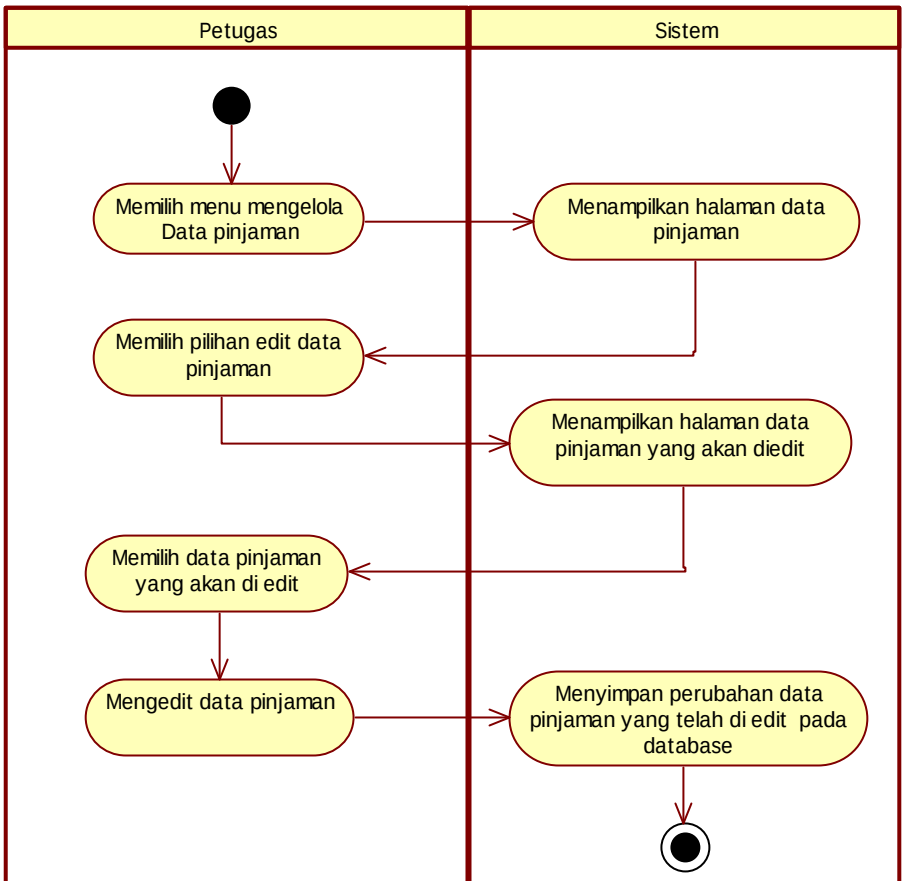
Petugas koperasi dan admin dapat melakukan pengelolaan pinjaman anggota. Ketika anggota ingin melakukan peminjaman ada syarat yang harus dipenuhi seperti keanggotan sudah selama 3 bulan, jumlah pinjaman maksimal 2 kali jumlah simpanan dan tidak sedang mempunyai pinjaman. Berikut ini adalah diagram aktivitas mengelola pinjaman anggota yang diusulkan:




Gambar 4.18. *Activity* Diagram Mengelola Pinjaman Anggota yang diusulkan

#### 4.2.3.3.12. *Activity* Diagram Mengelola Edit Pinjaman Anggota

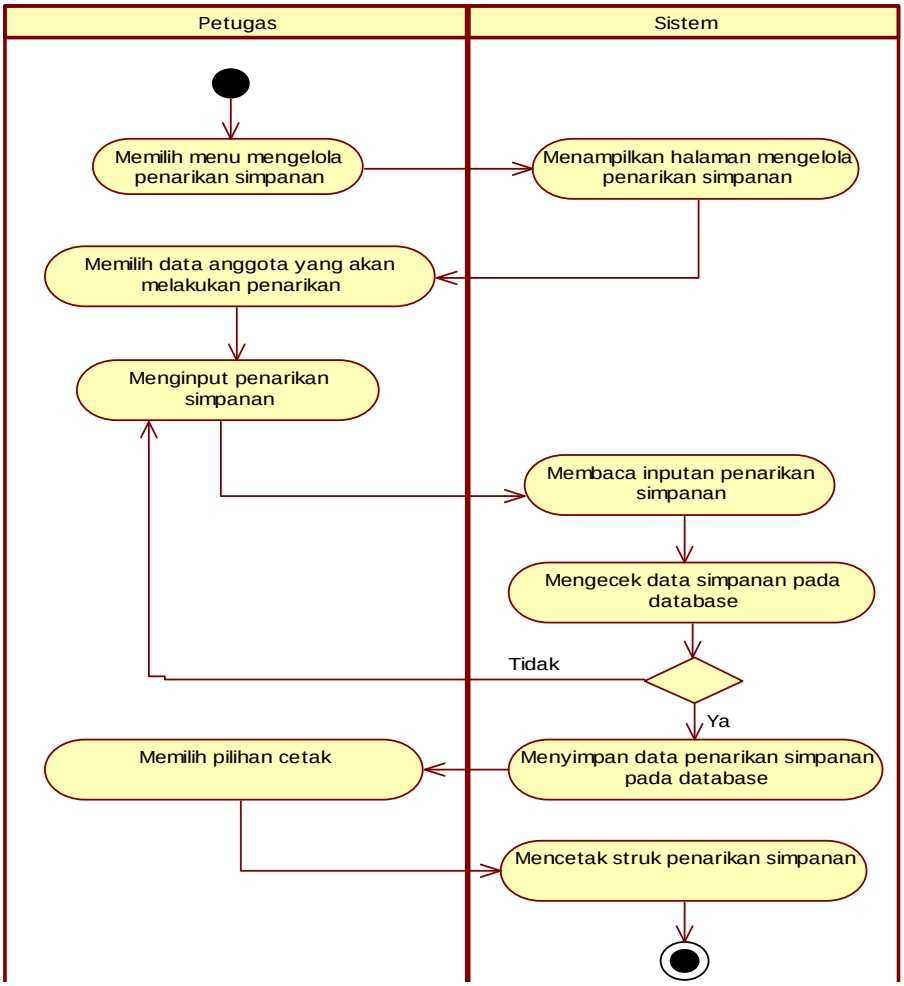
Petugas koperasi dan admin dapat melakukan pengelolaan edit pinjaman anggota, data pinjaman yang bisa diedit adalah data pinjaman yang belum melakukan angsuran. Berikut ini adalah diagram aktivitas mengelola edit pinjaman anggota yang diusulkan:




Gambar 4.19. *Activity* Diagram Mengelola Edit Data Pinjaman yang diusulkan

#### 4.2.3.3.13. *Activity* Diagram Mengelola Transaksi Penarikan Simpanan

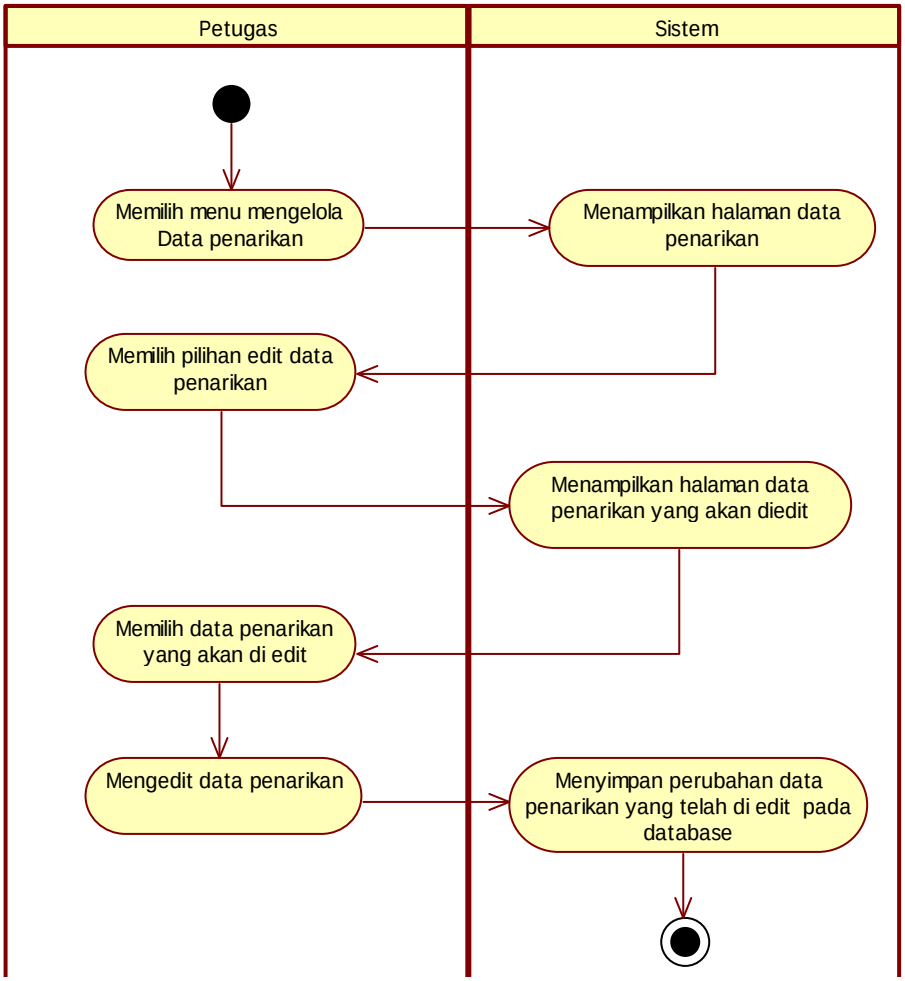
Petugas koperasi dan admin dapat melakukan pengelolaan penarikan simpanan, simpanan yang dapat diambil hanya simpanan sukarela saja dengan mengecek jumlah simpanan untuk mengetahui jumlah simpanan yang dapat diambil. Berikut ini adalah diagram aktivitas mengelola transaksi penarikan simpanan yang diusulkan:




Gambar 4.20. *Activity* Diagram Mengelola penarikan simpanan yang diusulkan

**4.2.3.3.14. *Activity* Diagram Mengelola Edit Transaksi Penarikan Simpanan.**

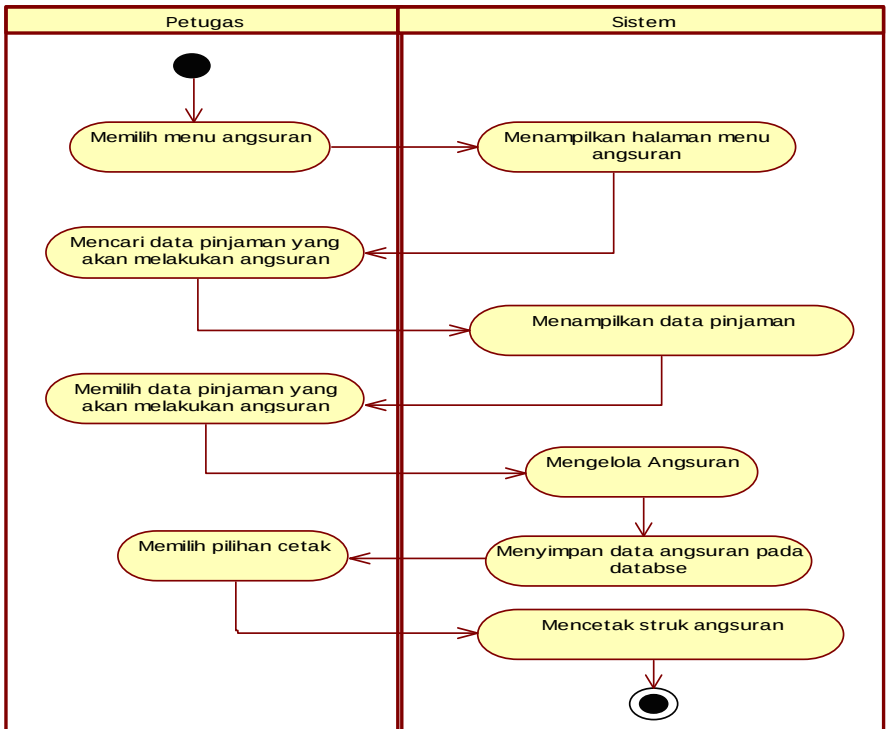
Petugas koperasi dan admin dapat melakukan pengelolaan edit penarikan simpanan. Berikut ini adalah diagram aktivitas mengelola edit transaksi penarikan simpanan yang diusulkan:




Gambar 4.21. *Activity* Diagram Mengelola Edit Data Penarikan yang diusulkan

#### 4.2.3.3.15. *Activity* Diagram Mengelola Transaksi Bayar Angsuran

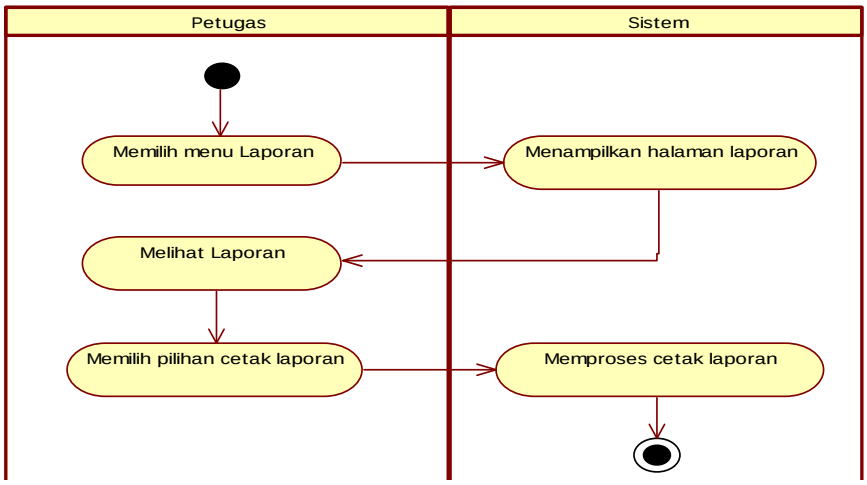
Petugas koperasi dan admin dapat melakukan pengelolaan angsuran dan juga denda, apabila telat mengangsur maka secara otomatis sistem akan menghitung jumlah denda perminggu yang harus dibayar. Berikut ini adalah diagram aktivitas mengelola transaksi bayar angsuran yang diusulkan:




Gambar 4.22. *Activity* Diagram Mengelola Angsuran yang diusulkan

#### 4.2.3.3.16. *Activity* Diagram Melihat Laporan

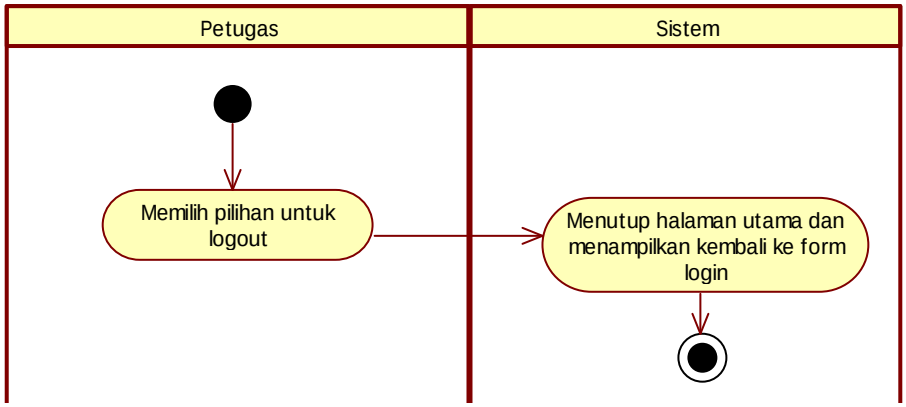
Petugas koperasi dapat melakukan pengelolaan laporan yaitu melihat dan mencetak laporan. Berikut ini adalah diagram aktivitas mengelola laporan yang diusulkan:




Gambar 4.23. *Activity* Diagram Mengelola Laporan yang diusulkan

#### 4.2.3.3.17. *Activity* Diagram *Logout*

Petugas koperasi dapat melakukan *Logout* untuk keluar dari sistem. Berikut ini adalah diagram aktivitas mengelola logout yang diusulkan:



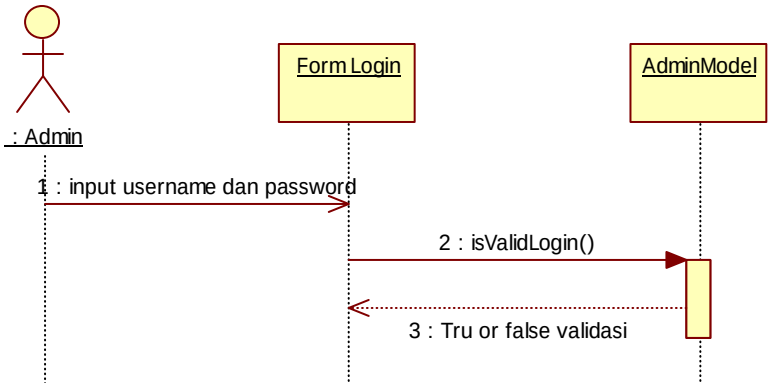

Gambar 4.24. *Activity* Diagram Logout yang diusulkan

### 4.2.3.4. *Sequence* Diagram yang Diusulkan

Pada *Sequence* Diagram menjelaskan aliran fungsionalitas dalam *use case*. Berikut gambar *sequence* diagramnya.

#### 4.2.3.4.1. *Sequence* Diagram Login

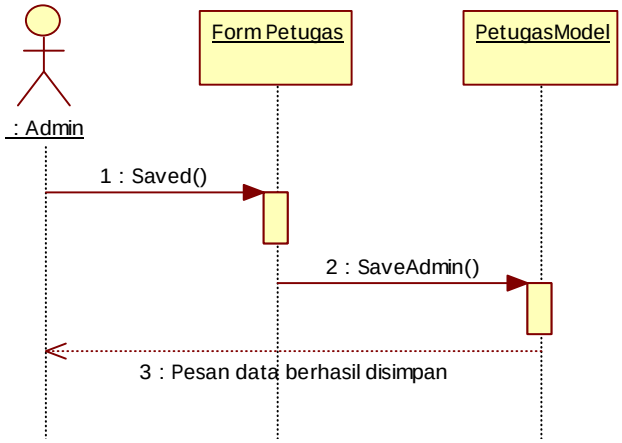
Pada *sequence* diagram login ini dijelaskan bahwa seorang *user* (Admin dan petugas) untuk masuk kehalamannya harus login terlebih dahulu dengan menginputkan *username* dan *password*. Berikut gambar *sequence* diagram loginnya:




Gambar 4.25. *Sequence* Diagram Login yang diusulkan

#### 4.2.3.4.2. *Sequence* Diagram Mengelola Tambah Data Petugas

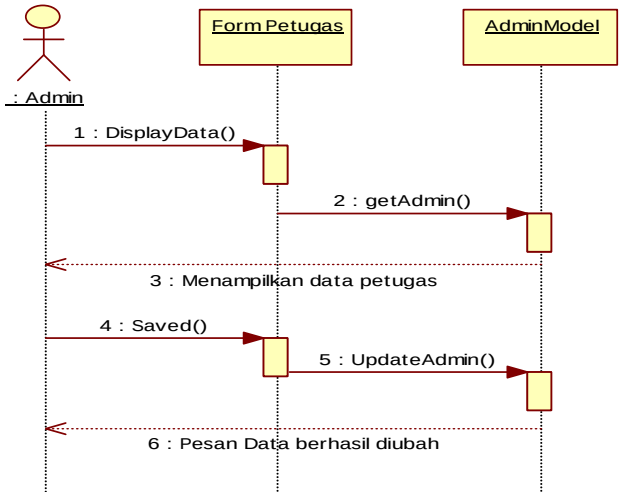
Pada *sequence* diagram mengelola data petugas, admin mengelola data petugas seperti menginputkan data petugas. Berikut gambar *sequence* diagram mengelola tambah data petugas:




Gambar 4.26. *Sequence* Diagram mengelola tambah data petugas yang diusulkan

#### 4.2.3.4.3. *Sequence* Diagram Mengelola Edit Data Petugas

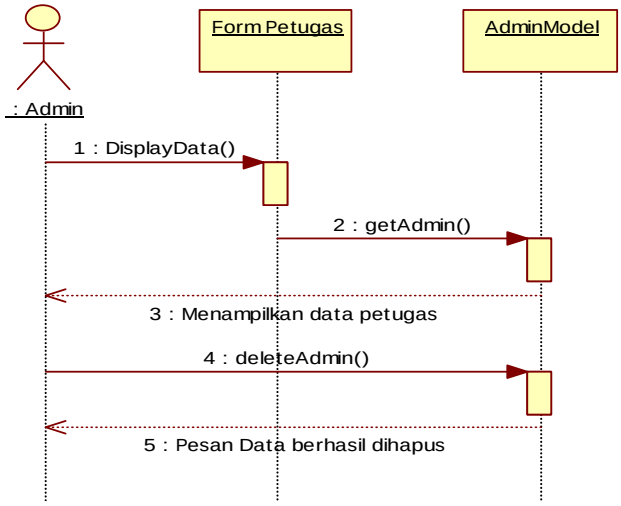
Pada *sequence* diagram mengelola edit data petugas, admin dapat mengelola edit data petugas seperti mengedit nama petugas, *username* dan *password*. Berikut gambar *sequence* diagram mengelola edit data petugas:




Gambar 4.27. *Sequence* Diagram mengelola edit data petugas yang diusulkan

#### 4.2.3.4.4. *Sequence* Diagram Mengelola Hapus Data Petugas

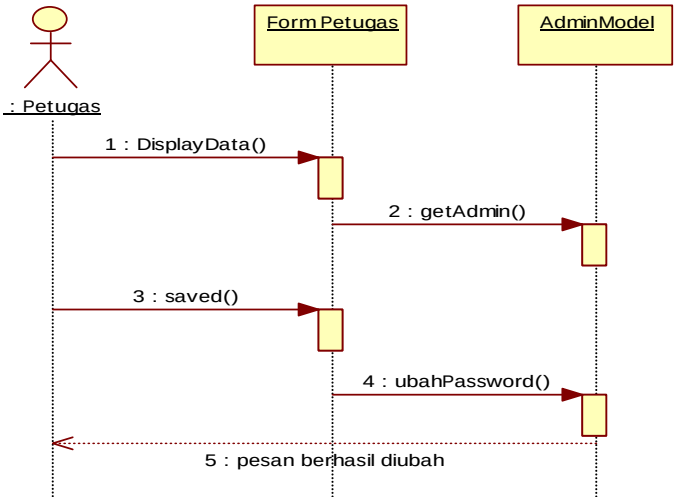
Pada *sequence* diagram mengelola hapus data petugas, admin dapat menghapus data petugas seperti. Berikut gambar *sequence* diagram mengelola hapus data petugas:




Gambar 4.28. *Sequence* Diagram mengelola hapus data petugas yang diusulkan

#### 4.2.3.4.5. *Sequence* Diagram Ubah *Password*

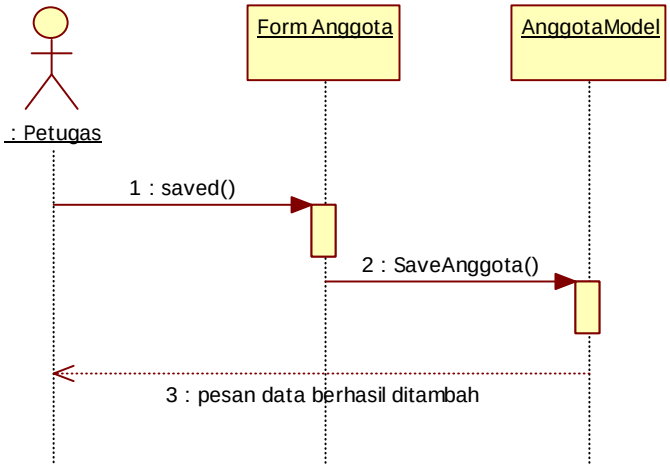
Pada *sequence* diagram ubah *password*, *user* dapat mengubah *password* baru. Berikut gambar *sequence* diagram ubah *password*:




Gambar 4.29. *Sequence* Diagram ubah password yang diusulkan

#### 4.2.3.4.6. *Sequence* Diagram Mengelola Tambah Data Anggota

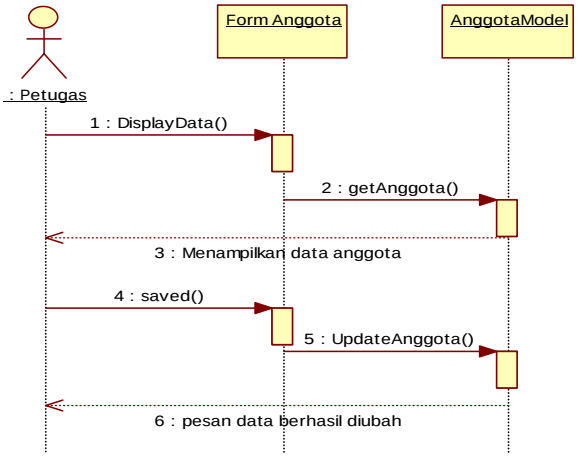
Pada *sequence* diagram mengelola data anggota setiap menambah anggota baru akan mendapatkan no anggota dan no simpanan secara otomatis. Berikut gambar *sequence* diagram mengelola tambah data anggota:




Gambar 4.30. *Sequence* Diagram mengelola tambah data anggota yang diusulkan

#### 4.2.3.4.7. *Sequence* Diagram Mengelola Edit Data Anggota

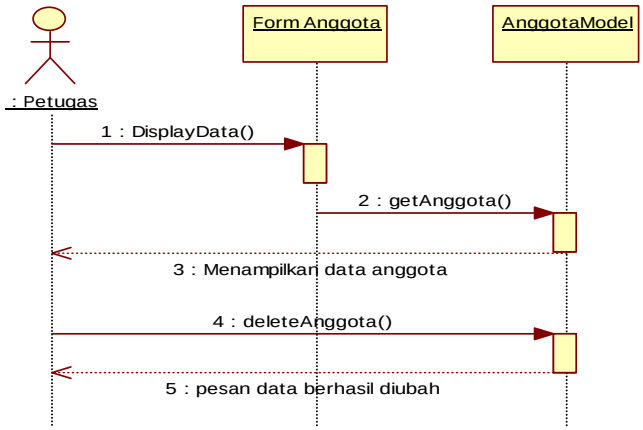
Pada *sequence* diagram mengelola edit data anggota, petugas dan admin dapat mengelola edit data petugas. Berikut gambar *sequence* diagram mengelola edit data anggota:




Gambar 4.31. *Sequence* Diagram mengelola edit data anggota yang diusulkan

#### 4.2.3.4.8. *Sequence* Diagram Mengelola Hapus Data Anggota

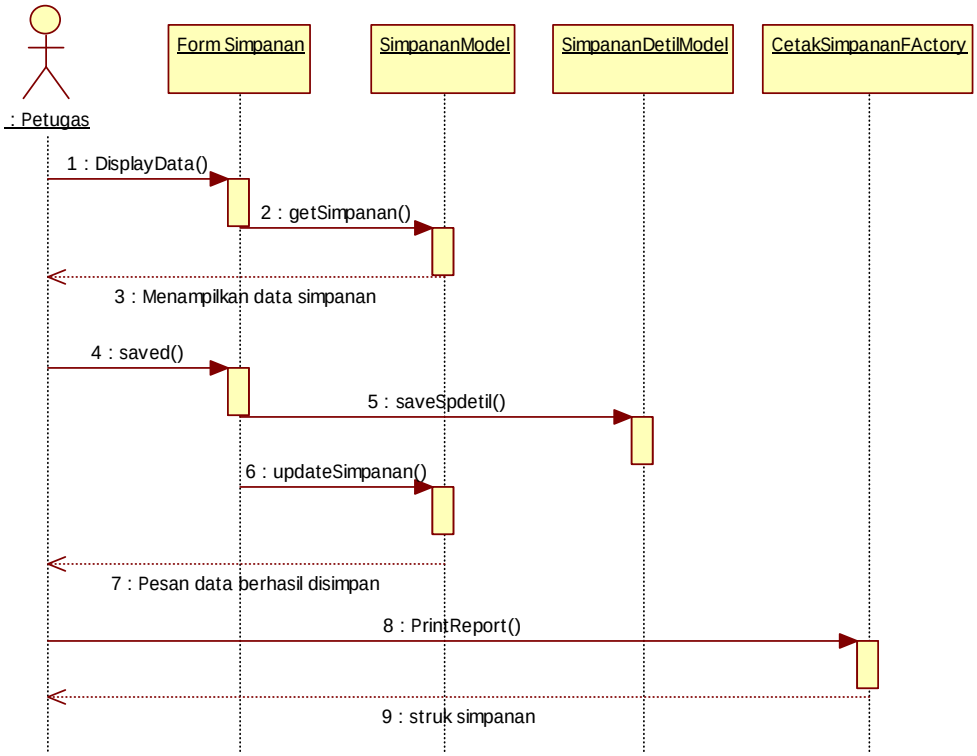
Pada *sequence* diagram mengelola edit data anggota, petugas dan admin dapat menghapus data anggota. Berikut gambar *sequence* diagram mengelola hapus data anggota:




Gambar 4.32. *Sequence* Diagram mengelola hapus data anggota yang diusulkan

#### 4.2.3.4.9. *Sequence* Diagram Simpanan Anggota

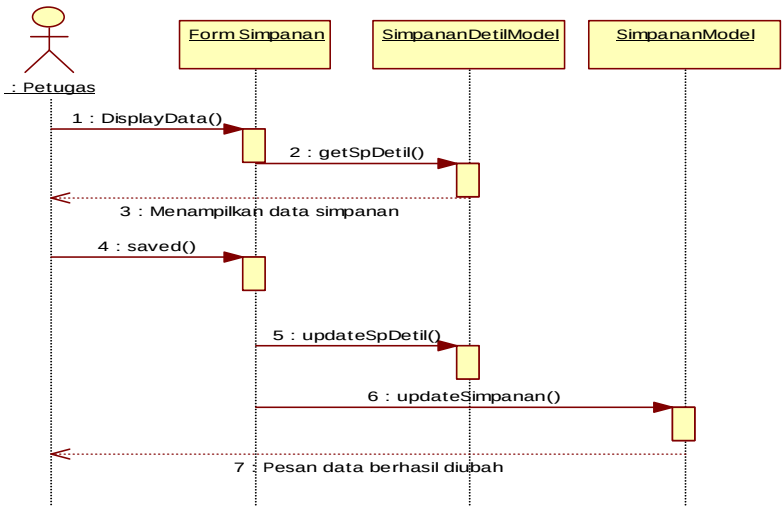
Pada *sequence* diagram simpanan anggota, terdapat tiga jenis simpanan yaitu simpanan pokok yang dilakukan satu kali ketika menjadi anggota koperasi, simpanan wajib setiap minggunya dan simpanan sukarela. Berikut gambar *sequence* diagram simpanan anggota:




Gambar 4.33. *Sequence* Diagram mengelola simpanan yang diusulkan

#### 4.2.3.4.10. *Sequence* Diagram Mengelola Edit Simpanan Anggota

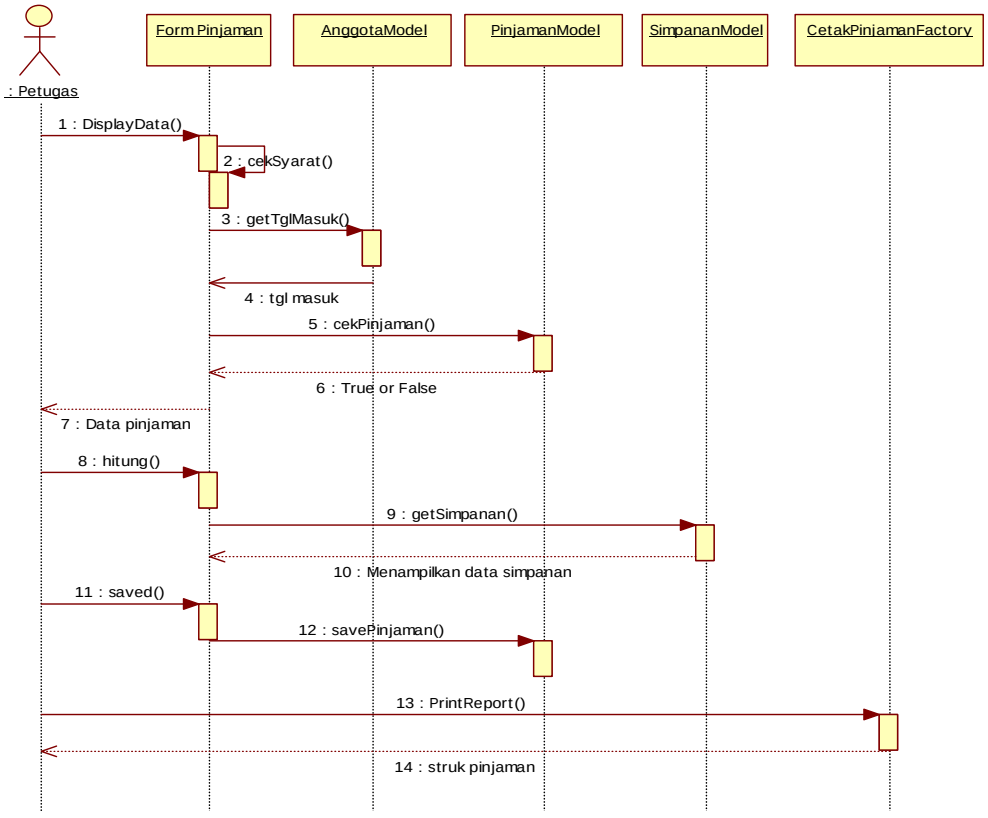
Pada *sequence* diagram simpanan anggota, admin dan petugas dapat melakukan pengelolaan edit data simpanan. Berikut gambar *sequence* diagram edit simpanan anggota:




Gambar 4.34. *Sequence* Diagram mengelola edit data simpanan yang diusulkan

#### 4.2.3.4.11. *Sequence* Diagram Pinjaman Anggota

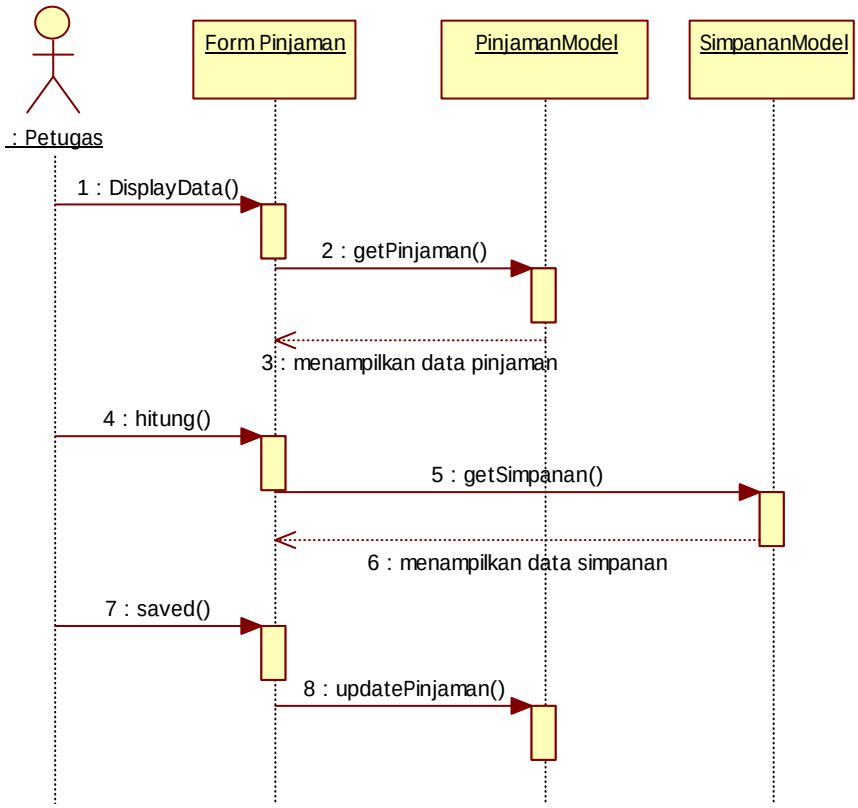
Pada *sequence* diagram pinjaman menggambarkan user untuk melakukan transaksi pinjaman. Berikut gambar *sequence* diagram pinjaman:




Gambar 4.35. *Sequence* Diagram mengelola pinjaman yang diusulkan

#### 4.2.3.4.12. *Sequence* Diagram Mengelola Edit Pinjaman Anggota

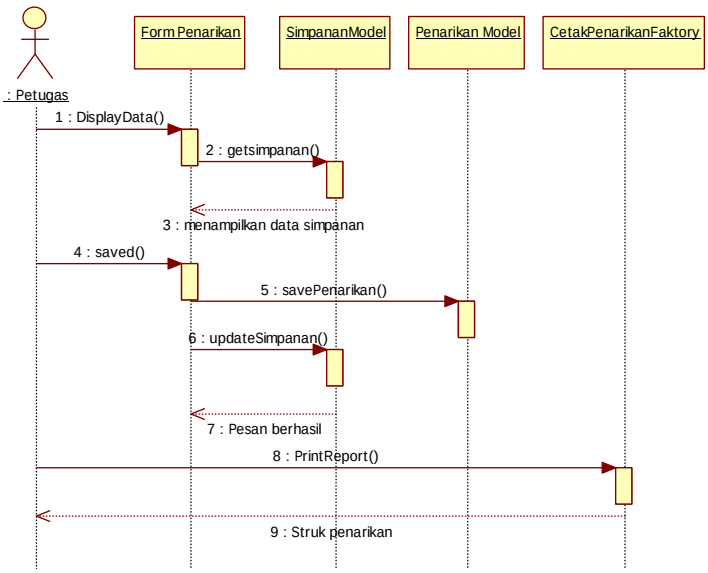
Pada *sequence* diagram pinjaman menggambarkan user untuk melakukan edit transaksi pinjaman tetapi data pinjan yang bisa diubah adalah pinjaman yang belum melakukan angsuran. Berikut gambar *sequence* diagram edit pinjaman:




Gambar 4.36. *Sequence* Diagram mengelola edit pinjaman yang diusulkan

#### 4.2.3.4.13. *Sequence* Diagram Mengelola Transaksi Penarikan Simpanan

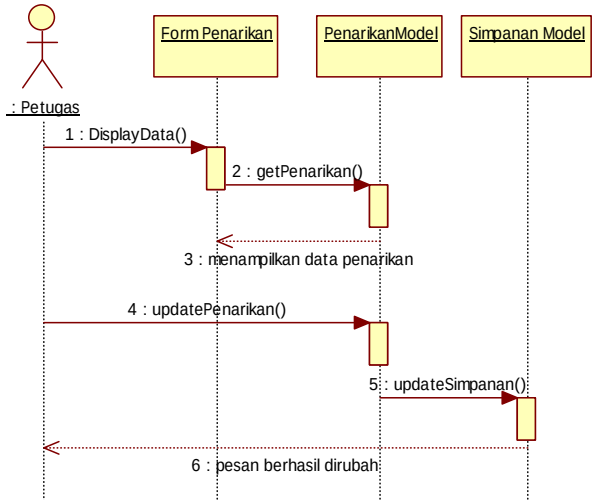
Pada *sequence* diagram transaksi penarikan simpanan menggambarkan user untuk melakukan transaksi penarikan simpanan. Berikut gambar *sequence* diagram transaksi penarikan simpanan:




Gambar 4.37. *Sequence* Diagram mengelola transaksi penarikan simpanan yang diusulkan

#### 4.2.3.4.14. *Sequence* Diagram Mengelola Edit Transaksi Penarikan Simpanan

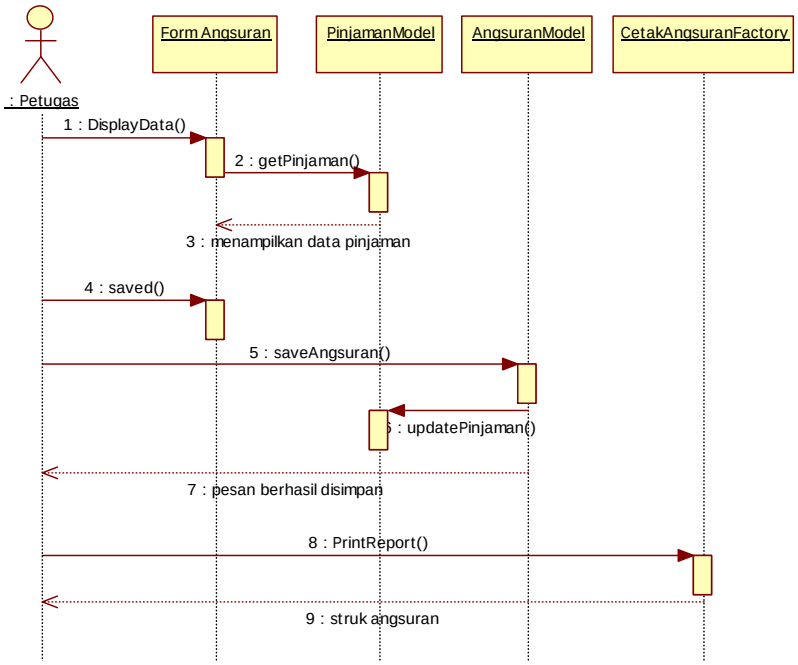
Pada *sequence* diagram eit transaksi penarikan simpanan menggambarkan admin atau petugas untuk melakukan edit transaksi penarikan simpanan. Berikut gambar *sequence* diagram edit transaksi penarikan simpanan:




Gambar 4.38. *Sequence* Diagram mengelola edit transaksi penarikan simpanan yang diusulkan

#### 4.2.3.4.15. *Sequence* Diagram Mengelola Transaksi Bayar Angsuran

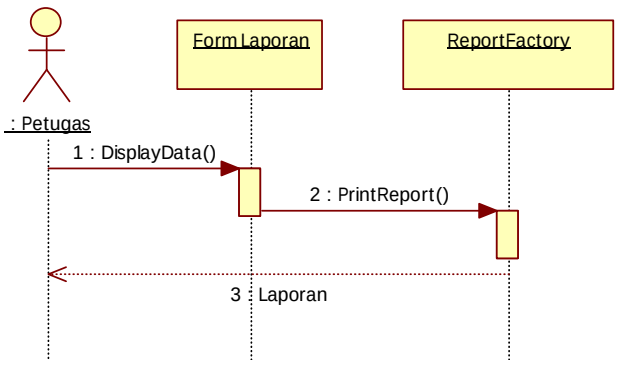
Pada *sequence* diagram mengelola transaksi bayar angsuran menggambarkan user untuk melakukan transaksi pembayaran angsuran. Berikut gambar *sequence* diagram transaksi bayar angsuran:




Gambar 4.39. *Sequence* Diagram mengelola transaksi angsuran yang diusulkan

#### 4.2.3.4.16. *Sequence* Diagram Melihat Laporan

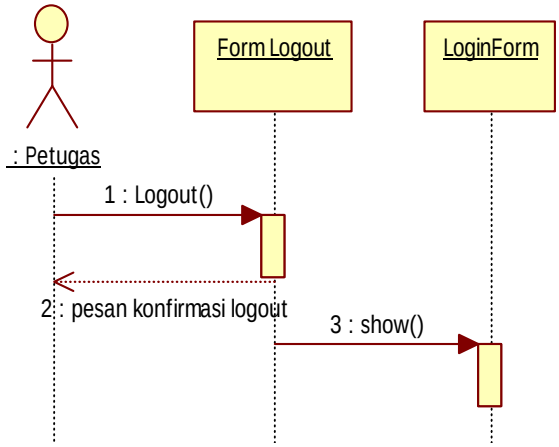
Pada *sequence* diagram melihat laporan user dapat melihat dan mencetak laporan. Berikut gambar *sequence* diagram melihat laporan:




Gambar 4.40. *Sequence* Diagram melihat laporan yang diusulkan

#### 4.2.3.4.17. *Sequence* Diagram *Logout*

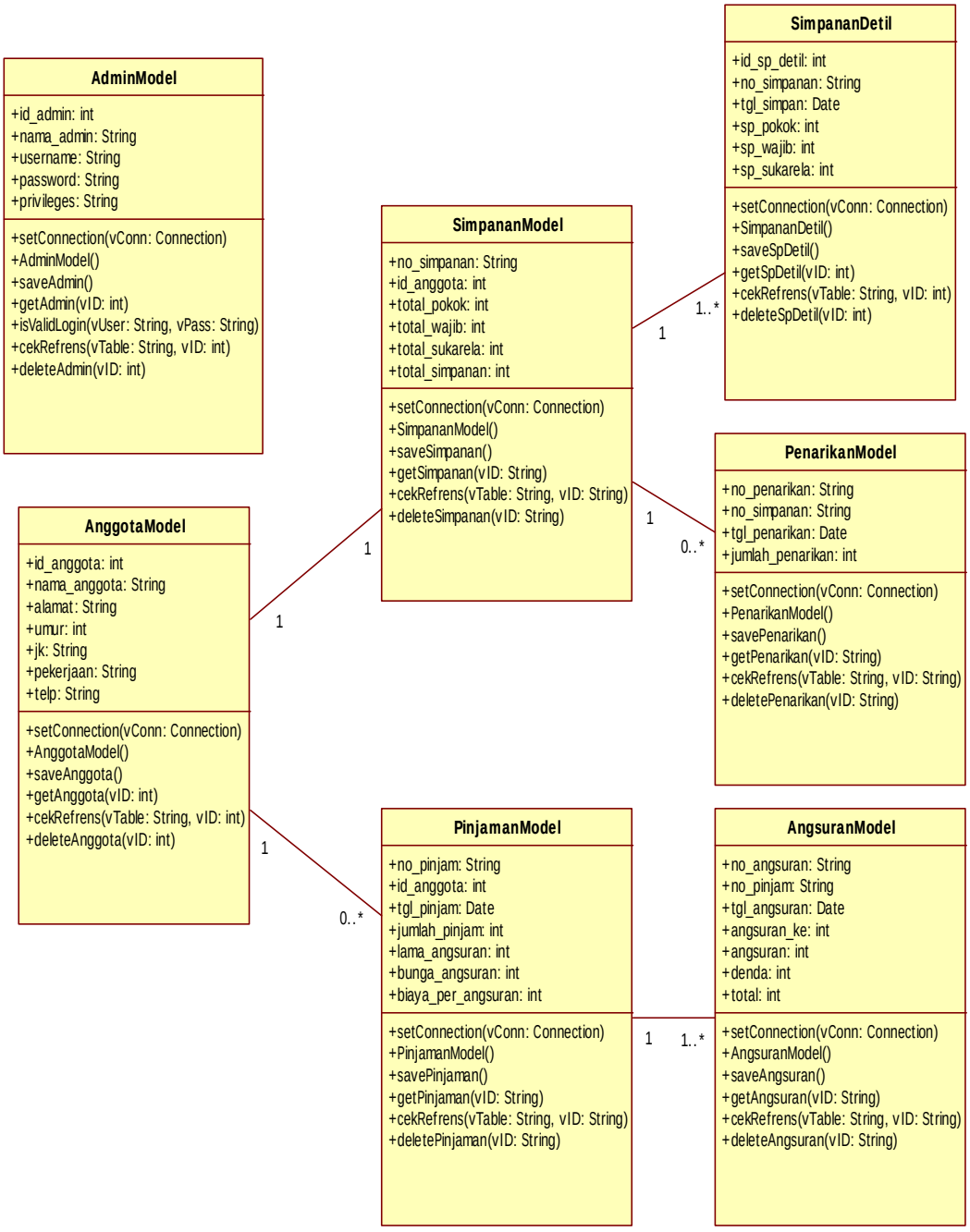
Pada *sequence* diagram *Logou*t, *user* dapat keluar dari halaman utama dan masuk kembali pada kembali pada form login. Berikut gambar *sequence* diagram melihat laporan:




Gambar 4.41. *Sequence* Diagram Logout

### 4.2.3.5. *Class* Diagram

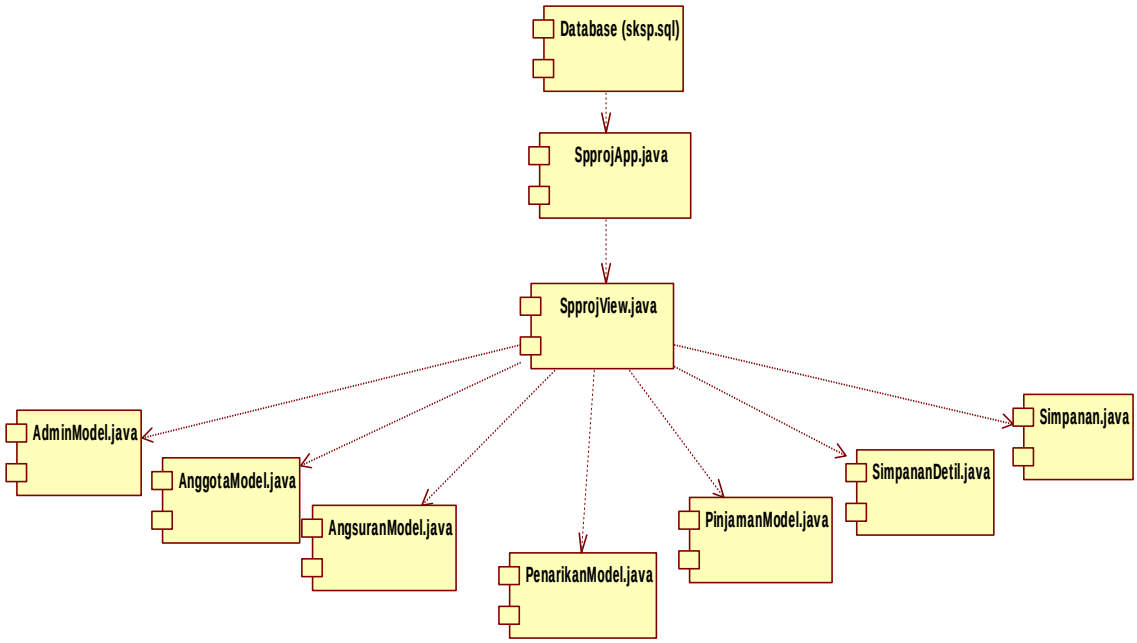
*Class* diagram membantu kita dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class* diagram memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain (dalam *logical view*) dari suatu sistem.




Gambar 4.42. *Class* diagram simpan pinjam yang diusulkan

### 4.2.3.6. *Component* Diagram

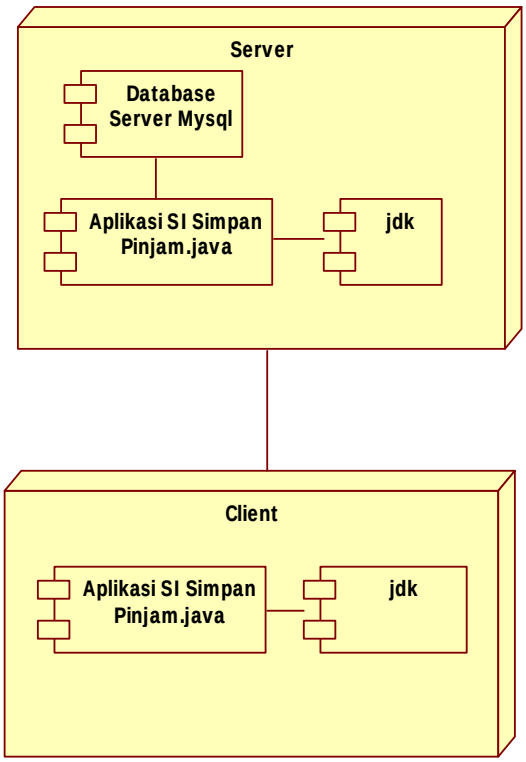
Diagram ini menampilkan komponen dalam sistem dan hubungan antar komponen tersebut. Adapun Diagram Komponen yang terbentuk adalah sebagai berikut:




Gambar 4.43. *Component* Diagram Simpan Pinjam yang diusulkan

### 4.2.3.7. *Deployment* Diagram

Diagram *deployment* menggambarkan bagaimana konfigurasi perangkat keras dimana sistem akan diinstalasikan. Adapun Diagram *deployment* yang terbentuk adalah sebagai berikut:




Gambar 4.44. *Deployment* Diagram Simpan Pinjam yang diusulkan

#### 4.2.3.8. Struktur File

Struktur file digunakan untuk merancang elemen data, yaitu panjang elemen, jenis data, serta ukuran dari data. Struktur file dari sistem informasi simpan pinjam pada koperasi pedagang Mitra Abadi adalah sebagai berikut:

**1. Tabel Anggota**

Nama File : tbl_nggota

Kunci : id

Jumlah field : 9

Panjang : 172bit

**Tabel 4.31. Tabel Anggota**

| No | Nama Field | Tipe Data | Panjang | Value |
|---|---|---|---|---|
| 1 | id | int | 3 | - |
| 2 | id_anggota | varchar | 5 | - |
| 3 | nama_anggota | varchar | 25 | - |
| 4 | alamat | varchar | 100 | - |
| 5 | umur | int | 2 | - |
| 6 | jk | enum | - | ‘p’,’l’ |
| 7 | pekerjaan | varchar | 25 | - |
| 8 | telepon | varchar | 12 | - |
| 9 | tgl_masuk | date | - | - |

**2. Tabel Admin**

Nama File : tbl_admin

Kunci : id

Jumlah field : 6

Panjang : 86bit

**Tabel 4.32. Tabel Admin**

| No | Nama Field | Tipe Data | Panjang |
|---|---|---|---|
| 1 | id | int | 3 |
| 2 | id_admin | varchar | 7 |
| 3 | nama | varchar | 25 |
| 4 | username | varchar | 25 |

**Tabel 4.32. Tabel Admin (Lanj)**

| No | Nama Field | Tipe Data | Panjang |
|---|---|---|---|
| 5 | password | varchar | 25 |
| 6 | privileges | int | 1 |

**3. Tabel Angsuran**

Nama File : tbl_angsuran

Kunci : id

Jumlah field : 9

Panjang : 24bit

**Tabel 4.33. Tabel Angsuran**

| No | Nama Field | Tipe Data | Panjang |
|---|---|---|---|
| 1 | id | int | 3 |
| 2 | no_angsuran | varchar | 5 |
| 3 | no_pinjam | varchar | 5 |
| 4 | tgl_tempo | date | - |
| 5 | tgl_angsuran | date | - |
| 6 | angsuran_ke | int | 11 |
| 7 | angsuran | Double | - |
| 8 | denda | Double | - |
| 9 | total | Double | - |

**4. Tabel Detil Simpanan**

Nama File : tbl_detil_simpanan

Kunci : id_detil_simpanan

Jumlah field : 7

Panjang : 10bit

**Tabel 4.34. Tabel Detil Simpan**

| No | Nama Field | Tipe Data | Panjang |
|---|---|---|---|
| 1 | id_detil_simpanan | int | 5 |
| 2 | no_simpanan | varchar | 5 |
| 3 | tgl_simpan | date | - |
| 4 | sp_pokok | Double | - |
| 5 | sp_wajib | Double | - |
| 6 | sp_sukarela | Double | - |
| 7 | total | Double | - |

**3. Tabel Penarikan**

Nama File : tbl_penarikan

Kunci : id

Jumlah field : 5

Panjang : 13bit

**Tabel 4.35. Tabel Penarikan**

| No | Nama Field | Tipe Data | Panjang |
|---|---|---|---|
| 1 | id | int | 3 |
| 2 | no_penarikan | varchar | 5 |
| 3 | no_simpanan | varchar | 5 |
| 4 | tgl_ambil | date | - |
| 5 | jumlah | Double | - |

**4. Tabel Pinjaman**

Nama File : tbl_pinjaman

Kunci : id

Jumlah field : 14

Panjang : 17bit

**Tabel 4.36. Tabel Pinjaman**

| No | Nama Field | Tipe Data | Panjang | Value |
|---|---|---|---|---|
| 1 | id | int | 3 | - |
| 2 | no_ pinjaman | varchar | 5 | - |
| 3 | id_anggota | varchar | 5 | - |
| 4 | tgl_pinjam | date | - | - |
| 5 | jml_pinjam | Double | - | - |
| 6 | lama_angsuran | int | 2 | - |
| 7 | tipe | enum | - | ‘m’,’b’ |
| 8 | jatuh_tempo | int | 2 | - |

**Tabel 4.36. Tabel Pinjaman (Lanj)**

| No | Nama Field | Tipe Data | Panjang | Value |
|---|---|---|---|---|
| 9 | pokok_angsuran | Double | - | - |
| 10 | bunga_angsuran | Double | - | - |
| 11 | total_ angsuran | Double | - | - |
| 12 | angsuran_masuk | Double | - | - |
| 13 | sisa_ angsuran | Double | - | - |
| 14 | keterangan | enum | - | ‘0’,’1’ |

**5. Tabel Simpanan**

Nama File : tbl_ simpanan

Kunci : id

Jumlah field : 7

Panjang : 13bit

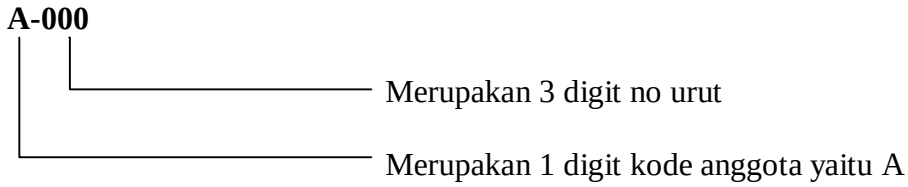
**Tabel 4.37. Tabel Simpanan**

| No | Nama Field | Tipe Data | Panjang |
|---|---|---|---|
| 1 | id | int | 3 |
| 2 | no_simpanan | varchar | 5 |
| 3 | id_anggota | varchar | 5 |
| 4 | total_pokok | Double | - |
| 5 | total _wajib | Double | - |
| 6 | total _sukarela | Double | - |
| 7 | total_simpanan | Double | - |

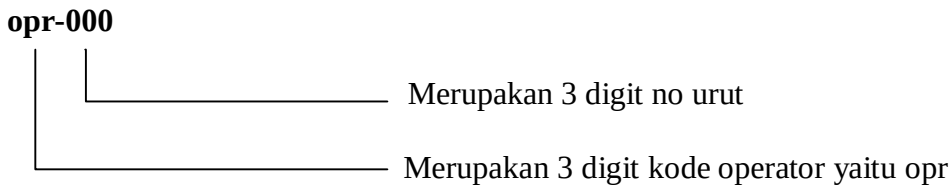
#### 4.2.3.9. Kodifikasi

Kodifikasi atau sistem pengkodean digunakan sebagai alat bantu pengolahan data yang dapat mempermudah dalam pemasukan dan pencarian data. Adapun beberapa pengkodean yang digunakan dalam pembuatan sistem ini, antara lain sebagai berikut:

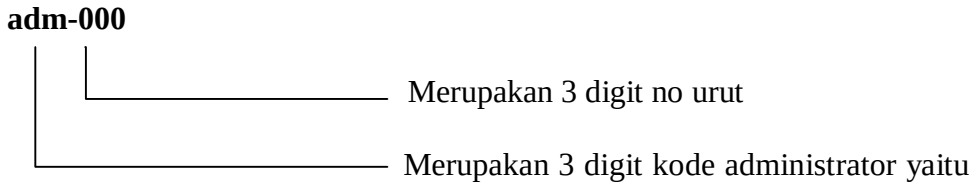
1. **ID Anggota**



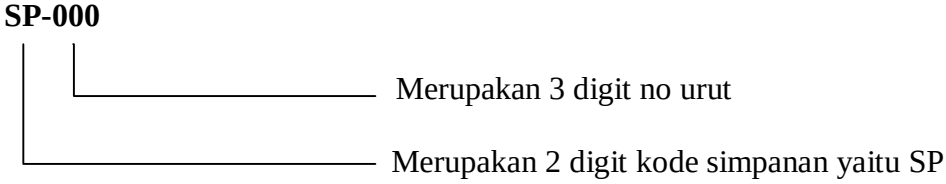

2. **ID Petugas**




3. **ID Administrator**

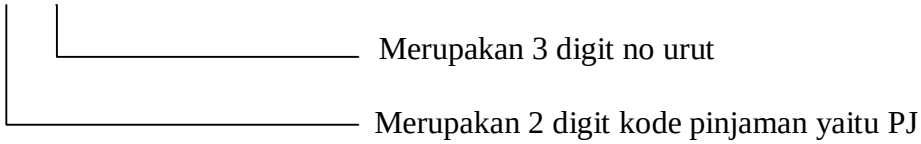



4. **No Simpanan**



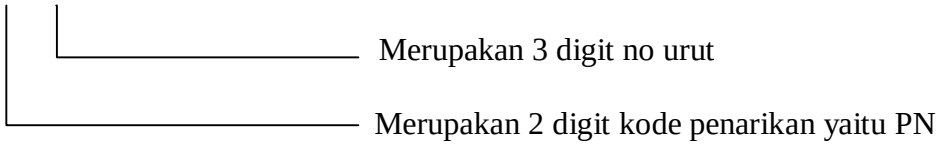
**5. No Pinjaman**

**PJ-000**



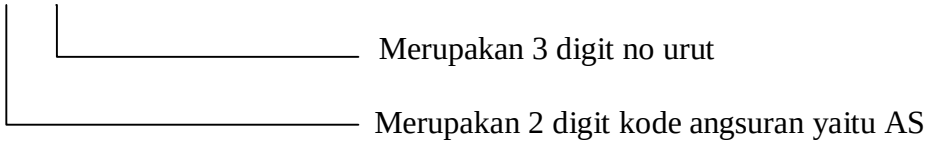

**6. No Penarikan**

**PN-000**




**7. No Angsuran**

**AS-000**

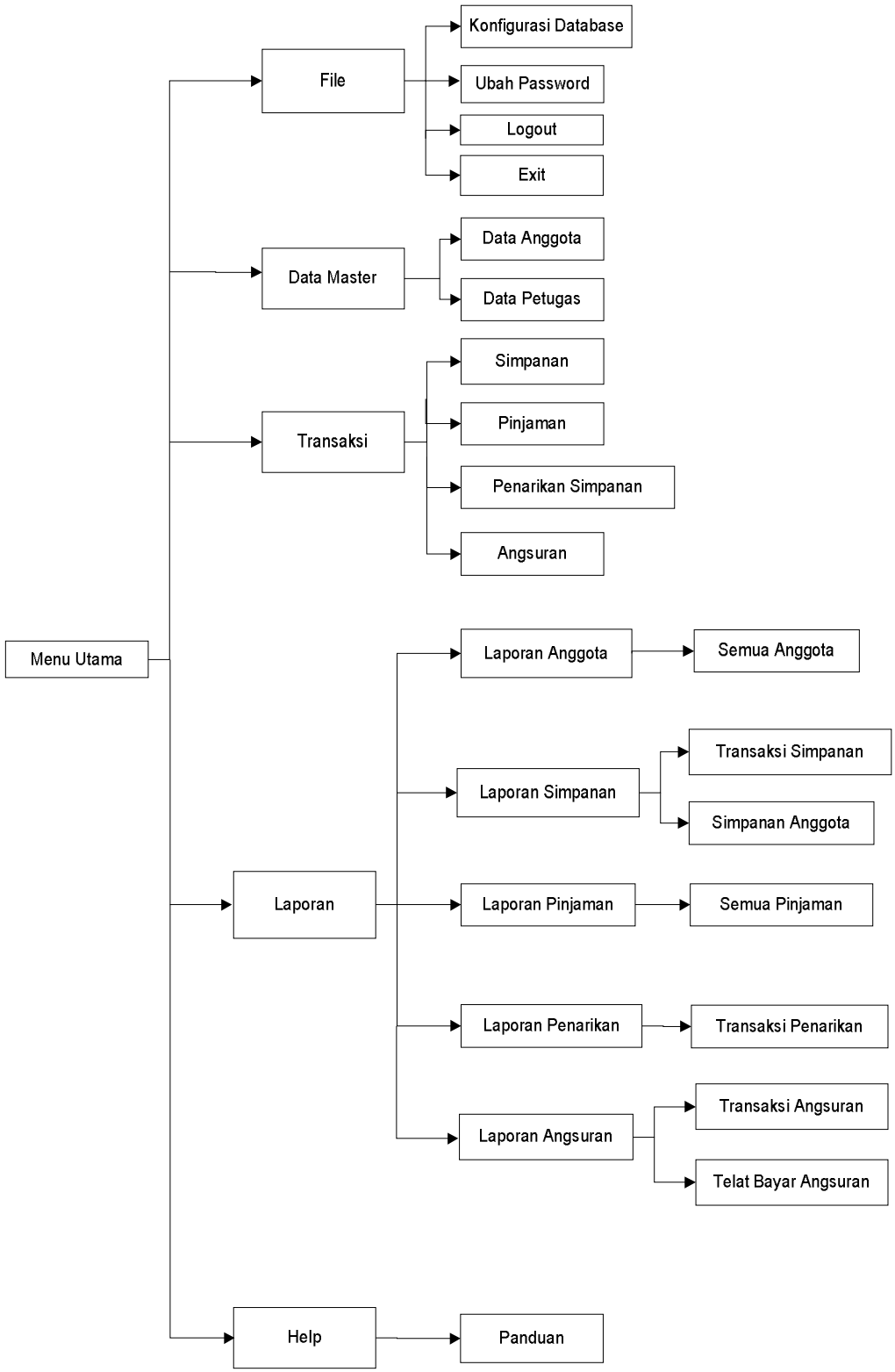



### 4.2.4. Perancangan Antar Muka

Pada perancangan antar muka atau *interface* ini akan menggambarkan rancangan tampilan apa saja yang ada dalam sistem ini. Adapun yang akan dibahas yaitu struktur menu, perancangan *input* dan perancangan *output*.

#### 4.2.4.1. Struktur Menu

Struktur menu menjelaskan bentuk rancangan menu dari *software* yang akan dibuat. Berikut ini adalah rancangan struktur menu :

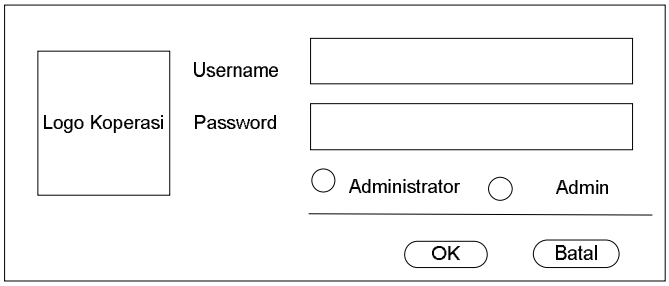



Gambar 4.45. Struktur Menu

### 4.2.4.2. Perancangan *Input*

Perancangan *input* merupakan perancangan desain untuk masukan yang akan diproses oleh *software*.

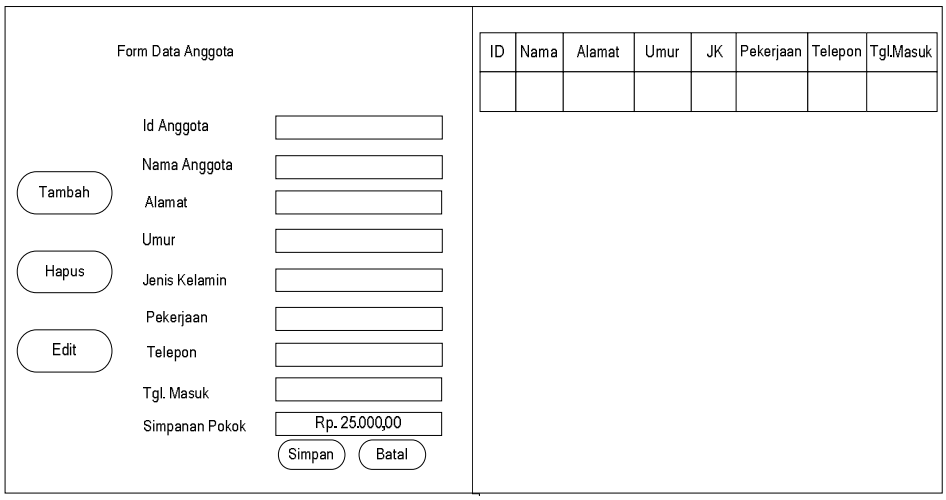
1. **Form Login**




Gambar 4.46. Rancangan Form Login

Form login dijalankan ketika aplikasi dijalankan. Form ini berfungsi sebagai penghubung ke menu utama dan juga sebagai pengatur hak akses dari operator dan admin, dengan memasukan *username*, *password* dan memilih akses mana yang akan akan dipilih.

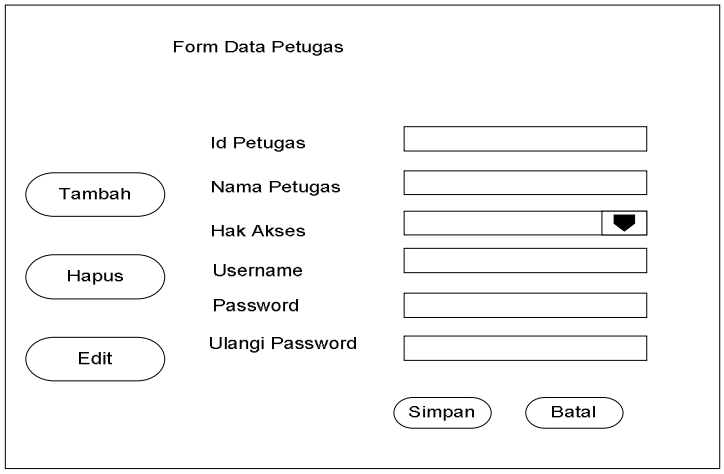
2. **Form Data Anggota**




Gambar 4.47. Rancangan Form Anggota

Form data anggota adalah form untuk menambah data anggota, menghapus dan mengedit data anggota dengan cara mencari terlebih dahulu data yang akan diedit.

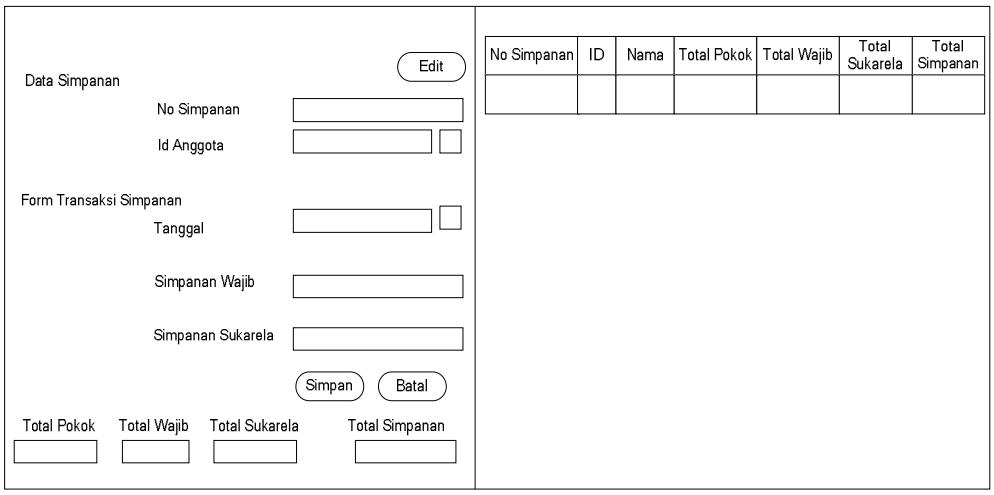
### 3. Form Data Petugas




Gambar 4.48. Rancangan Form Data Petugas

Form data petugas adalah form yang dapat menambah, mengubah dan menghapus data petugas atau admin.

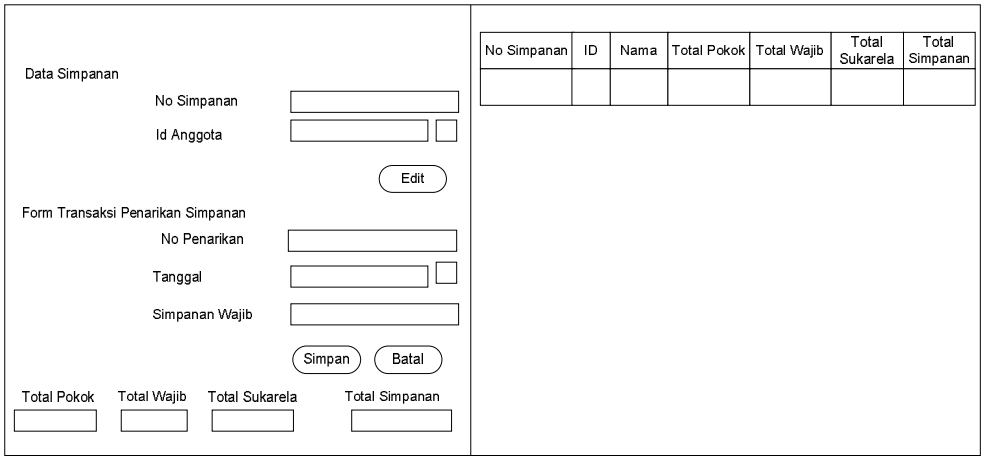
### 4. Form Simpanan




Gambar 4.49. Rancangan Form Simpanan

Form simpanan ini adalah form yang mengelola transaksi simpanan anggota baik simpanan pokok, simpanan wajib dan simpanan sukarela.

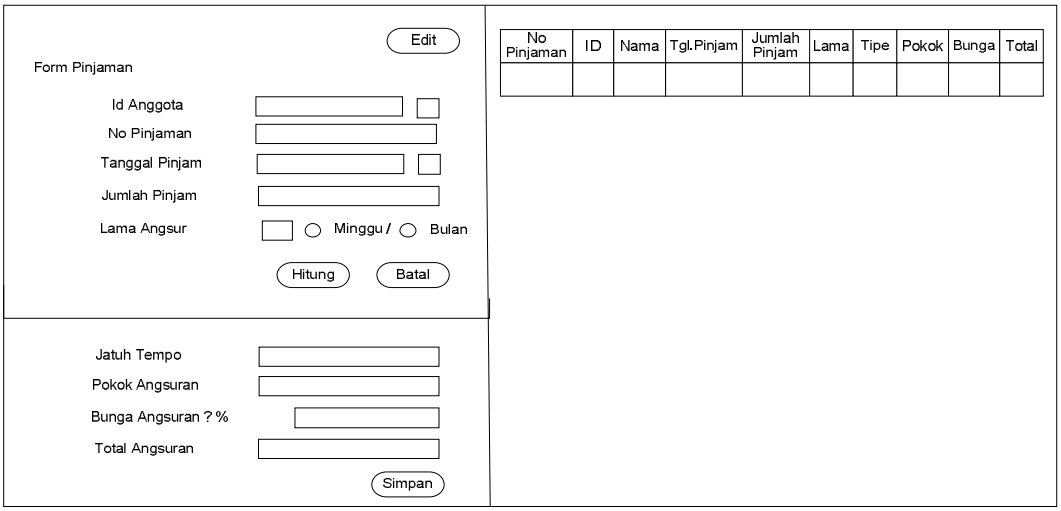
## 5. Form Form Penarikan Simpanan




Gambar 4.50. Rancangan Form Penarikan Simpanan

Form penarikan simpanan ini adalah form yang mengelola transaksi penarikan simpanan anggota baik simpanan sukarela.

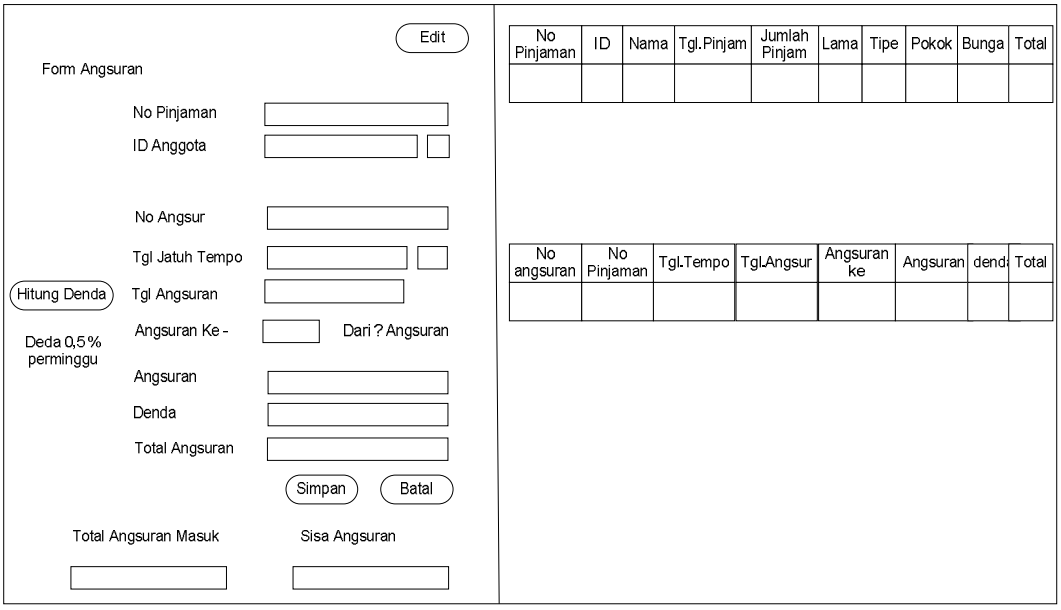
## 6. Form Pinjaman




Gambar 4.51. Rancangan Form Pinjaman

Form Pinjaman ini adalah form yang mengelola transaksi peminjaman.

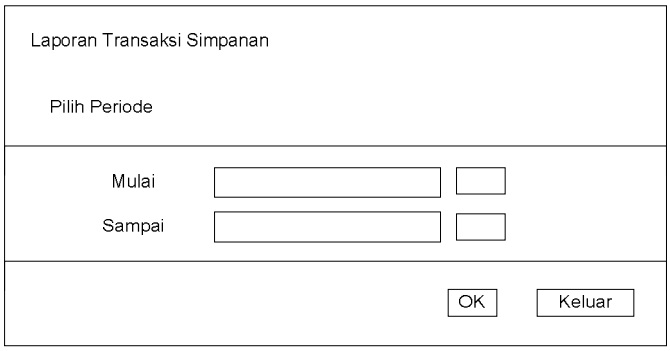
## 7. Form Angsuran




Gambar 4.52. Rancangan Form Angsuran

Form angsuran ini adalah form yang mengelola angsuran pinjaman anggota baik angsuran mingguan ataupun bulanan dan juga terdapat denda keterlambatan sebesar 0,5% perminggunya.

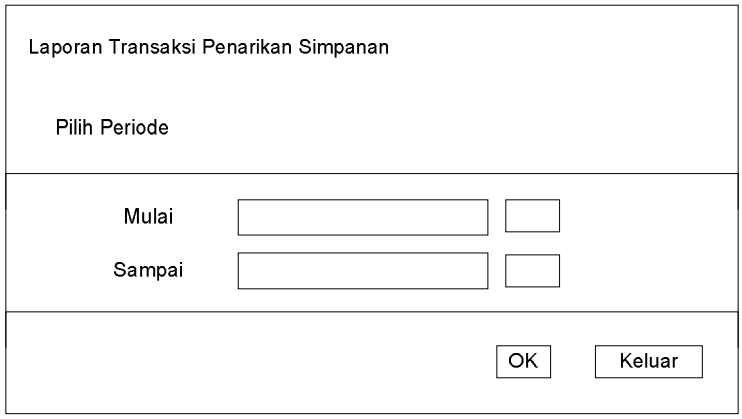
## 8. Form Pilih Periode Laporan Transaksi Simpanan




Gambar 4.53. Rancangan Form Pilih Periode Laporan Transaksi Simpanan

Form ini digunakan untuk mencetak laporan transaksi simpanan berdasarkan per periode.

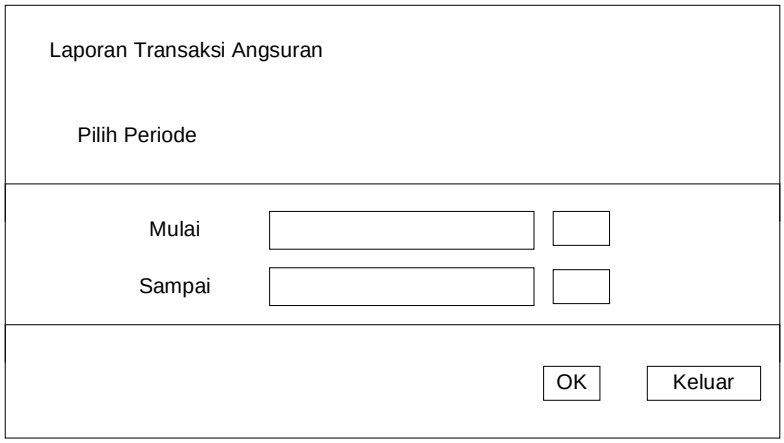
**9. Form Pilih Periode Laporan Transaksi Penarikan Simpanan**




Gambar 4.54. Rancangan Form Pilih Periode Laporan Transaksi penarikan Simpanan

Form ini digunakan untuk mencetak laporan transaksi penarikan simpanan berdasarkan per periode.

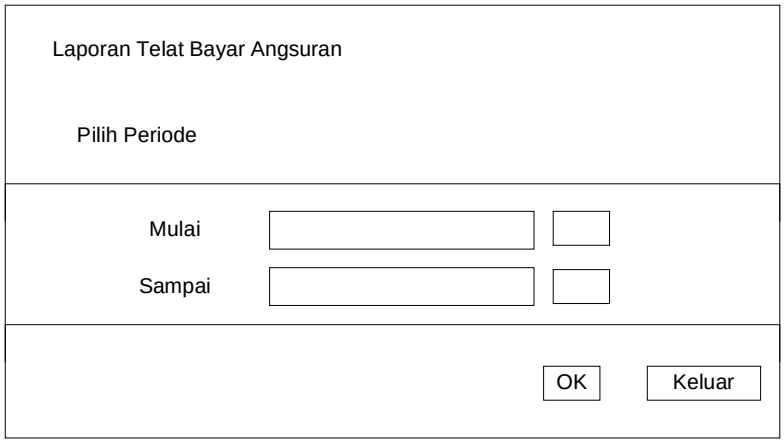
**10. Form Pilih Periode Laporan Transaksi Angsuran**




Gambar 4.55. Rancangan Form Pilih Periode Laporan Transaksi Angsuran

Form ini digunakan untuk mencetak laporan transaksi angsuran berdasarkan per periode.

**11. Form Pilih Periode Laporan Telat Bayar Angsuran**




Gambar 4.56. Rancangan Form Pilih Periode Laporan Telat Bayar Angsuran

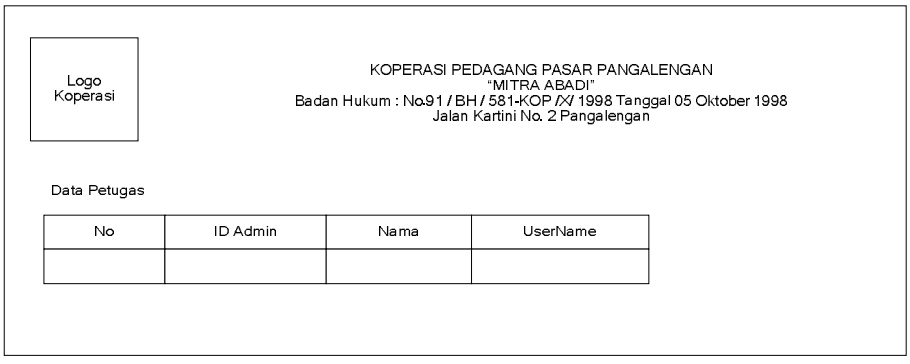
Form ini digunakan untuk mencetak laporan yang telat membayar angsuran berdasarkan per periode.

### 4.2.4.3. Peracangan *Output*

Perancangan *output* merupakan rancangan tampilan informasi yang dihasilkan dari aplikasi. Format laporan bisa dicetak ataupun hanyan ditampilkan dimonitor saja.

**1. Laporan Data Petugas**

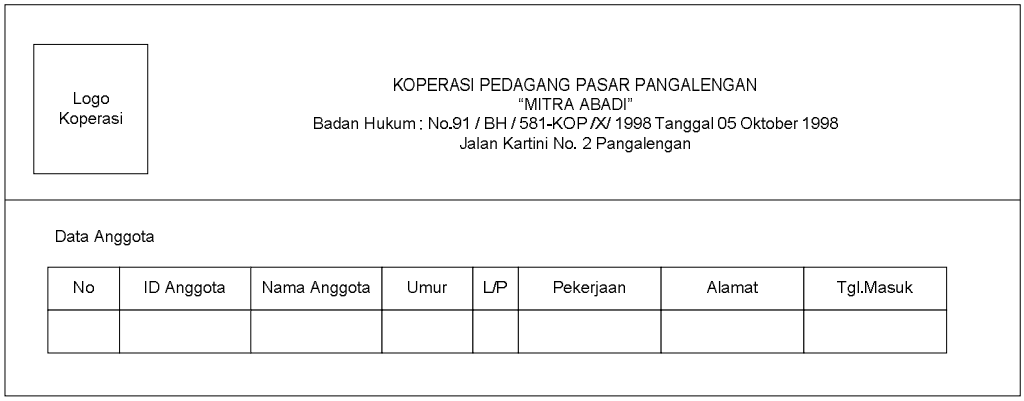
Perancangan ini untuk laporan data petugas yang dicetak oleh admin.




Gambar 4.57. Laporan Data Petugas

## 2. Laporan Data Anggota

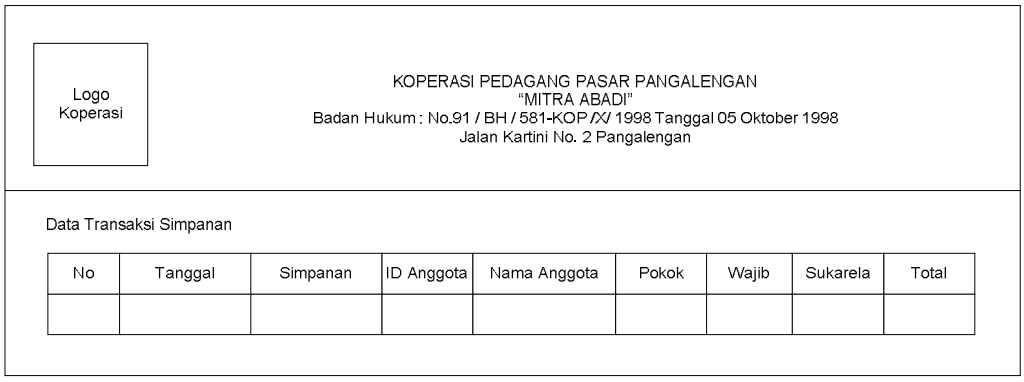
Perancangan ini untuk laporan data anggota yang dicetak oleh admin atau operator.




Gambar 4.58. Laporan Data Anggota

## 3. Laporan Data Transaksi Simpanan

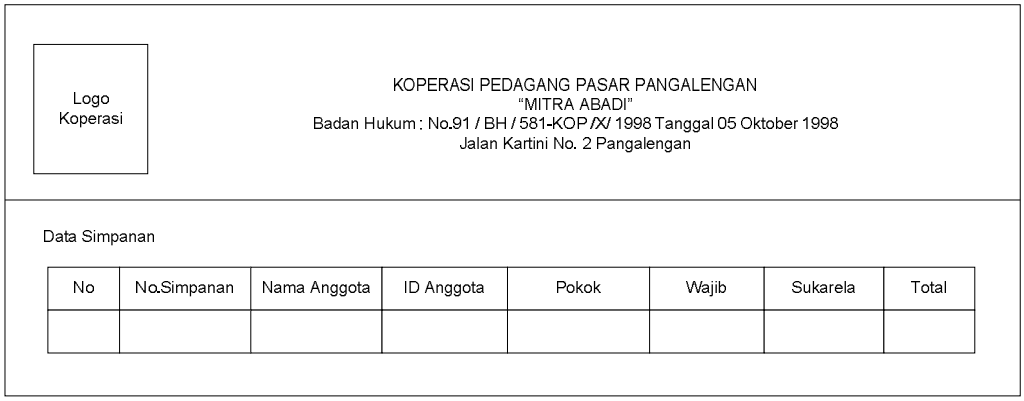
Perancangan ini untuk laporan data transaksi simpanan per periode yang dicetak oleh admin atau operator.




Gambar 4.59. Laporan Data Transaksi Simpanan

### 4. Laporan Data Simpanan Anggota

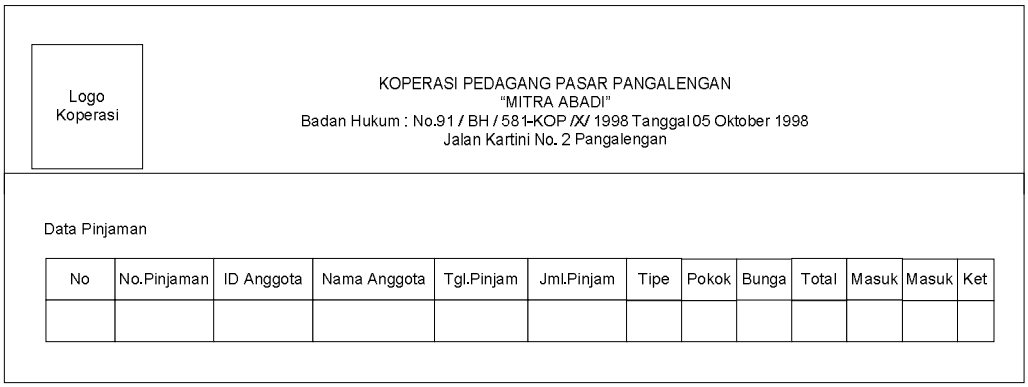
Perancangan ini untuk laporan data transaksi simpanan seluruh anggota yang dicetak oleh admin atau operator.




Gambar 4.60. Laporan Data Simpanan Anggota

### 5. Laporan Data Pinjaman

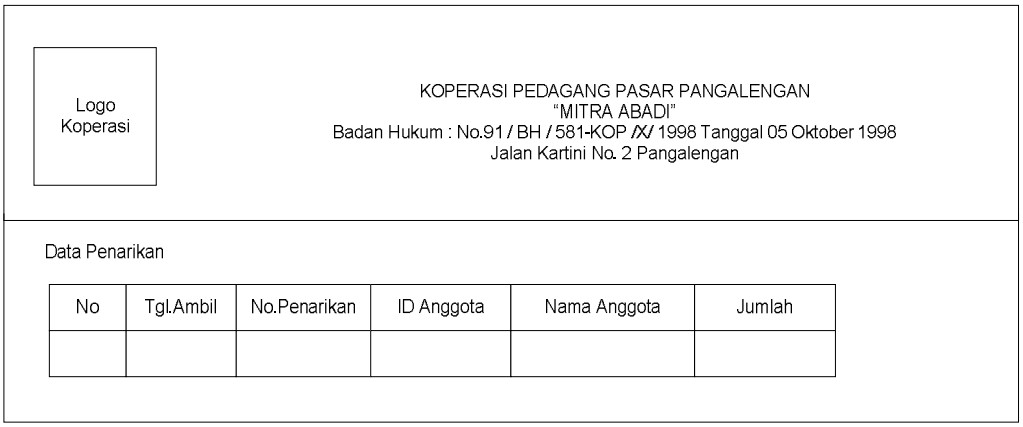
Perancangan ini untuk laporan data pinjaman anggota yang dicetak oleh admin atau operator.




Gambar 4.61. Laporan Data Pinjaman

### 6. Laporan Data Penarikan Simpanan

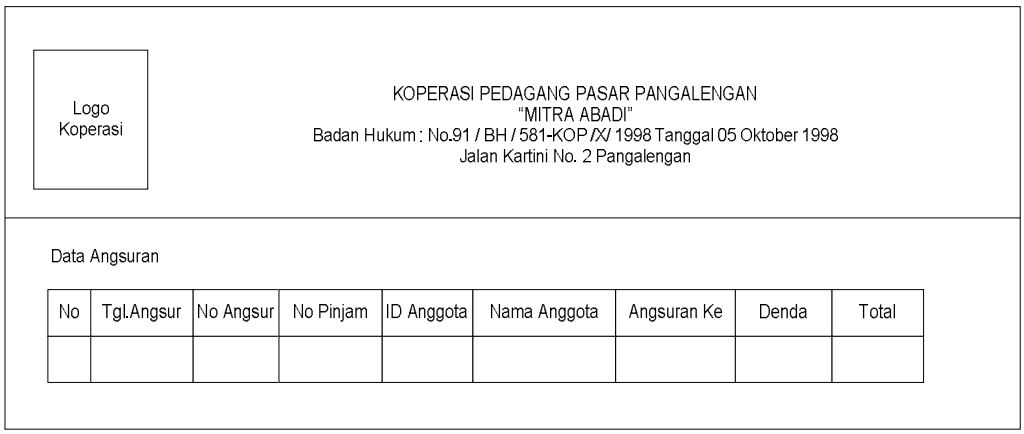
Perancangan ini untuk laporan data penarikan simpanan per periode yang dicetak oleh admin atau operator.




Gambar 4.62. Laporan Data Penarikan

### 7. Laporan Data Angsuran

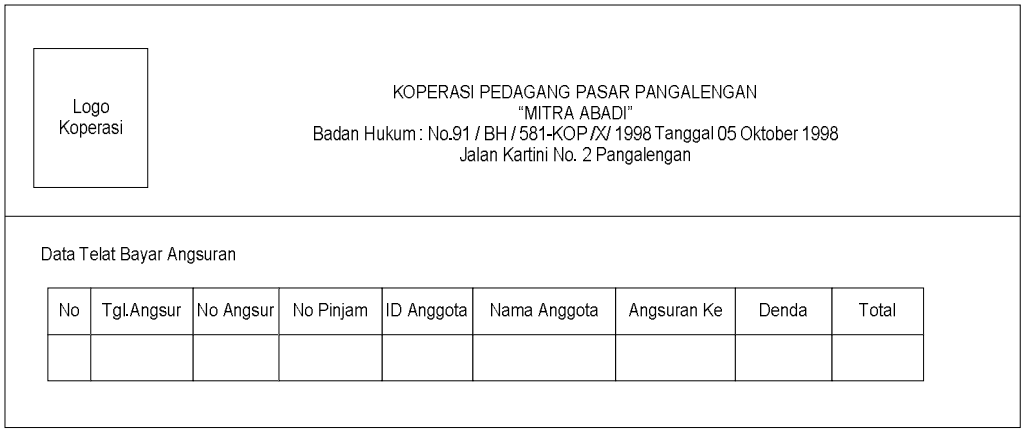
Perancangan ini untuk laporan data angsuran per periode yang dicetak oleh admin atau operator.




Gambar 4.63. Laporan Data Angsuran

### 8. Laporan Data Telat Bayar Angsuran

Perancangan ini untuk laporan data telat bayar angsuran per periode yang dicetak oleh admin atau operator.

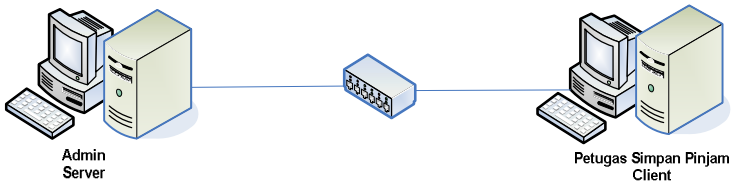



Gambar 4.64. Laporan Data Telat Bayar Angsuran

### 4.2.5. Perancangan Arsitektur Jaringan

Arsitektur jaringan yang diterapkan pada Koperasi Pedagang Mitra Abadi Pangalengan adalah dengan menggunakan topologi *star*. Topologi ini dipilih karena memiliki beberapa keuntungan sebagai berikut :

1. Paling unggul diantara topologi yang lain.
2. Mudah dikembangkan.
3. Kemudahan akses ke jaringan LAN lain.




Gambar 4.65. Arsitektur Jaringan