# BAB III

# PEMBAHASAN

## 3.1. Kegiatan Kerja Praktek

Kerja praktek ini dilaksanakan lebih kurang selama selama 29 hari ( 1 bulan 1 minggu ), terhitung mulai sejak minggu terakhir bulan oktober hingga akhir bulan November di PT. Informatika Reka Mandiri ( IRM ). Kami ditempatkan di bagian tim Dokumentasi Project, dimana bertugas mendokumentasikan program yang dibuat kedalam sebuah dokumen dalam bentuk user guide untuk client, membuat dokumen UAT ( User Acceptance Test ) yang digunakan untuk system pengujian aplikasi yang dibuat serta melakukan testing aplikasi. Selain itu kegiatan kerja praktek lainnya adalah membangun aplikasi koperasi untuk koperasi yang berada di PT. Informatika Reka Mandiri tersebut. Koperasi pada PT. IRM ini yang masih dijalankan secara konvensional, maka kami mencoba menganalisis kegiatan koperasi tersebut. Lalu melakukan wawancara dengan petugas koperasi dan akhirnya, memutuskan untuk membuat aplikasi koperasi simpan pinjam di PT. Informatika Reka Mandiri sebagai laporan kerja praktek. Namun kegiatan atau tugas kerja praktek lainnya yang kami dapatkan tetap dijalankan dengan sebaik mungkin.

Di dalam proses kehadiran kami diharuskan hadir setiap hari dihari kerja dengan waktu jam masuk dan jam keluar sama dengan jadwal karyawan di PT. Informatika Reka Mandiri. Dan melakukan tatap muka dengan pembimbing lapangan hampir setiap hari. Itupun dilakukan sesuai kebutuhan, dengan kata lain bersifat fleksibel.

Adapun rangkaian kegiatan kerja yang kami lakukan selama kerja praktek di PT. Informatika Reka Mandiri antara lain:

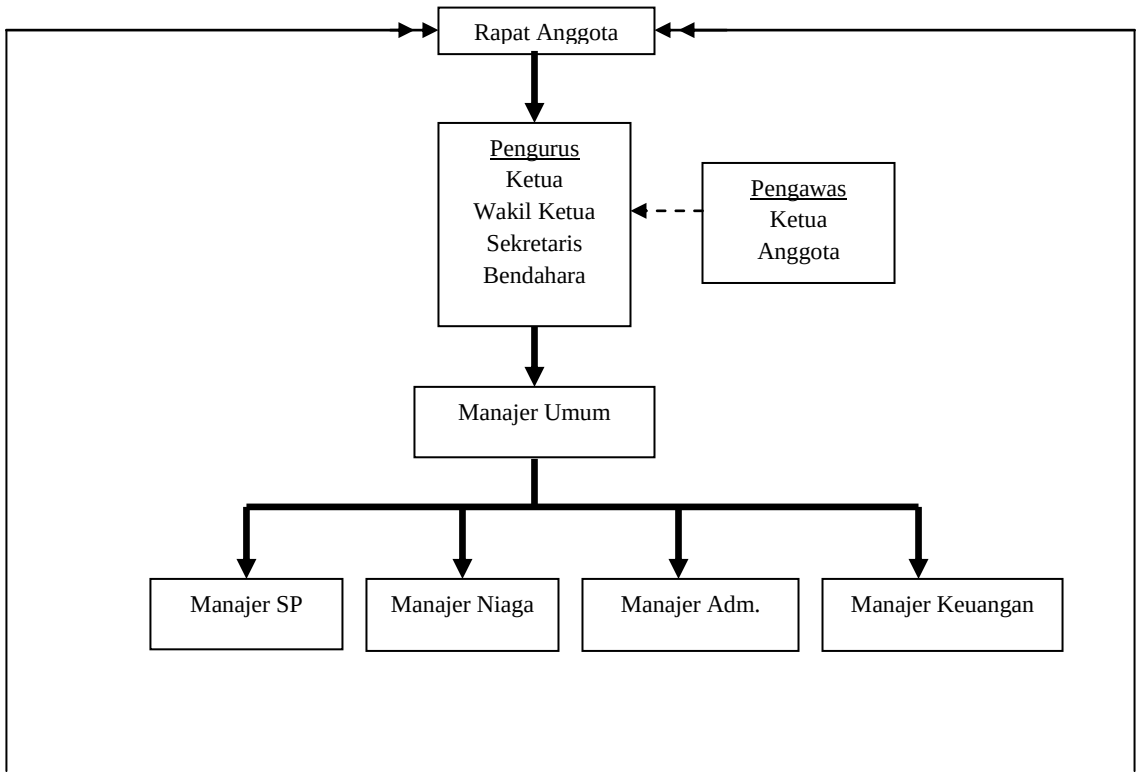
| Nama kegiatan | Bulan | | | | | |
|---|---|---|---|---|---|---|
| | Oktober | | November | | | |
| | m-3 | m-4 | m-1 | m-2 | m-3 | m-4 |

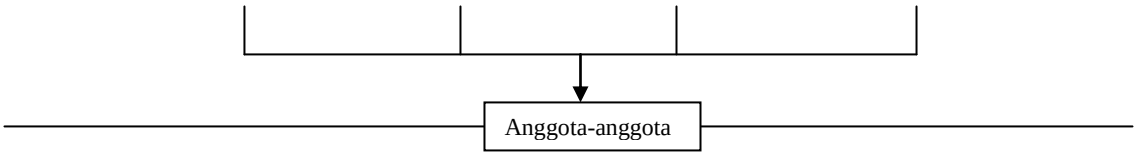
| | | | | | | |
|---|---|---|---|---|---|---|
| Wawancara langsung terkait tugas Kerja Praktek di PT. IRM | | | | | | |
| Pengumpulan data dan analisis tugas tim dokumentasi dengan tim yang lain. | | | | | | |
| Menjalankan tugas kerja praktek yaitu membuat dokumen user guide untuk para client dari beberapa aplikasi yang dibuat oleh tim IRM | | | | | | |
| Menganalisis kegiatan di PT. IRM yang belum dibuat aplikasinya. | | | | | | |
| Wawancara dengan pengurus koperasi, karena kami akan mengangkat tema koperasi simpan pinja sebagai judul kerja praktek di IRM | | | | | | |
| Pembuatan dokumen UAT untuk persiapan testing aplikasi | | | | | | |
| Testing aplikasi terhadap perangkat lunak yang dibangun. | | | | | | |
| Revisi seluruh dokumen sesuai update aplikasi dari programmer | | | | | | |
| Mulai merancang program KP yang diambil yaitu membuat aplikasi koperasi simpan pinjam PT. IRM | | | | | | |

*Tabel 1 Kegiatan Kerja Praktek*

### 3.1.1. Struktur Organisasi Koperasi Informatika Reka Mandiri

Struktur organisasi Koperasi Mandiri di PT Informatika Reka Mandiri dapat dilihat pada gambar di bawah ini :






*Gambar 1 Struktur Organisasi*

Keterangan :

: Garis Komando

: Garis Pengawasan

: Garis Pelayanan

: Garis Kepemilikan

Adapun beberapa deskripsi tugas dari struktur organisasi yang terlibat dalam Koperasi Mandiri yaitu diuraikan sebagai berikut :

**1. Rapat Anggota**

Rapat anggota adalah pertemuan para pemilik koperasi yang diselenggarakan secara demokratis dan merupakan kekuasaan tertinggi dalam koperasi. Rapat anggota dilaksanakan minimal satu kali dalam setahun. Rapat angota membahas dan memutuskan antara lain : program kerja, rancangan anggaran belanja koperasi, peraturan-peraturan, masa deepan koperasi, laporan pertanggungjawaban pengurus dan pengawas, pemilihan pengurus dan pengawas, pengesahan AD dan ART serta peraturan khusus lainnya.

**2. Pengawas**

Pengawas dipilih dari dan oleh anggota dalam Rapat Anggota. Tugasnya adalah melakukan pengawasan terhadap pelaksanaan terhadap pelaksanaan seluruh keputusan Rapat Anggota yang dilakukan oleh pengurus. Pengawas bertanggung jawab kepada Rapat Anggota.

**3. Pengurus**

Pengurus dipilih dari dan oleh anggota dalam Rapat Anggota. Tugasnya adalah memimpin jalannya organisasi dan usaha koperasi dengan berpedoman pada keputusan Rapat Anggota. Pengurus bertanggung jawab kepada Rapat

Anggota dan bertindak atas nama koperasi di muka hukum. Dalam mengelola usaha koperasi, pengurus dapat mengangkat manajer yang professional.

**4. Manajer**

Manajer diangkat oleh pengurus. Rencana pengangkatan diajukan kepada Rapat Anggota untuk mendapatkan persetujuan. Manajer diberi wewenang dan kuasa untuk mengelola usaha koperasi dalam rangka melayani anggota. Dalam pekerjaannya, manajer bertanggung jawab kepada pengurus.

Untuk menjalankan usaha koperasi, manajer dapat mengangkat karyawan atas dasar persetujuan pengurus. Manajer dan Karyawan menangani kegiatan operasional dan usaha koperasi, seperti pelaksana pengelola usaha sehari-hari, membuat laporan periodic, merancang anggaran pendapatan dan belanja, serta merundingkan dengan pengurus mengenai program dan kebijakan baru.

**5. Komite**

Pengurus dengan persetujuan Rapat Anggota dapat membentuk komite (panitia). Komite ini bertugas untuk melaksanakan kegiatan-kegiatan berdasarkan kebutuhan koperasi. Komite berasal dari anggota-anggota aktif koperasi yang memiliki kemampuan khusus di bidangnya masing-masing.

a. Komite audit. Tugasnya melakukan audit internal terhadap kekayaan koperasi
b. Komite pemilihan. Bertugas mempersiapkan, menangani dan mengawasi proses pemilihan pengurus dan pengawas, serta mengumumkan hasilnya kepada anggota.
c. Komite kredit. Bertugas melaksanakan kegiatan yang menyangkut pelayanan pinjaman bagi anggota. Misalnya melakukan penilaian terhadap kelayakan pinjaman yang diajukan anggota.
d. Komite diklat. Bertugas mengelola pendidikan dan pelatihan (Diklat) anggota, pengurus dan pengawas koperasi.
e. Komite penyempurnaan AD dan ART dalam upaya mengikuti perkembangan yang terjadi, maka sangat dimungkinkan untuk menyempurnakan AD dan ART.

## 3.2. Analisis Sistem

Analisis adalah penguraian dari suatu masalah atau objek yang akhirnya menghasilkan suatu kesimpulan, hal ini dimaksudkan untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan.

### 3.2.1. Analisis Masalah

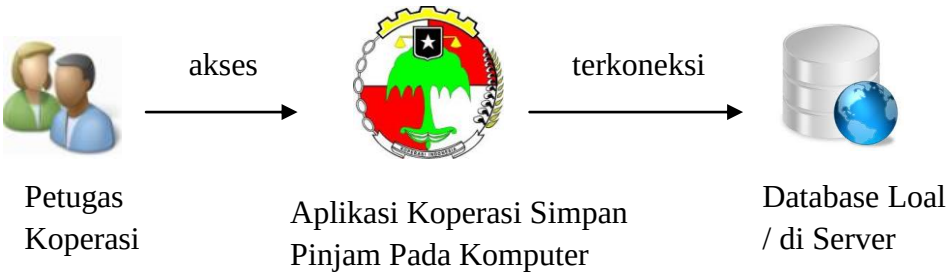
Masalah yang ada di PT.Informatika Reka Mandiri yaitu belum adanya suatu sistem aplikasi yang dapat membantu mempermudah berjalannya system koperasi di PT. Informatika Reka Mandiri tersebut, sehingga pengurus koperasi masih kesulitan dalam melakukan update data mengenai jumlah simpanan, pinjaman / piutang dan input data barang koperasi dengan system yang masih konvensional, untuk memecahkan permasalah tersebut akan dibuat aplikasi *Koperasi* untuk mempermudah update atau input data pengguna koperasi Informatika Reka Mandiri.

### 3.2.2. Analisis Sistem yang Sedang Berjalan

Analisis prosedur pada sistem yang sedang berjalan bertujuan untuk mengetahui lebih jelas bagaimana cara kerja sistem tersebut, sehingga kelebihan dan kekurangan sistem dapat diketahui.

### 3.2.3. Analisis Sistem Yang Akan Dibangun

Sistem yang akan dibangun yaitu aplikasi *Koperasi Simpan Pinjam Reka Mandiri*. Proses analisis *Koperasi Simpan Pinjam* bertujuan untuk mengetahui proses simpan pinjam yang terjadi pada koperasi Reka Mandiri, alur aplikasi *Koperasi Simpan Pinjam Reka mandiri* yang akan dibangun dapat dilihat pada gambar dibawah ini :




*Gambar 2 Arsitektur Sistem*

### 3.2.4. Analisis Prosedur Yang Sedang Berjalan

Analisis prosedur sangat diperlukan untuk mempermudah pembuatan perancangan sistem, karena dengan menganalisis mulai dari unit terkecil sampai dengan keseluruhan, Berdasarkan metode analisis yang diambil dan digunakan, maka langkah pertama yang dilakukan adalah menentukan kebutuhan dari pengguna dengan cara menganalisis sistem yang sedang berjalan, kemudian dievaluasi. untuk meningkatkan pelayanan terhadap simpan pinjam, untuk lebih jelasnya akan diuraikan sebagai berikut :

Analisis Prosedur Peminjaman adalah sebagai berikut :

1. Anggota yang belum menjadi anggota, mengisi formulir keanggotaan terlebih dahulu.
2. Formulir dan persyaratan di berikan kepada pengurus koperasi untuk memeriksa kelengkapan, apabila formulir dan kelengkaan sudah lengkap maka, formulir dan persyaratan di berikan kepada bendahara, dan apabila formulir dan persyatratan kurang lengkap maka pengurus koperasi akan memberkan kembali formlir dan persyaratan kepada anggota untuk di perbaiki/dilengkapi.
3. Apabila formulir dan persyaratan sudah lengkap, pengurus koperasi membuat kartu anggota dan diberikan kepada anggota.
4. Pengurus koperasi memberikan form pengajuan peminjaman kepada anggota.

5. Anggota menerima form pengajuan peminjaman dari pengurus koperasi dan mengisi form pengajuan peminjaman, dan diberikan kembali kepada pengurus koperasi.
6. Pengurus koperasi melakukan pengolahan form pengajuan pinjaman, jika tidak lengkap maka form tersebut di berikan lagi kepada anggota untuk di perbaiki.
7. Pengurus koperasi memberikan form pangajuan pinjaman kepada bendahara untuk di ACC dan melakukan pemeriksaan dan melakukan pengaktifan pemotongan gaji perbulan.
8. Pembayaran dilakukan perbulan, bendahara membuat laporan gaji anggota dan laporan keuangan usaha simpan pinjam (usp) bagi anggota yang melakukan peminjaman.
9. Kwitansi dan Dana cair merupakan bukti penerimaan kas dan input data pembayaran pinjaman anggota jika dana kurang maka bendahara melakukan update data pembayaran dan melakukan penambahan ke angsuran berikutnya.
10. Bendahara membuat laporan keuangan usp perbulan yang diserahkan kepada pimpinan koperasi untuk di arsipkan.

### 3.2.5. Analisis kebutuhan pengguna

Analisis kebutuhan pengguna,bertujuan untuk mengetahui pengguna yang akan berinteraksi atau mengoperasikan aplikasi sistem informasi data bantuan.pengguna yang terlibat adalah user.user akan bertanggung jawab dalam penginputan data tersebut adalah pegawai yang memiliki wewenang sepenuhnya dalam penginputan data bantuan tersebut.

### 3.2.6. Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional menggambarkan kebutuhan sistem yang menitik-beratkan pada properti perilaku yang dimiliki oleh sistem, diantaranya kebutuhan perangkat keras, perangkat lunak, serta *user* sebagai bahan

analisis kekurangan dan kebutuhan yang harus dipenuhi dalam perancangan sistem yang akan diterapkan.

### 3.2.6.1. Spesifikasi Perangkat Lunak

Perangkat lunak digunakan dalam sebuah sistem merupakan kumpulan perintah-perintah yang diberikan kepada perangkat keras agar saling berinteraksi untuk melakukan suatu tugas. Perangkat lunak yang dibutuhkan untuk membangun aplikasi Koperasi Simpan Pinjam ini adalah sebagai berikut:

1) Sistem operasi Windows
2) Netbeans 7.0 sebagai platform bahasa pemrograman JAVA
3) XAMPP sebagai database server dari MySQL
4) Browser ( Chrome / Firefox ) digunakan untuk memanggil struktur database SQL pada localserver phpmyadmin dari XAMPP

### 3.2.6.2. Spesifikasi Kebutuhan Perangkat Keras

Spesifikasi perangkat keras minimum yang digunakan dalam pembuatan aplikasi ini adalah :

| No | Perangkat Keras | Spesifikasi / Keterangan |
|---|---|---|
| | Processor | Prosesor 1,6 GHz |
| 2 | Monitor | Monitor Resolusi 1024 x 728 Pixels |
| 3 | VGA | 256 MB |
| 4 | Memori | 512 GB |
| 5 | Keyboard & Mouse | P/S atau USB |
| 6 | Hardisk | 40 GB |
| 7 | Jaringan / Modem | Untuk koneksi Internet dan akses ke server |

*Tabel 2 Spesifikasi Minimum Hardware Untuk User*

### 3.2.6.3. Spesifikasi Pengguna ( User )

Spesifikasi pengguna sistem dimaksudkan untuk mengetahui siapa saja *actor* yang terlibat dalam menjalankan system, yaitu :

1. Pengguna aplikasi koperasi simpan pinjam (*end user*) yaitu pengurus koperasi yang berlaku juga sebagai admin. Admin sebagai pengelola konten mempunyai kapabilitas dalam mengelola database seperti menambah, mengubah, dan menghapus data melalui aplikasi desktop koperasi simpan pinjam yang telah dibuat.

Berikut ini karakteristik pengguna sistem yang ditunjukkan oleh tabel berikut:

| Pengguna | Hak Akses | Pengalaman |
|---|---|---|
| Admin ( petugas Koperasi ) | Mengakses data konten aplikasi koperasi melalui aplikasi desktop. | Min. Mengatur dan memperbaiki data aplikasi koperasi simpan pinjam pada aplikasi desktop. |

*Tabel 3 Karakteristik Pengguna*

### 3.2.7. Analisis Kebutuhan Fungsional

#### 3.2.7.1. Perancangan Sistem Menggunakan *Object Oriented Design*

Setelah dilakukan analisa yang ada pada perangkat lunak bahwa untuk membangun aplikasi *forecasting* untuk koperasi simpan pinjam reka mandiri ini dengan menggunakan *Object Oriented Design* (OOD), maka untuk proses analisis sampai proses pelaporan dapat digambarkan dengan *use case diagram, Activity diagram,* dan *sequence diagram*.

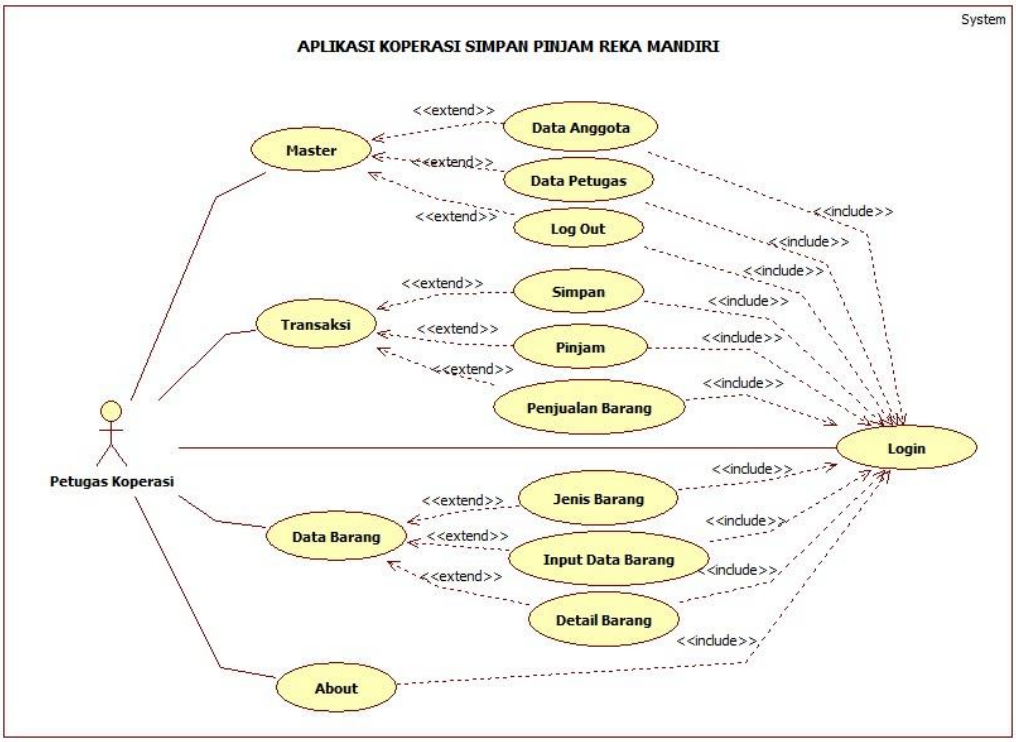
### 3.3. Pemodelan dan Perancangan Sistem

Pemodelan sistem berfungsi untuk menentukan fungsi-fungsi yang dapat dilakukan oleh sistem pada aplikasi serta menentukan kelas yang dibutuhkan untuk realisasi fungsi-fungsi sistem yang telah dianalisis sebelumnya dan mendeskripsikannya kedalam bentuk diagram.

Pemodelan *use case* adalah pemodelan sistem dari perspektif pandangan pemakai akhir (*end user*). Model *use case* adalah pandangan dari luar sistem, sementara model perancangan adalah pandangan dari dalam. Model *use case* menangkap penggunaan-penggunaan sistem, sedangkan model rancangan merepresentasikan pembangunan dari sistem.

*Use case* bekerja dengan skenario. Skenario mendeskripsikan urutan langkah-langkah (proses bisnis) yang dilakukan aktor terhadap sistem maupun sebaliknya, sistem terhadap aktor.

*Usecase* diagram aplikasi *koperasi simpan pinjam Reka Mandiri* untuk semua aktor dapat dilihat pada Gambar dibawah ini :




*Gambar 3 Use Case Diagram Aplikasi Koperasi*

### 3.3.1. Login

Fungsi ini melakukan login terhadap *user* yang menggunakan dan menjalankan program *aplikasi koperasi* untuk memastikan bahwa orang yang bersangkutan benar – benar operator yang memiliki hak akses terhadap perangkat lunak *aplikasi koperas*.

**a. *Use Case Login* (*Authentikasi User*)**

*Authentikasi user* merupakan proses pengecekan data *user* oleh sistem dengan memasukan *username* dan *password* kemudian *klik tombol Login*, untuk *use case login* (Authntikasi User) terdapat pada gambar dibawah ini :




*Gambar 4 Use Case Login (Authentikasi User)*

**b. Skenario *login***

Skenario merupakan suatu pengambaran langkah-langkah aksi aktor terhadap sistem *login*, untuk skenario *login* dapat dilihat pada tabel scenario login sebagai berikut:

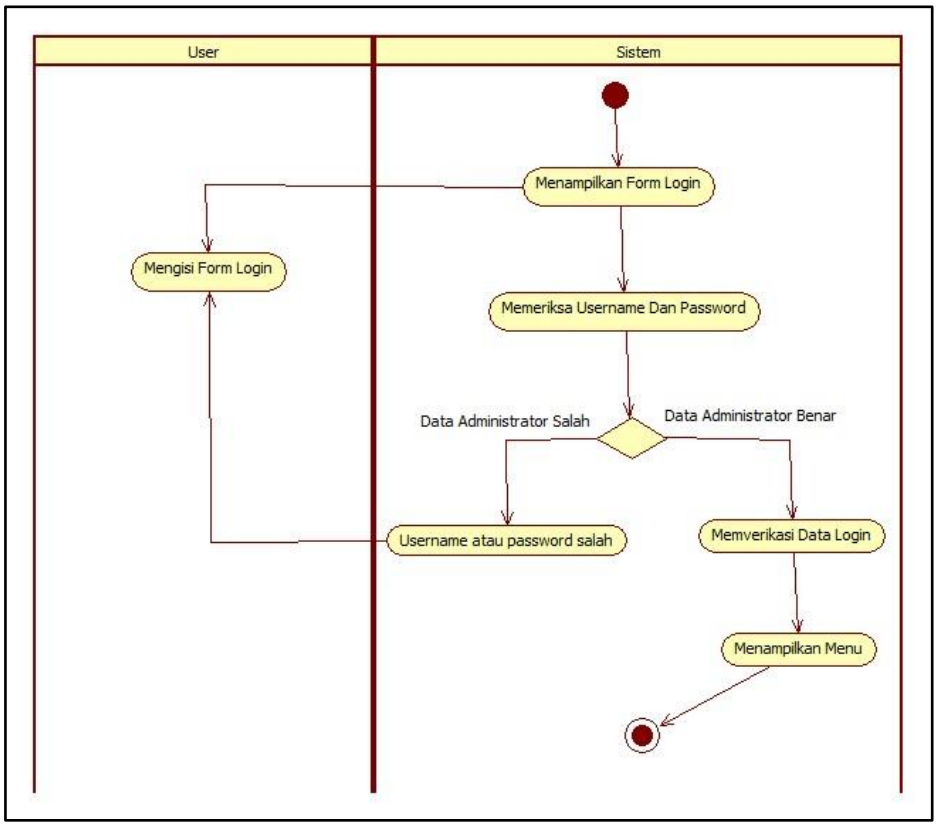
| **Identifikasi** | | |
|---|---|---|
| **Nomor** | 1 | |
| **Nama** | *Login* | |
| **Tujuan** | Memastikan hak akses *User* sebagai operator | |
| **Deskripsi** | Sistem menerima data identifikasi *user* dan berdasarkan data tersebut memutuskan apakah pengguna memiliki hak akses sebagai operator | |
| **Tipe** | | |
| **Aktor** | *user*, Aplikasi *koperasi* | |
| **Skenario Utama** | | |
| **Kondisi awal** | Aplikasi *koperasi* baru dinyalakan | |
| **Aksi Aktor** | | **Reaksi Sistem** |
| *User* memasukkan password | | |
| | | Mengecek Password |
| Sistem Perangkat Lunak *koperasi* | | |

| | |
|---|---|
| Memberikan Respon | |
| | Memeriksa respons, jika benar maka akan masuk ke dalam aplikasi *forecasting* dan jika salah maka akan keluar *show message* (pesan kesalahan) |
| | |
| **Skenario Alternatif – Autentikasi Gagal** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| Operator memasukkan lagi password | Menampilkan *show message* gagal dan akan menampilkan password kembali dengan kondisi *default* |
| **Kondisi akhir** | Hak akses pengguna sebagai *user* sudah dipastikan |

*Tabel 4 Skenario Login*

### c. *Activity Diagram Login*

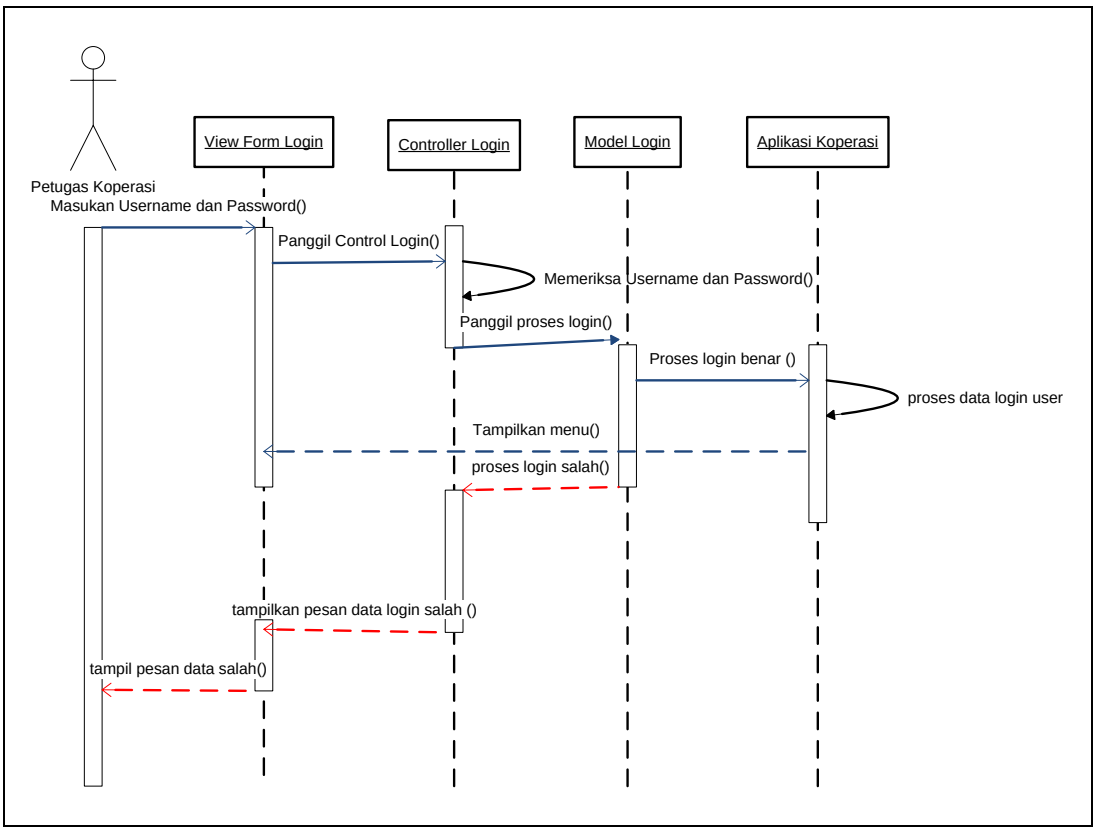
*Activity* diagram *login* menggambarkan aliran kejadian (*flow of events*) pada saat aktor melakukan login, untuk *actifity* diagram *login* dapat di lihat pada gambar *activity diagram*, sebagai berikut:




*Gambar 5 Activity Diagram Login*

### d. *Sequence* Diagram *Login*

*Sequence* diagram *login* digunakan untuk menunjukan aliran fungsionalitas dalam *use case login user*, untuk *sequence* diagram *login user* dapat dilihat pada gambar *sequence* diagram untuk *login user* sebagai berikut :

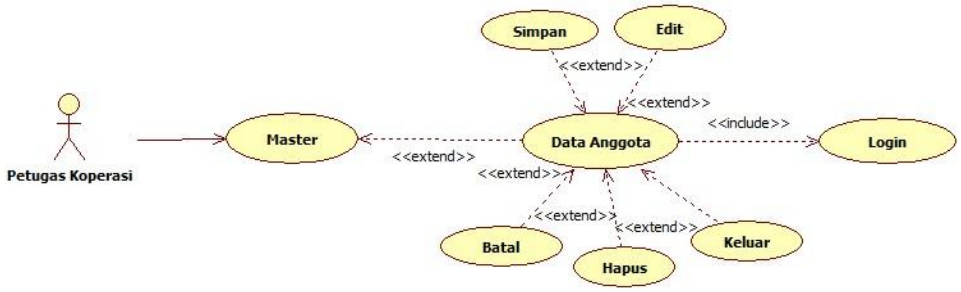



*Gambar 6 Sequence Diagram Data Login*

### 3.3.2. Data Anggota

#### a. *Use Case* Data Anggota

*Use case* data anggota merupakan suatu proses untuk memasukan identitas anggota yang dilakukan oleh *user*, tetapi untuk melakukan *input* data anggota harus melakukan *login* terlebih dahulu, *input* data anggota dilakukan dengan cara memilih menu Master dan pilih data anggota kemudian *klik* maka akan muncul *form* data anggota, *user* memasukan identitas anggota dan *klik* tombol *simpan* jika telah selesai maka datanya akan ditampilkan dalam tabel data anggota, untuk gambar *use case* data anggota dapat dilihat pada gambar dibawah sebagai berikut :




*Gambar 7 Use Case Data Anggota*

#### b. Skenario Data Anggota

Skenario data anggota merupakan suatu pengambaran langkah-langkah aksi aktor terhadap sistem untuk melakukan *input* data anggota, untuk skenario data anggota dapat dilihat pada tabel dibawah ini :

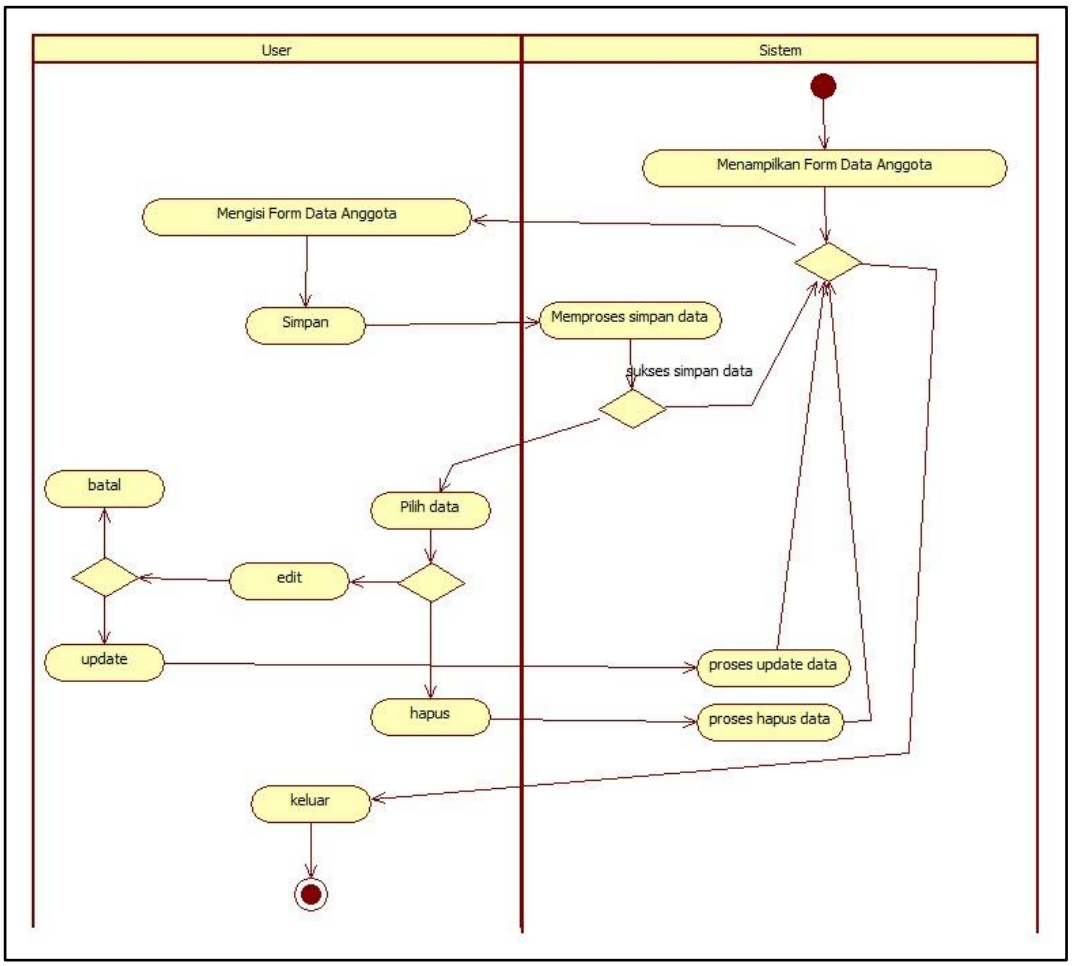
| **Identifikasi** | |
|---|---|
| **Nomor** | 2 |
| **Nama** | *D*ata Anggota |
| **Tujuan** | Melakukan pemasukkan data anggota. |
| **Deskripsi** | *User* memasukkan data anggota koperasi |
| **Tipe** | |
| **Aktor** | *User*, Sistem Aplikasi *koperasi* |
| **Skenario Utama** | |
| **Kondisi awal** | *User* sudah terautentikasi |

| Aksi Aktor | Reaksi Sistem |
|---|---|
| *User* memilih menu Master ”data anggota” | |
| | Menampilkan *form* data anggota |
| *User* memasukkan data anggota | |
| | Menampilkan tabel data anggota dalam *form* data anggota |
| Sistem memberikan respon | |
| | Memberitahu bahwa data yang dimasukkan sudah tersimpan |
| **Kondisi akhir** | Aplikasi *koperasi* menyimpan data anggota ke dalam *database* |

*Tabel 5 Skenario Data Anggota*

### c. *Activity Diagram* Data Anggota

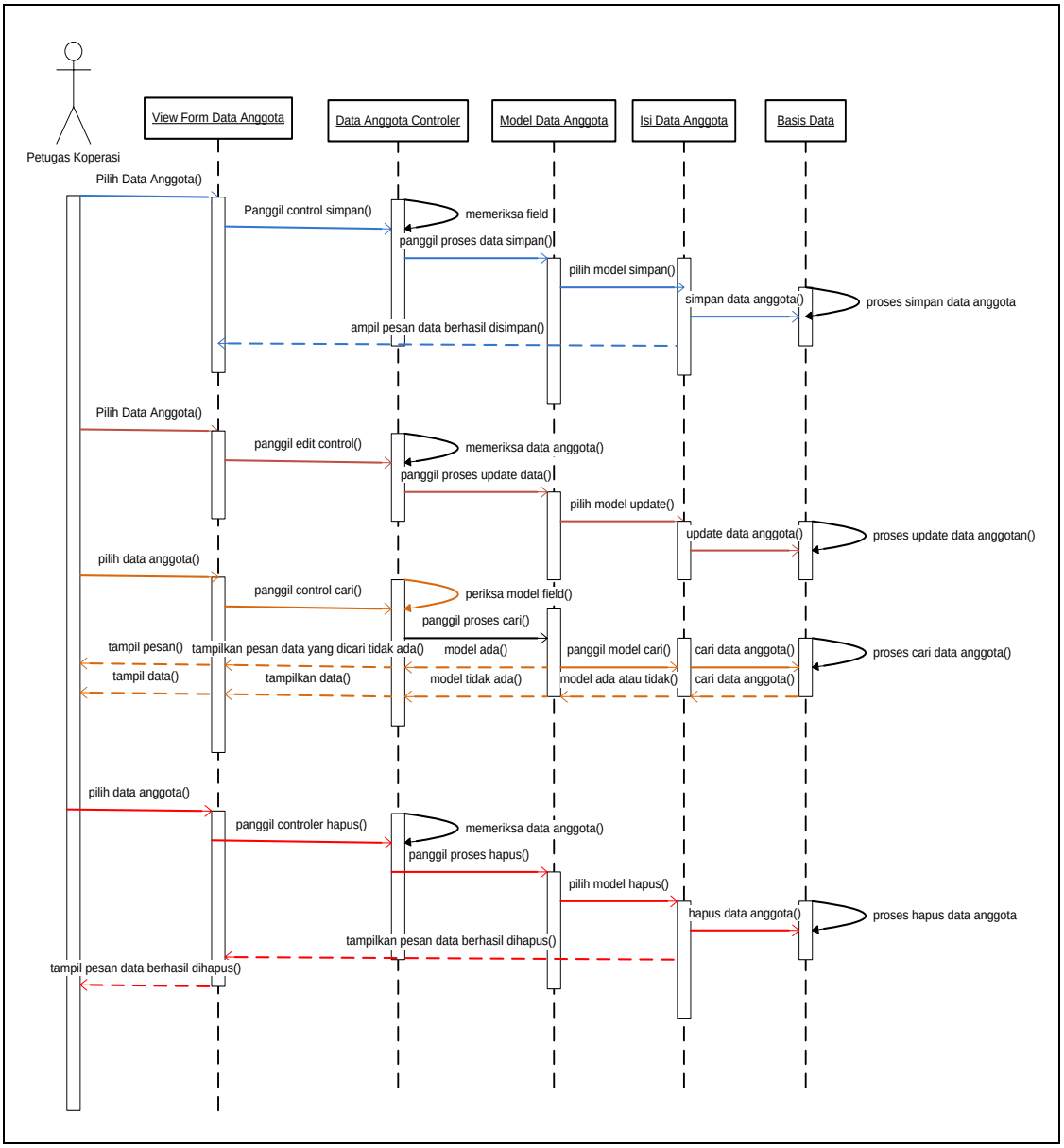
*Activity* diagram data anggota menggambarkan aliran kejadian (*flow of events*) pada saat user melakukan aktifitas *input* data anggota kedalam sistem, untuk *actifity* diagram data anggota dapat di lihat pada gambar dibawah ini:




*Gambar 8 Activity Diagram Anggota*

### d. *Sequence* Diagram Data Anggota

*Sequence* diagram data Anggota digunakan untuk menunjukan aliran fungsionalitas dalam *use case* data anggota, untuk *sequence* diagram data anggota dapat dilihat pada gambar dibawah ini :



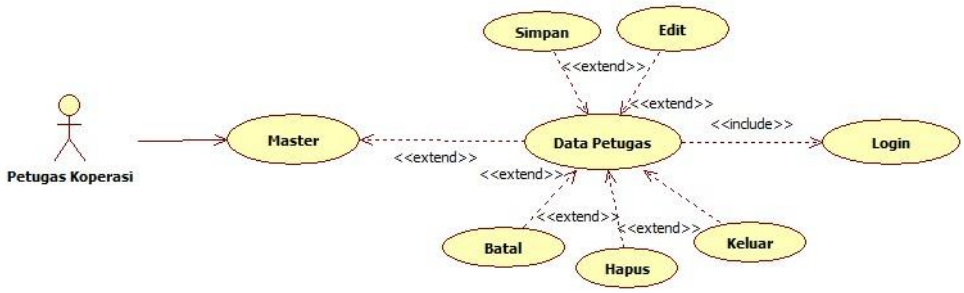

*Gambar 9 Sequence Diagram Data Anggota*

### 3.3.3. Data Petugas

#### a. *Use Case* Data Petugas

*Use case* data petugas merupakan suatu proses untuk membatasi aktifitas antara admin dan *user*, jika melakukan *input* data petugas harus *login* terlebih dahulu, kemudian pilih menu master kemudian *user* dan pilih

data petugas, maka akan muncul *form* data *petugas*, *user* memasukan identitas data petugas dan *klik* tombol *simpan* jika telah selesai maka datanya akan ditampilkan dalam tabel data petugas, untuk gambar *use case* data petugas dapat dilihat pada gambar dibawah sebagai berikut :




*Gambar 10 Use Case Data Petugas*

## b. Skenario Data Petugas

Skenario data petugas merupakan suatu pengambaran langkah-langkah aksi aktor terhadap sistem untuk melakukan *input* data petugas, untuk skenario data petugas dapat dilihat pada tabel dibawah ini :

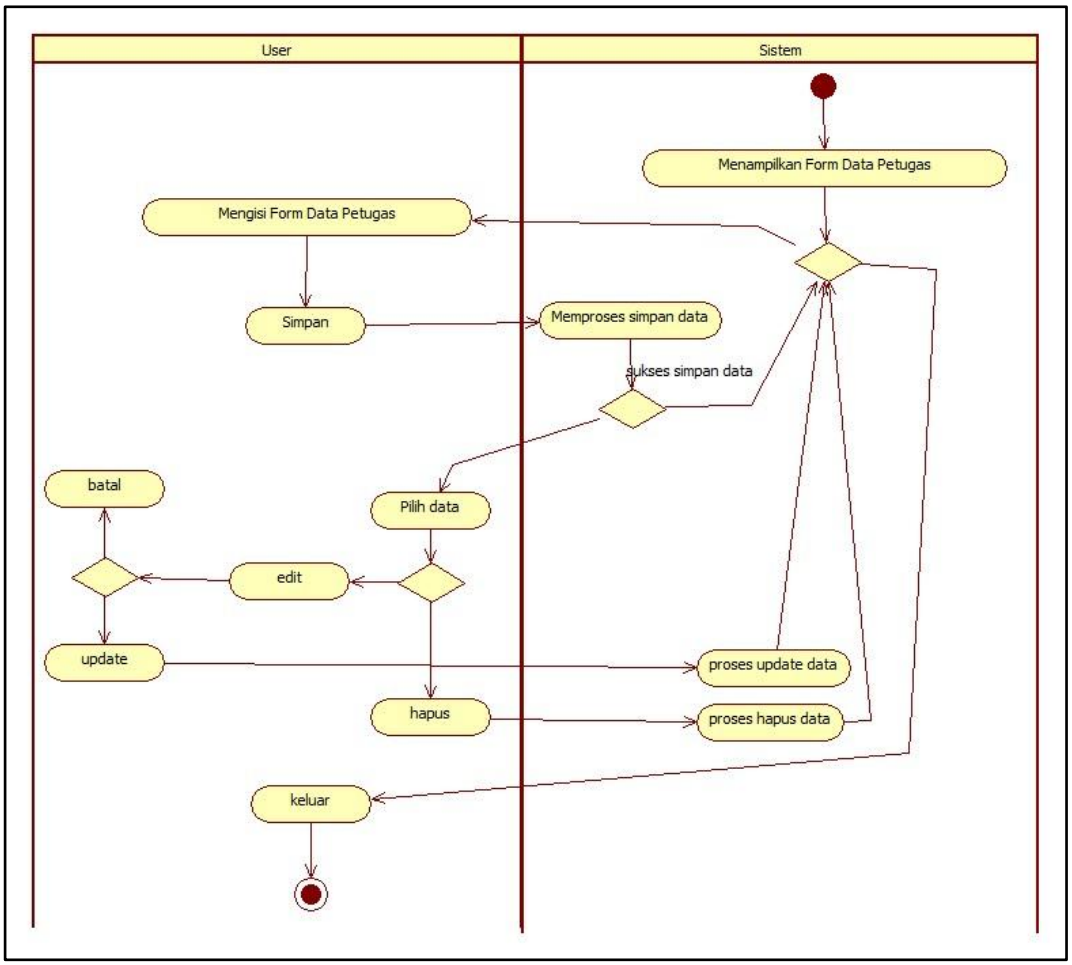
| **Identifikasi** | |
|---|---|
| **Nomor** | 3 |
| **Nama** | *D*ata Petugas |
| **Tujuan** | Melakukan pemasukkan data petugas |
| **Deskripsi** | *User* memasukkan data petugas koperasi |
| **Tipe** | |
| **Aktor** | *User,* Sistem Aplikasi *koperasi* |
| **Skenario Utama** | |
| **Kondisi awal** | *User* sudah terautentikasi |
| **Aksi Aktor** | **Reaksi Sistem** |
| *User* memilih menu Master ”data petugas” | |
| | Menampilkan *form* data petugas |
| *User* memasukkan data petugas | |
| | Menampilkan tabel data petugasdalam *form* data petugas |
| Sistem memberikan respon | |

| | |
|---|---|
| | Memberitahu bahwa data yang dimasukkan sudah tersimpan |
| **Kondisi akhir** | Aplikasi *koperasi* menyimpan data petugas ke dalam *database* |

*Tabel 6 Skenario Data Petugas*

### c. *Activity Diagram* Data Petugas

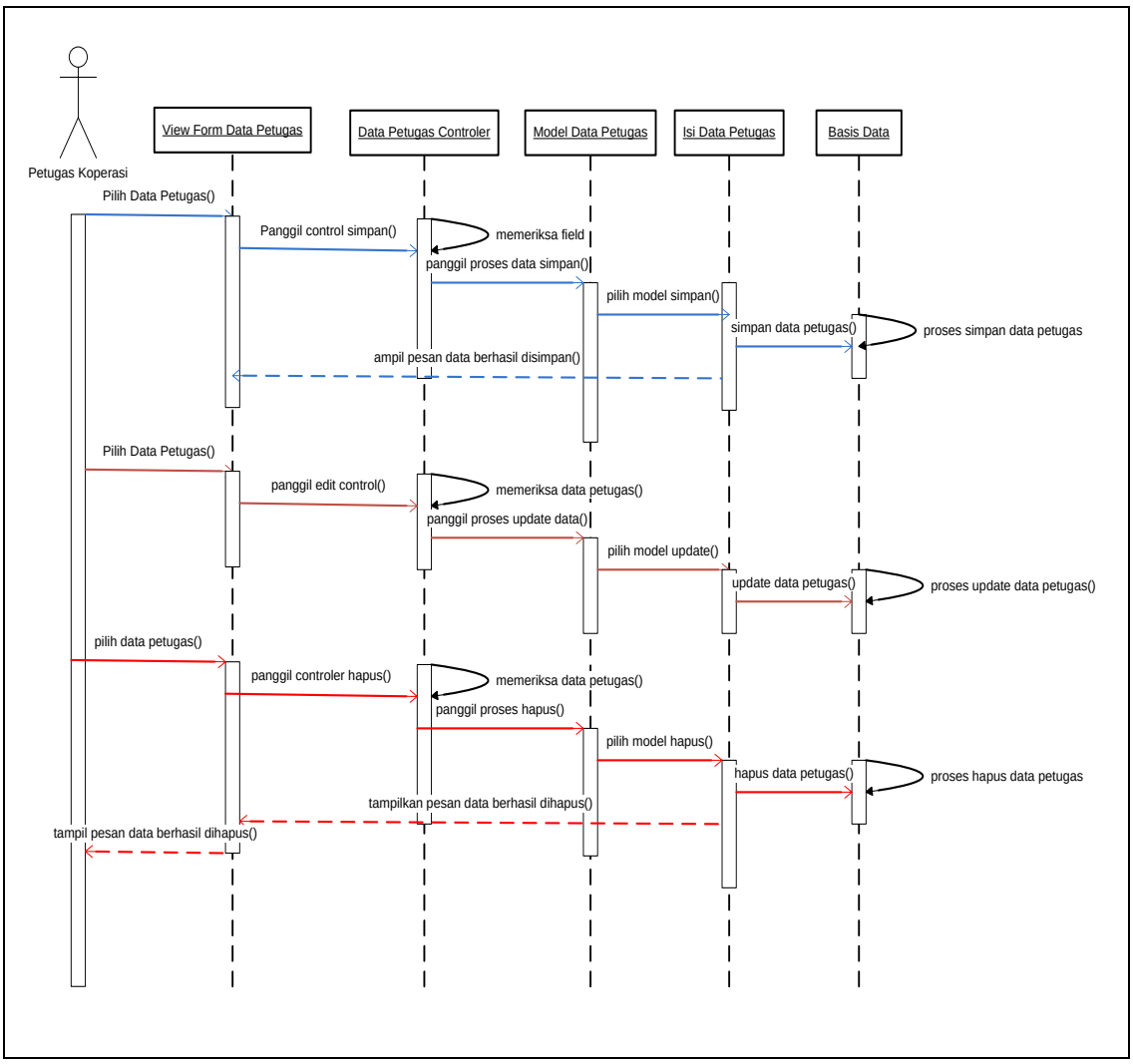
*Activity* diagram data petugas menggambarkan aliran kejadian (*flow of events*) pada saat user melakukan aktifitas *input* data anggota kedalam sistem, untuk *actifity* diagram data petugas dapat di lihat pada gambar dibawah ini:




*Gambar 11 Activity Diagram Data Petugas*

### d. *Sequence* Diagram Data Petugas

*Sequence* diagram data Petugas digunakan untuk menunjukan aliran fungsionalitas dalam *use case* data petugas, untuk *sequence* diagram data petugas dapat dilihat pada gambar dibawah ini :



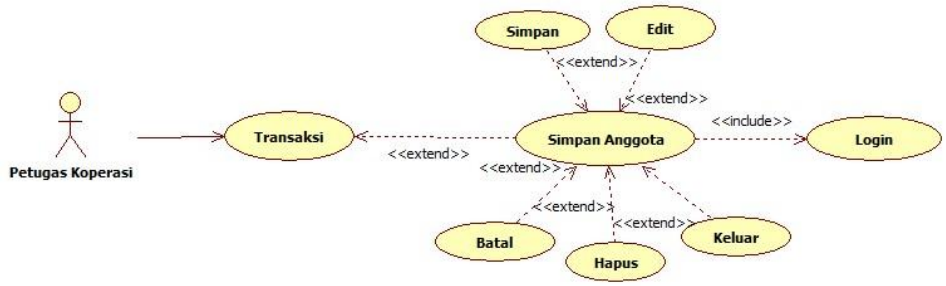

*Gambar 12 Sequence Diagram Data Petugas*

## 3.3.4. Data Transaksi Simpan

### a. *Use Case* Data Transaksi Simpan

*Use case* data transaksi simpan merupakan suatu proses untuk membatasi aktifitas antara admin dan *user*, jika melakukan *input* data

transaksi simpan harus *login* terlebih dahulu, kemudian pilih menu Transaksi kemudian pilih simpan, maka akan muncul *form* data transaksi simpan, *user* memasukan identitas data simpan dan *klik* tombol *simpan* jika telah selesai maka datanya akan ditampilkan dalam tabel data transaksi simpan, untuk gambar *use case* data transaksi simpan dapat dilihat pada gambar dibawah sebagai berikut :




*Gambar 13 Use Case Transaksi Simpan*

## b. Skenario Data Transaksi Simpan

Skenario data transaksi simpan merupakan suatu pengambaran langkah-langkah aksi aktor terhadap sistem untuk melakukan *input* data transaksi simpanan anggota, untuk skenario data transaksi simpanan anggota dapat dilihat pada tabel dibawah ini :

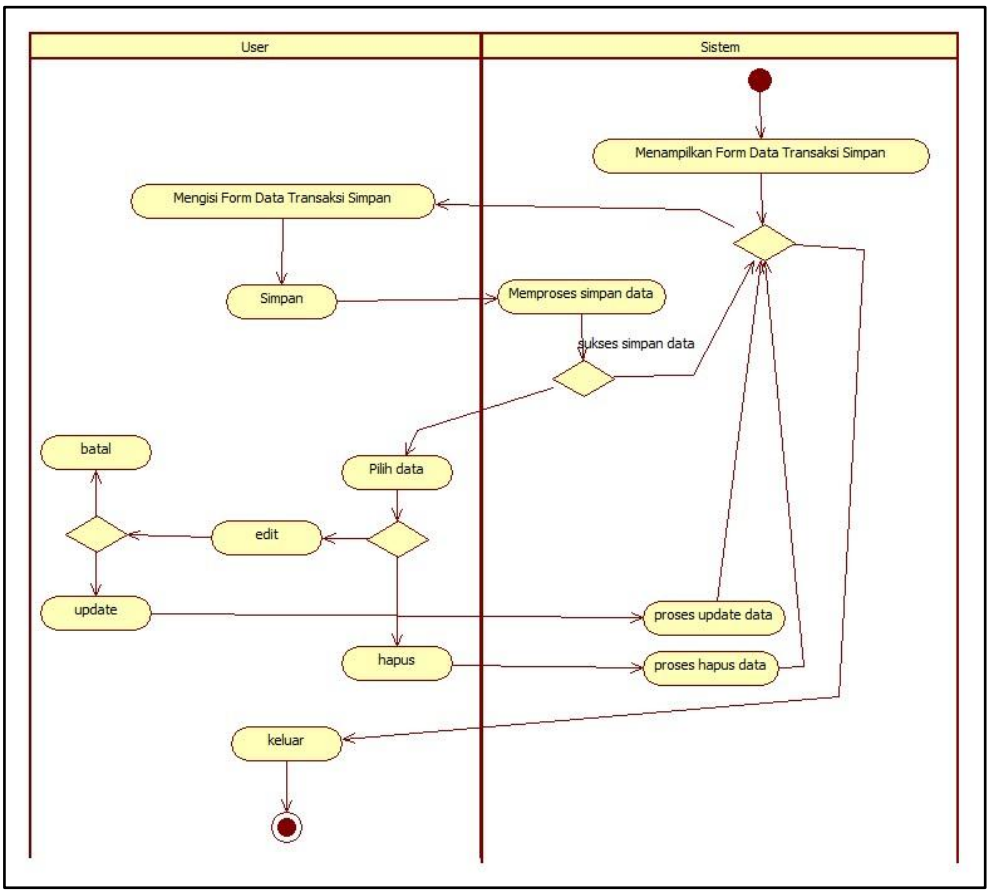
| **Identifikasi** | |
|---|---|
| **Nomor** | 4 |
| **Nama** | *D*ata Transaksi Simpan |
| **Tujuan** | Melakukan pemasukkan data transaksi simpanan anggota |
| **Deskripsi** | *User* memasukkan data transaksi simpanan anggota koperasi |
| **Tipe** | |
| **Aktor** | *User* , Sistem Aplikasi *koperasi* |
| **Skenario Utama** | |
| **Kondisi awal** | *User* sudah terautentikasi |
| **Aksi Aktor** | **Reaksi Sistem** |
| *User* memilih menu Transaksi ”Simpan ” | |
| | Menampilkan *form* data simpanan |
| *User* memasukkan data simpanan anggota | |

| | | |
|---|---|---|
| | | Menampilkan tabel data simpanan anggota dalam *form* data simpanan anggota |
| Sistem memberikan respon | | |
| | | Memberitahu bahwa data yang dimasukkan sudah tersimpan |
| **Kondisi akhir** | Aplikasi *koperasi* menyimpan data simpanan ke dalam *database* | |

*Tabel 7 Skenario Data Transaksi Simpan*

### c. *Activity Diagram* Data Transaksi Simpan

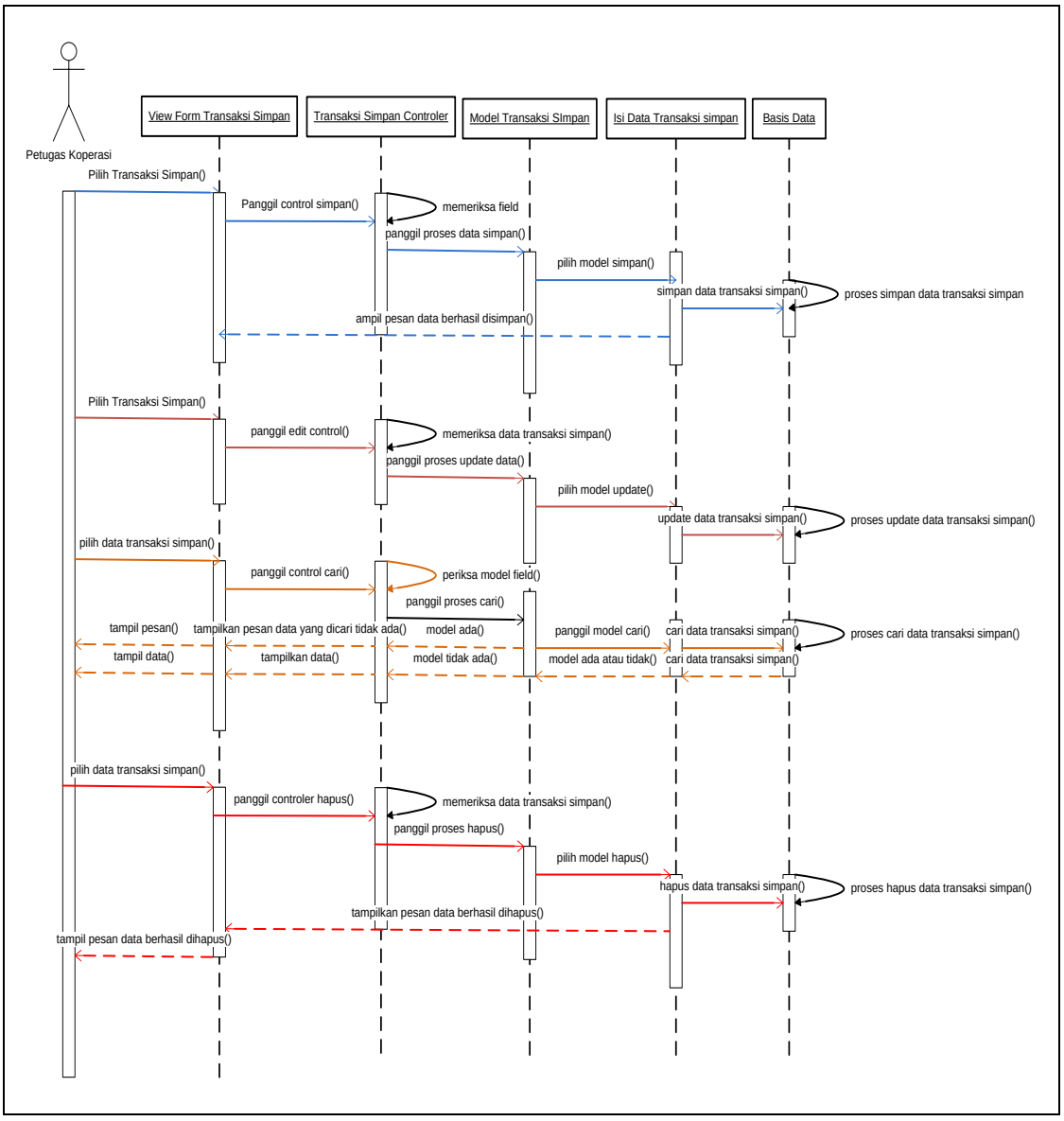
*Activity* diagram data transaksi simpanan anggota menggambarkan aliran kejadian (*flow of events*) pada saat user melakukan aktifitas *input* data transaksi simpanan anggota kedalam sistem, untuk *actifity* diagram data transaksi simpanan anggota dapat di lihat pada gambar dibawah ini:




*Gambar 14 Activity Diagram Transaksi Simpan*

### d. *Sequence* Diagram Data Transaksi Simpanan Anggota

*Sequence* diagram data transaksi simpanan anggota digunakan untuk menunjukan aliran fungsionalitas dalam *use case* data transaksi simpanan anggota, untuk *sequence* diagram data transaksi simpanan anggota dapat dilihat pada gambar dibawah ini :

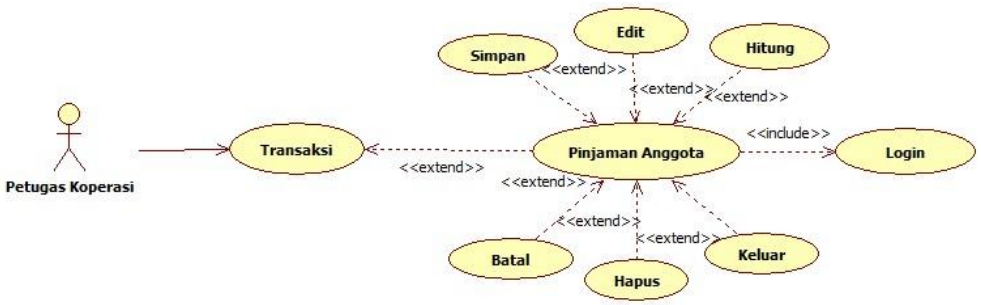



*Gambar 15 Sequence Diagram Transaksi Simpan*

### 3.3.5. Data Transaksi Pinjam

#### *a. Use Case* Data Transaksi Pinjam

*Use case* data transaksi pinjam merupakan suatu proses untuk membatasi aktifitas antara admin dan *user*, jika melakukan *input* data transaksi pinjam harus *login* terlebih dahulu, kemudian pilih menu Transaksi kemudian pilih pinjam, maka akan muncul *form* data transaksi pinjam, *user* memasukan identitas data pinjam dan *klik* tombol *simpan* jika telah selesai maka datanya akan ditampilkan dalam tabel data pinjam, untuk gambar *use case* data transaksi pinjam dapat dilihat pada gambar dibawah sebagai berikut :




*Gambar 16 Use Case Transaksi Pinjam*

#### b. Skenario Data Transaksi Pinjam

Skenario data transaksi pinjam merupakan suatu pengambaran langkah-langkah aksi aktor terhadap sistem untuk melakukan *input* data transaksi pinjaman anggota, untuk skenario data transaksi pinjaman anggota dapat dilihat pada tabel dibawah ini :

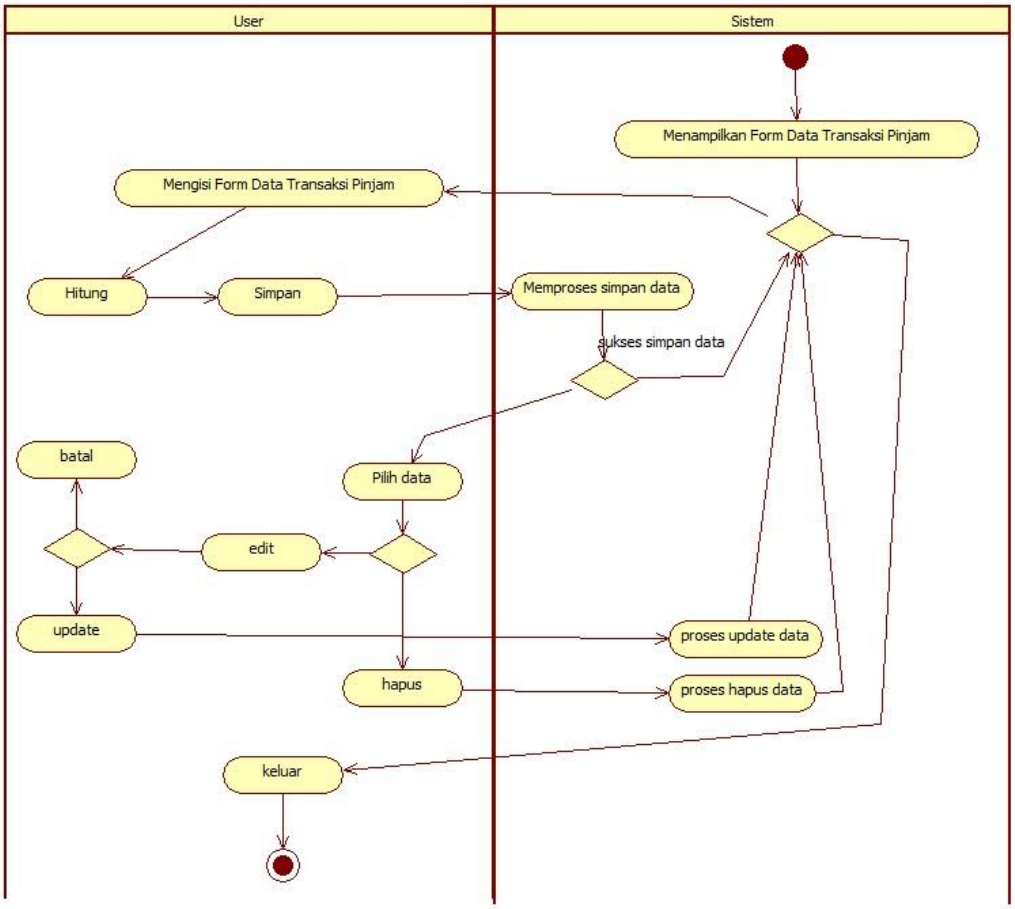
| **Identifikasi** | |
|---|---|
| **Nomor** | 5 |
| **Nama** | Data Transaksi Pinjam |
| **Tujuan** | Melakukan pemasukkan data transaksi pinjaman anggota |
| **Deskripsi** | *User* memasukkan data transaksi pinjaman anggota koperasi |

| | | |
|---|---|---|
| **Tipe** | | |
| **Aktor** | *User* , Sistem Aplikasi *koperasi* | |
| **Skenario Utama** | | |
| **Kondisi awal** | *User* sudah terautentikasi | |
| **Aksi Aktor** | | **Reaksi Sistem** |
| *User* memilih menu Transaksi ”pinjam” | | |
| | | Menampilkan *form* data pinjaman |
| *User* memasukkan data pinjaman | | |
| | | Menampilkan tabel data pinjaman dalam *form* data pinjaman |
| Sistem memberikan respon | | |
| | | Memberitahu bahwa data yang dimasukkan sudah tersimpan |
| **Kondisi akhir** | Aplikasi *koperasi* menyimpan data pinjaman ke dalam *database* | |

*Tabel 8 Skenario Data Transaksi Pinjam*

### c. *Activity Diagram* Data Transaksi Pinjam

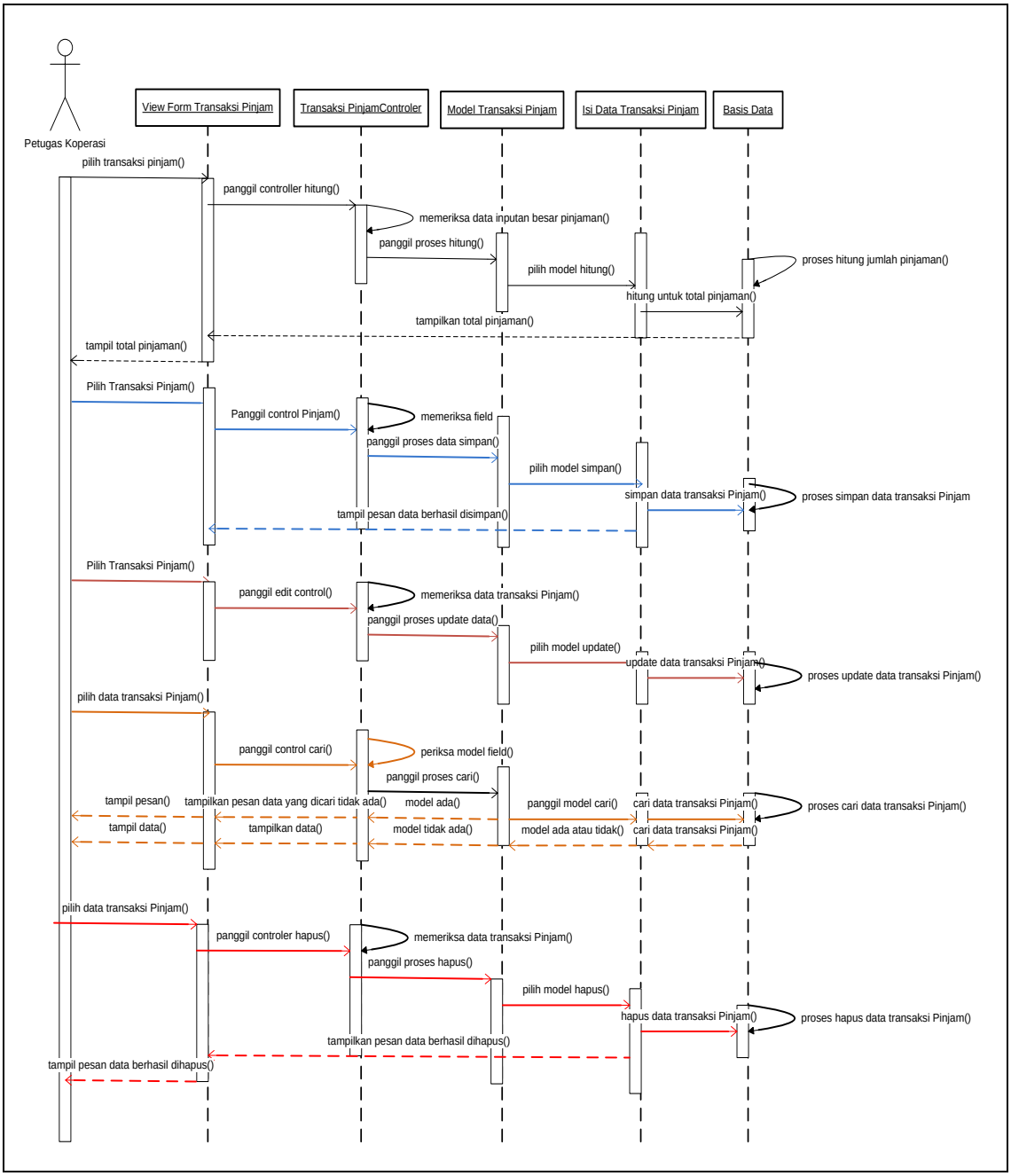
*Activity* diagram data transaksi pinjaman anggota menggambarkan aliran kejadian (*flow of events*) pada saat user melakukan aktifitas *input* data transaksi pinjaman anggota kedalam sistem, untuk *actifity* diagram data transaksi pinjaman anggota dapat di lihat pada gambar dibawah ini:




*Gambar 17 Activity Diagram Transaksi Pinjam*

### d. *Sequence* Diagram Data Transaksi Pinjaman Anggota

*Sequence* diagram data transaksi pinjaman anggota digunakan untuk menunjukan aliran fungsionalitas dalam *use case* data transaksi pinjaman anggota, untuk *sequence* diagram data transaksi pinjaman anggota dapat dilihat pada gambar dibawah ini :

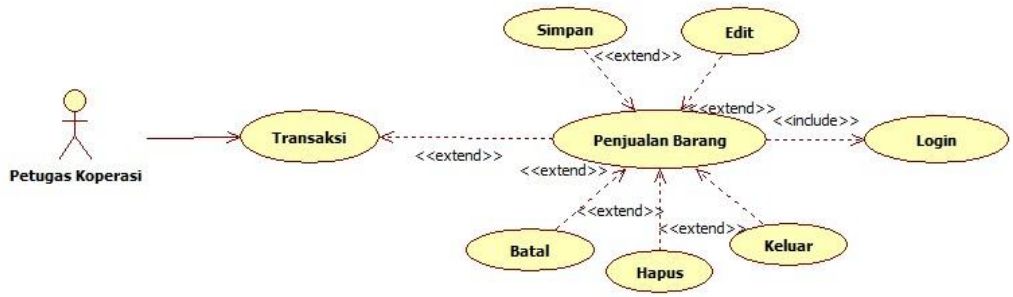



*Gambar 18 Sequence Diagram Tansaksi Pinjam*

### 3.3.6. Data Transaksi Penjualan Barang

#### *a. Use Case* Data Transaksi Penjualan Barang

*Use case* data transaksi penjualan barang merupakan suatu proses untuk membatasi aktifitas antara admin dan *user*, jika melakukan *input* data transaksi penjualan barang harus *login* terlebih dahulu, kemudian pilih menu Transaksi kemudian pilih penjualan barang, maka akan muncul *form* data transaksi penjualan barang, *user* memasukan identitas data penjualan barang dan *klik* tombol *simpan* jika telah selesai maka datanya akan ditampilkan dalam tabel data penjualan barang, untuk gambar *use case* data transaksi penjualan barang dapat dilihat pada gambar dibawah sebagai berikut :




*Gambar 19 Use Case Transaksi Penjualan Barang*

#### b. Skenario Data Transaksi Penjualan Barang

Skenario data transaksi penjualan barang merupakan suatu pengambaran langkah-langkah aksi aktor terhadap sistem untuk melakukan *input* data transaksi penjualan barang, untuk skenario data transaksi penjualan barang dapat dilihat pada tabel dibawah ini :

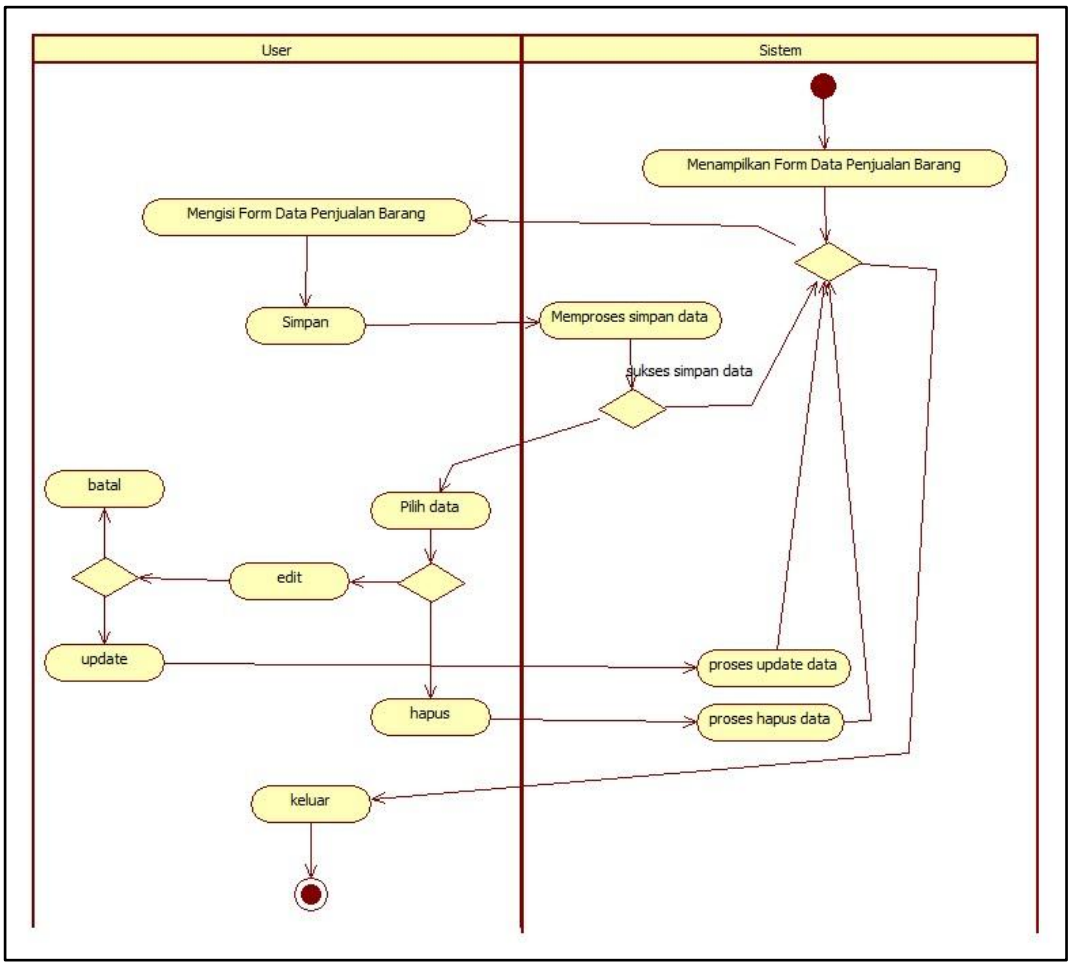
| **Identifikasi** | |
|---|---|
| **Nomor** | 6 |
| **Nama** | Data Transaksi Penjualan Barang |
| **Tujuan** | Melakukan pemasukkan data transaksi penjualan barang |
| **Deskripsi** | *User* memasukkan data transaksi penjualan barang koperasi |
| **Tipe** | |
| **Aktor** | *User* , Sistem Aplikasi *koperasi* |

| Skenario Utama | | |
|---|---|---|
| **Kondisi awal** | *User* sudah terautentikasi | |
| **Aksi Aktor** | | **Reaksi Sistem** |
| *User* memilih menu Transaksi ”penjualan barang” | | |
| | | Menampilkan *form* data penjualan barang |
| *User* memasukkan data penjualan barang | | |
| | | Menampilkan tabel data penjualan barang dalam *form* data penjualan barang |
| Sistem memberikan respon | | |
| | | Memberitahu bahwa data yang dimasukkan sudah tersimpan |
| **Kondisi akhir** | Aplikasi *koperasi* menyimpan data penjualan barang ke dalam *database* | |

*Tabel 9 Skenario Data Transaksi Pinjam*

### c. *Activity Diagram* Data Transaksi Penjualan Barang

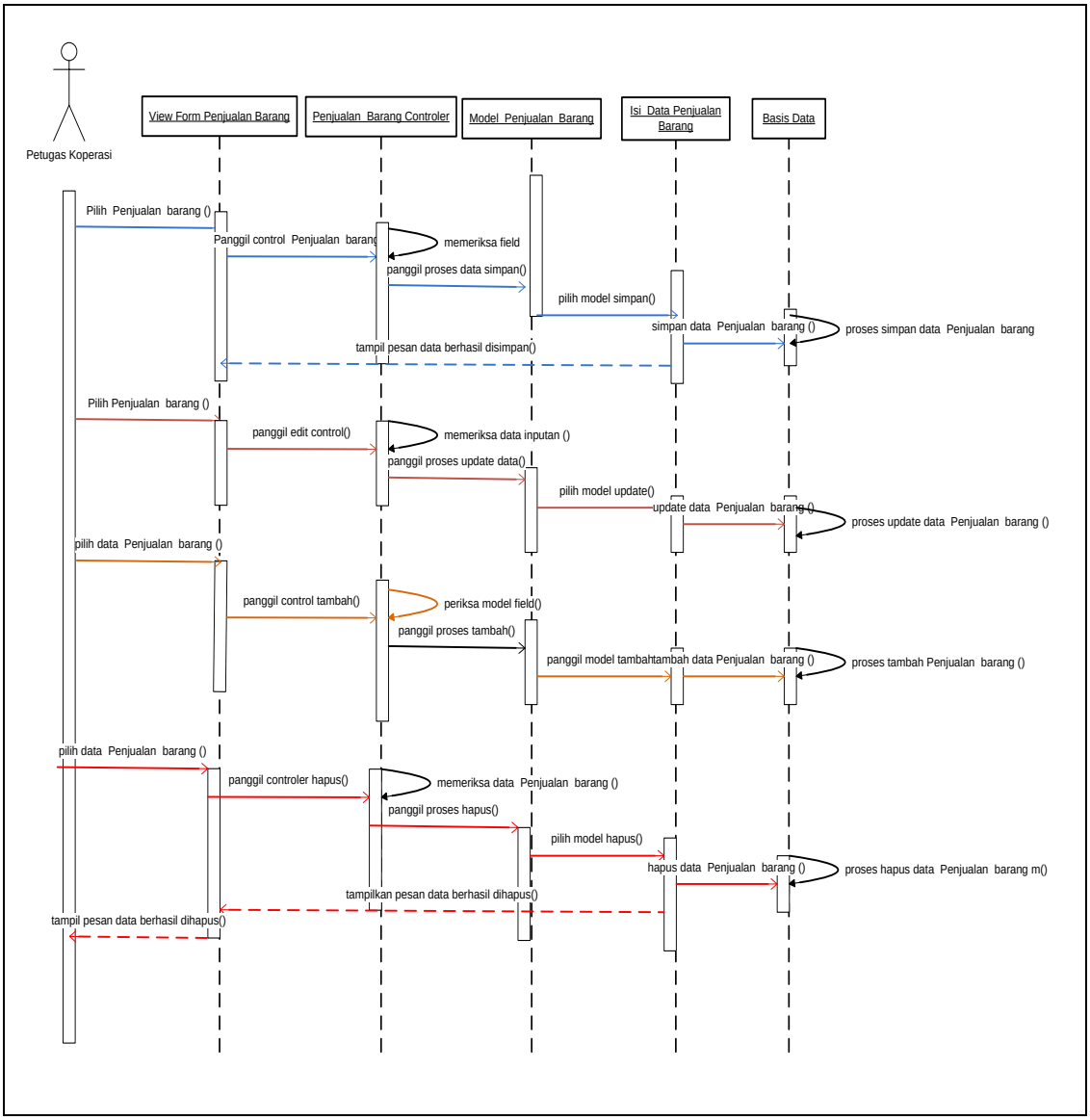
*Activity* diagram data transaksi penjualan barang menggambarkan aliran kejadian (*flow of events*) pada saat user melakukan aktifitas *input* data transaksi penjualan barang kedalam sistem, untuk *actifity* diagram data transaksi penjualan barang dapat di lihat pada gambar dibawah ini:




*Gambar 20 Activity Diagram Transaksi Penjualan Barang*

### d. *Sequence* Diagram Data Transaksi Penjualan Barang

*Sequence* diagram data transaksi penjualan barang digunakan untuk menunjukan aliran fungsionalitas dalam *use case* data transaksi penjualan barang, untuk *sequence* diagram data transaksi penjualan barang dapat dilihat pada gambar dibawah ini :



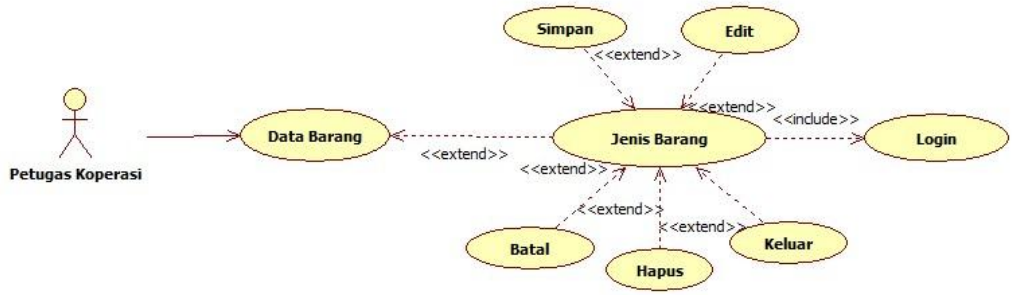

*Gambar 21 Sequence Diagram Transaksi Penjualan Barang*

### 3.3.7. Data Jenis Barang

#### *a. Use Case* Data Jenis Barang

*Use case* data jenis barang merupakan suatu proses untuk membatasi aktifitas antara admin dan *user*, jika melakukan *input* data jenis barang harus *login* terlebih dahulu, kemudian pilih menu Transaksi kemudian pilih jenis barang, maka akan muncul *form* data jenis barang, *petugas* memasukan

identitas data jenis barang dan *klik* tombol *simpan* jika telah selesai maka datanya akan ditampilkan dalam tabel data jenis barang, untuk gambar *use case* data transaksi jenis barang dapat dilihat pada gambar dibawah sebagai berikut :




*Gambar 22 Use Case Data Jenis Barang*

## b. Skenario Data Jenis Barang

Skenario data jenis barang merupakan suatu pengambaran langkah-langkah aksi aktor terhadap sistem untuk melakukan *input* data jenis barang, untuk skenario data jenis barang dapat dilihat pada tabel dibawah ini :

| **Identifikasi** | |
|---|---|
| **Nomor** | 7 |
| **Nama** | Data Jenis Barang |
| **Tujuan** | Melakukan pemasukkan data jenis barang |
| **Deskripsi** | *User* memasukkan data jenis barang di koperasi |
| **Tipe** | |
| **Aktor** | *User* , Sistem Aplikasi *koperasi* |
| **Skenario Utama** | |
| **Kondisi awal** | *User* sudah terautentikasi |

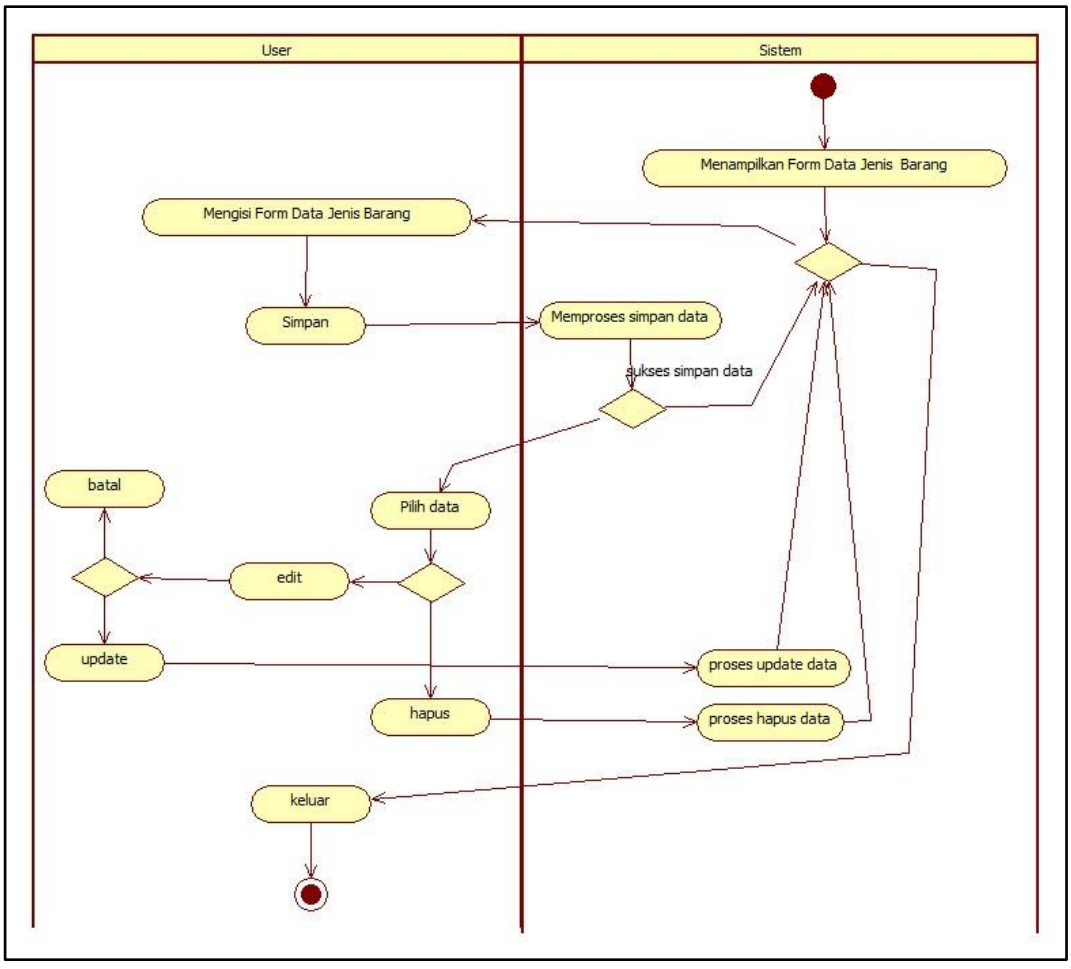
| **Aksi Aktor** | **Reaksi Sistem** |
|---|---|
| *User* memilih menu Transaksi ”jenis barang” | |
| | Menampilkan *form* data jenis barang |
| *User* memasukkan data jenis barang | |
| | Menampilkan tabel data jenis barang dalam *form* data jenis barang |
| Sistem memberikan respon | |

| | | |
|---|---|---|
| | | Memberitahu bahwa data yang dimasukkan sudah tersimpan |
| **Kondisi akhir** | Aplikasi *koperasi* menyimpan data jenis barang ke dalam *database* | |

*Tabel 10 Skenario Data Transaksi Pinjam*

### c. *Activity Diagram* Data Jenis Barang

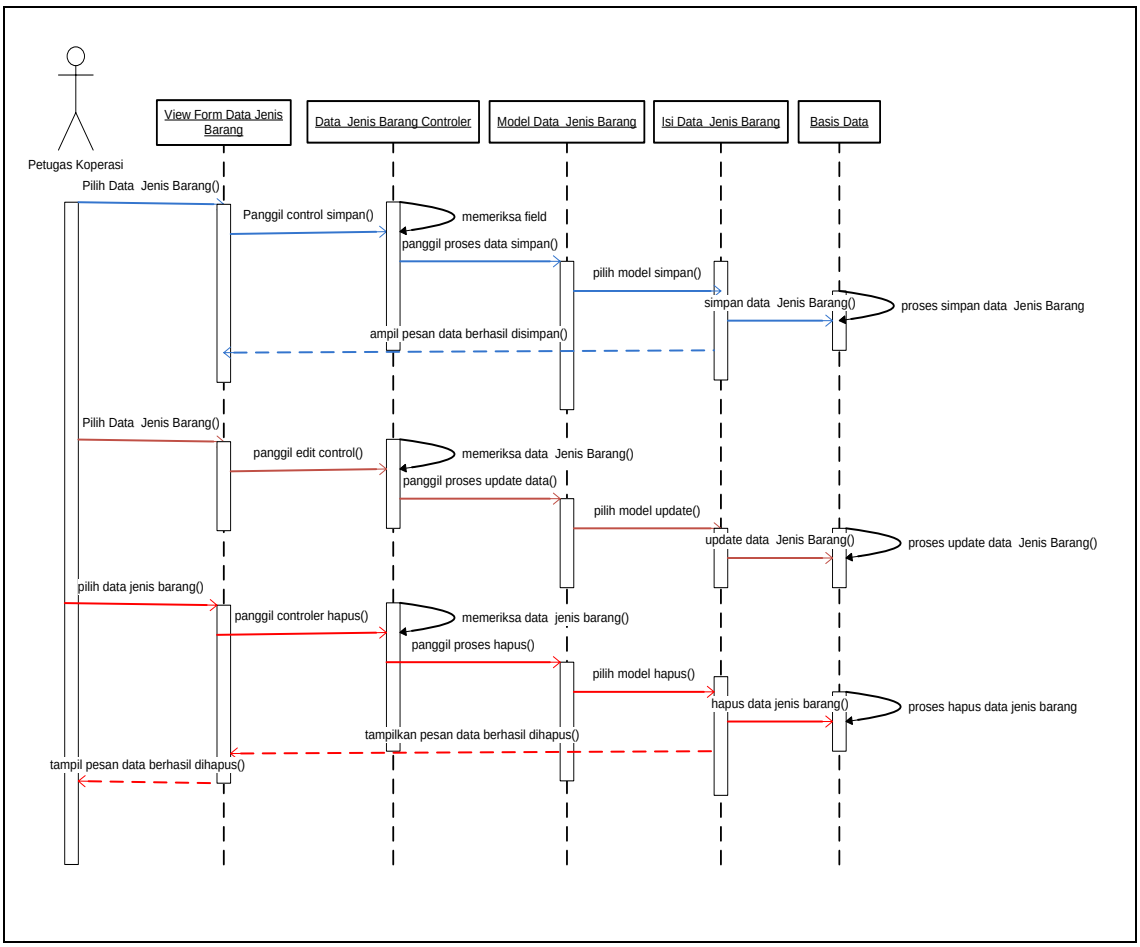
*Activity* diagram data jenis barang menggambarkan aliran kejadian (*flow of events*) pada saat user melakukan aktifitas *input* data jenis barang kedalam sistem, untuk *actifity* diagram jenis barang dapat di lihat pada gambar dibawah ini:




*Gambar 23 Activity Diagram Data Jenis Barang*

### d. *Sequence* Diagram Data Jenis Barang

*Sequence* diagram data jenis barang digunakan untuk menunjukan aliran fungsionalitas dalam *use case* data jenis barang, untuk *sequence* diagram data jenis barang dapat dilihat pada gambar dibawah ini :



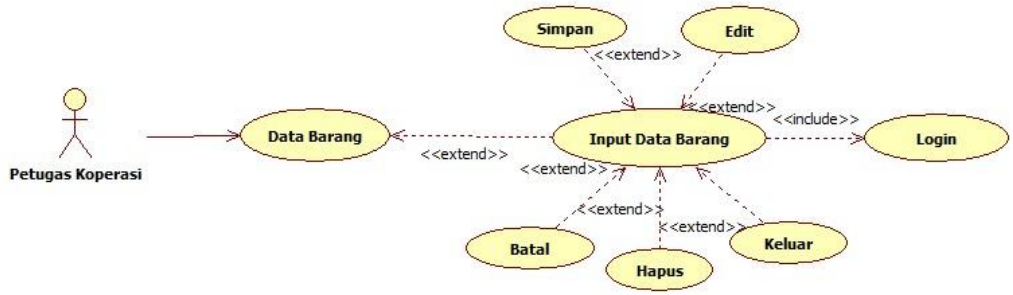

*Gambar 24 Sequence Diagram Input Nama Jenis Barang*

## 3.3.8. Input Data Pembelian Barang

### *a. Use Case Input* Data Pembelian Barang

*Use case* input data barang merupakan suatu proses untuk membatasi aktifitas antara admin dan *user*, jika melakukan *input* input data barang harus *login* terlebih dahulu, kemudian pilih menu Transaksi kemudian pilih input

data barang, maka akan muncul *form* input data barang, *petugas* memasukan identitas data barang dan *klik* tombol *simpan* jika telah selesai maka datanya akan ditampilkan dalam tabel data barang, untuk gambar *use case* input data barang dapat dilihat pada gambar dibawah sebagai berikut :




*Gambar 25 Use Case Input Data Pembelian Barang*

## b. Skenario Input Data Pembelian Barang

Skenario input data barang merupakan suatu pengambaran langkah-langkah aksi aktor terhadap sistem untuk melakukan input data barang, untuk skenario input data barang dapat dilihat pada tabel dibawah ini :

| **Identifikasi** | |
|---|---|
| **Nomor** | 8 |
| **Nama** | Input Data Barang |
| **Tujuan** | Melakukan pemasukkan input data barang |
| **Deskripsi** | *User* memasukkan input data barang di koperasi |
| **Tipe** | |
| **Aktor** | *User* , Sistem Aplikasi *koperasi* |
| **Skenario Utama** | |
| **Kondisi awal** | *User* sudah terautentikasi |

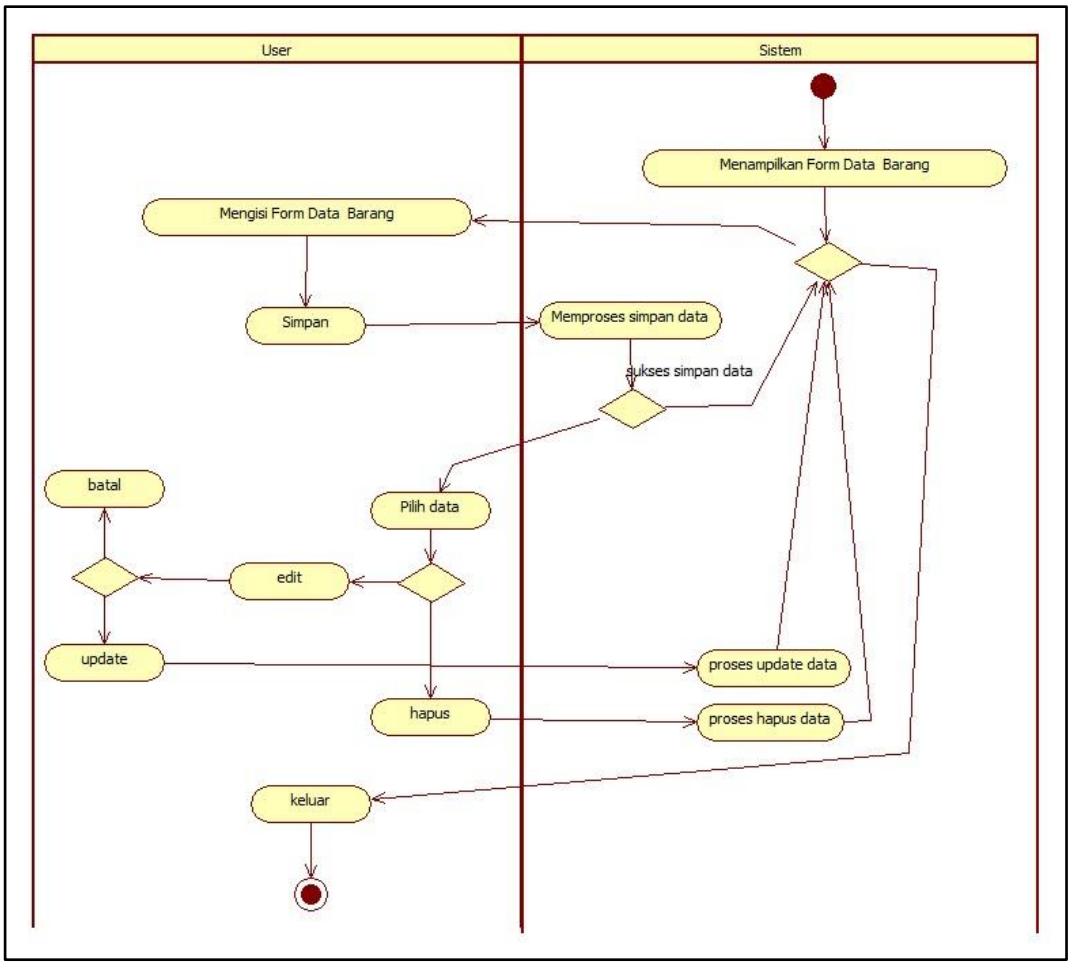
| **Aksi Aktor** | **Reaksi Sistem** |
|---|---|
| *User* memilih menu Transaksi ”input data barang” | |
| | Menampilkan *form* input data barang |
| *User* memasukkan data barang | |
| | Menampilkan tabel input data barang dalam *form* input data barang |
| Sistem memberikan respon | |

| | | |
|---|---|---|
| | | Memberitahu bahwa data yang dimasukkan sudah tersimpan |
| **Kondisi akhir** | Aplikasi *koperasi* menyimpan inputan data barang ke dalam *database* | |

*Tabel 11 Skenario Data Transaksi Pinjam*

### c. *Activity Diagram* Input Data Pembelian Barang

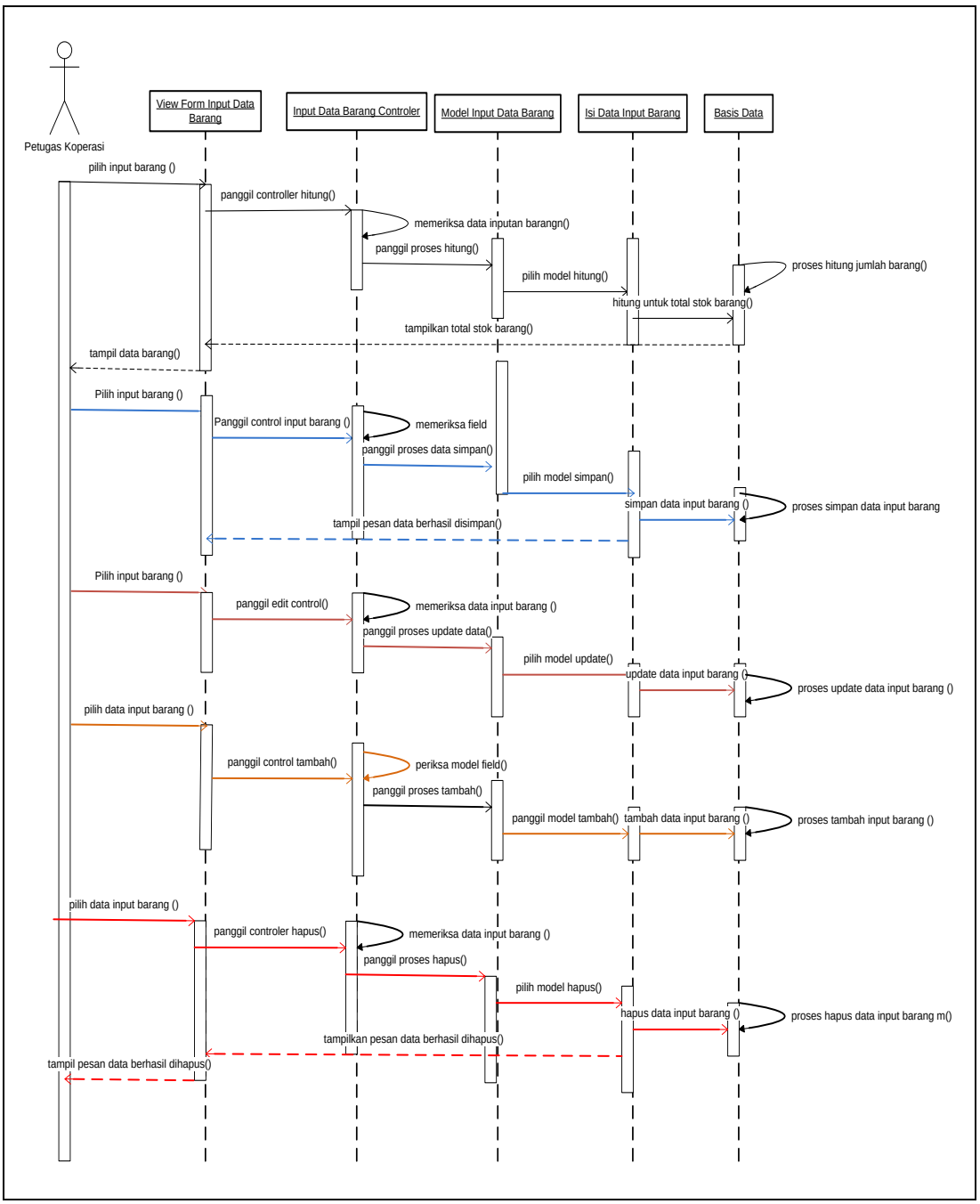
*Activity* diagram input data barang menggambarkan aliran kejadian (*flow of events*) pada saat user melakukan aktifitas *input* data barang kedalam sistem, untuk *actifity* diagram input data barang dapat di lihat pada gambar dibawah ini:




*Gambar 26 Activity Diagram Input Data Barang*

### d. *Sequence* Diagram Data Pembelian Barang

*Sequence* diagram input data barang digunakan untuk menunjukan aliran fungsionalitas dalam *use case* input data barang, untuk *sequence* diagram input data barang dapat dilihat pada gambar dibawah ini :

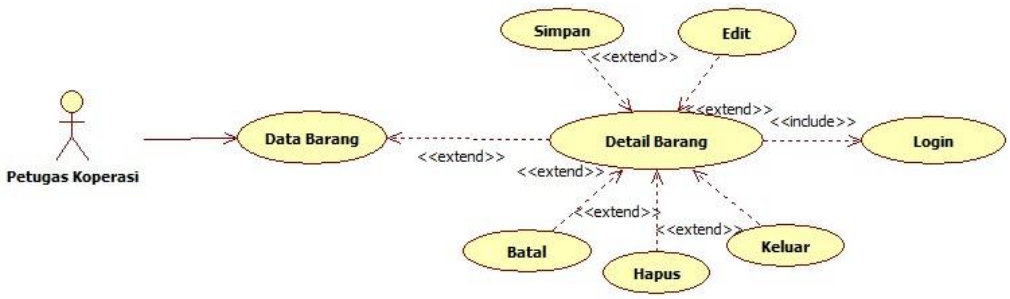



*Gambar 27 Sequence Diagram Input Data Barang*

### 3.3.9. Input Data Barang

#### *a. Use Case Input Data* Barang

*Use case* detail barang merupakan suatu proses untuk membatasi aktifitas antara admin dan *user*, jika melakukan *input* data detail barang harus *login* terlebih dahulu, kemudian pilih menu Transaksi kemudian pilih detail barang, maka akan muncul *form* detail barang, *petugas* memasukan identitas data detail barang dan *klik* tombol *simpan* jika telah selesai maka datanya akan ditampilkan dalam tabel detail barang, untuk gambar *use case* detail barang dapat dilihat pada gambar dibawah sebagai berikut :




*Gambar 28 Use Case Input Data Barang*

#### b. Skenario Input Data Barang

Skenario input detail barang merupakan suatu pengambaran langkah-langkah aksi aktor terhadap sistem untuk melakukan input detail barang, untuk skenario data detail barang dapat dilihat pada tabel dibawah ini :

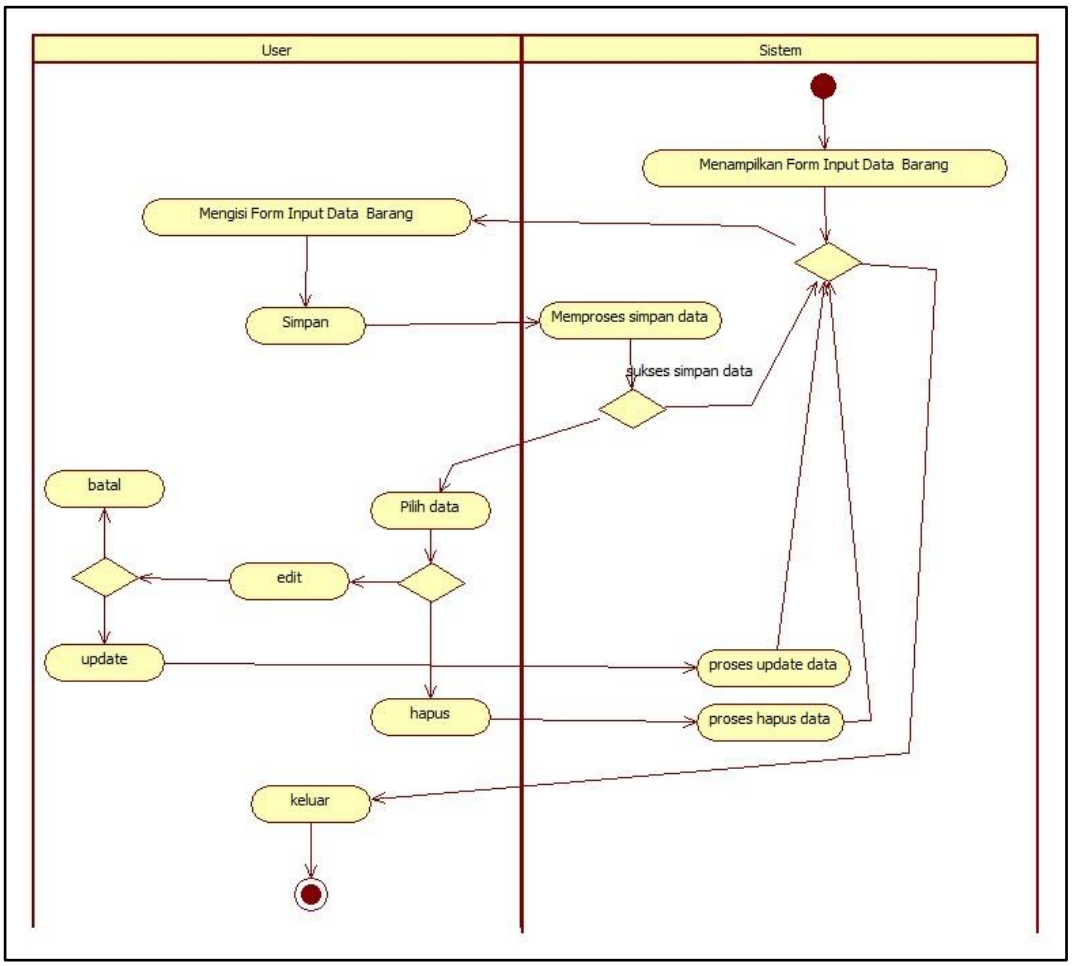
| **Identifikasi** | |
|---|---|
| **Nomor** | 9 |
| **Nama** | Input Detail Barang |
| **Tujuan** | Melakukan pemasukkan input detail barang |
| **Deskripsi** | *User* memasukkan input detail barang di koperasi |
| **Tipe** | |
| **Aktor** | *User* , Sistem Aplikasi *koperasi* |
| **Skenario Utama** | |
| **Kondisi awal** | *User* sudah terautentikasi |

| Aksi Aktor | | Reaksi Sistem |
|---|---|---|
| *User* memilih menu Transaksi ”detail barang” | | |
| | | Menampilkan *form* detail barang |
| *User* memasukkan data detail barang | | |
| | | Menampilkan tabel input data detail barang dalam *form* input data detail barang |
| Sistem memberikan respon | | |
| | | Memberitahu bahwa data yang dimasukkan sudah tersimpan |
| **Kondisi akhir** | Aplikasi *koperasi* menyimpan inputan data barang ke dalam *database* | |

*Tabel 12 Skenario Data Transaksi Pinjam*

### c. *Activity Diagram* Detail Barang

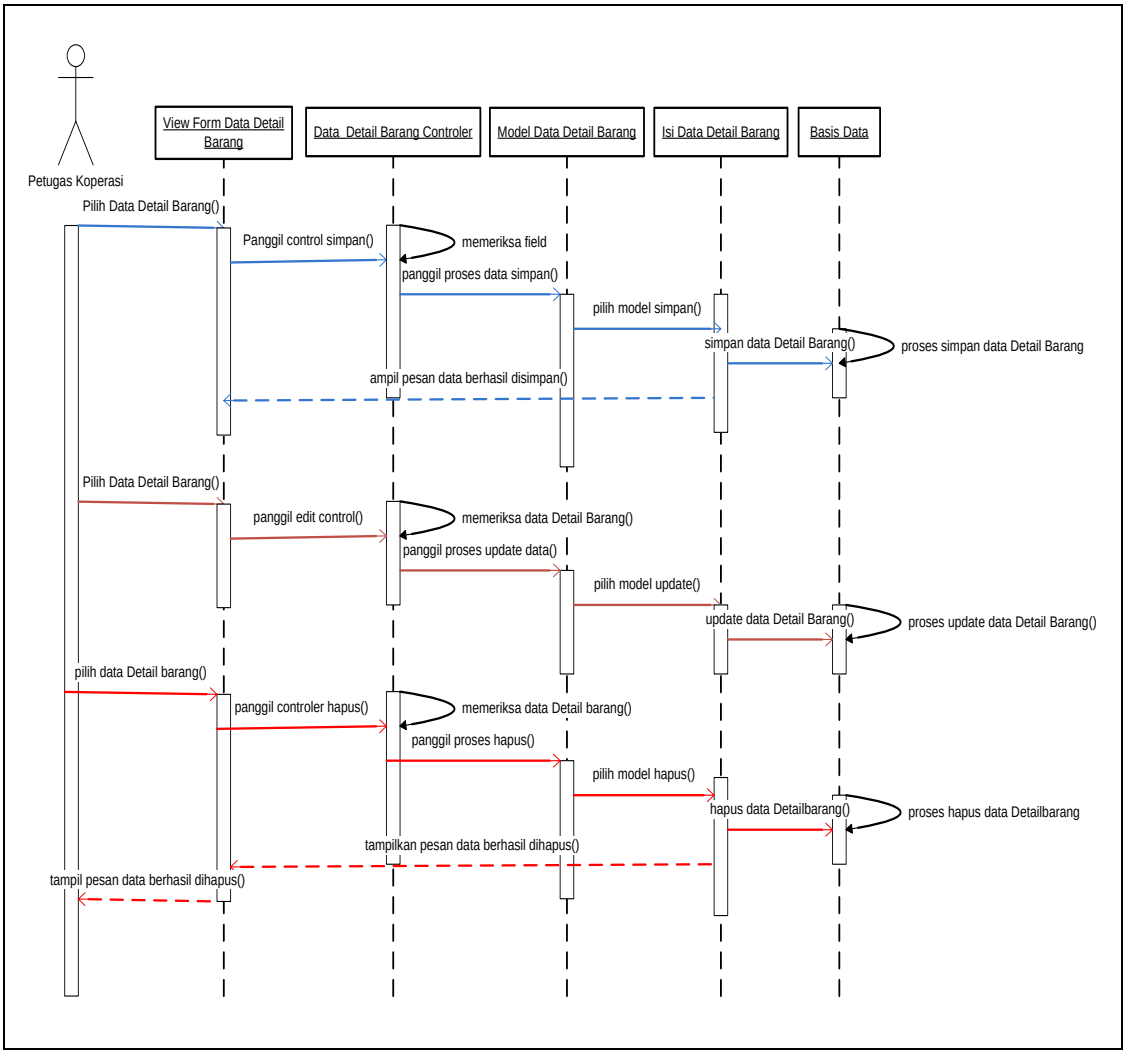
*Activity* diagram input detail barang menggambarkan aliran kejadian (*flow of events*) pada saat user melakukan aktifitas *input* detail barang kedalam sistem, untuk *actifity* diagram input data detail barang dapat di lihat pada gambar dibawah ini:




*Gambar 29 Activity Diagram Detail Barang*

### d. *Sequence* Diagram Detail Barang

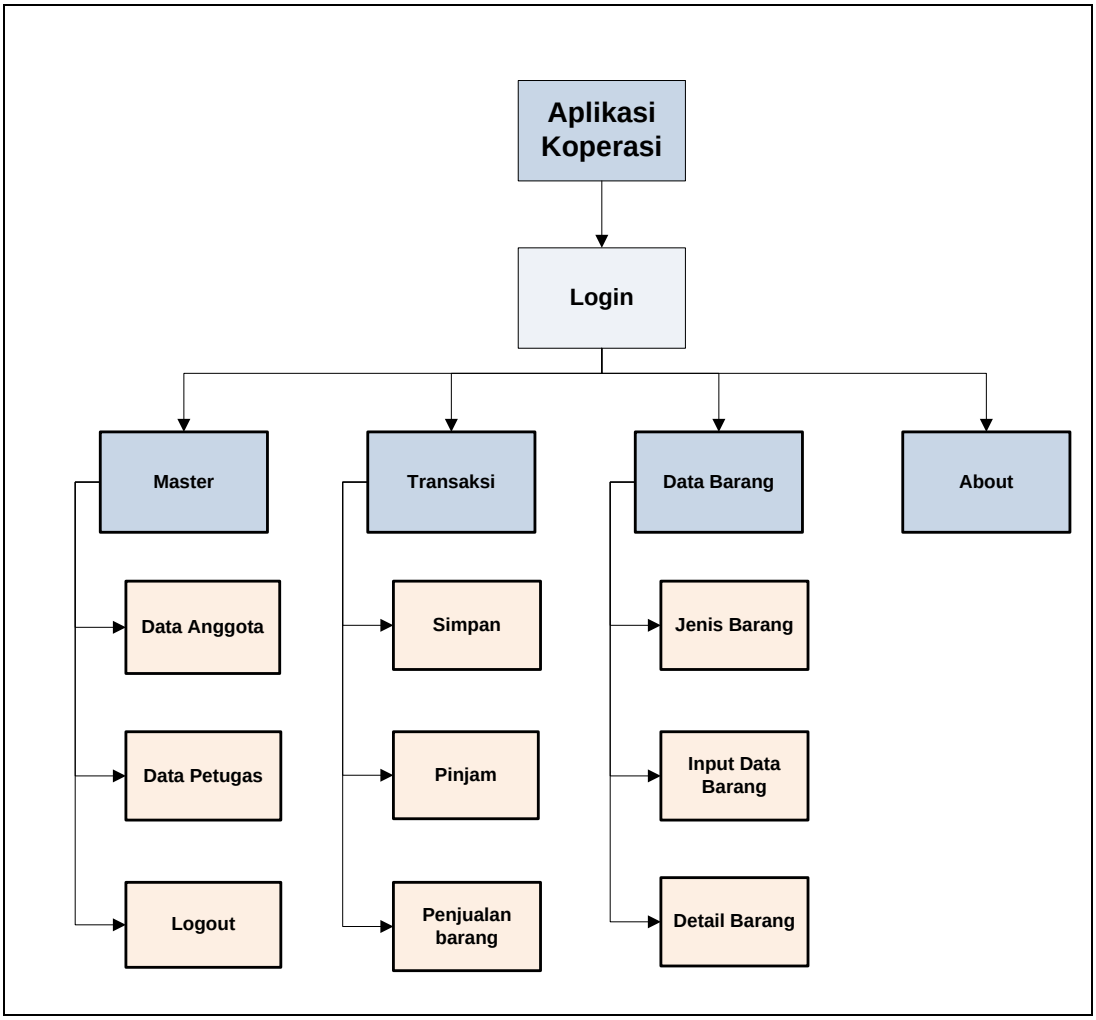
*Sequence* diagram input detail barang digunakan untuk menunjukan aliran fungsionalitas dalam *use case* input detail data barang, untuk *sequence* diagram input detail data barang dapat dilihat pada gambar dibawah ini :




*Gambar 30 Sequence Diagram Detail Barang*

## 3.4. Struktur Menu Program

Struktur menu program aplikasi *koperasi simpan pinjam reka mandiri* dapat dilihat pada gambar dibawah ini :




*Gambar 31 Struktur Menu Koperasi*

## 3.5. Perancangan Sistem

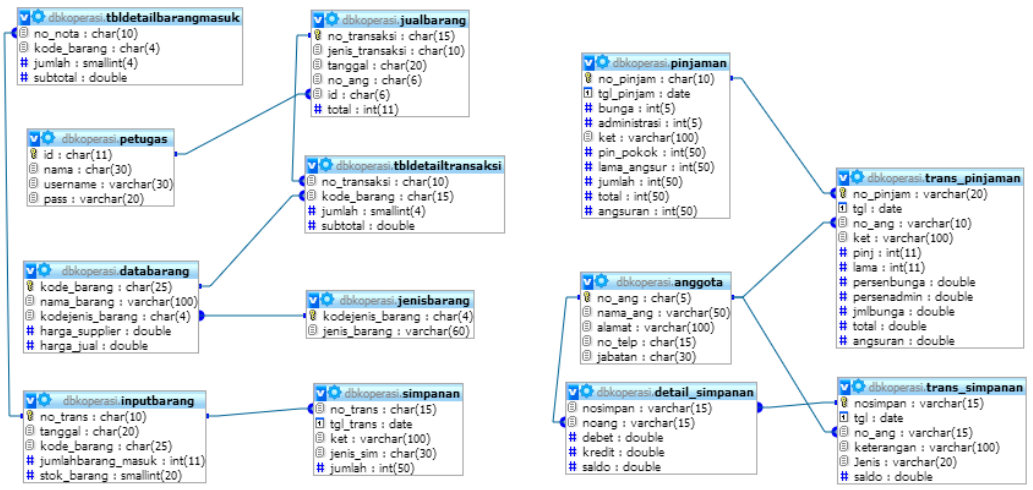
Perancangan akan dimulai setelah tahap analisis terhadap sistem selesai dilakukan. Perancangan dapat didefinisikan sebagai proses aplikasi berbagai teknik dan prinsip bagi tujuan pendefinisian suatu perangkat, suatu proses atau sistem dalam detail yang memadai untuk memungkinkan realisasi fisiknya. Perancangan digambarkan sebagai proses multi-langkah dimana representasi perancangan *data*, skema relasi, struktur *file*, perancangan antar muka, dan perancangan prosedural dari persyaratan informasi.

### 3.5.1. Perancangan *Data*

Perancangan *data* atau lebih dikenal dengan perancangan basis *data* yaitu menciptakan atau merancang *data* yang terhubung dan disimpan secara bersama-sama. Untuk menggambarkannya digunakanlah skema relasi dan struktur *file*. Dari dua hasil tersebut, implementasi basis *data* akan bisa dikerjakan.

**a. Skema Relasi**

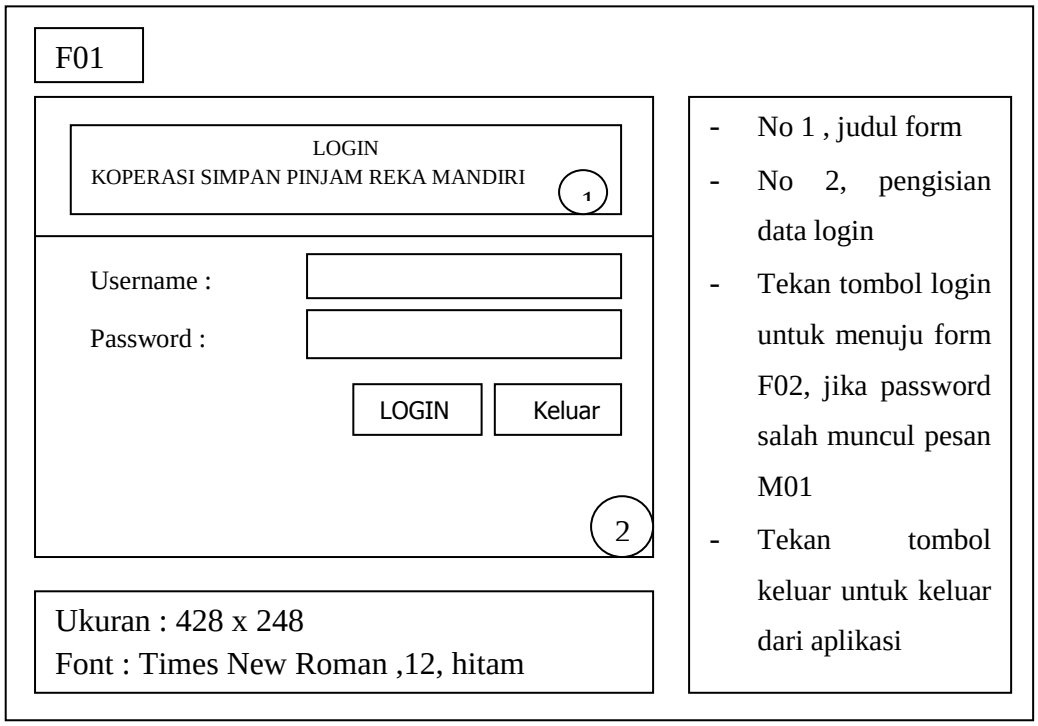
Model *data* relasional merupakan model *data* di mana hubungan antar *data*,arti *data* dan batasannya dijelaskan dengan baris dan kolom. Secara formal, kesemuanya itu digambarkan ke dalam skema relasi dan diagram skema. Adapun skema relasi yang terdapat didalam aplikasi ini adalah sebagai berikut:




*Gambar 32 Skema Relasi Database*

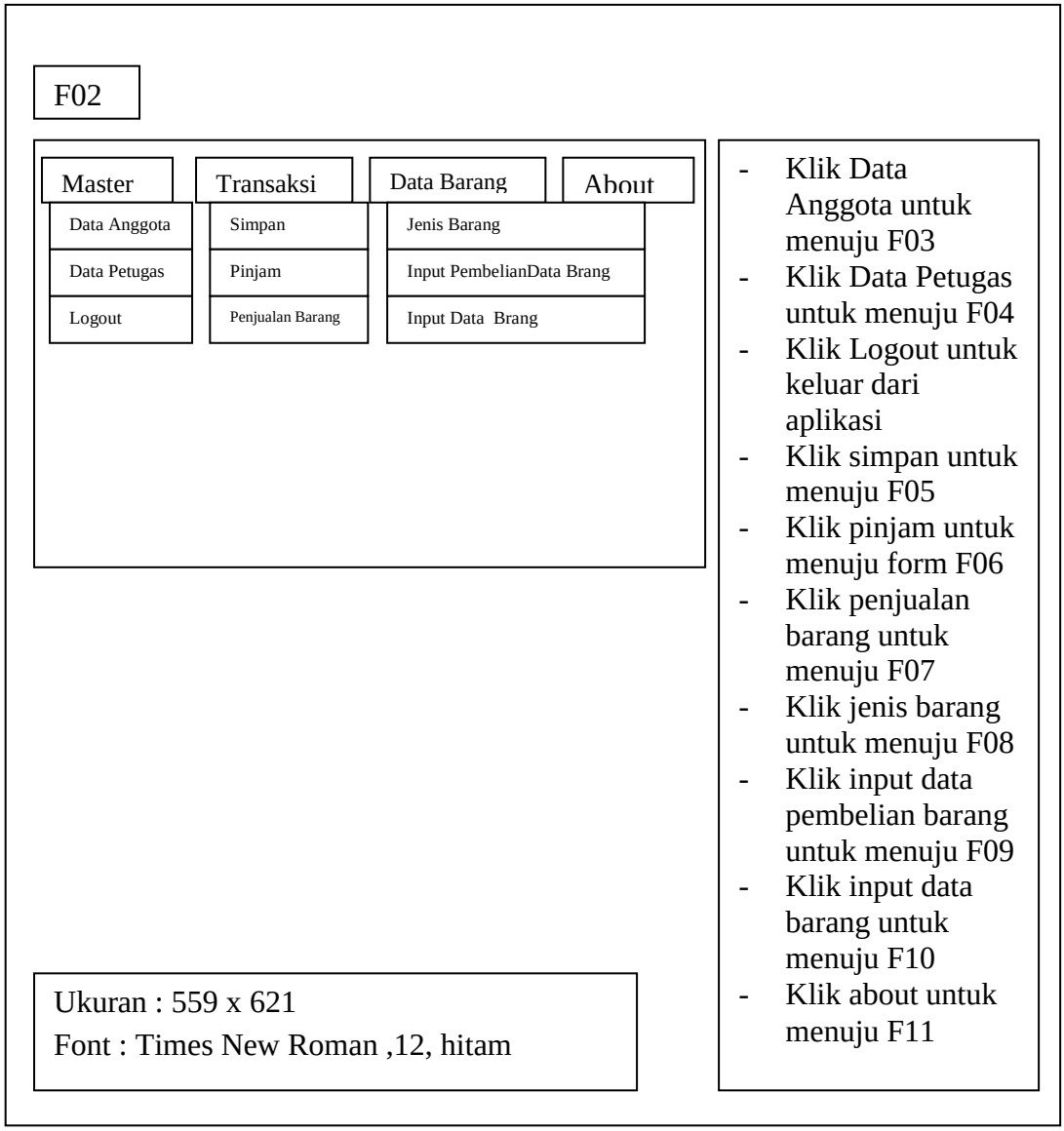
### 3.5.2. Perancangan *Form* Perangkat Lunak

a. Perancangan antarmuka form login



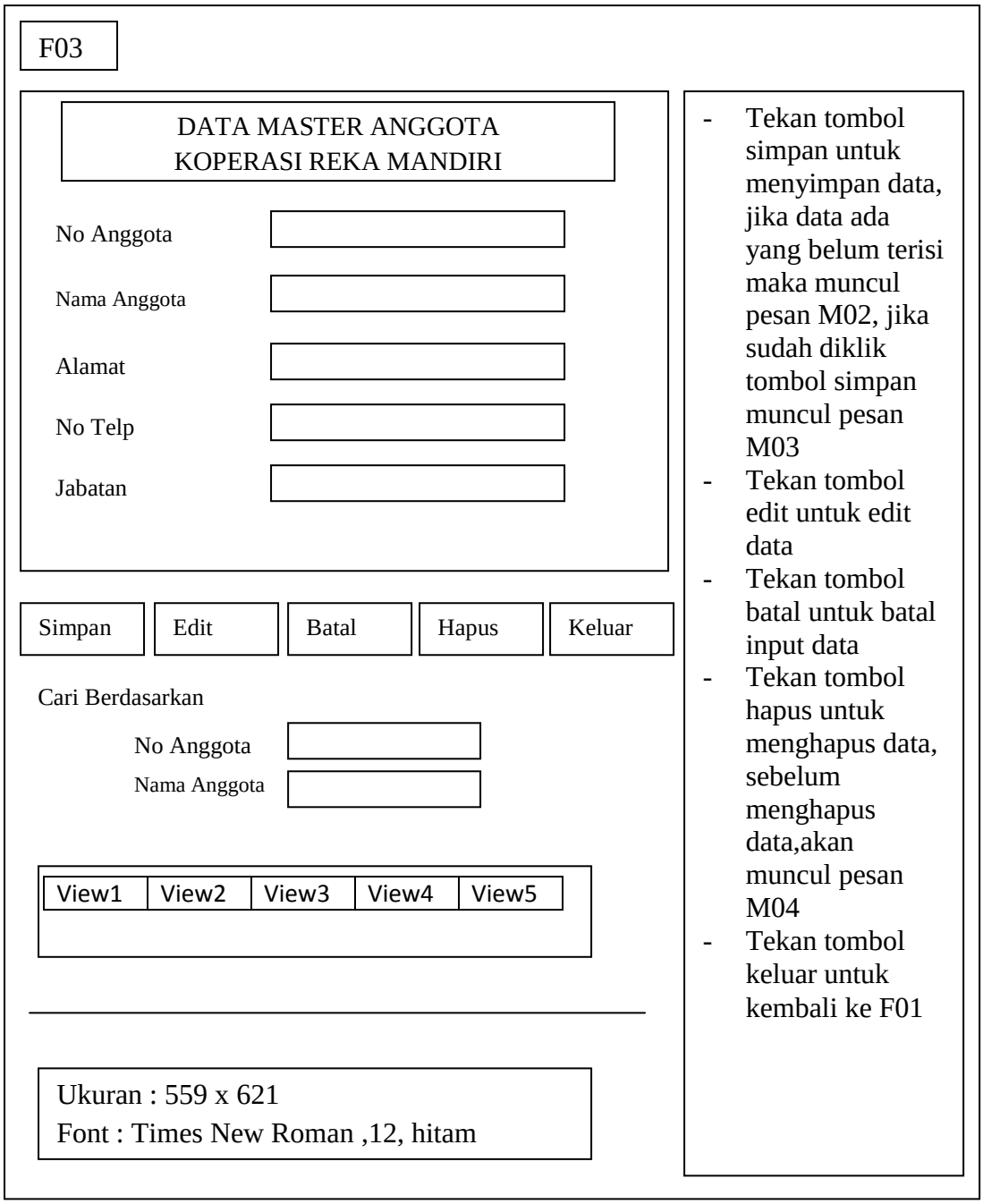

*Gambar 33 Perancangan antarmuka Login*

b. Perancangan antarmuka halaman utama koperasi



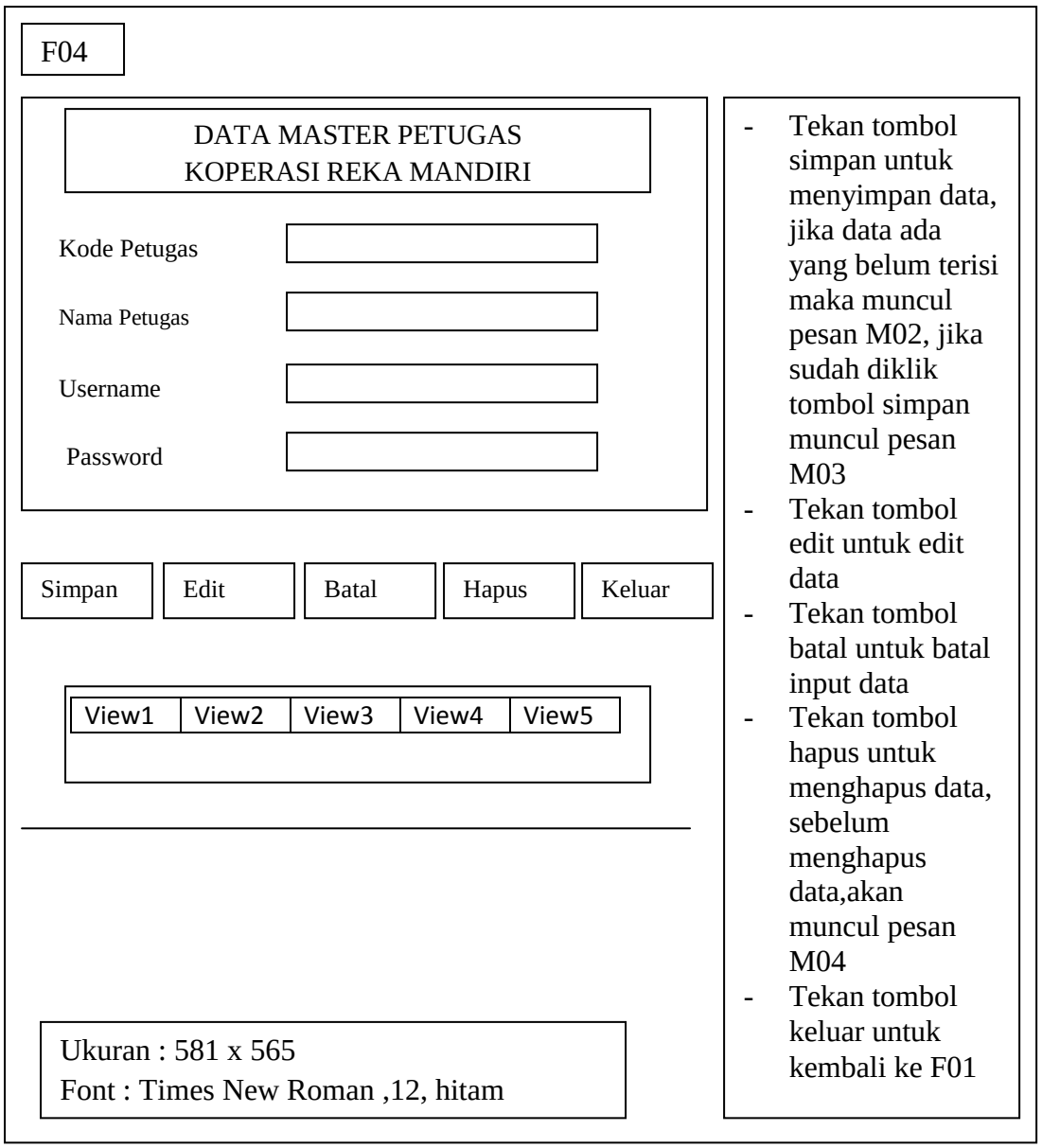

*Gambar 34 Perancangan antarmuka halaman utama*

d. perancangan antarmuka form Data Anggota



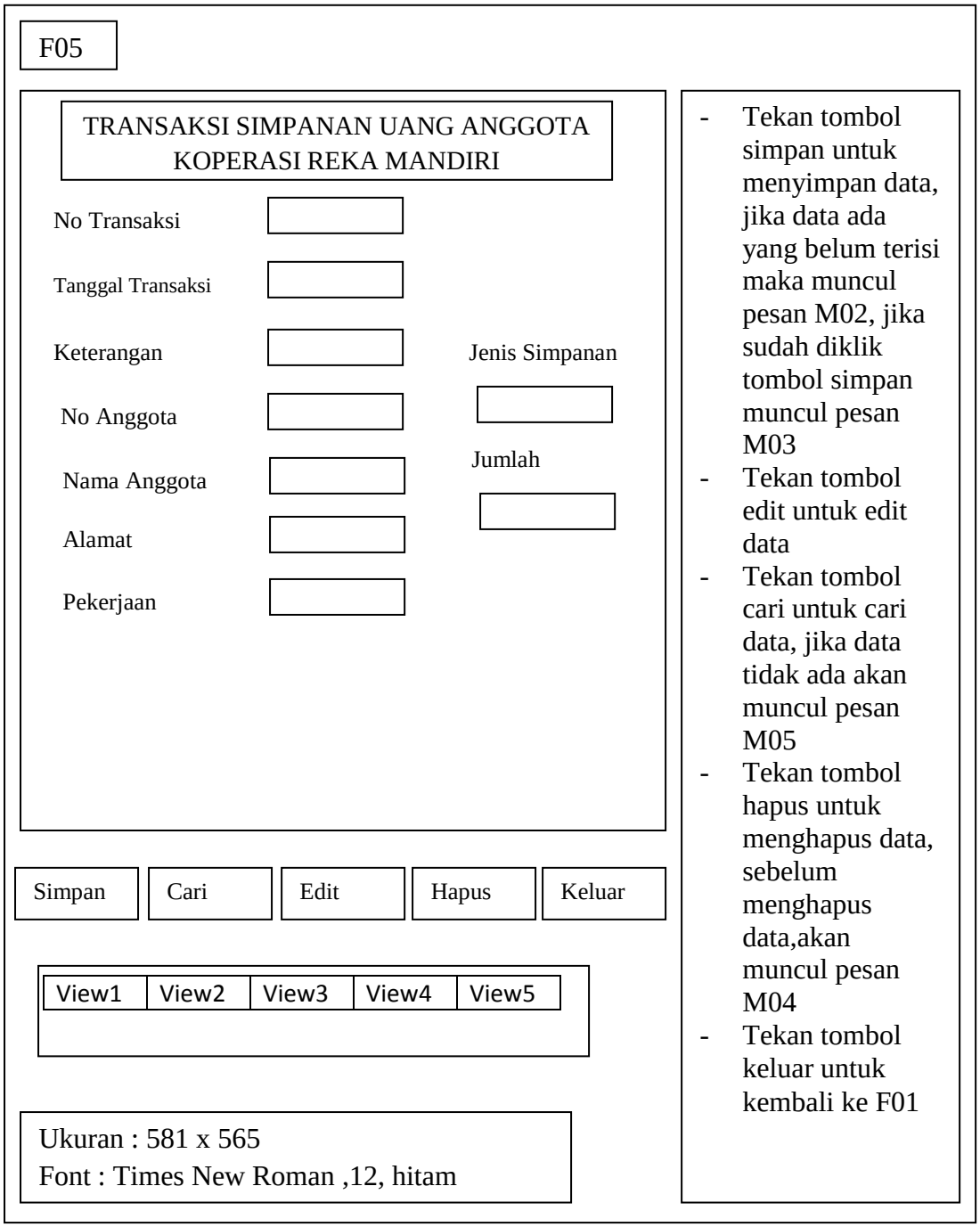

*Gambar 35 Perancangan antarmuka form Data anggota*

e. Perancangan antarmuka form data petugas



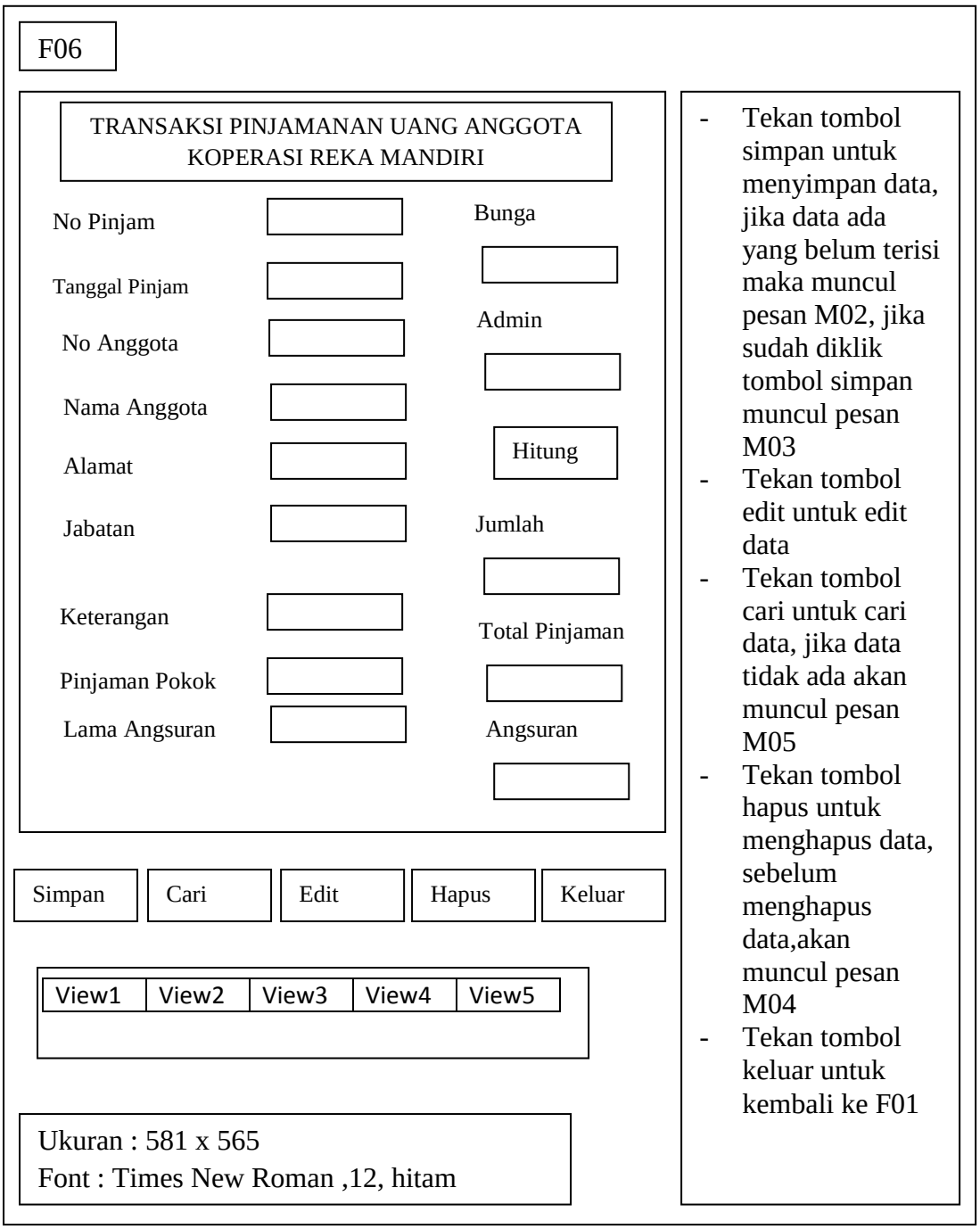

*Gambar 36 Perancangan antarmuka Form Data Petugas*

f. Perancagan antarmuka transaksi simpan



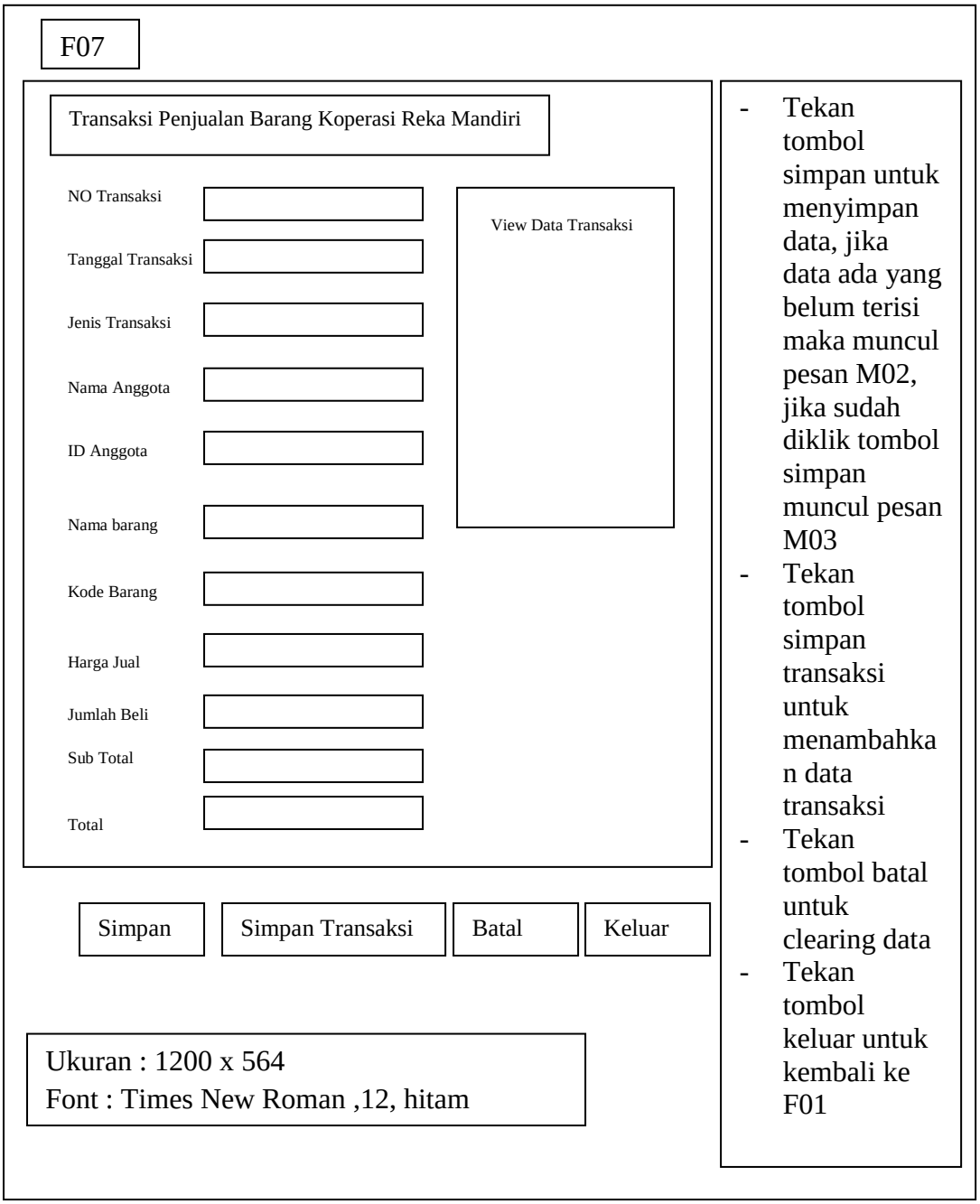

*Gambar 37 Perancangan antarmuka Form Transaksi Simpan*

g. Perancangan antarmuka transaksi pinjam



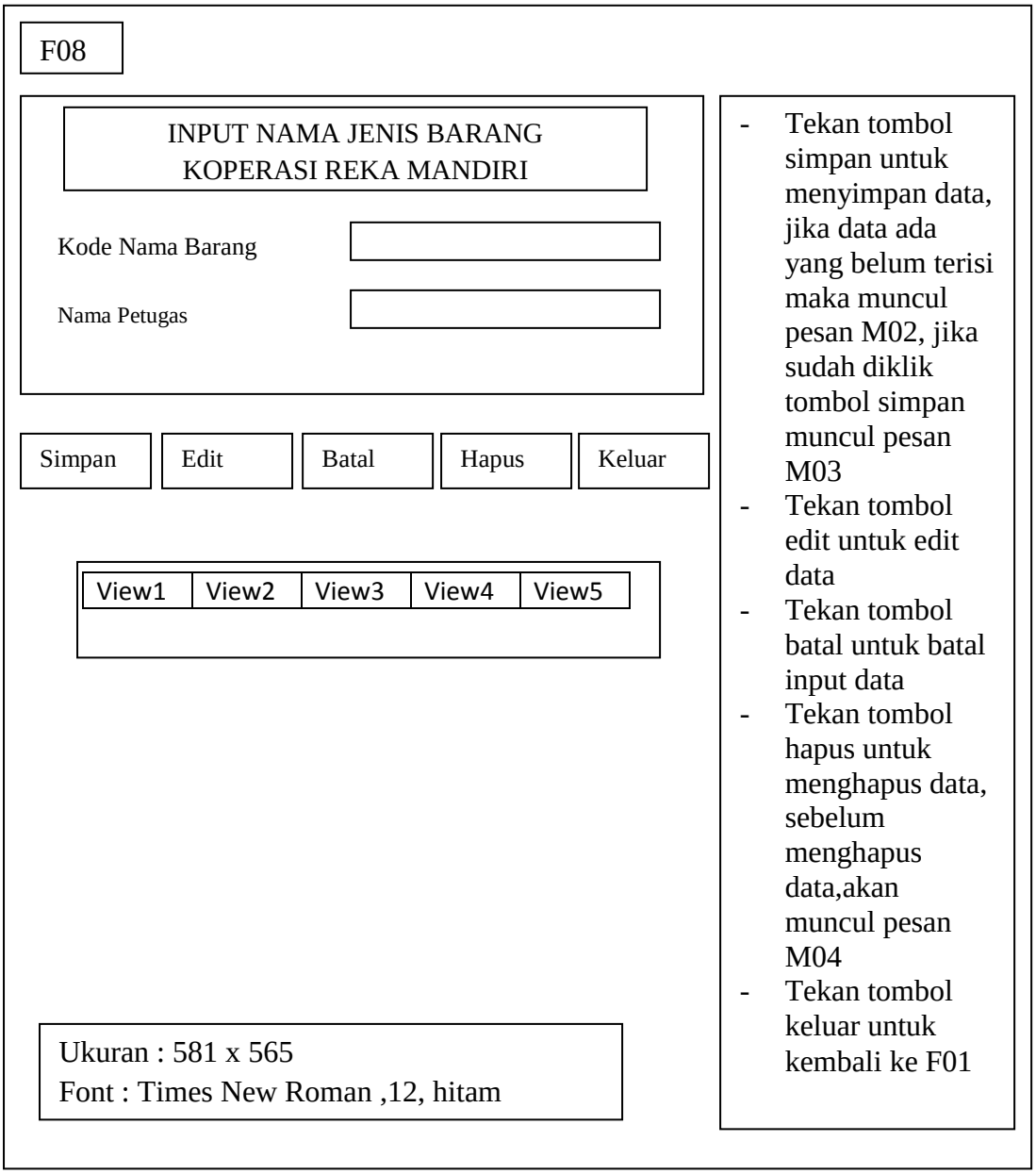

*Gambar 38 Perancangan Antarmuka Form Pinjam Anggota*

h. Perancangan antarmuka transaksi penjualan barang



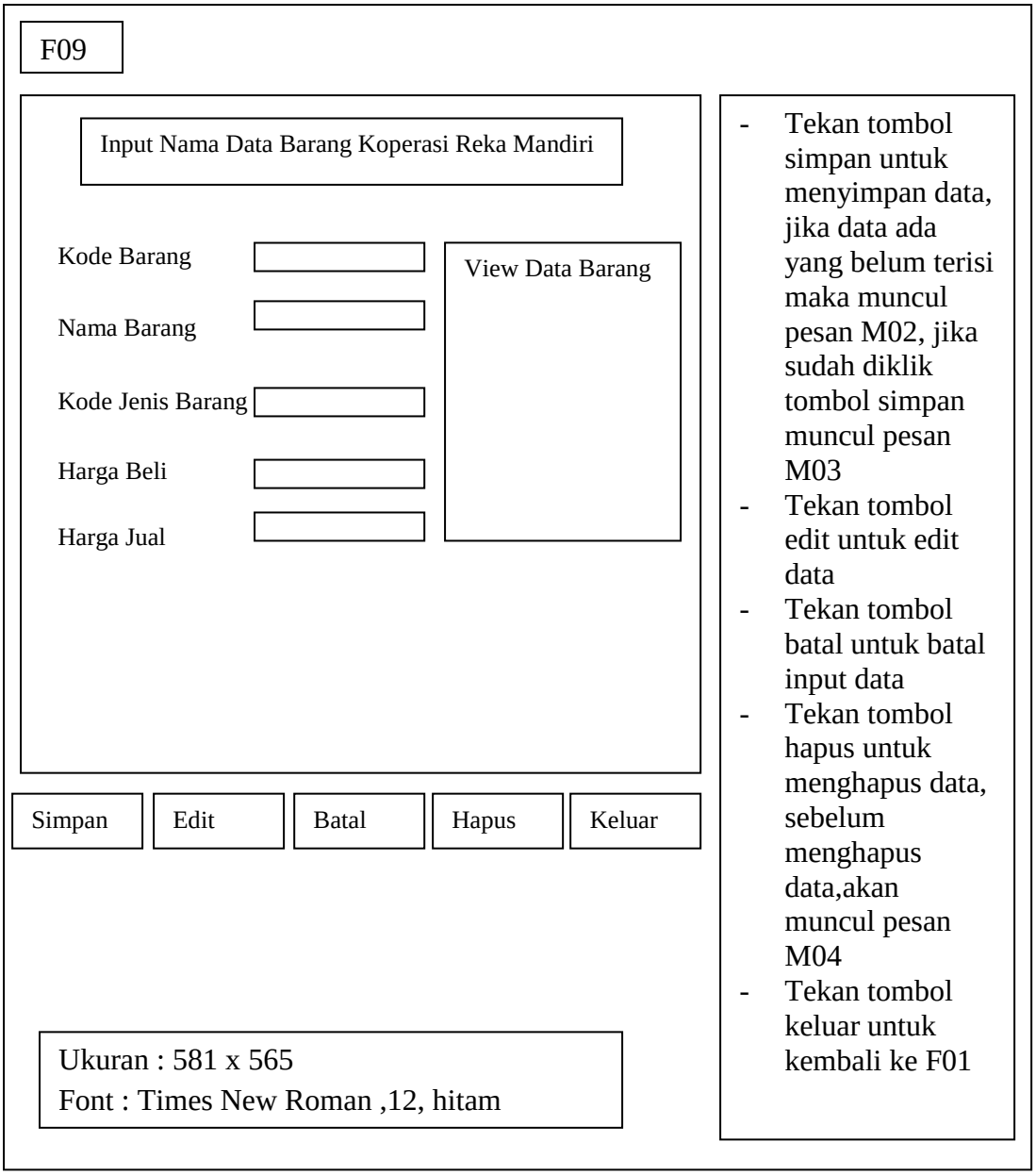

*Gambar 39 Perancangan Antarmuka Form Penjualan Barang*

i. Perancangan antarmuka form Jenis Barang



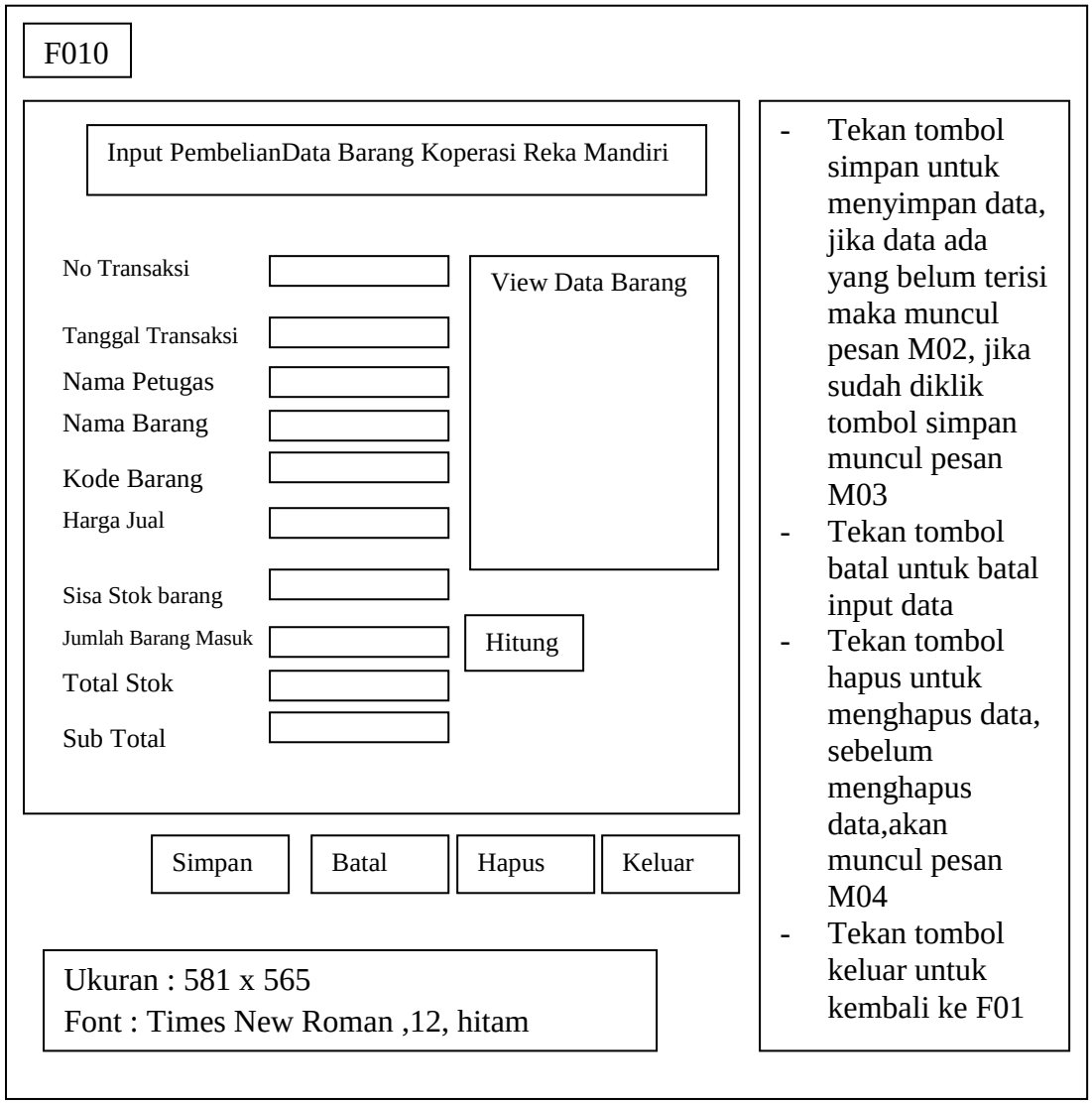

*Gambar 40 Perancangan antarmuka form Jenis Barang*

j. Perancangan antarmuka form input data barang



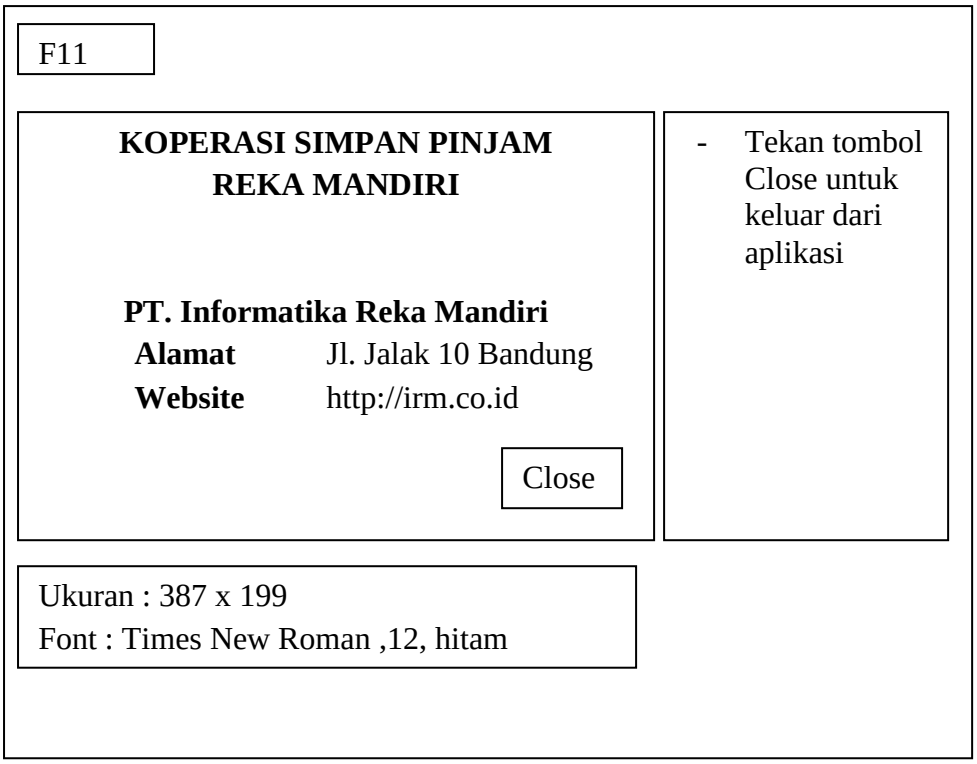

*Gambar 41 Perancangan Antarmuka Form Data Barang*

k. Perancangan antarmuka form pembelian barang




*Gambar 42 Perancangan Antarmuka Form Input Pembelian Data Barang*

l. Perancangan antarmuka form about




*Gambar 43 Perancangan antarmuka form about*

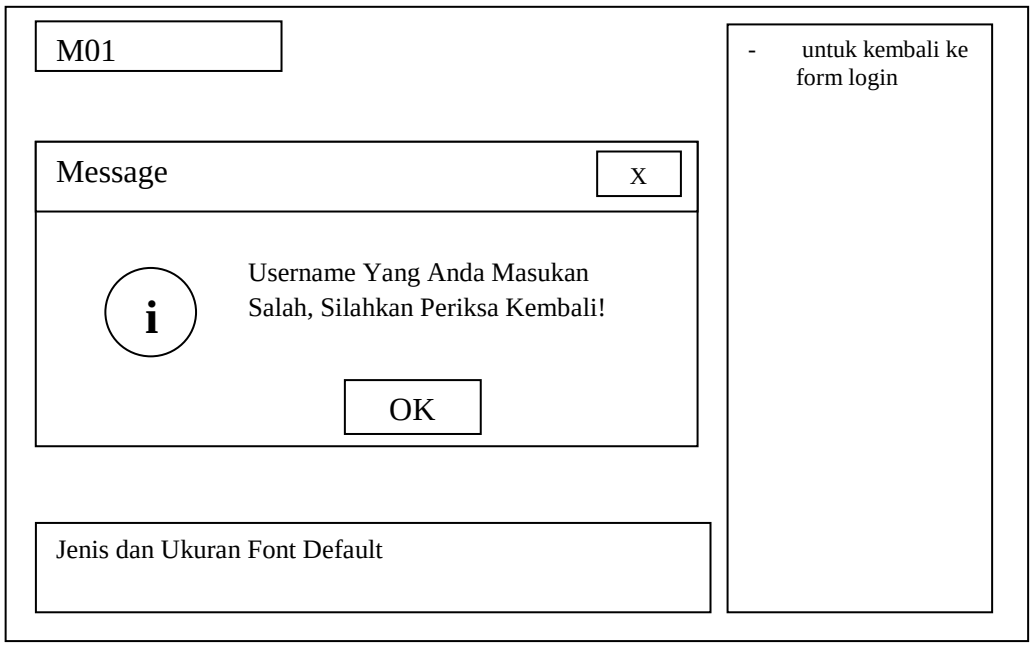
### 3.5.3. Perancangan Pesan

Perancangan pesan merupakan suatu perancangan tampilan pesan yang berupa informasi jika user menggunakan aplikasi koperasi, berikut ini adalah perancangan pesan-pesan informasi yang yang ada dalam aplikasi *koperasi* :

1. Pesan *Form Login*

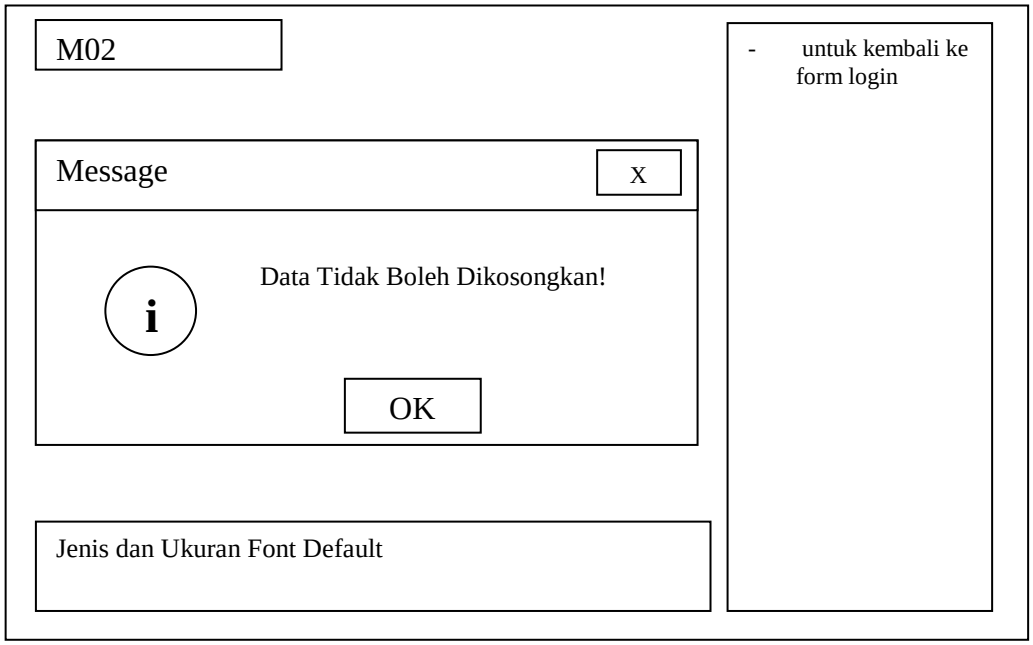
   Pesan form login merupakan suatu pesan yang berupa informasi pada saat *user* menggunakan *form login*, berikut ini adalah tampilan pesan dalam *form login* :

A. Pesan M01 akan muncul apabila *user* salah memasukan *username* atau salah memasukan *password*, berikut ini adalah perancangan pesan M01 :



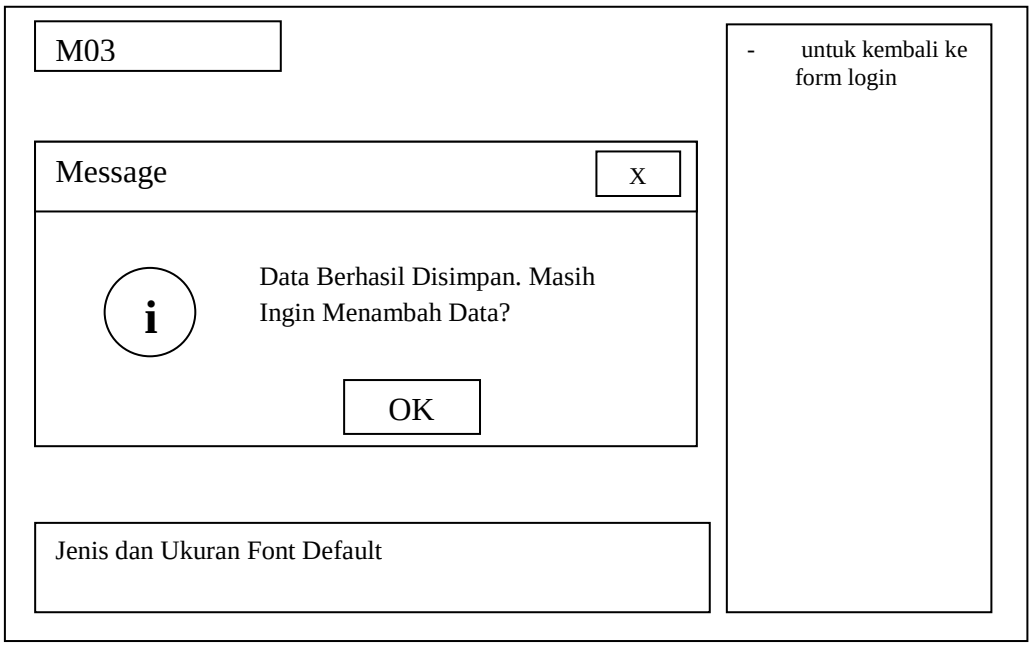

*Gambar 44 Perancangan Pesan M01*

B. Pesan M02 akan muncul apabila *user* tidak mengisi salah satu field ketika tombol Simpan diklik, berikut ini adalah perancangan pesan M02 :



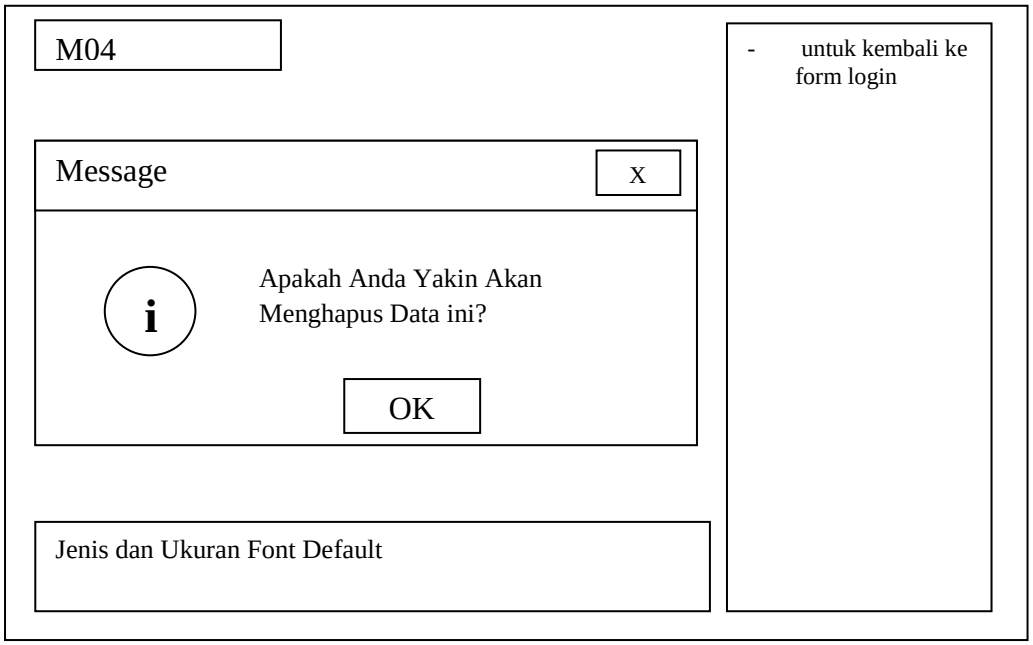

*Gambar 45 Perancangan Pesan M02*

C. Pesan M03 akan muncul apabila *user* telah mengisi semua field dan menekan tombol simpan, berikut ini adalah perancangan pesan M03 :



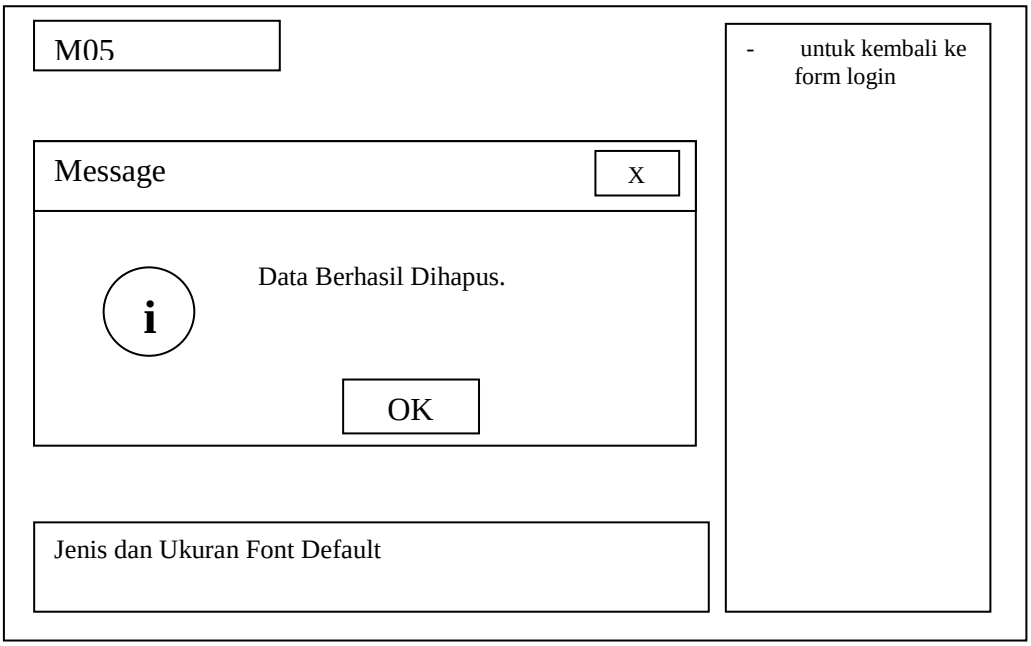

*Gambar 46 Perancangan Pesan M03*

D. Pesan M04 akan muncul apabila *user* ingin menghapus data dari database, berikut ini adalah perancangan pesan M04 :



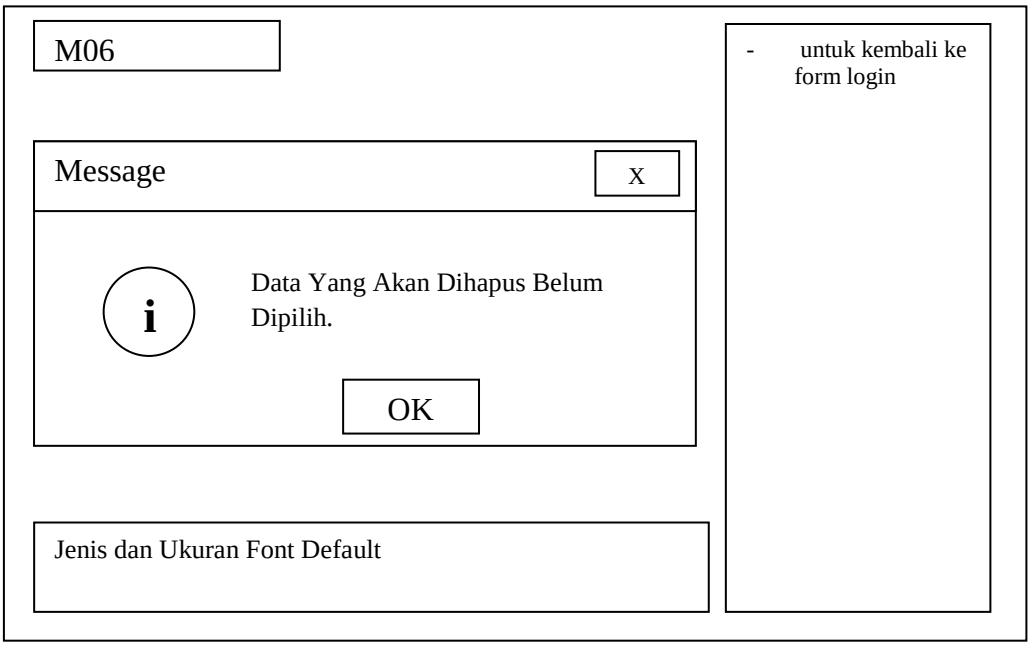

*Gambar 47 Perancangan Pesan M04*

E. Pesan M05 akan muncul apabila *user* telah meng-klik tombol hapus, menandakan data berhasil dihapus dari database, berikut ini adalah perancangan pesan M05 :



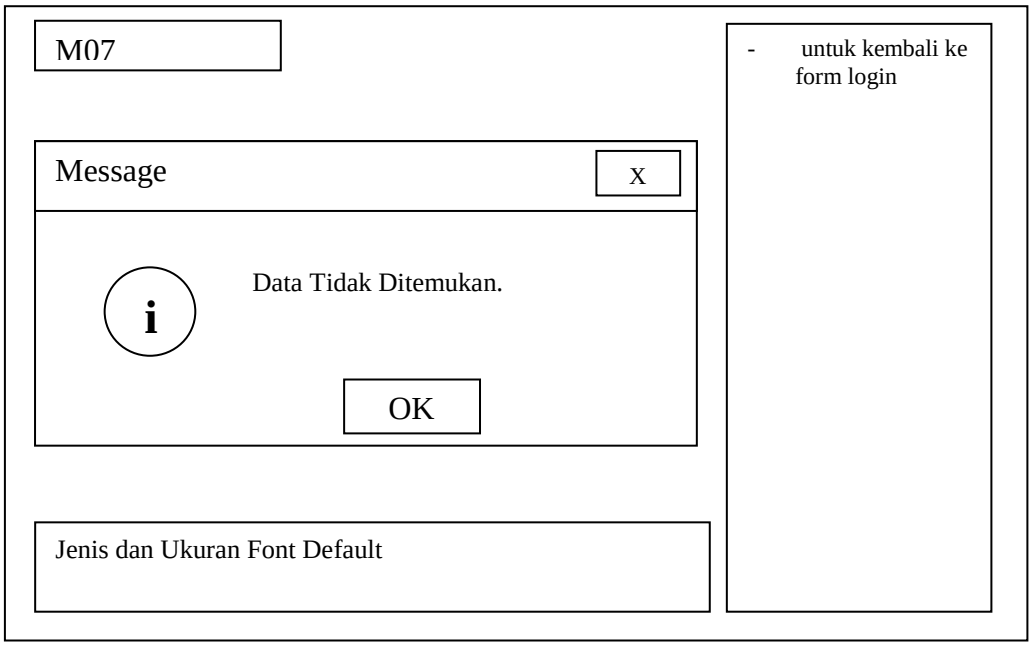

*Gambar 48 Perancangan Pesan M05*

F. Pesan M06 akan muncul apabila *user* telah meng-klik tombol hapus, namun belum memilih data dari database, berikut ini adalah perancangan pesan M06 :



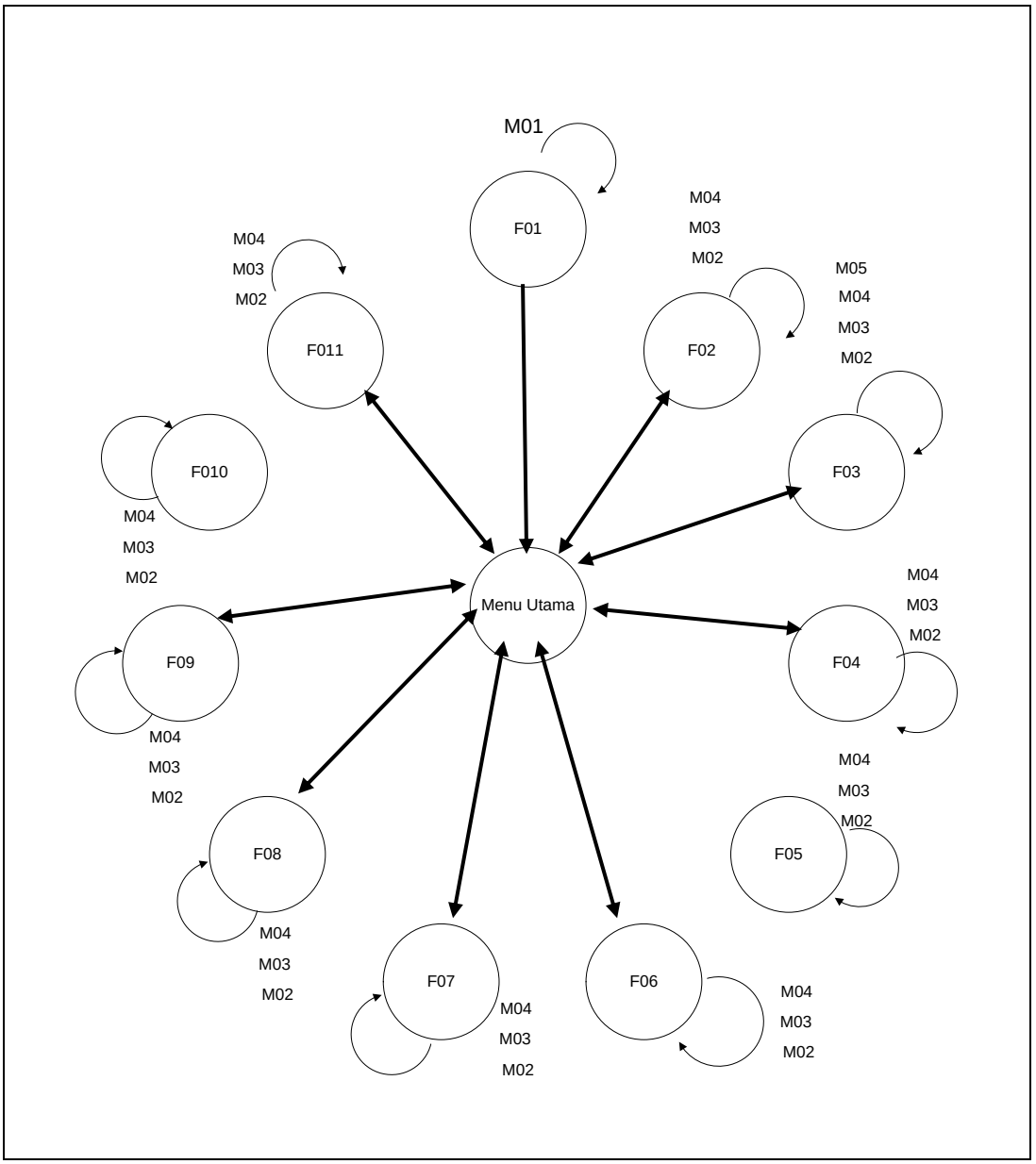

*Gambar 49 Perancangan Pesan M06*

G. Pesan M07 akan muncul apabila *user* mencari data, namun field untuk mencari data tidak isi atau data tidak ada di database, berikut ini adalah perancangan pesan M07 :




*Gambar 50 Perancangan Pesan M07*

### 3.5.4. Jaringan Semantik



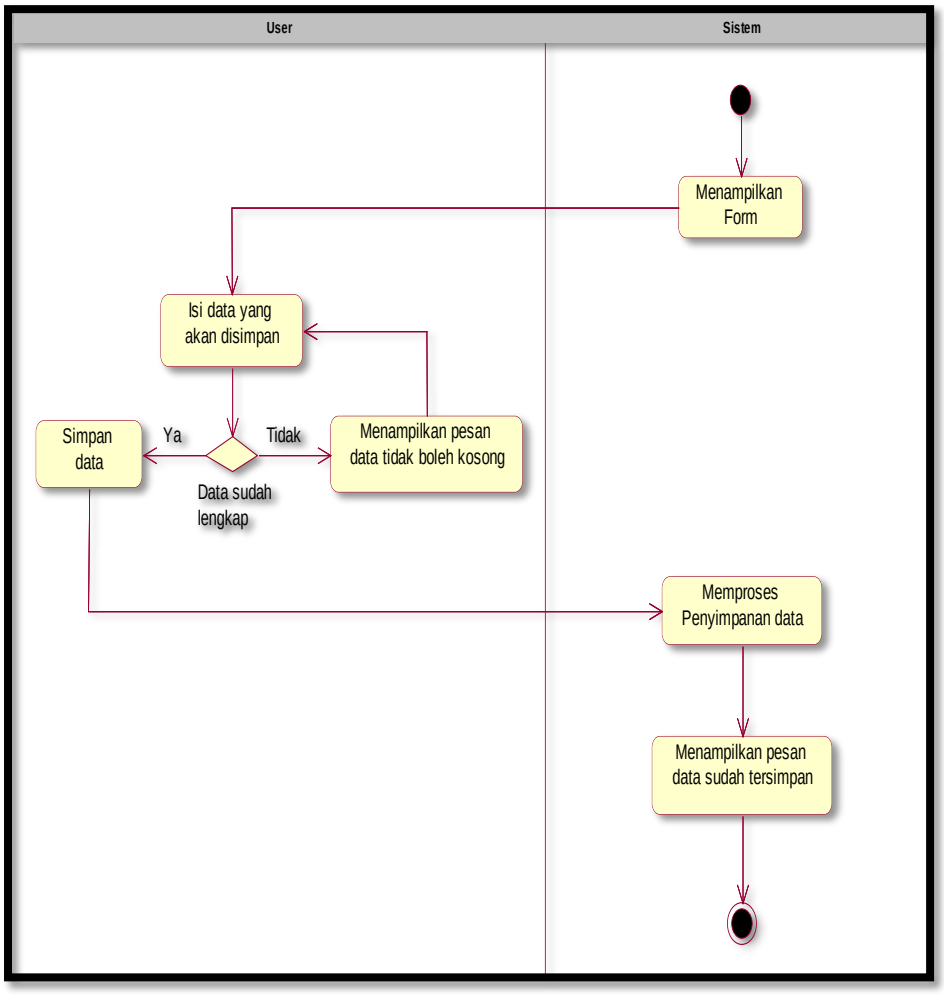

*Gambar 51 Jaringan Semantik Kopeasi SImpan Pinjam*

### 3.5.5. Perancangan Prosedural

Perancangan prosedural mentransformasi elemen-elemen objek dari arsitektur program ke dalam suatu deskripsi prosedural dari komponen-komponen perangkat lunak. Adapun perancangan prosedural untuk aplikasi *koperasi* yang akan dibangun adalah sebagai berikut:

1. Prosedural *Simpan*

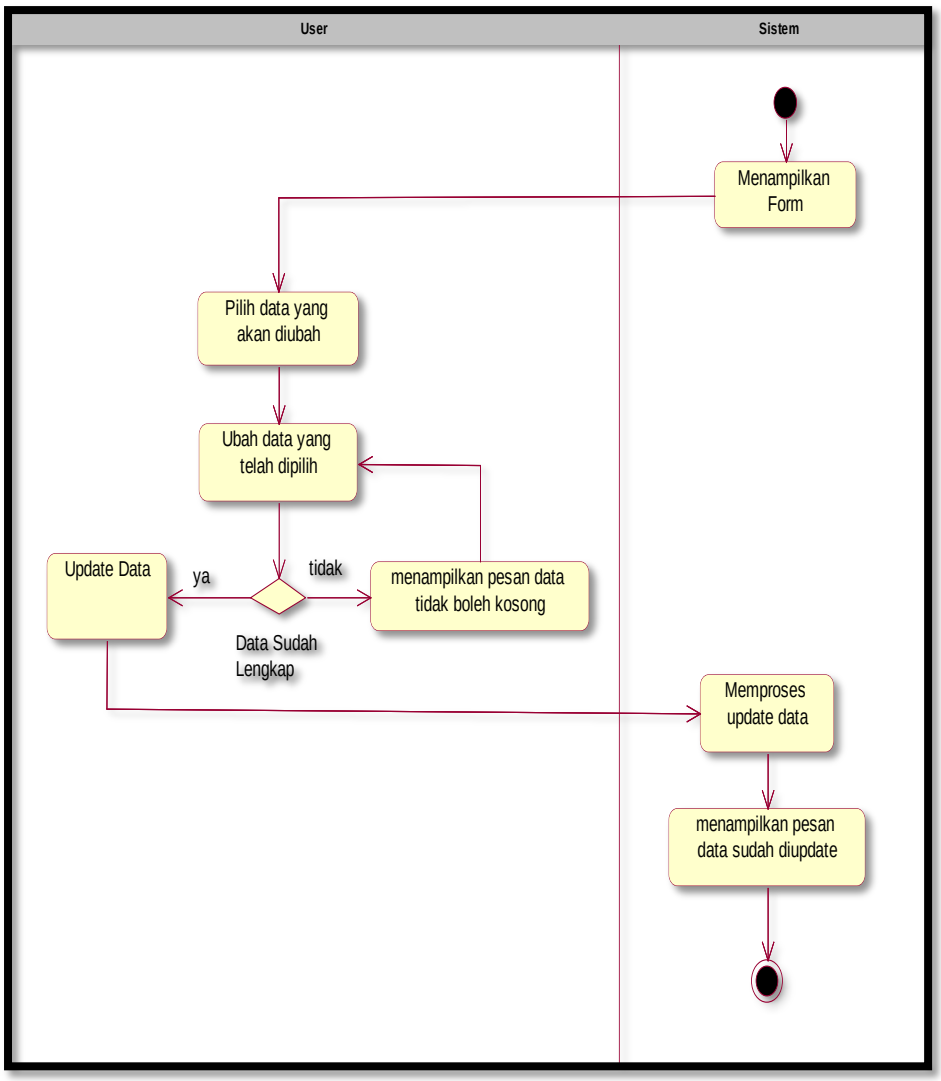
   Prosedur ini dilakukan ketika pengguna akan melakukan penyimpanan data, prosedural simpan dapat dilihat pada gambar di bawah ini :




*Gambar 52 Prosedural Simpan*

2. Prosedural *Edit*

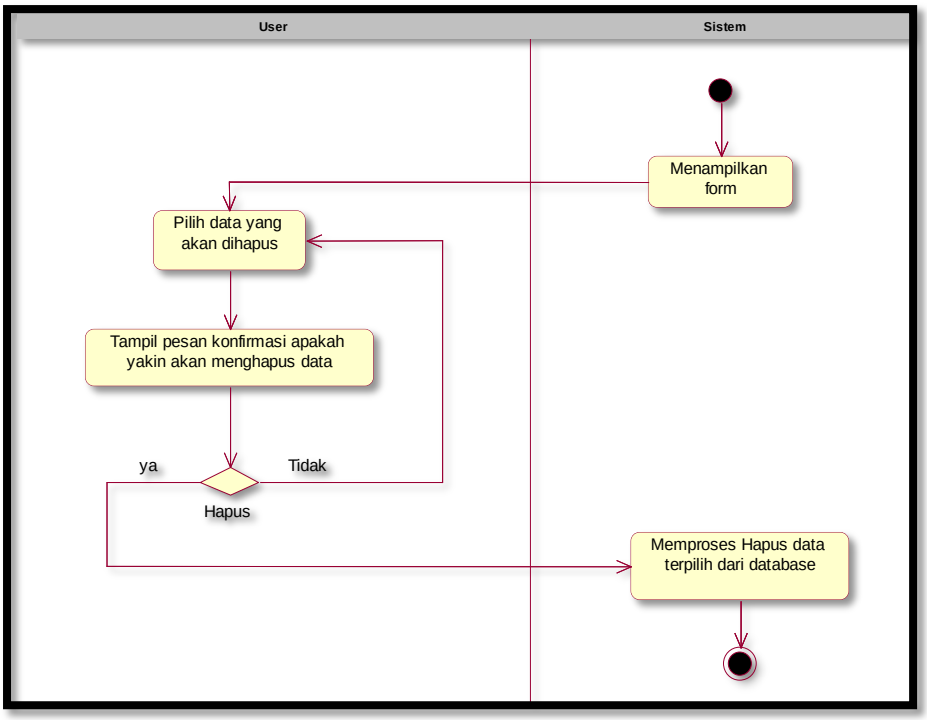
Prosedur ini dilakukan ketika pengguna akan melakukan perubahan data, prosedural edit dapat dilihat pada gambar di bawah ini :




*Gambar 53 Prosedural Edit*

3. Prosedural *Hapus*

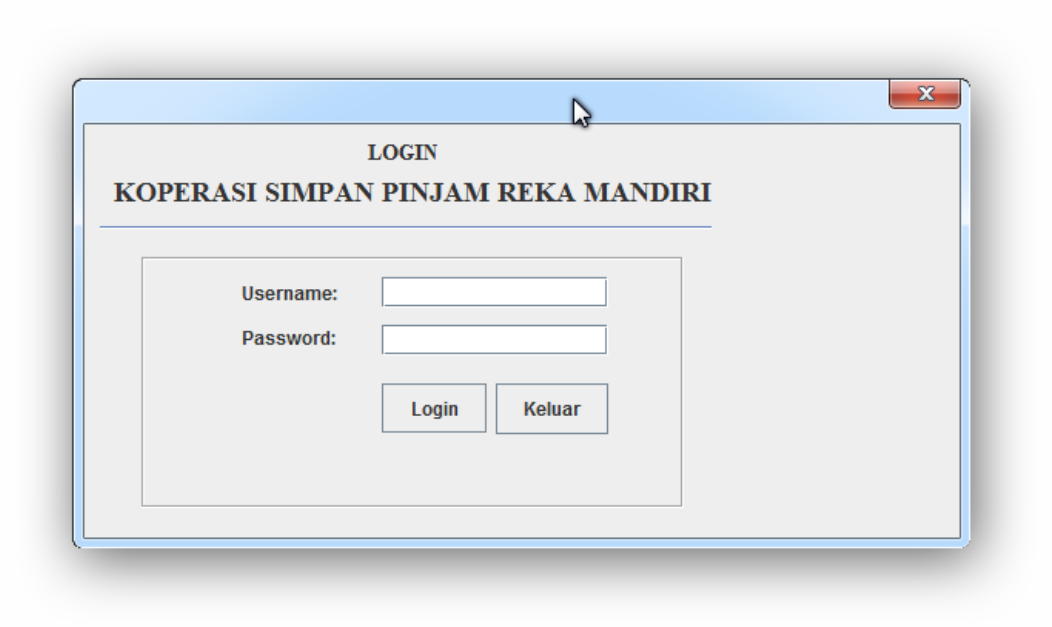
Prosedur ini dilakukan ketika *user* akan melakukan penghapusan data. Prosedural hapus dapat dilihat pada gambar sebagai berikut :




*Gambar 54 Prosedural Hapus*

## 3.6. Implementasi Perancangan Antarmuka

A. Implementasi antarmuka Login




*Gambar 55 Implementasi Login*

B. Implementasi Halman Utama




*Gambar 56 Implementasi Halaman Utama*

C. Implementasi Menu Master




*Gambar 57 Implementasi Menu Master*

D. Implementasi Menu Transaksi



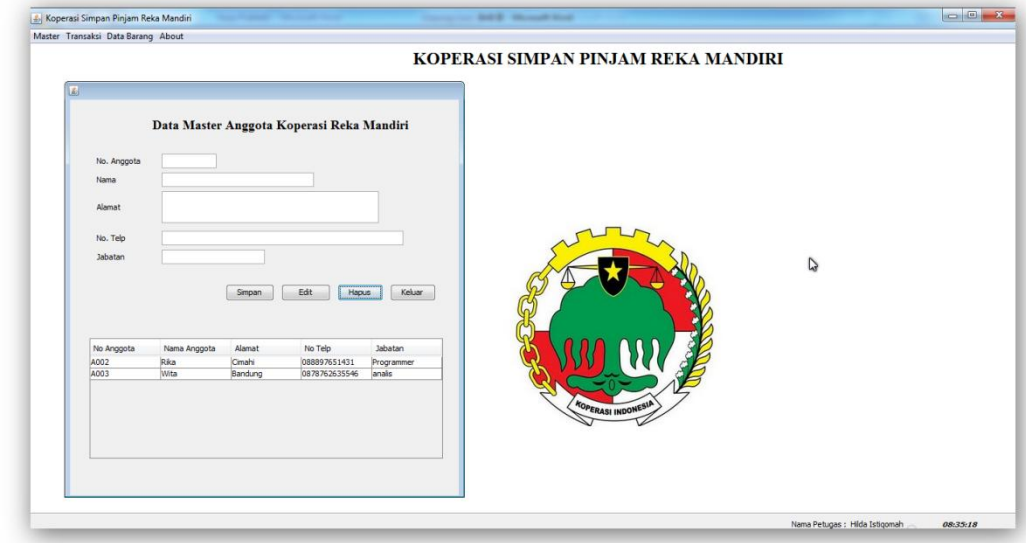

*Gambar 58 Implementasi Menu Transaksi*

E. Implementasi Menu Data Barang




*Gambar 59 Implementasi Menu Data Barang*

F. Implementasi Menu Data Anggota



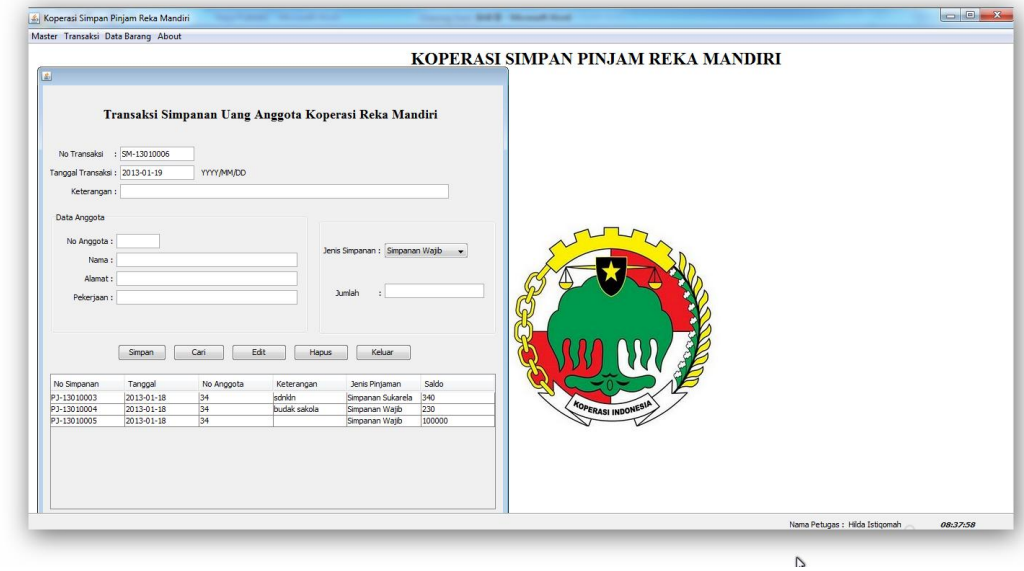

*Gambar 60 Implementasi Menu Data Anggota*

G. Implementasi Menu Data Petugas



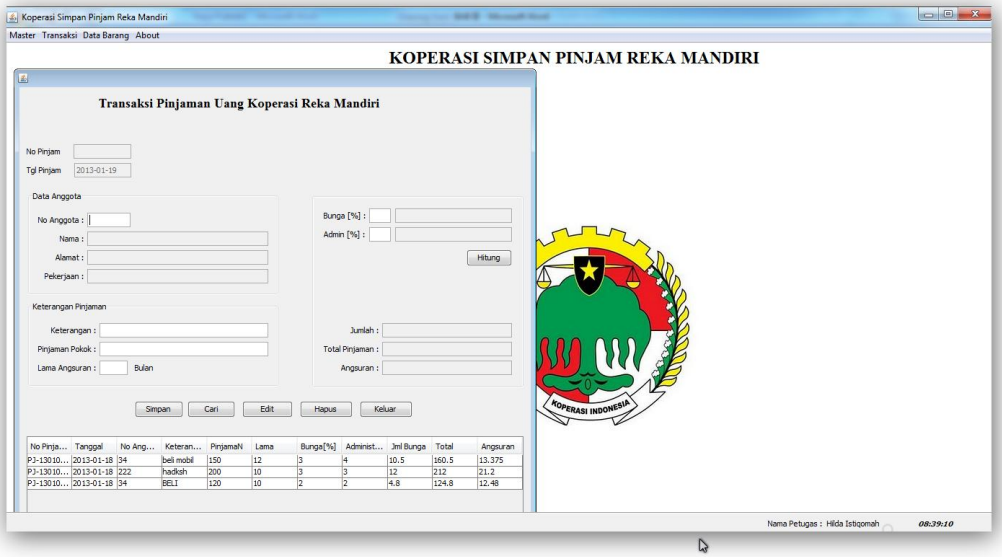

*Gambar 61 Implementasi Menu Data Petugas*

H. Implementasi Menu Transaksi Simpan



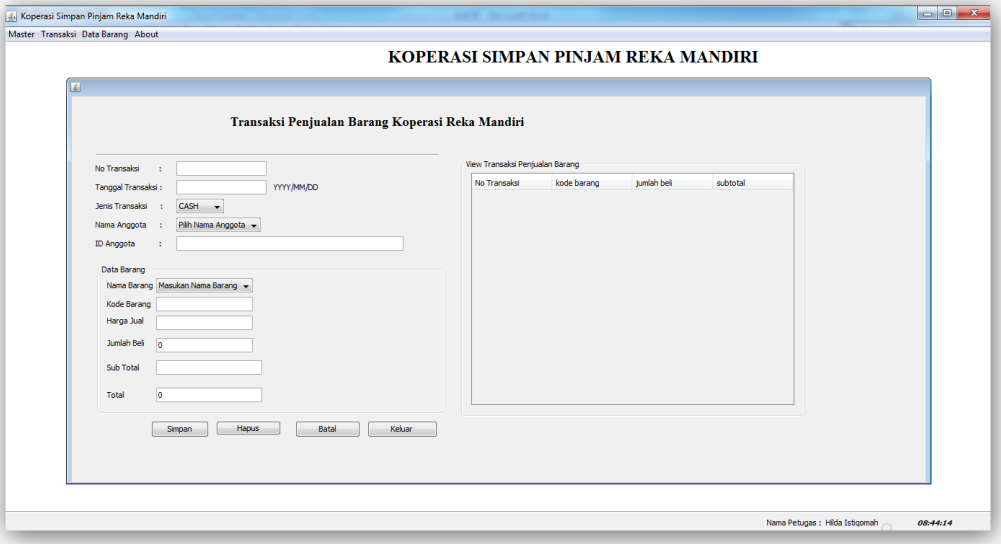

*Gambar 62 Implementasi Menu Transaksi Simpan*

I. Implementasi Menu Transaksi Pinjam



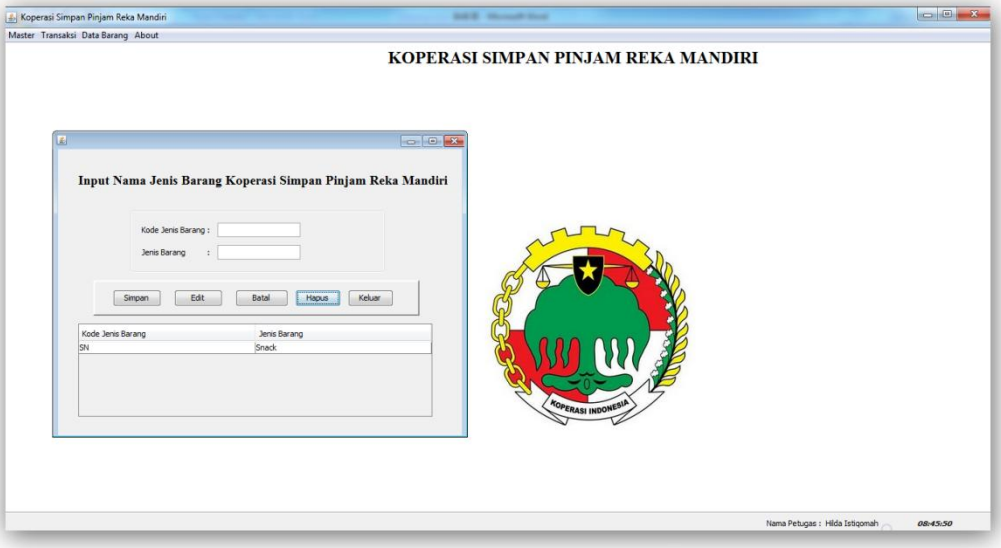

*Gambar 63 Implementasi Menu Transaksi Pinjam*

J. Implementasi Menu Transaksi Penjualan Barang



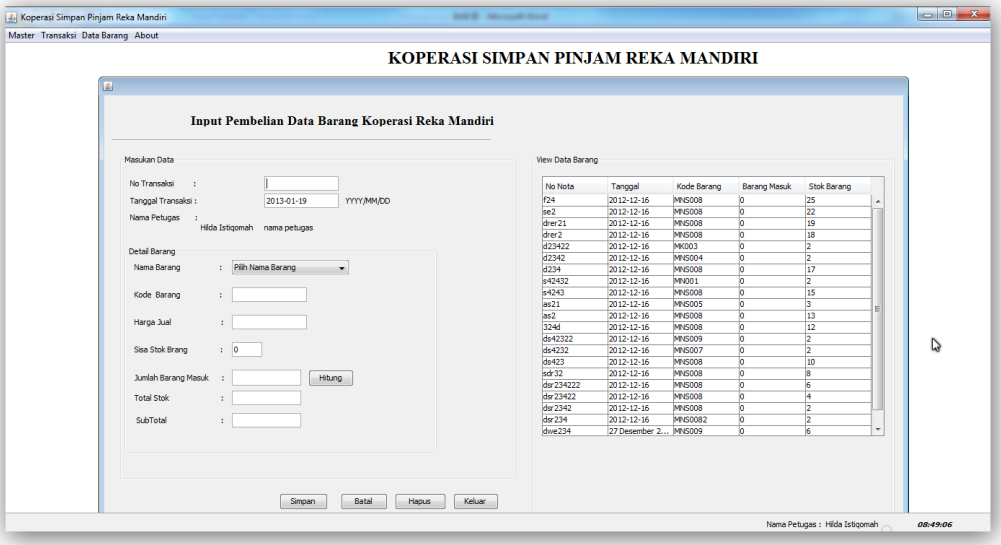

*Gambar 64 Implementasi Menu Transaksi Penjualan Barang*

K. Implementasi Menu Data Barang Jenis Barang



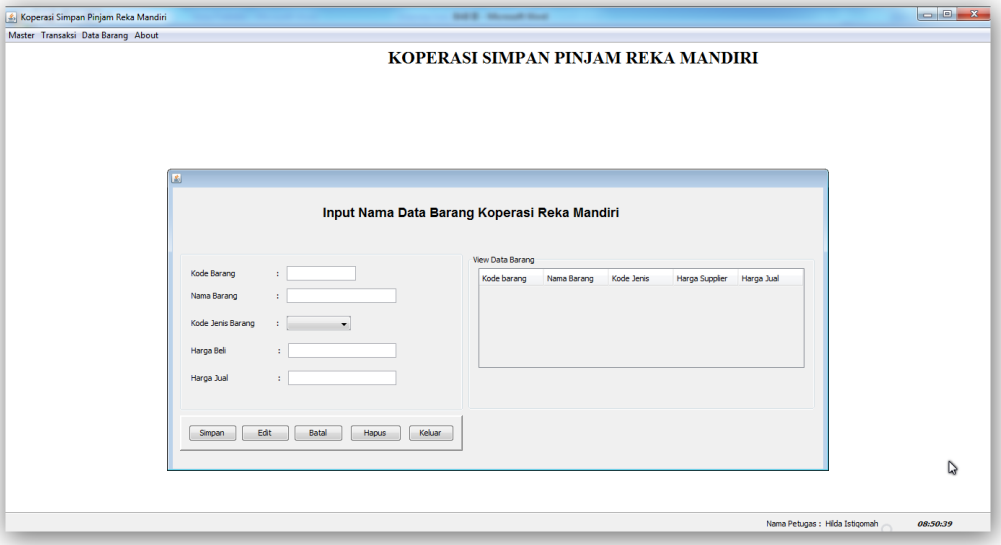

*Gambar 65 Implementasi Menu Data Barang Jenis Barang*

L. Implementasi Menu Data Barang Input Pembelian Data Barang




*Gambar 66 Implementasi Menu Data Barang Input Pembelian Data Barang*

M. Implementasi Menu Data barang Input Data Brang




*Gambar 67 Implementasi Menu Data barang Input Data Barang*