

BAB III

METODOLOGI

3.1 Alat dan Bahan Penelitian

Dalam penelitian dibutuhkan beberapa alat dan bahan untuk mendukung berjalannya perancangan dan implementasi aplikasi .

3.1.1 Alat

Alat yang digunakan berupa perangkat keras (*hardware*) dan perangkat lunak (*software*).

a. Perangkat Keras (*Hardware*)

1. *Personal Computer* (PC) atau Laptop.

64 bit architecture processor,

4 GB *Random Access Memmory* (RAM),

Sistem Operasi *Windows* 10.

2. *Printer* Dokumen untuk mencetak laporan.

b. Perangkat Lunak (*Software*)

1. *Microsoft Visual Studio 2013*

2. *Microsoft Visio 2013*

3. *SQL Server 2014*

3.1.2 Bahan

Dalam penelitian bahan yang digunakan adalah :

1. Data yang diperoleh melalui studi literatur berdasarkan penelitian sebelumnya yang masih memiliki keterkaitan dengan aplikasi desktop yang dikembangkan. Dari data yang diperoleh, maka didapatkan kebutuhan dari aplikasi pada saat pengembangan aplikasi.
2. Data anggota, simpanan, pinjaman dan angsuran Koperasi Sari Usaha Tani

3.2 Teknik Pengumpulan Data

3.2.1 Wawancara

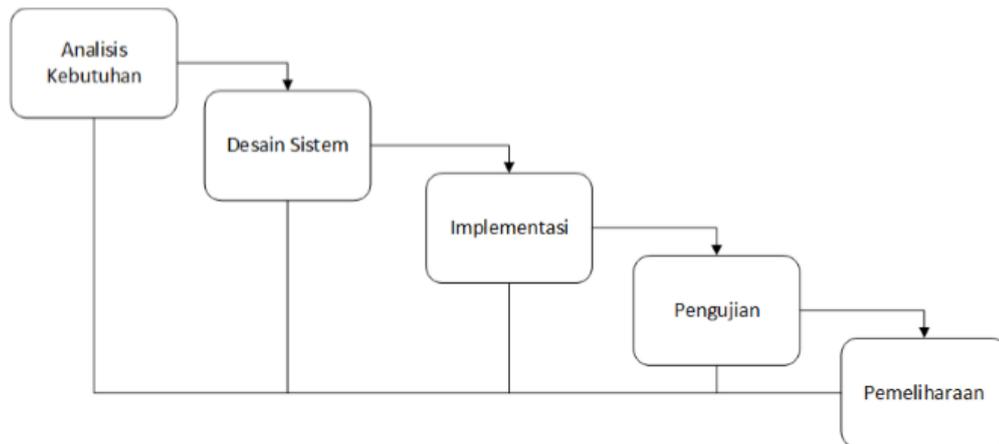
Wawancara merupakan proses tanya jawab yang bertujuan untuk mendapatkan data. Proses wawancara ini dilakukan langsung kepada salah satu pengurus Koperasi Sari Usaha Tani Bapak Muhadi Wijaya. Hal hal yang dibahas adalah mengenai persyaratan - persyaratan, proses perhitungan bunga, proses simpan, pinjam dan angsuran.

3.2.2 Observasi

Metode observasi ini dilakukan untuk mengamati secara langsung objek penelitian agar peneliti dapat mengumpulkan data dan menyimpulkan data secara langsung. Pada penelitian ini peneliti melakukan pengamatan terhadap pembukuan Koperasi Sari Usaha Tani.

3.2.3 Alur Penelitian

Sistem informasi Koperasi Simpan Pinjam Sari Usaha Tani berbasis desktop ini menggunakan metode *System Development Life Cycle (SDLC)* model *waterfall*. Model *Waterfall* merupakan salah satu metode dalam SDLC yang mempunyai ciri khas pengerjaan setiap fase dalam *waterfall* harus diselesaikan terlebih dahulu sebelum melanjutkan ke fase selanjutnya. Artinya fokus terhadap masing-masing fase dapat dilakukan maksimal karena tidak adanya pengerjaan yang sifatnya *parallel* itu alasan kenapa metode ini digunakan dalam mengerjakan sistem ini. Tahapan tahapan dari metode *waterfall* ditunjukkan pada Gambar 3.1:



Gambar 3.1 *Waterfall*

1. Analisis kebutuhan

Analisis kebutuhan mencakup kebutuhan perangkat keras, perangkat lunak, dan bahan untuk menu pada aplikasi. Sebelum membuat program aplikasi berbasis *desktop*, terlebih dahulu dilakukan pengumpulan data dengan melakukan wawancara dan observasi terhadap pengurus koperasi.

2. Desain sistem

Sebelum aplikasi dibuat, penulis merancang desain aplikasi. Pembuatan rancangan ini ditujukan agar sistem yang dibuat sesuai dengan kebutuhan yang telah ditentukan.

3. Implementasi

Perancangan yang telah dibuat sebelumnya diterjemahkan dalam *coding* yang membentuk sebuah aplikasi berbasis *desktop*. Pembuatan aplikasi *desktop* pun dilakukan dengan menggunakan peralatan yang telah ditentukan dari analisis kebutuhan.

4. Pengujian

Sistem akan diuji sesuai dengan *requirement* yang telah ditentukan sebelumnya. Pengujian aplikasi *desktop* dilakukan dengan menggunakan metode pengujian fungsional.

5. Pemeliharaan

Pemeliharaan *software* termasuk didalamnya adalah pengembangan. Ketika *software* dijalankan mungkin masih ada *error* atau *fitur* yang perlu ditambahkan. Pengembangan pun perlu dilakukan untuk memperbaiki *software* menjadi lebih baik.

3.3 Analisis Kebutuhan

Analisis kebutuhan didapat dari studi literatur dan diskusi bersama pengelola Koperasi Sari Usaha Tani. Berikut ini merupakan analisis kebutuhan pada aplikasi desktop

1. Terdapat *form Login* untuk *admin* dan superadmin.
2. Terdapat *form Registrasi admin* untuk *admin* baru.
3. Terdapat *form Registrasi anggota*, beserta atribut-atributnya.
4. Terdapat *form Simpanan*, yang terdiri dari Simpanan wajib, dan Simpanan sukarela. Saat melakukan registrasi anggota maka terlebih dahulu melakukan simpanan pokok.
5. Terdapat *form Pinjaman*
6. Terdapat *form Angsuran*
7. Terdapat *form* untuk Laporan dan untuk mencetak data

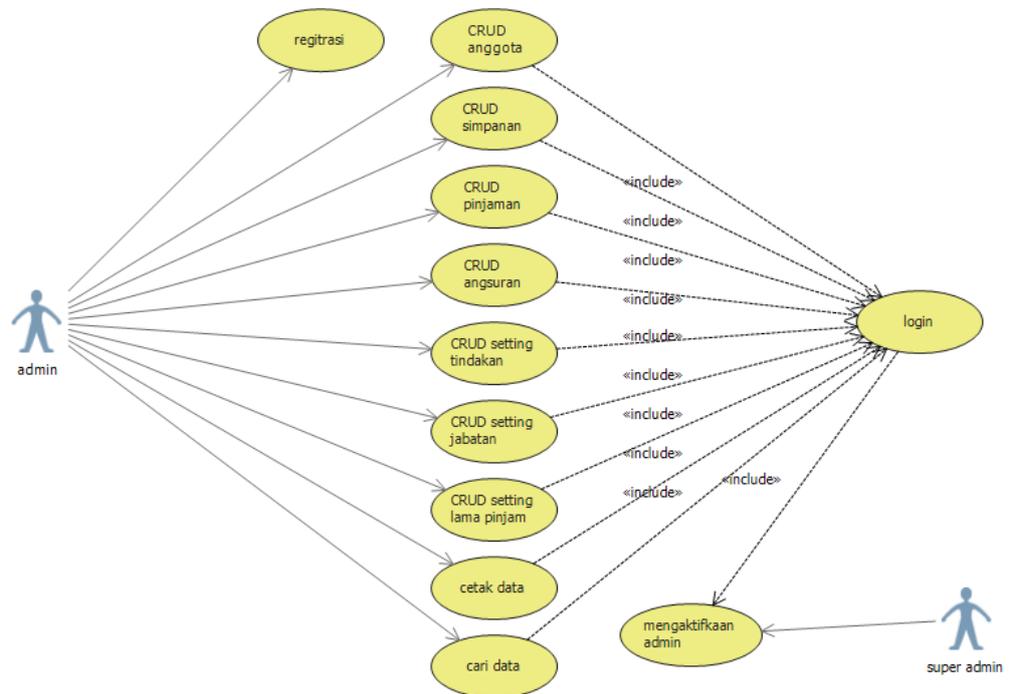
3.4 Rancangan Sistem

3.4.1 Rancangan Sistem Aplikasi *Desktop*

Perancangan sistem yang di lakukan untuk mengembangkan sistem aplikasi desktop ini adalah dengan menggunakan diagram *Unified Modeling Language* (UML). *Unified Modeling Language* (UML) adalah bahasa spesifikasi standar yang di gunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan sebuah perangkat lunak. *Unified Modeling Language* (UML) merupakan metodologi untuk mengembangkan sistem berorientasi objek dan juga merupakan *tool* untuk mendukung pengembangan sistem. Metode UML yang dipakai yaitu *Use Case Diagram*, *ER Diagram*, *Class Diagram*, dan *Activity Diagram*

3.4.1.1 *Use Case Diagram*

Use Case Diagram digunakan untuk memodelkan proses bisnis berdasarkan pandangan pengguna sistem. Use case diagram lebih menekankan pada “siapa” melakukan “apa” dalam lingkungan sistem perangkat lunak yang akan di bangun. Use case diagram untuk aplikasi ini dapat dilihat pada Gambar 3.2.

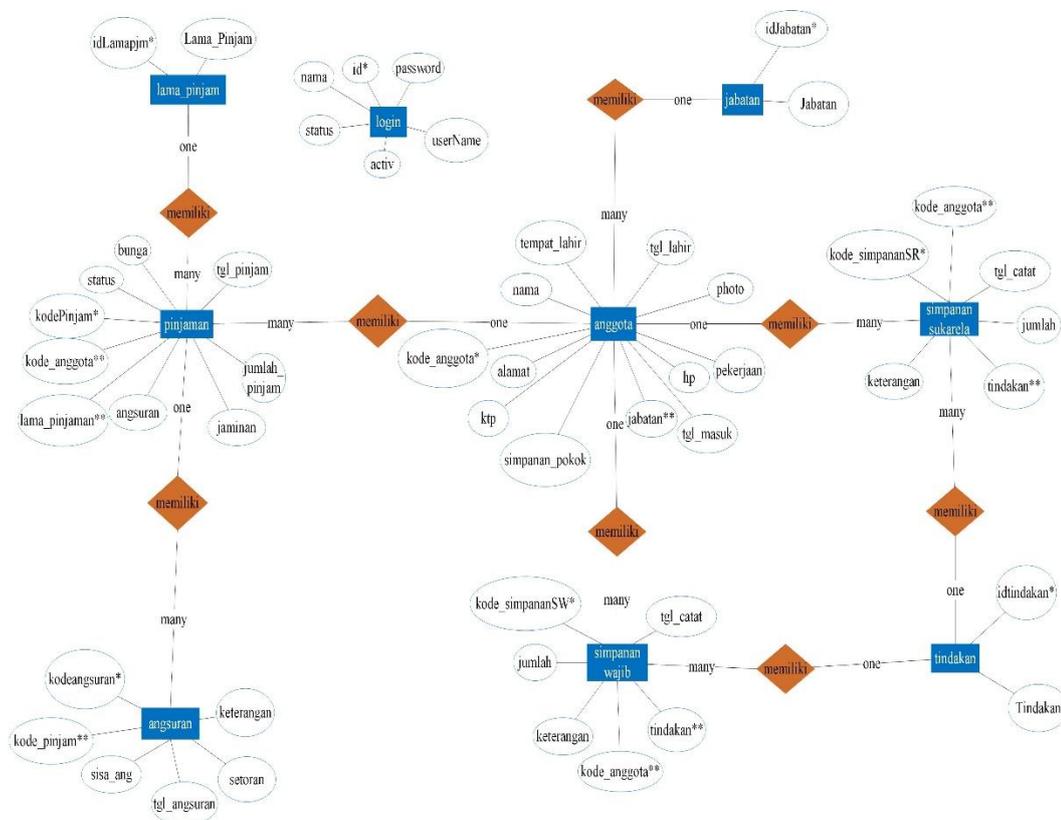


Gambar 3. 2 Use Case Diagram

Berikut penjelasan gambar 3.2

1. Terdapat dua aktor pada *use case diagram* yaitu *admin* dan *super admin*
2. Pada sistem yang dibuat *Admin* memiliki wewenang untuk mengelola seluruh kegiatan mulai dari tambah, hapus, edit data, lihat data dan cetak data anggota, simpanan, pinjaman, angsuran, *Setting* tindakan, *Setting* jabatan serta *Setting* lama pinjam.
3. *Super admin* memiliki wewenang mengaktifkan *admin* baru agar dapat mengakses halaman utama sistem.

3.4.1.2 ER Diagram



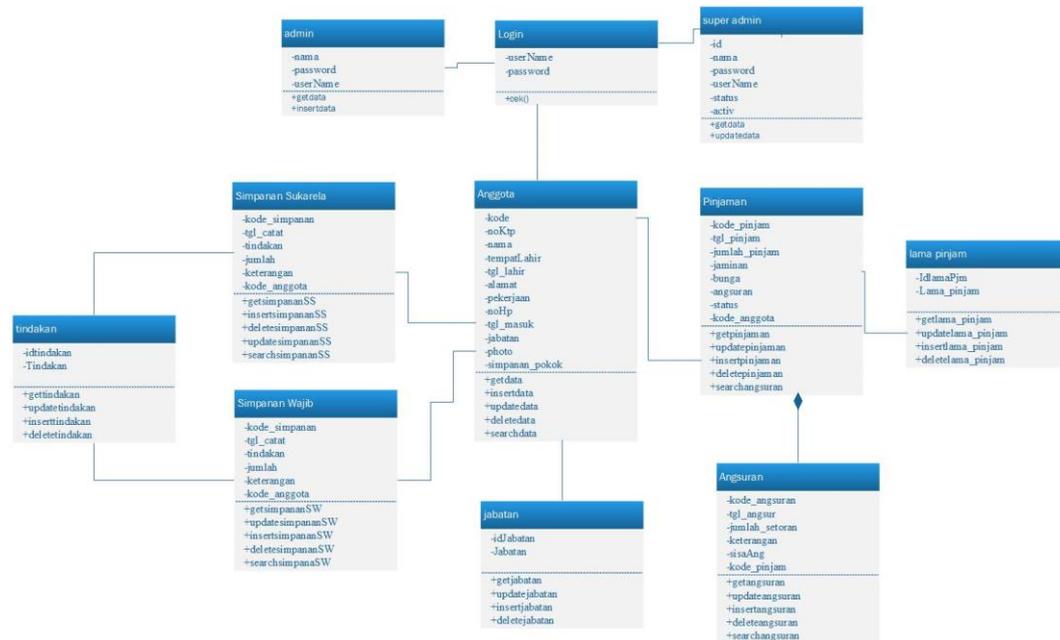
Gambar 3.3 ER Diagram

Gambar diatas adalah *ER Diagram* Koperasi Sari Usaha Tani yang menunjukkan entitas dengan atribut dan relasinya. Entitas ini yang nantinya akan menjadi rancangan tabel pada *database* program. Entitas diatas memiliki relasi dari satu dengan yang lain. Beberapa relasi akan dijelaskan sebagai berikut:

1. Entitas *login* tidak memiliki relasi dengan entitas lain. Jadi entitas *login* tidak bergantung pada entitas lain.
2. Entitas *anggota*, memiliki relasi *many-to-one* terhadap entitas *jabatan*. Karena transaksi ini bisa dilakukan kapan saja dengan satu anggota.
3. Entitas *pinjaman*, memiliki relasi *many-to-one* terhadap entitas *anggota*. Karena pada transaksi ini satu anggota boleh mempunyai banyak pinjaman.

4. Entitas Angsuran, memiliki relasi *many-to-one* terhadap entitas Pinjaman. Karena setiap pinjaman memiliki banyak angsuran.
5. Entitas tindakan, memiliki relasi *many-to-one* terhadap entitas anggota, karena pada transaksi ini banyak tindakan dapat dilakukan oleh satu anggota.
6. Entitas lama pijam, memiliki relasi *one-to-many* terhadap entitas pinjaman. Karena pada transaksi ini satu lama pinjam bisa dimiliki banyak pinjaman.
7. Entitas simpanan sukarela, memiliki relasi *many-to-one* terhadap entitas tindakan, Karena transaksi ini dapat dilakukan berkali kali hanya dengan satu tindakan. Dan entitas simpanan wajib memiliki relasi *many-to-one* terhadap entitas anggota, dimana satu anggota dapat memiliki banyak simpanan sukarela.
8. Entitas simpanan wajib, memiliki relasi *many-to-one* terhadap entitas tindakan. Karena transaksi ini dapat dilakukan berkali kali hanya dengan satu tindakan. Dan entitas simpanan wajib memiliki relasi *many-to-one* terhadap entitas anggota, dimana satu anggota dapat memiliki banyak simpanan wajib.

3.4.1.3 Class Diagram



Gambar 3.4 Class Diagram

Penjelasan fungsi :

1. Class Admin

Berfungsi untuk memasukkan dan menampilkan data pengguna yang dapat menggunakan aplikasi.

2. Class Superadmin

Pada class superadmin berfungsi menampilkan dan mengaktifkan admin baru.

3. Class Login

Pada class login berfungsi memvalidasi *username* dan *password* yang terdapat pada kelas *admin* dan kelas *superadmin*.

4. Class Anggota

Pada class anggota terdapat fungsi *edit* data, simpan data, hapus data, lihat data dan cari data. *Edit* data digunakan untuk mengedit data, simpan data berfungsi untuk menyimpan data, hapus data berfungsi menghapus data, lihat data digunakan untuk melihat data sedangkan cari data berfungsi mencari data.

5. *Class* Pinjaman

Pada *class* pinjaman terdapat fungsi *edit* data, simpan data, hapus data, cari data dan lihat data. *Edit* data digunakan untuk mengedit data, simpan data berfungsi untuk menyimpan data, hapus data berfungsi menghapus data, cari data berfungsi untuk mencari data sedangkan lihat data digunakan untuk melihat data.

6. *Class* Angsuran

Pada *class* angsuran terdapat fungsi *edit* data, simpan data, hapus data, cari data, dan lihat data. *Edit* data digunakan untuk mengedit data, simpan data berfungsi untuk menyimpan data, hapus data berfungsi menghapus data, cari data berfungsi untuk mencari data sedangkan lihat data digunakan untuk melihat data.

7. *Class* Simpanan Sukarela

Pada *class* simpanan sukarela terdapat fungsi *edit* data, simpan data, hapus data, cari data, dan lihat data. *Edit* data digunakan untuk mengedit data, simpan data berfungsi untuk menyimpan data, hapus data berfungsi menghapus data, cari data berfungsi untuk mencari data sedangkan lihat data digunakan untuk melihat data.

8. *Class* Simpanan Wajib

Pada *class* simpanan wajib terdapat fungsi edit data, simpan data, hapus data, cari data, dan lihat data. *Edit* data digunakan untuk mengedit data, simpan data berfungsi untuk menyimpan data, hapus data berfungsi menghapus data, cari data berfungsi untuk mencari data sedangkan lihat data digunakan untuk melihat data.

9. *Class* Tindakan

Pada *class* tindakan terdapat fungsi *edit* data, simpan data, hapus data, dan lihat data. *Edit* data digunakan untuk mengedit data, simpan data berfungsi untuk menyimpan data, hapus data berfungsi menghapus data sedangkan lihat data digunakan untuk melihat data.

10. *Class* Jabatan

Pada kelas jabatan terdapat fungsi edit data, simpan data, hapus data, dan lihat data. *Edit* data digunakan untuk mengedit data, simpan data berfungsi untuk menyimpan data, hapus data berfungsi menghapus data sedangkan lihat data digunakan untuk melihat data.

11. *Class* Lama Pinjam

Pada *class* lama pinjam terdapat fungsi *edit* data, simpan data, hapus data, dan lihat data. *Edit* data digunakan untuk mengedit data, simpan data berfungsi untuk menyimpan data, hapus data berfungsi menghapus data sedangkan lihat data digunakan untuk melihat data.

Berikut penjelasan *Class Diagram* gambar 3.4 Aplikasi Desktop Sistem Informasi Simpan Pinjam Koperasi :

- a. Semua kelas pada diagram digunakan untuk inisialisasi ke dalam tabel model dan basis data aplikasi.
- b. Kelas admin memiliki asosiasi dengan *login* dimana pada kelas *login* dilakukan validasi kelas *admin*.
- c. Kelas super admin memiliki asosiasi dengan *login* dimana pada kelas *login* dilakukan validasi kelas super admin.
- d. Kelas anggota memiliki asosiasi dengan kelas simpanan sukarela dan simpanan wajib. Anggota boleh melakukan simpanan lebih dari satu kali.
- e. Kelas simpanan wajib dan simpanan sukarela memiliki asosiasi dengan kelas tindakan. Anggota boleh melakukan tindakan simpanan lebih dari satu kali pada simpanan wajib atau simpanan sukarela .
- f. Kelas jabatan memiliki asosiasi dengan anggota, artinya setiap jabatan bisa dimiliki oleh beberapa anggota
- g. Kelas pinjaman memiliki *composition* dengan kelas angsuran, artinya kelas angsuran merupakan bagian dari kelas pinjaman. Kelas angsuran tidak dapat berdiri sendiri apabila kelas pinjaman tidak ada.

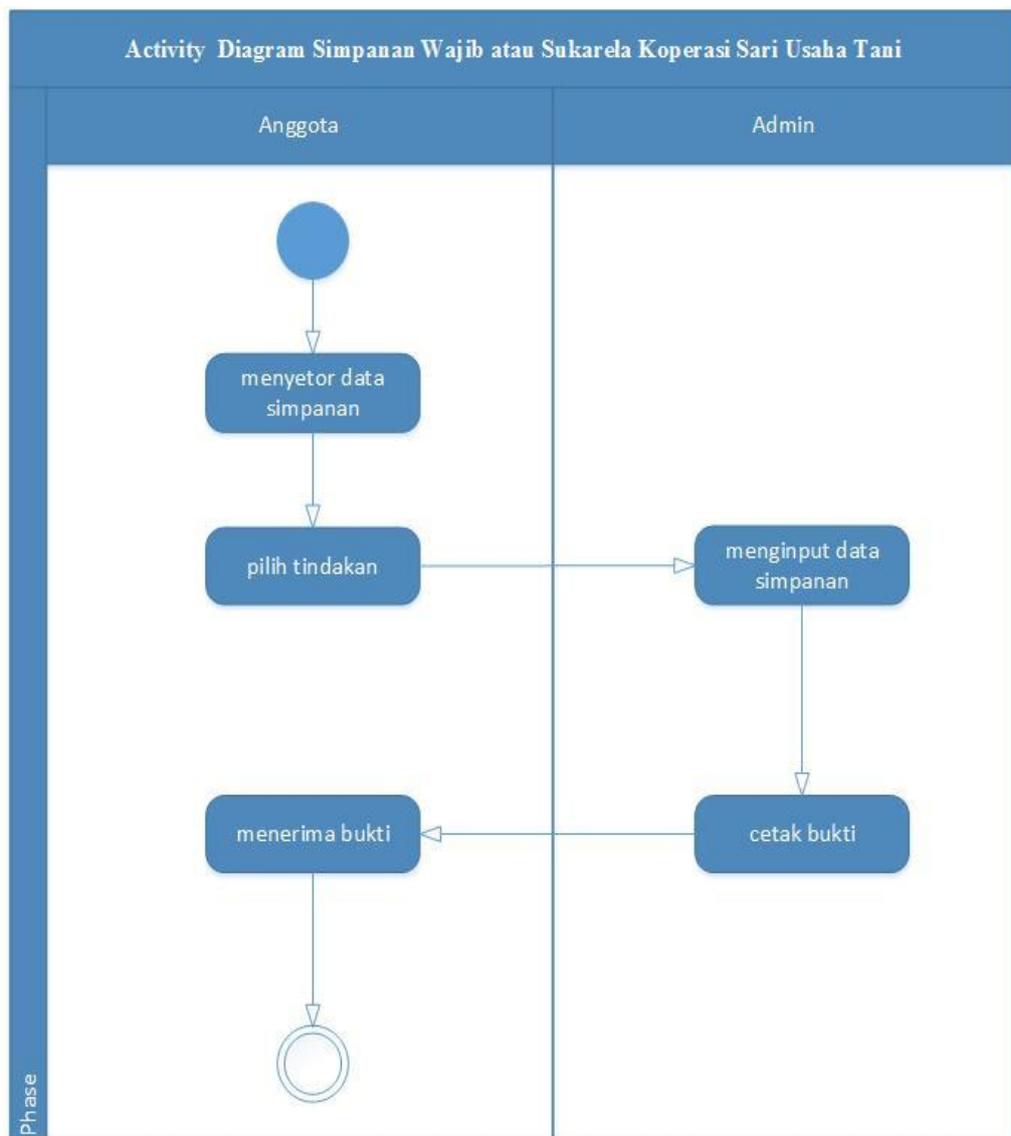
- h. Kelas lama pinjam memiliki asosiasi dengan kelas pinjaman, artinya setiap lama pinjam bisa dimiliki oleh beberapa peminjam.

3.4.1.4 Activity Diagram

Activity diagram dalam aplikasi dibagi menjadi dua bagian yaitu *activity diagram* simpanan wajib atau simpanan sukarela dan *activity diagram* pinjaman.

- a. *Activity Diagram* Simpanan Wajib atau Simpanan Sukarela

Gambaran *Activity Diagram* yang digunakan dalam aplikasi dapat dilihat pada Gambar 3.5.



Gambar 3.5 Activity Diagram Simpanan Wajib atau Simpanan Sukarela

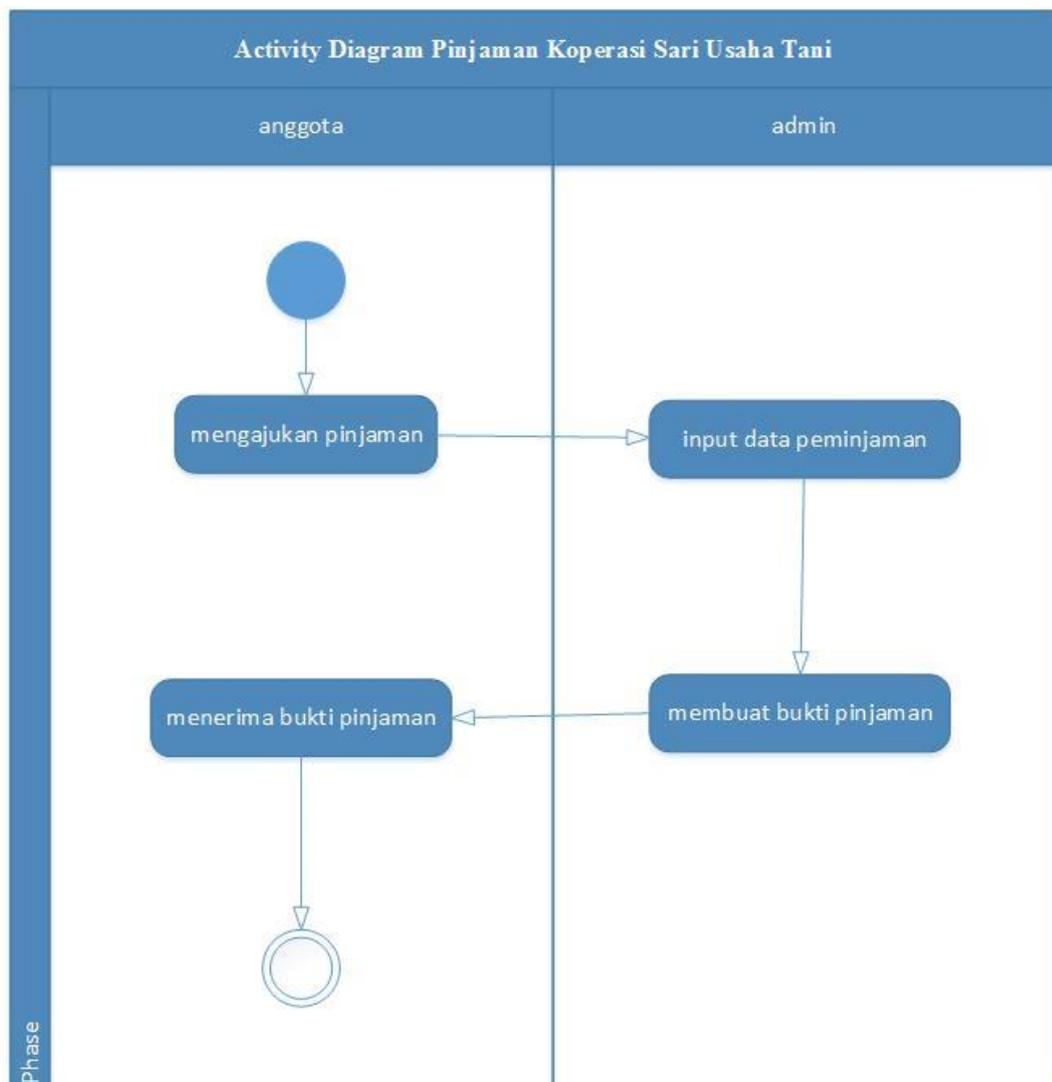
Berikut penjelasan dari gambar 3.5 :

1. Alur dari kegiatan transaksi simpanan wajib atau simpanan sukarela yaitu dimulai dari anggota yang membawa bukti kode anggota dan memilih tindakan mau melakukan simpanan atau penarikan simpanan.
2. Kemudian *admin* melakukan *input* data simpanan anggota dan membuat bukti simpanann

3. Anggota mendapatkan bukti simpanan pada slip atau buku bukti simpanan yang terdapat jumlah saldo simpanan.

b. *Activity Diagram* Pinjaman

Gambaran *Activity Diagram* yang digunakan dalam aplikasi dapat dilihat pada Gambar 3.6.



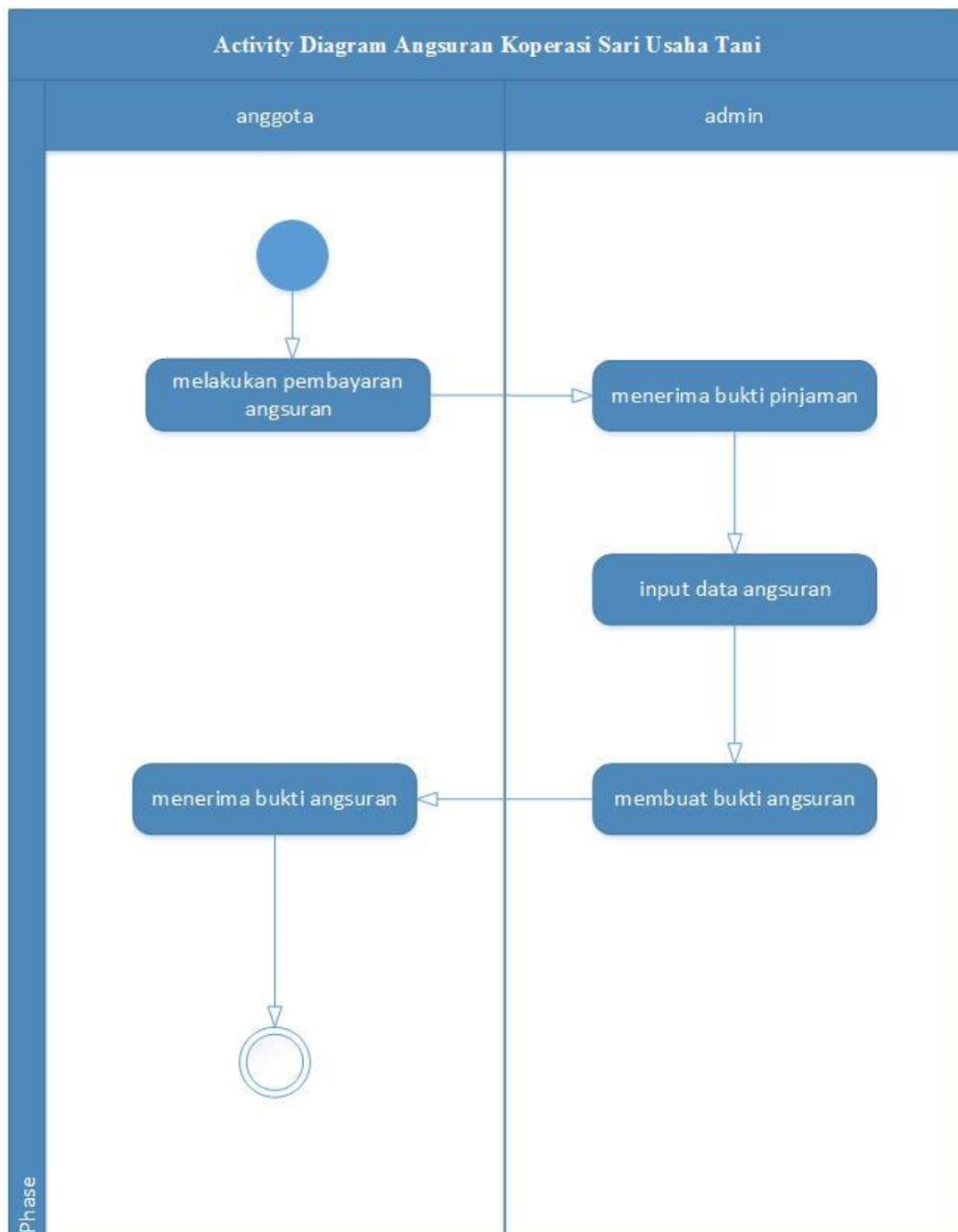
Gambar 3.6 *Activity Diagram* Pinjaman

Berikut penjelasan dari gambar 3.6 :

1. Terdapat alur dari kegiatan peminjaman, yaitu dimulai dari anggota mengajukan permohonan pinjaman.
2. Jika permohonan telah memenuhi syarat dan disetujui maka *admin* memeriksa kode anggota dan *input* data nominal pinjaman.
3. *Admin* membuat arsip dan slip bukti pinjaman dengan kode pinjaman yang kemudian akan digunakan anggota pada saat melakukan pembayaran angsuran.
4. Anggota menerima bukti pinjaman yang nantinya akan digunakan disaat melakukan angsuran setiap bulannya.

c. *Activity Diagram* Angsuran

Gambaran *Activity Diagram* yang digunakan dalam aplikasi dapat dilihat pada Gambar 3.7.



Gambar 3.7 Activity Diagram Angsuran

Berikut penjelasan dari gambar 3.7 :

1. Terdapat alur dari kegiatan angsuran, yaitu dimulai dari anggota melakukan pembayaran angsuran setiap bulannya sesuai dengan

ketentuan lamanya angsuran dan memberikan slip pinjaman ke *admin*

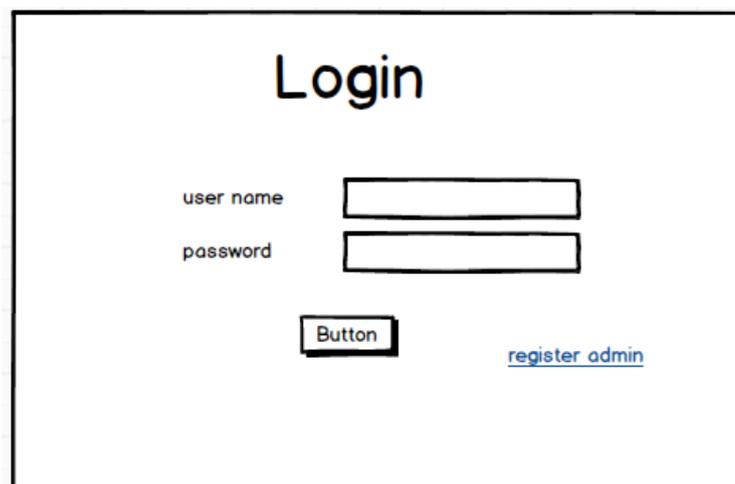
2. *Admin* memasukan nominal yang anggota setorkan dan membuat slip bukti angsuran
3. Anggota menerima slip bukti angsuran yang berisikan informasi sisa pinjaman

3.5 Rancangan Antarmuka (*User Interface*)

User Interface sangat penting dalam suatu aplikasi karena merupakan bagian dari perangkat lunak yang menjadi sarana komunikasi antar pengguna dengan sistem serta dapat membantu pengguna dalam melakukan aktivitasnya.

3.5.1 Rancangan Antarmuka Halaman *Login*

Rancangan antarmuka halaman *login* digunakan oleh pihak *admin* koperasi untuk dapat masuk ke halaman utama aplikasi. Gambaran rancangan antarmuka menu login dapat dilihat pada Gambar 3.8 :



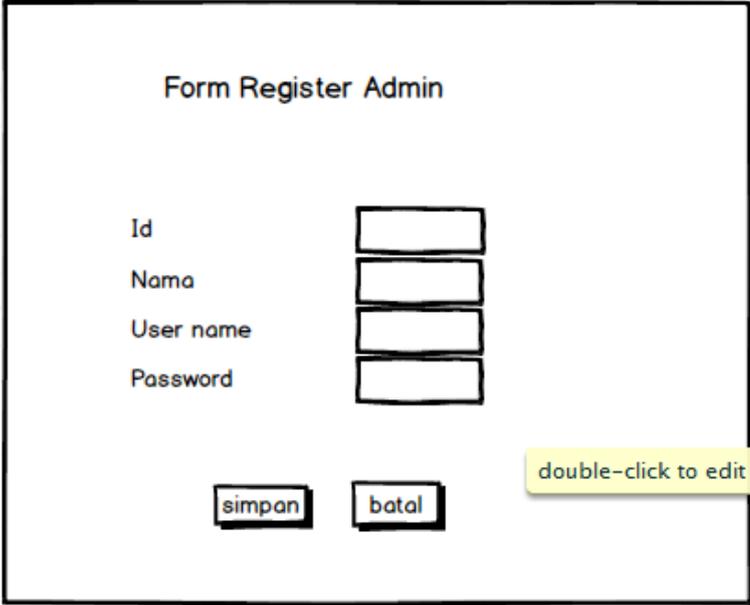
The image shows a login form with the following elements:

- Header: "Login" in a large, bold font.
- Input fields: Two rectangular boxes for "user name" and "password".
- Buttons: A rectangular button labeled "Button" and a blue text link labeled "register admin".

Gambar 3.8 Rancangan Antarmuka Halaman *Login*

3.5.2 Rancangan Antarmuka Halaman Register Admin

Rancangan antarmuka halaman registrasi *admin* digunakan untuk mendaftarkan *admin* agar dapat mengakses halaman utama. Gambaran rancangan antarmuka menu register admn dapat dilihat pada Gambar 3.9 :

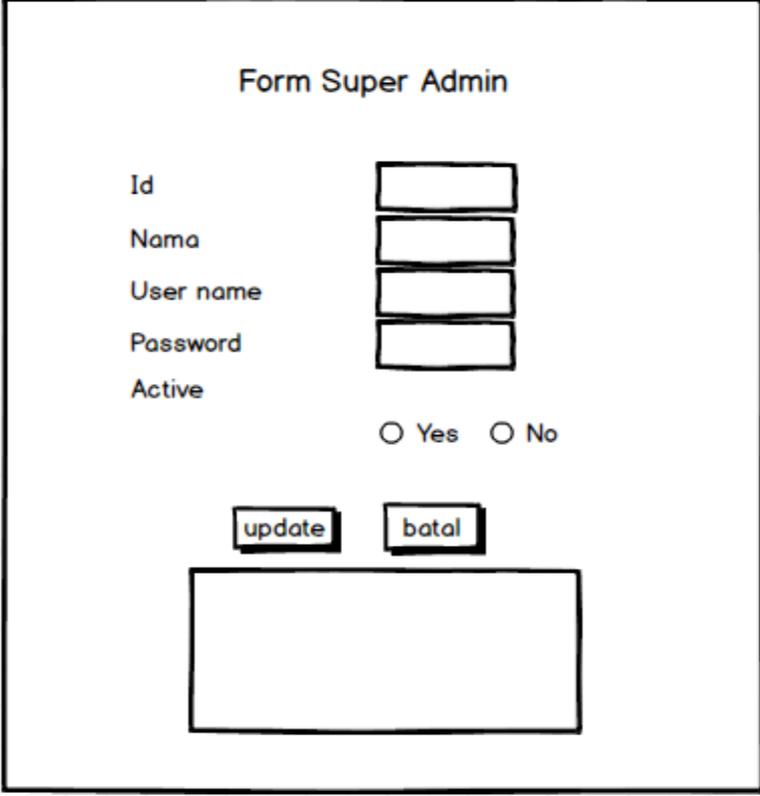


The image shows a wireframe of an 'Admin Registration Form'. The title 'Form Register Admin' is centered at the top. Below the title, there are four input fields stacked vertically, each preceded by a label: 'Id', 'Nama', 'User name', and 'Password'. At the bottom of the form, there are two buttons: 'simpan' (save) on the left and 'batal' (cancel) on the right. A yellow callout box with the text 'double-click to edit' is positioned to the right of the 'batal' button.

Gambar 3.9 Rancangan Antarmuka Halaman Register Admin

3.5.3 Rancangan Antarmuka Halaman Super admin

Rancangan antarmuka halaman super *admin* digunakan mengaktifkan *admin* yang sudah mendaftar di halaman register *admin*, untuk membatasi siapa saja yang dapat mengakses menu utama. Gambaran rancangan antarmuka menu register admn dapat dilihat pada Gambar 3.10 :

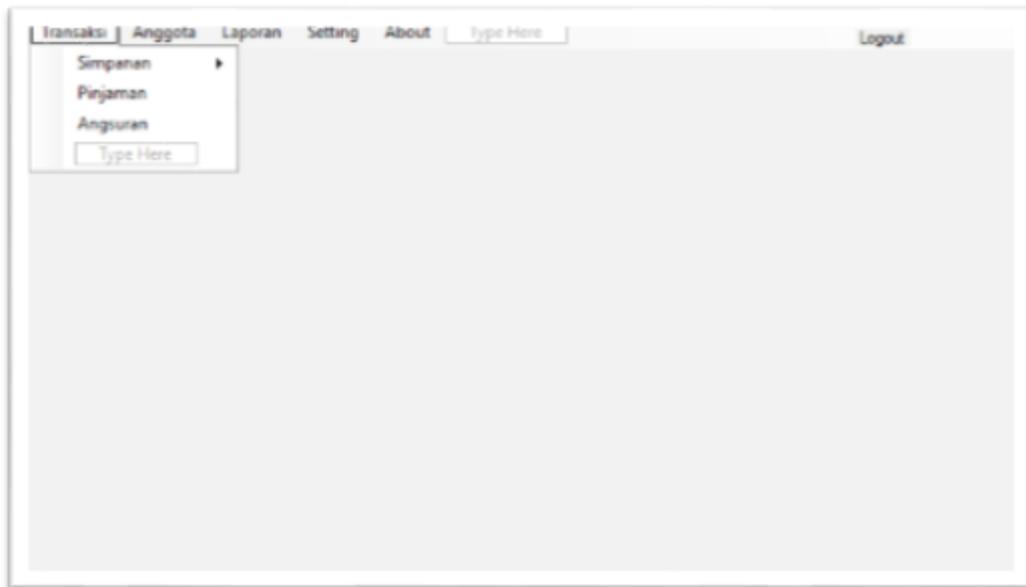


The image shows a wireframe for a 'Form Super Admin'. It is enclosed in a rectangular border. At the top center, the title 'Form Super Admin' is displayed. Below the title, there are five labels on the left side: 'Id', 'Nama', 'User name', 'Password', and 'Active'. To the right of each label is a rectangular input field. The 'Active' label is followed by two radio buttons, one labeled 'Yes' and one labeled 'No'. Below the input fields and radio buttons, there are two buttons: 'update' and 'batal'. At the bottom of the form, there is a large, empty rectangular box.

Gambar 3.10 Rancangan Antarmuka Halaman Super Admin

3.5.4 Rancangan Antarmuka Halaman Utama

Rancangan antarmuka halaman utama merupakan tampilan utama dari aplikasi yang dapat dibuka oleh *admin*. Gambaran rancangan antarmuka halaman utama dapat dilihat pada Gambar 3.11.



Gambar 3.11 Rancangan Antarmuka Halaman Utama

Halaman utama pada aplikasi menggunakan tampilan menu strip, terdapat menu-menu yang dapat dibuka sesuai dengan pengguna. Jika pengguna sudah berhasil login, semua menu akan aktif. Fungsi setiap menu dapat dilihat pada Tabel 3.1

Tabel 3.1 Tabel Fungsi Halaman Utama

No	Menu	Sub Menu	Keterangan
1	Transaksi	Simpanan	Sub menu yang didalamnya terdapat transaksi simpanan wajib dan simpanan sukarela
		Pinjaman	Sub menu ini didalamnya terdapat transaksi pinjaman anggota
		Angsuran	Sub menu ini didalamnya terdapat transaksi angsuran anggota

No	Menu	Sub Menu	Keterangan
2	Anggota	Registrasi	Menu ini berfungsi untuk menampilkan halaman data anggota.
3	Laporan	Laporan simpanan wajib	Sub menu data laporan simpanan wajib berfungsi untuk menampilkan halaman data laporan dari transaksi simpanan wajib.
		Laporan simpanan sukarela	Sub menu data laporan simpanan sukarela berfungsi untuk menampilkan halaman data laporan dari transaksi simpanan sukarela.
		Laporan data pinjaman	Sub menu data laporan data pinjaman berfungsi untuk menampilkan halaman data laporan dari transaksi pinjaman.
		Laporan data angsuran	Sub menu data laporan data angsuran berfungsi untuk menampilkan halaman data laporan dari transaksi angsuran.
		Laporan data anggota	Sub menu data laporan data anggota berfungsi untuk menampilkan halaman data laporan dari registrasi anggota.

No	Menu	Sub Menu	Keterangan
4	<i>Setting</i>	<i>Ganti password</i>	Sub menu ini berfungsi apabila <i>admin</i> ingin merubah kata sandi
		<i>Setting jabatan</i>	Sub menu ini berfungsi untuk menampilkan jabatan
		<i>Setting lama pinjam</i>	Sub menu ini berfungsi untuk menampilkan lama pinjam
		<i>Setting Tindakan</i>	Sub menu ini berfungsi untuk menampilkan jabatan
5	<i>About</i>	-	Menu ini berfungsi untuk menampilkan tentang koperasi
6	<i>Logout</i>		Menu <i>logout</i> berfungsi untuk keluar dari aplikasi dan kembali ke halaman <i>login</i> .

3.5.5 Rancangan Antarmuka Halaman Data Anggota

Rancangan antarmuka halaman data anggota merupakan tampilan yang menampilkan seluruh data anggota serta dapat melakukan penambahan data anggota dimulai dari kode anggota otomatis oleh aplikasi, nama dan data diri anggota, simpanan pokok yang harus disetorkan oleh anggota dan tanggal masuk menjadi anggota koperasi, serta dapat melakukan edit data, menghapus dan mencari data anggota. Gambaran rancangan antarmuka halaman data anggota dapat dilihat pada Gambar 3.12.

Gambar 3.12 Rancangan Antarmuka Halaman Data Anggota

3.5.6 Rancangan Antarmuka Halaman Simpanan Wajib

Rancangan antarmuka halaman data simpanan wajib merupakan tampilan yang menampilkan seluruh data simpanan wajib anggota serta dapat melakukan pengelolaan transaksi simpanan wajib anggota, kode simpanan otomatis dari aplikasi, dapat mencatat dan menarik simpanan wajib, serta dapat mengedit dan menghapus data simpanan wajib. Gambaran rancangan antarmuka halaman data simpanan wajib dapat dilihat pada Gambar 3.13.

Gambar 3.13 Rancangan Antarmuka Halaman Data Simpanan Wajib

3.5.7 Rancangan Antarmuka Halaman Simpanan Sukarela

Rancangan antarmuka halaman data simpanan sukarela merupakan tampilan yang menampilkan seluruh data simpanan sukarela anggota serta dapat melakukan pengelolaan transaksi simpanan sukarela anggota, kode simpanan otomatis dari aplikasi, dapat mencatat dan menarik simpanan sukarela, serta dapat mengedit dan menghapus data simpanan sukarela. Gambaran rancangan antarmuka halaman data simpanan sukarela dapat dilihat pada Gambar 3.14.

Form simpanan sukarela

pencarian

Data anggota

kode anggota

nama

saldo

Data lainya

kode simpanan

tanggal simpan

tindakan

jumlah

keterangan

simpan edit hapus refresh

Gambar 3.14 Rancangan Antarmuka Halaman Data Simpanan Sukarela

3.5.8 Rancangan Antarmuka Halaman Pinjaman

Rancangan antarmuka halaman data pinjaman merupakan tampilan yang menampilkan seluruh data pinjaman anggota serta dapat melakukan pengelolaan transaksi pinjaman anggota dimulai dari kode anggota dan kode pinjam otomatis oleh aplikasi, mencatat jumlah pinjam, barang jaminan, status dan lama pinjam yang kemudian sistem secara otomatis menampilkan bunga dan jumlah angsuran setiap bulan. Dapat juga melakukan edit data, menghapus dan mencari data pinjaman. Gambaran rancangan antarmuka halaman data pinjaman dapat dilihat pada Gambar 3.15.

Gambar 3.15 Rancangan Antarmuka Halaman Data Pinjaman

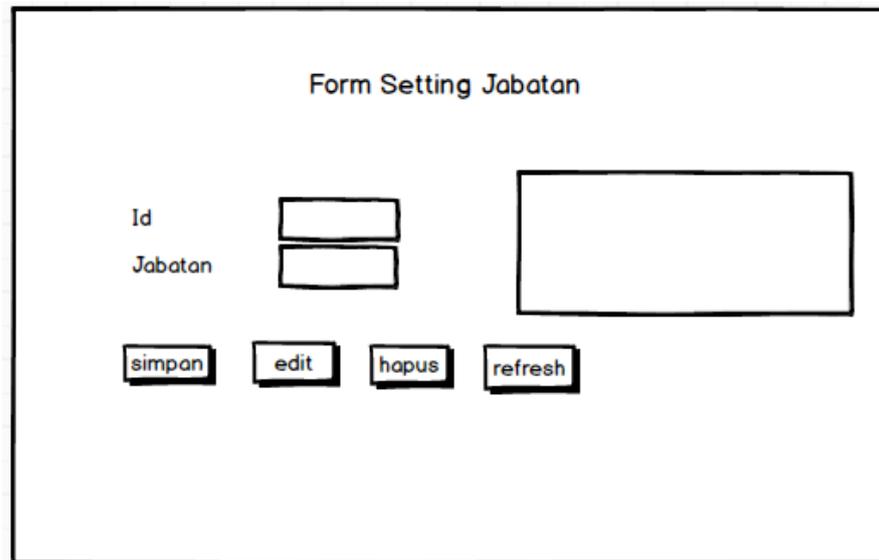
3.5.9 Rancangan Antarmuka Halaman Angsuran

Rancangan antarmuka halaman data angsuran merupakan tampilan yang menampilkan seluruh data angsuran anggota, sisa angsuran dan status anggota. Serta dapat melakukan edit data, menghapus dan mencari data angsuran. Gambaran rancangan antarmuka halaman data angsuran dapat dilihat pada Gambar 3.16.

Gambar 3.16 Rancangan Antarmuka Halaman Data Angsuran

3.5.10 Rancangan Antarmuka Halaman *Setting* Jabatan

Rancangan antarmuka halaman *setting* jabatan merupakan tampilan yang menampilkan jabatan serta dapat melakukan *edit*, tambah dan hapus data jabatan. Gambaran rancangan antarmuka halaman *setting* jabatan dapat dilihat pada Gambar 3.17.

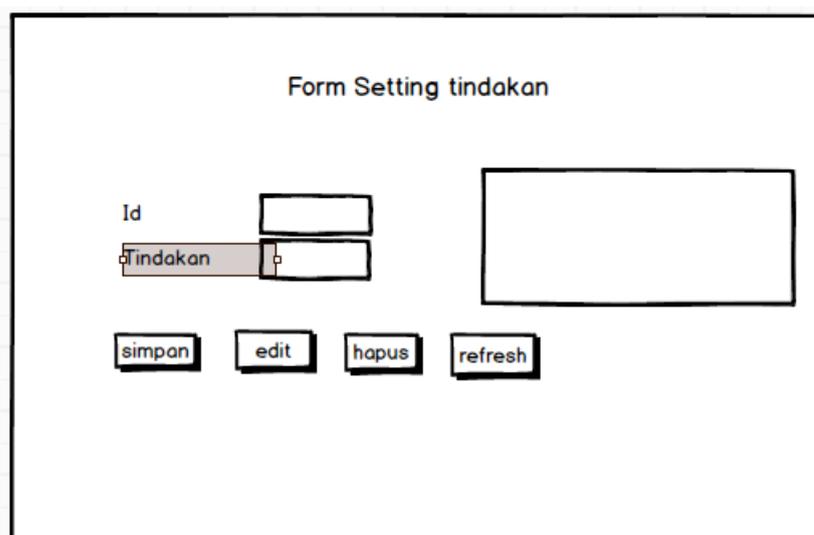


The image shows a wireframe for a 'Form Setting Jabatan' (Job Setting Form). It features a title 'Form Setting Jabatan' at the top center. Below the title, there are two input fields: one labeled 'Id' and one labeled 'Jabatan'. To the right of these fields is a large, empty rectangular box. At the bottom of the form, there are four buttons: 'simpan', 'edit', 'hapus', and 'refresh'.

Gambar 3.17 Rancangan Antarmuka Halaman *Setting* Jabatan

3.5.11 Rancangan Antarmuka Halaman *Setting* Tindakan

Rancangan antarmuka halaman *Setting* tindakan merupakan tampilan yang menampilkan jabatan serta dapat melakukan *edit*, tambah dan hapus data tindakan. Gambaran rancangan antarmuka halaman *Setting* tindakan dapat dilihat pada Gambar 3.18.

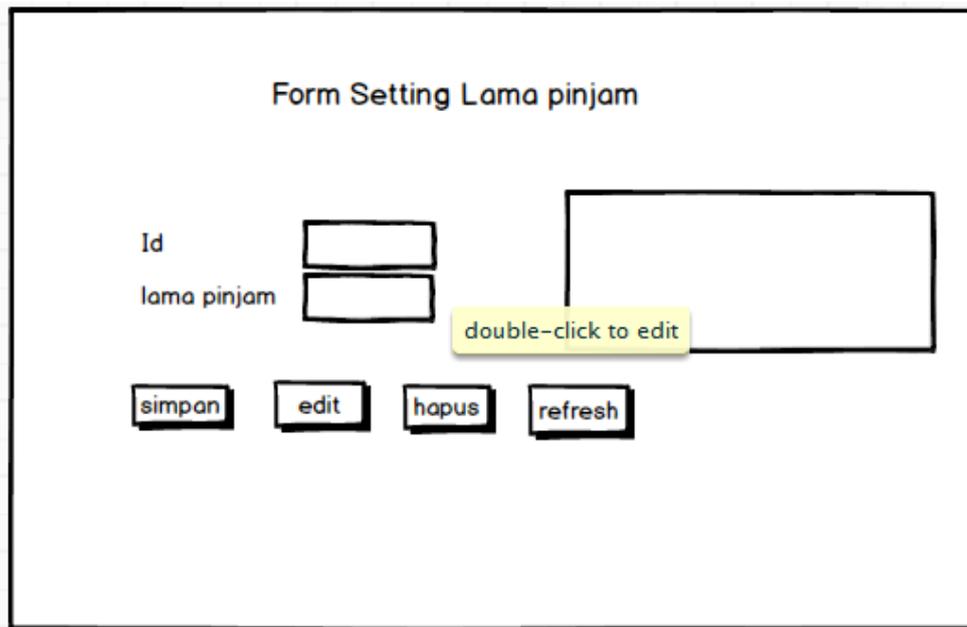


The image shows a wireframe for a 'Form Setting tindakan' (Action Setting Form). It features a title 'Form Setting tindakan' at the top center. Below the title, there are two input fields: one labeled 'Id' and one labeled 'Tindakan'. To the right of these fields is a large, empty rectangular box. At the bottom of the form, there are four buttons: 'simpan', 'edit', 'hapus', and 'refresh'.

Gambar 3.18 Rancangan Antarmuka Halaman *Setting* tindakan

3.5.12 Rancangan Antarmuka Halaman *Setting* Lama Pinjam

Rancangan antarmuka halaman *setting* lama pinjam merupakan tampilan yang menampilkan lama pinjam serta dapat melakukan *edit*, tambah dan hapus data lama pinjam. Gambaran rancangan antarmuka halaman *setting* lama pinjam dapat dilihat pada Gambar 3.19.



The image shows a wireframe for a 'Form Setting Lama pinjam'. At the top center, the title 'Form Setting Lama pinjam' is displayed. Below the title, there are two input fields: one labeled 'Id' and another labeled 'lama pinjam'. To the right of these fields is a larger rectangular area, likely for a list of data, with a yellow tooltip that says 'double-click to edit'. At the bottom of the form, there are four buttons: 'simpan', 'edit', 'hapus', and 'refresh'.

Gambar 3.19 Rancangan Antarmuka Halaman *Setting* Lama Pinjam

