BAB IV

HASIL DAN PEMBAHASAN

4.1. Perancangan Sistem

Perancangan sistem merupakan suatu kegiatan pengembangan prosedur dan proses yang sedang berjalan untuk menghasilkan suatu sistem yang baru untuk meningkatkan efektifitas kerja agar dapat memenuhi hasil yang diharapkan dengan memanfaatkan teknologi dan fasilitas yang tersedia. Perancangan sistem dilakukan apabila tahap dari analisis sistem telah selesai. 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.

4.1.1. Tujuan perancangan sistem

Perancangan sistem bertujuan untuk memberikan gambaran umum pengembangan sistem yang jelas dan rancang sistem yang sesuai dengan kebutuhan pemakai sistem itu sendiri. Adapun tujuan perancangan sistem yang akan diusulkan yaitu untuk menghasilkan suatu rancangan sistem yang dapat membantu proses pengolahan data anggota dalam menangani proses simpan-pinjam dan pelaporan yang semuanya akan terintegrasi dengan satu *database* dan aplikasi, dimana dapat mencapai sasaran kebutuhan demi kepuasan, kenyamanan dan kemudahan. Selain itu perancangan sistem yang di usulkan juga bertujuan untuk mempermudah petugas koperasi dalam melakukan pelayanan simpan pinjam.

4.1.2. Gambaran umum sistem yang diusulkan

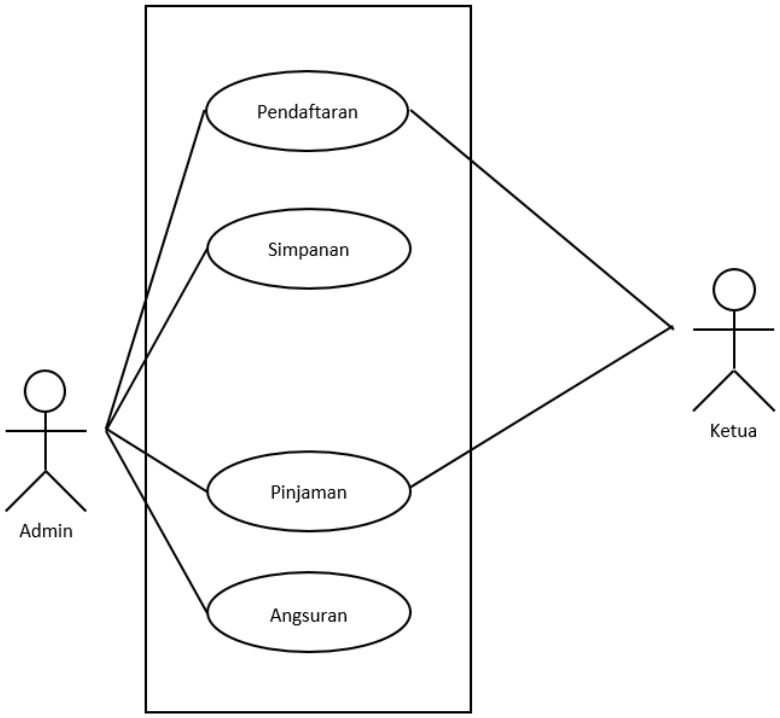
Gambaran umum tentang sistem yang diusulkan pada proses perancangan ini adalah suatu sistem yang merubah seluruh proses sistem yang manual menjadi terkomputerisasi. Proses sistem yang dimaksud adalah pengolahan data anggota, proses penyimpanan, proses pembayaran angsuran, proses peminjaman, perhitungan denda dan cicilan secara otomatis, hingga proses pelaporan atau pengarsipan. Sistem yang diusulkan ini diharapkan dapat membantu efektifitas dan optimalisasi waktu pada seluruh proses bisnis yang berjalan serta dapat membantu mengatasi permasalahan yang ada dan yang paling penting dapat menghasilkan informasi yang cepat, tepat dan akurat.

4.1.3. Perancangan prosedur yang diusulkan

Perancangan prosedur yang diusulkan merupakan tahap untuk memperbaiki atau meningkatkan efisiensi kerja. Dimana dalam perancangan prosedur yang diusulkan ini akan menjelaskan tentang bagaimana pengurus koperasi melakukan aktivitas simpan pinjam dengan sistem yang diusulkan.

4.1.3.1.*Use case* diagram

Use Case Diagram memperlihatkan hubungan-hubungan yang terjadi antara aktor dengan *use case* di dalam sistem. Berikut adalah gambar model *use case* diagram sistem simpan pinjam yang diusulkan :



Gambar 4.1 *Use Case* Diagram Sistem Simpan Pinjam yang diusulkan

4.1.3.2.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. Adapun tahapan-tahapan sekenario *use case* Simpan Pinjam yang diusulkan adalah sebagai berikut :

1. Nama *Use Case* : Pendaftaran

   Aktor : Admin, Ketua

   Tujuan : Aktor menambah, mengedit, atau menghapus data anggota

Tabel 4.1 Skenario *Use Case* Pendaftaran

AKTOR	REAKSI SISTEM
1. Aktor memilih menu data anggota	
	2. Menampilkan halaman data anggota
3. Aktor mengisi data anggota baru kemudian menekan tombol tambah/simpan.	
	4. Memeriksa *field*, apa telah di isi dengan sesuai.
	5. Mengkonfirmasi kesesuaian, jika sesuai akan menambahkan penyimpanan data anggota yang baru pada *database*.
	6. Menyimpan data anggota yang baru pada *database.*

2. Nama *Use Case* : Simpanan

   Aktor : Admin

   Tujuan : Aktor menambah data simpanan

Tabel 4.2 Skenario *Use Case* Simpanan

AKTOR	REAKSI SISTEM
1. Aktor memilih menu simpanan	
	2. Menampilkan halaman simpanan

Tabel 4.2 Skenario *Use Case* Simpanan (Lanjutan)

AKTOR	REAKSI SISTEM
3. Aktor menekan tombol cari, untuk mengisi nomor dan nama anggota	
	4. Menampilkan data simpanan untuk nomor/nama anggota yang dipilih.
5. Aktor memilih jenis simpanan dan mengisi *field* besar simpanan kemudian menekan tombol Simpan	
	6. memeriksa *field*, apa telah di isi dengan sesuai.
	7. Mengkonfirmasi kesesuaian, jika sesuai akan menambahkan penyimpanan data simpanan yang baru pada *database* .
	8. Menyimpan data penyimpanan yang baru pada *database*.
	9. Menampilkan data *history* simpanan dari anggota tersebut

3. Nama *Use Case* : Pinjaman

   Aktor : Ketua, Admin

Tujuan : Aktor menambah data permohonan pinjaman, menyutujui maupun tidak menyetujui pinjaman, serta menambah data pinjaman.

Tabel 4.3 Skenario *Use Case* Pinjaman

AKTOR	REAKSI SISTEM
1. Aktor memilih menu permohonan pinjaman	
	2. Menampilkan halaman permohonan pinjaman
3. Aktor menekan tombol cari, untuk mengisi nomor dan nama anggota	
	4. Menampilkan info saldo simpanan dan info peminjaman untuk nomor/nama anggota yang dipilih.
5. Aktor mengisi *field* jumlah pinjaman kemudian menekan tombol cetak	
	6. Mengkonfirmasi penyimpanan, dan menambahkan data permohonan pinjaman yang baru pada *database* .
	7. Menyimpan data permohonan pinjaman yang baru pada *database*.
	8. Menampilkan/mencetak surat permohonan pinjaman dari anggota tersebut

Tabel 4.3 Skenario *Use Case* Pinjaman (Lanjutan)

AKTOR	REAKSI SISTEM
9. Selanjutnya, setelah membuat data permohonan pinjaman, aktor memilih menu persetujuan pinjaman	
	10. Menampilkan halaman persetujuan pinjaman
11. Aktor menekan tombol cari, untuk mencari permohonan pijaman anggota yang dimaksud	
	12. Menampilkan data permohonan pinjaman anggota yang dimaksud.
13. Setelah itu, aktor memilih permohonan pinjaman anggota yang dimaksud, kemudian menekan tombol setujui/tolak	
	14. Mengkonfirmasi kesesuaian, jika disetujui akan menambahkan data persetujuan pinjaman anggota yang dimaksud pada *database* data peminjaman. Jika tidak disetujui akan menghapus data persetujuan pinjaman anggota yang dimaksud dari *database*.

Tabel 4.3 Skenario *Use Case* Pinjaman (Lanjutan)

AKTOR	REAKSI SISTEM
	15. Menyimpan data persetujuan pinjaman yang baru pada *database.*
16. Terakhir, untuk membuat data pinjaman, aktor memilih menu data pinjaman	
	17. Menampilkan halaman data pinjaman
18. Aktor menekan tombol cari, untuk mengisi *field* nomor dan nama anggota.	
	19. Menampilkan isi *field* nomor dan nama anggota.
20. Aktor mengisi *field* lama angsuran pinjaman	
	21. Menampilkan isi *field* jasa, besar angsuran, provisi dan *field* simpanan khusus.
22. Aktor menekan tombol simpan	
	23. Mengkonfirmasi kesesuaian, jika sesuai akan menambahkan data pinjaman yang baru pada *database*..
	24. Menyimpan data pinjaman yang baru pada *database*

	25. Menampilkan data *history* pinjaman yang baru dari anggota tersebut.

4. Nama *Use Case* : Angsuran

   Aktor : Admin

   Tujuan : Aktor menambah data angsuran

Tabel 4.4 Skenario *Use Case* Angsuran

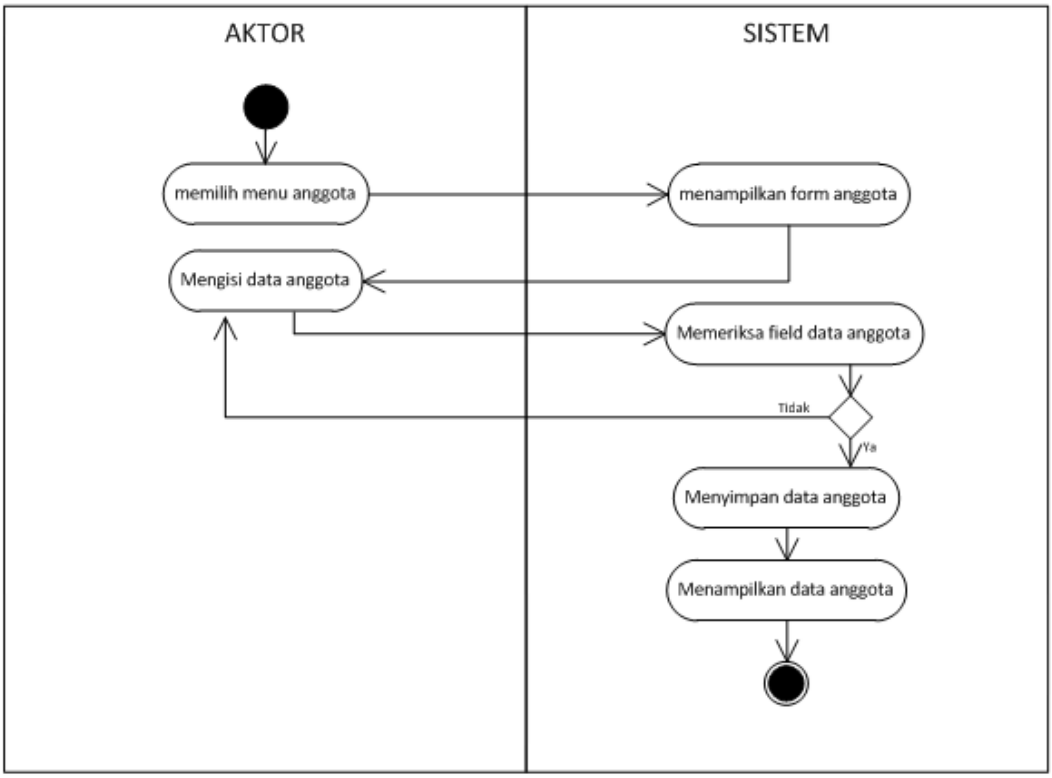
AKTOR	REAKSI SISTEM
1. Aktor memilih menu angsuran	
	2. Menampilkan halaman angsuran
3. Aktor menekan tombol cari, untuk mengisi nomor dan nama anggota	
	4. Menampilkan data angsuran untuk nomor/nama anggota yang dipilih.
5. Setelah itu, aktor mengisi *field* jumlah bayar serta mengisi detail pembayaran dan menekan tombol Simpan	
	6. memeriksa *field*, apa telah di isi dengan sesuai.
	7. Mengkonfirmasi kesesuaian, jika sesuai akan menambahkan data angsuran yang baru pada *database* .

	8. Menyimpan data angsuran yang baru pada *database*.
	9. Menampilkan data *history* angsuran dari anggota tersebut

4.1.3.3.*Activity* diagram

1. Pendaftaran

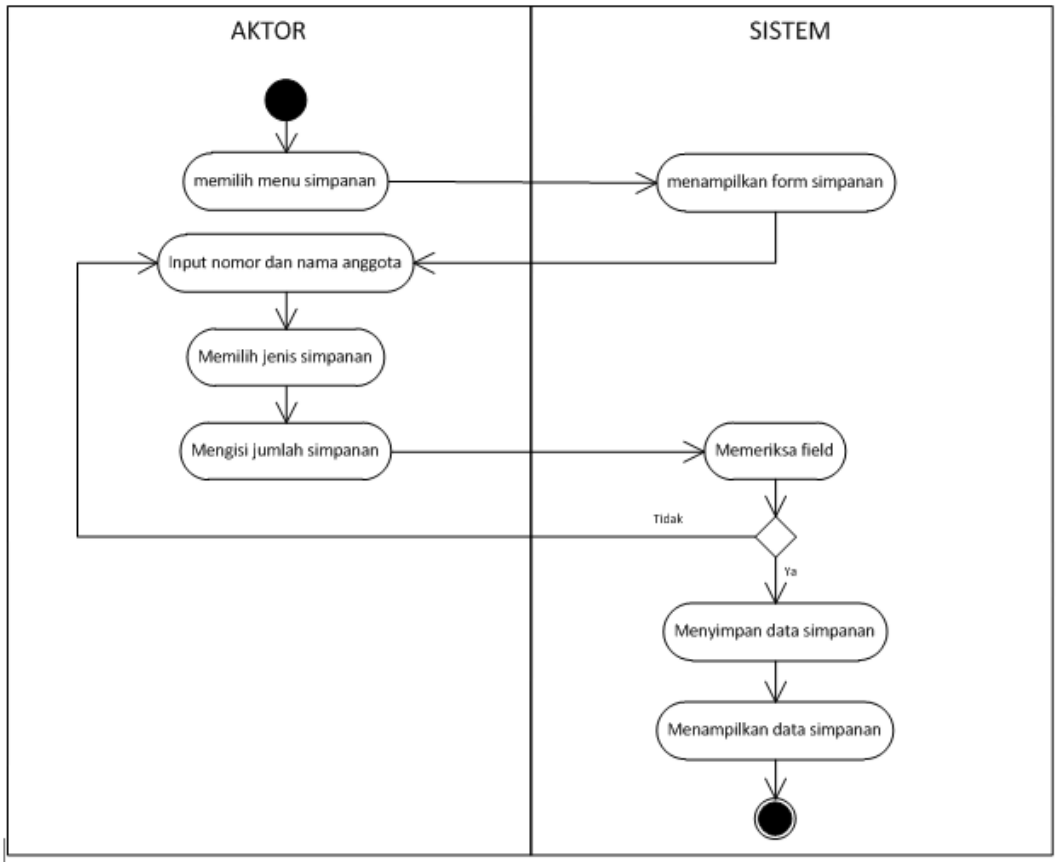
   Berikut ini penggambaran *activity* diagram Pendaftaran yang diusulkan adalah sebagai berikut :



Gambar 4.2 *Activity* Diagram Pendaftaran yang Diusulkan

2. Simpanan

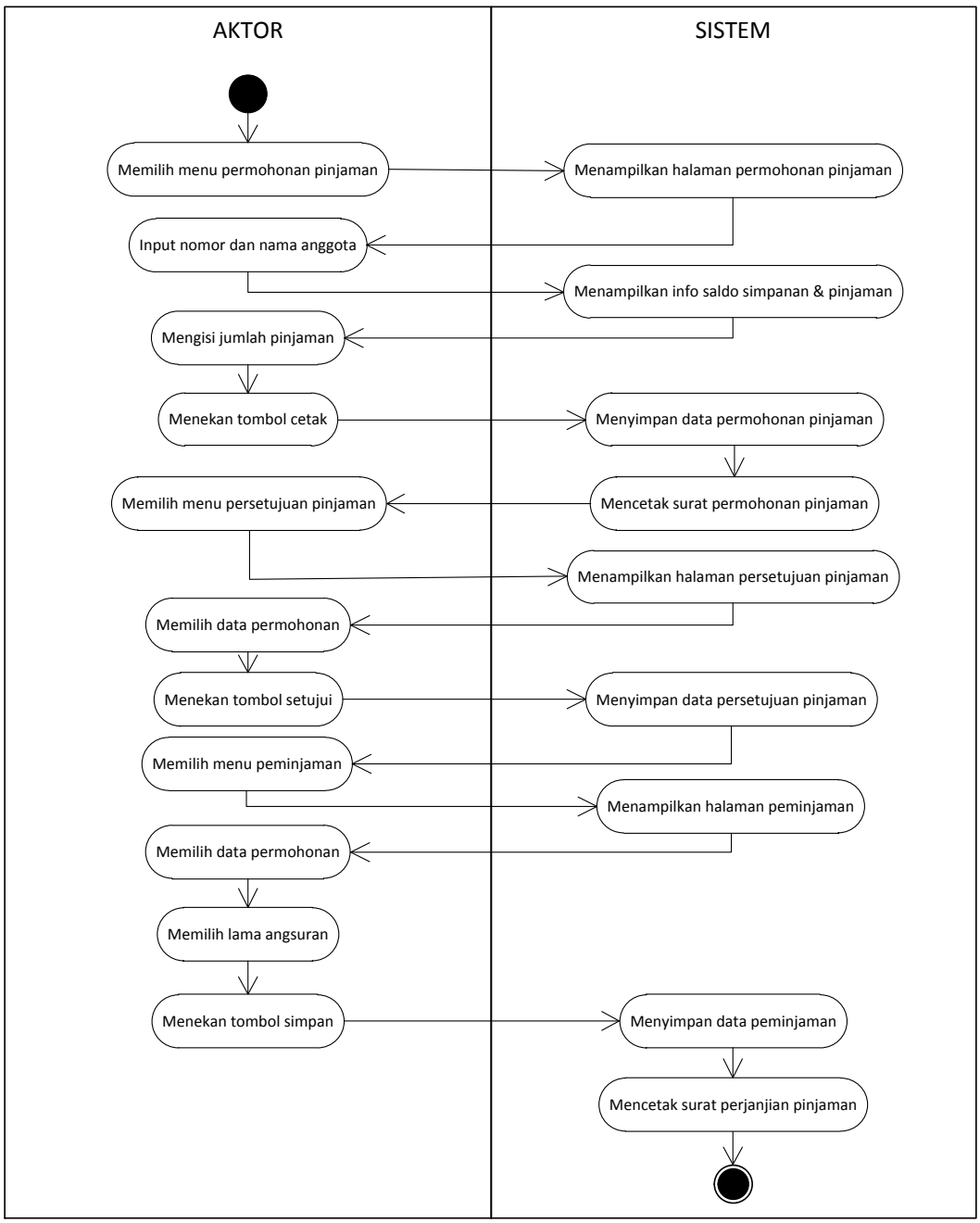
   Berikut ini penggambaran *activity* diagram Simpanan yang diusulkan adalah sebagai berikut



Gambar 4.3 *Activity* Diagram Simpanan yang Diusulkan

3. Pinjaman

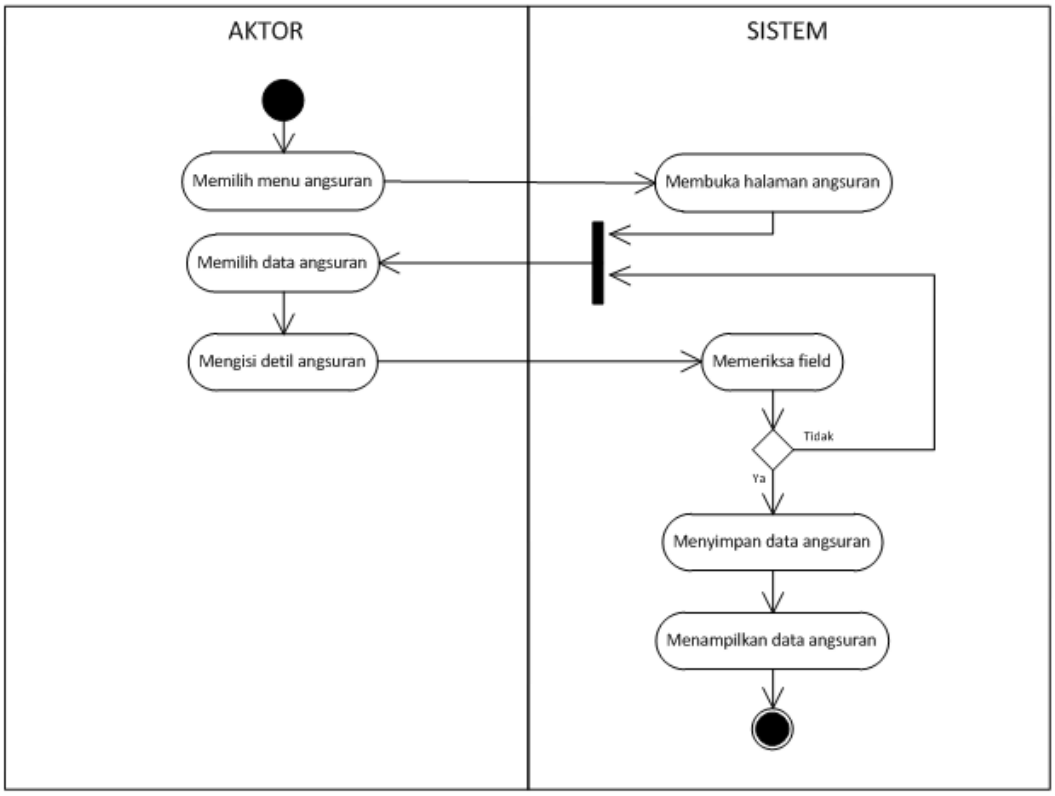
   Berikut ini penggambaran *activity* diagram Pinjaman yang diusulkan adalah sebagai berikut :



Gambar 4.4 *Activity* Diagram Pinjaman yang Diusulkan

4. Angsuran

Berikut ini penggambaran *activity* diagram Angsuran yang diusulkan adalah sebagai berikut :



Gambar 4.5 *Activity* Diagram Angsuran yang Diusulkan

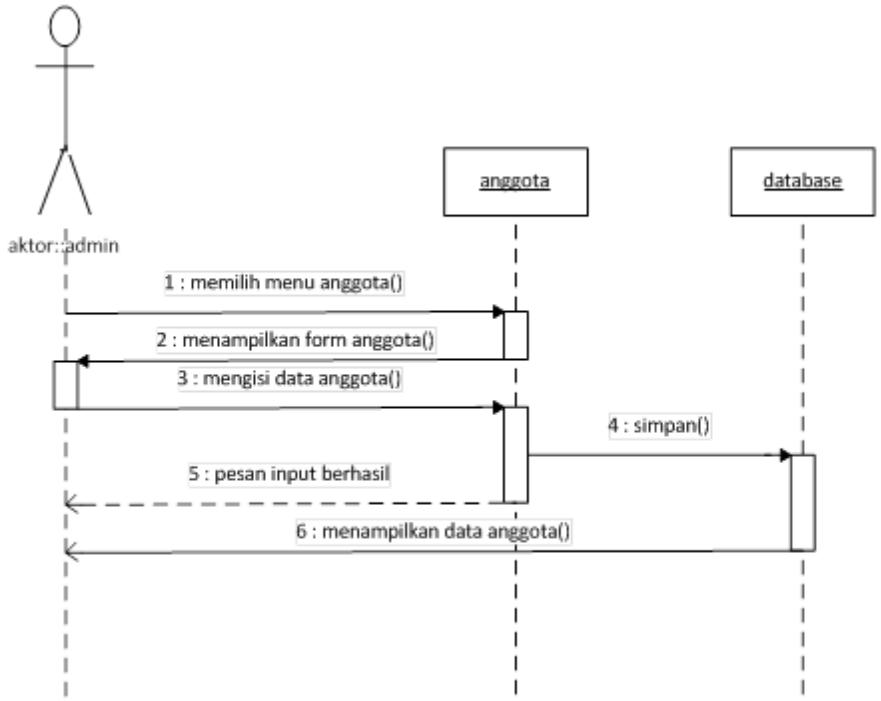
4.1.4. Perancangan basis data

Adapun perancangan basis data yang diusulkan pada Koperasi Harapan Warga adalah sebagai berikut :

4.1.4.1.*Sequence* diagram

1. *Sequence* Diagram Pendaftaran yang Diusulkan

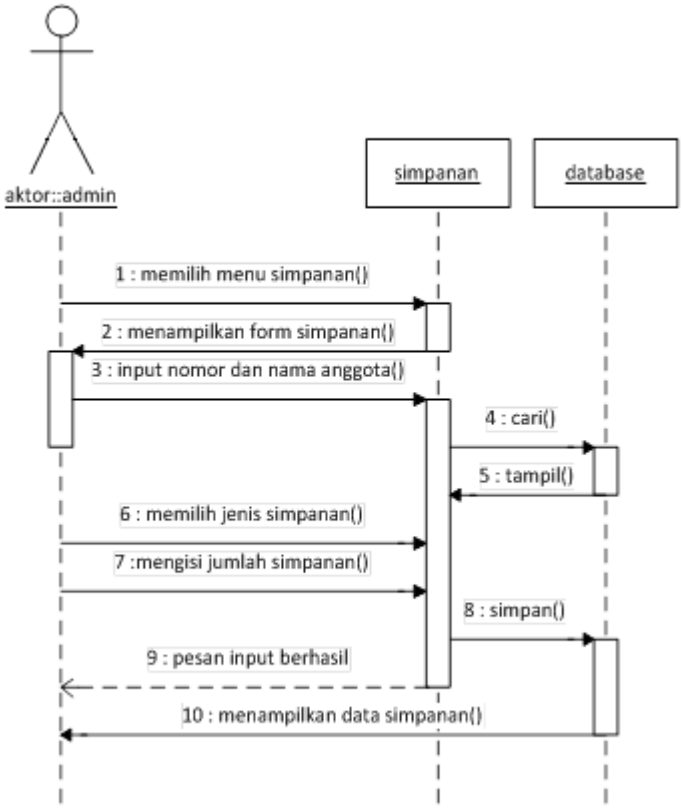
   Berikut ini penggambaran *Sequence* Diagram Pendaftaran yang diusulkan adalah sebagai berikut :



Gambar 4.6 *Sequence* Diagram Pendaftaran yang Diusulkan

2. *Sequence* Diagram Simpanan yang Diusulkan

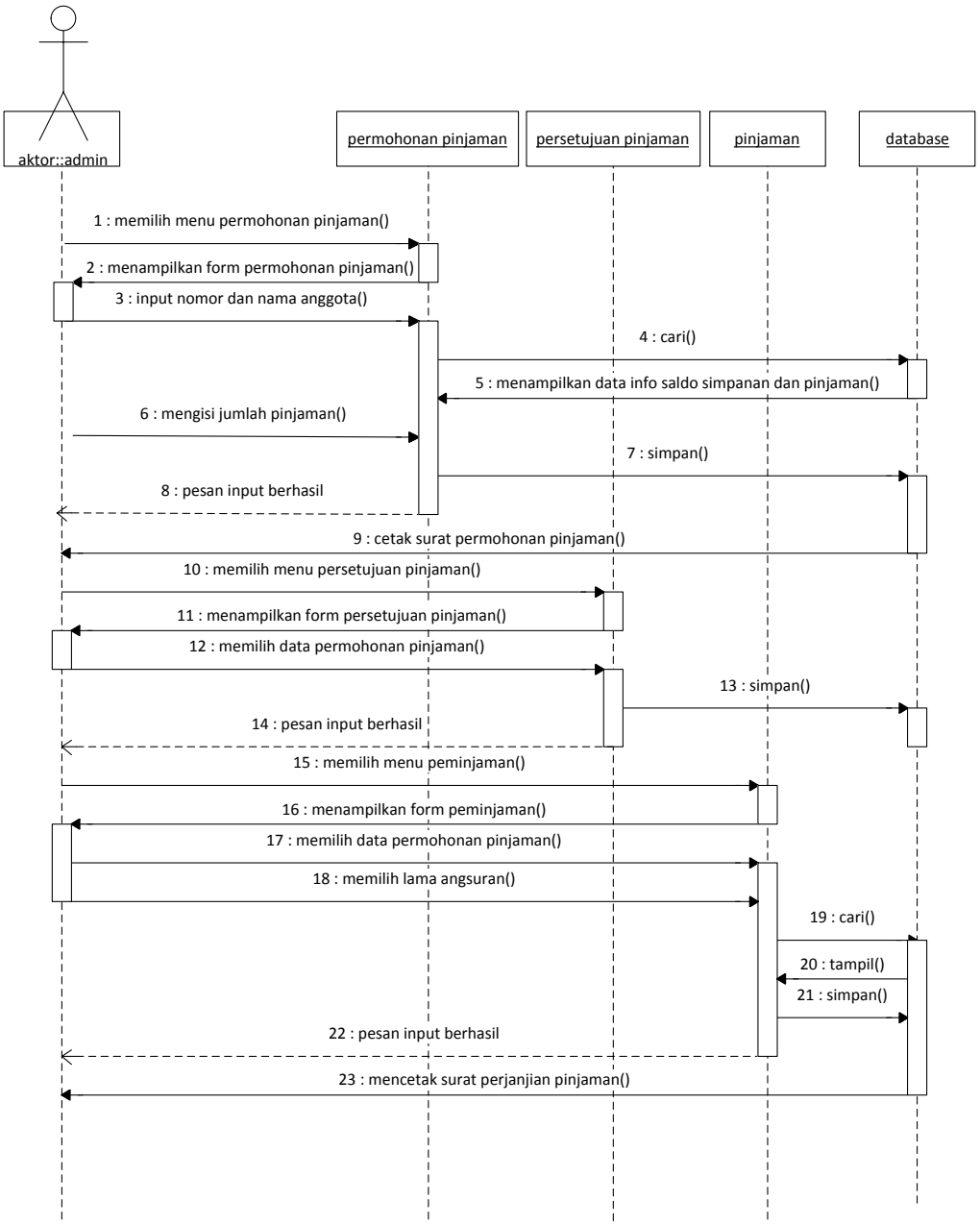
   Berikut ini penggambaran *Sequence* Diagram Simpanan yang diusulkan adalah sebagai berikut :



Gambar 4.7 *Sequence* Diagram Simpanan yang Diusulkan

3. *Sequence* Diagram Pinjaman yang Diusulkan

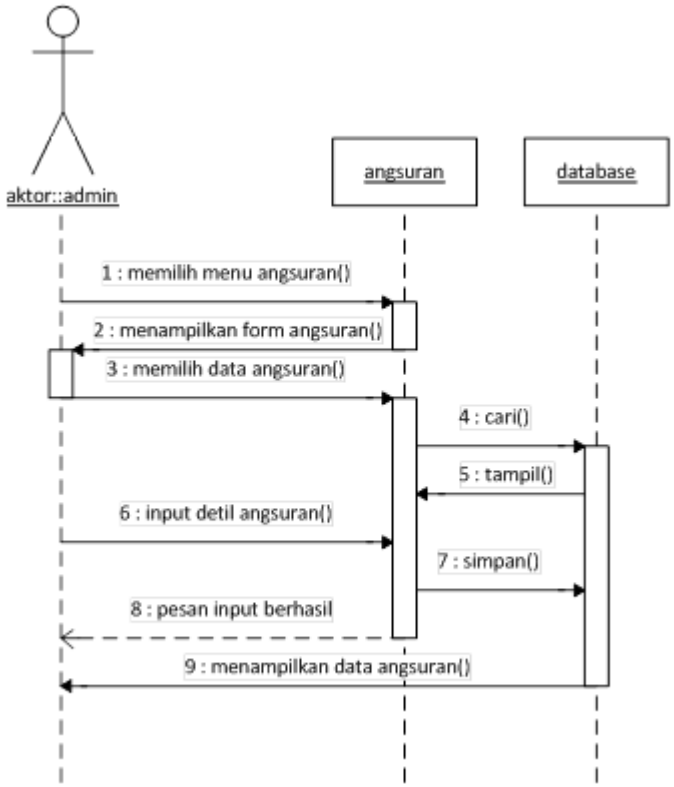
   Berikut ini penggambaran *Sequence* Diagram Olah Data Pinjaman yang diusulkan adalah sebagai berikut :



Gambar 4.8 Sequence Diagram Pinjaman yang Diusulkan

4. *Sequence* Diagram Angsuran yang Diusulkan

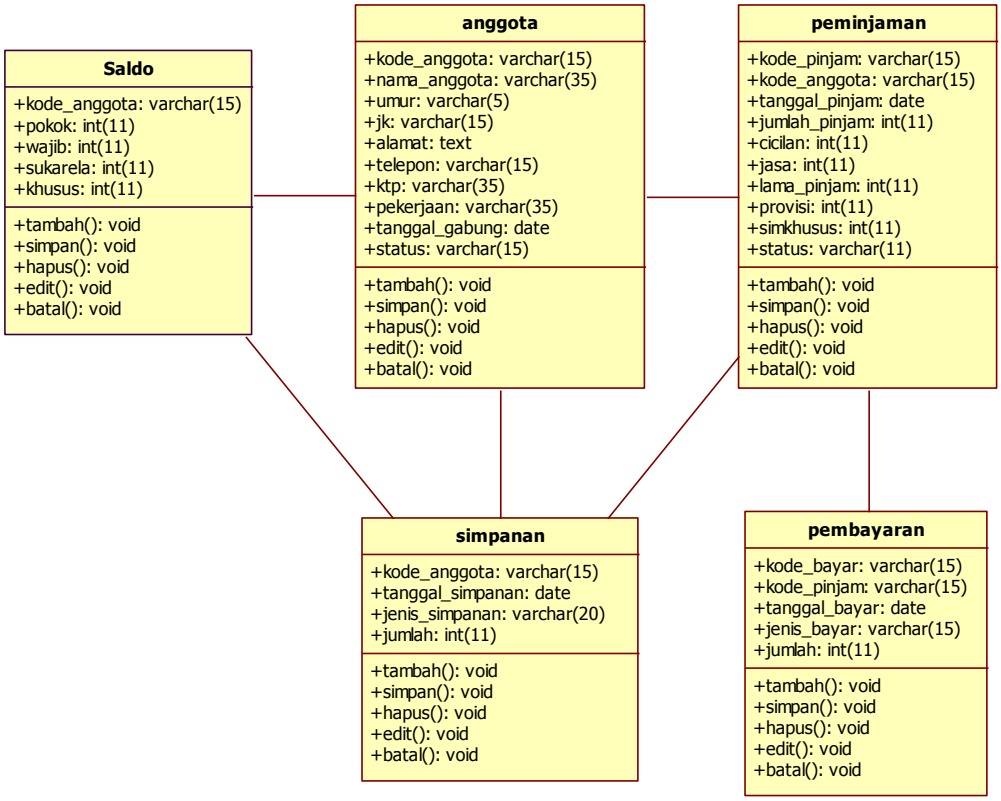
   Berikut ini penggambaran *Sequence* Diagram Angsuran yang diusulkan adalah sebagai berikut :



Gambar 4.9 *Sequence* Diagram Angsuran yang Diusulkan

4.1.4.2.*Class* diagram

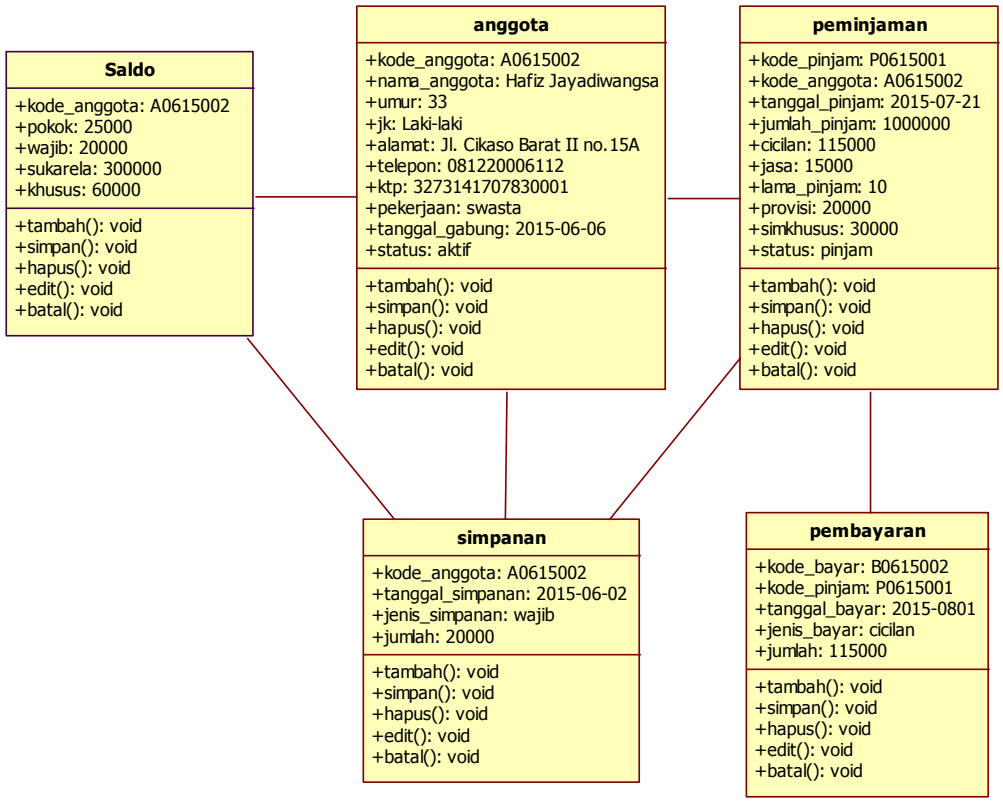
Menggambarkan struktur statis *class* di dalam sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem. Berikut *class* diagram yang diusulkan pada Sistem Informasi Koperasi Harapan Warga :



Gambar 4.10 *Class* Diagram

4.1.4.3.*Object* diagram

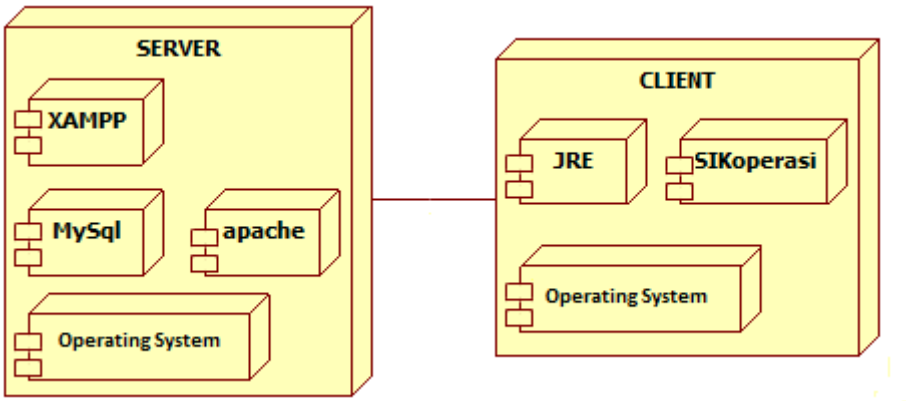
Menggambarkan struktur model sebuah sistem dalam kurun waktu tertentu, yang berasal dari class diagram sehingga bergantung pada *class* diagram. *Object* diagram merepresentasikan hubungan yang kompleks antara objek. Dengan melihat *class* diagram yang ada, maka dapat dibuat *object* diagram. Berikut *object* diagram yang diusulkan pada Sistem Informasi Koperasi Harapan Warga :



Gambar 4.11 *Object* Diagram

4.1.4.4.*Deployment* diagram

Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan. Diagram ini memuat simpul-simpul beserta komponen-komponen yang ada di dalamnya. *Deployment* diagram berhubungan dengan diagram komponen, dimana *deployment* diagram memuat satu atau lebih komponen-komponen.



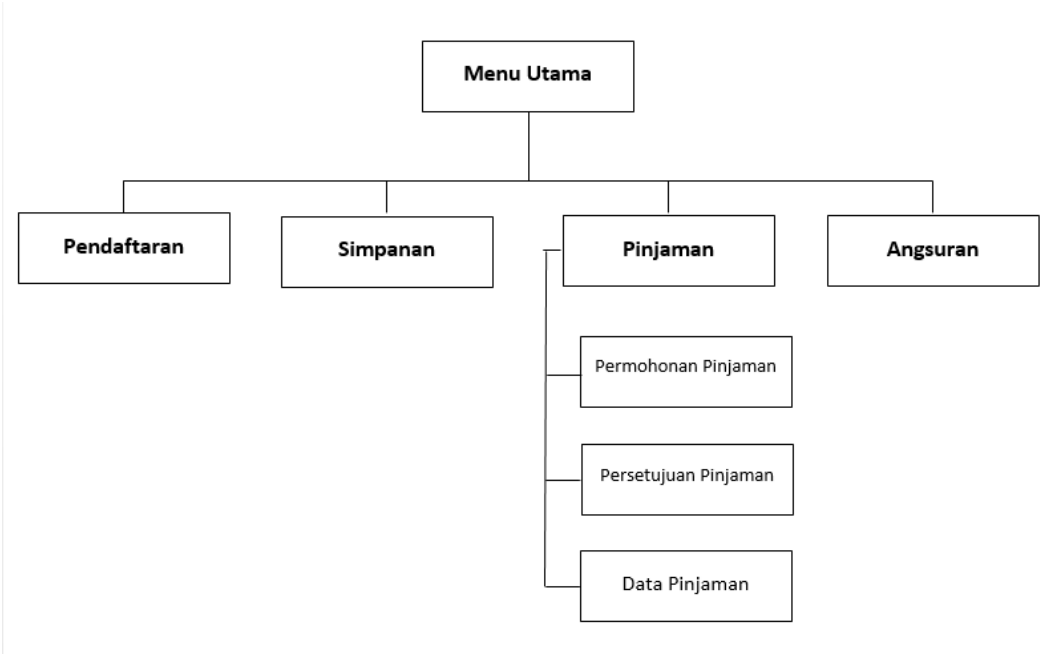
Gambar 4.12 *Deployment* Diagram

4.2. Perancangan Antar Muka

Perancangan antar muka merupakan perancangan yang dibuat sebelum program aplikasi dibuat, perancangan antar muka pada sistem informasi Koperasi Harapan Warga adalah sebagai berikut:

4.2.1. Struktur menu

Perancangan menu dibuat sebagai alat antar muka dengan pengguna untuk memudahkan pengoperasian perangkat lunak. Berikut rancangan menu perangkat lunak ini :



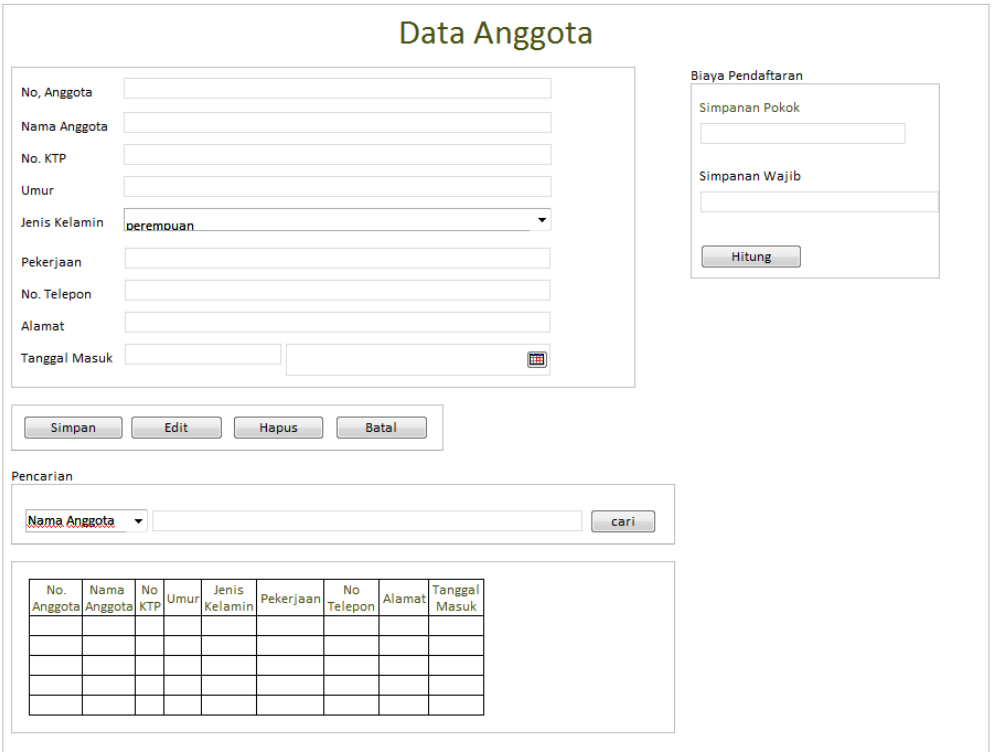
Gambar 4.13 Perancangan Struktur Menu Utama

4.2.2. Perancangan *input*

Perancangan input merupakan perancangan tampilan yang akan digunakan guna memasukkan data pada sistem untuk kemudian diproses. Dalam perancangan input ini, data yang dimasukkan akan mempengaruhi hasil yang ditampilkan. Adapun perancangan-perancangan input yang ada dalam perancangan ini adalah :

1. Perancangan Tampilan Pendaftaran Anggota

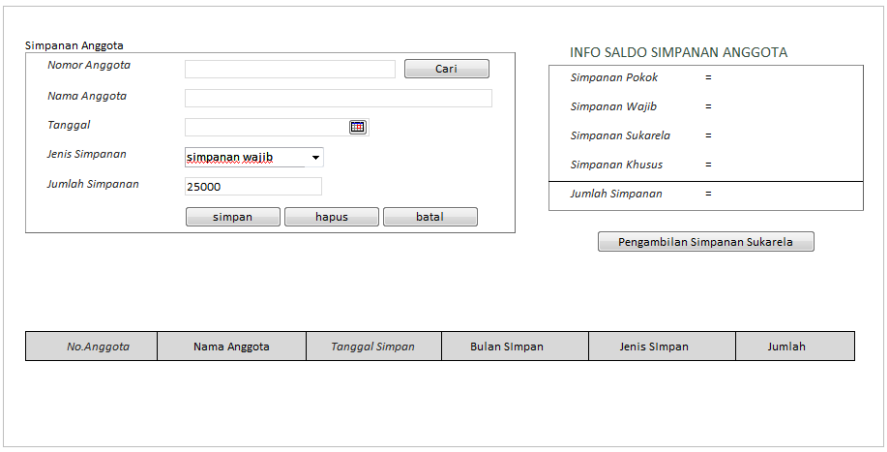
   Berikut ini perancangan tampilan pendaftaran anggota :



Gambar 4.14 Perancangan Tampilan Pendaftaran Anggota

2. Perancangan Tampilan Simpanan

Berikut ini perancangan tampilan simpanan :



Gambar 4.15 Perancangan Tampilan Simpanan

3. Perancangan Tampilan Permohonan Pinjaman

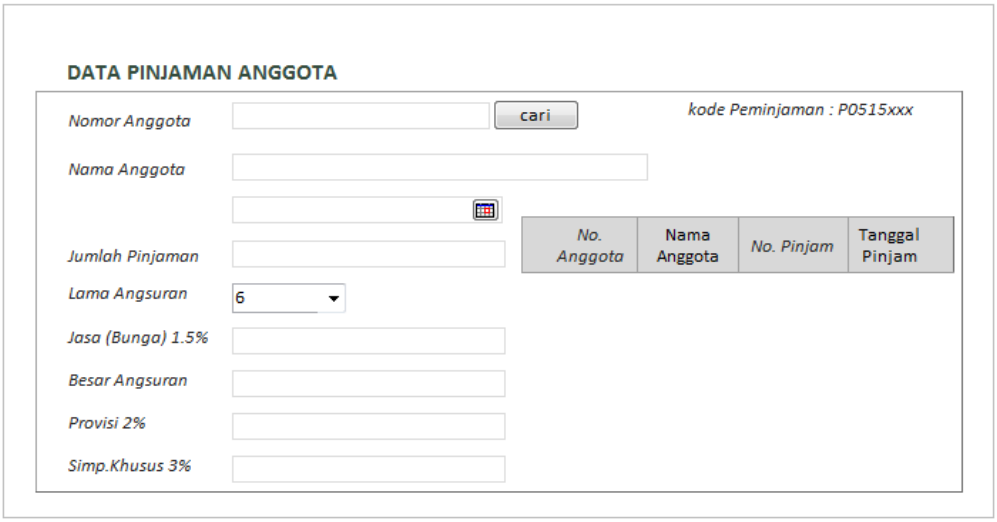
Berikut ini perancangan tampilan permohonan pinjaman :



Gambar 4.16 Perancangan Tampilan Permohonan Pinjaman

4. Perancangan Tampilan Data Pinjaman

Berikut ini perancangan tampilan data pinjaman :



Gambar 4.17 Perancangan Tampilan Data Pinjaman

5. Perancangan Tampilan Persetujuan Pinjaman

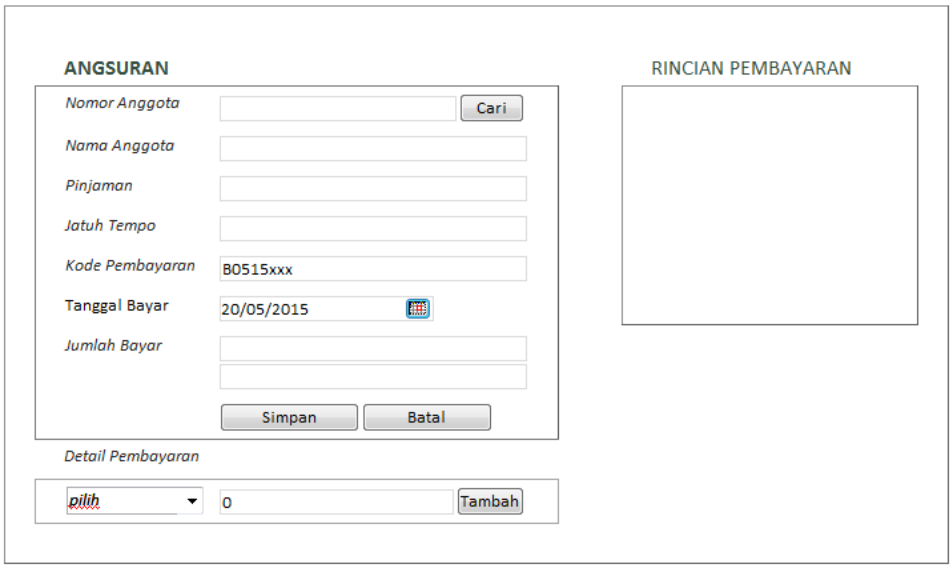
Berikut ini perancangan tampilan persetujuan pinjaman :



Gambar 4.18 Perancangan Tampilan Persetujuan Pinjaman

6. Perancangan Tampilan Pembayaran Angsuran

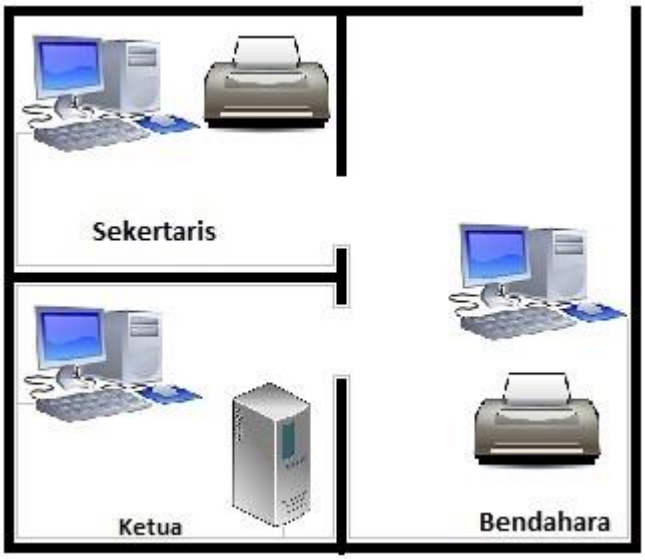
Berikut ini perancangan tampilan pembayaran angsuran :



Gambar 4.19 Perancangan Tampilan Pembayaran Angsuran

4.3. Perancangan Arsitektur Jaringan

Arsitektur jaringan yang digunakan adalah sistem *client/server* mempunyai dua komponen utama yaitu komputer *server* dan komputer *client*. *Server* merupakan komputer induk yang melakukan pemrosesan terbanyak untuk memenuhi permintaan-permintaan dari komputer client dan bertindak sebagai *server database* yang menyimpan data. *Client* yaitu komputer yang melakukan pengiriman permintaan-permintaan data pada *server* kemudian menampilkan data tersebut pada *interface* aplikasi yang dimilikinya.



Gambar 4.20 Arsitektur jaringan yang diusulkan

Gambar 4.20 menunjukkan arsitektur jaringan. Disini penulis menggunakan topologi star dengan alasan, jika salah satu computer *client* rusak, komputer yang lain masih dapat diakses dan control management lebih mudah karena semuanya terpusat ke server.

4.4. Pengujian

Pengujian dapat berarti proses untuk memeriksa apakah suatu perangkat lunak yang dihasilkan sudah dapat dijalankan sesuai dengan standar tertentu. Metode pengujian adalah cara atau teknik untuk menguji perangkat lunak, mempunyai mekanisme untuk menentukan data uji yang dapat menguji perangkat lunak secara lengkap dan mempunyai kemungkinan tinggi untuk menemukan kesalahan.

4.4.1. Rencana pengujian

Adapun hal-hal yang akan diuji melalui teknik pengujian black box adalah sebagai berikut :

Tabel 4.5 Rencana Pengujian

Kelas Uji	Butir Uji	Tingkat Pengujian	Jenis Pengujian
Pendaftaran	Input Data Anggota	Integrasi	*black box*
Simpanan	Input Data Simpanan	Integrasi	*black box*
Pinjaman	Input Data Pinjaman	Integrasi	*black box*
Angsuran	Input Data Angsuran	Integrasi	*black box*

4.4.2. Kasus dan Hasil Pengujian

1. Pengujian input data anggota

Pengujian ini adalah pengujian dengan memasukkan sejumlah data ke dalam database, dengan beberapa validasi.

Tabel 4.6 Pengujian *Input* Data Anggota

Kasus dan Hasil Uji Data Anggota (Data Normal)			
Data Masukkan	Yang Diharapkan	Pengamatan	Kesimpulan
Isi data Anggota	Data masuk ke form isian data anggota dan dapat kode secara otomatis.	Data masuk ke form isian data anggota dan dapat kode secara otomatis.	[X] Diterima [ ] Ditolak
Klik Tombol Simpan	Data Anggota masuk ke dalam *Database*.	Data Anggota masuk ke dalam *Database*.	[X] Diterima [ ] Ditolak
Kasus dan Hasil Uji Data Anggota (Data Salah)			
Data Masukkan	Yang Diharapkan	Pengamatan	Kesimpulan
Data tidak diisi secara lengkap, tekan tombol simpan	Gagal dalam menyimpan dalam *database* dan muncul kotak pesan dimana lokasi *field* belum diisi.	Gagal dalam menyimpan dalam *database* dan muncul kotak pesan dimana lokasi *field* belum diisi.	[X] Diterima [ ] Ditolak

2. Pengujian input data simpanan

Pengujian ini adalah pengujian dengan memasukkan sejumlah data ke dalam database, dengan beberapa validasi. Dengan mengisikan jumlah simpanan sukarela di form simpanan.

Tabel 4.7 Pengujian *Input* Data Simpanan

Kasus dan Hasil Uji Data Simpanan (Data Normal)			
Data Masukkan	Yang Diharapkan	Pengamatan	Kesimpulan
Isi data simpanan dan klik tombol simpan	Data masuk ke form isian dan nomor simpanan, jumlah simpanan wajib otomatis muncul dalam form, kemudian data tersimpan dalam *database*.	Data masuk ke form isian dan nomor simpanan, jumlah simpanan wajib otomatis muncul dalam form, kemudian data tersimpan dalam *database*.	[X] Diterima [ ] Ditolak
Kasus dan Hasil Uji Data Simpanan (Data Salah)			
Data Masukkan	Yang Diharapkan	Pengamatan	Kesimpulan
Data tidak diisi secara lengkap, tekan tombol simpan	Gagal dalam menyimpan dalam *database* dan muncul kotak pesan dimana lokasi *field* belum diisi.	Gagal dalam menyimpan dalam *database* dan muncul kotak pesan dimana lokasi *field* belum diisi.	[X] Diterima [ ] Ditolak

3. Pengujian input data pinjaman

Pengujian ini adalah pengujian dengan memasukkan sejumlah data ke dalam database, dengan beberapa validasi. Dengan mengisikan data permohonan pinjaman, persetujuan pinjaman dan data pinjaman di form pinjaman.

Tabel 4.8 Pengujian *Input* Data Pinjaman

Kasus dan Hasil Uji Data Pinjaman (Data Normal)			
Data Masukkan	Yang Diharapkan	Pengamatan	Kesimpulan
Isi data permohonan pinjaman dan klik tombol simpan	Data masuk ke form isian, kemudian data permohonan pinjaman tersimpan dalam *database*.	Data masuk ke form isian, kemudian data permohonan pinjaman tersimpan dalam *database*.	[X] Diterima [ ] Ditolak
pilih data permohonan pinjaman dan klik tombol setujui	Data persetujuan pinjaman tersimpan dalam *database*.	Datapersetujuan pinjaman tersimpan dalam *database*.	[X] Diterima [ ] Ditolak
Isi data pinjaman dan pilih lama angsuran	Data masuk ke form isian dan jasa, besar angsuran, provisi, simpanan khusus otomatis muncul dalam form.	Data masuk ke form isian dan jasa, besar angsuran, provisi, simpanan khusus otomatis muncul dalam form.	[X] Diterima [ ] Ditolak
Klik Tombol Simpan	Data pinjaman masuk ke dalam *Database*.	Data pinjaman masuk ke dalam *Database*.	[X] Diterima [ ] Ditolak
Kasus dan Hasil Uji Data Pinjaman (Data Salah)			
Data Masukkan	Yang Diharapkan	Pengamatan	Kesimpulan
Data permohonan pinjaman tidak diisi secara lengkap, tekan tombol simpan	Gagal dalam menyimpan dalam *database* dan muncul kotak pesan dimana lokasi *field* belum diisi.	Gagal dalam menyimpan dalam *database* dan muncul kotak pesan dimana lokasi *field* belum diisi.	[X] Diterima [ ] Ditolak
Tidak memilih data permohonan pinjaman, klik tombol setujui	Gagal dalam menyimpan dalam *database* dan muncul kotak pesan dimana lokasi *field* belum diisi.	Gagal dalam menyimpan dalam *database* dan muncul kotak pesan dimana lokasi *field* belum diisi.	[X] Diterima [ ] Ditolak
Data pinjaman tidak diisi secara lengkap,pilih lama angsuran, tekan tombol simpan	Gagal dalam menyimpan dalam *database* dan muncul kotak pesan dimana lokasi *field* belum diisi.	Gagal dalam menyimpan dalam *database* dan muncul kotak pesan dimana lokasi *field* belum diisi.	[X] Diterima [ ] Ditolak

4. Pengujian input data angsuran

Pengujian ini adalah pengujian dengan memasukkan sejumlah data ke dalam database, dengan beberapa validasi. Dengan mengisikan data anguran di form angsuran.

Tabel 4.9 Pengujian *Input* Data Angsuran

Kasus dan Hasil Uji Data Angsuran (Data Normal)			
Data Masukkan	Yang Diharapkan	Pengamatan	Kesimpulan
Isi data angsuran dan Klik Tombol Simpan	Data masuk ke form isian dan nomor angsuran otomatis muncul dalam form kemudian data angsuran masuk ke dalam *Database*	Data masuk ke form isian dan nomor angsuran otomatis muncul dalam form kemudian data angsuran masuk ke dalam *Database*	[X] Diterima [ ] Ditolak
Kasus dan Hasil Uji Data Angsuran (Data Salah)			
Data Masukkan	Yang Diharapkan	Pengamatan	Kesimpulan
Data tidak diisi secara lengkap, tekan tombol simpan	Gagal dalam menyimpan dalam *database* dan muncul kotak pesan dimana lokasi *field* belum diisi.	Gagal dalam menyimpan dalam *database* dan muncul kotak pesan dimana lokasi *field* belum diisi.	[X] Diterima [ ] Ditolak

4.4.3. Kesimpulan hasil pengujian

Berdasarkan hasil pengujian dengan kasus uji *sample* yang telah dilakukan di atas dapat disimpulkan bahwa perangkat lunak Sistem Informasi Simpan Pinjam pada Koperasi Harapan Warga layak dan sesuai dengan fungsi yang dibutuhkan serta dapat berjalan dengan baik juga mengeluarkan hasil sesuai dengan yang diharapkan. Diharapkan dari pengujian tersebut dapat mewakili pengujian fungsi lainnya dalam perangkat lunak yang digunakan Koperasi Harapan Warga.

4.5. Implementasi

Implementasi merupakan proses untuk memastikan pelaksanaan ide, program atau seperangkat aktivitas baru dengan harapan orang lain dapat menerima dan melakukan perubahan. Keberhasilan implementasi dapat dilihat dari program tersebut dirancang dengan landasan yang jelas, dengan kelompok sasaran, dan tujuan yang jelas.

4.5.1. Batasan implementasi (optional)

Dalam implementasi perangkat lunak, pemantauan dan pengendalian aplikasi ini ada beberapa hal yang menjadi batasan, yaitu :

1. Implementasi perangkat lunak ini difokuskan pada proses pendaftaran (data anggota), simpanan, penarikan, peminjaman, angsuran serta pembuatan laporan.
2. Implementasi yang dilakukan hanya membahas implementasi secara langsung di Koperasi Harapan Warga dan perancangan pada tahap pembuatan aplikasi.
3. Implementasi pembuatan *software* Sistem Informasi Simpan Pinjam ini menggunakan *Java Development Kit* 6.9 (JDK6.9) dan menggunakan *tools Netbeans IDE 8.0.2* sebagai editor serta *database* yang digunakan adalah Xampp dan MySql.

4.5.2. Implementasi perangkat lunak

Unruk implementasi perangkat lunak pada *server* menggunakan MySql, apache, XAMPP. Sedangkan perangkat lunak yang diperlukan *client* adalah JRE, dan aplikasi SIKoperasi. *Operating System* yang digunakan adalah *Microsoft Windows 7*

4.5.3. Implementasi perangkat keras

Perangkat keras yang dibutuhkan berdasarkan kebutuhan minimal yang harus dipenuhi adalah sebagai berikut :

1. *Server* :
    a. *Proccessor* Intel Pentium IV 1.7 Mhz atau setara.
    b. *Memory* 1024 MB atau lebih.
    c. *Harddisk* 80 GB atau lebih.
    d. *Mouse, Keyboard, monitor* dan *Printer*
    e. CD-ROM Drive.
2. *Client* :
    a. *Proccessor* Intel Pentium IV 1.7 Mhz atau setara.
    b. *Memory* 512 MB atau lebih.
    c. *Harddisk* 40 GB atau lebih.
    d. *Mouse, Keyboard, monitor* dan *Printer*
    e. CD-ROM Drive.
3. Perangkat Jaringan
    a. Kabel UTP dan Konektor RJ45
    b. Hub 5 Port

4.5.4. Implementasi basis data (sintaks SQL)

Pembahasan pembuatan basis data akan dibahas dengan menampilkan desain *physical* dari basis data, dimana aplikasi pemrograman yang dipakai adalah *mySQL*, Implementasi basis data dalam *mySQL* adalah sebagai berikut :

```
--
-- Struktur dari tabel `anggota`
--

CREATE TABLE IF NOT EXISTS `anggota` (
  `kode_anggota` varchar(15) NOT NULL,
  `nama_anggota` varchar(35) NOT NULL,
  `umur` varchar(5) NOT NULL,
  `jk` varchar(15) NOT NULL,
  `alamat` text NOT NULL,
  `telepon` varchar(15) NOT NULL,
  `ktp` varchar(35) NOT NULL,
  `pekerjaan` varchar(35) NOT NULL,
  `tanggal_gabung` date NOT NULL,
  `status` varchar(15) NOT NULL,
  PRIMARY KEY (`kode_anggota`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;


--
-- Struktur dari tabel `simpanan`
--

CREATE TABLE IF NOT EXISTS `simpanan` (
  `kode_anggota` varchar(15) NOT NULL,
  `tanggal_simpan` date NOT NULL,
  `jenis_simpanan` varchar(20) NOT NULL,
  `jumlah` int(11) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
--
-- Struktur dari tabel `peminjaman`
--

CREATE TABLE IF NOT EXISTS `peminjaman` (
  `kode_pinjam` varchar(15) NOT NULL,
  `kode_anggota` varchar(15) NOT NULL,
  `tanggal_pinjam` date NOT NULL,
  `jumlah_pinjam` int(11) NOT NULL,
  `cicilan` int(11) NOT NULL,
  `jasa` int(11) NOT NULL,
  `lama_pinjam` int(11) NOT NULL,
  `provisi` int(11) NOT NULL,
  `simkhusus` int(11) NOT NULL,
  `status` varchar(11) NOT NULL,
  PRIMARY KEY (`kode_pinjam`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;


--
-- Struktur dari tabel `pembayaran`
--

CREATE TABLE IF NOT EXISTS `pembayaran` (
  `kode_bayar` varchar(15) NOT NULL,
  `kode_pinjam` varchar(15) NOT NULL,
  `tanggal_bayar` date NOT NULL,
  `jenis_bayar` varchar(15) NOT NULL,
  `jumlah` int(11) NOT NULL,
  PRIMARY KEY (`kode_bayar`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

4.5.5. Implementasi antar muka

Berikut adalah penjelasan implementasi antar muka struktur menu pada perangkat lunak Sistem Informasi Simpan Pinjam Koperasi Harapan Warga.

4.5.5.1.Implementasi Halaman Utama

Halaman utama berisi menu-menu yang dapat digunakan oleh pengguna dalam mengelola kegiatan simpan pinjam. Berikut ini adalah uraian menu yang terdapat pada halaman utama.

Tabel 4.10 Implementasi Halaman Utama

Sub Menu	Keterangan	Nama File
Pendaftaran	Digunakan untuk memanggil *form* anggota	Menu.java
Simpanan	Digunakan untuk memanggil *form* simpanan	Menu.java
Pinjaman	Digunakan untuk memanggil *form* pinjaman	Menu.java
Angsuran	Digunakan untuk memanggil *form* angsuran	Menu.java

4.5.5.2.Implementasi *form* pinjaman

Form pinjaman berisi kegiatan permohonan pinjaman, persetujuan pinjaman dan data pinjaman.

Tabel 4.11 Implementasi *Form* Pinjaman

Sub Menu	Keterangan	Nama File
Permohonan Pinjaman	Digunakan untuk memanggil *form* permohonan pinjaman	Peminjaman.java
Persetujuan Pinjaman	Digunakan untuk memanggil *form* persetujuan pinjaman	Peminjaman.java
Data Pinjaman	Digunakan untuk memanggil *form* data pinjaman	Peminjaman.java

4.5.6. Implementasi instalasi program

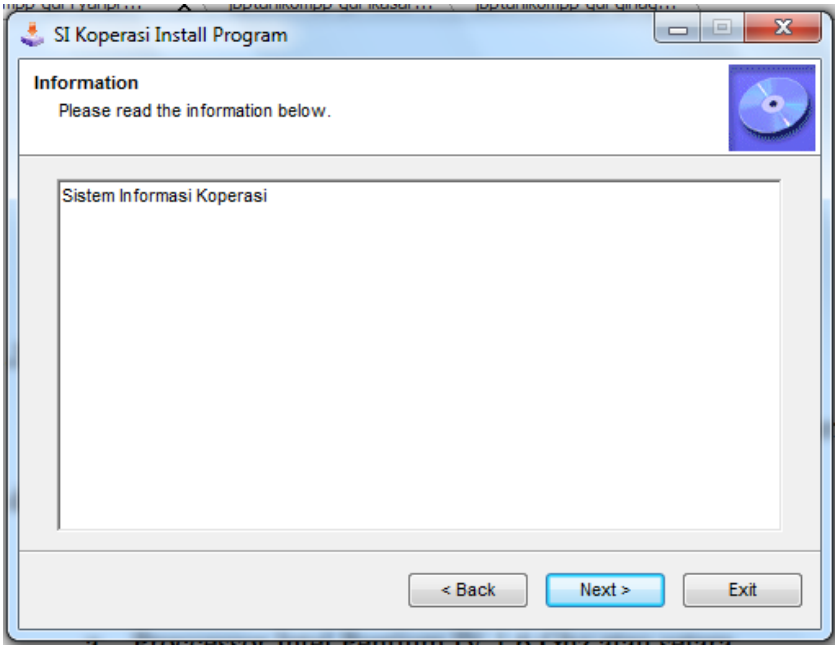
Berikut ini adalah implementasi instalasi program Sistem Informasi Simpan Pinjam pada Koperasi Harapan Warga.

1. Ketika menjalankan setup.exe maka akan muncul tampilan pertama kali sebagai berikut, kemudian pilih next.



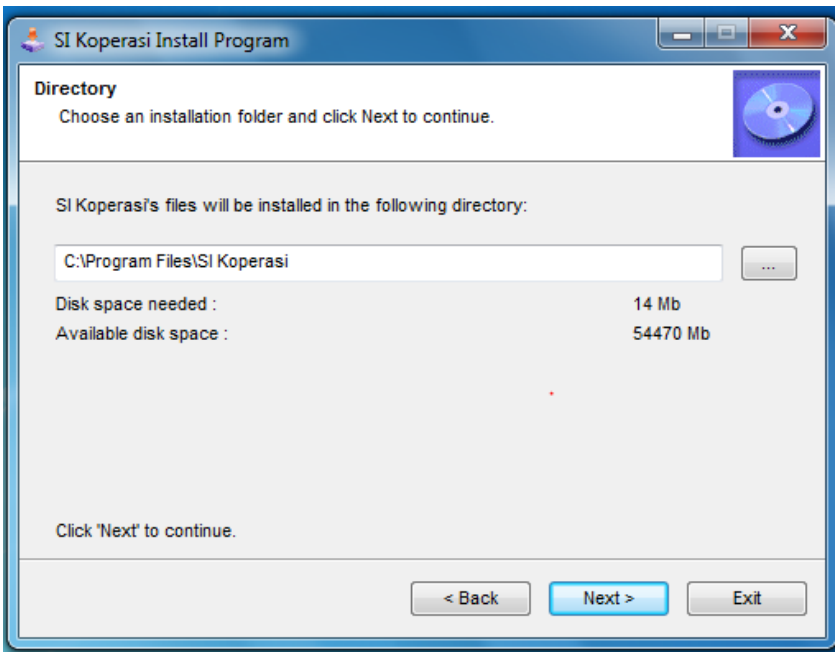
Gambar 4.21 Kotak *Dialog Setup Welcome*

2. Kemudian akan muncul tampilan seperti gambar 4.22, kemudian pilih next.



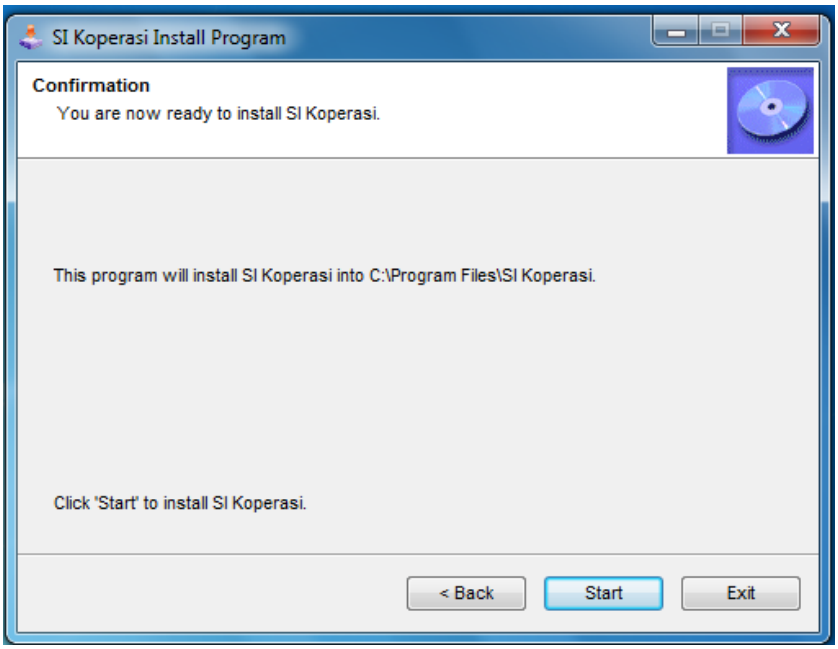
Gambar 4.22 Kotak *Dialog Setup Information*

3. Setelah itu akan muncul tampilan setup *directory* untuk memilih tempat penyimpanan folder instalasi. Kemudian klik next.



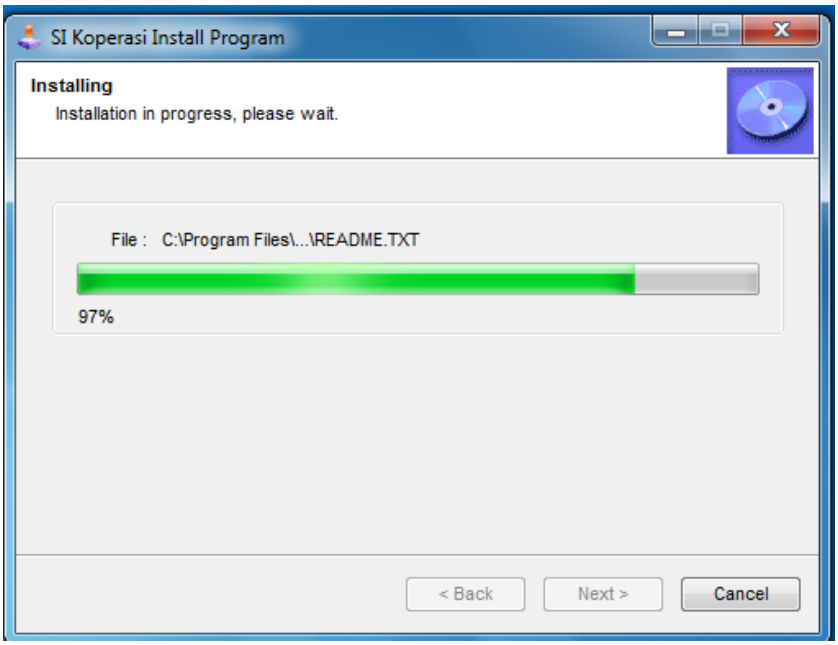
Gambar 4.23 Kotak *Dialog Setup Directory*

4. Lalu klik tombol start untuk memulai installasi pada gambar 4.24 seperti berikut ini.



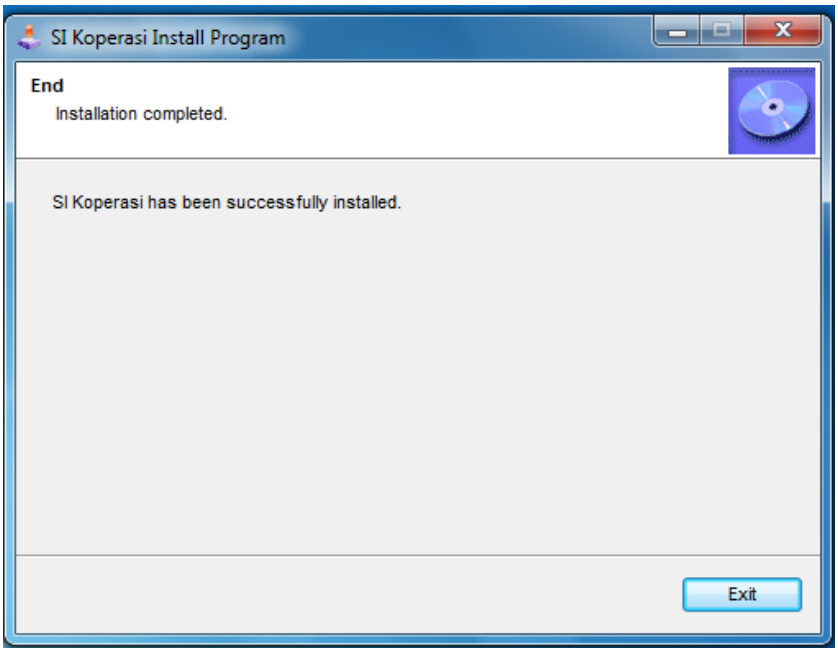
Gambar 4.24 Kotak *Dialog Confirmation*

5. Berikut adalah proses instalasi, tunggu hingga proses selesai.



Gambar 4.25 Kotak *Dialog Installing*

6. Setelah menunggu beberapa menit sampai proses installasi selesai, kemudian akan muncul seperti gambar di bawah ini lalu klik Exit dan Aplikasi siap digunakan.



Gambar 4.26 Kotak *Dialog End*

4.5.7. Penggunaan program

Kegiatan ini bertujuan untuk menerangkan secara singkat penggunaan program Simpa Pinjam. Adapun cara pengoperasiannya adalah sebagai berikut :

1. *Login*

   Ketika pertama kali program applikasi ini dijalankan maka yang pertama kali muncul adalah *form login* seperti di bawah ini. Halaman ini merupakan halaman yang berisikan menu masuk pada *form* utama untuk melakukan pengolahan data sesuai dengan hak akses yang dimiliki *user*.



Gambar 4.27 *Form Login*

2. *Form* Utama

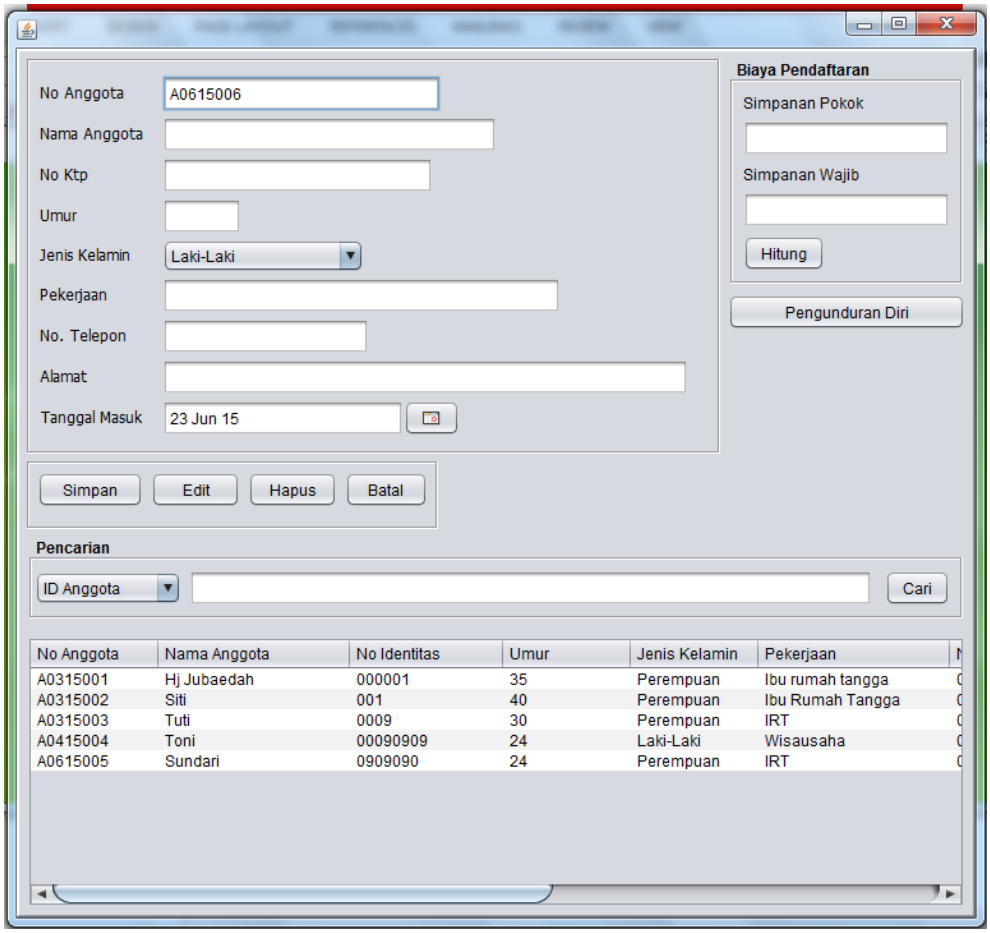
Jika *user* menekan tombol login pada *form logi*n dan berhasil melakukan proses autentikasi *user*, maka *form* utama akan tampil. *Form* utama ini berisi menu-menu yang dapat diakses oleh *user* tersebut.



Gambar 4.28 *Form* Utama

3. *Form* Anggota

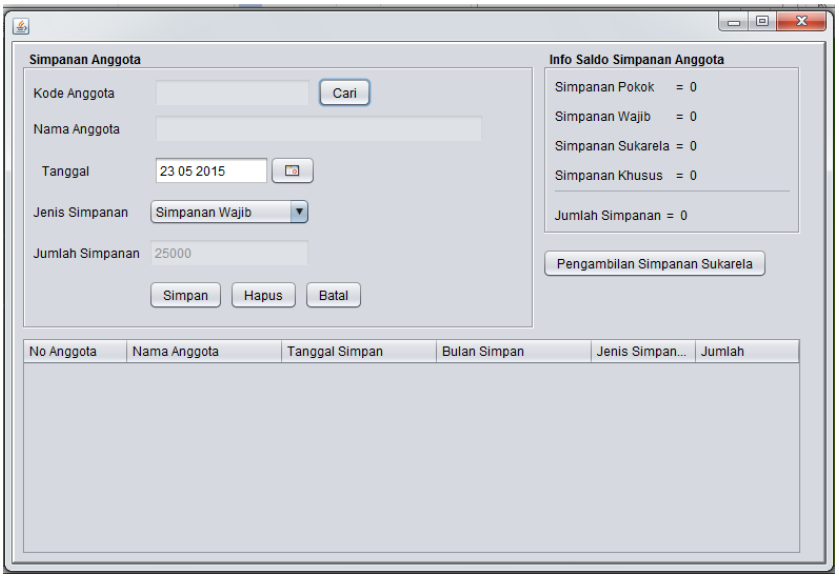
Berikut adalah *form* pendaftaran anggota yang berfungsi untuk menambahkan, mengedit data ataupun menghapus data anggota.



Gambar 4.29 *Form* Anggota

4. *Form* Simpanan

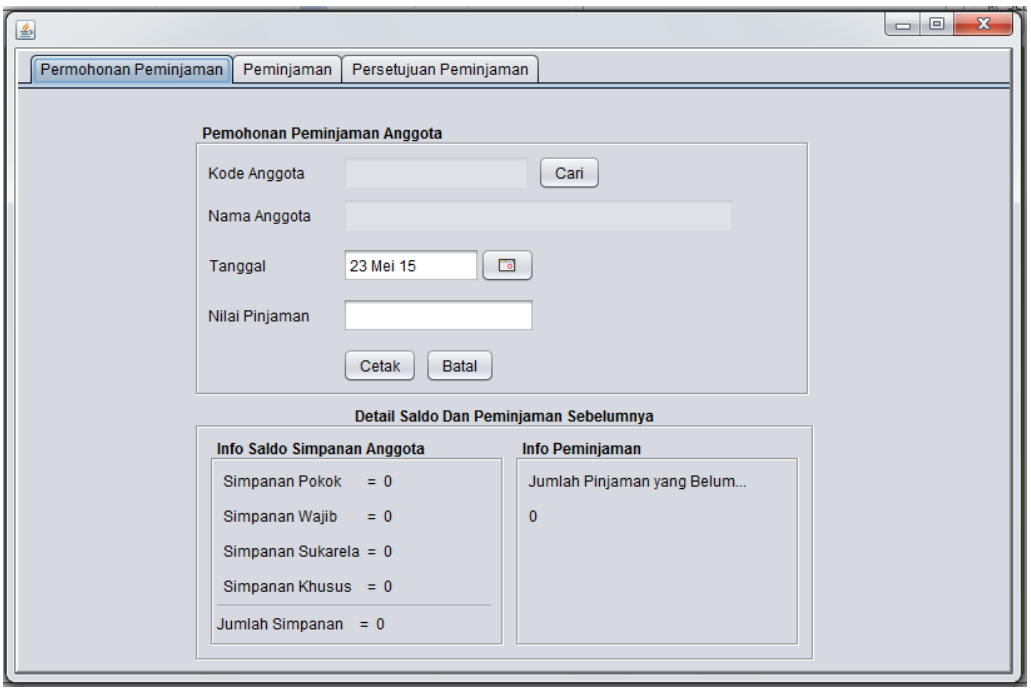
Berikut adalah *Form* Simpanan yang berfungsi untuk melakukan penambahan simpanan anggota dan pemilihan *Form* penarikan simpanan.



Gambar 4.30 *Form* Simpanan

5. *Form* Permohonan Pinjaman

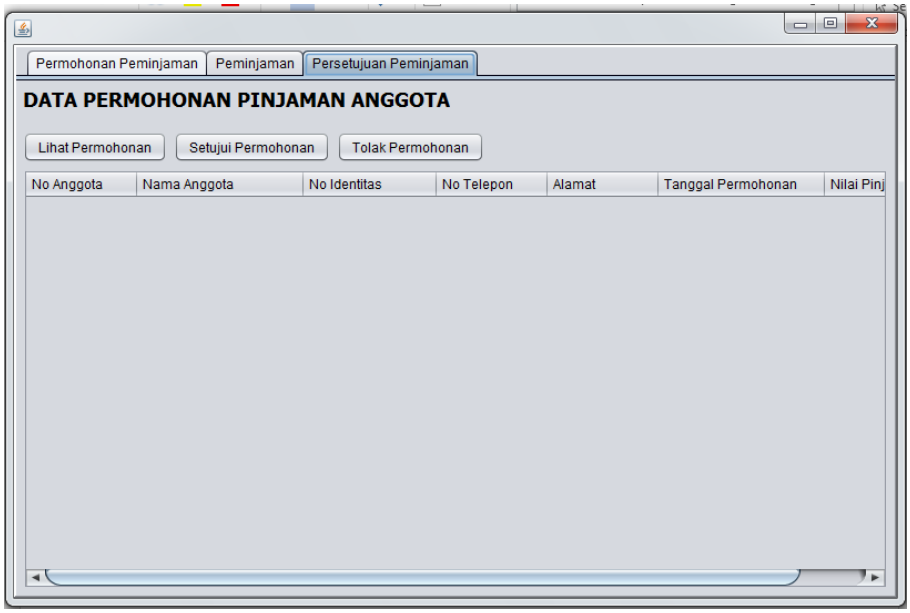
Berikut adalah *Form* permohonan pinjaman yang berfungsi untuk melakukan penambahan permohonan pinjaman.



Gambar 4.31 *Form* Permohonan Pinjaman

6. *Form* Persetujuan Pinjaman

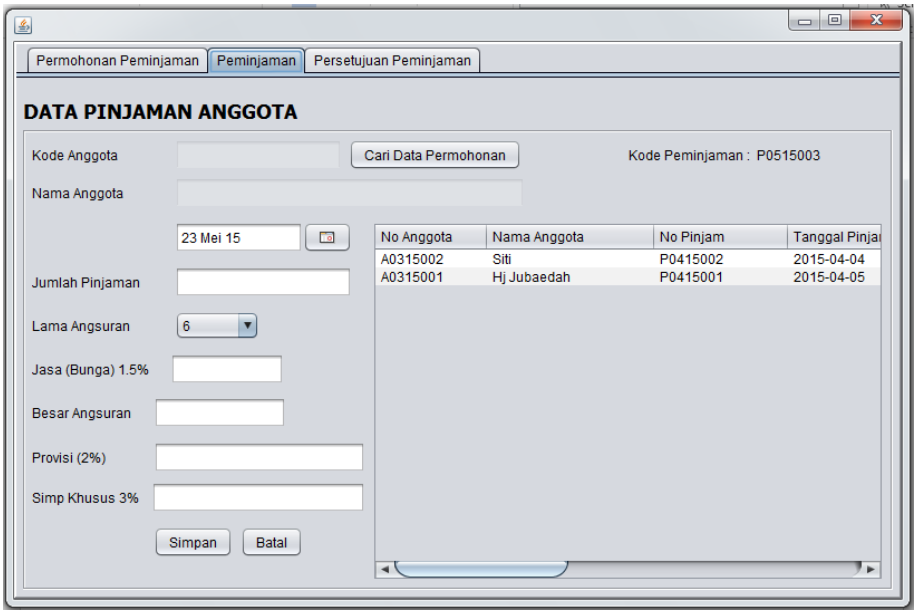
Berikut adalah *Form* persetujuan simpanan yang berfungsi untuk melakukan persetujuan pinjaman.



Gambar 4.32 *Form* Persetujuan Pinjaman

7. *Form* Data Pinjaman

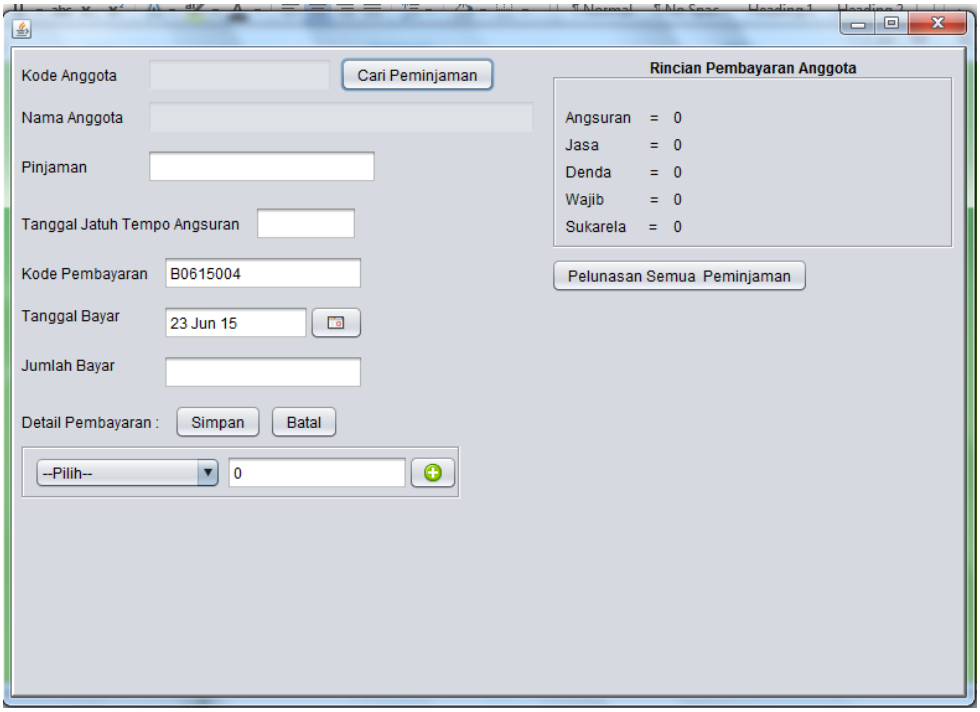
Berikut adalah *Form* data pinjaman yang berfungsi untuk menentukan besar angsuran dan lama angsuran, jika permohonan pinjaman telah disetujui.



Gambar 4.33 *Form* Permohonan Pinjaman

8. *Form* Angsuran

Berikut adalah *Form* Angsuran yang berfungsi untuk melakukan penambahan pembayaran angsuran anggota.



Gambar 4.34 *Form* Angsuran