**BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang Masalah**

Pada zaman globalisasi saat ini, kemajuan atas segala bidang aspek kehidupan, telah menampakkan pengaruhnya pada kehidupan masyarakat. Namun, perubahan pesat yang begitu jelas terlihat dan dapat dirasakan yakni perubahan pada teknologi. Dimana dapat dibuktikan bahwa gaya hidup hampir seluruh lapisan masyarakat saat ini tidak dapat lepas dari peran teknologi, baik setiap individu maupun suatu organisasi. Mulai dari telepon genggam yang mempunyai fitur canggih dan memudahkan penggunanya dalam mendapatkan informasi hingga komputer dengan sistem canggih yang dapat mengatur kinerja dari suatu perusahaan sehingga pekerjaan para karyawan menjadi lebih efektif, dan perubahan lainnya yang tanpa disadari telah memudahkan manusia dalam melakukan sesuatu.

Namun, di zaman yang identik dengan perubahan pesat ini, masih saja ditemukan individu maupun organisasi/lembaga yang membudayakan cara “lama” atau “manual” dalam kinerjanya. Salah satu bukti nyata yakni Koperasi Karyawan DKB Kantor Pusat (Kapus) pada PT. Dok & Perkapalan Kodja Bahari. Adapun Koperasi Karyawan DKB Kantor Pusat (Kapus) merupakan koperasi yang tidak hanya menyajikan jasa simpan pinjam bagi karyawan maupun anggota koperasi, melainkan dapat pula menyajikan pengadaan kebutuhan perusahaan seperti alat tulis (ATK), atau pembuatan kartu nama hingga pengadaan jasa ketenagaan pekerja seperti tenaga supir, hingga tukang kebun.

Kemudian, dalam mengelola data-data transaksi, maupun mengelola data-data anggotanya, pihak koperasi masih menuangkannya dalam sebuah buku atau lembaran dokumen. Hal ini tentunya mengakibatkan seringkali terjadinya kekeliruan atau kesalahan dalam pencatatan data transaksi, dan bahkan mempersulit anggota koperasi dalam mencari data-data yang dibutuhkan.

Adapun pengertian dari Koperasi merupakan suatu organisasi bisnis yang dimiliki serta di operasikan oleh sekelompok individu dengan tujuan demi kepentingan bersama serta mewujudkan asas-asas dari suatu organisasi koperasi. Kemudian, kegiatan-kegiatan yang biasanya dilakukan oleh para anggota koperasi terkait dengan mengelola keuangan setiap anggota. Sehingga dapat diartikan bahwa kegiatan yang dilakukan sangatlah kompleks, sensitif, dan membutuhkan tingkat ketelitian yang maksimal.

Maka dari itu, setelah mengetahui kegiatan beserta faktor-faktor penghambat tersebut, penulis tertarik untuk membantu menyelesaikan permasalahan yang ada pada Koperasi Karyawan DKB Kantor Pusat (Kapus). Dengan dibuatnya sistem komputerisasi transaksi simpan pinjam, diharapkan dapat pula untuk dapat semakin meningkatkan kinerja dari para anggotanya terutama dalam hal penyimpanan data atau dokumen-dokumen penting agar dapat relatif cepat serta akurat dalam hal penyajian datanya.

* 1. **Alasan Pemilihan Objek**

Alasan penulis memilih objek yang terkait yakni :

1. Tempat observasi yang bersangkutan merupakan tempat bekerja dari salah seorang anggota keluarga, sehingga lebih mudah untuk mendapatkan dokumen-dokumen yang dibutuhkan guna kebutuhan penyusunan Tugas Akhir.
2. Ingin mengetahui lebih mendalam mengenai sistem transaksi simpan pinjam pada suatu organisasi Koperasi.
3. Objek yang terkait dianggap patut dibahas oleh penulis. Hal ini dikarenakan masih ditemukan adanya kesenjangan antara harapan dari pihak koperasi mengenai sistem transaksi simpan pinjam yang diterapkan dengan keadaan yang sebenarnya, setelah sistem tersebut diterapkan.

* 1. **Tujuan dan Manfaat Penulisan**
     1. **Tujuan Penulisan**

1. Untuk mengetahui sistem transaksi simpan pinjam yang saat ini diterapkan pada Koperasi Karyawan DKB Kantor Pusat (Kapus).
2. Untuk mengetahui faktor penghambat yang didapat anggota koperasi selama menerapkan sistem transaksi yang saat ini berjalan.
3. Untuk menemukan solusi dan merancang sebuah sistem informasi transaksi simpan pinjam yang sesuai dengan harapan pihak Koperasi Karyawan DKB Kantor Pusat (Kapus).
   * 1. **Manfaat Penulisan**
4. **Bagi Perusahaan**
5. Sebagai bahan pertimbangan atas masalah-masalah yang didapat guna mengoptimalkan kembali mutu serta kualitas pelayanan pada Koperasi Karyawan DKB Kantor Pusat (Kapus).
6. Dapat menghasilkan laporan transaksi yang akurat.
7. Dapat mempermudah para anggota koperasi dalam mengelola data-data yang bersangkutan.
8. Dapat mengurangi adanya kekeliruan atau kesalahan dalam pencatatan data transaksi.
9. **Bagi Dunia Pendidikan**
10. Sebagai bahan referensi (informasi tambahan) bagi mahasiswa/i atas berbagai laporan-laporan sejenis yang telah dibuat sebelumnya.
11. Dapat menjadi bahan pertimbangan bagi lembaga-lembaga pendidikan guna nantinya diterapkan dalam dunia pendidikan serta sebagai bagian dari solusi terhadap permasalahan pendidikan yang ada.
12. **Bagi Penulis**
13. Menambah wawasan dalam bidang perancangan suatu sistem komputerisasi simpan pinjam pada organisasi koperasi.
14. Melatih kemampuan penulis dalam menganalisa suatu permasalahan, merancang suatu program yang diharapkan pihak *client*, berpikir kritis dalam memutuskan, dsb.
15. Dapat menerapkan segala ilmu pengetahuan yang telah didapat selama kegiatan perkuliahan berlangsung.
    1. **Identifikasi Masalah**

Berdasarkan latar belakang masalah yang telah dipaparkan, maka didapat beberapa identifikasi masalah guna membantu penulis dalam memecahkan permasalahan yang ada serta membantu dalam perancangan sistem yang diharapkan oleh pihak Koperasi Karyawan DKB Kantor Pusat (Kapus). Adapun identifikasi masalah yang didapat yakni sebagai berikut.

1. Bagaimana sistem transaksi simpan pinjam yang sedang berjalan pada Koperasi Karyawan DKB Kantor Pusat (Kapus) ?
2. Apa saja hambatan-hambatan yang didapat dalam sistem transaksi simpan pinjam yang sedang berjalan pada Koperasi Karyawan DKB Kantor Pusat (Kapus) ?
3. Bagaimana solusi sistem transaksi simpan pinjam yang diharapkan oleh pihak Koperasi Karyawan DKB Kantor Pusat (Kapus) ?
   1. **Batasan Masalah**

Guna mempermudah penulis dalam penyusunan Tugas Akhir ini, ruang lingkup atas masalah-masalah yang telah diutarakan saat pengamatan dilakukan, maka didapat batasan-batasan masalah dimana sesuai dengan pokok judul yang telah ditentukan penulis. Adapun batasan masalah meliputi :

1. Prosedur Pendaftaran Anggota Koperasi
2. Prosedur Simpanan Pokok/Wajib/Sukarela
3. Prosedur Permohonan Pinjaman
4. Prosedur Pembayaran Angsuran Pinjaman
   1. **Metodologi Penulisan**

Metode yang digunakan penulis dalam pengumpulan data/dokumen yang berkaitan dengan transaksi simpan pinjam koperasi yakni sebagai berikut.

* + 1. **Studi Kepustakaan (*Library Research*)**

Merupakan metode pengumpulan data dengan mencari referensi-referensi melalui media cetak yang berhubungan dengan penelitian yang dilakukan. Adapun metode studi pustaka merupakan metode penunjang daripada metode pengamatan serta metode wawancara. Untuk kegiatan yang dilakukan penulis dalam hal melaksanakan metode studi pustaka yakni dengan mengklasifikasikan bahan-bahan yang behubungan dengan permasalahan, baik dari sumber dokumen maupun media cetak.

* + 1. **Studi Lapangan (*Field Research*)**

1. **Pengamatan (*Observation*)**

Merupakan metode pengumpulan data dengan mengamati suatu objek secara langsung menggunakan panca indra. Adapun dengan melakukan pengamatan (*Observation*), penulis dapat lebih mengetahui tentang fakta dari suatu alur maupun data. Dalam metode ini, kegiatan yang dilakukan penulis yakni mencatat objek-objek yang berkaitan dengan transaksi simpan pinjam, serta mengamati dan mencatat prosedur-prosedur untuk melakukan simpan pinjam pada Koperasi Karyawan DKB Kantor Pusat (Kapus).

1. **Wawancara (*Interview*)**

Merupakan metode pengumpulan data dengan jalan mengadakan komunikasi secara lisan dengan pihak narasumber. Komunikasi yang biasa dilakukan yakni dengan bertatap muka dan melakukan tanya-jawab secara langsung antara pihak narasumber dan pengumpul data. Adapun bersamaan dengan penjelasan diatas, guna melaksanakan metode pengumpulan data dengan wawancara, kegiatan yang dilakukan penulis meliputi melakukan tatap muka dengan pihak-pihak pengurus Koperasi seperti Kepala Koperasi, bendahara, serta salah satu anggota koperasi. Selain itu, kegiatan yang dilakukan penulis yakni melakukan tanya-jawab kepada pihak-pihak yang terkait mengenai transaksi simpan pinjam pada Koperasi Karyawan DKB Kantor Pusat (Kapus).

* 1. **Sistematika Penulisan**

Dalam penyusunan Tugas Akhir, penulis membagi pokok bahasan menjadi beberapa bab, yakni sebagai berikut.

**BAB I PENDAHULUAN**

Pada bab ini, penulis akan menjelaskan mengenai Latar Belakang Masalah, Alasan Pemilihan Objek, Identifikasi Masalah, Tujuan dan Manfaat Penulisan, Batasan Masalah, Metodologi Penulisan, dan Sistematika Penulisan.

**BAB II LANDASAN TEORI**

Untuk bab ini, hal yang dibahas yakni menyangkut tentang teori-teori yang digunakan dalam menyelesaikan permasalahan. Selain itu, membahas pula mengenai pengertian dari istilah-istilah yang digunakan dalam penulisan laporan.

**BAB III PROFIL PERUSAHAAN**

Didalam bab ini, penulis menjelaskan mengenai profil dari perusahaan, dimana tempat dilaksanakannya pengamatan (*Observation*). Pada bab ini, akan dibahas hal-hal yang meliputi Sejarah Perusahaan, Struktur Organisasi, hingga penjelasan detail mengenai pekerjaan dari masing-masing jabatan.

**BAB IV PEMBAHASAN**

Pada bab ini, penulis mengungkapkan, menjelaskan, serta membahas karakteristik dari objek yang telah diamati secara statistik. Didalamnya meliputi analisa sistem berjalan, merancang basis data, merancang sistem program komputerisasi mulai dari format *input* hingga format *output,* dsb.

**BAB V PENUTUP**

Didalam bab penutup terdapat sub bab kesimpulan serta saran. Dimana didalam setiap sub bab, penulis menyatakan pemahamannya mengenai masalah-masalah yang telah diteliti. Adapun pada sub bab kesimpulan, penulis menjelaskan kesimpulan yang didapat setelah tahap penyusunan Tugas Akhir ini. Kemudian, pada sub bab saran, penulis mengungkapkan saran-sarannya baik kepada pihak penulis lainnya, perusahaan, maupun pihak Politeknik LP3I Jakarta Kampus Pondok Gede.

**BAB II**

**LANDASAN TEORI**

* 1. **Perancangan Sistem Informasi**

Perancangan sistem informasi diibaratkan dengan pengembangan sistem baru dari sistem yang lama/sudah ada, dimana masalah-masalah yang terjadi pada sistem lama diharapkan sudah dapat teratasi pada sistem yang baru.

Adapun pengertian dari Perancangan dan Sistem Informasi secara terpisah yakni sebagai berikut :

* + 1. **Pengertian Perancangan**

Dalam bukunya, I Putu Eka Pratama (2014:33) mengungkapkan mengenai pengertian dari perancangan yakni bahwa :

“Perancangan merupakan suatu tahap yang mencakup desain dan implementasi perangkat lunak beserta dengan desain dan implementasi jaringan komputer untuk media pelengkap perangkat lunak yang bersangkutan (jika diperlukan)”.

* + 1. **Pengertian Sistem Informasi**

Sistem informasi adalah kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi itu untuk mendukung operasi dan manajemen. Adapun pandangan lain mengenai pengertian sistem informasi menurut Nazruddin Safaat H.MT. (2015:8) yang mengungkapkan bahwa :

“Sesungguhnya, yang dimaksud dengan sistem informasi tidak harus melibatkan komputer. Sistem Informasi mencakup sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran dan tujuan”.

Di dalam suatu sistem informasi terdiri atas komponen-komponen yang disebut dengan blok bangunan (*building block*), yang terdiri atas komponen *input*, komponen model, komponen *output,* komponen teknologi, komponen *hardware*, komponen *software,* komponen basis data serta komponen kontrol. Adapun, komponen-komponen tersebut saling berinteraksi antara yang satu dengan yang lainnya untuk dapat membentuk suatu kesatuan guna mencapai sasaran.

Berikut ini merupakan komponen-komponen yang terdapat pada sebuah sistem yakni meliputi :

1. Komponen *Input* mewakili data yang masuk ke dalam sistem informasi. *Input* dalam hal ini termasuk metode dan media untuk menangkap data yang akan dimasukkan, dan biasanya dapat berupa dokumen-dokumen dasar.
2. KomponenModelmerupakan komponen yang terdiri atas kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara yang telah ditentukan untuk menghasilkan keluaran yang diinginkan.
3. Komponen *Output* merupakan hasil informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.
4. Komponen Teknologi adalah alat dalam sistem informasi yang digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan *output* serta memantau pengendalian sistem secara keseluruhan.
5. Komponen *Hardware* merupakan komponen yang dijadikan sebagai tempat/sarana untuk menampung *database* atau lebih mudah dikatakan sebagai sumber data dan informasi untuk memperlancar dan mempermudah kerja dari sistem informasi.
6. Komponen *Software* adalah suatu komponen yang berfungsi sebagai tempat untuk mengolah, menghitung, dan memanipulasi data yang diambil dari *hardware* untuk dapat menciptakan suatu informasi.
7. Komponen Basis Data merupakan komponen yang terdiri atas kumpulan data yang saling berhubungan serta yang tersimpan di dalam komputer dan menggunakan *software* untuk memanipulasinya.
8. Komponen Kontrol merupakan komponen yang dapat mengendalikan gangguan terhadap sistem informasi.

Selain, komponen yang terdapat dalam suatu sistem informasi, terdapat pula karakteristik dari sistem informasi. Dimana suatu sistem bisa dikatakan sebagai sebuah sistem informasi apabila memenuhi karakteristik utama dari sebuah sistem informasi.

Berikut ini merupakan karakteristik yang dimiliki suatu sistem informasi.

1. Memiliki komponen

Komponen bisa disebut juga sebagai subsistem di dalam sebuah sistem informasi yang memiliki sifat untuk menjalankan fungsi-fungsi tertentu di dalam sebuah sistem informasi. Jadi, apabila komponen dari suatu sistem tidak dapat bekerja secara optimal, maka keseluruhan sistem yang diimplementasikan tidak akan dapat berjalan secara optimal.

1. Memiliki Batasan

Batasan yang dimaksudkan adalah batasan dari sebuah sistem informasi dengan sistem informasi lainnya, yang membuat sistem informasi tersebut menjadi satu kesatuan sistem informasi yang utuh, dan menunjukkan ruang lingkup yang dimiliki oleh sistem informasi tersebut.

1. Memiliki Lingkungan Luar dari Sistem

Lingkungan luar yang dimaksud merupakan keseluruhan sistem dan juga lingkungan yang berada di luar batasan dari sebuah sistem informasi.

1. Memiliki *Interface*

*Interface* atau antarmuka merupakan media yang digunakan untuk dapat menghubungkan sebuah komponen atau subsistem yang terdapat pada sebuah sistem informasi.

1. Memiliki *Input* system

Merupakan jenis energi yang digunakan untuk dimasukkan ke dalam suatu sistem.

1. Memiliki *Output* atau Keluaran dari Sistem

*Output* merupakan keluaran energi atau hasil yang diteruskan oleh *input.* Adapun hasil atau *output* yang dihasilkan bisa berupa tampilan data dan juga informasi yang muncul pada *display user*, yang berisi informasi.

1. Memiliki Pengolah dan Pemrosesan Sistem

Pemrosesan sistem dapat membantu proses pengolahan data secara keseluruhan yang terdapat di dalam sistem informasi, lalu mentransmisikan hasil dari pengolahan data tersebut menuju *output* yang dikeluarkan oleh sistem dan dapat diakses oleh *user*.

1. Memiliki Sasaran dari Sistem

Sasaran dari suatu sistem merupakan analisis berupa siapa saja yang akan menggunakan sistem informasi tersebut. Tanpa adanya sasaran dari pembuatan sistem, maka sistem informasi tersebut tidak akan bermanfaat dan juga berguna.

* + 1. **Perancangan Sistem Informasi**

Tahap perancangan sistem biasanya dilakukan setelah tahao analisis sistem selesai dilakukan. Dalam arti, setelah mendapatkan gambaran denga jelas mengenai rencana kerja yang akan dilakukan, maka langkah selanjutnya yakni memikirkan bagaimana membentuk sistem tersebut. Tahap dapat juga disebut dengan perancangan sistem.

Adapun tujuan utama dilakukannya perancangan sistem yaitu sebagai berikut.

1. Untuk memenuhi kebutuhan pemakai sistem
2. Untuk memberikan gambaran yang jelas mengenai rancang bangun kepada pihak *programmer* dan para ahli teknik lainnya yang terkait serta pihak *user* sendiri.

Desain sebuah sistem merupakan persiapan dari desain secara terperinci. Desain terperinci yang dimaksud berguna bagi pihak *programmer* serta ahli teknik lainnya yang nantinya akan mengimplementasikan sistem tersebut. Terdapat pula komponen dari sistem informasi yang perlu untuk dilakukan tahap desain yaitu meliputi model, *output, input, database,* teknologi, dan kontrol.

* + 1. ***Flowchart***

Dalam karyanya, penulis Indrajani (2015:36) menjelaskan mengenai pengertian *flowchart/* diagram alir yakni bahwa :

“*Flowchart* merupakan penggambaran secara grafis dari langkah-langkah dan urutan prosedur suatu program. Biasanya mempermudah penyelesaian masalah, khususnya yang perlu dipelajari dan dievaluasi lebih lanjut”.

Berikut ini akan dijelaskan mengenai urutan simbol-simbol dari suatu *flowchart* beserta dengan deskripsi dari masing-masing *flowchart*.

|  |  |
| --- | --- |
| **SIMBOL** | **DESKRIPSI** |
| ***Terminator*** | Merupakan suatu simbol yang digunakan untuk menunjukan awal dan akhir dari suatu proses program. |
| ***Process*** | Merupakan simbol yang digunakan untuk menyatakan kegiatan yang kaan ditampilkan dalam suatu *flowchart.* |
| ***Read / Write*** | Merupakan simbol yang digunakan untuk mewakili adanya proses *input* dan *output* pada suatu program. |
| ***Decision*** | Merupakan simbol yang digunakan sebagai proses/langkah dimana diperlukan adanya keputusan terhadap suatu kondisi tertentu. Adapun pada simbol ini, terdapat dua hasil *output* yang nantinya dilanjutkan pada aliran yang berbeda. |
| ***Preparation*** | Merupakan salah satu simbol yang digunakan pada alur *flowchart* guna mempersiapkan tempat penyimpanan yang akan digunakan sebagai tempat pengelohan didalam *storage*. |
| ***Subprogram*** | Merupakan simbol yang digunakan untuk menjalankan proses dari suatu bagian atau prosedur. |
| ***On-page Connector*** | Merupakan simbol yang digunakan sebagai penghubung dari suatu bagian alur *flowchart* yang berada pada halaman yang sama (satu halaman). |
| ***Off-page Connector*** | Merupakan simbol yang digunakan sebagai penghubung dari suatu alur *flowchart* yang berada pada halaman yang berbeda. |
| ***Arrow*** | Merupakan suatu simbol yang berguna untuk menunjukkan arah aliran suatu proses terhadap proses lainnya. |
| **Manual *Operation*** | Merupakan simbol yang berfungsi sebagai penanda bahwa suatu tindakan / proses dari suatu bagian alur *flowchart* tidak dilakukan oleh komputer atau dalam arti dilakukan secara manual. |

**Tabel 2.1 Simbol *Flowchart***

* + 1. ***UML Modeling***

Pengertian bahasa UML (*Unified Modeling Language*) menurut sudut pandang penulis Rosa A.S dan M. Shalahuddin (2014:137) dalam buku yang ditulisnya, keduanya mengungkapkan bahwa “UML merupakan bahasa *visual* untuk permodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung”.

Bahasa UML mencakup banyak diagram didalamnya, dengan tiga diagram utama yakni *Sequence Diagram, Class Diagram,* dan *Use Case Diagram.* Adapun, pendapat mengenai sebab munculnya bahasa UML menurut penulis Rosa A.S dan M. Shalahuddin (2014:137) yakni :

“UML muncul karena adanya kebutuhan permodelan visual guna kegiatan *menspesifikasikan*, menggambarkan, membangun, serta mendokumentasikan dari suatu sistem perangkat lunak”.

Adapun, berikut ini akan dijelaskan secara detail mengenai diagram-diagram yang tergabung ke dalam bahasa UML.

1. ***Use Case Diagram***

“Mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *Use Case Diagram* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi serta siapa saja yang berhak menggunakan fungsi-fungsi itu”. Rosa A.S dan M. Shalahuddin (2014:155).

Tak hanya pengertian, terdapat pula syarat penamaan pada *Use Case Diagram.* Dimana persyaratan penamaan yakni nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama untuk penamaan *Use Case* yaitu mendefinisikan apa yang disebut aktor dan *Use Case.*

1. Aktor

Merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, diluar dari sistem informasi yang akan dibuat sendiri. Jadi, walaupun simbol dari aktor adalah gambar orang, namun belum tentu suatu aktor merupakan orang.

1. *Use Case*

Merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut ini adalah simbol-simbol yang ada pada diagram *Use Case* :

|  |  |
| --- | --- |
| **SIMBOL** | **DESKRIPSI** |
| ***Use Case*** | Merupakan simbol deskripsi dari urutan aksi-aksi yang ditampilkan sistem. yang menghasilkan suatu hasil yang terukur bagi suatu aktor. |
| **Aktor**  **NAMA AKTOR** | Merupakan segala sesuatu yang perlu berinteraksi dengan sistem untuk pertukaran informasi. Biasanya dapat berupa manusia, sistem, atau *device.* |
| **Asosiasi** | Merupakan simbol yang digunakan untuk menghubungkan *actor* dengan *use case.* |
| **Ektensi**  **<<extend>>** | Merupakan simbol yang menunjukkan bahwa suatu *use case* berperan sebagai tambahan fungsional dari *use case* lainnya jika suatu kondisi dapat/telah terpenuhi. |
| **Generalisasi** | Hubungan dimana obyek anak (*descendant*) berbagi perilaku dan struktur data dari obyek yang ada di atasnya atau dapat disebut juga obyek induk (*ancestor*). |
| ***Include***    **<<include>>** | Relasi tambahan untuk ditujukan kepada *use case,* dimana *use case* yang ditambahkan membutuhkan  *use case* ini untuk menjalankan fungsinya atau dapat sebagai syarat untuk menjalankan  *use case* tersebut. |
| **Proses** | Simbol Proses menyatakan batasan suatu sistem didalam relasi dengan *actor-actor* yang mengguakannya (di luar sistem) dan fitur-fitur yang harus disediakan (dalam sistem). |

**Tabel 2.2 Simbol *Use Case* Diagram**

1. ***Scenario Use Case***

“*Scenario Use Case* merupakan skenario alur dari jalannya proses *use case* dari sisi aktor maupun sistem”. Rosa A.S dan M. Shalahuddin (2014:155). Adapun *Scenario Use Case* dibuat per *use case* terkecil. Berikut ini adalah contoh format tabel *Scenario Use Case* beserta dengan penjelasannya*.*

1. Skenario Normal merupakan sebuah skenario yang apabila suatu sistem dapat berjalan normal atau tanpa terjadi kesalahan (*error*).
2. Skenario Alternatif merupakan skenario yang apabila suatu sistem tidak dapat berjalan normal atau mengalami *error.*

|  |  |
| --- | --- |
| **REAKSI AKTOR** | **REAKSI SISTEM** |
| Skenario Normal |  |
|  |  |
| Skenario Alternatif |  |
|  |  |

**Tabel 2.3 Format Tabel *Scenario Use Case***

Selain itu, baik skenario normal maupun skenario alternatif dapat memiliki nilai lebih dari satu. Dan, nantinya alur dari *scenario* *use case* tersebut akan menjadi dasar pembuatan diagram *sequence.*

1. ***Activity Diagram***

“*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak*“.* Rosa A.S dan M. Shalahuddin (2014:161). Namun, jika diperhatikan lebih lanjut diagram aktivitas merupakan wujud penggambaran dari aktivitas suatu sistem atau aktivitas yang dapat dilakukan oleh sisem tersebut.

Adapun, diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut ini :

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokkan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan tampilan antarmuka.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada suatu perangkat lunak.

Berikut ini urutan simbol yang digunakan dalam pembuatan suatu Diagram *Activity* dan dilengkapi dengan penjelasan dari masing-masing simbol*.*

|  |  |
| --- | --- |
| **SIMBOL** | **DESKRIPSI** |
| **Status Awal** | Merupakan sebuah simbol yang berperan sebagai awal dari suatu aktivitas modul sistem aplikasi. |
| **Aktivitas** | Merupakan suatu simbol yang memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain. |
| **Percabangan / *Decision*** | Merupakan suatu simbol yang digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu. |
| **Penggabungan / *Join*** | Merupakan suatu simbol yang menjadi asosiasi penggabungan dimana lebih dari satu aktivitas yang digabungkan menjadi satu bagian. |
| **Status Akhir** | Merupakan sebuah simbol yang berperan sebagai akhir dari suatu aktivitas modul sistem aplikasi. |
| ***Swimlane***  **atau** | Merupakan sebuah simbol yang dapat menunjukkan siapa yang bertanggung jawab dalam melakukan aktivitas pada suatu diagram. |

**Tabel 2.4 Simbol *Activity Diagram***

1. ***Sequence Diagram***

“Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek”. Rosa A.S dan M. Shalahuddin (2014:165). Berdasarkan pengertian tersebut, maka untuk menggambarkan diagram sekuen diharuskan mengetahui objek-objek yang terlibat dalam sebuah *use case* beserta dengan metode-metode yang dimiliki setiap kelas. Selain itu, didalam membuat diagram *use case* diperlukan untuk melihat skenario yang ada pada *use case.*

Berikut ini merupakan penjelasan dari simbol yang terdapat pada *Sequence Diagram.*

|  |  |
| --- | --- |
| **SIMBOL** | **DESKRIPSI** |
| **Aktor**  actor  **NAMA AKTOR**  **Atau** | Merupakan simbol yang digunakan untuk menggambarkan *user* / pengguna. Biasanya dapat berupa orang, proses, atau sistem lain yang tentunya dapat berinteraksi dengan sistem informasi. Selain itu, penempatan pada simbol aktor ini, dapat ditempatkan pada bagian atas diagram. |
| **Garis hidup / *Lifeline*** | Merupakan suatu simbol yang menandakan kehidupan atas sebuah objek. Adapun untuk mengakhirinya ditandai dengan tanda X pada titik dimana kelas tidak lagi berinteraksi. |
| **Objek** | Merupakan simbol yang digunakan guna dapat berpartisipasi secara berurutan dengan langkah mengirim dan / atau menerima pesan. |
| **Waktu Aktif** | Merupakan simbol yang menandakan bahwa suatu objek dalam keadaan aktif dan berinteraksi dengan semua yang berhubungan dengan tahapan-tahapan yang terdapat didalamnya. |
| **Pesan Tipe *Create***    **<<*create*>>** | Merupakan simbol yang menyatakan bahwa satu objek dapat membuat objek yang lain. Dan untuk menggambarkannya, arah panah mengarah pada objek yang dibuat. |
| **Pesan Tipe *Call***    **1 : nama\_metode ()** | Merupakan suatu simbol yang dapat menyatakan bahwa suatu objek dapat memanggil operasi / metode yang ada pada objek lain atau terhadap dirinya sendiri. |
| **Pesan Tipe *Send***  **1 : masukan** | Merupakan suatu simbol yang menyatakan bahwa suatu objek dapat mengirimkan data / masukan / informasi kepada objek lainnya. |
| **Pesan Tipe *Return***    **1 : keluaran** | Merupakan suatu simbol yang menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode dapat menghasilkan kembali ke objek tertentu. |
| **Pesan Tipe *Destroy***    **<<*destroy*>>** | Merupakan suatu simbol yang dapat mengakhiri hidup objek lain. Untuk menggambarkan simbol tersebut, arah panah mengarah pada objek yang di akhiri. |

**Tabel 2.5 Simbol *Sequence Diagram***

1. ***Class Diagram***

“*Class Diagram* merupakan diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sebuah sistem”. Rosa A.S dan M. Shalahuddin (2014:141). Adapun, kelas-kelas yang ditampilkan pada *Class Diagram* dapat berupa implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metode. *Interface* tidak dapat langsung diinstansikan, tetapi harus diimplementasikan terlebih dahulu hingga menjadi sebuah *class*. Dengan demikian, *interface* dapatmendukung resolusi metode pada saat *run-time.*

Berikut ini merupakan detail penjelasan dari simbol-simbol *Class Diagram.*

|  |  |
| --- | --- |
| **SIMBOL** | **DESKRIPSI** |
| **Kelas / *Class*** | Merupakan simbol yang terdiri dari himpunan objek-objek yang berbagi atribut serta operasi yang sama. |
| **Antarmuka / *Interface***  **Nama *Interface*** | Sama dengan konsep *interface* dalam pemrograman berorientasi objek. Sebuah *interface* dapat menentukan operasi eksternal yang dapat dilakukan oleh sebuah *class* dan/ atau komponen. Dan biasanya suatu *interface* hanya dapat menentukan bagian tertentu dari suatu *class.* |
| **Asosiasi / *Association*** | Merupakan simbol relasi antar kelas dengan makna umum, asosiasi biasanya juga di sertai dengan *multiplicity.* Hal ini dapat terlihat pada saat beberapa kelas saling terhubung satu sama lain secara konseptual. |
| **Asosiasi berarah / *Directed Association*** | Merupakan sebuah simbol relasi seperti asosiasi antar kelas namun menggambarkan objek atau aliran kejadian yang berasal dari salah satu kelas, sedangkan kelas lainnya hanya bersifat pasif. |
| **Generalisasi** | Relasi antar kelas dengan makna generalisasi spesialisasi (urutan khusus). Adapun, suatu kelas (*child class*) dapat diturunkan dari kelas lain dan mewarisi semua atribut dan method induknya (*parent class*) serta dapat menambah method atau atribut baru. |
| **Kebergantungan / *Defendency*** | Merupakan hubungan ketergantungan antar kelas-kelas yang ada. Dalam arti, suatu kelas dapat memiliki ketergantungan terhadap kelas lain, namun tidak berlaku sebaliknya. |
| **Agregasi / *Aggregation*** | Merupakan simbol berbentuk khusus dari suatu asosiasi, dimana induk (*parent class*) terhubung dengan bagian-bagiannya. |

**Tabel 2.6 Simbol *Class Diagram***

* + 1. **Perancangan *Database***

Perancangan *Database* adalah proses pembuatan (*develop*) struktur *database* sesuai dengan data yang dibutuhkan oleh *user.* Kemudian, adapun tiga fase perancangan basis data menurut Indrajani (2011:53) yaitu :

1. ***Conceptual Database Design***

Merupakan suatu proses pembentukan model yang dibangun menggunakan informasi dalam spesifikasi kebutuhan *user* dan merupakan sumber informasi untuk fase desain logikal.

1. ***Logical Database Design***

Merupakan proses pembentukan model dengan memperbaiki dan memetakan model data konseptual yang telah dibuat sebelumnya.

1. ***Physical Database Design***

Merupakan proses yang menghasilkan deskripsi implementasi basis data pada penyimpanan sekunder, gambaran struktur penyimpanan, dan metode akses yang digunakan untuk mencapai akses yang efisien terhadap data. Desain fisikal juga bisa dikatakan sebagai cara pembuatan menuju suatu DBMS tertentu.

1. **Pengertian *Database***

Basis Data atau *Database* merupakan kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut.

Adapun tujuan utama dari basis data atau *database* menurut Fathansyah (2012:3) yakni :

“Basis data dan lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data/arsip”.

Dalam lingkungan basis data, pengguna memiliki alat untuk mengakses data. Semua pengguna melakukan semua tipe pekerjaan dan keperluan mereka seperti menggali data (*data mining*), memodifikasi data, atau berusaha membuat data baru. Data yang dimaksud biasanya bersifat *integrated* dan *shared.*

1. Terpadu (*Integrated*), berkas-berkas data yang ada pada basis data saling terkait.
2. Berbagi Data (*Shared*), data yang sama dapat dipakai oleh sejumlah pengguna dalam waktu yang bersamaan. Hal ini dapat disebut juga system *multiuser.*
3. **Normalisasi**

Menurut Indrajani (2011:57) dalam karyanya mengungkapkan mengenai pengertian dari normalisasi yakni:

“Normalisasi adalah teknik dengan pendekatan *bottom-up* yang digunakan untuk membantu mengidentifikasi hubungan. Dimulai dari menguji hubungan, yaitu *functional dependencies* antaraatribut”.

Selain pengertian, adapun tujuan yang dimiliki proses pembuatan suatu normalisasi yakni beberapa diantaranya seperti untuk meminimalisir adanya kerangkapan (*redudancy*) data, untuk menghilangkan kompleksitas, untuk mempermudah dilakukannya pemodifikasian data-data terkait, serta yang utama adalah dapat mengidentifikasi kesesuaian hubungan yang mendukung guna memenuhi kebutuhan suatu perusahaan/organisasi.

Setelah mengetahui pengertian serta tujuan dari pembuatan normalisasi, terdapat pula level/ tingkatan dari proses suatu normalisasi. Dimana level/tingkatan tersebut meliputi :

1. ***Unnormalized Form* (UNF)**

Merupakan suatu level normalisasi yang memiliki tabel yang berisikan satu atau lebih grup berulang. Langkah untuk membuat *Unnormalized Form* yaitu dengan memindahkan seluruh data dari sumber informasi.

1. ***First Norm Form* (1NF)**

Merupakan suatu relasi dimana setiap baris dan kolom hanya memiliki nilai tunggal serta tidak ada pengulangan pada grup atribut dalam suatu baris. Adapun tahapan proses perpindahan dari UNF kepada 1NF yakni sebagai berikut.

1. Memisahkan serta menghilangkan duplikasi kolom pada setiap tabel yang sama.
2. Membuat beberapa tabel menjadi terpisah pada masing-masing kelompok data yang terkait.
3. Mengidentifikasi setiap baris dan kolom yang berulang guna menentukan *primary key.*
4. ***Second Norm Form* (2NF)**

Merupakan level normalisasi yang memiliki suatu relasi didalam level 1NF dan setiap atribut yang bernilai *non primary key* bersifat *fully fungsionally dependent* terhadap *primary key.*

1. ***Third Norm Form* (3NF)**

Merupakan level normalisasi yang memiliki sebuah relasi terhadap level 1NF serta 2NF, dimana tidak terdapat atribut *non primary key* yang bersifat *transitively dependent* terhadap *primary key.*

1. ***Entity Relational Diagram* (ERD)**

Menurut Indrajani (2011:109) mengenai *Entity Relational Diagram* (ERD)mengungkapkan bahwa :

“ERD adalah komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan seluruh fakta dari dunia nyata yang kita tinjau.”

Selain atribut, *Entity Relational Diagram* (ERD) juga memiliki kardinalitas relasi yang terdiri dari 3 jenis kardinalitas. Adapun pengertian dari Kardinalitas Relasi sendiri adalah penggambaran mengenai banyaknya jumlah maksimum entitas yang dapat melakukan relasi dengan suatu entitas terhadap himpunan entitas yang lainnya. Berikut akan dijelaskan mengenai jenis-jenis dari Kardinalitas Relasi yakni sebagai berikut.

1. ***One to One***

Merupakan tingkat relasi dari satu ke satu. Dimana hal ini dapat dinyatakan dengan hanya mempunyai satu hubungan kejadian pada entitas yang kedua dan berlaku pula untuk sebaliknya.

1. ***One to Many***

Merupakan tingkat relasi dari satu ke banyak. Dimana hal ini dapat dinyatakan apabila satu kejadian pada entitas yang pertama dapat memiliki banyak hubungan dengan kejadian pada entitas yang kedua.

1. ***Many to One***

Merupakan tingkat relasi dari banyak ke satu. Hal tersebut dapat dinyatakan apabila satu kejadian pada entitas kedua memiliki banyak hubungan relasi dengan kejadian pada entitas pertama.

1. ***Many to Many***

Merupakan tingkat relasi dari banyak ke banyak. Hal ini dapat terjadi apabila suatu kejadian pada sebuah entitas mempunyai banyak hubungan dengan kejadian pada entitas yang lainnya, baik dari sisi entitas pertama maupun entitas kedua.

* + 1. **Perancangan *User Interface***

Merupakansuatu kegiatan proses penggambaran yang berbanding lurus pada sebuah bagian sistem dengan langkah membuat gambaran antarmuka yang diinginkan pihak pengguna untuk menerima informasi dari pengguna/*user* dan memberikan informasi kepada pengguna/*user* untuk membantu mengarahkan alur penelusuran masalah sampai ditemukan suatu solusi.

Tujuan dari perancangan antarmuka pengguna adalah merancang *interface* yang efektif untuk suatu sistem perangkat lunak. Efektif yang dimaksud artinya siap digunakan dan hasilnya sesuai dengan kebutuhan pengguna. Pengguna sering menilai sistem dari *interface* bukan dengan fungsinya melainkan dari segi *user interface*. Saat ini salah satu *interface* yang banyak digunakan dalam *software* adalah GUI (*Graphical User Interface*).

* 1. **Koperasi**
     1. **Pengertian Koperasi**

Menurut Hendrojogi (2015:46) pada salah satu karya sastranya mengungkapkan mengenai pengertian koperasi yakni bahwa :

“Koperasi adalah perkumpulan otonom dari orang-orang yang bergabung secara sukarela untuk memenuhi kebutuhan dan aspirasi ekonomi, sosial, dan budaya mereka yang sama melalui perusahaan yang dimiliki dan diawasi secara demokratis”.

Adapun nilai-nilai yang terkandung dalam suatu organisasi koperasi yakni melandaskan nilai-nilai menolong diri sendiri, bertanggung jawab kepada diri sendiri, demokrasi, persamaan, keadilan, dan solidaritas.

* + 1. **Prinsip-prinsip Koperasi**

Adapun prinsip-prinsip koperasi yakni sebagai berikut.

1. Keanggotaan yang sukarela dan terbuka

Koperasi adalah organisasi yang bersifat sukarela, terbuka bagi semua orang yang bersedia menggunakan jasa-jasanya dan bersedia menerima tanggung jawab keanggotaan, tanpa membedakan jenis kelamin, latar belakang sosial, ras, politik, atau agama.

1. Pengawasan demokratis oleh anggota

Koperasi adalah organisasi demokratis yang diawali oleh para anggotanya, yang secara aktif menetapkan suatu kebijakan serta membuat keputusan. Dalam hal ini, para anggota koperasi memiliki hak suara yang sama (satu anggota berhak memiliki satu suara).

1. Partisipasi anggota dalam kegiatan ekonomi

Dalam prinsip ini, para anggota koperasi memberikan kontribusi permodalan secara adil dan melakukan pengawasan secara demokratis terhadap modal tersebut. Adapun pada modal tersebut, setidaknya sebagiannya adalah milik anggota guna nantinya akan diberikan kembali kepada pihak anggota secara seimbang dengan transaksi yang telah dilakukan anggota dengan koperasi ataupun dapat juga digunakan untuk memenuhi kebutuhan koperasi dengan tujuan mengembangkan koperasi menjadi lebih maju dan berkembang.

1. Otonomi dan kemandirian (*Independence*)

Koperasi adalah organisasi otonom, menolong diri sendiri namun diawasi oleh para anggotanya. Sebagai contoh, pada saat suatu koperasi dapat mengadakan suatu perjanjian dengan pihak luar/ organisasi lainnya atau melakukan penanaman modal, maka diharuskan tetap dalam pengawasan oleh para anggota lainnya dan berdasarkan otonomi dari koperasi tersebut.

1. Pendidikan, pelatihan, dan penerangan

Organisasi koperasi memberikan pendidikan serta pelatihan bagi pihak-pihak yang tergabung dalam koperasi tersebut guna dapat melakukan tugas-tugasnya dengan efektif dan profesional serta dapat terus meningkatkan perkembangan suatu koperasi tersebut. Pendidikan yang dimaksudkan adalah penjelasan mengenai hakikat dunia perkoperasian serta keuntungan-keuntungan yang didapat dalam menjalankan suatu koperasi.

1. Kerja sama antar koperasi

Dalam koperasi, melayani para anggotanya secara kolektif dianggap dapat memperkuat suatu organisasi koperasi. Hal ini dapat terwujud dengan dilakukannya kerja sama antar internal organisasi tersebut maupun eksternal organisasi yang tentunya masih tergabung/terkait dalam organisasi tersebut.

1. Kepeduliaan terhadap masyarakat

Pihak koperasi diajarkan serta diharuskan pula untuk tetap berbagi kepada sesama. Hal ini biasanya dilakukan oleh pihak koperasi secara berkelanjutan dan berdasarkan dengan kebijakan-kebijakan yang telah ditetapkan pada RAT (Rapat Anggota Tahunan).

* + 1. **Kelengkapan Koperasi**

Berikut ini merupakan kelengkapan-kelengkapan koperasi yakni sebagai berikut :

1. Anggota

Adapun anggota koperasi dapat meliputi perorangan yang secara sukarela menjadi bagian dari anggota koperasi. Dan adapula badan hukum koperasi, dimana suatu koperasi yang menjadi anggota dari koperasi yang ruang lingkupnya lebih luas.

1. Pengurus koperasi

Pengurus koperasi dan pengawas koperasi dipilih oleh anggota dalam RAT (Rapat Anggota Tahunan). Adapun dalam RAT tersebut, diungkapkan pula mengenai wewenang dan tugas-tugas dari seorang pengurus koperasi.

1. RAT (Rapat Anggota Tahunan)

Merupakan kekuasaan tertinggi dari suatu organisasi koperasi. Tujuan dari adanya RAT adalah :

1. Meminta pertanggung jawaban pengurus serta pengawas koperasi dalam hal pengelolaan koperasi.
2. Menetapkan anggaran dasar koperasi
3. Mengesahkan rencana kerja
4. Menetapkan pembagian SHU
5. Sebagai sarana untuk memilih dan memberhentikan selaku pengurus dan pengawas koperasi.
   * 1. **Jenis Koperasi**

Terdapat berbagai macam/tipe koperasi yang terlihat dari jenisnya, keanggotaannya, tingkatan, maupun fungsinya. Berikut ini akan dijelaskan salah satu macam koperasi berdasarkan jenisnya yakni sebagai berikut.

1. Koperasi Konsumsi

Merupakan organisasi koperasi yang memiliki ruang lingkup melakukan usaha produksi atau menghasilkan barang.

1. Koperasi Simpan Pinjam

Merupakan organisasi koperasi yang bertugas melayani para anggotanya untuk menabung atau melakukan peminjaman uang.

1. Koperasi Serba Usaha

Merupakan organisasi koperasi yang terdiri dari berbagai jenis usaha di dalamnya.

* + 1. **Koperasi Simpan Pinjam**

Koperasi simpan pinjam dikelola dengan cara yang sama dengan koperasi pada umumnya, hanya ada beberapa bagian teknis yang berbeda. Adapun ruang lingkup kegiatan usaha koperasi pinjam adalah penghimpunan dan penyaluran dana dalam bentuk penyaluran pinjaman bagi para anggotanya. Adapun bentuk penghimpunan dana koperasi dapat berupa tabungan dan simpanan sedangkan penghimpunan dana yang berasal dari masyarakat sendiri yakni berupa pinjaman modal usaha.

Membahas tentang simpanan, suatu organisasi koperasi memiliki beberapa jenis simpanan yakni meliputi :

1. Simpanan Pokok

Merupakan suatu jenis simpanan dimana sejumlah uang yang nominalnya bersifat tetap dan wajib dibayarkan kepada pihak koperasi pada saat seseorang menjadi bagian anggota koperasi.

1. Simpanan Wajib

Merupakan jenis simpanan yang dibayarkan setiap bulannya kepada pihak koperasi, dimana besaran dari simpanan wajib telah ditetapkan atau disepakati oleh seluruh anggota koperasi melalui RAT. Adapun tata cara penarikan simpanan ini, pada saat anggota tersebut mengundurkan diri/ keluar sebagai anggota koperasi.

1. Simpanan Sukarela

Merupakan jenis simpanan koperasi yang besaran nominalnya dibebaskan sekehendak para anggotanya. Untuk tahap penarikan simpanan ini, pada saat anggota tersebut benar-benar membutuhkannya guna keperluan investasi atau keperluan lainnya yang tentunya berdasarkan persetujuan dari anggota dan pengurus koperasi.

1. Tabungan Koperasi

Merupakan jenis simpanan pada koperasi yang penyetorannya dilakukan berangsur-angsur. Untuk tahap penarikan dari simpanan, hanya dapat dilakukan oleh anggota yang bersangkutan.

1. Simpanan Berjangka Koperasi

Merupakan jenis simpanan koperasi yang tahap penyetorannya dilakukan satu kali untuk jangka waktu tertentu sesuai dengan perjanjian antara pihak anggota / penyimpan dengan pihak koperasi. Selain itu, penarikan simpanan tidak dapat diambil sebelum jangka waktu tersebut berakhir.

* 1. **Bahasa Pemrograman**

Menurut David Naista (2016:2) dalam bukunya mengungkapkan mengenai pengertian bahasa pemrograman bahwa “Bahasa Pemrograman adalah bahasa yang menghubungkan antara kita dengan komputer.”

Berikut ini merupakan rincian dari beberapa bahasa pemrograman yang digunakan oleh penulis guna melaksanakan perancangan sistem informasi Koperasi Simpan Pinjam, yakni meliputi :

* + 1. ***CodeIgniter***

*Codeigniter* merupakan salah satu dari sekian banyak *framework* PHP yang kini mulai banyak digunakan dalam mengembangkan aplikasi berbasis *web*. Sebelumnya, Awan Pribadi Basuki (2014:13) mengungkapkan mengenai pengertian *framework* yang berarti bahwa :

“*Framework* dapat diartikan sebagai kumpulan potongan-potongan program (kelas dan fungsi) yang disusun dan di organisasikan sedemikian rupa, sehingga dapat digunakan kembali untuk membantu membuat aplikasi utuh tanpa harus membuat semua kodenya dari awal”.

Adapun alasan-alasan mengapa penulis atau seorang *developer* menggunakan *framework* *Codeigniter* dalam pembuatan sebuah website yakni sebagai berikut.

1. Lingkup komunitas yang besar
2. Sisi dokumentasi codeigniter yang lengkap dan jelas dimengerti
3. Bersifat fleksibel dan tidak mengharuskan pengguna untuk mematuhi aturan untuk penggunaannya.
4. Tersedianya *Library* dan *Helper* yang cukup lengkap.
5. Minimalnya konfigurasi
6. Tidak mengharuskan menggunakan *command line*
7. Adanya konsep MVC (*Model-View-Controller*)
   * 1. **PHP (*Hypertext Prepocessor*)**

David Naista (2016:4) mengungkapkan mengenai pengertian dari bahasa pemrograman PHP yakni bahwa “*Hypertext Prepocessor* atau PHP adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML”. Dengan bahasa pemrograman PHP, pengguna/*developer* dimudahkan untuk membuat *website* menjadi lebih dinamis. Dan adapun, manfaat dari bahasa pemrograman PHP yaitu dapat membangun sebuah CMS (*Content Management System*).

* + 1. **HTML (*HyperText Markup Language*)**

HTML merupakan bahasa pemrograman yang hanya menjadi tampilan dasar atau tampilan depan dari sebuah *website.* Terdapat pula sudut pandang penulis lainnya yakni David Naista (2016:2) yang beranggapan mengenai pengertian bahasa pemrograman HTML bahwa :

“HTML merupakan bahasa *markup* yang digunakan untuk membuat sebuah halaman *website*, menampilkan berbagai informasi di dalam sebuah penjelajah *web internet* dan pemformatan *hiperteks* sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi”.

* + 1. **CSS (*Cascading Style Sheet*)**

Mengenai bahasa pemrograman CSS, adapun dari sudut pandang penulis David Naista (2016:3) mengungkapkan mengenai pengertian bahasa pemrograman CSS dalam salah satu karyanya yakni bahwa “CSS (*Cascading Style Sheet*) merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur dan seragam”.

Maka dari itu, dengan adanya bahasa pemrograman CSS (*Cascading Style Sheet*), pengguna dapat menambahkan *style* pada *script* HTML, dan sehingga tampilan sebuah *website* atau suatu program dapat terlihat menjadi lebih menarik.

* + 1. ***JavaScript***

*JavaScript* merupakan bahasa skrip *client side* yang populer di *Internet* dan dapat bekerja di sebagian besar *website* browser populer seperti *Internet Explorer* (IE), *Mozilla Firefox*, *Netscape*, dan *Opera*. Adapun penulis David Naista (2016:4) mengungkapkan mengenai bahasa pemrograman *JavaScript* yaitu:

“Kode *JavaScript* dapat disisipkan dalam halaman *website* menggunakan *tag <script>*. Dengan adanya *Javascript* ini, pengguna dapat membuat tampilan sebuah *website* menjadi lebih menarik dikarenakan dilengkapi dengan animasi-animasi yang dibuat interaktif”.

* + 1. ***JQuery***

Menurut Adi Chandra Setiawan (2015:48) dalam salah satu bukunya mengungkapkan bahwa “*JQuery* adalah *library javascript* untuk memudahkan pengguna dalam pembuatan *website* dengan HTML yang berjalan di sisi *client* ”. *Script JQuery* dibuat untuk memudahkan pengaturan dokumen seperti menyeleksi objek dengan elemen DOM (*Document Object Model*) serta dalam membuat aplikasi dengan AJAX, menerapkan animasi, dan mengaplikasikan *events.*

Selain itu, dengan adanya *script JQuery,* pihak *developer* dapat menciptakan berbagai *plugin* berbasis *library JavaScript.* Dan dengan *plugin-plugin* tersebut, pihak *developer* mampu menyusun sejumlah abstraksi guna kebutuhan interaksi dan animasi sederhana, serta beberapa efek yang cukup kompleks dan berbagai *widget* yang tentunya dapat dikonfigurasikan.

* + 1. **Perangkat Lunak (*Software*)**

Perangkat lunak (*software*) merupakan sekumpulan data elektronik yang disimpan dan diatur oleh komputer. Adapun data elektronik yang disimpan dapat berupa program atau instruksi yang akan menjalankan suatu perintah. Dan maka dari itu, melalui adanya *software* atau perangkat lunak, komputer dapat menjalankan suatu perintah.

Selain itu, fungsi dari perangkat lunak (*software*) sendiri adalah memproses data atau perintah hingga mendapat hasil atas menjalankan perintah tersebut, serta dapat berfungsi sebagai sarana interaksi yang menghubungkan antara pihak pengguna komputer dengan perangkat keras (*hardware*).

Selain bahasa pemrograman, adapun beberapa perangkat lunak yang digunakan penulis guna merancang program aplikasi yakni sebagai berikut.

1. ***XAMPP***

Merupakan perangkat lunak yang bersifat bebas, dimana mendukung untuk banyak sistem operasi dan merupakan kompilasi dari beberapa program. *XAMPP* merupakan program yang tersedia dalam GNU (*General Public License).* Atau dalama arti, mudah digunakan untuk meenampilkan halaman *web* yang dinamis. Selain itu, *XAMPP* terdiri dari beberapa program didalamnya, yakni meliputi :

1. *Apache HTTP Server,*
2. *MySQL Database,*
3. Penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*.
4. **MySQL**

Merupakan sebuah perangkat lunak sistem manajemen basis data *SQL* yang sifatnya *multithread* dan *multiuser. MySQL* memiliki struktur tabel yang lebih fleksibel dalam menangani *alter table* dibandingkan dengan *database* lainnya. Adapun beberapa kelebihan yang dimiliki *MySQL*, antara lain :

1. Dapat berjalan stabil pada berbagai sistem operasi seperti *Windows, Linux*, dan lain-lain.
2. Dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
3. Memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana.
4. Memiliki tipe kolom yang sangat kompleks, seperti *integer, double, char, date, text*, dan lain-lain.
5. Memiliki operator dan fungsi secara penuh yang mendukung perintah *select* dan *where* dalam *query.*
6. ***Notepad*++**

Merupakan sebuah program aplikasi pengembang yang berguna sebagai media editor teks dari berbagai macam *script* dan kode pemrograman. *Notepad++* memiliki beberapa kelebihan pada peningkatan kemampuan dalam sebuah program *text editor,* lebih dari sekedar program *notepad* bawaan dari *windows. Notepad++* dapat mengenal berbagai *tag* dan kode dalam berbagai bahasa pemrograman.

**BAB III**

**PROFIL PERUSAHAAN**

* 1. **Sejarah Singkat Koperasi Karyawan DKB Kantor Pusat (Kapus)**

Koperasi Karyawan DKB Kantor Pusat (Kapus) didirikan berdasarkan Akta Pendirian No. 2990/BH/I tanggal 28 Agustus 1992 serta berdasarkan Surat Keputusan Kepala Kantor Wilayah (Kakanwil) Departemen Koperasi DKI Jaya No. 146/BPL/X/VIII/1992. Adapun, koperasi Karyawan DKB Kapus merupakan koperasi yang bersifat fungsional/ terbatas. Terbatas yang dimaksud yakni mengacu kepada segi keanggotaannya. Dikarenakan didirikan di lingkungan Kantor Pusat PT Dok & Perkapalan Kodja Bahari (Persero), maka keanggotaan dari Koperasi Karyawan DKB Kapus hanya diperuntukkan bagi karyawan yang melakukan tugas-tugasnya di lingkungan Kantor Pusat PT. Dok & Perkapalan Kodja Bahari. Namun, pada kenyataannya, anggota dari Koperasi Karyawan DKB Kantor Pusat (Kapus) telah tersebar di PT DKB Group seperti Galangan, Cabang, serta Anak Perusahaan. Hal ini dikarenakan adanya mutasi yang didapatkan anggota tersebut sebagai karyawan PT. Dok & Perkapalan Kodja Bahari (Persero). Dapat dilihat bahwa pada tahun 2013, jumlah anggota dari Koperasi Karyawan DKB Kantor Pusat (Kapus) tercatat 343 anggota.

Untuk dari segi kepengurusan, Koperasi Karyawan DKB Kantor Pusat (Kapus) terdiri dari 3 unsur yaitu Unsur Pengurus, Unsur Pengawas, serta Unsur Penasehat. Hal tersebut diterapkan berdasarkan keputusan RAT Koperasi Tahun 2014 serta pengukuhan yang telah dilakukan oleh pihak Sudin Koperasi, Usaha Mikro, Kecil, dan Menengah dan Perdagangan Jakarta Administrasi Utara No. 961/-1.829 tanggal 1 Oktober 2014.

* 1. **Visi dan Misi Koperasi Karyawan DKB Kantor Pusat (Kapus)**
     1. **Visi**

Mewujudkan pelayanan yang optimal guna meningkatkan kesejahteraan para anggota.

* + 1. **Misi**

1. Mengoptimalkan sumber daya yang ada guna meningkatkan pelayanan dan usaha koperasi.
2. Meningkatkan mutu manajemen dan tata kelola yang transparan, profesional, serta akuntabilitas.
3. Meningkatkan jiwa partisipasi aktif bagi para anggota, baik sebagai kepemilikan atas koperasi maupun sebagai pengguna jasa dari koperasi.
   1. **Aspek Kegiatan Usaha Koperasi Karyawan DKB Kantor Pusat (Kapus)**

Berikut ini merupakan uraian bidang usaha pokok yang telah dijalankan pihak Koperasi Karyawan DKB Kantor Pusat (Kapus) yakni :

1. Perdagangan kebutuhan anggota / karyawan maupun perusahaan PT. Dok & Perkapalan Kodja Bahari (Persero) yang meliputi :
2. Pengadaan kebutuhan rumah tangga
3. Pengadaan alat tulis kantor (ATK)
4. Pengadaan barang teknik
5. Pengadaan kebutuhan kapal
6. Usaha Pinjaman untuk anggota

Selain menjalankan usaha di bidang pengadaan kebutuhan anggota serta perusahaan, Koperasi Karyawan DKB Kantor Pusat (Kapus) juga menjalankan usaha jasa di bidang pinjaman. Dimana, pihak koperasi memberikan pinjaman bagi anggota koperasi yang membutuhkan.

1. Menjalankan usaha di bidang jasa, seperti diantaranya :
2. Jasa penyedia Tenaga Kerja
3. Jasa penyedia Cleaning Service
4. Jasa penyedia Pemeliharaan Taman
5. Jasa Penyewaan Mobil Dinas
6. Jasa Borongan
7. Jasa Sewa Foto Copy, dsb.
   1. **Struktur Organisasi Koperasi Karyawan DKB Kantor Pusat (Kapus)**

RAPAT ANGGOTA TAHUNAN

**PENASEHAT**

**BADAN PENGAWAS**

**PENGURUS**

**KEPALA DIVISI LOGISTIK & UMUM**

**KETUA**

**KETUA**

**KEPALA DIVISI KEUANGAN**

**WAKIL KETUA**

**SEKRETARIS**

**ANGGOTA**

**SEKRETARIS**

**I**

**SEKRETARIS**

**II**

**BENDAHARA**

**MANAGER**

**BID. ADMIN**

**BID. P. KARY**

**BID. USAHA**

**Gambar 3.1 Struktur Organisasi**

**Koperasi Karyawan DKB Kantor Pusat (Kapus)**

* 1. **Deskripsi Kerja**

Adapun berikut ini pula akan dijelaskan mengenai deskripsi kerja dari masing-masing bagian yang terkait dengan organisasi Koperasi Karyawan DKB Kantor Pusat (Kapus) yakni meliputi :

1. **RAT (Rapat Anggota Tahunan)**

Merupakan pemegang kekuasaan tertinggi dalam kehidupan suatu koperasi. Bagi para anggotanya, forum RAT dijadikan sebagai lahan untuk mengkritik dan menyampaikan gagasan-gagasan yang sifatnya membangun guna pelayanan serta usaha koperasi dapat tetap berjalan dengan baik. Maka dari itu, dalam setiap pelaksanaan RAT (Rapat Anggota Tahunan) ini, setiap anggota diminta untuk menyampaikan suaranya. Kemudian, dari sisi mengambil keputusan dalam rapat, keputusan yang diambil sedapat mungkin berdasarkan keputusan bersama. Namun, apabila tidak mencapai kata mufakat, maka keputusan dapat diambil berdasarkan suara terbanyak dari anggota-anggota yang hadir dalam rapat. Adapun topik pembahasan serta keputusan yang biasanya dibahas dan diambil dalam rapat adalah :

1. Anggaran Dasar.
2. Pemilihan pengurus dan pengawas, pengangkatan jabatan, hingga pemberhentian pengurus.
3. Pembagian SHU (Sisa Hasil Usaha).
4. Rencana kerja, Rencana Anggaran Pendapatan, Belanja Koperasi, dan kegiatan pengesahan Laporan Keuangan.
5. Penggabungan, peleburan, pembagian, danpembubaran koperasi.
6. **Ketua Koperasi**

Tugas serta tanggung jawab yang wajib dilakukan oleh seorang Ketua Koperasi yakni :

1. Membuat dan mengesahkan seluruh keputusan dan kebijakan organisasi yang bersifat strategis (politis).
2. Memimpin rapat pengurus pleno dan rapat pengurus harian.
3. Mengkoordinasikan serta mengawasi program kerja dari seluruh penyelenggara organisasi.
4. Bertanggung jawab secara internal terhadap forum RPP pada akhir masa baktinya.
5. Memberikan pokok pikiran yang merupakan sebagai strategi dan kebijakan koperasi dalam rangka melaksanakan program kerja demi mencapai cita-cita dan tujuan organisasi.
6. Mengoptimalkan fungsi dan peran sebagai Wakil Ketua agar tercapai efesiensi dan efektivitas kerja organisasi.
7. **Wakil Ketua**

Tugas serta tanggung jawab yang wajib dilakukan oleh seorang Wakil Koperasi yakni :

1. Membuat dan mengesahkan keputusan serta kebijakan, baik di bidang pengembangan sumber daya manusia, hingga pengembangan usaha yang dijalankan pihak koperasi.
2. Mengusulkan dan merumuskan program kerja setiap tahunnya.
3. Bertindak sebagai kepala personalia, dimana memiliki kewenangan mengatur dan mengkoordinasikan sistem kepersonaliaan organisasi.
4. Menyusun konsep rencana anggaran pendapatan dan belanja organisasi setiap tahunnya dengan menggunakan pertimbangan serta indikator teknis yang ada.
5. Membangun hubungan kerja sama kemitraan dengan pihak lain guna mengembangkan ekonomi skala kecil, menengah, dan koperasi.
6. **Sekretaris I**

Tugas serta tanggung jawab yang wajib dilakukan oleh seorang Sekretaris I yakni :

1. Membuat dan mengesahkan keputusan dan kebijakan organisasi bersama dengan Kepala Koperasi dalam bidang administrasi dan penyelenggaraan roda organisasi.
2. Mengkoordinasikan seluruh penyelenggara roda organisasi di bidang administrasi dan tata kerja organisasi.
3. Menangani surat masuk dan keluar bersama dengan Kepala Koperasi.
4. Bertanggung jawab pada setiap aktifitas di bagian administrasi dan tata kerja organisasi.
5. Mengusulkan kepada Kepala Koperasi atas segala peraturan organisasi di bidang administrasi dan tata kerja organisasi.
6. Membuat ringkasan dalam setiap rapat organisasi baik Rapat Pengurus Pleno maupun Rapat Pengurus Harian.
7. Mengawasi seluruh penyelenggaraan aktifitas organisasi di bagian administrasi dan tata kerja.
8. **Sekretaris II**

Tugas serta tanggung jawab yang wajib dilakukan oleh seorang Sekretaris II yakni :

1. Mengesahkan segala keputusan dan kebijakan organisasi terutama dalam hal kesekretariatan, bersama dengan Sekretaris I.
2. Mengatasi segala tugas dari Sekretariat I apabila yang bersangkutan berhalangan hadir dalam sebuah pertemuan dalam bidang kesekretariatan dan tata usaha organisasi.
3. Memberikan fasilitas kebutuhan ketatausahaan internal organisasi dan antar unit kerja.
4. Mengawasi seluruh penyelenggaraan aktifitas organisasi dibagian logistik, akomodasi, dan urusan perjalanan organisasi.
5. Menyelenggarakan aktifitas Pengembangan Usaha Kesejahteraan Sosial dalam berbagai bentuk seperti santunan.
6. **Bendahara**

Tugas serta tanggung jawab yang wajib dilakukan oleh seorang Bendahara yakni :

1. Membuat dan mengesahkan keputusan serta kebijakan organisasi, bersama dengan Kepala Koperasi dalam bidang kekayaan dan keuangan organisasi.
2. Memberikan fasilitas kebutuhan pembiayaan program kerja dan roda organisasi.
3. Menyelenggarakan aktifitas pembukuan terhadap transaksi pengeluaran dan pemasukan keuangan organisasi secara rutin.
4. Menyusun neraca laba rugi keuangan organisasi setiap 1 bulan sekali.
5. Menyelenggarakan aktifitas pembukuan terhadap transaksi pengeluaran dan pemasukan keuangan secara rutin.
6. Menyimpan keuangan organisasi di lembaga keuangan yang disepakati forum RPP dan menyisihkannya untuk kepentingan operasional.
7. Mendata dan menginventarisir asset dan harta kekayaan organisasi untuk kepentingan perhitungan modal dan laba/rugi organisasi.
8. **Badan Pengawas**

Tugas serta tanggung jawab yang wajib dilakukan oleh seorang Badan Pengawas yakni :

1. Mengesahkan seleuruh keputusan dan kebijakan organisasi dalam sistem pengawasan.
2. Melakukan pengawasan terhadap tata kehidupan koperasi, termasuk organisasi, manajemen, usaha, keuangan, permodalan, dsb.
3. Mengawasi kebijakan operasional pengurus yang meliputi bidang organisasi, bidang usaha dan bidang keuangan koperasi.
4. Memeriksa, meneliti ketepatan dan kebenaran catatan organisasi, usaha, dan keuangan untuk dibandingkan dengan kenyataan yang ada.
5. Membuat laporan pemeriksanaan secara tertulis, memberikan pendapatnya, dan memberikan saran perbaikan dalam menyajikan laporan kepada rapat anggota tahunan.
6. Bertanggung jawab atas kegiatan pemeriksaan dan hasil pemeriksaan tersebut serta merahasiakannya kepada pihak ketiga.
7. **Manajer**

Tugas serta tanggung jawab yang wajib dilakukan oleh seorang Manajer yakni :

1. Memimpin dan mengkoordinasikan semua kegiatan usaha koperasi.
2. Memberikan pengarahan dan mengawasi pelaksanaan rencana kerja yang telah digariskan.
3. Mengesahkan pengeluaran kas sampai batas wewenang yang diberikan oleh Pengurus.
4. Mengambil langkah pengamanan uang dan barang koperasi.
5. Membahas serta menyiapkan rencana kerja dan anggaran serta menjajaki kemungkinan perluasan usaha baru untuk diajukan kepada rapat anggota tahunan.
6. **Bidang Administrasi**

Tugas serta tanggung jawab yang wajib dilakukan oleh seorang Bidang Administrasi yakni :

1. Melaksanakan segala sesuatu yang berhubungan dengan urusan rumah tangga koperasi.
2. Melaksanakan segala sesuatu yang berhubungan dengan surat-menyurat.
3. Melaksanakan segala sesuatu yang berhubungan dengan urusan karyawan.
4. Bertanggung jawab atas pelaksanaan tugasnya kepada Manajer.
5. **Bidang Usaha**

Tugas serta tanggung jawab yang wajib dilakukan oleh seorang Bidang Usaha yakni :

1. Melaksanakan segala sesuatu yang berhubungan dengan urusan usaha.
2. Membuat daftar/list jadwal kegiatan usaha.
3. Menyiapkan administrasi penagihan.
4. Mencari order kedalam dan atau keluar perusahaan.
5. Bertanggung jawab atas pelaksanaan tugasnya kepada Manajer.
6. **Bidang Pengawasan Karyawan**

Tugas serta tanggung jawab yang wajib dilakukan oleh seorang Bidang Pengawasan Karyawan yakni :

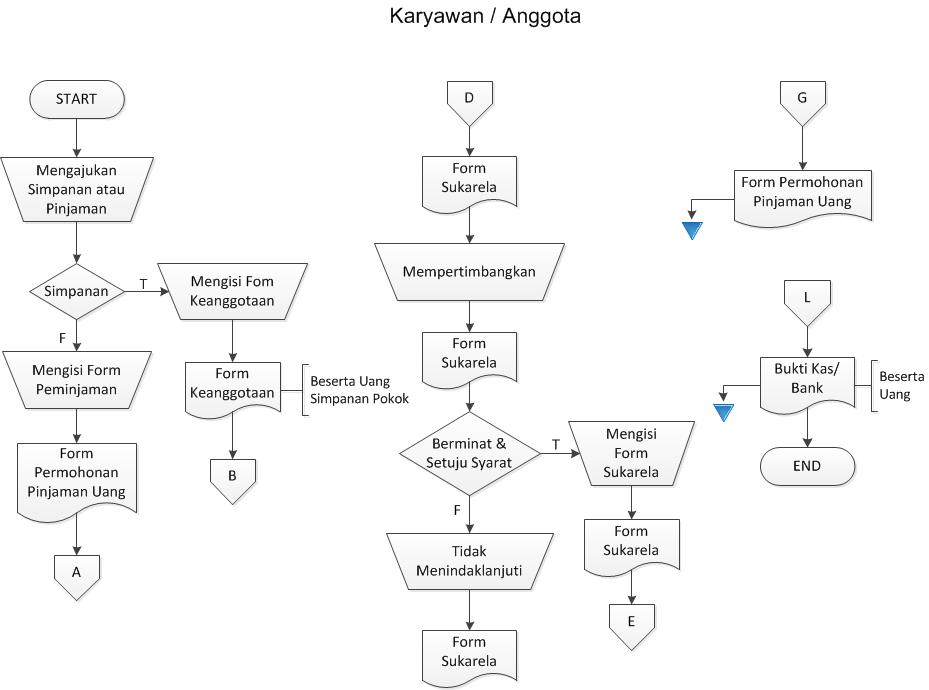
1. Mengkoordinasikan seluruh karyawan yang berada dibawah naungannya.
2. Mengatur penempatan karyawan sesuai dengan bidang tugasnya.
3. Mengatur mutasi/rotasi antar bagian.

**BAB IV**

**PEMBAHASAN**

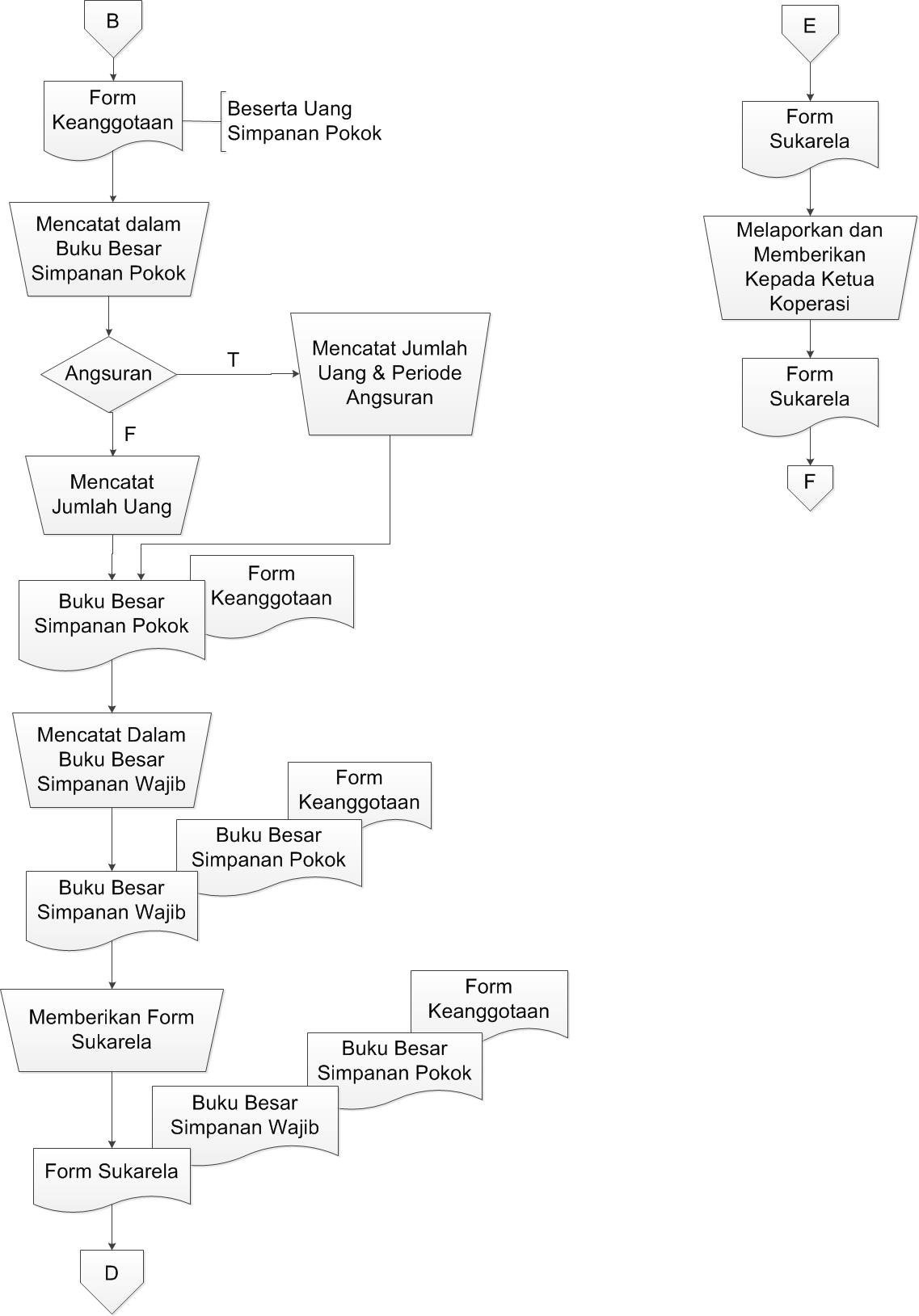
* 1. ***Flowchart* Sistem Berjalan Simpan Pinjam pada Koperasi Karyawan DKB Kantor Pusat (Kapus)**
     1. ***Flowchart* Sistem Berjalan**

1. ***Flowchart* Karyawan/Anggota**



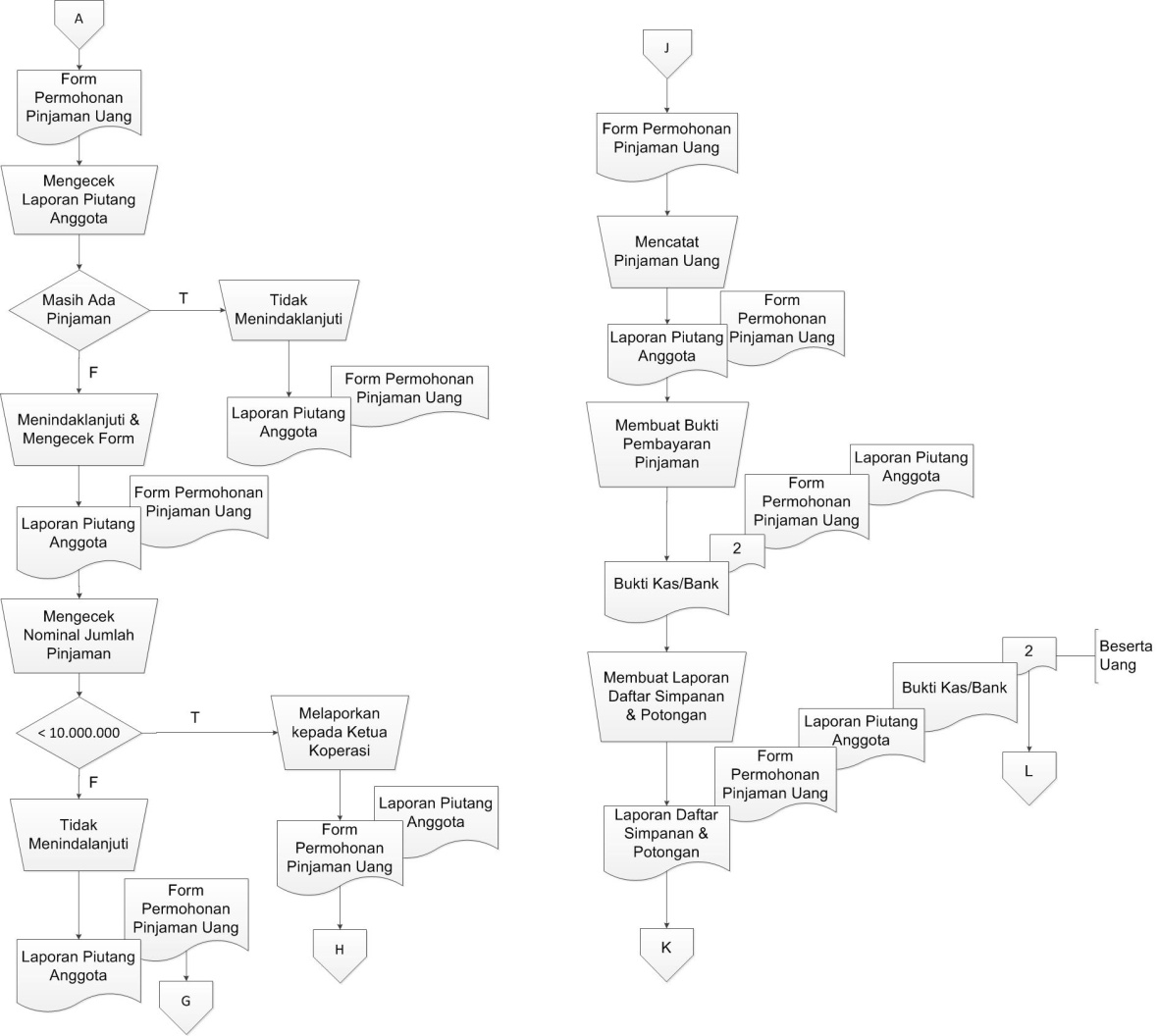
**Gambar 4.1 *Flowchart* Karyawan / Anggota**

1. ***Flowchart* Pengurus Koperasi (Simpanan)**



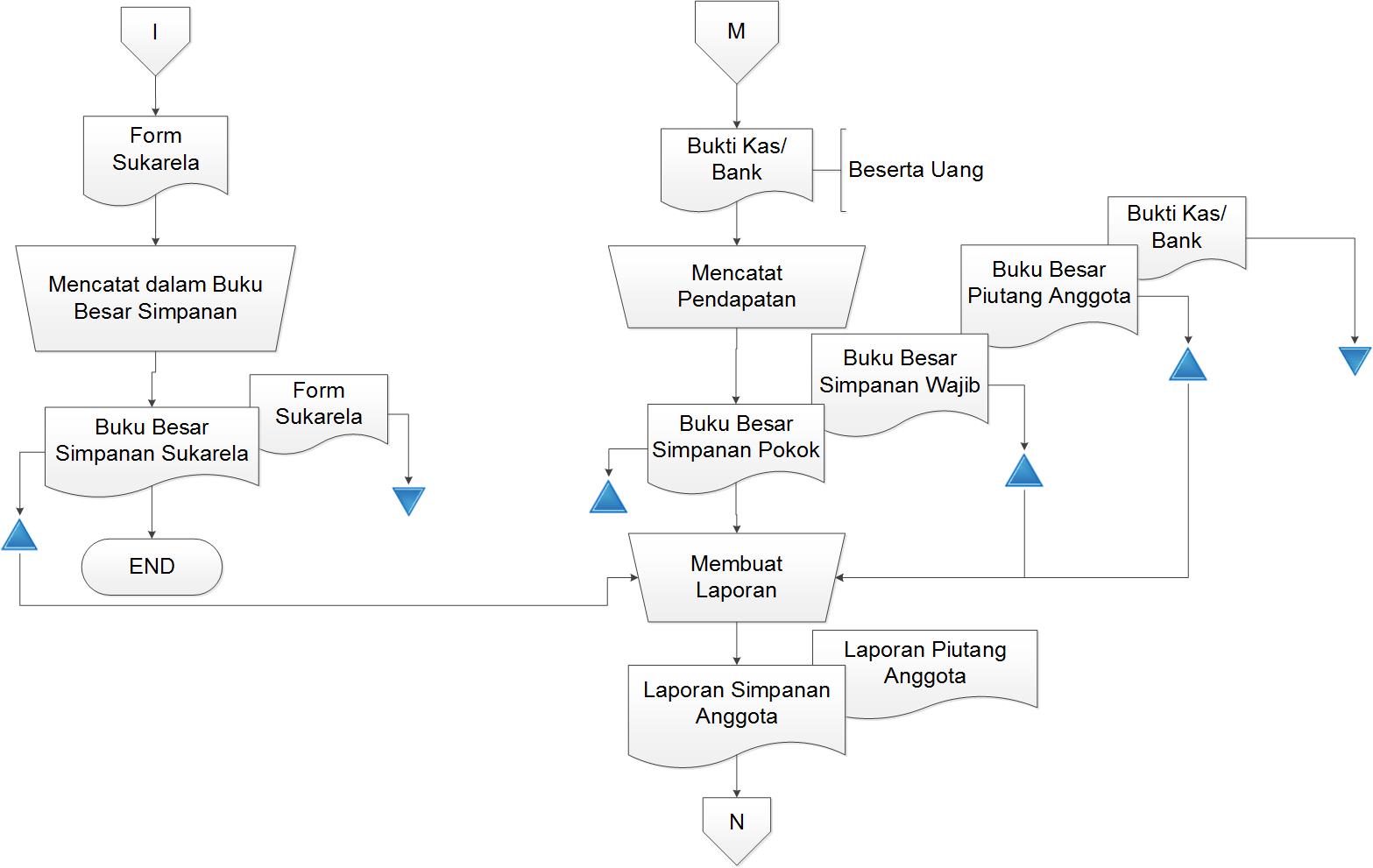
**Gambar 4.2 *Flowchart* Pengurus Koperasi (Simpanan)**

1. ***Flowchart* Pengurus Koperasi (Peminjaman)**



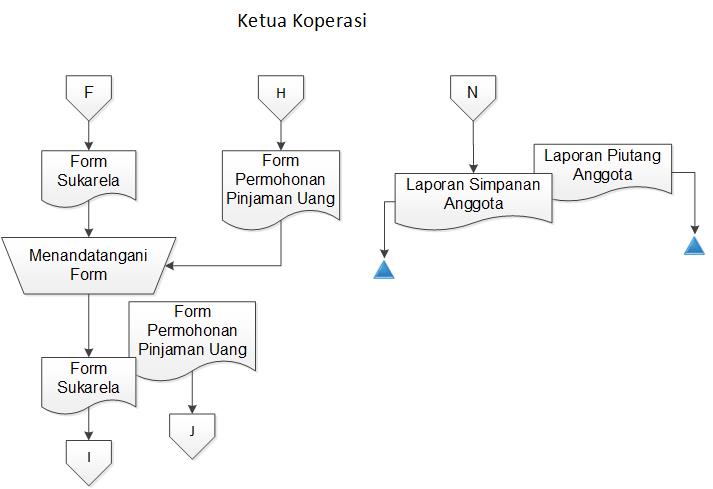
**Gambar 4.3 *Flowchart* Pengurus Koperasi (Peminjaman)**

1. ***Flowchart* Pengurus Koperasi (Laporan)**



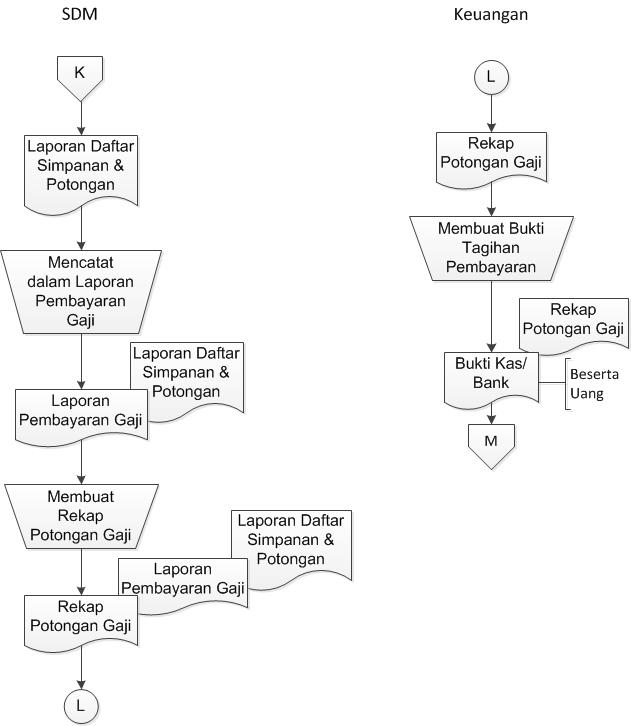
**Gambar 4.4 *Flowchart* Pengurus Koperasi**

1. ***Flowchart* Ketua Koperasi**



**Gambar 4.5 *Flowchart* Ketua Koperasi**

1. ***Flowchart* SDM & Keuangan**



**Gambar 4.6 *Flowchart* SDM & Keuangan**

* + 1. **Narasi Sistem Berjalan**

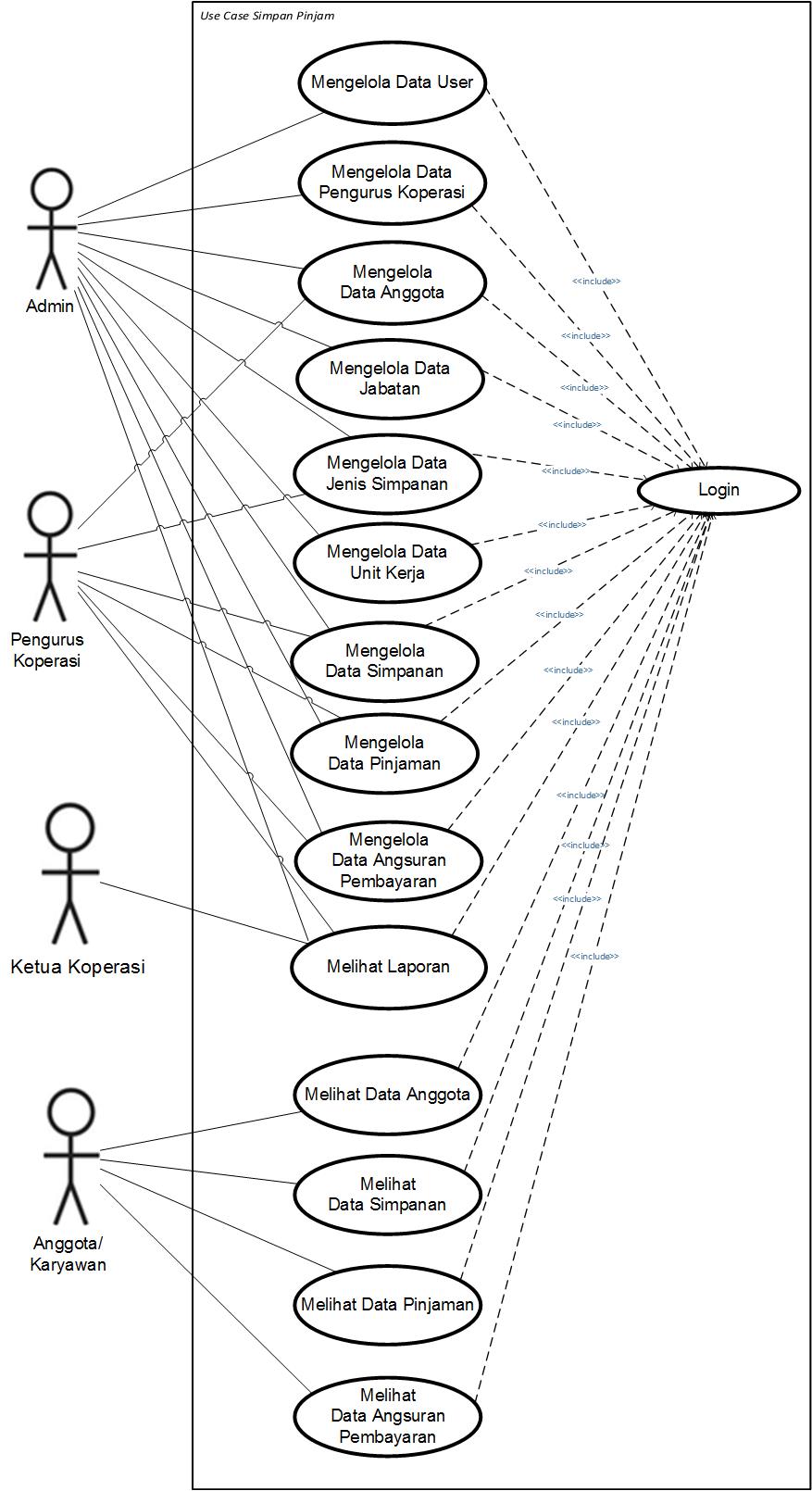
1. Langkah awal dalam sistem simpan pinjam pada Koperasi Karyawan DKB Kantor Pusat (Kapus), pihak karyawan DKB mendatangi koperasi dengan tujuan mengajukan simpanan maupun pinjaman.
2. Adapun, untuk melakukan simpanan baik wajib, pokok, maupun sukarela, pihak karyawan DKB diharuskan menjadi anggota terlebih dahulu dikarenakan proses transaksi simpanan yang bersamaan dengan proses pendaftaran menjadi anggota. Ataupun jika sudah menjadi anggota koperasi, pihak anggota tidak perlu mendatangi koperasi guna untuk melakukan penyetoran simpanan. Hal ini dikarenakan proses penyetoran yang dilakukan dengan melalui potongan gaji karyawan.
3. Kemudian, pihak pengurus koperasi akan memberikan form keanggotaan kepada pihak karyawan DKB. Dimana data-data yang tertera pada form diwajibkan bagi calon anggota untuk mengisinya secara lengkap.
4. Namun, jika pihak calon anggota tidak setuju dengan persyaratan-persyaratan yang telah tertera pada form, maka tidak mengharuskan bagi calon anggota untuk menindaklanjuti proses pendaftaran.
5. Apabila pihak calon anggota setuju dengan persyaratan yang berlaku, maka pihak calon anggota dapat menandatangani form keanggotaan beserta dengan memberikan sejumlah uang guna simpanan pokok.
6. Setelah pihak pengurus menerima form keanggotaan kembali beserta dengan uang simpanan pokok, maka pihak pengurus akan mencatatnya dalam Buku Besar Simpanan. Terdapat 2 cara pembayaran yang diterapkan untuk proses pembayaran uang simpanan pokok yakni secara tunai maupun angsuran. Apabila pihak calon anggota menerapkan sistem pembayaran angsuran, maka pihak pengurus akan mencatat sejumlah uang yang diterimanya beserta dengan mencatat jumlah periode angsuran yang diajukan oleh calon anggota. Namun, apabila pihak calon anggota menerapkan sistem pembayaran tunai, maka pihak pengurus hanya mencatat sejumlah uang yang diterimanya ke dalam Buku Besar Simpanan.
7. Dalam hal ini, proses pendaftaran dinyatakan selesai. Kemudian, pihak pengurus memberikan sebuah form yang berisikan surat pernyataan ikut berpartisipasi dalam Simpanan Sukarela. Untuk simpanan sukarela, pihak anggota diijinkan untuk tidak ikut berpartisipasi atau dalam arti simpanan sukarela bersifat tidak wajib bagi anggota koperasi.
8. Namun, apabila pihak anggota berminat untuk berpartisipasi dalam simpanan sukarela, maka diwajibkan untuk mengisi formulir serta menandatanganinya.
9. Kemudian, pihak pengurus akan mencatatnya dalam Buku Besar Simpanan Sukarela guna sebagai catatan hutang (tagihan) yang harus dibayarkan oleh pihak perusahaan DKB kepada koperasi.
10. Bersamaan dengan itu, pihak pengurus juga mencatat dalam Buku Besar Simpanan Wajib guna mencatat tagihan yang harus dibayarkan oleh pihak perusahaan DKB.
11. Kemudian, untuk proses peminjaman, langkah pertama dilakukan oleh pihak pengurus koperasi yang mengecek terlebih dahulu data anggota guna mengetahui periode/jangka waktu karyawan DKB tersebut menjadi anggota koperasi.
12. Apabila telah 1 tahun menjadi anggota, maka pihak pengurus menindaklanjuti proses peminjaman dengan memberikan Form Permohonan Pinjaman Uang kepada pihak anggota.
13. Setelah menerima form peminjaman, pihak anggota diminta untuk mengisi form tersebut sesuai dengan kebutuhan serta menyetujui persyaratan pembayaran.
14. Namun, apabila pihak anggota tidak menyetujui persyaratan pembayaran yang telah tertera, maka diharuskan untuk tidak menindaklanjuti pengisian form.
15. Sedangkan, apabila pihak anggota setuju dengan persyaratan pembayaran, maka setelah mengisi form dengan lengkap, pihak anggota dapat menandatangani form tersebut.
16. Setelah itu, form permohonan pinjaman yang telah diisi serta ditanda tangani oleh anggota, akan kembali di cek oleh pihak pengurus beserta dengan pengecekan nominal pinjaman.
17. Apabila nominal peminjaman yang diajukan oleh anggota kurang dari Rp. 10.000.000,- maka pihak pengurus dapat menindaklanjuti proses peminjaman dan memberikan form peminjaman kepada Ketua Koperasi guna di tanda tangani sebagai bukti persetujuan antara pihak Ketua Koperasi dengan pihak anggota.
18. Setelah form permohonan pinjaman diberikan kembali, pihak pengurus akan membuat bukti pembayaran dalam form Bukti Kas/Bank. Adapun form Bukti Kas/Bank dibuat sebanyak 2 rangkap. Dimana nantinya form rangkap pertama tersebut akan diberikan kepada pihak anggota yang bersangkutan beserta dengan uang pinjaman yang diajukan, sedangkan form rangkap kedua dijadikan sebagai arsip bagi pihak pengurus guna membuat laporan simpanan dan potongan.
19. Kemudian, setelah proses peminjaman selesai, pihak pengurus akan membuat Laporan Daftar Simpanan dan Potongan guna sebagai laporan tagihan yang akan diberikan kepada divisi SDM.
20. Setelah menerimanya, divisi SDM akan memasukkan data-data simpanan serta potongan yang ada ke dalam Laporan Pembayaran Gaji guna proses penghitungan gaji.
21. Kemudian, selain membuat Laporan Pembayaran Gaji, divisi SDM akan membuat Laporan Rekap Potongan Gaji guna diberikan kepada divisi Keuangan sebagai laporan penagihan pembayaran atas data potongan-potongan yang ada.
22. Maka setelah menerima Laporan Rekap Potongan Gaji, pihak keuangan akan membuat bukti pembayaran yakni berupa Bukti Kas/Bank. Adapun form Bukti Kas/Bank dibuat sebanyak 2 rangkap. Dimana nantinya form rangkap pertama tersebut akan diberikan kepada pihak pengurus koperasi yang bersangkutan beserta dengan uang yang tertera pada Laporan Rekap Potongan Gaji, sedangkan form rangkap kedua dijadikan sebagai arsip bagi divisi keuangan.
23. Kemudian, setelah pihak pengurus koperasi menerima sejumlah uang pembayaran dan Bukti Kas/Bank, maka pihak pengurus koperasi akan mencatat pendapatan ke dalam setiap buku besar simpanan dan piutang. Dalam arti, pihak pengurus koperasi meng-*update* tagihan setiap anggotanya terhadap simpanan maupun pinjaman.
24. Tidak hanya itu, setelah meng-*update* buku besar, pihak pengurus koperasi juga akan membuat laporan simpanan dan piutang anggota guna diberikan kepada Ketua Koperasi.
    1. **Kendala Sistem Berjalan Simpan Pinjam pada Koperasi Karyawan DKB Kantor Pusat (Kapus)**

Adapun setelah melakukan kegiatan *observasi* pada Koperasi Karyawan DKB Kantor Pusat (Kapus), ditemukan beberapa kendala dalam sistem simpan pinjam yang saat ini diterapkan yakni meliputi :

1. Proses pencatatan transaksi simpanan dan pinjaman yang masih dilakukan dengan cara pembukuan, serta proses penghitungan transaksi simpanan, pinjaman, maupun pembayaran angsuran atas peminjaman uang masih dilakukan dengan menggunakan program aplikasi *Microsoft Office Excel* yang dapat memungkinkan terjadinya kesalahan atau kekeliruan dalam perhitungan.
2. Penyajian bukti-bukti transaksi simpanan anggota maupun transaksi peminjaman uang masih dituangkan dalam bentuk lampiran-lampiran dokumen, sehingga dalam hal ini mengakibatkan pengurus koperasi kesulitan dalam proses pencarian data serta kurang efektifnya proses pembuatan rekapitulasi atau laporan yang nantinya akan diberikan kepada pihak Ketua Koperasi.

* 1. **Perancangan Sistem Informasi Simpan Pinjam pada Koperasi Karyawan DKB Kantor Pusat (Kapus)**
     1. **Perancangan Model**

1. ***Use Case Diagram***



**Gambar 4.7 *Use Case* Diagram**

1. ***Scenario Use Case***

|  |  |
| --- | --- |
| **Aktor** | **Deskripsi** |
| Administrator | Merupakan salah seorang pihak yang memiliki hak akses terpenting dalam suatu program aplikasi. Dimana pihak administrator dapat mengelola semua data yang terdapat dalam suatu aplikasi. |
| Pengurus Koperasi | Merupakan pihak penting lainnya yang dapat mengelola suatu aplikasi dan bertugas untuk melakukan semua aktivitas yang berkaitan dengan transaksi. |
| Ketua Koperasi | Merupakan pihak yang bertugas untuk melihat hasil akhir dari suatu pengelolaan data. Dimana hasil akhir tersebut biasanya berupa laporan periode suatu transaksi. |
| Anggota | Merupakan pihak yang hanya dapat mengakses data yang berkaitan dengan identitas masing-masing anggota saja. Aktivitas yang dapat dilakukan oleh pihak anggota biasanya seperti melihat data anggota, melihat data simpanan dan data pinjaman. |

**Tabel 4.1 Aktor *Scenario Use Case***

1. ***Scenario Use Case Login***

|  |  |
| --- | --- |
| **Reaksi Aktor** | **Reaksi Sistem** |
| **Skenario Normal** | |
| 1. Membuka Aplikasi Koperasi Simpan Pinjam dengan mengetikkan alamat *URL* yang terkait. |  |
|  | 1. Menampilkan halaman *Login.* |
|  | 1. Meminta pengguna memasukan *Username* serta *Password* dari masing-masing pengguna. |
| 1. Memasukan *Username* serta *Password*. |  |
| 1. *Click button Login.* |  |
|  | 1. Memeriksa *valid* atau tidaknya data *Username* serta *Password* yang terdapat pada tabel *Login.* |
|  | 1. Menampilkan halaman *Dashboard.* |
| **Skenario Alternatif** | |
| 1. Membuka Aplikasi Koperasi Simpan Pinjam dengan mengetikkan alamat *URL* yang terkait. |  |
|  | 1. Menampilkan halaman *Login.* |
|  | 1. Meminta pengguna memasukkan *Username* serta *Password* dari masing-masing pengguna. |
| 1. Memasukkan *Username* serta *Password*. |  |
| 1. *Click button Login.* |  |
|  | 1. Memeriksa *valid* atau tidaknya data *Username* serta *Password* yang terdapat pada tabel *Login.* |
|  | 1. Menampilkan notifikasi bahwa data *login* tidak *valid.* |
| 1. Memasukkan *Username* serta *Password* yang *valid.* |  |
| 1. *Click button Login.* |  |
|  | 1. Memeriksa *valid* atau tidaknya data *Username* serta *Password* yang terdapat pada tabel *Login.* |
|  | 1. Menampilkan halaman *Dashboard.* |

**Tabel 4.2 *Scenario* *Use Case* *Login***

1. ***Scenario Use Case* Mengelola Data Pengurus**

|  |  |
| --- | --- |
| **Reaksi Aktor** | **Reaksi Sistem** |
| **Skenario Normal** | |
| 1. Membuka Aplikasi Koperasi Simpan Pinjam dengan mengetikkan alamat *URL* yang terkait. |  |
| 1. *Login.* |  |
|  | 1. Menampilkan halaman *Dashboard* Aplikasi Koperasi Simpan Pinjam. |
| 1. *Click* Menu User. |  |
|  | 1. Menampilkan sub menu User. |
| 1. Memilih sub menu Data Pengurus. |  |
|  | 1. Menampilkan halaman List Pengurus Koperasi. |
| 1. *Click button* Tambah Data Pengurus. |  |
|  | 1. Menampilkan halaman Form Tambah Data Pengurus Koperasi. |
| 1. Memasukkan data-data Pengurus sesuai dengan form. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menyimpan data Pengurus ke tabel Pengurus. |
|  | 1. Menampilkan notifikasi bahwa data telah berhasil disimpan. |
|  | 1. Menampilkan halaman List Pengurus Koperasi*.* |
| 1. Melakukan pencarian data Pengurus yang akan diubah. |  |
|  | 1. Menampilkan data Pengurus yang dicari pada List Pengurus Koperasi. |
| 1. *Click button* Edit untuk melakukan perubahan pada data Pengurus yang akan diubah. |  |
|  | 1. Menampilkan halaman Form Edit Data Pengurus. |
| 1. Mengubah data Pengurus. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah diubah. |
|  | 1. Menyimpan data Pengurus yang telah diubah ke tabel Pengurus. |
|  | 1. Menampilkan notifikasi bahwa data Pengurustelah berhasil diubah. |
|  | 1. Menampilkan halaman List Pengurus Koperasi. |
| 1. Melakukan pencarian data Pengurus yang akan dihapus. |  |
|  | 1. Menampilkan data Pengurus yang dicari pada List Pengurus Koperasi. |
| 1. *Click button* Hapus untuk menghapus data Pengurus. |  |
|  | 1. Menampilkan pesan konfirmasi hapus data. |
| 1. *Click option button* Hapus. |  |
|  | 1. Menampilkan notifikasi bahwa data berhasil dihapus. |
|  | 1. Melakukan *refresh* terhadap halaman List Pengurus. |
| **Skenario Alternatif** | |
| 1. Membuka Aplikasi Koperasi Simpan Pinjam dengan mengetikkan alamat *URL* yang terkait. |  |
| 1. *Login.* |  |
|  | 1. Menampilkan halaman *Dashboard* Aplikasi Koperasi Simpan Pinjam. |
| 1. *Click* Menu User. |  |
|  | 1. Menampilkan sub menu User. |
| 1. Memilih sub menu Data Pengurus. |  |
|  | 1. Menampilkan halaman List Pengurus Koperasi. |
| 1. *Click button* Tambah Data Pengurus. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menampilkan notifikasi bahwa data yang disimpan tidak *valid*/ sudah tersedia. |
| 1. Memperbaiki data Pengurus dengan yang belum tersedia/nilainya *valid.* |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menyimpan data Pengurus ke tabel Pengurus. |
|  | 1. Menampilkan notifikasi bahwa data telah berhasil disimpan. |
|  | 1. Menampilkan halaman List Pengurus. |
| 1. Melakukan pencarian data Pengurus yang akan diubah. |  |
|  | 1. Menampilkan data Pengurus yang dicari pada List Pengurus. |
| 1. *Click button* Edit untuk melakukan perubahan pada data Pengurus yang akan diubah. |  |
|  | 1. Menampilkan halaman Form Edit Data Pengurus. |
| 1. Mengubah data Pengurus sesuai dengan kebutuhan. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah diubah. |
|  | 1. Menampilkan notifikasi bahwa data yang diubah tidak *valid.* |
| 1. Memperbaiki data Pengurus dengan nilai yang *valid.* |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah diubah. |
|  | 1. Menyimpan data Pengurus yang telah diubah ke tabel Pengurus. |
|  | 1. Menampilkan notifikasi bahwa data telah berhasil diubah. |
|  | 1. Menampilkan halaman List Pengurus. |
| 1. Melakukan pencarian data Pengurus yang akan dihapus. |  |
|  | 1. Menampilkan data Pengurus yang dicari pada List Pengurus. |
| 1. *Click button* Hapus untuk menghapus data Pengurus. |  |
|  | 1. Menampilkan pesan konfirmasi hapus data. |
| 1. *Click button Cancel.* |  |
|  | 1. Melakukan *refresh* halaman List Pengurus. |

**Tabel 4.3 *Scenario* *Use Case* Mengelola Data Pengurus**

1. ***Scenario Use Case* Mengelola Data *User***

|  |  |
| --- | --- |
| **Reaksi Aktor** | **Reaksi Sistem** |
| **Skenario Normal** | |
| 1. Membuka Aplikasi Koperasi Simpan Pinjam dengan mengetikkan alamat *URL* yang terkait. |  |
| 1. *Login.* |  |
|  | 1. Menampilkan halaman *Dashboard* Aplikasi Koperasi Simpan Pinjam. |
| 1. *Click* Menu *Settings*. |  |
|  | 1. Menampilkan sub menu *Settings*. |
| 1. Memilih sub menu User. |  |
|  | 1. Menampilkan halaman List User. |
| 1. *Click button* Tambah User. |  |
|  | 1. Menampilkan halaman Form Tambah User. |
| 1. Memasukkan data-data *User* sesuai dengan form. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menyimpan data *User* ke tabel User. |
|  | 1. Menampilkan notifikasi bahwa data telah berhasil disimpan. |
|  | 1. Menampilkan halaman List User*.* |
| 1. Melakukan pencarian data *User* yang akan diubah. |  |
|  | 1. Menampilkan data *User* yang dicari pada List User. |
| 1. *Click button* Edit untuk melakukan perubahan pada data *User* yang akan diubah. |  |
|  | 1. Menampilkan halaman Form Edit User. |
| 1. Mengubah data *User*. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah diubah. |
|  | 1. Menyimpan data *User* yang telah diubah ke tabel User. |
|  | 1. Menampilkan notifikasi bahwa data *User* telah berhasil diubah. |
|  | 1. Menampilkan halaman List User. |
| 1. Melakukan pencarian data *User* yang akan dihapus. |  |
|  | 1. Menampilkan data *User* yang dicari pada List User. |
| 1. *Click button* Hapus untuk menghapus data *User*. |  |
|  | 1. Menampilkan pesan konfirmasi hapus data. |
| 1. *Click option button* Hapus. |  |
|  | 1. Menampilkan notifikasi bahwa data berhasil dihapus. |
|  | 1. Melakukan *refresh* terhadap halaman List User. |
| **Skenario Alternatif** | |
| 1. Membuka Aplikasi Koperasi Simpan Pinjam dengan mengetikkan alamat *URL* yang terkait. |  |
| 1. *Login.* |  |
|  | 1. Menampilkan halaman *Dashboard* Aplikasi Koperasi Simpan Pinjam. |
| 1. *Click* Menu *Settings*. |  |
|  | 1. Menampilkan sub menu *Settings*. |
| 1. Memilih sub menu User. |  |
|  | 1. Menampilkan halaman List User. |
| 1. *Click button* Tambah User. |  |
|  | 1. Menampilkan halaman Form User. |
| 1. Memasukkan data-data *User* sesuai dengan form. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menampilkan notifikasi bahwa data yang disimpan tidak *valid*/ sudah tersedia. |
| 1. Memperbaiki data *User* dengan yang belum tersedia/nilainya *valid.* |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menyimpan data *User* ke tabel User. |
|  | 1. Menampilkan notifikasi bahwa data telah berhasil disimpan. |
|  | 1. Menampilkan halaman List User. |
| 1. Melakukan pencarian data *User* yang akan diubah. |  |
|  | 1. Menampilkan data *User* yang dicari pada List User. |
| 1. *Click button* Edit untuk melakukan perubahan pada data *User* yang akan diubah. |  |
|  | 1. Menampilkan halaman Form Edit User. |
| 1. Mengubah data *User* sesuai dengan kebutuhan. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah diubah. |
|  | 1. Menampilkan notifikasi bahwa data yang diubah tidak *valid.* |
| 1. Memperbaiki data *User* dengan nilai yang *valid.* |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah diubah. |
|  | 1. Menyimpan data *User* yang telah diubah ke tabel User. |
|  | 1. Menampilkan notifikasi bahwa data telah berhasil diubah. |
|  | 1. Menampilkan halaman List User. |
| 1. Melakukan pencarian data *User* yang akan dihapus. |  |
|  | 1. Menampilkan data *User* yang dicari pada List User. |
| 1. *Click button* Hapus untuk menghapus data *User*. |  |
|  | 1. Menampilkan pesan konfirmasi hapus data. |
| 1. *Click button Cancel.* |  |
|  | 1. Melakukan *refresh* halaman List User. |

**Tabel 4.4 *Scenario* *Use Case* Mengelola Data *User***

1. ***Scenario Use Case* Mengelola Data Anggota**

|  |  |
| --- | --- |
| **Reaksi Aktor** | **Reaksi Sistem** |
| **Skenario Normal** | |
| 1. Membuka Aplikasi Koperasi Simpan Pinjam dengan mengetikkan alamat *URL* yang terkait. |  |
| 1. *Login.* |  |
|  | 1. Menampilkan halaman *Dashboard* aplikasi Koperasi Simpan Pinjam. |
| 1. *Click* Menu Data Master. |  |
|  | 1. Menampilkan sub menu Data Master. |
| 1. Memilih sub menu Anggota. |  |
|  | 1. Menampilkan halaman List Anggota. |
| 1. *Click* button Tambah Anggota. |  |
|  | 1. Menampilkan halaman Form Keanggotaan. |
| 1. Memasukkan data-data anggota sesuai dengan form. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menyimpan data Anggota ke tabel Anggota. |
|  | 1. Menampilkan notifikasi bahwa data telah berhasil disimpan. |
|  | 1. Menampilkan halaman List Anggota. |
| 1. Melakukan pencarian data Anggota yang akan diubah. |  |
|  | 1. Menampilkan data Anggota yang dicari pada List Anggota. |
| 1. *Click button* Edit untuk melakukan perubahan pada data anggota yang akan diubah. |  |
|  | 1. Menampilkan halaman Form Edit Anggota. |
| 1. Mengubah data Anggota sesuai dengan kebutuhan. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah diubah. |
|  | 1. Menyimpan data Anggota yang telah diubah ke tabel Anggota. |
|  | 1. Menampilkan notifikasi bahwa data Anggota telah berhasil diubah. |
|  | 1. Menampilkan halaman List Anggota. |
| 1. Melakukan pencarian data Anggota yang akan dihapus. |  |
|  | 1. Menampilkan data Anggota yang dicari pada List Anggota. |
| 1. *Click button* Hapus untuk menghapus data anggota. |  |
|  | 1. Menampilkan pesan konfirmasi hapus data. |
| 1. *Click option button* Hapus. |  |
|  | 1. Menampilkan notifikasi bahwa data berhasil dihapus. |
|  | 1. Melakukan *refresh* terhadap halaman List Anggota. |
| 1. Melakukan pencarian data Anggota. |  |
|  | 1. Menampilkan data Anggota yang dicari pada List Anggota. |
| 1. *Click button* Detail. |  |
|  | 1. Menampilkan halaman View Detail Keanggotaan. |
| **Skenario Alternatif** | |
| 1. Membuka Aplikasi Koperasi Simpan Pinjam dengan mengetikkan alamat *URL* yang terkait. |  |
| 1. *Login.* |  |
|  | 1. Menampilkan halaman *Dashboard* Aplikasi Koperasi Simpan Pinjam. |
| 1. *Click* Menu Data Master. |  |
|  | 1. Menampilkan sub menu Data Master |
| 1. Memilih sub menu Anggota. |  |
|  | 1. Menampilkan halaman List Anggota. |
| 1. *Click button* Tambah Anggota. |  |
|  | 1. Menampilkan halaman Form Keanggotaan. |
| 1. Memasukkan data-data Anggota sesuai dengan form. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menampilkan notifikasi bahwa data Anggota yang dimasukkan sudah tersedia*/*tidak *valid.* |
| 1. Memperbaiki data Anggota dengan yang belum tersedia/*valid.* |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menyimpan data Anggota ke tabel Anggota. |
|  | 1. Menampilkan notifikasi bahwa data telah berhasil disimpan. |
|  | 1. Menampilkan halaman List Anggota. |
| 1. Melakukan pencarian data anggota yang akan diubah. |  |
|  | 1. Menampilkan data anggota yang dicari pada List Anggota. |
| 1. *Click button* Edit untuk melakukan perubahan pada data anggota yang akan diubah. |  |
|  | 1. Menampilkan halaman Form Edit Anggota. |
| 1. Mengubah data anggota sesuai dengan kebutuhan. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah diubah. |
|  | 1. Menampilkan notifikasi bahwa data yang diubah tidak *valid.* |
| 1. Memperbaiki data anggota dengan nilai yang *valid.* |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah diubah. |
|  | 1. Menyimpan data Anggota yang telah diubah ke tabel Anggota. |
|  | 1. Menampilkan notifikasi bahwa data telah berhasil diubah dan tampilan akan kembali ke halaman List Anggota. |
| 1. Melakukan pencarian data Anggota yang akan dihapus. |  |
|  | 1. Menampilkan data Anggota yang dicari pada List Anggota. |
| 1. *Click button* Hapus untuk menghapus data Anggota. |  |
|  | 1. Menampilkan pesan konfirmasi hapus data. |
| 1. *Click button Cancel.* |  |
|  | 1. Melakukan *refresh* terhadap halaman List Anggota. |

**Tabel 4.5 *Scenario* *Use Case* Mengelola Data Anggota**

1. ***Scenario Use Case* Mengelola Data Jabatan**

|  |  |
| --- | --- |
| **Reaksi Aktor** | **Reaksi Sistem** |
| **Skenario Normal** | |
| 1. Membuka Aplikasi Koperasi Simpan Pinjam dengan mengetikkan alamat *URL* yang terkait. |  |
| 1. *Login.* |  |
|  | 1. Menampilkan halaman *Dashboard* aplikasi Koperasi Simpan Pinjam. |
| 1. *Click* Menu Data Master. |  |
|  | 1. Menampilkan sub menu Data Master. |
| 1. Memilih sub menu Jabatan. |  |
|  | 1. Menampilkan halaman List Jabatan. |
| 1. *Click button* Tambah Jabatan. |  |
|  | 1. Menampilkan halaman Form Tambah Jabatan. |
| 1. Memasukkan data-data Jabatan sesuai dengan form. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menyimpan data Jabatan ke tabel Jabatan. |
|  | 1. Menampilkan notifikasi bahwa data telah berhasil disimpan. |
|  | 1. Menampilkan halaman List Jabatan. |
| 1. Melakukan pencarian data Jabatan yang akan diubah. |  |
|  | 1. Menampilkan data Jabatan yang dicari pada List Jabatan. |
| 1. *Click button* Edit untuk melakukan perubahan pada data Jabatan. |  |
|  | 1. Menampilkan halaman Form Edit Jabatan. |
| 1. Mengubah data Jabatan sesuai dengan kebutuhan. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah diubah. |
|  | 1. Menyimpan data Jabatan yang telah diubah ke tabel Jabatan. |
|  | 1. Menampilkan notifikasi bahwa data Jabatan telah berhasil diubah. |
|  | 1. Menampilkan halaman List Jabatan. |
| 1. Melakukan pencarian data Jabatan yang akan dihapus. |  |
|  | 1. Menampilkan data Jabatan yang dicari pada List Jabatan. |
| 1. *Click button* Hapus untuk menghapus data Jabatan. |  |
|  | 1. Menampilkan pesan konfirmasi hapus data. |
| 1. *Click option button* Hapus. |  |
|  | 1. Menampilkan notifikasi bahwa data berhasil dihapus. |
|  | 1. Melakukan *refresh* pada halaman List Jabatan. |
| **Skenario Alternatif** | |
| 1. Membuka Aplikasi Koperasi Simpan Pinjam dengan mengetikkan alamat *URL* yang terkait. |  |
| 1. *Login.* |  |
|  | 1. Menampilkan halaman *Dashboard* Aplikasi Koperasi Simpan Pinjam. |
| 1. *Click* Menu Data Master. |  |
|  | 1. Menampilkan sub menu Data Master. |
| 1. Memilih sub menu Jabatan. |  |
|  | 1. Menampilkan halaman List Jabatan. |
| 1. *Click button* Tambah Jabatan. |  |
|  | 1. Menampilkan halaman Form Tambah Jabatan. |
| 1. Memasukkan data-data Jabatan sesuai dengan form. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menampilkan notifikasi bahwa data yang disimpan tidak *valid*/ sudah tersedia. |
| 1. Memperbaiki data Jabatan dengan yang belum tersedia/nilainya *valid.* |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menyimpan data Jabatan ke tabel Jabatan. |
|  | 1. Menampilkan notifikasi bahwa data telah berhasil disimpan. |
|  | 1. Menampilkan halaman List Jabatan. |
| 1. Melakukan pencarian data Jabatan yang akan diubah. |  |
|  | 1. Menampilkan data Jabatan yang dicari pada List Jabatan. |
| 1. *Click button* Edit untuk melakukan perubahan pada data Jabatan yang akan diubah. |  |
|  | 1. Menampilkan halaman Form Edit Jabatan. |
| 1. Mengubah data Jabatan sesuai dengan kebutuhan. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah diubah. |
|  | 1. Menampilkan notifikasi bahwa data yang diubah tidak *valid/* sudah tersedia*.* |
| 1. Memperbaiki data Jabatan dengan nilai yang *valid/* belum tersedia*.* |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah diubah. |
|  | 1. Menyimpan data Jabatan yang telah diubah ke tabel Jabatan. |
|  | 1. Menampilkan notifikasi bahwa data telah berhasil diubah. |
|  | 1. Menampilkan halaman List Jabatan. |
| 1. Melakukan pencarian data Jabatan yang akan dihapus. |  |
|  | 1. Menampilkan data Jabatan yang dicari pada List Jabatan. |
| 1. *Click button* Hapus untuk menghapus data Jabatan. |  |
|  | 1. Menampilkan pesan konfirmasi hapus data. |
| 1. *Click button Cancel.* |  |
|  | 1. Melakukan *refresh* pada halaman List Jabatan. |

**Tabel 4.6 *Scenario* *Use Case* Mengelola Data Jabatan**

1. ***Scenario Use Case* Mengelola Data Jenis Simpanan**

|  |  |
| --- | --- |
| **Reaksi Aktor** | **Reaksi Sistem** |
| **Skenario Normal** | |
| 1. Membuka Aplikasi Koperasi Simpan Pinjam dengan mengetikkan alamat URL yang terkait. |  |
| 1. *Login.* |  |
|  | 1. Menampilkan halaman *Dashboard* Aplikasi Koperasi Simpan Pinjam. |
| 1. *Click* Menu Data Master. |  |
|  | 1. Menampilkan sub menu Data Master. |
| 1. Memilih sub menu Jenis Simpanan. |  |
|  | 1. Menampilkan halaman List Jenis Simpanan. |
| 1. *Click button* Tambah Jenis Simpanan. |  |
|  | 1. Menampilkan halaman Form Jenis Simpanan. |
| 1. Memasukkan data-data Jenis Simpanan sesuai dengan form. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menyimpan data Jenis Simpanan ke tabel Jenis Simpanan. |
|  | 1. Menampilkan notifikasi bahwa data telah berhasil disimpan. |
|  | 1. Menampilkan halaman List Jenis Simpanan. |
| 1. Melakukan pencarian data Jenis Simpanan yang akan diubah. |  |
|  | 1. Menampilkan data Jenis Simpanan yang dicari pada List Jenis Simpanan. |
| 1. *Click button* Edit untuk melakukan perubahan pada data Jenis Simpanan. |  |
|  | 1. Menampilkan halaman Form Edit Jenis Simpanan. |
| 1. Mengubah data Jenis Simpanan sesuai dengan kebutuhan. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah diubah. |
|  | 1. Menyimpan data Jenis Simpanan yang telah diubah ke tabel Jenis Simpanan. |
|  | 1. Menampilkan notifikasi bahwa data Jenis Pinjaman telah berhasil diubah. |
|  | 1. Menampilkan halaman List Jenis Simpanan. |
| 1. Melakukan pencarian data Jenis Simpanan yang akan dihapus. |  |
|  | 1. Menampilkan data jenis simpanan yang dicari pada List Jenis Simpanan. |
| 1. *Click button* Hapus untuk menghapus data jenis simpanan. |  |
|  | 1. Menampilkan pesan konfirmasi hapus data. |
| 1. *Click option button* Hapus. |  |
|  | 1. Menampilkan notifikasi bahwa data berhasil dihapus. |
|  | 1. Melakukan *refresh* pada halaman List Jenis Simpanan. |
| 1. Melakukan pencarian data Jenis Simpanan. |  |
|  | 1. Menampilkan data Jenis Simpanan yang dicari pada List Jenis Simpanan. |
| 1. *Click button* Detail. |  |
|  | 1. Menampilkan halaman List Data Simpanan Anggota. |
| **Skenario Alternatif** | |
| 1. Membuka Aplikasi Koperasi Simpan Pinjam dengan mengetikkan alamat URL yang terkait. |  |
| 1. *Login.* |  |
|  | 1. Menampilkan halaman *Dashboard* Aplikasi Koperasi Simpan Pinjam. |
| 1. *Click* Menu Data Master. |  |
|  | 1. Menampilkan sub menu Data Master. |
| 1. Memilih sub menu Jenis Simpanan. |  |
|  | 1. Menampilkan halaman List Jenis Simpanan. |
| 1. *Click button* Tambah Jenis Simpanan. |  |
|  | 1. Menampilkan halaman Form Jenis Simpanan. |
| 1. Memasukkan data-data Jenis Simpanan sesuai dengan form. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menampilkan notifikasi bahwa data yang disimpan tidak *valid*/ sudah tersedia. |
| 1. Memperbaiki data Jenis Simpanan dengan yang belum tersedia/nilainya *valid.* |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menyimpan data Jenis Simpanan ke tabel Jenis Simpanan. |
|  | 1. Menampilkan notifikasi bahwa data telah berhasil disimpan. |
|  | 1. Menampilkan halaman List Jenis Simpanan. |
| 1. Melakukan pencarian data Jenis Simpanan yang akan diubah. |  |
|  | 1. Menampilkan data Jenis Simpanan yang dicari pada List Jenis Simpanan. |
| 1. *Click button* Edit untuk melakukan perubahan pada data jenis simpanan yang akan diubah. |  |
|  | 1. Menampilkan halaman Form Edit Jenis Simpanan. |
| 1. Mengubah data Jenis Simpanan sesuai dengan kebutuhan. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah diubah. |
|  | 1. Menampilkan notifikasi bahwa data yang diubah tidak *valid.* |
| 1. Memperbaiki data Jenis Simpanan dengan nilai yang *valid.* |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah diubah. |
|  | 1. Menyimpan data Jenis Simpanan yang telah diubah ke tabel Jenis Simpanan. |
|  | 1. Menampilkan notifikasi bahwa data telah berhasil diubah. |
|  | 1. Menampilkan halaman List Jenis Simpanan. |
| 1. Melakukan pencarian data jenis simpanan yang akan dihapus. |  |
|  | 1. Menampilkan data jenis simpanan yang dicari pada List Anggota. |
| 1. *Click button* Hapus untuk menghapus data jenis simpanan. |  |
|  | 1. Menampilkan pesan konfirmasi hapus data. |
| 1. *Click button Cancel.* |  |
|  | 1. Melakukan *refresh* pada halaman List Jenis Simpanan. |

**Tabel 4.7 *Scenario* *Use Case* Mengelola Data Jenis Simpanan**

1. ***Scenario Use Case* Mengelola Data Unit Kerja**

|  |  |
| --- | --- |
| **Reaksi Aktor** | **Reaksi Sistem** |
| **Skenario Normal** | |
| 1. Membuka Aplikasi Koperasi Simpan Pinjam dengan mengetikkan alamat *URL* yang terkait. |  |
| 1. *Login.* |  |
|  | 1. Menampilkan halaman *Dashboard* Aplikasi Koperasi Simpan Pinjam. |
| 1. *Click* Menu Data Master. |  |
|  | 1. Menampilkan sub menu Data Master. |
| 1. Memilih sub menu Unit Kerja. |  |
|  | 1. Menampilkan halaman List Unit Kerja. |
| 1. *Click button* Tambah Unit Kerja. |  |
|  | 1. Menampilkan halaman Form Tambah Unit Kerja. |
| 1. Memasukkan data-data Unit Kerja sesuai dengan form. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menyimpan data Unit Kerja ke tabel Unit Kerja. |
|  | 1. Menampilkan notifikasi bahwa data telah berhasil disimpan. |
|  | 1. Menampilkan halaman List Unit Kerja. |
| 1. Melakukan pencarian data Unit Kerja yang akan diubah. |  |
|  | 1. Menampilkan data Unit Kerja yang dicari pada List Unit Kerja. |
| 1. *Click button* Edit untuk melakukan perubahan pada data Unit Kerja. |  |
|  | 1. Menampilkan halaman Form Edit Unit Kerja. |
| 1. Mengubah data Unit Kerja sesuai dengan kebutuhan. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah diubah. |
|  | 1. Menyimpan data Unit Kerja yang telah diubah ke tabel Unit Kerja. |
|  | 1. Menampilkan notifikasi bahwa data Unit Kerja telah berhasil diubah. |
|  | 1. Menampilkan halaman List Unit Kerja. |
| 1. Melakukan pencarian data Unit Kerja yang akan dihapus. |  |
|  | 1. Menampilkan data Unit Kerja yang dicari pada List Unit Kerja. |
| 1. *Click button* Hapus untuk menghapus data Unit Kerja. |  |
|  | 1. Menampilkan pesan konfirmasi hapus data. |
| 1. *Click option button* Hapus. |  |
|  | 1. Menampilkan notifikasi bahwa data berhasil dihapus. |
|  | 1. Melakukan *refresh* pada halaman List Unit Kerja. |
| **Skenario Alternatif** | |
| 1. Membuka Aplikasi Koperasi Simpan Pinjam dengan mengetikkan alamat *URL* yang terkait. |  |
| 1. *Login.* |  |
|  | 1. Menampilkan halaman *Dashboard* Aplikasi Koperasi Simpan Pinjam. |
| 1. *Click* Menu Data Master. |  |
|  | 1. Menampilkan sub menu Data Master. |
| 1. Memilih sub menu Unit Kerja. |  |
|  | 1. Menampilkan halaman List Unit Kerja. |
| 1. *Click button* Tambah Unit Kerja. |  |
|  | 1. Menampilkan halaman Form Unit Kerja. |
| 1. Memasukkan data-data Unit Kerja sesuai dengan form. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menampilkan notifikasi bahwa data yang disimpan tidak *valid*/ sudah tersedia. |
| 1. Memperbaiki data Unit Kerja dengan yang belum tersedia /nilainya *valid.* |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menyimpan data Unit Kerja ke tabel Unit Kerja. |
|  | 1. Menampilkan notifikasi bahwa data telah berhasil disimpan. |
|  | 1. Menampilkan halaman List Unit Kerja. |
| 1. Melakukan pencarian data Unit Kerja yang akan diubah. |  |
|  | 1. Menampilkan data Unit Kerja yang dicari pada List Unit Kerja. |
| 1. *Click button* Edit untuk melakukan perubahan pada data Unit Kerja yang akan diubah. |  |
|  | 1. Menampilkan halaman Form Edit Unit Kerja. |
| 1. Mengubah data Unit Kerja sesuai dengan kebutuhan. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah diubah. |
|  | 1. Menampilkan notifikasi bahwa data yang diubah tidak *valid.* |
| 1. Memperbaiki data Unit Kerja dengan nilai yang *valid.* |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah diubah. |
|  | 1. Menyimpan data Unit Kerja yang telah diubah ke tabel Unit Kerja. |
|  | 1. Menampilkan notifikasi bahwa data telah berhasil diubah. |
|  | 1. Menampilkan halaman List Unit Kerja. |
| 1. Melakukan pencarian data Unit Kerja yang akan dihapus. |  |
|  | 1. Menampilkan data Unit Kerja yang dicari pada List Unit Kerja. |
| 1. *Click button* Hapus untuk menghapus data Unit Kerja. |  |
|  | 1. Menampilkan pesan konfirmasi hapus data. |
| 1. *Click button Cancel.* |  |
|  | 1. Melakukan *refresh* pada halaman List Unit Kerja. |

**Tabel 4.8 *Scenario* *Use Case* Mengelola Data Unit Kerja**

1. ***Scenario Use Case* Mengelola Data Simpanan**

|  |  |
| --- | --- |
| **Reaksi Aktor** | **Reaksi Sistem** |
| **Skenario Normal** | |
| 1. Membuka Aplikasi Koperasi Simpan Pinjam dengan mengetikkan alamat *URL* yang terkait. |  |
| 1. *Login.* |  |
|  | 1. Menampilkan halaman *Dashboard* Aplikasi Koperasi Simpan Pinjam. |
| 1. *Click* Menu Simpanan. |  |
|  | 1. Menampilkan List Simpanan Anggota. |
| 1. *Click button* Tambah Data Simpanan. |  |
|  | 1. Menampilkan halaman Form Tambah Data Simpanan. |
| 1. Memasukkan data Anggota beserta nominal masing-masing Simpanan yang ada. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menyimpan data Simpanan Anggota ke tabel Simpanan. |
|  | 1. Menampilkan notifikasi transaksi telah berhasil disimpan. |
|  | 1. Menampilkan halaman List Simpanan Anggota. |
| 1. Melakukan pencarian data Simpanan yang akan diubah. |  |
|  | 1. Menampilkan data Simpanan yang dicari pada List Simpanan Anggota. |
| 1. *Click button* Edit untuk melakukan perubahan pada data Anggota maupun data Simpanan. |  |
|  | 1. Menampilkan halaman Form Edit Simpanan. |
| 1. Mengubah data-data sesuai dengan kebutuhan. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah diubah. |
|  | 1. Menyimpan data Simpanan Anggota yang telah diubah ke tabel Simpanan. |
|  | 1. Menampilkan notifikasi bahwa data Simpanan telah berhasil diubah. |
|  | 1. Menampilkan halaman List Simpanan Anggota. |
| 1. Melakukan pencarian data Simpanan. |  |
|  | 1. Menampilkan data Simpanan yang dicari pada List Simpanan Anggota. |
| 1. *Click button* Transaksi untuk melakukan pembayaran Simpanan. |  |
|  | 1. Menampilkan halaman Form Pembayaran Simpanan Anggota. |
| 1. Memasukkan data-data Pembayaran Simpanan sesuai dengan form. |  |
| 1. Memilih Jenis Pembayaran Simpanan. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menyimpan data Pembayaran Simpanan ke tabel Bayar Simpanan. |
|  | 1. Menampilkan notifikasi bahwa transaksi pembayaran telah berhasil disimpan. |
|  | 1. Menampilkan halaman List Simpanan Anggota*.* |
| **Skenario Alternatif** | |
| 1. Membuka Aplikasi Koperasi Simpan Pinjam dengan mengetikkan alamat URL yang terkait. |  |
| 1. *Login.* |  |
|  | 1. Menampilkan halaman *Dashboard* Aplikasi Koperasi Simpan Pinjam. |
| 1. *Click* Menu Simpanan. |  |
|  | 1. Menampilkan halaman List Simpanan Anggota. |
| 1. *Click button* Tambah Data Simpanan. |  |
|  | 1. Menampilkan halaman Form Tambah Data Simpanan. |
| 1. Memasukkan data Anggota beserta nominal masing-masing Simpanan yang ada. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menampilkan notifikasi data yang dimasukkan tidak *valid*. |
| 1. Memperbaiki data Simpanan Anggota dengan data yang nilainya *valid.* |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menyimpan data Simpanan Anggota ke table Simpanan. |
|  | 1. Menampilkan notifikasi transaksi telah berhasil disimpan. |
|  | 1. Menampilkan halaman List Simpanan Anggota. |
| 1. Melakukan pencarian data Simpanan yang akan diubah. |  |
|  | 1. Menampilkan data Simpanan yang dicari pada List Simpanan Anggota. |
| 1. *Click button* Edit untuk melakukan perubahan pada data Anggota maupun data Simpanan. |  |
|  | 1. Menampilkan halaman Form Edit Simpanan. |
| 1. Mengubah data-data sesuai dengan kebutuhan. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah diubah. |
|  | 1. Menampilkan notifikasi bahwa data yang diubah tidak *valid.* |
| 1. Memperbaiki data Simpanan Anggota dengan nilai yang *valid.* |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah diubah. |
|  | 1. Menyimpan data Simpanan yang telah diubah ke tabel Simpanan. |
|  | 1. Menampilkan notifikasi bahwa data telah berhasil diubah. |
|  | 1. Menampilkan halaman List Simpanan Anggota. |
| 1. Melakukan pencarian data Simpanan. |  |
|  | 1. Menampilkan data Simpanan yang dicari pada List Simpanan Anggota. |
| 1. *Click button* Transaksi untuk melakukan pembayaran Simpanan. |  |
|  | 1. Menampilkan halaman Form Pembayaran Simpanan Anggota. |
| 1. Memasukkan data-data Pembayaran Simpanan sesuai dengan form. |  |
| 1. Memilih Jenis Pembayaran Simpanan. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menampilkan notifikasi bahwa data yang dimasukkan tidak *valid.* |
| 1. Memperbaiki data Pembayaran Simpanan dengan nilai yang *valid.* |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menyimpan data Pembayaran Simpanan ke tabel Bayar Simpanan. |
|  | 1. Menampilkan notifikasi bahwa data telah berhasil disimpan. |
|  | 1. Menampilkan halaman List Simpanan Anggota. |

**Tabel 4.8 *Scenario* *Use Case* Data Simpanan**

1. ***Scenario Use Case* Mengelola Data Pinjaman**

|  |  |
| --- | --- |
| **Reaksi Aktor** | **Reaksi Sistem** |
| **Skenario Normal** | |
| 1. Membuka Aplikasi Koperasi Simpan Pinjam dengan mengetikkan alamat URL yang terkait. |  |
| 1. *Login.* |  |
|  | 1. Menampilkan halaman *Dashboard* aplikasi Koperasi Simpan Pinjam. |
| 1. *Click* Menu Pinjaman. |  |
|  | 1. Menampilkan halaman List Pinjaman Anggota. |
| 1. *Click button* Tambah Pinjaman. |  |
|  | 1. Menampilkan halaman Form Peminjaman Uang. |
| 1. Memasukkan data-data anggota yang bersangkutan beserta nominal dan keterangan peminjaman uang. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menyimpan data Pinjaman ke tabel Pinjaman. |
|  | 1. Menampilkan notifikasi bahwa transaksi peminjaman uang telah berhasil disimpan. |
|  | 1. Menampilkan halaman List Pinjaman Anggota. |
| 1. Melakukan pencarian data transaksi Pinjaman Anggota yang akan diubah. |  |
|  | 1. Menampilkan data Pinjaman Anggota yang dicari pada List Pinjaman Anggota. |
| 1. *Click button* Edit untuk melakukan perubahan pada data Pinjaman Anggota. |  |
|  | 1. Menampilkan halaman Form Edit Peminjaman Uang. |
| 1. Mengubah data Anggota atau data transaksi Pinjaman Anggota. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah diubah. |
|  | 1. Menyimpan data Peminjaman yang telah diubah ke tabel Pinjaman. |
|  | 1. Menampilkan notifikasi bahwa data transaksi Peminjaman telah berhasil diubah. |
|  | 1. Menampilkan halaman List Peminjaman. |
| 1. Melakukan pencarian data transaksi Pinjaman Anggota yang akan hapus. |  |
|  | 1. Menampilkan data Pinjaman Anggota yang dicari pada List Pinjaman Anggota. |
| 1. *Click button* Hapus untuk menghapus data transaksi Pinjaman Anggota. |  |
|  | 1. Menampilkan pesan konfirmasi hapus data. |
| 1. *Click button* Hapus. |  |
|  | 1. Menampilkan notifikasi bahwa data berhasil dihapus. |
|  | 1. Melakukan *refresh* terhadap halaman List Pinjaman Anggota. |
| 1. Melakukan pencarian data transaksi Pinjaman Anggota. |  |
|  | 1. Menampilkan data Pinjaman Anggota yang dicari pada List Pinjaman Anggota. |
| 1. *Click button* Detail. |  |
|  | 1. Menampilkan halaman Form Detail Pinjaman Anggota. |
| **Skenario Alternatif** | |
| 1. Membuka Aplikasi Koperasi Simpan Pinjam dengan mengetikkan alamat URL yang terkait. |  |
| 1. *Login.* |  |
|  | 1. Menampilkan halaman *Dashboard* aplikasi Koperasi Simpan Pinjam. |
| 1. *Click* Menu Pinjaman. |  |
|  | 1. Menampilkan halaman List Pinjaman Anggota. |
| 1. *Click button* Tambah Pinjaman. |  |
|  | 1. Menampilkan halaman Form Peminjaman Uang. |
| 1. Memasukkan data-data anggota yang bersangkutan beserta nominal dan keterangan peminjaman uang. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menampilkan notifikasi bahwa data yang disimpan tidak *valid*. |
| 1. Memperbaiki data Pinjaman dengan nilai yang *valid.* |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah dimasukkan. |
|  | 1. Menyimpan data Pinjaman Anggota ke tabel Pinjaman. |
|  | 1. Menampilkan notifikasi bahwa data telah berhasil disimpan. |
|  | 1. Menampilkan halaman List Pinjaman Anggota. |
| 1. Melakukan pencarian data transaksi Pinjaman Anggota yang akan diubah. |  |
|  | 1. Menampilkan data Pinjaman Anggota yang dicari pada List Pinjaman Anggota. |
| 1. *Click button* Edit untuk melakukan perubahan pada data Pinjaman Anggota. |  |
|  | 1. Menampilkan halaman Form Edit Peminjaman Uang. |
| 1. Mengubah data Anggota atau data transaksi Pinjaman Anggota. |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah diubah. |
|  | 1. Menampilkan notifikasi bahwa data transaksi Peminjaman yang diubah tidak *valid*. |
| 1. Memperbaiki data Anggota atau transaksi Pinjaman dengan nilai yang *valid.* |  |
|  | 1. Memeriksa *valid* atau tidaknya data yang telah diubah. |
|  | 1. Menyimpan data Pinjaman yang telah diubah ke tabel Pinjaman. |
|  | 1. Menampilkan notifikasi bahwa data telah berhasil diubah. |
|  | 1. Menampilkan halaman List Pinjaman Anggota. |
| 1. Melakukan pencarian data transaksi Pinjaman Anggota yang akan dihapus. |  |
|  | 1. Menampilkan data Pinjaman Anggota yang dicari pada List Pinjaman Anggota. |
| 1. *Click button* Hapus untuk menghapus data transaksi Pinjaman Anggota. |  |
|  | 1. Menampilkan pesan konfirmasi hapus data. |
| 1. *Click button Cancel*. |  |
|  | 1. Melakukan *refresh* terhadap halaman List Pinjaman Anggota. |

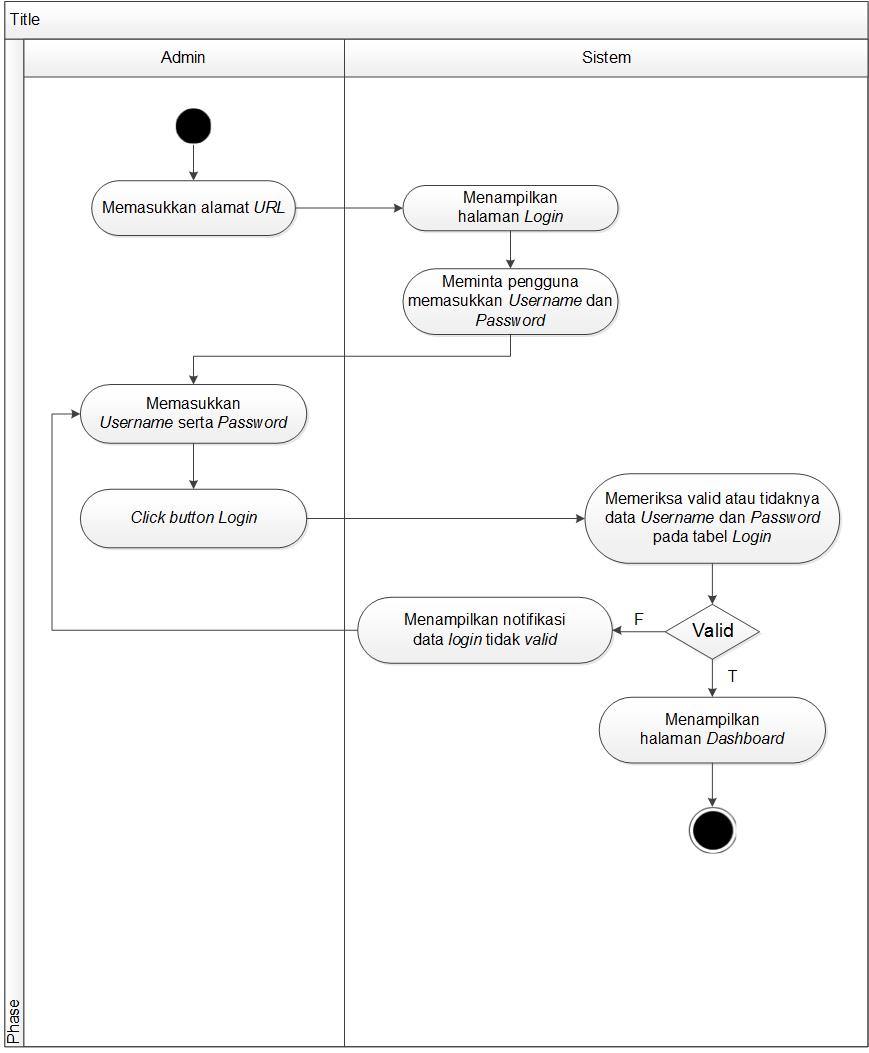
**Tabel 4.10 *Scenario* *Use Case* Mengelola Data Pinjaman**

1. ***Scenario Use Case* Mengelola Data Angsuran Pinjaman**

|  |  |
| --- | --- |
| **Reaksi Aktor** | **Reaksi Sistem** |
| **Skenario Normal** | |
| 1. Membuka Aplikasi Koperasi Simpan Pinjam dengan mengetikkan alamat *URL* yang terkait. |  |
| 1. *Login.* |  |
|  | 1. Menampilkan halaman *Dashboard* aplikasi Koperasi Simpan Pinjam. |
| 1. *Click* Menu Angsuran Pinjaman. |  |
|  | 1. Menampilkan halaman List Pinjaman Anggota. |
| 1. Melakukan pencarian data Pinjaman Anggota. |  |
|  | 1. Menampilkan data transaksi Pinjaman Anggota. |
| 1. *Click* button Detail Pinjaman. |  |
|  | 1. Menampilkan halaman List Detail Pinjaman Anggota. |
| 1. Memilih data Cicilan Pembayaran Pinjaman Anggota yang akan dibayarkan. |  |
| 1. *Click button* Pembayaran. |  |
|  | 1. Menampilkan Form Pembayaran Angsuran Pinjaman. |
| 1. Memasukkan nominal pembayaran angsuran pinjaman. |  |
| 1. *Click button* Bayar. |  |
|  | 1. Menyimpan data Pinjaman ke tabel Pinjaman. |
|  | 1. Menampilkan notifikasi transaksi pembayaran angsuran pinjaman berhasil disimpan. |
|  | 1. Menampilkan halaman List Detail Pinjaman Anggota. |

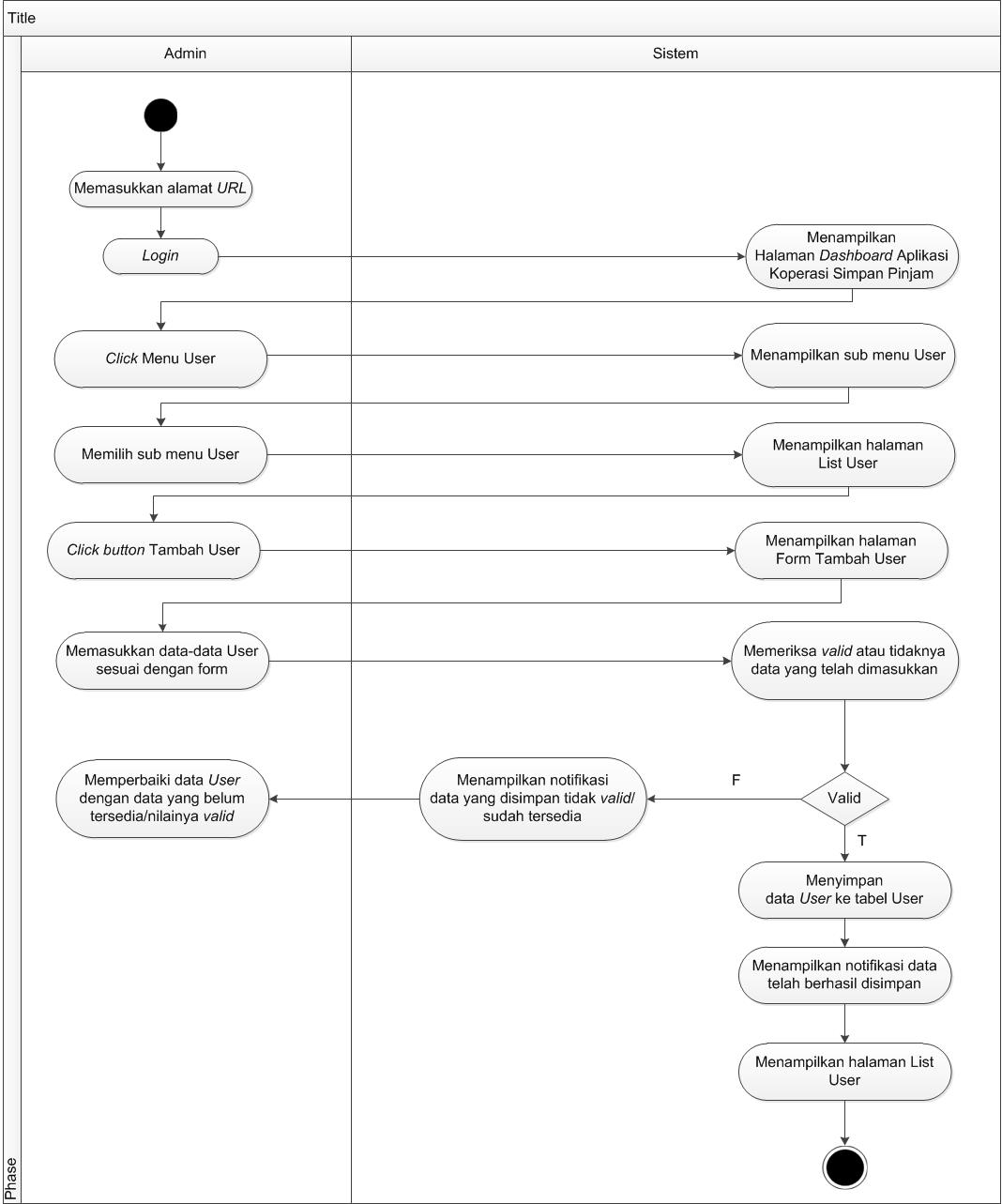
**Tabel 4.11 *Scenario* *Use Case* Mengelola Data Angsuran Pinjaman**

1. ***Activity Diagram***
   1. ***Activity Diagram Login***



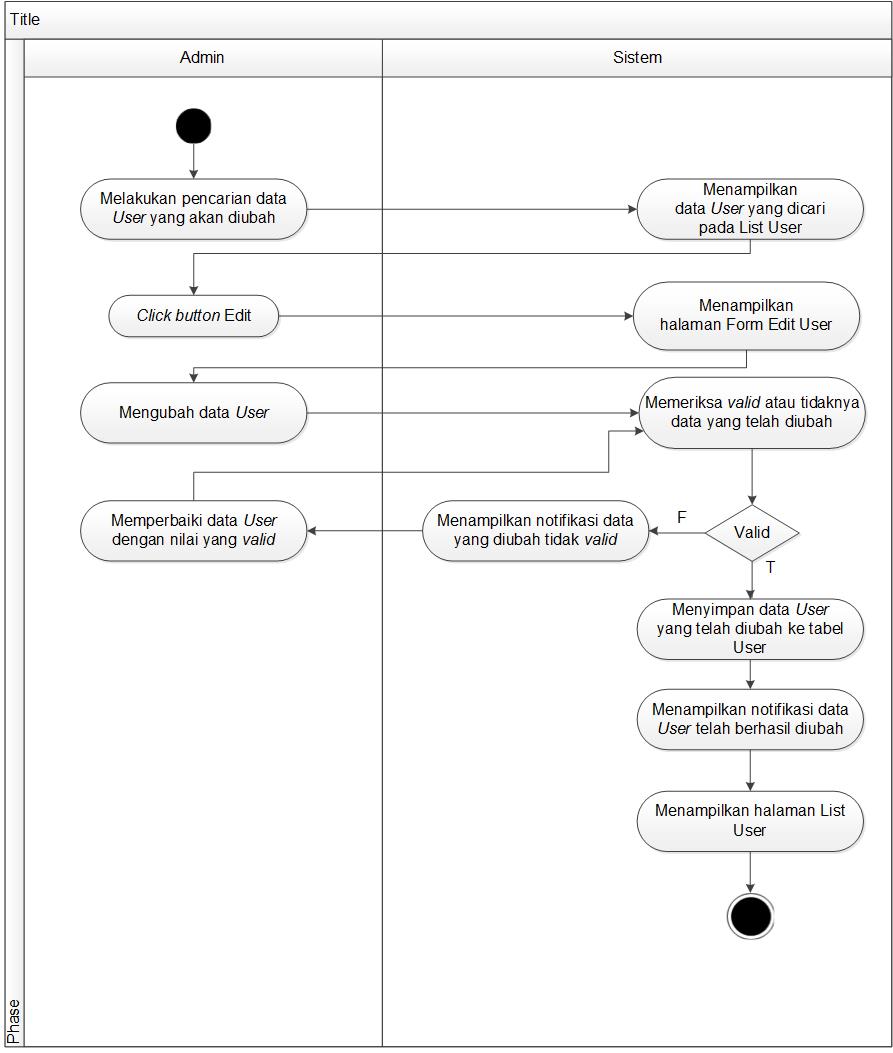
**Gambar 4.8 *Activity Diagram Login***

* 1. ***Activity Diagram* Tambah Data *User***



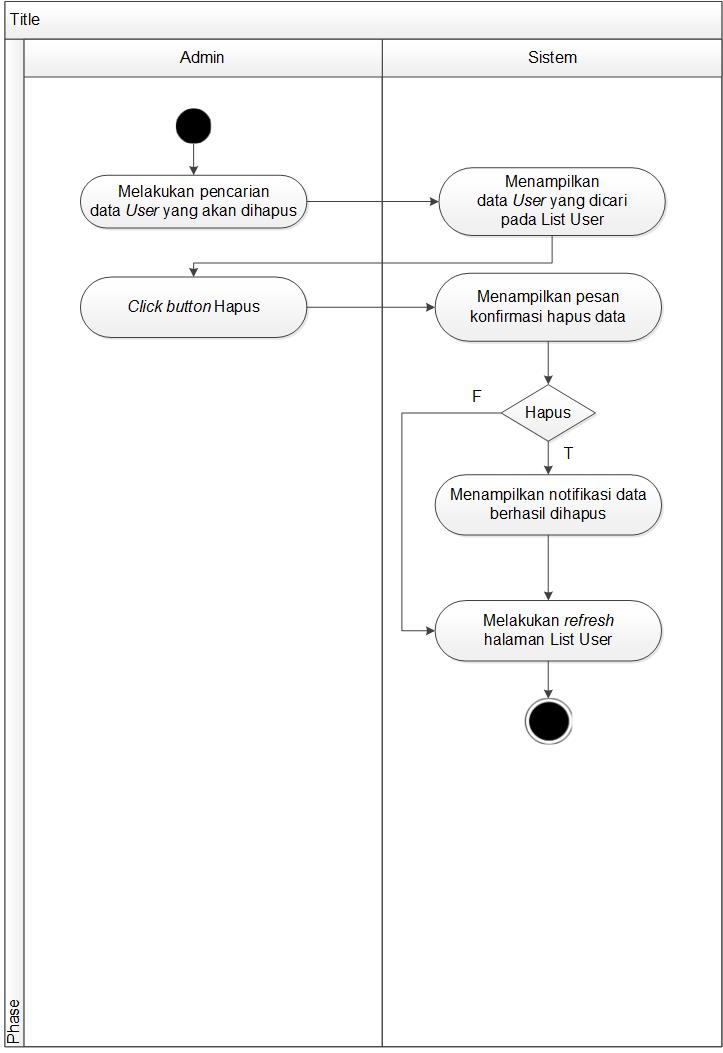
**Gambar 4.9 *Activity Diagram* Tambah Data *User***

* 1. ***Activity Diagram* Edit Data *User***



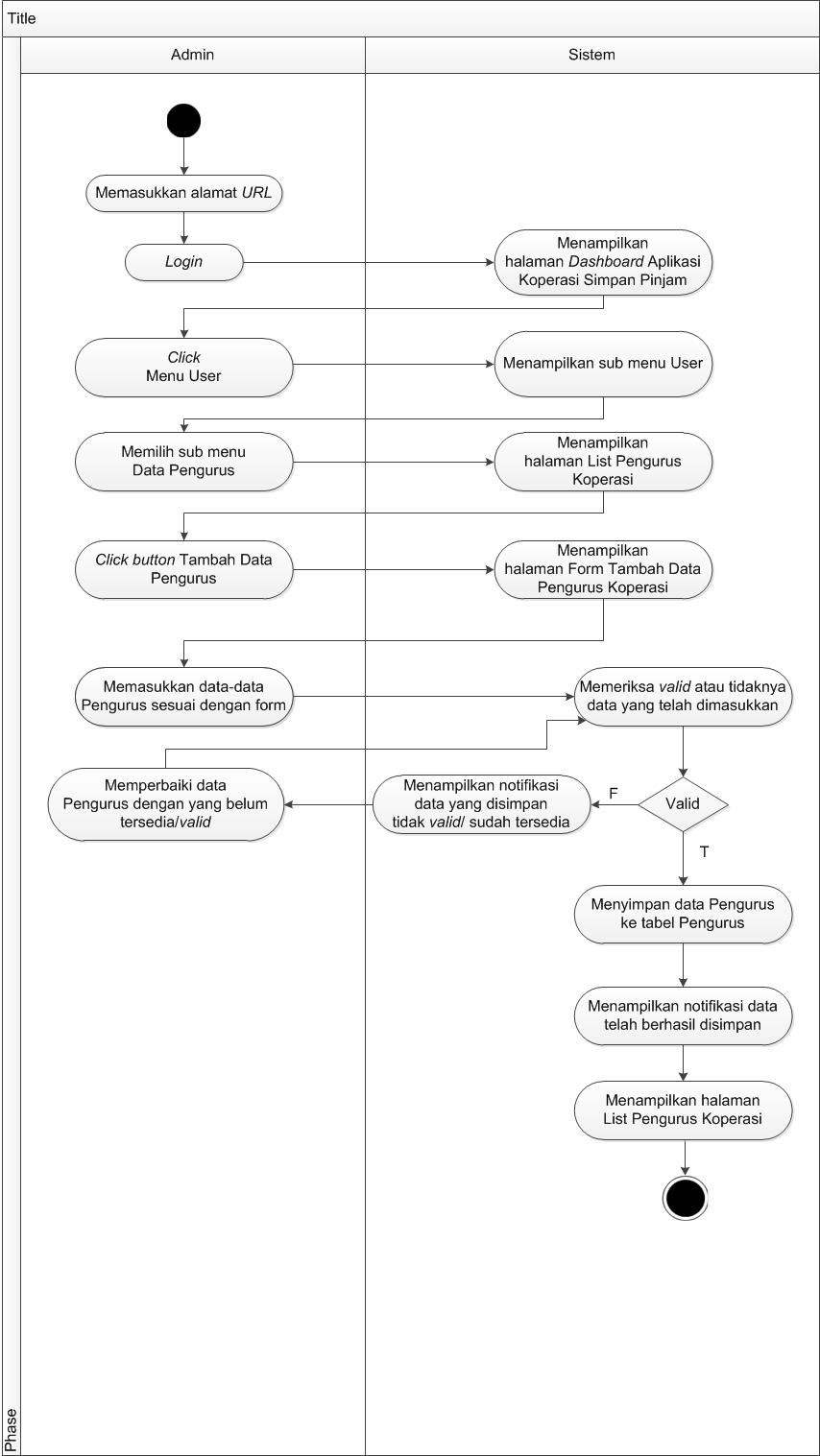
**Gambar 4.10 *Activity Diagram* Edit Data *User***

* 1. ***Activity Diagram* Hapus Data *User***



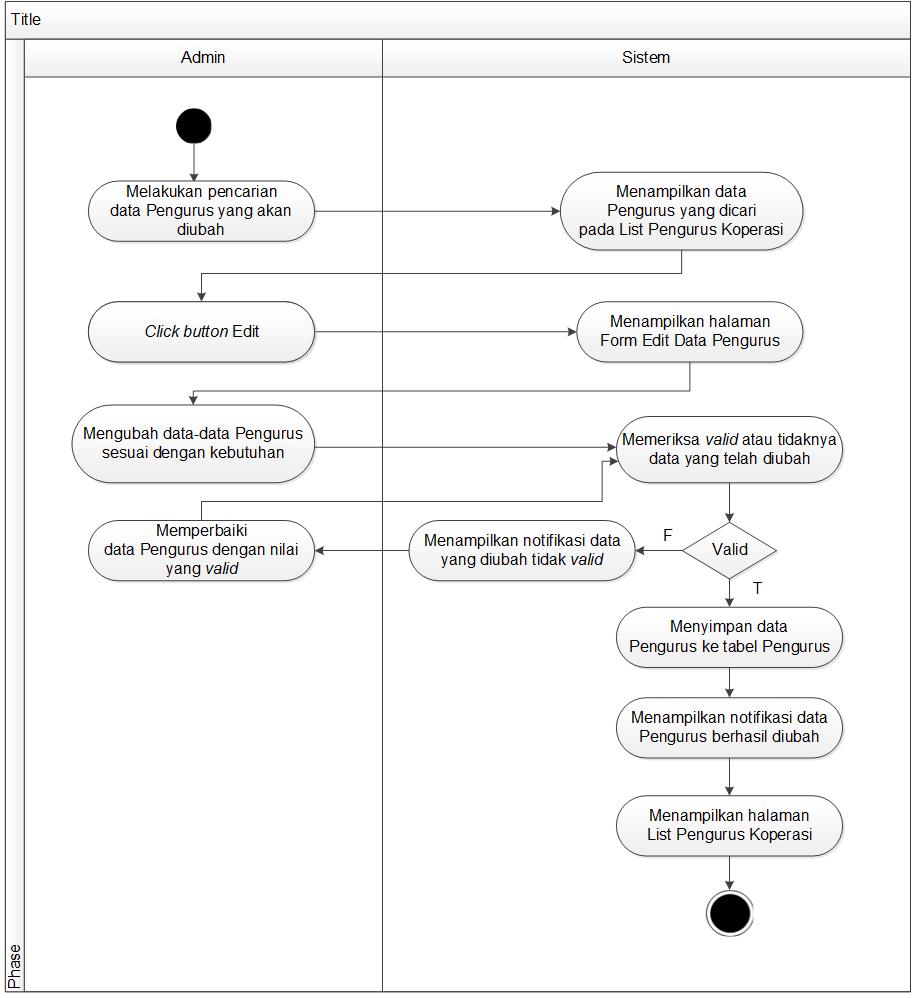
**Gambar 4.11 *Activity Diagram* Hapus Data *User***

* 1. ***Activity Diagram* Tambah Data Pengurus Koperasi**



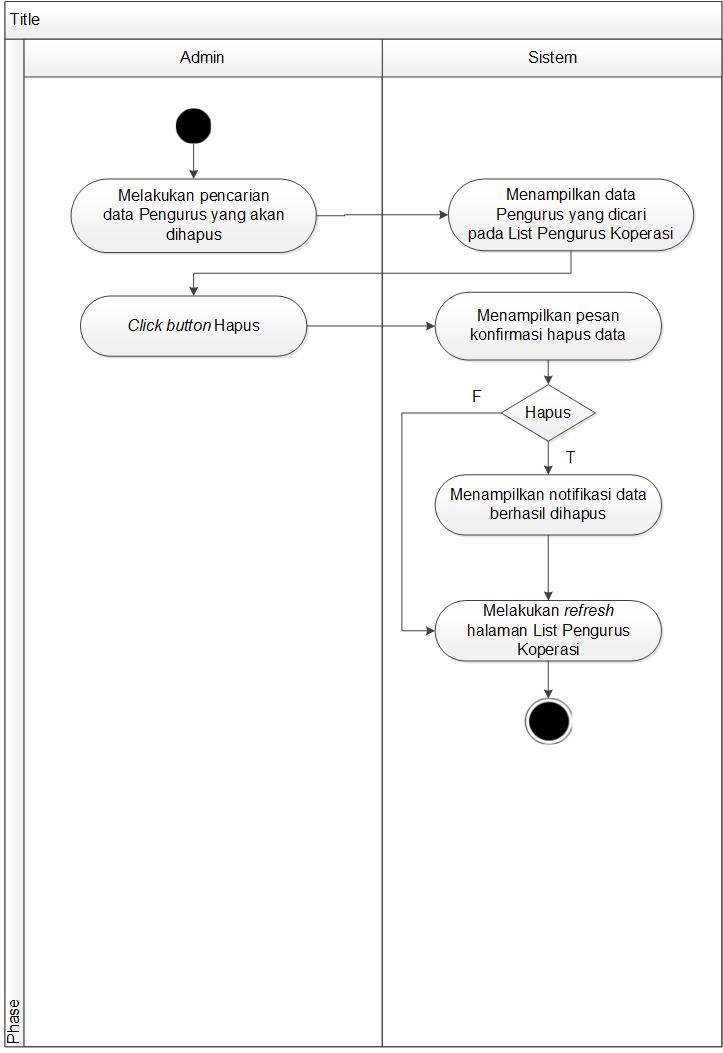
**Gambar 4.12 *Activity Diagram* Tambah Data Pengurus Koperasi**

* 1. ***Activity Diagram* Edit Data Pengurus Koperasi**



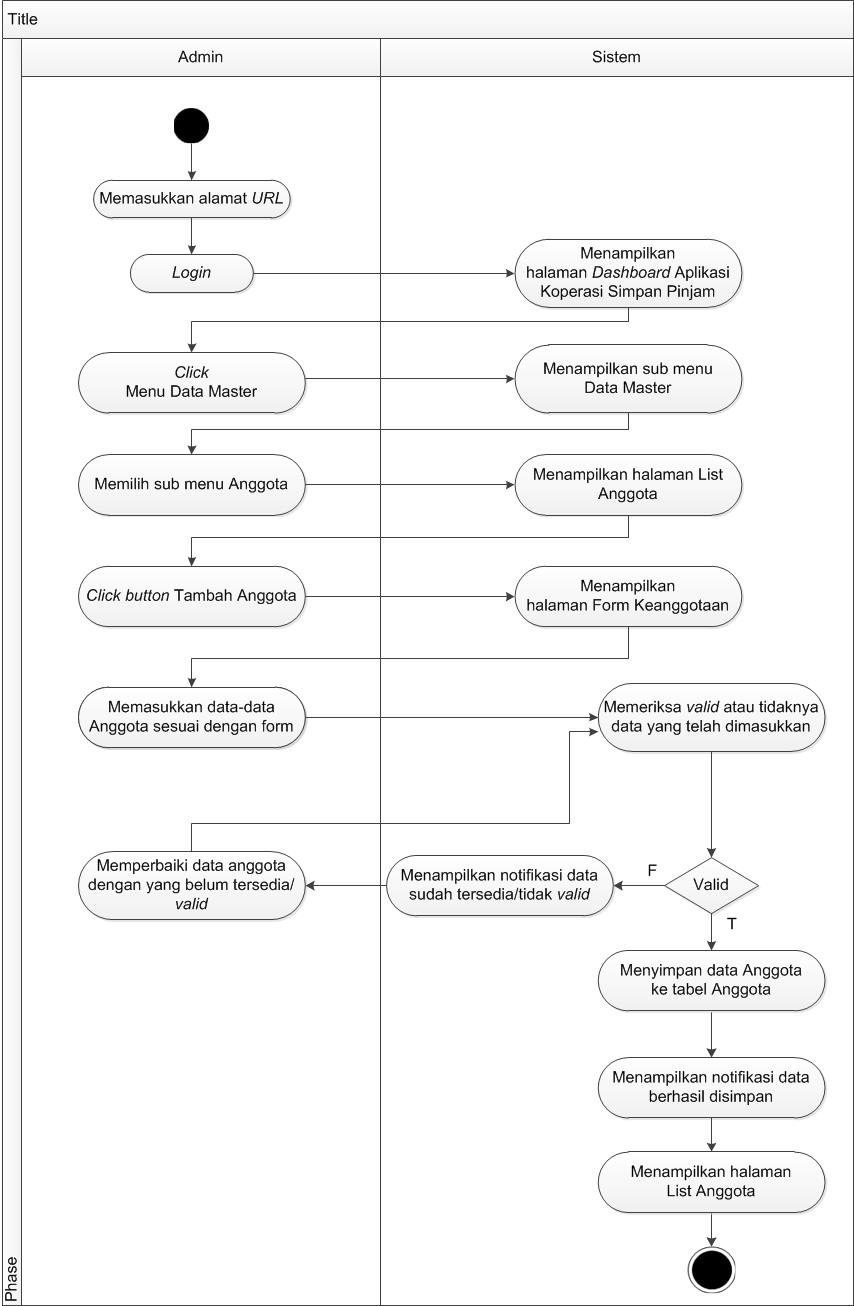
**Gambar 4.13 *Activity Diagram* Edit Data Pengurus Koperasi**

* 1. ***Activity Diagram* Hapus Data Pengurus Koperasi**



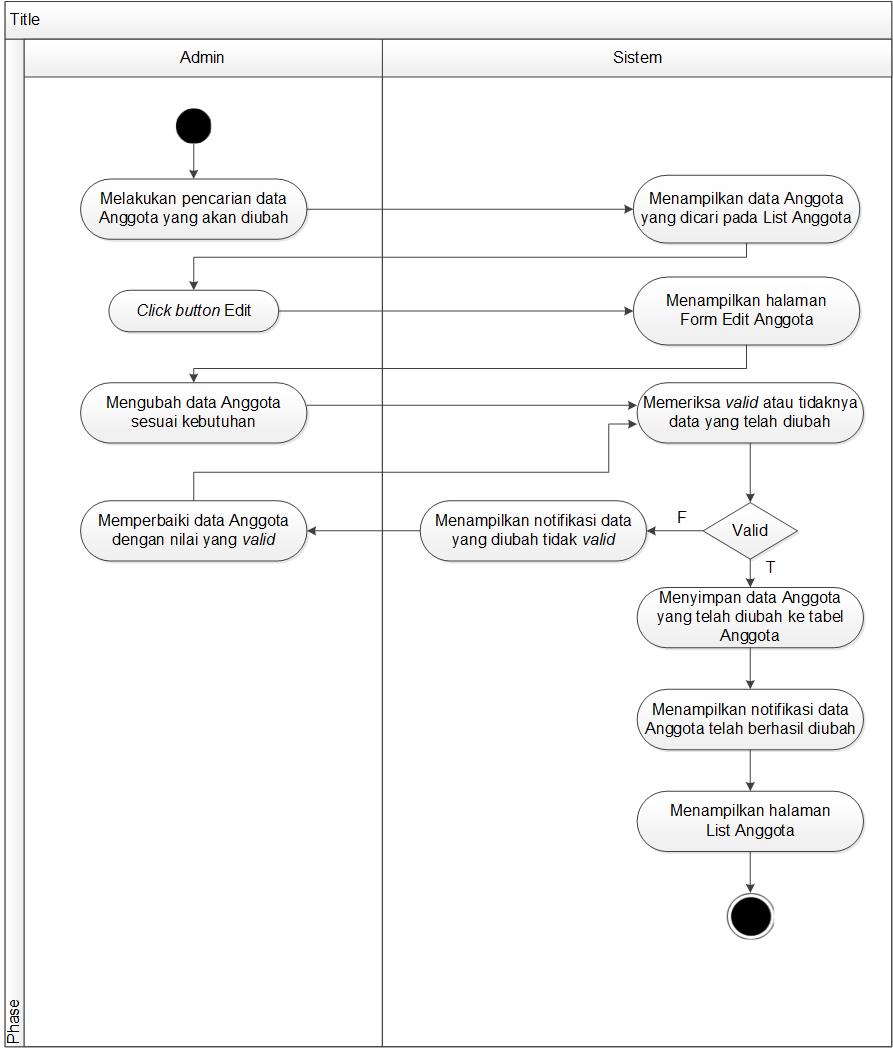
**Gambar 4.14 *Activity Diagram* Hapus Data Pengurus Koperasi**

* 1. ***Activity Diagram* Tambah Data Anggota**



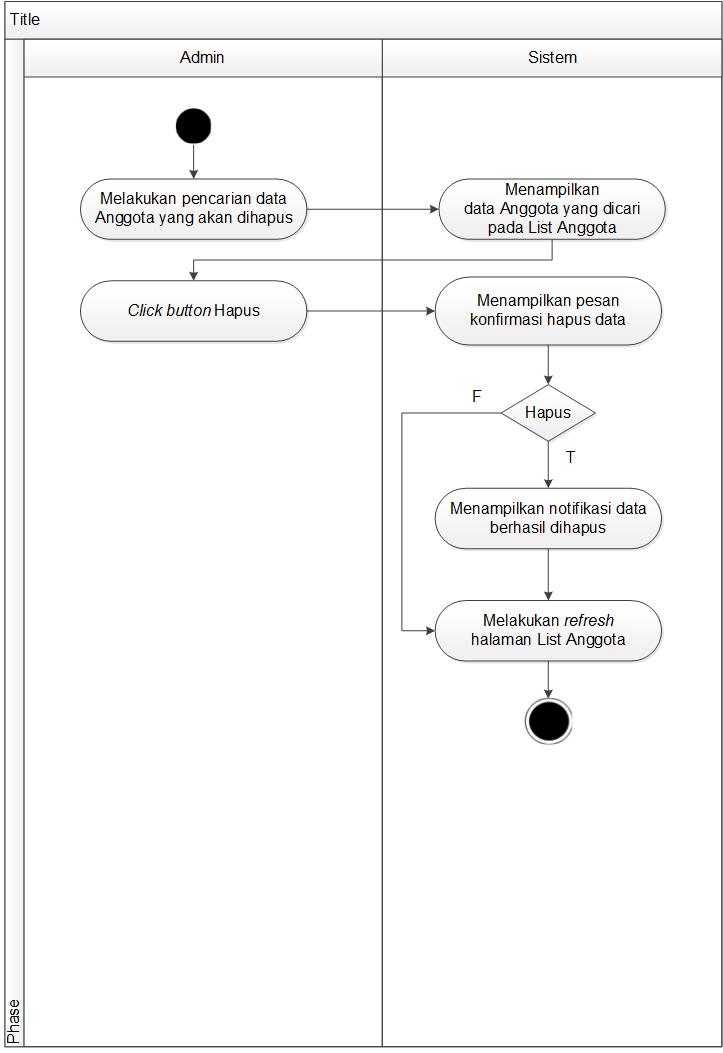
**Gambar 4.15 *Activity Diagram* Tambah Data Anggota**

* 1. ***Activity Diagram* Edit Data Anggota**

****

**Gambar 4.16 *Activity Diagram* Edit Data Anggota**

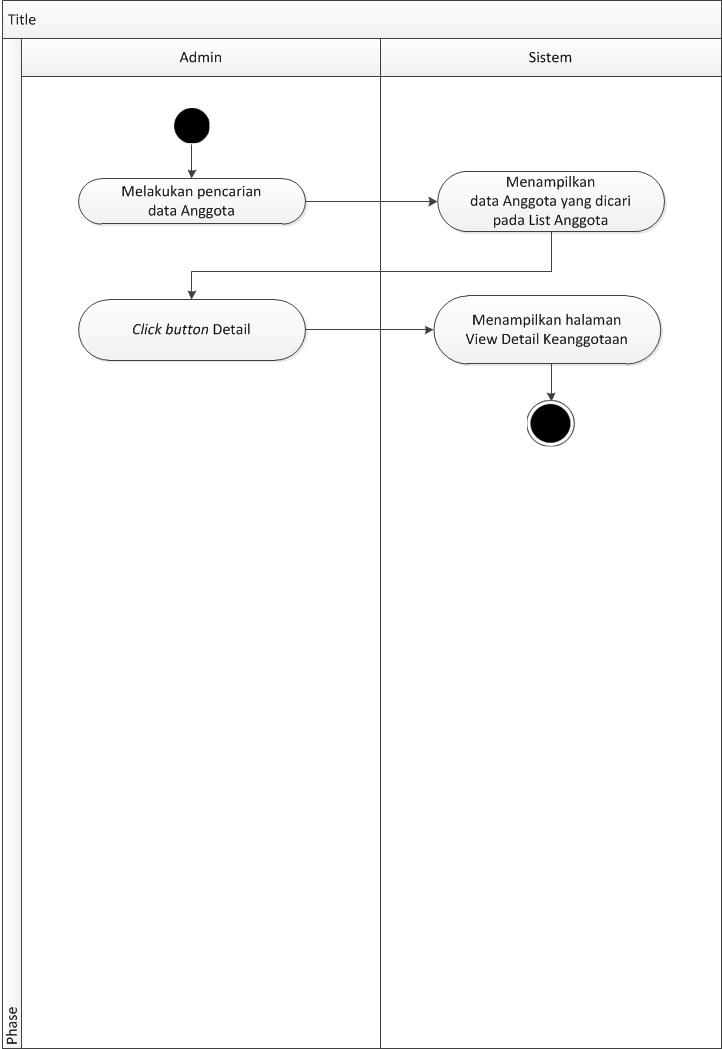
* 1. ***Activity Diagram* Hapus Data Anggota**

****

**Gambar 4.17 *Activity Diagram* Hapus Data Anggota**

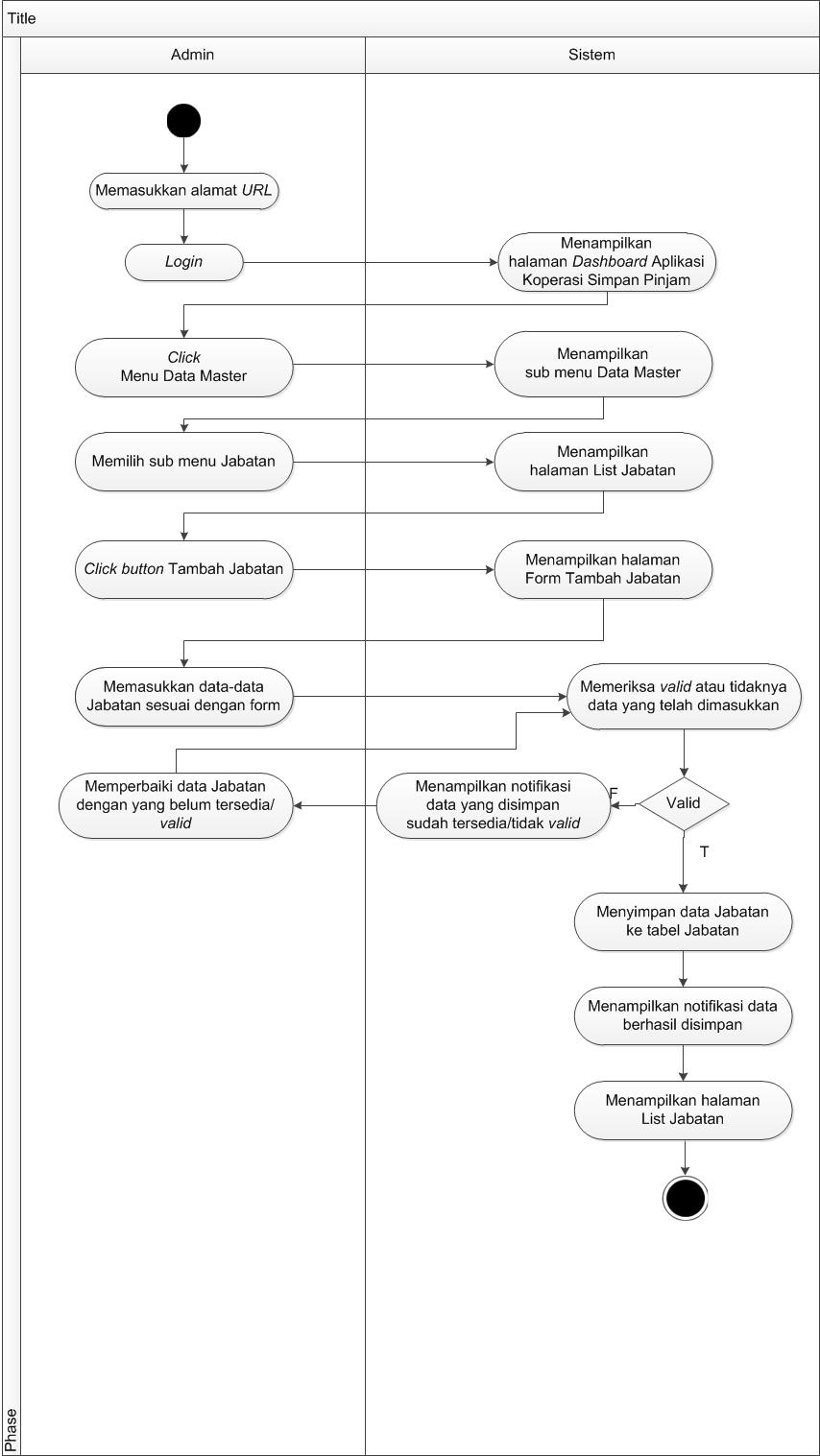
**081317628562**

* 1. ***Activity Diagram* Detail Data Anggota**



**Gambar 4.18 *Activity Diagram* Detail Data Anggota**

* 1. ***Activity Diagram* Tambah Data Jabatan**



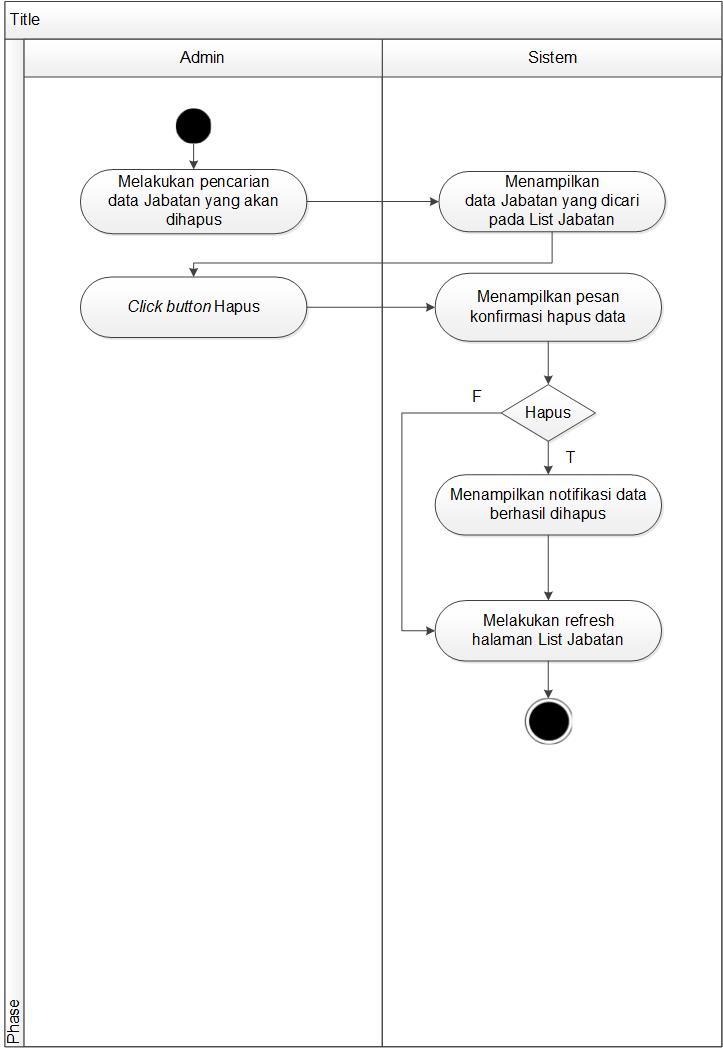
**Gambar 4.19 *Activity Diagram* Tambah Data Jabatan**

* 1. ***Activity Diagram* Edit Data Jabatan**



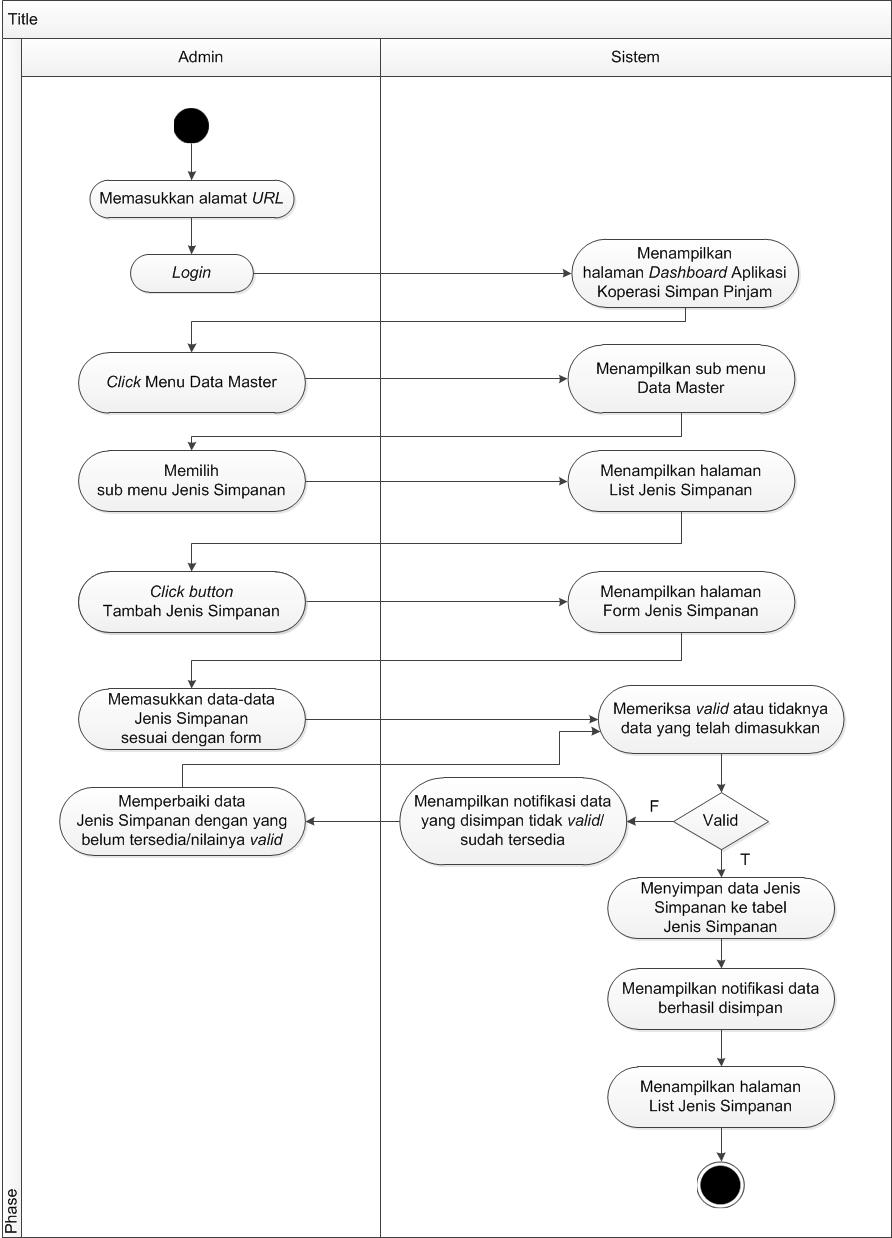
**Gambar 4.20 *Activity Diagram* Edit Data Jabatan**

* 1. ***Activity Diagram* Hapus Data Jabatan**



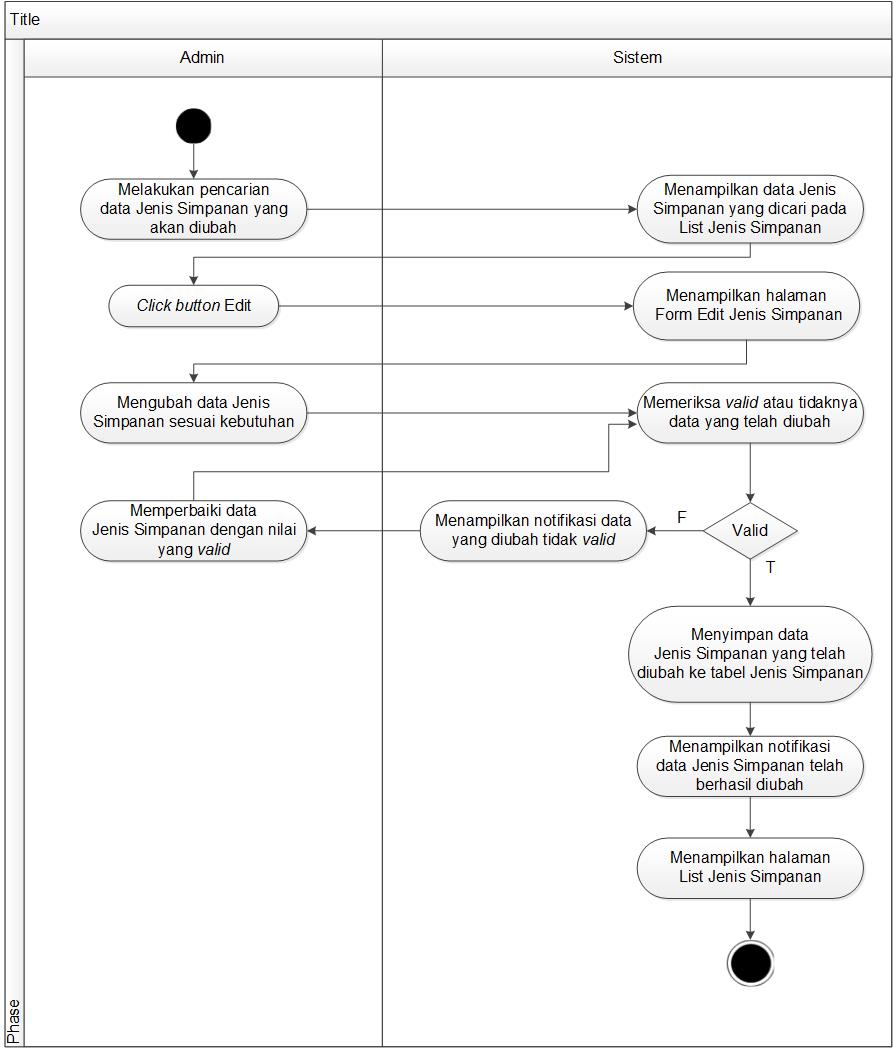
**Gambar 4.21 *Activity Diagram* Hapus Data Jabatan**

* 1. ***Activity Diagram* Tambah Data Jenis Simpanan**



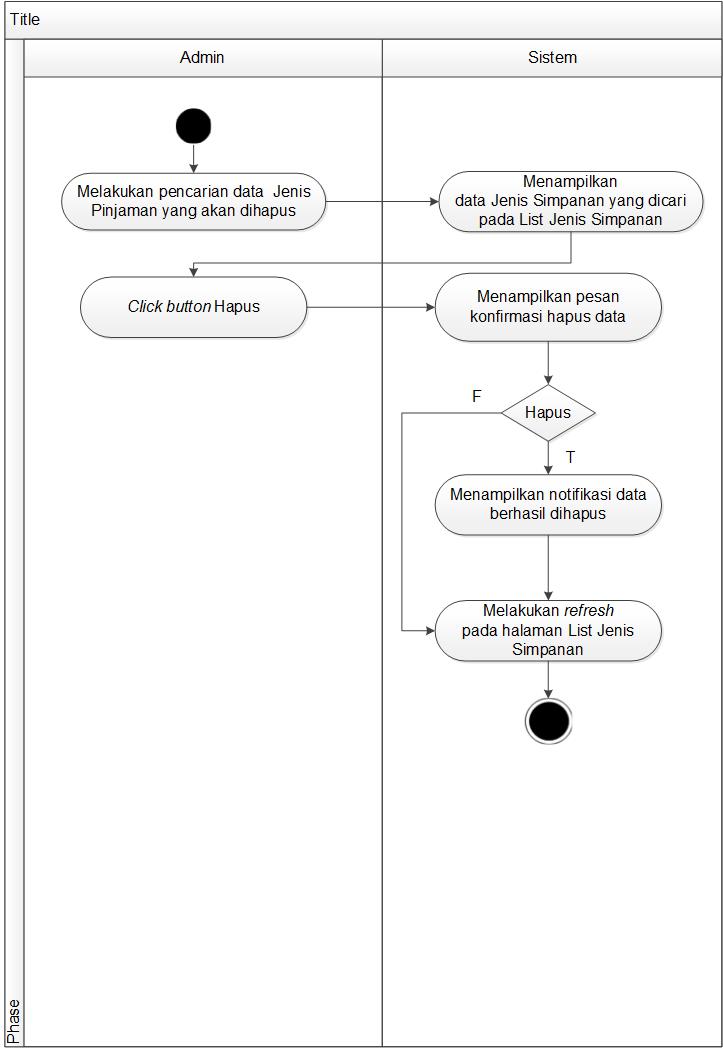
**Gambar 4.22 *Activity Diagram* Tambah Data Jenis Simpanan**

* 1. ***Activity Diagram* Edit Data Jenis Simpanan**



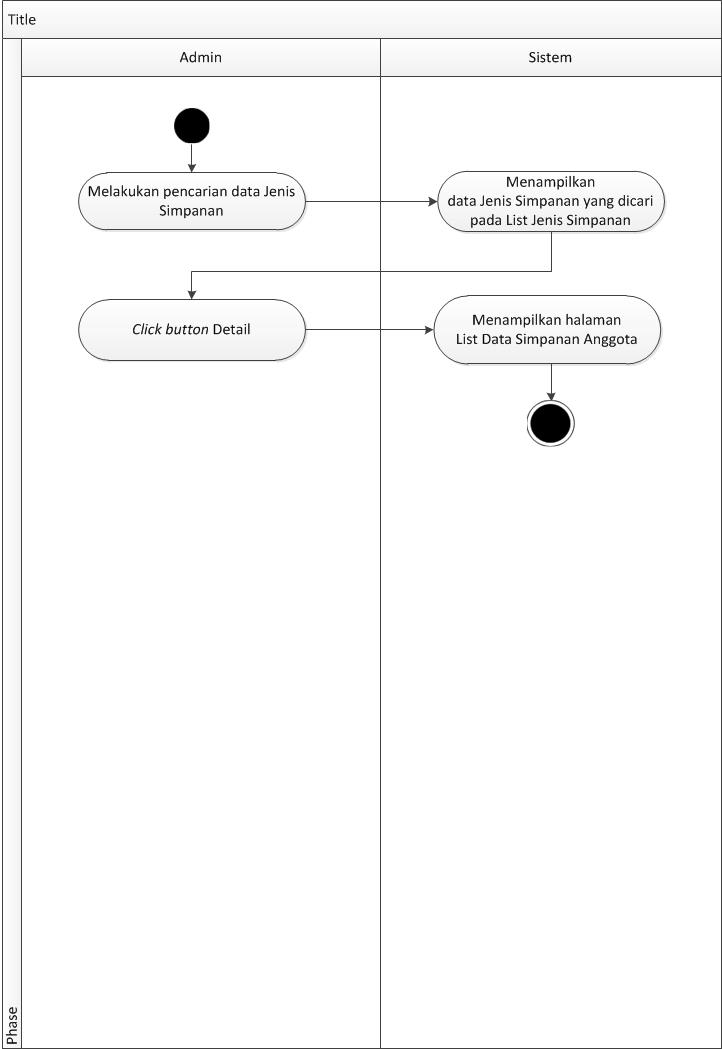
**Gambar 4.23 *Activity Diagram* Edit Data Jenis Simpanan**

* 1. ***Activity Diagram* Hapus Data Jenis Simpanan**



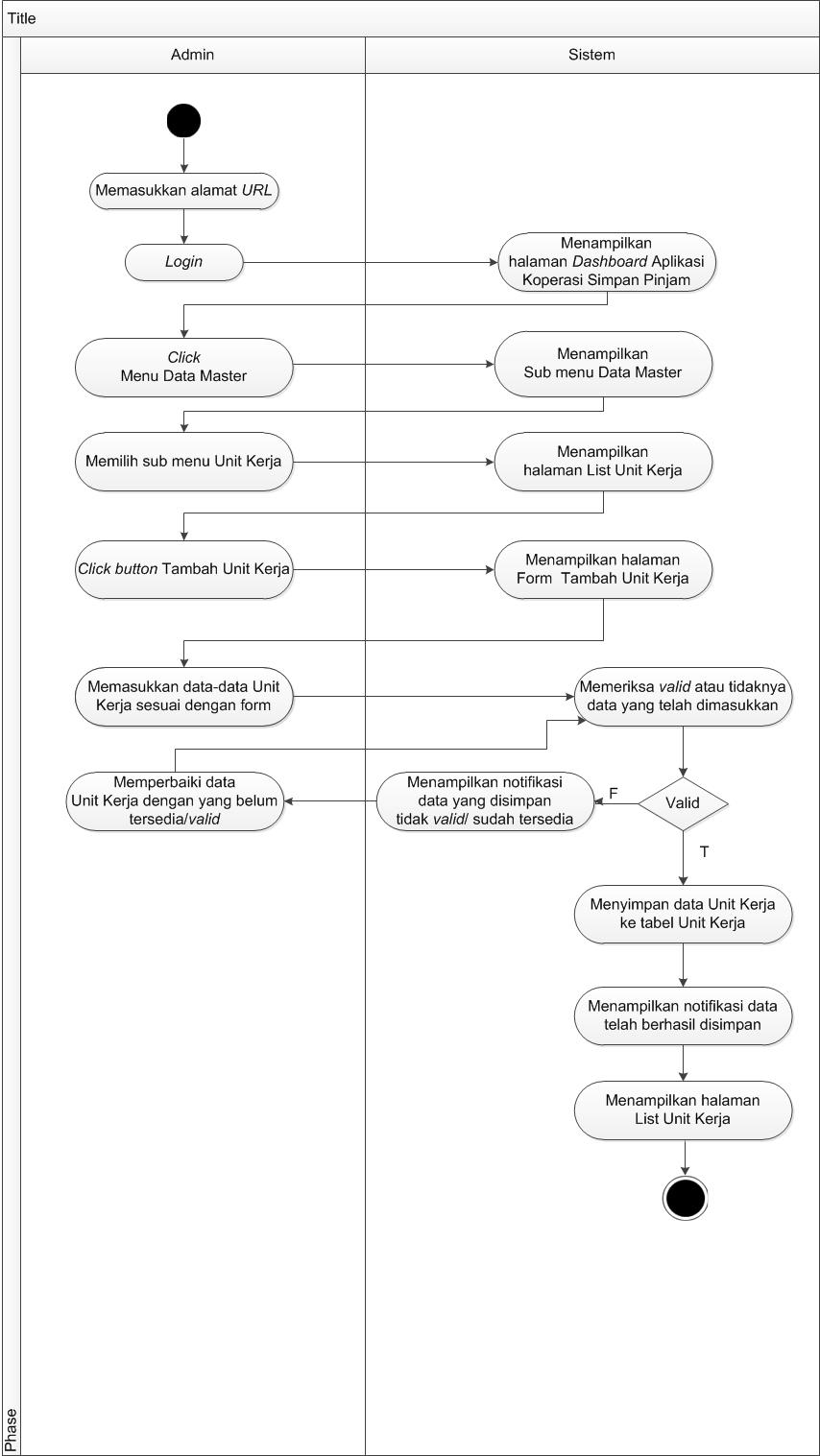
**Gambar 4.24 *Activity Diagram* Hapus Data Jenis Simpanan**

* 1. ***Activity Diagram* Detail Data Jenis Simpanan**



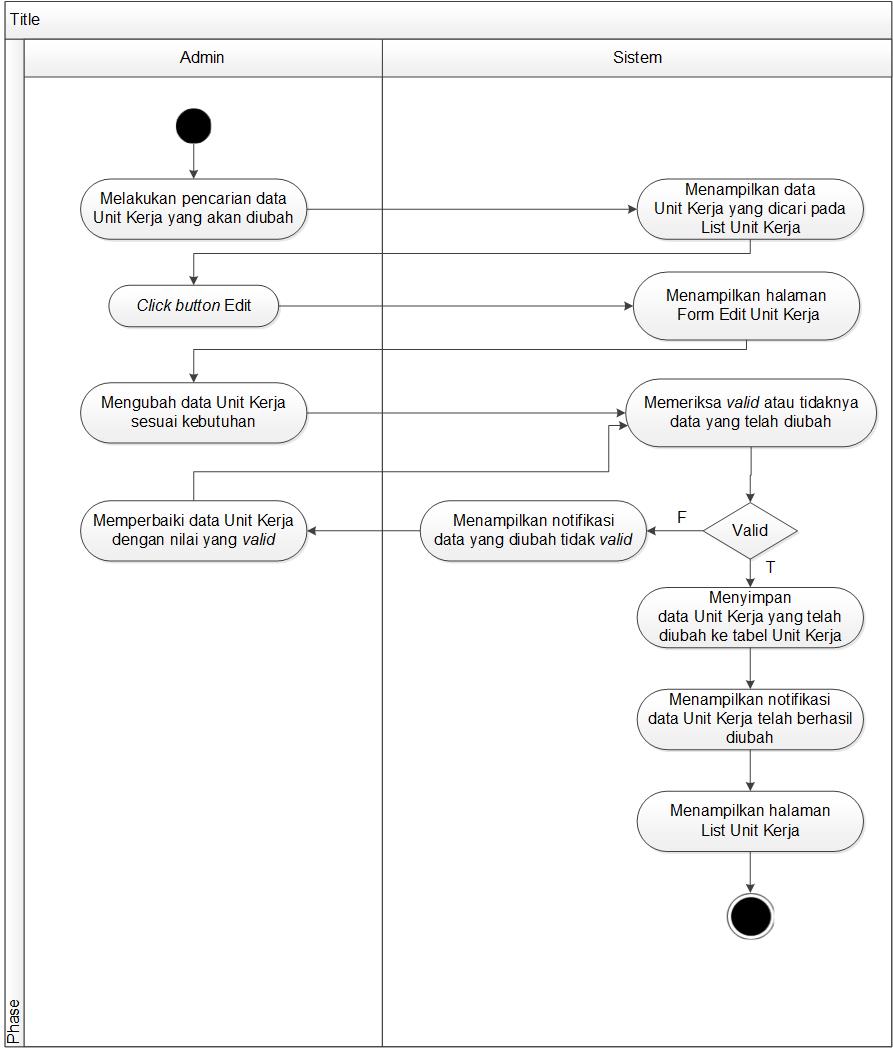
**Gambar 4.25 *Activity Diagram* Detail Data Jenis Simpanan**

* 1. ***Activity Diagram* Tambah Data Unit Kerja**



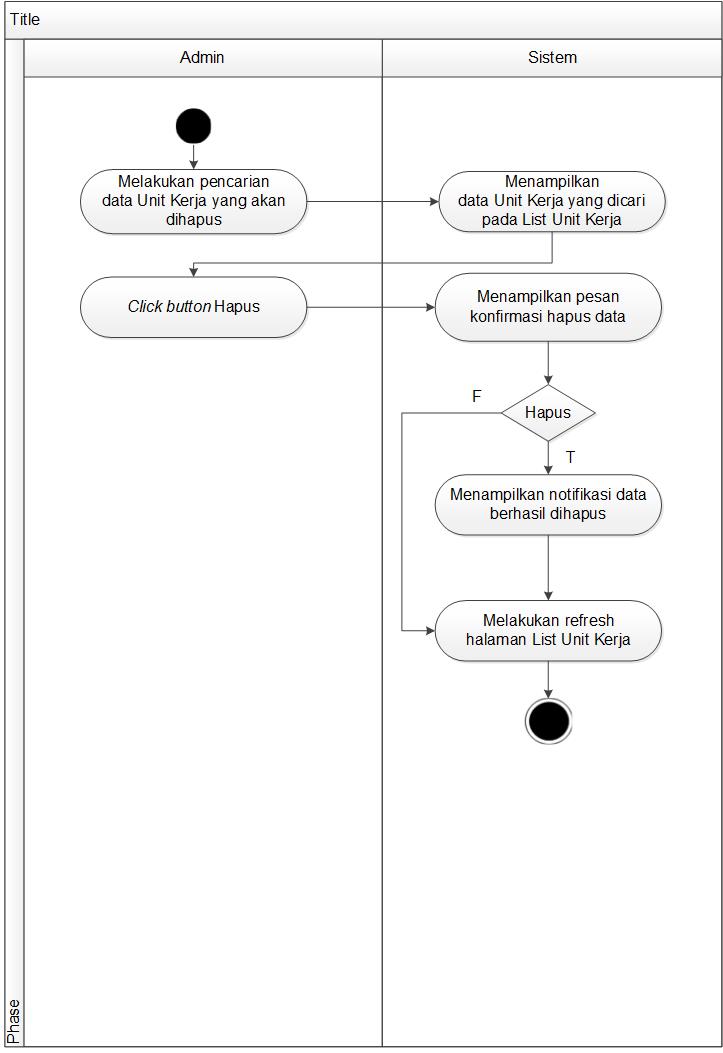
**Gambar 4.26 *Activity Diagram* Tambah Data Unit Kerja**

* 1. ***Activity Diagram* Edit Data Unit Kerja**



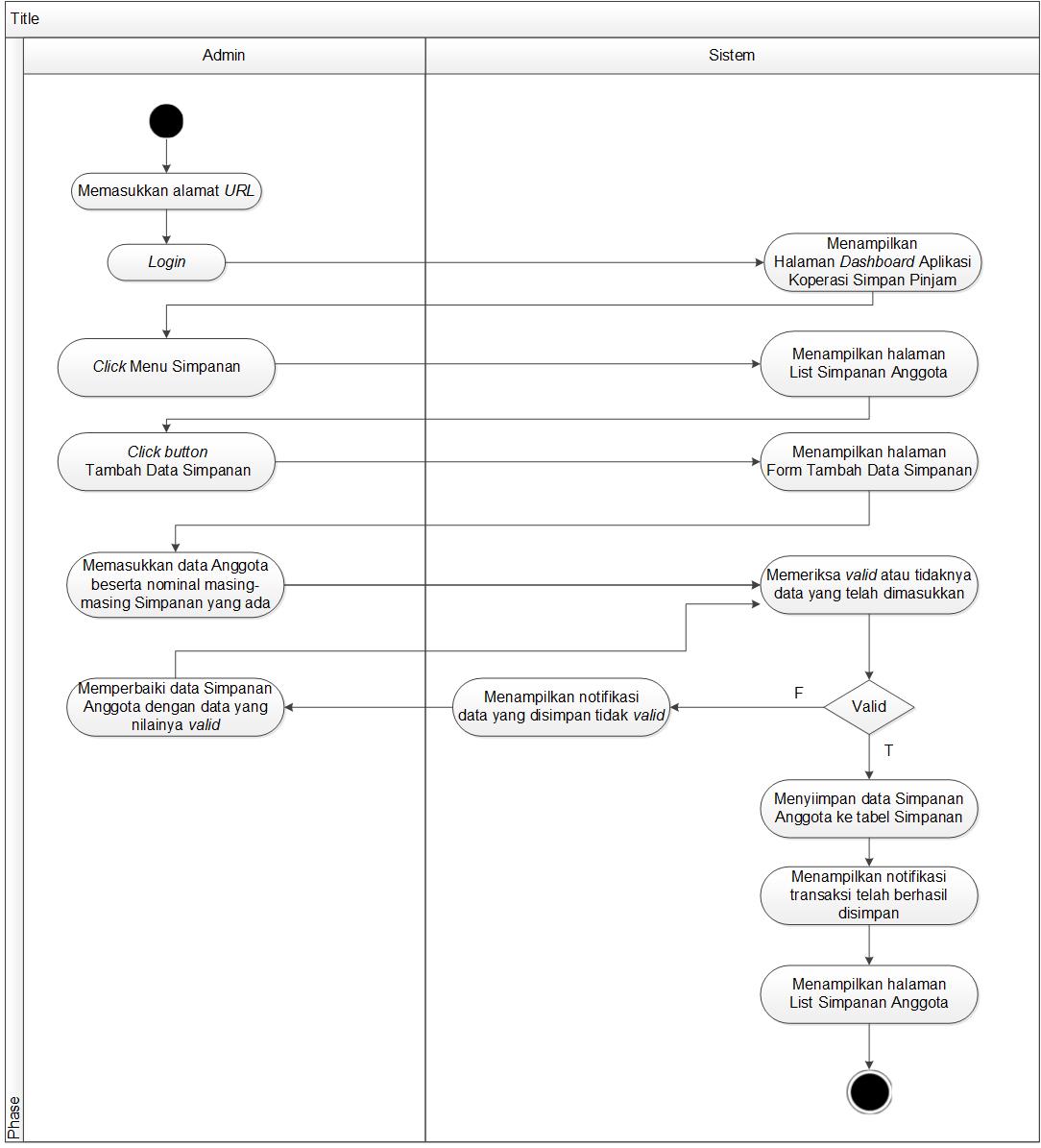
**Gambar 4.27 *Activity Diagram* Edit Data Unit Kerja**

* 1. ***Activity Diagram* Hapus Data Unit Kerja**



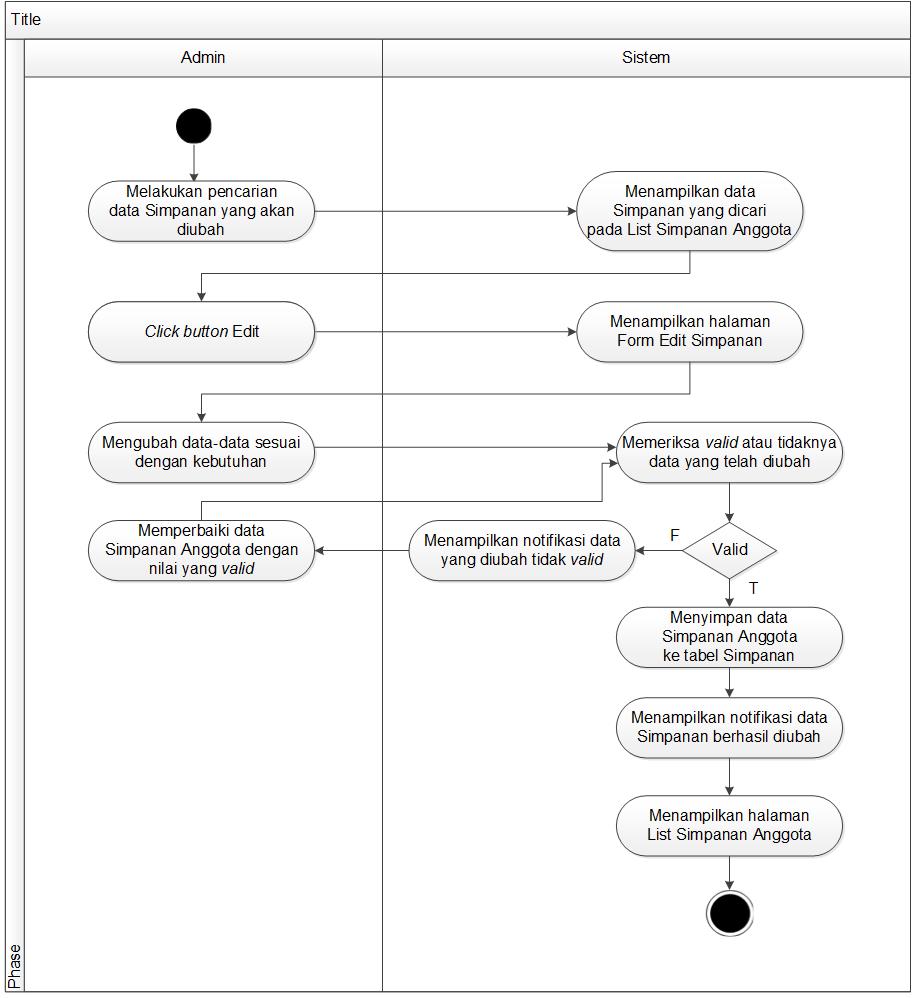
**Gambar 4.28 *Activity Diagram* Hapus Data Unit Kerja**

* 1. ***Activity Diagram* Tambah Data Simpanan**



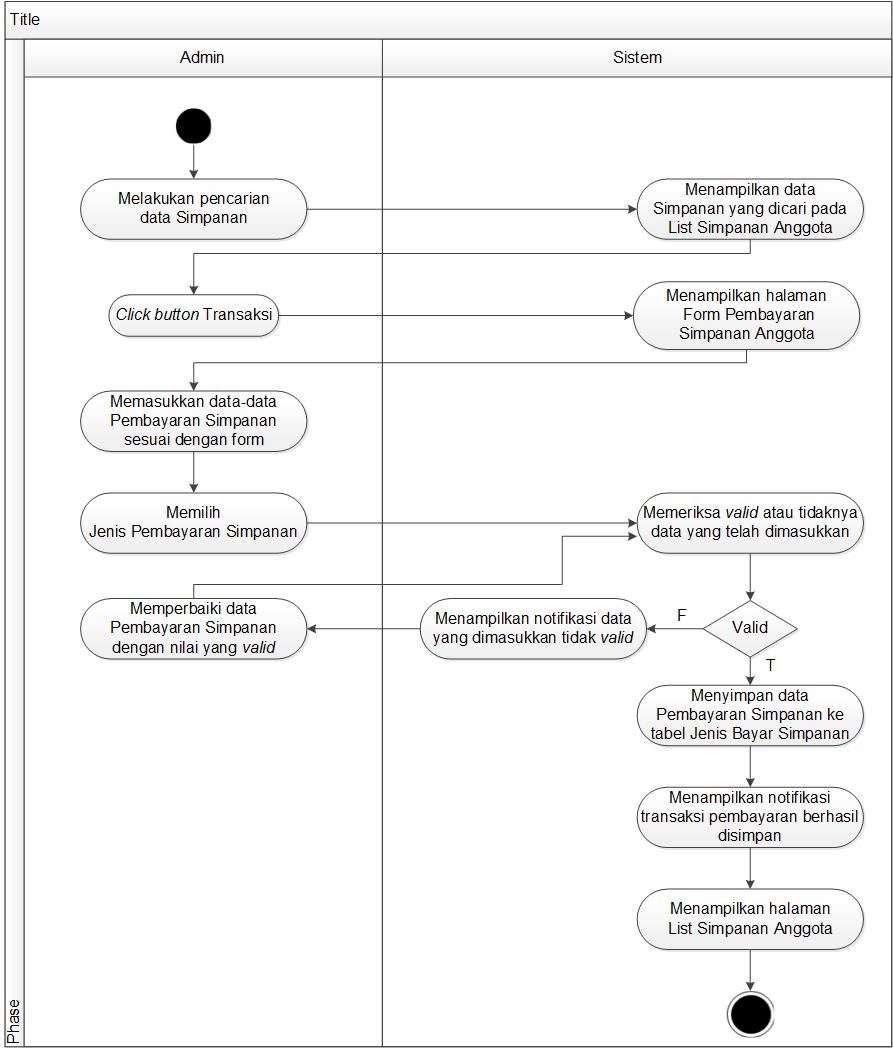
**Gambar 4.29 *Activity Diagram* Tambah Data Simpanan**

* 1. ***Activity Diagram* Edit Data Simpanan**



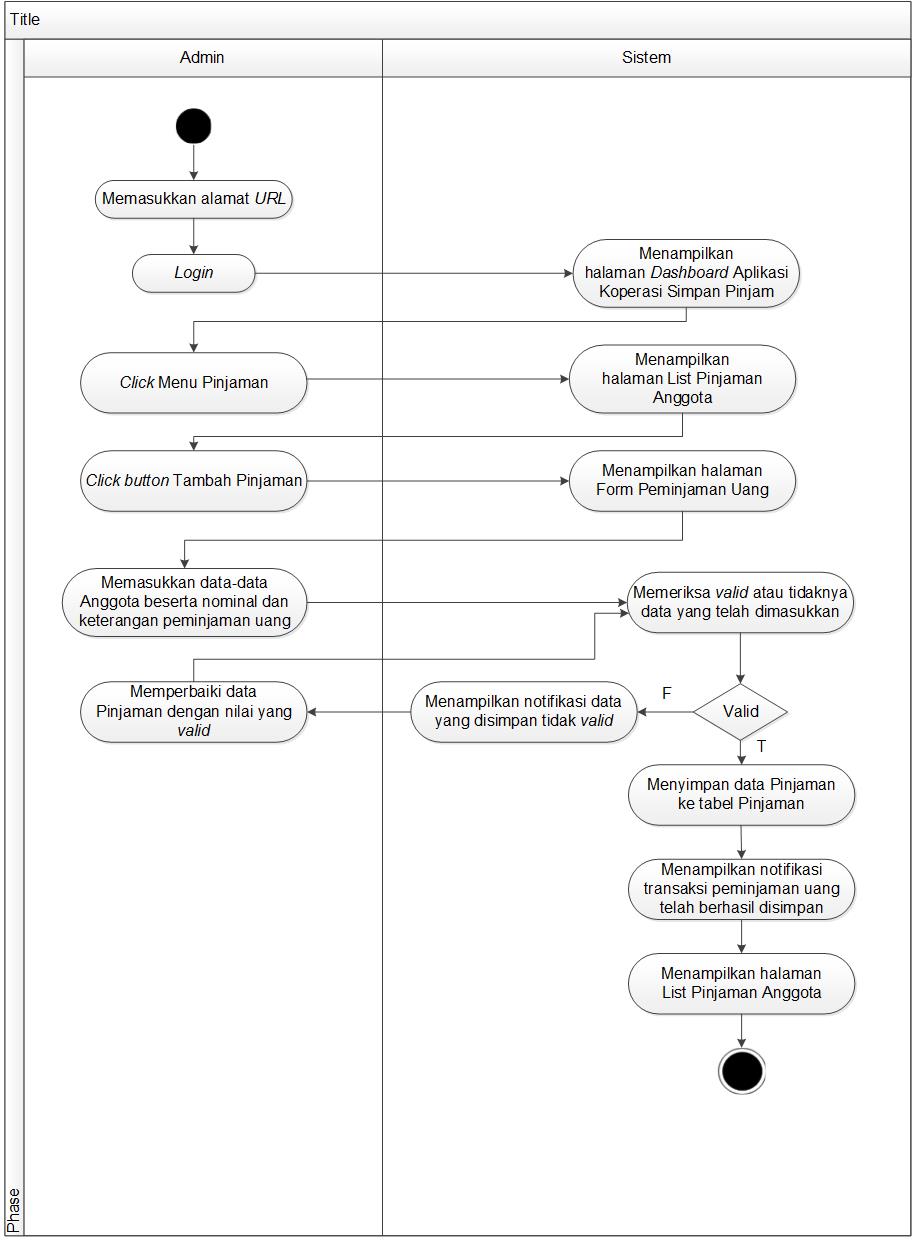
**Gambar 4.30 *Activity Diagram* Edit Data Simpanan**

* 1. ***Activity Diagram* Transaksi Simpanan (Pembayaran)**



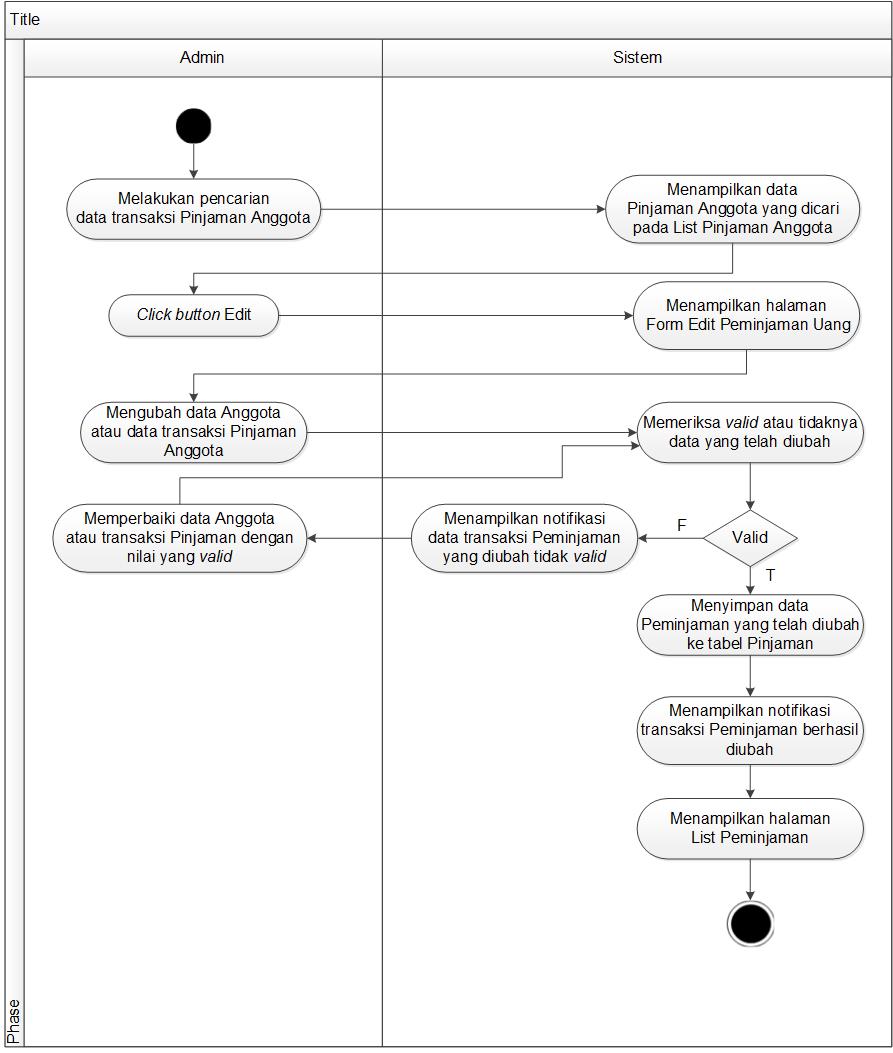
**Gambar 4.31 *Activity Diagram* Transaksi Simpanan (Pembayaran)**

* 1. ***Activity Diagram* Tambah Pinjaman**



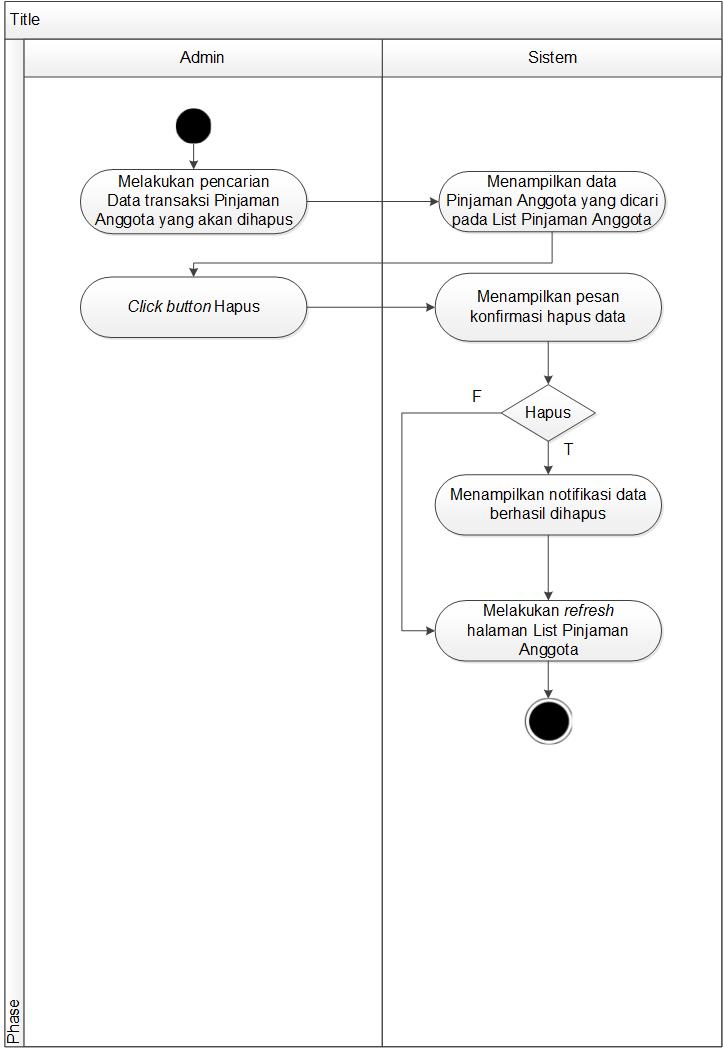
**Gambar 4.32 *Activity Diagram* Tambah Pinjaman**

* 1. ***Activity Diagram* Edit Pinjaman**



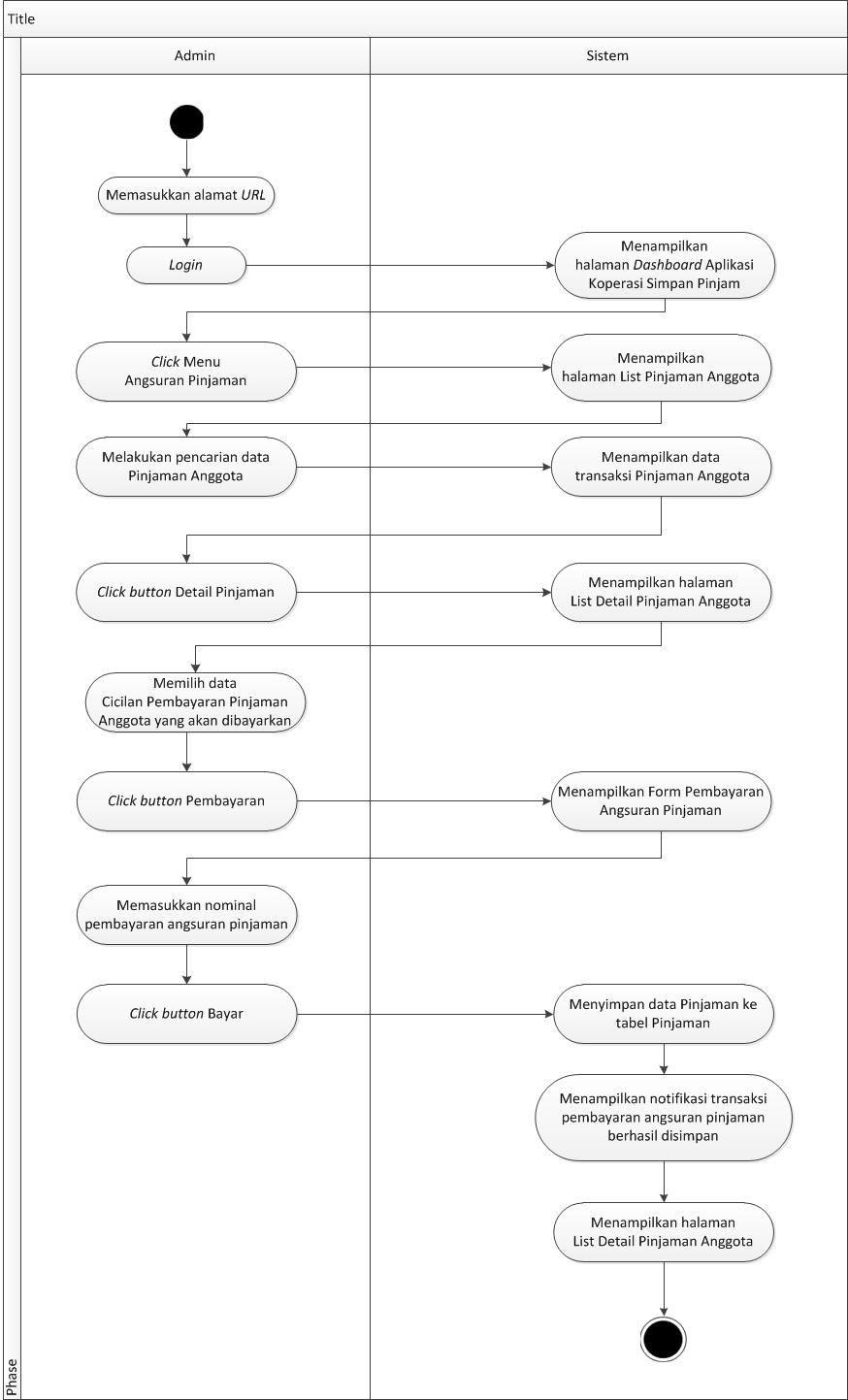
**Gambar 4.33 *Activity Diagram* Edit Pinjaman**

* 1. ***Activity Diagram* Hapus Pinjaman**



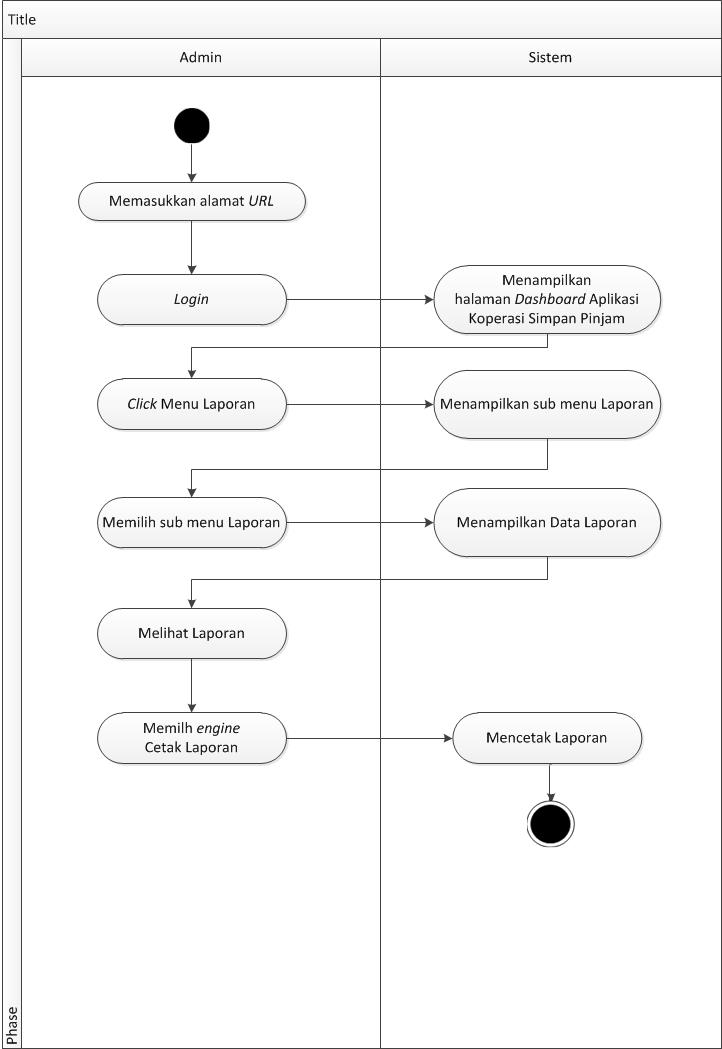
**Gambar 4.34 *Activity Diagram* Hapus Pinjaman**

* 1. ***Activity Diagram* Angsuran Pembayaran (Pinjaman)**



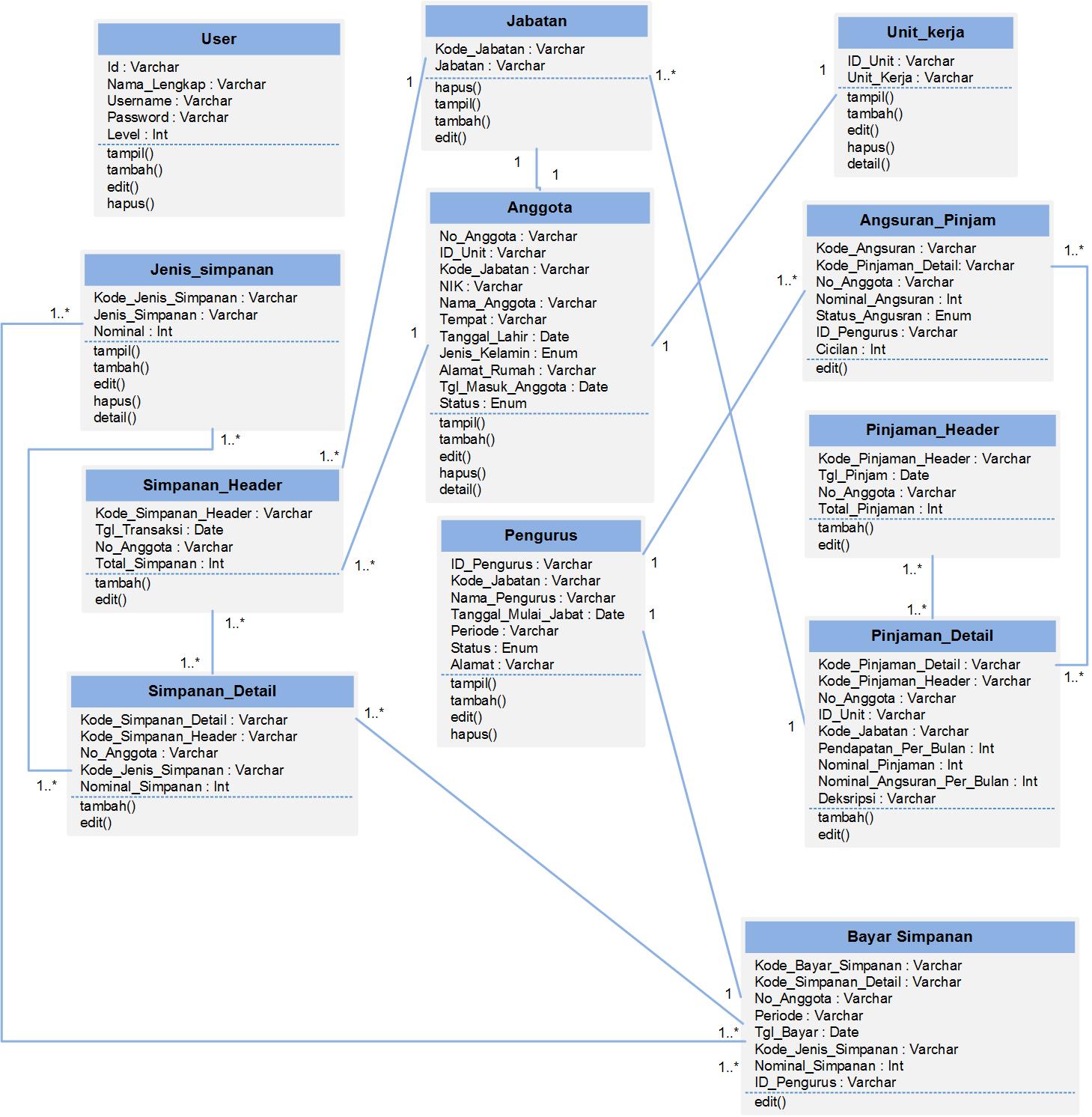
**Gambar 4.35 *Activity Diagram* Angsuran Pembayaran (Pinjaman)**

* 1. ***Activity Diagram View* Laporan**



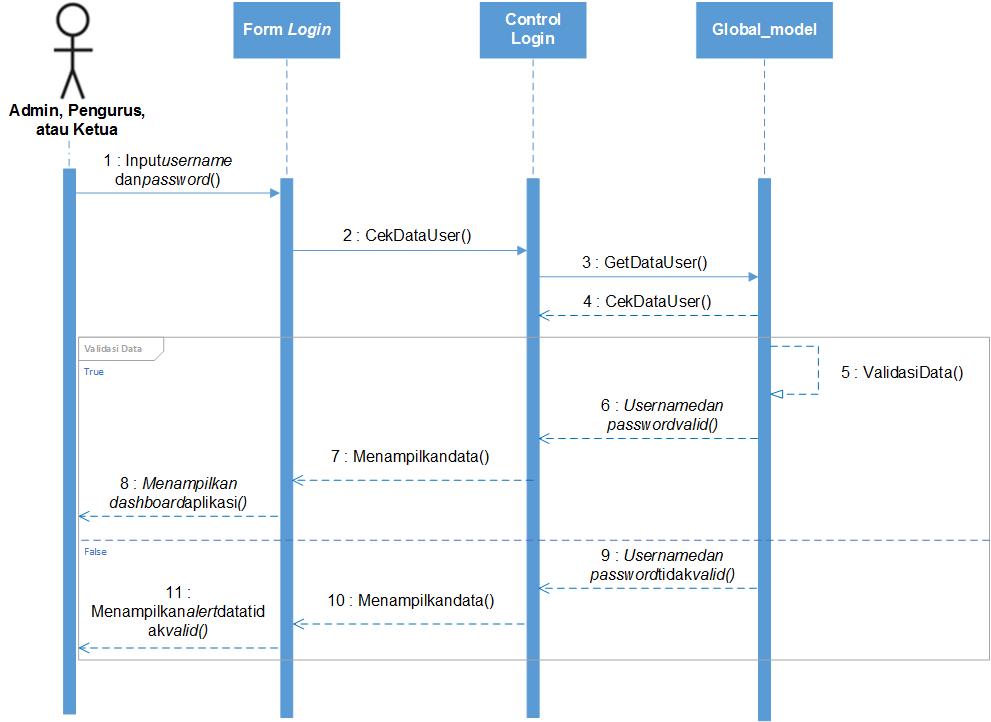
**Gambar 4.36 *Activity Diagram View* Laporan**

1. ***Class Diagram***



**Gambar 4.37 *Class Diagram***

1. ***Sequence Diagram***
2. ***Sequence Diagram Login***

****

**Gambar 4.38 *Sequence Diagram Login***

1. ***Sequence Diagram* Data Pengurus Koperasi**

****

**Gambar 4.39 *Sequence Diagram* Pengurus Koperasi**

1. ***Sequence Diagram* Data Anggota**

****

**Gambar 4.40 *Sequence Diagram* Data Anggota**

1. ***Sequence Diagram* Data Jabatan**

****

**Gambar 4.41 *Sequence Diagram* Data Jabatan**

1. ***Sequence Diagram* Data Jenis Simpanan**

****

**Gambar 4.42 *Sequence Diagram* Data Jenis Simpanan**

1. ***Sequence Diagram* Data Unit Kerja**

****

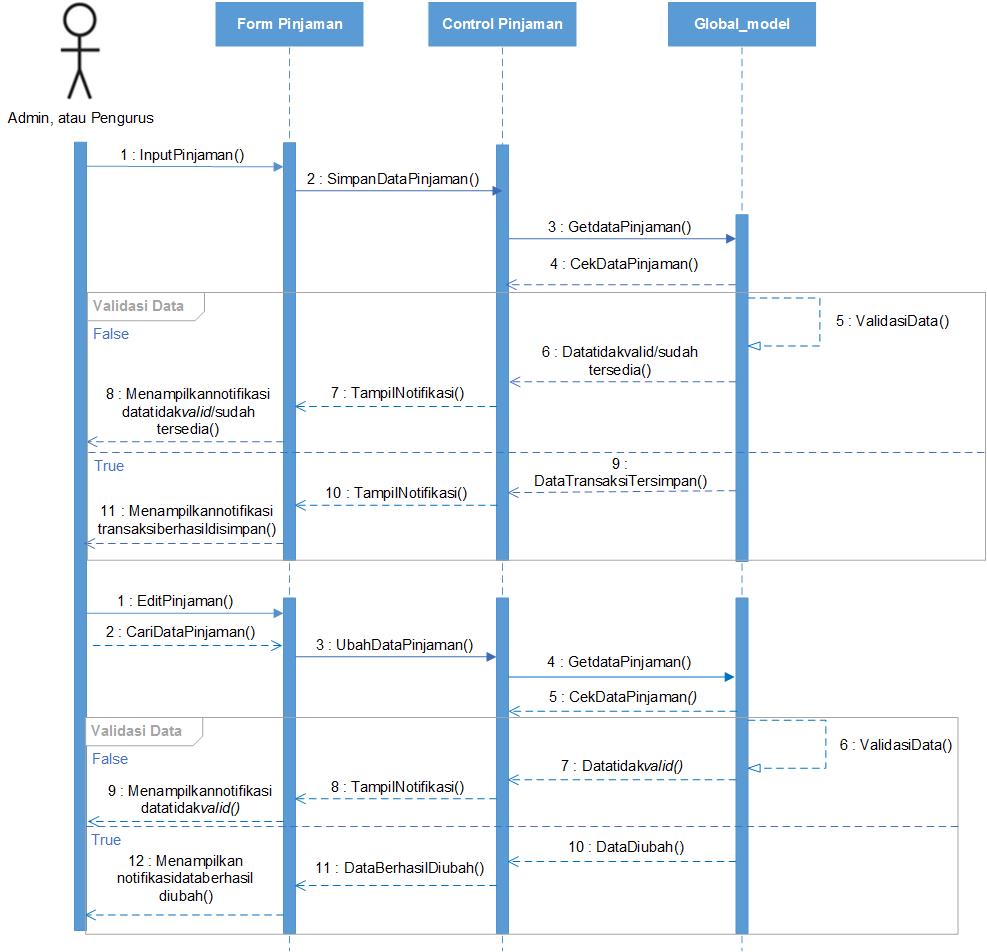
**Gambar 4.43 *Sequence Diagram* Data Unit Kerja**

1. ***Sequence Diagram* Data Simpanan**



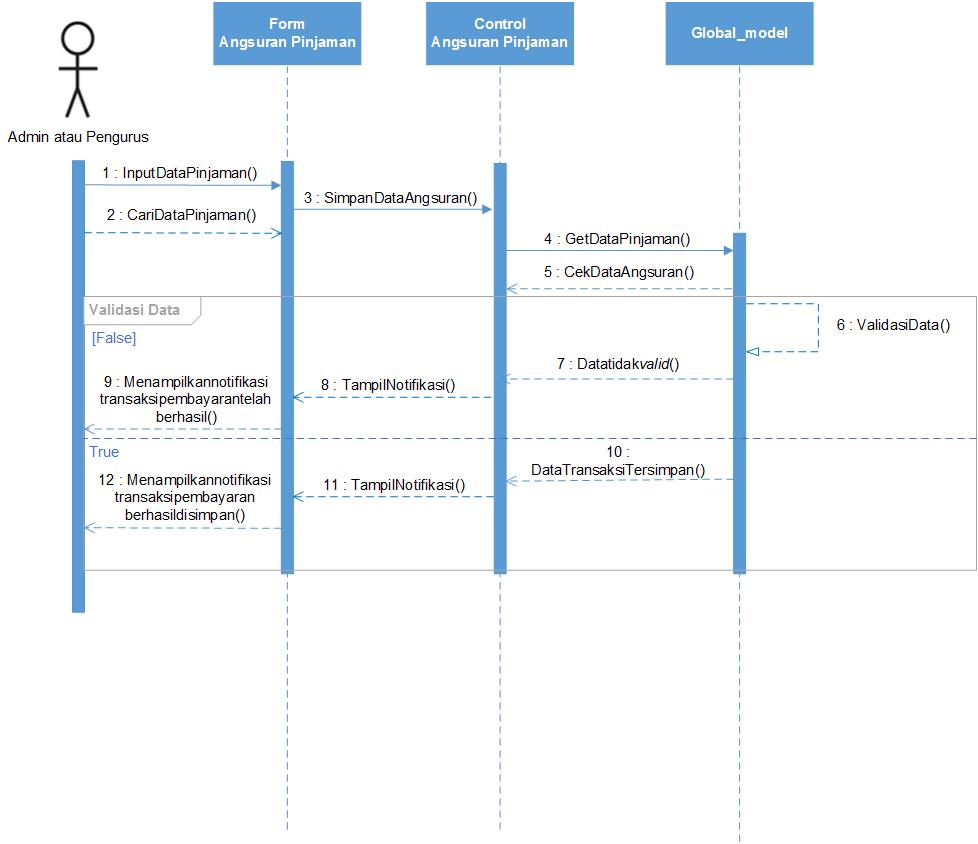
**Gambar 4.44 *Sequence Diagram* Data Simpanan**

1. ***Sequence Diagram* Data Pinjaman**

****

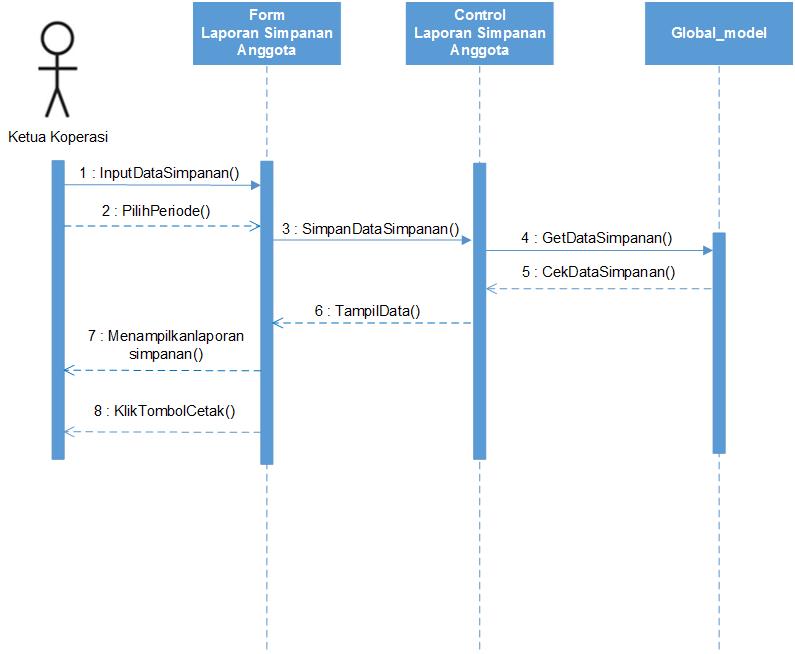
**Gambar 4.45 *Sequence Diagram* Data Pinjaman**

1. ***Sequence Diagram* Data Angsuran Pembayaran Pinjaman**



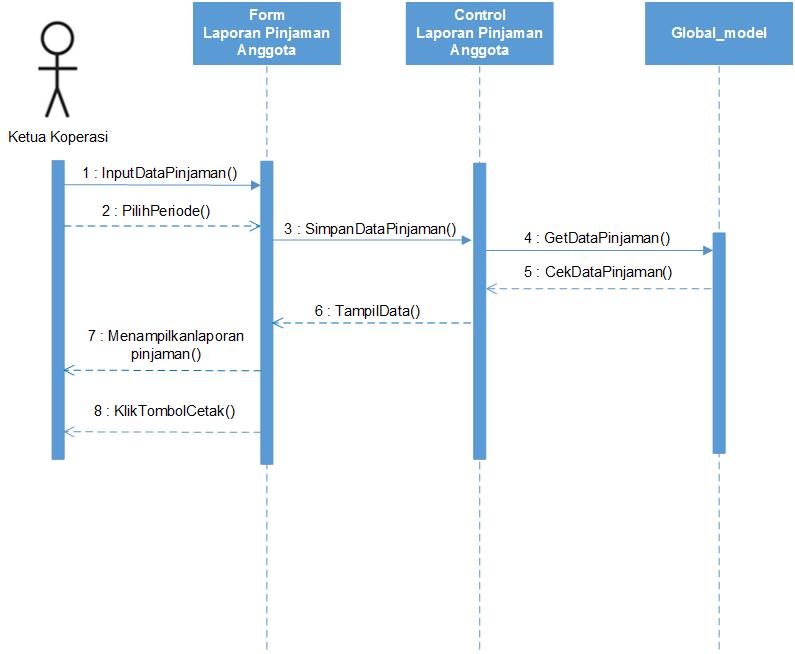
**Gambar 4.46 *Sequence Diagram* Data Angsuran Pinjaman**

1. ***Sequence Diagram* Melihat Laporan Simpanan Anggota**



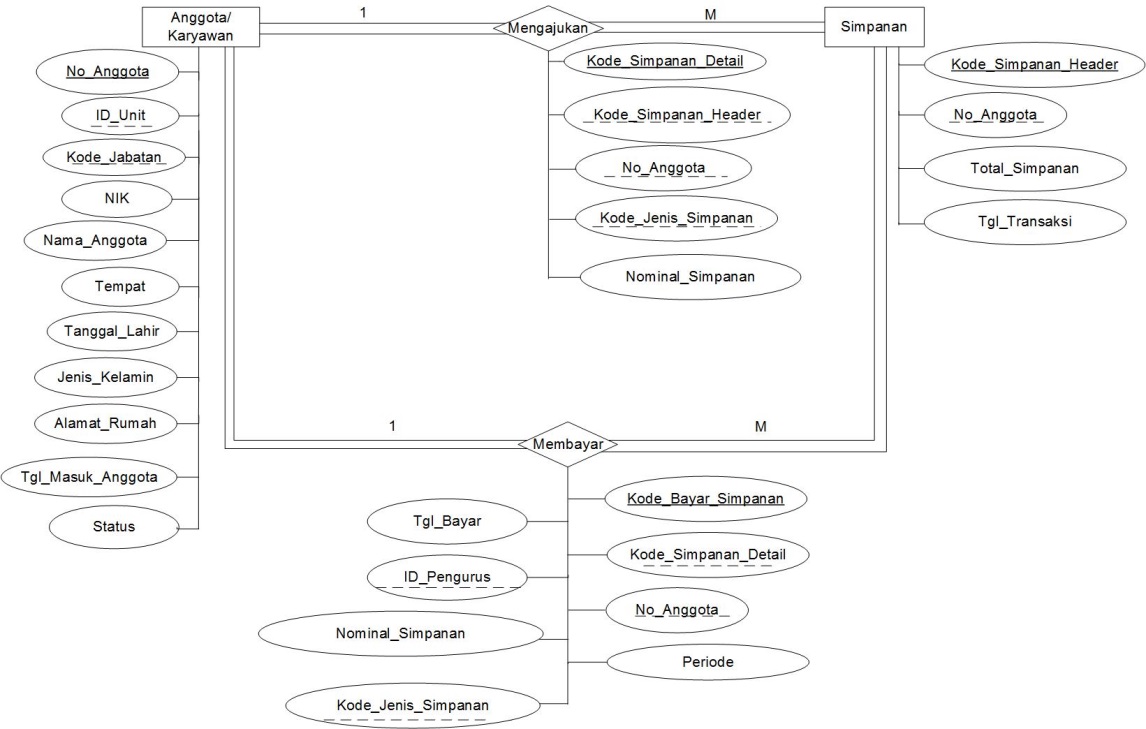
**Gambar 4.47 *Sequence Diagram* Melihat Laporan Simpanan**

1. ***Sequence Diagram* Melihat Laporan Pinjaman Anggota**

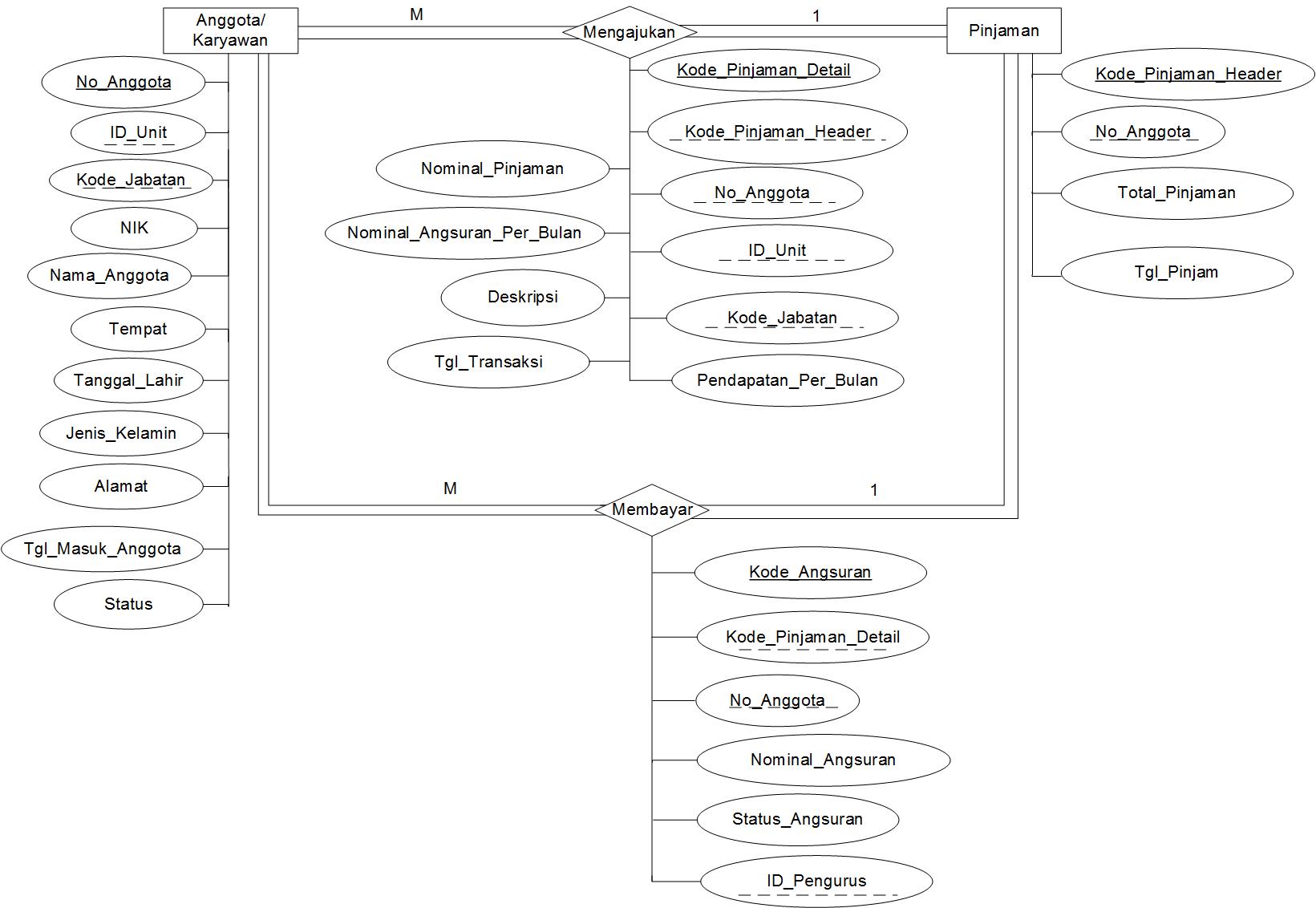


**Gambar 4.48 *Sequence Diagram* Melihat Laporan Pinjaman**

* + 1. **Perancangan Basis Data (*Database*)**

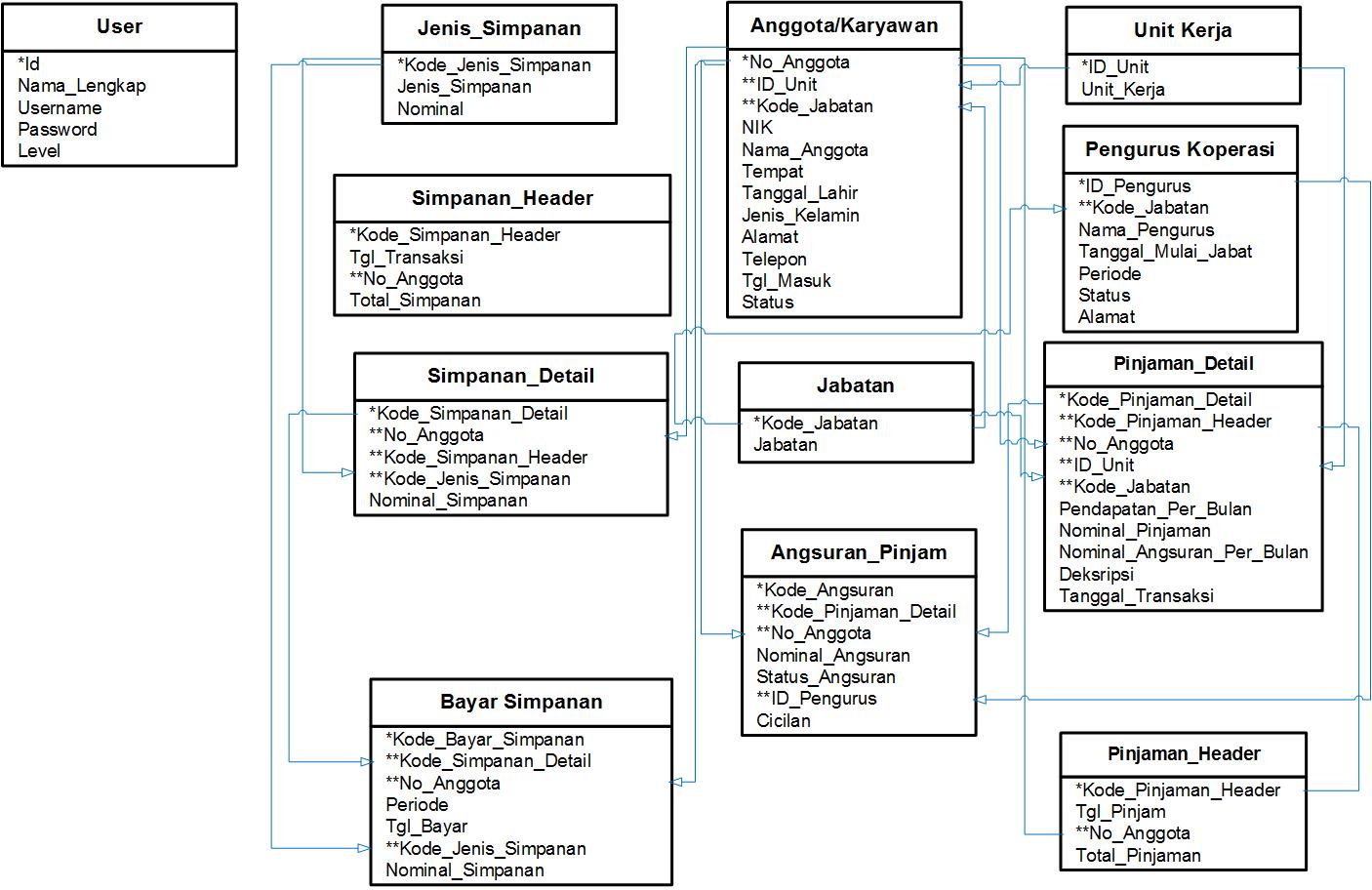
1. ***Entity Relational Diagram***

**Gambar 4.53 *Entity Relational Diagram* (ERD) Proses Simpanan**



**Gambar 4.54 *Entity Relational Diagram* (ERD) Proses Pinjaman**

1. **Relasi Antar Tabel**



**Gambar 4.55 Relasi Antar Tabel**

1. **Klasifikasi File**
2. Tabel Admin

Nama Tabel : user

*Type File* : Master

*Primary Key* : Id

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *File*** | **Tipe Data** | **Ukuran** | **Ket** |
| Id | *Varchar* | 10 | *PK* |
| Nama\_Lengkap | *Varchar* | 50 |  |
| Username | *Varchar* | 10 |  |
| Password | *Varchar* | 10 |  |
| Level | *Int* | 11 |  |

**Tabel 4.11 Data Tabel User**

1. Tabel Pengurus Koperasi

Nama Tabel : pengurus

*Type File* : Master

*Primary Key* : ID\_Pengurus

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *File*** | **Tipe Data** | **Ukuran** | **Ket** |
| ID\_Pengurus | *Varchar* | 10 | PK |
| Kode\_Jabatan | *Varchar* | 10 | FK |
| Nama\_Pengurus | *Varchar* | 30 |  |
| Tanggal\_Mulai\_Jabat | *Date* |  |  |
| Periode | *Varchar* | 12 |  |
| Status | *Enum* | ‘Aktif’,’Tidak\_Aktif’ |  |
| Alamat | *Varchar* | 100 |  |

**Tabel 4.12 Data Tabel Pengurus Koperasi**

1. Tabel Anggota

Nama Tabel : anggota

*Type File* : Master

*Primary Key* : ID\_Anggota

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *File*** | **Tipe Data** | **Ukuran** | **Ket** |
| No\_Anggota | *Varchar* | 10 | *PK* |
| ID\_Unit | *Varchar* | 10 | *FK* |
| Kode\_Jabatan | *Varchar* | 10 | *FK* |
| NIK | *Varchar* | 20 |  |
| Nama\_Anggota | *Varchar* | 30 |  |
| Tempat | *Varchar* | 20 |  |
| Tanggal\_Lahir | *Date* |  |  |
| Jenis\_Kelamin | *Enum* | ‘Pria’,’Wanita’ |  |
| Alamat\_Rumah | *Varchar* | 100 |  |
| Tgl\_Masuk\_Anggota | *Date* |  |  |
| Status | *Enum* | ‘Aktif’,’Tidak\_Aktif’ |  |

**Tabel 4.13 Data Tabel Anggota**

1. Tabel Jabatan

Nama Tabel : jabatan

*Type File* : Master

*Primary Key* : Kode\_Jabatan

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *File*** | **Tipe Data** | **Ukuran** | **Ket** |
| Kode\_Jabatan | *Varchar* | 10 | *PK* |
| Jabatan | *Varchar* | 20 |  |

**Tabel 4.14 Data Tabel Jabatan**

1. Tabel Jenis Simpanan

Nama Tabel : jenis\_simpanan

*Type File* : Master

*Primary Key* : Kode\_Jenis\_Simpanan

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *File*** | **Tipe Data** | **Ukuran** | **Ket** |
| Kode\_Jenis\_Simpanan | *Varchar* | 10 | *PK* |
| Jenis\_Simpanan | *Varchar* | 20 |  |
| Nominal | *Int* | 12 |  |

**Tabel 4.15 Data Tabel Jenis Simpanan**

1. Unit Kerja

Nama Tabel : unit\_kerja

*Type File* : Master

*Primary Key* : ID\_Unit

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *File*** | **Tipe Data** | **Ukuran** | **Ket** |
| ID\_Unit | *Varchar* | 10 | *PK* |
| Unit\_Kerja | *Varchar* | 20 |  |

**Tabel 4.16 Data Tabel Unit Kerja**

1. Tabel\_Simpanan\_Header

Nama Tabel : simpanan\_header

*Type File* : Transaksi

*Primary Key* : Kode\_Simpanan\_Header

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *File*** | **Tipe Data** | **Ukuran** | **Ket** |
| Kode\_Simpanan\_Header | *Varchar* | 10 | PK |
| Tgl\_Transaksi | *Date* |  |  |
| No\_Anggota | *Varchar* | 10 | FK |
| Total\_Simpanan | *Int* | 12 |  |

**Tabel 4.17 Data Tabel Simpanan Header**

1. Tabel Simpanan\_Detail

Nama Tabel : simpanan\_detail

*Type File* : Transaksi

*Primary Key* : Kode\_Simpanan\_Detail

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *File*** | **Tipe Data** | **Ukuran** | **Ket** |
| Kode\_Simpanan\_Detail | *Varchar* | 10 | PK |
| Kode\_Simpanan\_Header | *Varchar* | 10 | FK |
| No\_Anggota | *Varchar* | 10 | FK |
| Kode\_Jenis\_Simpanan | *Varchar* | 10 | FK |
| Nominal\_Simpanan | *Int* | 12 |  |

**Tabel 4.18 Data Tabel Simpanan Detail**

1. Tabel Pembayaran Simpanan

Nama Tabel : bayar\_simpanan

*Type File* : Transaksi

*Primary Key* : Kode\_Bayar\_Simpanan

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *File*** | **Tipe Data** | **Ukuran** | **Ket** |
| Kode\_Bayar\_Simpanan | Varchar | 10 | PK |
| Kode\_Simpanan\_Detail | *Varchar* | 10 | FK |
| No\_Anggota | *Varchar* | 10 | FK |
| Periode | *Varchar* | 10 |  |
| Tgl\_Bayar | *Date* |  |  |
| Kode\_Jenis\_Simpanan | *Varchar* | 10 | FK |
| Nominal\_Simpanan | *Int* | 12 |  |
| ID\_Pengurus | *Varchar* | 10 | FK |

**Tabel 4.19 Data Tabel Pembayaran Simpanan**

1. Tabel\_Pinjaman\_Header

Nama Tabel : pinjaman\_header

*Type File* : Transaksi

*Primary Key* : Kode\_Pinjaman\_Header

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *File*** | **Tipe Data** | **Ukuran** | **Ket** |
| Kode\_Pinjaman\_Header | *Varchar* | 10 | PK |
| Tgl\_Pinjam | *Date* |  |  |
| No\_Anggota | *Varchar* | 10 | FK |
| Total\_Pinjaman | *Int* | 12 |  |

**Tabel 4.20 Data Tabel Pinjaman Header**

1. Tabel Pinjaman\_Detail

Nama Tabel : pinjaman\_detail

*Type File* : Transaksi

*Primary Key* : Kode\_Pinjaman\_Detail

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *File*** | **Tipe Data** | **Ukuran** | **Ket** |
| Kode\_Pinjaman\_Detail | *Varchar* | 10 | PK |
| Kode\_Pinjaman\_Header | *Varchar* | 10 | FK |
| No\_Anggota | *Varchar* | 10 | FK |
| ID\_Unit | *Varchar* | 10 | FK |
| Kode\_Jabatan | *Varchar* | 10 | FK |
| Pendapatan\_Per\_Bulan | *Int* | 12 |  |
| Nominal\_Pinjaman | *Int* | 12 |  |
| Nominal\_Angsuran\_Per\_Bulan | *Int* | 12 |  |
| Deskripsi | *Varchar* | 1000 |  |

**Tabel 4.21 Data Tabel Pinjaman Detail**

1. Tabel Angsuran Pembayaran Peminjaman

Nama Tabel : angsuran\_pinjam

*Type File* : Transaksi

*Primary Key* : Kode\_Angsuran

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *File*** | **Tipe Data** | **Ukuran** | **Ket** |
| Kode\_Angsuran | *Varchar* | 10 | PK |
| Kode\_Pinjaman\_Detail | *Varchar* | 10 | FK |
| No\_Anggota | *Varchar* | 10 |  |
| Nominal\_Angsuran | *Int* | 12 |  |
| Status\_Angsuran | *Enum* | ‘Lunas’,’Belum\_Lunas’ |  |
| ID\_Pengurus | *Varchar* | 10 | FK |
| Cicilan | *Int* |  |  |

**Tabel 4.22 Data Tabel Angsuran Pembayaran Pinjaman**

* + 1. **Perancangan *User Interface* (Antar Muka Pengguna)**
  1. **Desain Input (Sketsa)**
     + - 1. **Desain *Input Login***

|  |
| --- |
| Form\_Login |
| Username : :  Password :   |  |  | | --- | --- | | Login | Cancel | |

**Gambar 4.50 *Desain Input* *Login***

* + - * 1. **Desain Form Input Data Master Anggota**

|  |
| --- |
| Form Keanggotaan |
| No\_Anggota : Tanggal Masuk Anggota :  NIM : Jenis Kelamin : Pria Wanita  Nama Anggota : Unit Kerja :  Tempat : Alamat :  Tanggal Lahir :   |  |  | | --- | --- | | Submit | Cancel | |

**Gambar 4.51 *Desain Input* Data Master Anggota**

1. **Desain Form Input Data Master Jenis Simpanan**

|  |
| --- |
| Form Jenis Simpanan |
| Kode Jenis Simpanan :  Jenis Simpanan :  Nominal Simpanan :   |  |  | | --- | --- | | Submit | Cancel | |

**Gambar 4.52 *Desain Input* Data Master Jenis Simpanan**

1. **Desain Form Input Data Master Unit Kerja**

|  |
| --- |
| Form Unit Kerja |
| ID Unit :  Unit Kerja :   |  |  | | --- | --- | | Submit | Cancel | |

**Gambar 4.53 *Desain Input* Data Master Unit Kerja**

1. **Desain Form Input Data Master Jabatan**

|  |
| --- |
| Form Jabatan |
| Kode Jabatan :  Jabatan :   |  |  | | --- | --- | | Submit | Cancel | |

**Gambar 4.54 *Desain Input* Data Master Jabatan**

1. **Desain Form Input Data *User***

|  |
| --- |
| Form User |
| Username :  Nama Lengkap :  Password :  Level : Admin Pengurus Ketua   |  |  | | --- | --- | | Submit | Cancel | |

**Gambar 4.55 *Desain Input* Data User**

1. **Desain Form Input Data Pengurus**

|  |
| --- |
| Form Pengurus |
| ID Pengurus :  Nama Pengurus :  Telepon :  Alamat :   |  |  | | --- | --- | | Submit | Cancel | |

**Gambar 4.56 *Desain Input* Data Pengurus**

1. **Desain Input Form Transaksi Simpanan**

|  |
| --- |
| Form Simpanan |
| Kode Simpanan Header :  Cari  Nomor Anggota :  Total Simpanan :  Tambah Simpanan  Kode Simpanan Detail :  Nominal Simpanan Pokok :  Nominal Simpanan Wajib :  Nominal Simpanan Sukarela :   |  |  | | --- | --- | | Submit | Cancel | |

**Gambar 4.57 *Desain Input* Data Transaksi Simpanan**

1. **Desain Input Transaksi Pinjaman**

|  |
| --- |
| Form Peminjaman |
| Kode Pinjaman Header :  Cari  Nomor Anggota :  Total Pinjaman :  Tambah Data Pinjaman  Kode Pinjaman Detail : Nominal Angsuran :  Unit Kerja : Tanggal Transaksi :    Jabatan : Deksripsi :  Pendapatan Per Bulan :  Nominal Pinjaman :   |  |  | | --- | --- | | Submit | Cancel | |

**Gambar 4.58 *Desain Input* Data Peminjaman**

1. **Desain Input Transaksi Pembayaran Simpanan**

|  |
| --- |
| Form Input Transaksi Pembayaran Simpanan |
| Kode Bayar Angsuran :  Cari  Kode Simpanan Detail :  No Anggota :  Nama Anggota :  Nominal Angsuran :  Status Angsuran : Lunas Belum Lunas  Nama Pengurus :     |  |  | | --- | --- | | Submit | Cancel | |

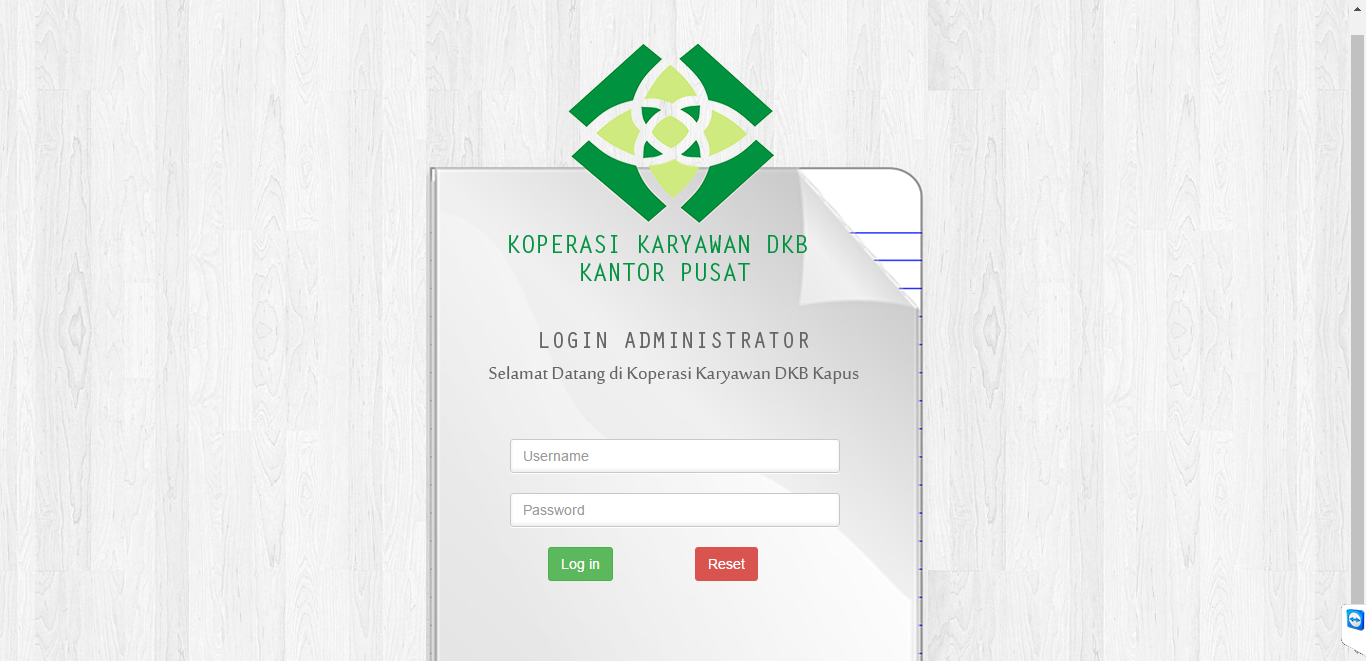
**Gambar 4.59 *Desain Input* Data Pembayaran Simpanan**

1. **Desain Input Transaksi Pembayaran Angsuran**

|  |
| --- |
| Form Input Transaksi Pembayaran Angsuran Pinjaman |
| Kode Angsuran :  Kode Pinjaman Detail :  Cari  No Anggota :  Nama Anggota :  Nominal Angsuran :  Status Angsuran : Lunas Belum Lunas  Nama Pengurus :   |  |  | | --- | --- | | Submit | Cancel | |

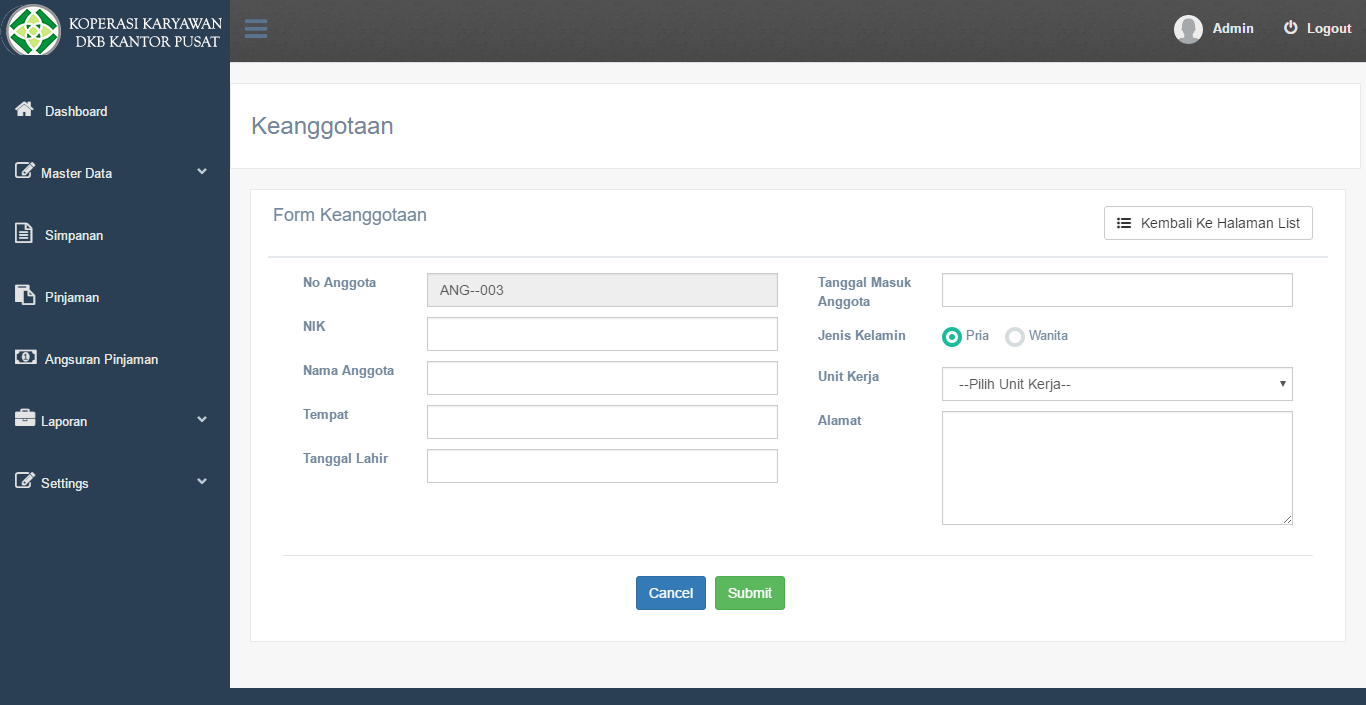
**Gambar 4.60 *Desain Input* Data Pembayaran Angsuran**

1. **Desain Input (Program Aplikasi)**
   * + - 1. ***Form Input Login***



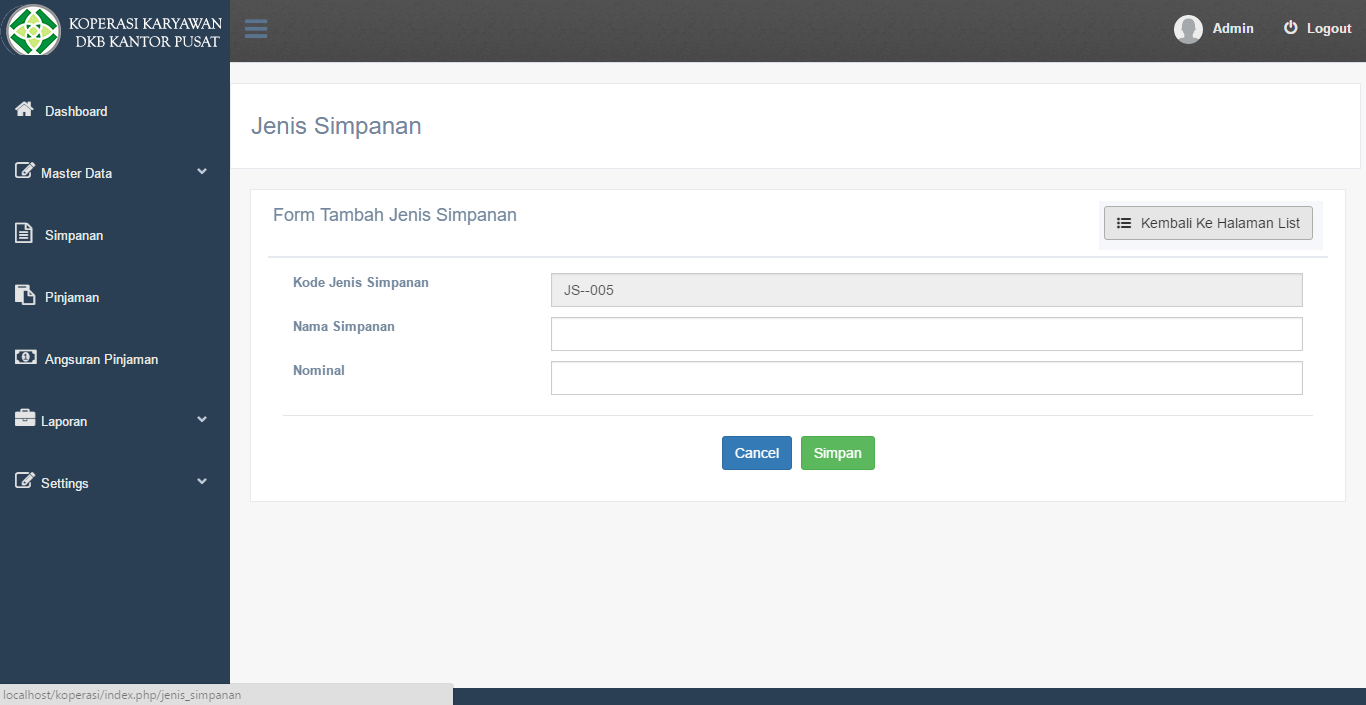
**Gambar 4.61 *Form Input Login***

* + - * 1. ***Form Input* Data Master Anggota**



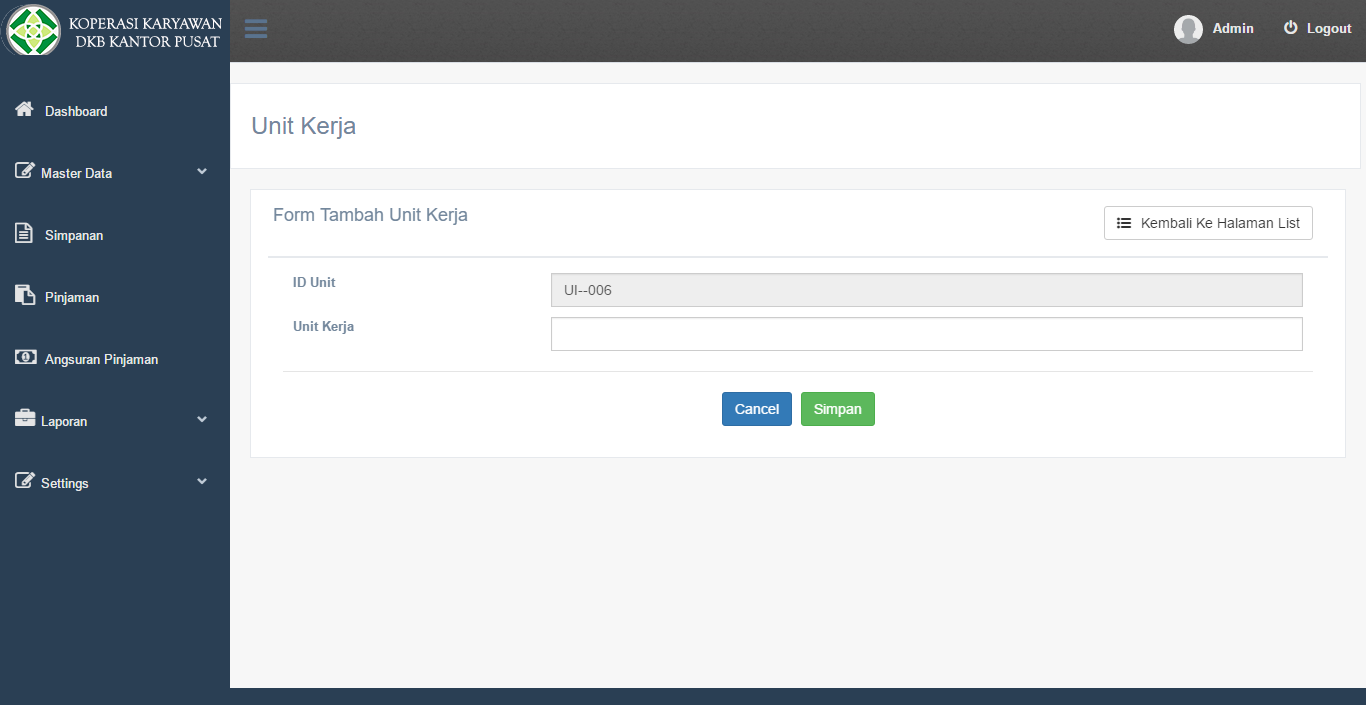
**Gambar 4.62 *Form Input* Data Master Anggota**

* + - * 1. ***Form Input* Data Master Jenis Simpanan**

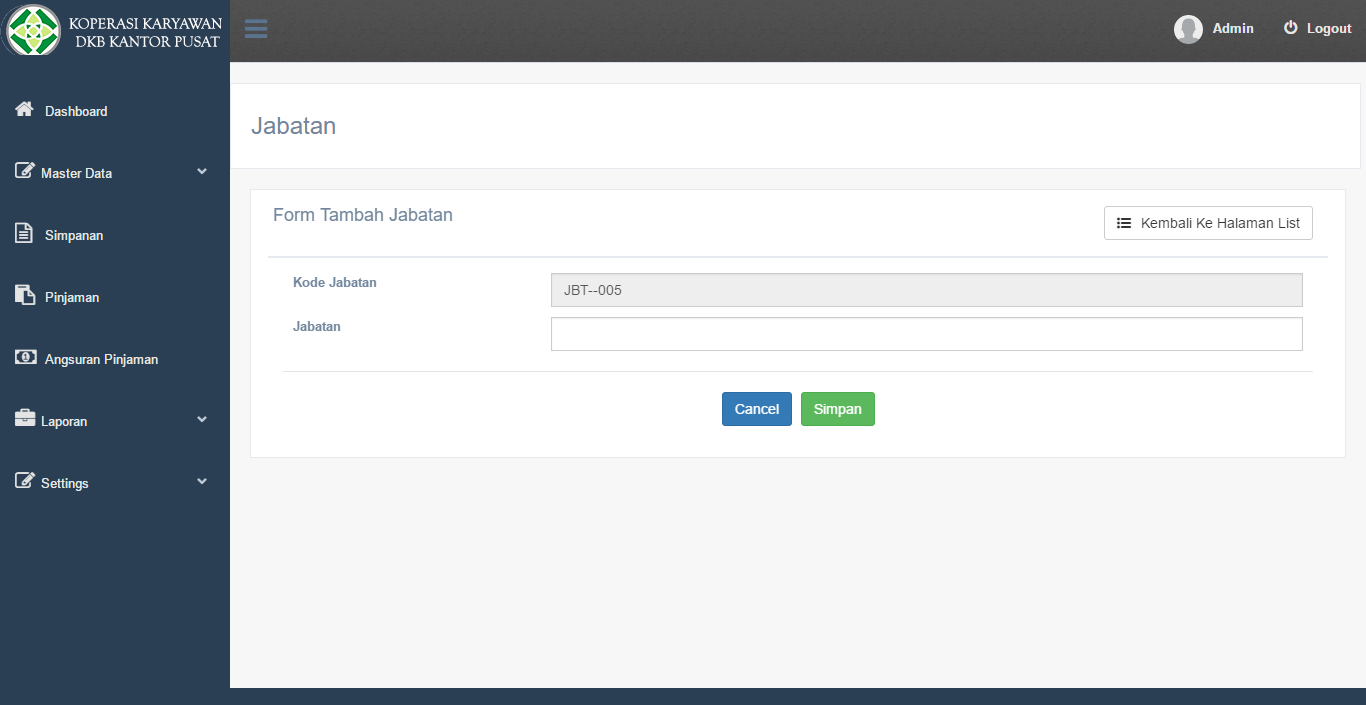


**Gambar 4.63 *Form Input* Data Master Jenis Simpanan**

* + - * 1. ***Form Input* Data Master Unit Kerja**



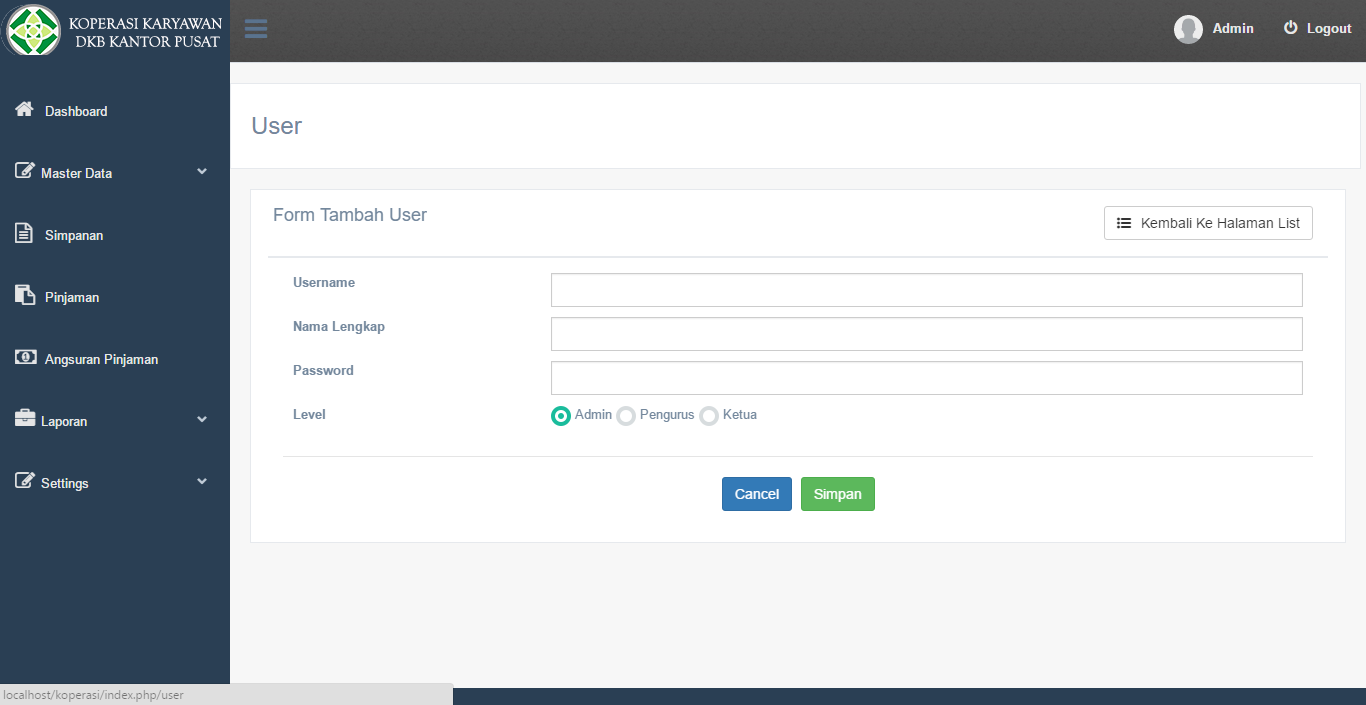
**Gambar 4.64 *Form Input* Data Master Unit Kerja**



* + - * 1. ***Form Input* Data Master Jabatan**

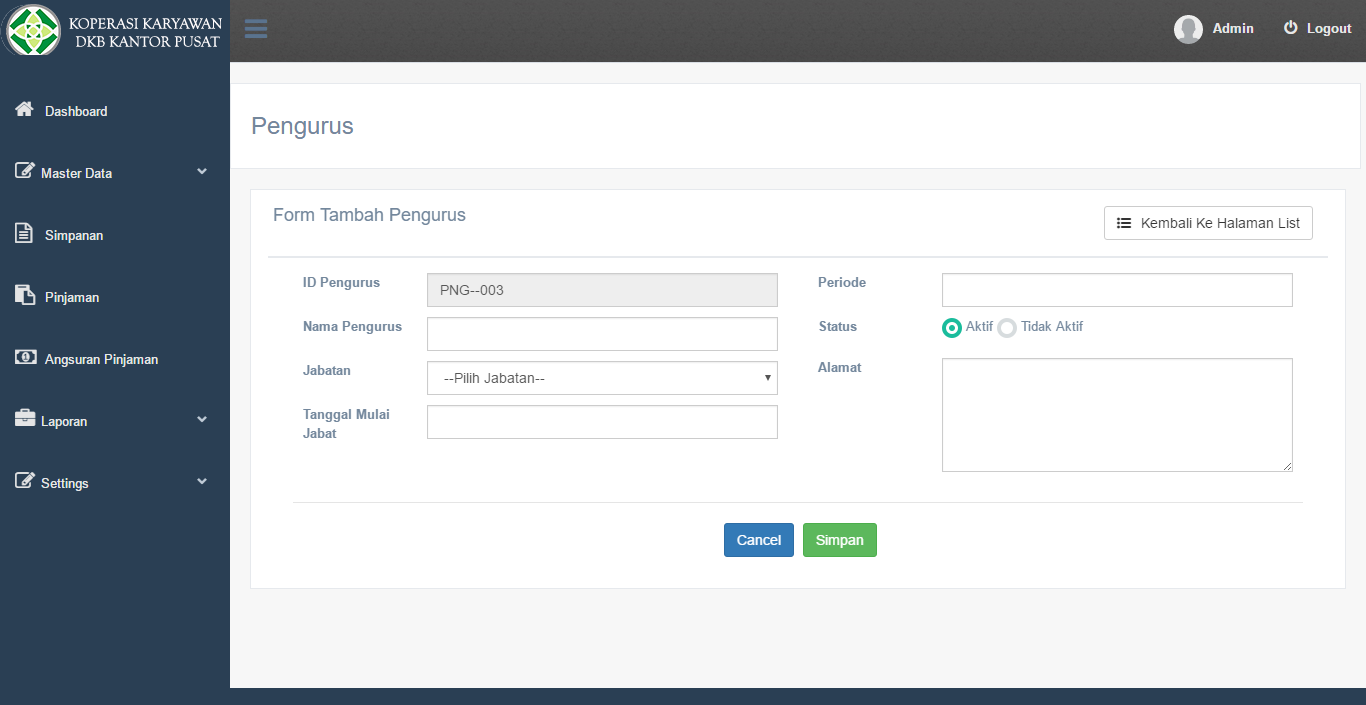
**Gambar 4.65 *Form Input* Data Master Jabatan**

**f. *Form Input* Data User**



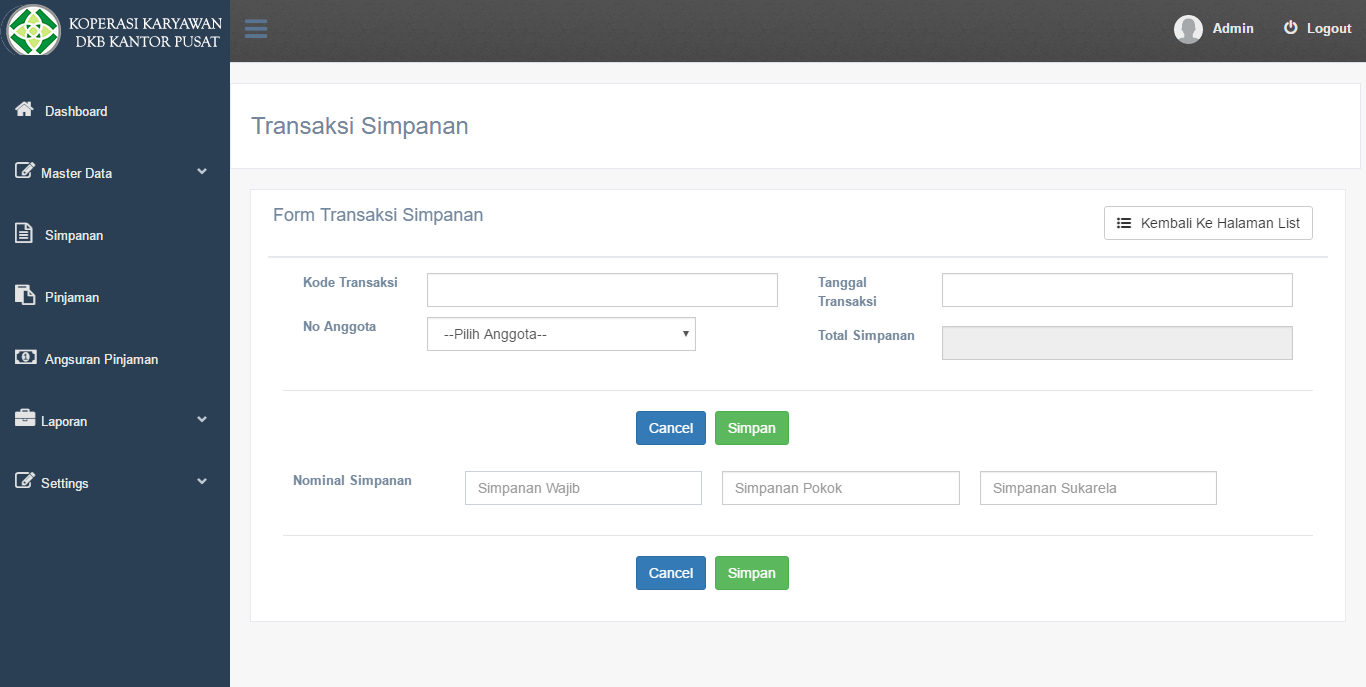
**Gambar 4.66 *Form Input* Data *User***

**g. *Form Input* Data Pengurus**



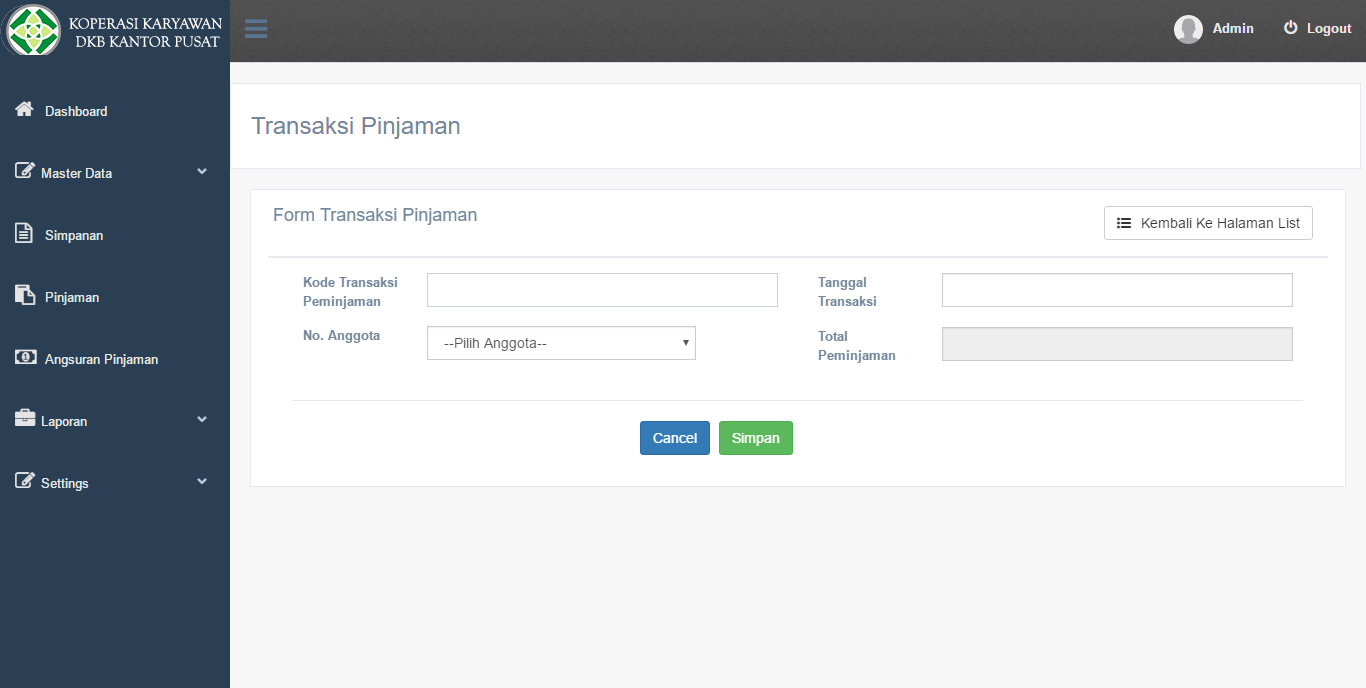
**Gambar 4.67 *Form Input* Data Pengurus**

**h. *Form Input* Data Transaksi Simpanan**



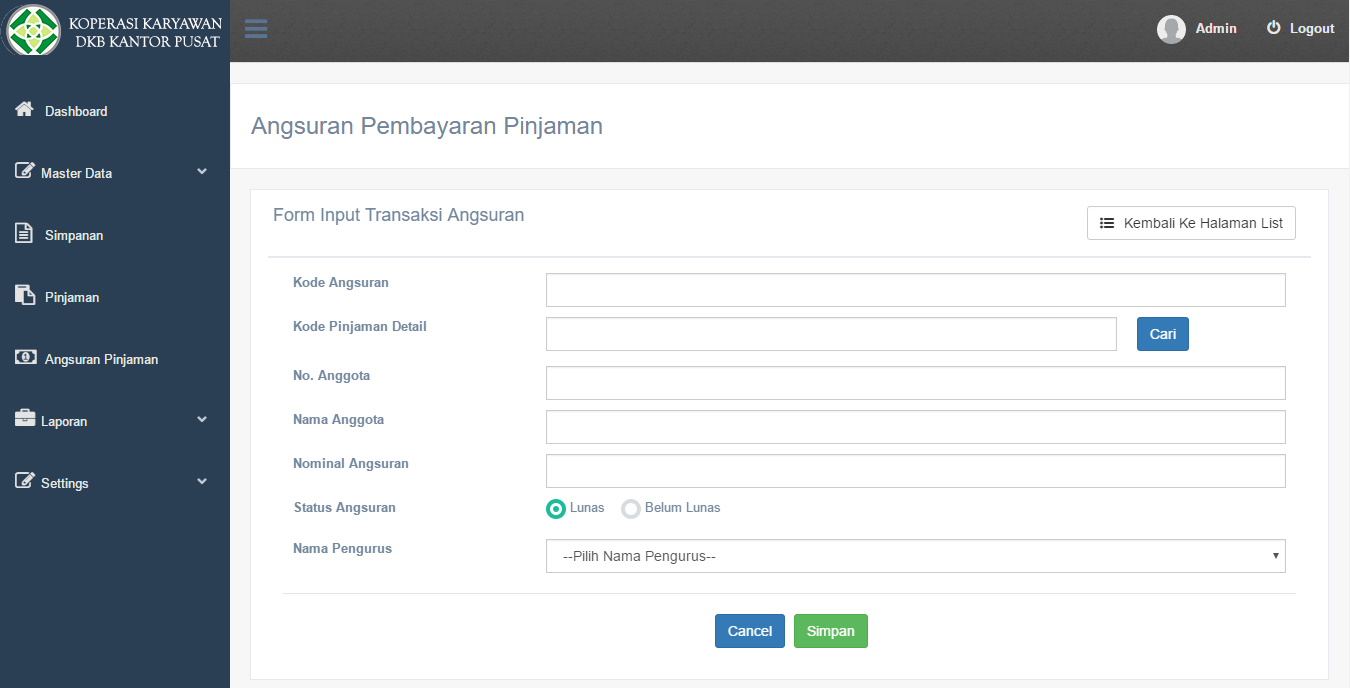
**Gambar 4.68 *Form Input* Data Transaksi Simpanan**

**i. *Form Input* Data Pinjaman**



**Gambar 4.69 *Form Input* Data Transaksi Pinjaman**

**j. *Form Input* Transaksi Pembayaran Angsuran**



**Gambar 4.70 *Form Input* Data Transaksi Pembayaran Angsuran**

1. **Perancangan Laporan**
2. **Desain Sketsa Laporan Simpanan**

|  |
| --- |
| Laporan Simpanan Anggota |
| Periode : XXXXX   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | No Anggota | Nama Anggota | Simpanan Anggota | | | | | Pokok | Wajib | Sukarela | Total Simpanan Anggota | | XXXXX | XXXXX | 99999 | 99999 | 99999 | 99999 |   Total Pendapatan Simpanan : 99999 |

**Gambar 4.71 Desain Sketsa Laporan Simpanan**

1. **Desain Sketsa Laporan Pinjaman**

|  |
| --- |
| Laporan Pinjaman Anggota |
| Periode : XXXXX   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | No Anggota | Nama Anggota | Pinjaman Anggota | | | | | Nominal Pinjaman | Nominal Angsuran Per Bulan | Status Bayar | Total Pinjaman | | XXXXX | XXXXX | 99999 | 99999 | XXXXX | 99999 |   Total Pendapatan Simpanan : 99999 |

**Gambar 4.72 Desain Sketsa Laporan Pinjaman**

* + 1. **Kebutuhan Infrastruktur**

Dalam merancang Aplikasi Koperasi Simpan Pinjam ini, dibutuhkan beberapa perangkat pendukung agar tercipta aplikasi transaksi simpanan serta peminjaman yang sesuai dengan harapan pihak Koperasi Karyawan DKB Kantor Pusat (Kapus) maupun sesuai pula dengan harapan penulis. Berikut ini merupakan rangkaian kebutuhan infrastruktur baik perangkat keras (*hardware*) maupun perangkat lunak (*software*) yakni meliputi :

1. **Perangkat Keras (*Hardware*)**
2. *Processor* AMD *Quad Core* A8-6410 2GHZ
3. *Memory* DDR3 4GB
4. *Harrdisk* 500GB
5. *Keyboard*
6. *Printer*
7. *Mouse*
8. **Perangkat Lunak (*Software*)**
9. Sistem Operasi : Windows 10
10. Program : *CodeIgniter* / *Framework* PHP
11. *Database* : MySQL
12. *Web Server* : XAMPP *version* 3.2.1
13. *Web Browser* : Google Chrome
14. Aplikasi *Editor* : Notepad++
15. Aplikasi *Design* : Microsoft Office Visio 2010 dan

Microsoft Office Visio2013