

LAPORAN KERJA PRAKTEK
SISTEM INFORMASI PENJUALAN
TOKO KOPERASI KARYA AUDITA BPKP
YOGYAKARTA

Diijukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Informatika



Disusun oleh:

Jan Faris Majd

08650052

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
YOGYAKARTA

2012

PENGESAHAN LAPORAN KERJA PRAKTIK

**Perancangan & Pembangunan Sistem Informasi Penjualan
Di Koperasi Karya Audita BPKP Yogyakarta**

Disusun oleh :

Nama : Jan Faris Majd

NIM : 08650052

Telah diseminarkan pada tanggal : 1 Juni 2012

Dosen Pembimbing



Shofwatul Uyun, S.T, M.Kom

NIP. 19820511-200604-2-002

Dosen Penguji


M. Mufarid

NIP. 19790331 200501 1 004

Mengetahui,

a.n Dekan

Ketua Program Studi



Andi Mulyanto, S.Si, M.Kom.

NIP. 19710823 199903 1 003

PENGESAHAN LAPORAN KERJA PRAKTIK

**Perancangan & Pembangunan Sistem Informasi Penjualan
Di Koperasi Karya Audita BPKP Yogyakarta**

Disusun oleh :

Nama : Jan Faris Majd

NIM : 08650052

Telah diseminarkan pada tanggal :

Dosen Pembimbing

Dosen Penguji

Shofwatul Uyun, S.T, M.Kom

NIP. 19820511-200604-2-002

NIP.

Mengetahui,

a.n Dekan

Ketua Program Studi

Agus Mulyanto, S.Si, M.Kom.

NIP. 19710823 199903 1 003

KATA PENGANTAR

Segala puji syukur bagi Allah SWT yang telah memberikan banyak kenikmatan serta pertunjuk-Nya dalam setiap kesulitan yang ada selama pelaksanaan kerja praktek. Atas berkat rahmat-Nya, pelaksanaan kerja praktek yang dilakukan di perusahaan Batik Seno dapat terselesaikan dengan baik.

Kerja praktek merupakan studi lapangan yang bertujuan untuk memberikan suatu gambaran kepada seluruh mahasiswa yang telah menyelesaikan sebagian dari teori yang didapatkan di bangku kuliah. Dengan adanya pelaksanaan kerja praktek ini dapat memberikan masukan dan praktek di lapangan tentang bagaimana situasi dan kondisi yang ada di tempat pelaksanaan kerja praktek dan dapat mengimplementasikan ilmu yang didapat di bangku kuliah sesuai dengan konsentrasi yang ditawarkan di Program Studi Teknik *Informatika* UIN Sunan Kalijaga Yogyakarta. Oleh karena itu, penyusun mengucapkan terima kasih kepada

1. Prof. Drs. H. Akh. Minhaji, MA., Ph.D., selaku Dekan Fakultas Sains dan Teknologi, UIN Sunan Kalijaga Yogyakarta
2. Bapak Agus Mulyanto, S.Si., M.Kom., selaku Kaprodi Teknik *Informatika*
3. Ibu Shofwatul 'Uyun, S.T, M.Kom selaku dosen pembimbing yang telah banyak memberi dukungan serta pengarahan demi kelancaran pelaksanaan kerja praktek.

4. Ibu Emprah Handayani selaku Sekretaris Koperasi Karya Audita dan pembimbing lapangan yang telah memberikan kesempatan untuk melaksanakan kerja praktek di Koperasi Karya Audita.
5. Ayah dan Ibu yang selalu setia memberikan dukungan serta doa yang menjadi sumber motivasi dan inspirasi.
6. Teman-teman Prodi Teknik *Informatika* UIN Sunan Kalijaga yang telah banyak membantu dalam pelaksanaan kerja praktek dan penyusunan laporannya.

Penulis menyadari masih banyak kekurangan dan kelemahan dalam pelaksanaan dan penyusunan laporan kerja praktek ini. Semoga pelaksanaan kerja praktek ini dapat menjadi pengalaman yang berharga bagi penulis dan bermanfaat untuk masyarakat yang lebih luas.

Yogyakarta, 19 Mei 2012

DAFTAR ISI

LEMBAR PENGESAHAN	i
KATA PENGANTAR	ii
DAFTAR ISI	iv
DAFTAR TABEL.....	viii
DAFTAR GAMBAR	ix
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Batasan Masalah	2
1.3 Tujuan Kerja Praktek	3
1.4 Manfaat Kerja Praktek	3
BAB II TEMPAT KERJA PRAKTEK	
2.1 Gambaran Umum BPKP	4
2.1.1. Sejarah Singkat BPKP.....	4
2.1.2. Visi dan Misi.....	5
2.1.3. Struktur Organisasi.....	5
2.1.4. Struktur Organisasi BPKP DIY.....	6
2.2 Gambaran Umum Koperasi Karya Audita	
2.2.1 Sejarah Singkat.....	8

2.2.2 Identitas	8
2.2.3 Kepengurusan.....	8
2.2.4 Pengelola Unit dan Kekaryawanan.	9
BAB III LANDASAN TEORI	
3.1 Konsep Sistem Informasi	10
3.1.1. Sistem.	10
3.1.2. Informasi.....	10
3.1.3. Sistem Informasi.....	10
3.2 Konsep Basis Data	11
3.2.1. Database Server dengan MySQL.....	11
3.2.2. Database MySQL dengan SQL.....	12
3.2.3. DBPaket.....	12
3.3 Java	14
3.4 UML.....	14
3.4.1. <i>Usecase</i>	15
3.4.2. <i>Class Diagram</i>	16
3.4.3. <i>Sequence Diagram</i>	19
3.4.4. <i>Activity Diagram</i>	20

BAB IV	HASIL DAN PEMBAHASAN	
4.1	Hasil	23
4.1.1.	<i>Usecase Diagram</i>	23
4.1.2.	<i>Activity Diagram</i>	42
4.1.3.	<i>Sequence Diagram</i>	48
4.1.4.	<i>Class Diagram</i>	75
4.1.5.	Desain Tabel	78
4.2	Implementasi Sistem	82
4.2.1.	<i>Form Home</i>	82
4.2.2.	<i>Form Login</i>	82
4.2.3.	<i>Form Barang</i>	83
4.2.4.	<i>Form Anggota</i>	83
4.2.5.	<i>Form Pilihan Kredit</i>	84
4.2.6.	<i>Form Kredit</i>	85
4.2.7.	<i>Form Pelunasan Kredit</i>	85
4.2.8.	<i>Form Penjualan</i>	86
4.2.9.	<i>Form Pembelian</i>	87
4.2.10.	<i>Form Laporan</i>	88
4.2.11.	<i>Form User</i>	89

4.2.12. <i>Form</i> Cari Anggota.....	89
4.2.13. <i>Form</i> Cari Barang.....	90
4.2.14. <i>Form</i> Ganti Password.....	91
BAB V KESIMPULAN DAN SARAN	92
5.1 Kesimpulan	92
5.2 Saran	92
DAFTAR PUSTAKA	93
LAMPIRAN	94

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam kegiatan perkoperasian, usaha sampingan yang lumrah dilakukan adalah dengan membuka usaha jual beli, baik berupa sembako hingga keperluan kantor. Demi kelancaran usaha jual beli, hal yang dapat dilakukan untuk menunjangnya ialah dengan pencatatan dan penginventarisasian yang baik. Untuk membuat suatu sistem yang baik haruslah dibangun suatu sistem yang secara spesifik bertujuan untuk melakukan pencatatan jual-beli dan penginventorisasian.

Berdasarkan pertimbangan diatas, maka Koperasi Karya Audita Badan Pemeriksa Keuangan (BPKP) sebagai salah satu badan usaha yang bergerak dalam bidang Perkoperasian dapat mengoptimalkan kinerjanya dengan membangun Sistem Informasi Jual-Beli yang dapat diimplementasikan tidak hanya di koperasi, melainkan dapat diimplementasikan di took-toko skala menengah kebawah.

Sistem Informasi Jual-Beli adalah suatu sistem yang secara khusus ditujukan untuk melakukan pencatatan jual-beli dan penginventorisasian. Beberapa fitur yang terdapat dalam sistem ini adalah pencatatan jual-beli yang jelas, manajemen barang yang mudah dan lengkap, serta pembuatan report yang mudah dan tertata. Selain itu statistic penjualan juga dapat dilihat sehingga pihak koperasi dapat memutuskan strategi yang akan diambil kedepannya. Penggunaan sistem ini dibatasi hanya dapat

digunakan oleh admin yang bertugas sebagai penginput data yang berasal dari penjaga toko koperasi.

Aplikasi ini dikembangkan dengan basis Java. Keputusan ini diambil berdasarkan kelengkapan fitur Java yang mencakup segala aspek yang diperlukan untuk mengembangkan Sistem Informasi ini. Selain itu, Java juga mudah untuk dikembangkan untuk kedepannya.

Sistem Informasi Jual-Beli ini dikembangkan dari sistem yang dipakai sebelumnya, yaitu Microsoft (Ms.) Office Excel yang hanya sekedar melakukan pencatatan manual dan kekurangan fitur-fitur yang dibutuhkan dalam mengoptimalkan pencatatan. Fitur-fitur yang ada dari sistem sebelumnya akan tetap dipakai, namun dioptimalkan dan ditambah dengan fitur-fitur yang belum ada. Sistem yang lama kekurangan otomatisasi pembuatan laporan, tidak adanya sistem statistic, dan sistem manajemen yang kurang tertata.

1.2 Batasan Masalah

Adapun lingkup materi untuk Sistem Informasi Jual-Beli sebagai berikut:

1. Manajemen barang yang dibeli dan akan dijual.
2. Pencatatan jual-beli.
3. Otomatisasi pembuatan laporan penjualan, barang, dan pembelian.

1.3 Tujuan Kerja Praktek

Tujuan kerja praktek ini adalah untuk menyelesaikan salah satu syarat untuk memperoleh gelar sarjana di Universitas Islam Negeri Sunan Kalijaga Yogyakarta.

Adapun tujuan khusus yang dapat diambil ialah :

1. Membangun suatu sistem informasi penjualan untuk Koperasi karya Audita.
2. Pencatatan data transaksi dan barang, baik penjualan maupun pembelian secara sistematis bagi Koperasi Karya Audita melalui pembangunan aplikasi Sistem Informasi Penjualan.
3. Pembuatan Laporan tagihan dan tahunan yang bersifat otomatis.

1.4 Manfaat Kerja Praktek

Manfaat Pelaksanaan Kerja Praktek di Koperasi Karya Audita BPKP Yogyakarta adalah:

1. Membantu Koperasi Karya Audita dalam pembuatan aplikasi pencatatan Jual-Beli dan Inventorisasi.
2. Membantu Koperasi Karya Audita dalam pembuatan laporan transaksi Toko melalui aplikasi sistem informasi penjualan toko.

BAB II

TEMPAT KERJA PRAKTEK

2.1 Gambaran Umum BPKP

2.1.1 Sejarah Singkat BPKP

Dengan diterbitkan Keputusan Presiden Nomor 31 Tahun 1983 tanggal 30 Mei 1983. Direktorat Djendral Pengawasan Keuangan Negara (DJPKN) ditransformasikan menjadi BPKP, sebuah lembaga pemerintah non departemen (LPND) yang berada di bawah dan bertanggung jawab langsung kepada Presiden. Salah satu pertimbangan dikeluarkannya Keputusan Presiden Nomor 31 Tahun 1983 tentang BPKP adalah diperlukannya badan atau lembaga pengawasan yang dapat melaksanakan fungsinya secara leluasa tanpa mengalami kemungkinan hambatan dari unit organisasi pemerintah yang menjadi obyek pemeriksaannya. Keputusan Presiden Nomor 31 Tahun 1983 tersebut menunjukkan bahwa Pemerintah telah meletakkan struktur organisasi BPKP sesuai dengan proporsinya dalam konstelasi lembaga-lembaga Pemerintah yang ada. BPKP dengan kedudukannya yang terlepas dari semua departemen atau lembaga sudah barang tentu dapat melaksanakan fungsinya secara lebih baik dan obyektif.

Sesuai arahan Presiden RI tanggal 11 Desember 2006, BPKP melakukan reposisi dan revitalisasi fungsi yang kedua kalinya. Reposisi dan revitalisasi BPKP diikuti dengan penajaman visi, misi, dan strategi.

Visi BPKP yang baru adalah "Auditor Intern Pemerintah yang Proaktif dan Terpercaya dalam Mentransformasikan Manajemen Pemerintahan Menuju Pemerintahan yang Baik dan Bersih".

2.1.2 Visi dan Misi

Visi BPKP:

"Auditor Presiden yang responsif, interaktif, dan terpercaya untuk mewujudkan akuntabilitas keuangan negara yang berkualitas."

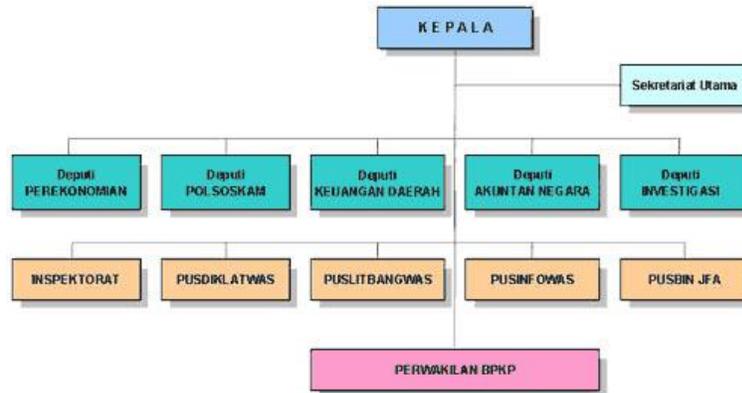
Misi BPKP:

1. Menyelenggarakan pengawasan intern terhadap akuntabilitas keuangan negara yang mendukung tata kelola pemerintahan yang baik dan bebas KKN.
2. Membina penyelenggaraan Sistem Pengendalian Intern Pemerintah.
3. Mengembangkan kapasitas pengawasan intern pemerintah yang profesional dan kompeten.
4. Menyelenggarakan sistem dukungan pengambilan keputusan yang andal bagi presiden/pemerintah.

2.1.3 Struktur Organisasi

BPKP dalam melaksanakan tugas pengawasannya mempunyai Kantor Perwakilan di 25 Provinsi seluruh Indonesia dan 8 Kantor

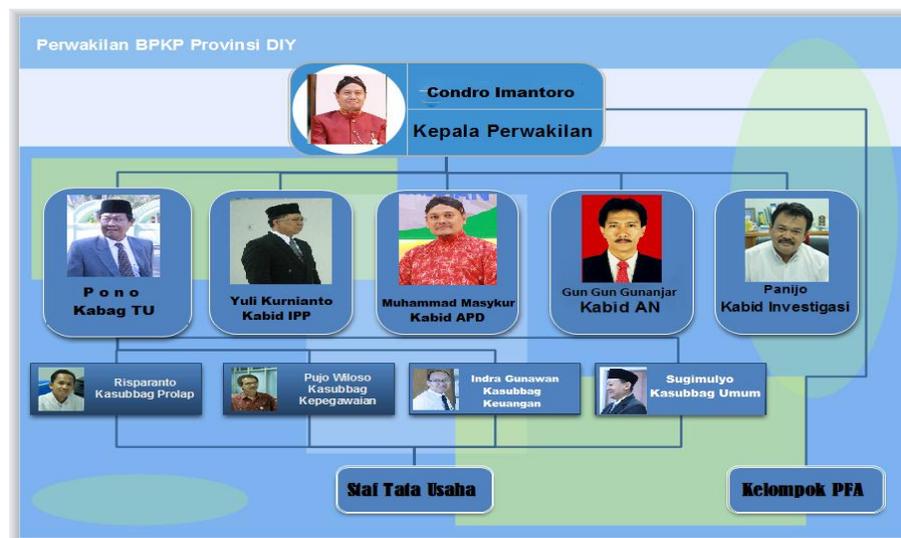
Perwakilan Madya di provinsi pemekaran. Struktur dari organisasi BPKP akan dijelaskan pada gambar 2.1



Gambar 2.1. Struktur organisasi BPKP

2.1.4 Struktur Organisasi BPKP DIY

Sedangkan struktur organisasi BPKP DIY dijelaskan pada gambar 2.2



Gambar 2.2. Struktur Organisasi BPKP DIY

Susunan Organisasi Perwakilan BPKP Provinsi D.I Yogyakarta terdiri dari :

Bagian Tata Usaha, terdiri dari :

1. Sub Bagian Program dan Pelaporan
2. Sub Bagian Kepegawaian
3. Sub Bagian Keuangan
4. Sub Bagian Umum
5. Bidang Pengawasan Intansi Pemerintah Pusat;
6. Bidang Akuntabilitas Pemerintah Daerah
7. Bidang Akuntan Negara
8. Bidang Investigasi

Selain struktur organisasi tersebut di atas, terdapat Kelompok Jabatan Fungsional mempunyai tugas melakukan kegiatan sesuai dengan jabatan fungsional masing-masing berdasarkan peraturan perundang-undangan yang berlaku. Di Perwakilan BPKP Provinsi D.I. Yogyakarta, Kelompok Jabatan Fungsional terdiri dari Jabatan Fungsional Auditor dan sejumlah jabatan fungsional lainnya yaitu Kelompok Jabatan Fungsional Analis Kepegawaian, Kelompok Jabatan Fungsional Arsiparis dan Kelompok Jabatan Fungsional Pranata Komputer.

2.2 Gambaran Umum Koperasi Karya Audita

2.2.1 Sejarah Singkat

Koperasi Karya Audita perwakilan BPKP Yogyakarta berdiri tahun 1997 dengan jumlah anggota awal hanya 25 orang. Kini, Koperasi Karya Audita telah berkembang hingga mencapai 180 orang anggota.

2.2.2 IDENTITAS

- a. Nama Koperasi : Koperasi Karya Audita Perwakilan BPKP DIY
- b. No dan Tgl B.Hk : No 178/BH/KWK.12/VII/1997 Tgl.14 Juli
1997
- c. Alamat : Jl. Parangtritis KM 5,5 Sewon Bantul. Telp. 0274-
385323

2.2.3 KEPENGURUSAN

- a. Susunan Pengurus Koperasi Karya Audita masa bakti 2007 –
2009 adalah sebagai berikut :

Ketua : Tahun 2007-2008 sdr Saji Hartana. Tahun 2009
sd saat ini sdr Basah Setiawan

Sekretaris : Emprah Handayani

Bendahara : Hartoyo

b. Susunan Badan Pengawas masa bakti 2007 -2009 :

Ketua : Drs. Chabib

Anggota : Dra. Sri Purwatinah

Ahmad Noorhidayat

2.2.4 PENGELOLA UNIT DAN KEKARYAWANAN

a. Pengelola Unit Toko : Hadiyani

Wiwik Widyaningrum

b. Pengelola Unit Akuntansi : Agus Waluyo

F.Rina

Yuni Diah

c. Keuangan : Puji Estriningsih

Sumiyani

d. Unit Simpan Pinjam: Hartoyo

Sri Widiyati

e. Karyawan Toko : Surati

BAB III

LANDASAN TEORI

3.1 Konsep Sistem Informasi

3.1.1 Sistem

Sebuah sistem menurut Kadir (2003) adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Putranta (2004) mengemukakan bahwa suatu sistem adalah sekumpulan dari subsistem-subsistem atau komponen-komponen yang saling berinteraksi atau saling bekerja sama.

3.1.2 Informasi

Menurut Davis (1999), informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang.

Data sering kali disebut sebagai bahan mentah informasi (Kadir, 2003). Melalui suatu proses transformasi, data dibuat menjadi lebih bermakna. Pengolahan data menjadi proses akan digambarkan oleh gambar 3.1.



Gambar 3.1 Transformasi data menjadi informasi

3.1.3 Sistem Informasi

Ada beberapa definisi mengenai sistem informasi. Sistem informasi adalah suatu sistem yang saling berintegrasi dan berinteraksi

atau berelasi satu sama lain, secara teratur, baik, dan rapi sehingga sistem dapat memberikan informasi yang bermanfaat yang dibutuhkan oleh pengguna dalam mengambil keputusan pada saat ini ataupun untuk masa mendatang (Supriyanto, 2005).

3.2 Konsep Basis Data

Basis data didefinisikan sebagai suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi dan mengatasi problem pada sistem yang memakai pendekatan berbasis berkas (Kadir, 2003).

Untuk mengelola basis data diperlukan sebuah perangkat lunak yang disebut DBMS (*Data Base Management System*). DBMS adalah perangkat lunak sistem yang memungkinkan pemakai membuat, memelihara, mengontrol, dan mengakses basis data dengan cara yang praktis dan efisien (Kadir, 2003). Beberapa contoh DBMS yang terkenal yaitu Access (Microsoft Corporation), Sybase (Sybase Inc.), MySQL (The MySQL AB Company), dan Oracle (Oracle Corporation).

3.2.1 Database Server dengan MySQL

MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi user* serta mampu menggunakan perintah standar SQL (*Structured Query Language*). MySQL selain sebagai *database server*, dapat juga berperan sebagai *client* sehingga sering disebut *database client-server*,

yang *open source* dengan kemampuan dapat berjalan baik di OS (*Operating System*) manapun, dengan *platform Windows* maupun LINUX (Nugroho, 2004).

3.2.2 Database MySQL dengan SQL

MySQL adalah sebuah program database server yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi user serta mampu menggunakan perintah standar SQL (*Structured Query Language*). MySQL selain sebagai *database server*, dapat juga berperan sebagai *client* sehingga sering disebut *database client-server*.

Menurut Kadir (2003), SQL singkatan dari *Structured Query Language* adalah bahasa yang digunakan untuk mengakses basis data yang tergolong relasional. SQL dibuat sebagai bahasa yang dapat merelasikan beberapa tabel dalam *database* maupun merelasikan antar *database* (Nugroho, 2004). Pada dasarnya, SQL tidak terbatas hanya untuk mengambil data (*query*), tetapi juga dapat digunakan untuk menciptakan tabel, menghapus tabel, menambahkan data ke tabel, menghapus data pada tabel, mengganti data pada tabel, dan berbagai operasi lainnya.

3.2.3 DBPaket

DBPaket adalah sebuah paket untuk melakukan akses dengan cara yang mudah dengan kelas yang sudah umum dipakai dalam

perancangan aplikasi database. Paket ini berisi objek dan kelas-kelas yang dipakai untuk keperluan akses ke dalam system database.

a. DriverManager

Kelas ini untuk menangani koneksi dengan driver DBMS, dan instant kelas ini akan selalu dibutuhkan oleh objek-objek yang terkait dengan database lainnya.

b. DBTable

Kelas ini dikhususkan untuk menangani hubungan dengan table saja.

c. DBQuery

Kelas ini hampir sama dengan kelas DBTable, karena diturunkan dari induk yang sama. Namun, kelas ini memiliki kelebihan khusus, yaitu mampu menjalankan perintah SQL (Insert, Update, dan Delete).

d. MyTable

Komponen ini adalah sebuah kontainer panel yang memuat kelas JScrollPane dan JTable dengan model tabelnya adalah MyModelTable. Komponen ini menampilkan data dari database ke dalam bentuk table dibungkus dalam satu kelas agar memudahkan dalam pengembangan dan mengurangi kompleksitas program.

3.3 JAVA

Sebagai sebuah bahasa pemrograman, Java dapat membuat seluruh bentuk aplikasi, desktop, web dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain.

Java adalah bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai platform sistem operasi. Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai system operasi dan bersifat open source.

Terdapat dua komponen utama dari *Deployment Environment*. Yang pertama adalah JRE (*Java Runtime Environment*), mengandung kelas-kelas untuk semua paket teknologi Java yang meliputi kelas dasar dari Java, komponen GUI (*Graphical User Interface*) dan sebagainya. Komponen yang lain terdapat pada Web Browser. Hampir seluruh Web Browser komersial menyediakan interpreter dan runtime environment dari teknologi Java.

3.4 UML (*Unified Modeling Language*)

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak.

UML cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET dapat juga digunakan untuk modeling aplikasi prosedural dalam VB atau C.

3.4.1. Use Case

Diagram ini memperlihatkan himpunan *Use Case* dan *actor-actor*. Diagram ini terutama sangat penting untuk memodelkan perilaku dari suatu system yang dibutuhkan serta diharapkan pengguna. Beberapa symbol yang digunakan antara lain:

1. Aktor

Aktor adalah pengguna sistem dan digambarkan seperti dibawah. Peran pengguna tertulis di bawah ikon. Pelaku tidak terbatas pada manusia. Jika suatu sistem berkomunikasi dengan aplikasi lain, dan akan memberikan masukan atau keluaran, maka aplikasi tersebut juga dapat dianggap sebagai aktor. Penggambaran aktor akan digambarkan oleh gambar 3.2.



Gambar 3.2 *Actor*

2. Use Case

Use Case adalah fungsi yang disediakan oleh sistem, (misalnya Daftar Mobil, Hapus Pengguna). *Use Case* digambarkan dengan elips. Nama *Use Case* ditulis dalam elips. Penggambaran dari simbol *Use Case* akan digambarkan oleh gambar 3.3.



Gambar 3.3 *Use Case*

3. Association

Asosiasi digunakan untuk menghubungkan Aktor dengan *Use Case*, dan menunjukkan bahwa sebuah Aktor berpartisipasi di dalam beberapa form *Use Case*. Asosiasi yang digambarkan oleh garis menghubungkan Aktor dan *Uses Case*. Penggambaran dari simbol *Association* akan digambarkan oleh gambar 3.4.



Gambar 3.4 *Association*

3.4.2. *Class Diagram*

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain (Huda dan Komputer, 2010).

Diagram ini memperlihatkan himpunan kelas, antarmuka, kolaborasi serta relasi.

1. *Class*

Kelas adalah bangunan blok di pemrograman berorientasi objek. Kelas digambarkan menggunakan persegi panjang dibagi menjadi tiga bagian. Bagian atas adalah nama Kelas. Bagian tengah mendefinisikan properti dari Kelas. Di bagian bawah daftar metode kelas.

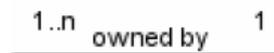
Penggambaran dari simbol *Class* akan digambarkan oleh gambar 3.5.



Gambar 3.5 *Class*

2. Association

Asosiasi adalah hubungan umum antara dua kelas, dan dimodelkan oleh baris menghubungkan dua kelas. Baris ini bisa memenuhi syarat dengan jenis hubungan, dan juga dapat fitur baragam aturan (misalnya, satu-ke-satu, satu-ke-banyak, banyak-ke-banyak) untuk hubungan. Penggambaran dari simbol *Association* akan digambarkan oleh gambar 3.6.



Gambar 3.6 *Association*

3. Composition

Jika satu kelas tidak dapat berdiri dengan sendirinya, melainkan harus menjadi anggota kelas lain, maka kelas tersebut memiliki hubungan Komposisi dengan kelas yang lain. Komposisi hubungan itu ditunjukkan dengan garis yang diisi dengan berlian. Penggambaran dari simbol *Composition* akan digambarkan oleh gambar 3.7.



Gambar 3.7 *Composition*

4. *Depedency*

Bila menggunakan kelas kelas yang lain, mungkin sebagai anggota variabel atau parameter, dan sebagainya "tergantung" di kelas itu, sebuah hubungan ketergantungan dibentuk. Ketergantungan hubungan itu ditunjukkan dengan panah garis putus putus. Penggambaran dari simbol *Depedency* akan digambarkan oleh gambar 3.8.



Gambar 3.8 *Depedency*

5. *Aggregation*

Aggregations menunjukkan seluruh bagian-hubungan, dan dikenal sebagai "memiliki-sebuah" hubungan. Penggabungan sebuah hubungan itu ditunjukkan dengan garis dengan berlian. Penggambaran dari simbol *Aggregation* akan digambarkan oleh gambar 3.9.



Gambar 3.9 *Aggregation*

6. *Generalization*

Hubungan generalisasi adalah sama dengan sebuah warisan hubungan berorientasi objek. Hubungan generalisasi ditunjukkan oleh sebuah panah dengan panah yang berongga pada, atau "orang tua", kelas. Penggambaran dari simbol *Generalization* akan digambarkan oleh gambar 3.10.



Gambar 3.10 *Generalization*

3.4.3. *Sequence Diagram*

Diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu. Beberapa symbol *Sequence Diagram* antara lain:

1. *Object*

Objek adalah contoh dari kelas, dan dapat diatur horizontal. Yang menggambarkan perwakilan untuk Objek adalah kotak dengan nama. Penggambaran dari simbol *Object* akan digambarkan oleh gambar 3.11.



Gambar 3.11 *Object*

2. *Actor*

Aktor juga dapat berkomunikasi dengan objek, sehingga mereka juga dapat terdaftar sebagai kolom. Aktor adalah sebuah model yang ada di mana-mana menggunakan simbol seperti gambar 3.12.



Gambar 3.12 *Actor*

3. *Lifeline*

Lifeline yang mengidentifikasi keberadaan objek dari waktu ke waktu. Penggambaran dari simbol *Lifeline* akan digambarkan oleh gambar 3.13.



Gambar 3.13 *Lifeline*

4. *Activation*

Activations sebagai model kotak segi empat di atas *lifeline*, menunjukkan bila objek melakukan tindakan. Penggambaran dari simbol *Activation* akan digambarkan oleh gambar 3.14.



Gambar 3.14 *Activation*

5. *Message*

Pesan, dimodelkan panah horisontal antara aktifitas, menunjukkan komunikasi antara objek. Penggambaran dari simbol *Message* akan digambarkan oleh gambar 3.15.



Gambar 3.15 *Message*

3.4.4. *Activity Diagram*

Diagram aktifitas ini terutama penting dalam pemodelan

fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek.

1. *Activity State*

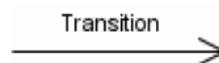
Aktifitas ditandai tindakan oleh obyek. Notasi untuk keadaan ini adalah bulatan. Penggambaran dari simbol *Action State* akan digambarkan oleh gambar 3.16.



Gambar 3.16 *Action State*

2. *Transition*

Bila aktifitas sudah selesai, proses berpindah ke aktifitas yang lain. Transisi digunakan untuk menandai gerakan ini. Transisi adalah model menggunakan panah. Penggambaran dari simbol *Transition* akan digambarkan oleh gambar 3.17.



Gambar 3.17 *Transition*

3. *Initial State*

Initial State menandai awal masuk dan titik awal. Notasinya lingkaran yang penuh. Hanya ada satu Initial State pada diagram. Penggambaran dari simbol *Initial State* akan digambarkan oleh gambar 3.18.



Gambar 3.18 *Initial State*

4. *Final State*

Final State menandai akhir model alur kerja. Ada beberapa *Final State* pada diagram, dan menggunakan model lingkaran yang penuh dikelilingi oleh lingkaran lain. Penggambaran dari simbol *Final State* akan digambarkan oleh gambar 3.19.



Gambar 3.19 *Final State*

5. *Decision State*

Decision state menandai pencabangan kondisi dari suatu proses. Penggambaran dari *Decision state* digambarkan oleh gambar 3.20.



Gambar 3.20 *Decision State*

6. *Synchronization Bar*

Kegiatan sering dapat dilakukan secara paralel. Untuk memecah proses ("*fork*"), atau untuk melanjutkan pemrosesan ketika beberapa kegiatan telah selesai ("*join*"), *Synchronization Bar* digunakan. Penggambaran dari *Synchronization Bar* digambarkan oleh gambar 3.21.



Gambar 3.21 *Synchronization Bar*

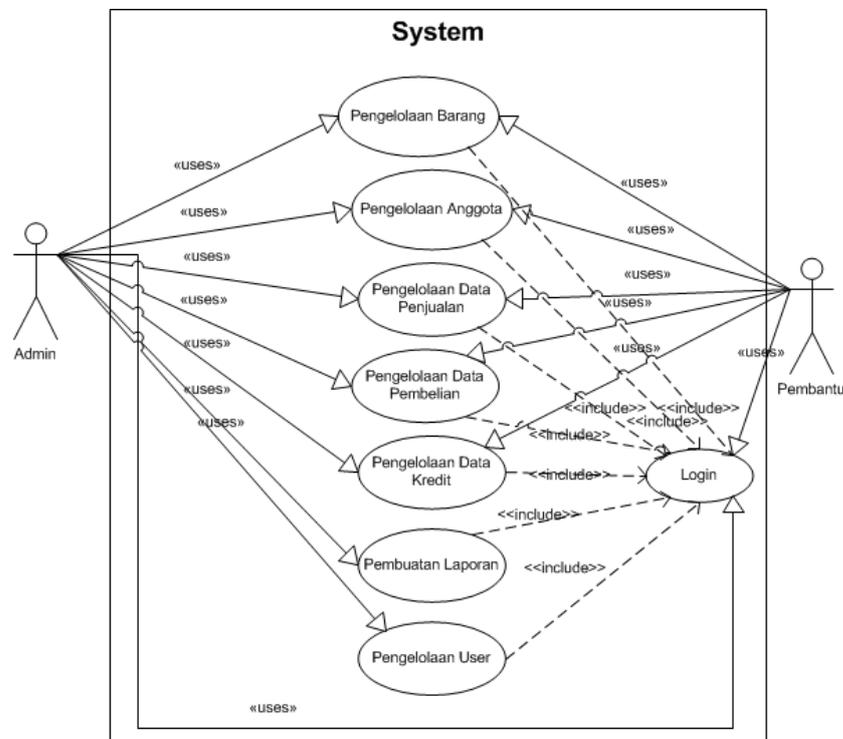
BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

4.1.1 Usecase Diagram

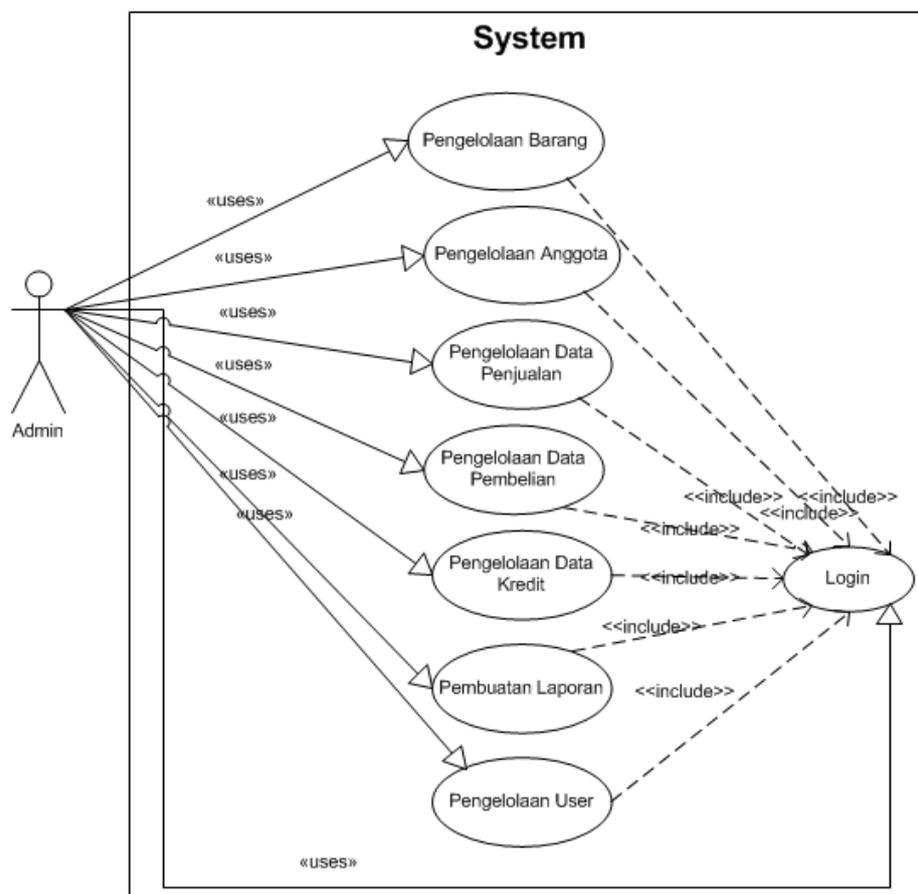
Fungsionalitas dari sistem penjualan Toko Koperasi yang mempresentasikan interaksi antara aktor (*Admin* atau *Pembantu*) dengan sistem dapat dilihat pada gambar 4.1



Gambar 4.1 Usecase diagram

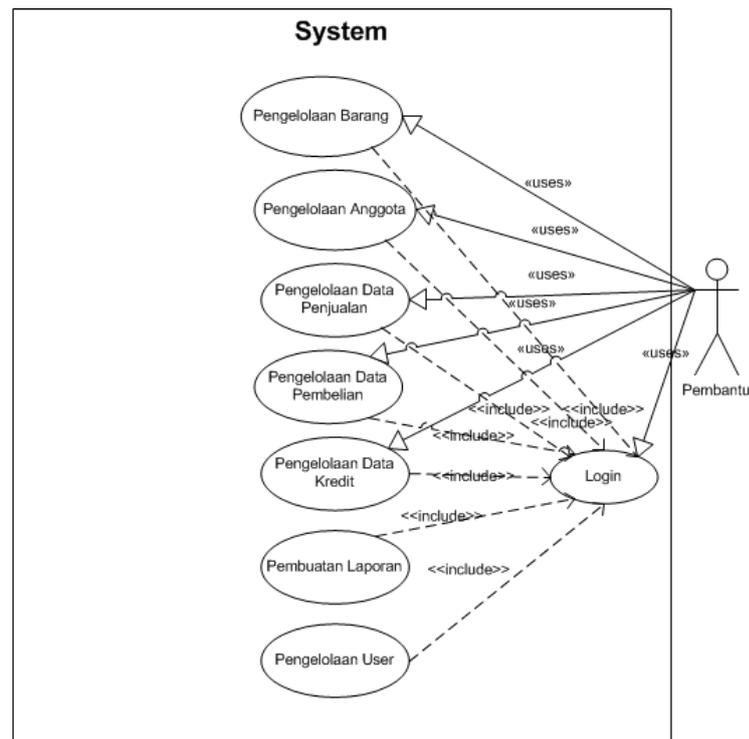
Dari gambar 4.1 dapat disimpulkan terdapat dua aktor dalam sistem Penjualan Toko Koperasi ini, yaitu *Admin* dan *Pembantu*. *Admin* dalam hal ini adalah Anggota dari Koperasi Karya Audita, sedangkan *Pembantu* adalah anggota Koperasi Karya Audita atau karyawan lepas. *Admin* dapat mengakses kedalam seluruh fasilitas yang ada didalam

aplikasi ini dengan cara *Login* sebagai *Admin* terlebih dahulu Untuk penjelasan hak aksesnya akan dijelaskan oleh gambar 4.2.



Gambar 4.2 Usecase diagram (*Admin*)

Sedangkan Pembantu dapat mengakses ke semua fasilitas aplikasi kecuali manajemen *User* dan tidak dapat mengakses fasilitas Laporan. Untuk penjelasan hak aksesnya akan dijelaskan oleh gambar 4.3.



Gambar 4.3 Usecase diagram (Pembantu)

A. Deskripsi Usecase

Berikut ini adalah penjelasan detail mengenai skenario dari masing-masing *usecase* melalui beberapa tabel deskripsi. Masing-masing tabel terdiri :

1. Skenario yang merupakan alur *usecase*.
2. *Brief description* adalah rincian deskripsi dari *usecase*
3. *Actor* menyatakan *actor* yang terlibat didalam *usecase*.
4. *Related Usecase* adalah *usecase* yang berhubungan dengan *usecase* tersebut.
5. *Precondition* adalah kondisi awal yang harus ada sebelum skenario dijalankan.
6. *Post Condition* adalah kondisi ketika skenario telah dijalankan.

7. *Flow of Event* adalah rangkaian kegiatan yang dilakukan aktor dan sistem yang terlibat.

8. *Exception Condition* adalah kondisi pengecualian yang diberikan ketika sistem tidak berjalan sesuai skenario yang diberikan.

Alur *usecase* untuk proses *login* akan dijelaskan oleh Tabel 4.1.

Tabel 4.1 Deskripsi *Usecase Login*

<i>Usecase Name</i>	<i>Login</i>	
<i>Skenario</i>	<i>Login ke Sistem Penjualan Toko Koperasi</i>	
<i>Brief Description</i>	<ul style="list-style-type: none"> - <i>Admin</i> memasukkan <i>Username</i> dan <i>password</i> pada <i>form Login</i>. - Sistem melakukan pengecekan <i>Username</i> dan <i>password</i> yang dimasukkan dengan data yang tersimpan di <i>database</i>. 	
<i>Actors</i>	<i>Admin, Pembantu</i>	
<i>Related Usecase</i>	-	
<i>Stakeholder</i>	Anggota Koperasi Karya Audita	
<i>Precondition</i>	- Data <i>Admin</i> harus sudah ada, didalam sistem sudah tersimpan data <i>Admin default</i> yang tersimpan..	
<i>PostCondition</i>	Fasilitas-fasilitas Sistem Penjualan Toko Koperasi akan terbuka	
<i>Flows of Events</i>	<i>Actors</i>	<i>System</i>
	<i>Admin</i> membuka Sistem Penjualan Toko Koperasi <i>Admin</i> memasukkan <i>Username</i> dan <i>password</i>	Sistem melakukan pengecekan <i>Username</i> dan <i>password</i> yang dimasukkan
<i>Exception Condition</i>	-	

Proses dimulai dari *Admin* memasukkan *username* dan *password*, setelah itu sistem akan melakukan pemeriksaan terhadap *username* dan *password* yang dimasukkan.

Sedangkan Tabel 4.2 akan menjelaskan *usecase* untuk mengelola barang.

Tabel 4.2 Deskripsi *Usecase* Mengelola Barang

<i>Usecase Name</i>	Mengelola barang	
Skenario	<ul style="list-style-type: none"> - <i>Admin</i> Menambah data barang - <i>Admin</i> Mengubah data barang - <i>Admin</i> Menghapus data barang 	
<i>Brief Description</i>	<ul style="list-style-type: none"> - <i>Admin</i> atau Pembantu membuka <i>form</i> barang - Sistem menampilkan data barang. - <i>Admin</i> atau pembantu mengisi data barang, kemudian menekan tombol tambah - Sistem akan menambahkan data barang tersebut ke dalam <i>database</i> - <i>Admin</i> atau pembantu memilih data barang pada tabel data, lalu mengubah data yang ada di <i>field</i> yang telah tersedia kemudian menekan tombol ubah. - Sistem akan menyimpan hasil perubahan data ke dalam <i>database</i> - <i>Admin</i> atau pembantu memilih data barang pada tabel data, kemudian klik tombol hapus. - Sistem akan menghapus data barang yang dipilih oleh <i>Admin</i> atau pembantu. 	
<i>Actors</i>	<i>Admin</i> , Pembantu	
<i>Related Usecase</i>	<i>Login</i>	
<i>Stakeholder</i>	Anggota Koperasi Karya Audita	
<i>Precondition</i>	Data barang yang ada dalam <i>database</i> sudah ditampilkan.	
<i>PostCondition</i>	Data barang bertambah Data barang berkurang Data barang berubah	
<i>Flows of Events</i>	Actors atau pembantu	<i>System</i>
	<ol style="list-style-type: none"> 1. <i>Admin</i> atau pembantu membuka <i>form</i> barang 2. <i>Admin</i> atau pembantu menambah data barang 	<ol style="list-style-type: none"> 1. Sistem menampilkan data barang 2. Sistem menyimpan data barang

	3. <i>Admin</i> atau pembantu mengubah data barang	baru 3. Sistem menyimpan hasil perubahan data barang
	4. <i>Admin</i> atau pembantu menghapus data barang	4. Sistem menghapus data barang
<i>Exception Condition</i>		

Admin atau pembantu dapat merubah, menambah serta menghapus data Anggota dari sistem. Untuk lebih jelas dari pengelolaan anggota akan dijelaskan oleh tabel 4.3.

Tabel 4.3 Deskripsi *Usecase* Mengelola Anggota

<i>Usecase Name</i>	Mengelola barang
Skenario	<ul style="list-style-type: none"> - <i>Admin</i> Menambah data Anggota - <i>Admin</i> Mengubah data Anggota - <i>Admin</i> Menghapus data Anggota
<i>Brief Description</i>	<ul style="list-style-type: none"> - <i>Admin</i> atau Pembantu membuka <i>form</i> Anggota - Sistem menampilkan data Anggota. - <i>Admin</i> atau pembantu mengisi data Anggota, kemudian menekan tombol tambah - Sistem akan menambahkan data Anggota tersebut ke dalam <i>database</i> - <i>Admin</i> atau pembantu memilih data Anggota pada tabel data, lalu mengubah data yang ada di <i>field</i> yang telah tersedia kemudian menekan tombol ubah. - Sistem akan menyimpan hasil perubahan data ke dalam <i>database</i> - <i>Admin</i> atau pembantu memilih data Anggota pada tabel data, kemudian klik tombol hapus. - Sistem akan menghapus data Anggota yang dipilih oleh <i>Admin</i> atau pembantu.
<i>Actors</i>	<i>Admin</i> , Pembantu
<i>Related Usecase</i>	<i>Login</i>

<i>Stakeholder</i>	Anggota Koperasi Karya Audita	
<i>Precondition</i>	Data Anggota yang ada dalam <i>database</i> sudah ditampilkan.	
<i>PostCondition</i>	Data Anggota bertambah Data Anggota berkurang Data Anggota berubah	
<i>Flows of Events</i>	<i>Actors</i> atau pembantu	<i>System</i>
	1. <i>Admin</i> atau pembantu membuka <i>form</i> Anggota 2. <i>Admin</i> atau pembantu menambah data Anggota. 3. <i>Admin</i> atau pembantu mengubah data Anggota 4. <i>Admin</i> atau pembantu menghapus data Anggota	1. Sistem menampilkan data Anggota 2. Sistem menyimpan data Anggota baru 3. Sistem menyimpan hasil perubahan data Anggota 4. Sistem menghapus data Anggota
<i>Exception Condition</i>		

Admin dapat menambahkan data kredit anggota baru, merubahnya, dan menghapusnya. Untuk lebih jelas akan dijelaskan oleh tabel 4.4.

Tabel 4.4 Deskripsi *Usecase* Mengelola Kredit.

<i>Usecase Name</i>	Mengelola Kredit
Skenario	<ul style="list-style-type: none"> - <i>Admin</i> Membuat kredit baru untuk anggota - <i>Admin</i> Mengubah data kredit anggota - <i>Admin</i> Menghapus data kredit
<i>Brief Description</i>	<ul style="list-style-type: none"> - <i>Admin</i> menekan tombol kredit - <i>Admin</i> memilih antara kredit baru dan pelunasan kredit, pilih kredit baru - <i>Admin</i> membuka <i>form</i> kredit - Untuk menambahkan kredit baru, <i>Admin</i> memasukkan data-data kredit baru yang akan dimasukkan, klik <i>submit</i>.

	<ul style="list-style-type: none"> - Untuk mengubah data, <i>Admin</i> memilih salah satu data yang ada di tabel, ubah data yang diperlukan, dan klik ubah. - Untuk menghapus data, pilih salah satu data yang ada di tabel, klik hapus. 	
<i>Actors</i>	<i>Admin/Pembantu</i>	
<i>Related Usecase</i>	<i>Login</i>	
<i>Stakeholder</i>	Anggota Koperasi Karya Audita	
<i>Precondition</i>	<ul style="list-style-type: none"> - Kredit Baru : - - Ubah Kredit : Data kredit yang telah ada. - Hapus Kredit : Data kredit yang telah ada 	
<i>PostCondition</i>	<ul style="list-style-type: none"> - Kredit Baru : Data kredit baru akan ditambahkan ke tabel. - Ubah Kredit : Data kredit lama akan berubah. - Hapus kredit : Data kredit lama akan terhapus 	
<i>Flows of Events</i>	Actors	System
	<ul style="list-style-type: none"> - Kredit Baru : 1. <i>Admin</i> menekan tombol Kredit. 2. <i>Admin</i> memilih kredit baru. 3. <i>Admin</i> memasukkan data-data kredit baru di <i>form</i> yang telah tersedia, lalu tekan tombol <i>submit</i>. 4. <i>Admin</i> memilih data yang ada, merubah data yg ada di <i>form</i> dan menekan tombol ubah untuk menyimpan perubahan. 	<ul style="list-style-type: none"> - Kredit Baru: 1.Sistem memunculkan pilihan kredit baru dan pelunasan. 2. Sistem menampilkan <i>form</i> kredit baru. 3. Sistem menyimpan data baru ke dalam <i>database</i>. 4. Sistem menyimpan perubahan data kedalam <i>database</i>.

	5. <i>Admin</i> memilih data lama yang berada di dalam tabel, tekan tombol hapus untuk menghapus.	5. Sistem akan menghapus data kredit yang ada di dalam <i>database</i> .
<i>Exception Condition</i>	-	

Tabel 4.4 menjelaskan mengenai *usecase* pengelolaan kredit.

Bagaimana cara *Admin* menambahkan data pelunasan kredit yang telah dibayar oleh anggota akan dijelaskan oleh tabel 4.5.

Tabel 4.5 Deskripsi *Usecase* Mengelola *password Admin*

<i>Usecase Name</i>	Pelunasan kredit
Skenario	<ul style="list-style-type: none"> - <i>Admin</i> Menambah data pelunasan kredit sembako. - <i>Admin</i> Menambah data pelunasan kredit konsinyasi.
<i>Brief Description</i>	<ul style="list-style-type: none"> - <i>Admin</i> menekan tombol kredit. - Sistem akan menampilkan pilihan antara kredit baru dan pelunasan kredit, pilih pelunasan kredit. - Sistem akan menampilkan <i>form</i> pelunasan kredit. - <i>Admin</i> memilih anggota yang bersangkutan dengan menekan tombol cari, pilih dari tabel yg muncul, dan tekan masukkan. - Setelah anggota terpilih, <i>Admin</i> memilih kredit yang akan dibayarkan, kredit sembako atau kredit konsinyasi. - Jika <i>Admin</i> memilih kredit sembako maka sistem otomatis akan menampilkan jumlah tagihan sembako untuk anggota yang terpilih. - <i>Admin</i> menekan tombol bayar untuk menambahkan data pelunasan kredit sembako. - Jika <i>Admin</i> memilih kredit konsinyasi, maka sistem akan memunculkan tabel konsinyasi untuk anggota terpilih. - <i>Admin</i> memilih kredit konsinyasi yang akan dilunasi dan tekan tombol masukkan. - Sistem akan menampilkan jumlah tertagih

	<p>untuk konsinyasi terpilih.</p> <ul style="list-style-type: none"> - <i>Admin</i> menekan tombol bayar untuk melunasi. 	
<i>Actors</i>	<i>Admin, Pembantu</i>	
<i>Related Usecase</i>	<i>Login</i>	
<i>Stakeholder</i>	Anggota Koperasi Karya Audita	
<i>Precondition</i>	Data Kredit yang telah ada di dalam <i>database</i> .	
<i>PostCondition</i>	Sistem menyimpan data pelunasan Kredit dalam <i>database</i> .	
<i>Flows of Events</i>	Actors	System
	<ol style="list-style-type: none"> 1. <i>Admin</i> menekan tombol kredit 2. <i>Admin</i> memilih pelunasan kredit. 3. <i>Admin</i> memilih anggota yang bersangkutan dengan menekan tombol cari. 4. <i>Admin</i> memilih anggota dengan memilih salah satu anggota di tabel dan tekan tombol masukkan. 5. <i>Admin</i> memilih kredit sembako. 6. <i>Admin</i> menekan tombol bayar. 	<ol style="list-style-type: none"> 1. Sistem menampilkan pilihan antara kredit baru dan pelunasan kredit 2. Sistem menampilkan <i>form</i> pelunasan kredit. 3. Sistem akan menampilkan tabel anggota. 4. Sistem akan menutup tabel anggota dan kembali ke <i>form</i> pelunasan dan menampilkan nomer dan nama anggota. 5. Sistem akan menampilkan data tertagih untuk kredit sembako anggota terpilih. 6. Sistem akan menambahkan data pelunasan kredit.

	<p>7. <i>Admin</i> memilih kredit konsinyasi.</p> <p>8. <i>Admin</i> memilih kredit yang akan dibayar dari tabel.</p> <p>9. <i>Admin</i> menekan tombol masukkan.</p> <p>10. <i>Admin</i> menekan tombol bayar.</p>	<p>7. Sistem akan menampilkan tabel konsinyasi.</p> <p>8. Sistem akan memilih data anggota.</p> <p>9. Sistem akan menutup tabel konsinyasi dan menampilkan data kredit tertagih.</p> <p>10. Sistem akan menambahkan data pelunasan kredit kedalam <i>database</i>.</p>
<i>Exception Condition</i>	-	

Untuk menambahkan data penjualan sembako, akan dijelaskan oleh tabel 4.6

Tabel 4.6 Deskripsi *Usecase* Menambahkan data penjualan.

<i>Usecase Name</i>	Menambah data penjalan.
Skenario	- <i>Admin</i> menambahkan data penjualan.
<i>Brief Description</i>	<ul style="list-style-type: none"> - <i>Admin</i> membuka <i>form</i> penjualan - Sistem menampilkan <i>form</i> penjualan - <i>Admin</i> memilih anggota. - Sistem menampilkan tabel anggota. - <i>Admin</i> memilih anggota ,kemudian menekan tombol masukkan. - Sistem memasukkan data anggota yang dipilih ke dalam <i>field</i> nomer dan nama anggota. - <i>Admin</i> memilih tipe pembelian, kredit atau tunai, jika barang yang dibeli langsung dibayar, maka pilih tunai, jika tidak maka kredit. - <i>Admin</i> memilih barang yang dijual ke anggota dengan menekan tombol cari

	barang. - Sistem akan menampilkan tabel barang. - <i>Admin</i> memilih barang, kemudian tekan tombol masukkan. - Sistem akan menutup tabel barang dan menampilkan spesifikasi barang. - <i>Admin</i> mengisi jumlah yang dijual. - <i>Admin</i> menekan tombol tambah. - Sistem akan menambahkan data barang jual sementara ke tabel penjualan. - Apabila ada barang yang ingin dibatalkan, maka pilih barang dan tekan tombol hapus. - Sistem akan menghapus data barang yang terpilih dari tabel penjualan. - Setelah selesai memilih barang terjual, maka <i>Admin</i> menekan tombol proses. - Sistem akan menambahkan data transaksi ke <i>database</i> dan mengurangi jumlah stok dari data barang yang terjual.	
<i>Actors</i>	<i>Admin</i> , Pembantu	
<i>Related Usecase</i>	<i>Login</i>	
<i>Stakeholder</i>	Anggota Koperasi Karya Audita	
<i>Precondition</i>	Sistem menampilkan <i>form</i> penjualan	
<i>PostCondition</i>	- Sistem menyimpan data transaksi penjualan. - Data stok barang akan berkurang.	
<i>Flows of Events</i>	Actors	System
	1. <i>Admin</i> menekan tombol penjualan. 2. <i>Admin</i> memilih anggota dengan menekan tombol anggota. 3. <i>Admin</i> memilih anggota yang ada di dalam tabel dan menekan tombol masukkan. 4. <i>Admin</i> menekan	1. Sistem menampilkan <i>form</i> penjualan. 2. Sistem menampilkan tabel anggota. 3. Sistem akan memilih data dan menampilkan di <i>form</i> penjualan. 4. Sistem

	<p>tombol cari barang.</p> <p>5. <i>Admin</i> memilih barang yang ada di dalam tabel barang dan menekan tombol masukkan.</p> <p>6. <i>Admin</i> mengisi jumlah yang diinginkan di <i>field</i> jumlah.</p> <p>7. <i>Admin</i> menekan tombol tambah.</p> <p>8. <i>Admin</i> menekan tombol proses.</p>	<p>menampilkan tabel barang.</p> <p>5. Sistem akan memilih data barang yang dipilih dan menampilkan di <i>form</i> penjualan.</p> <p>6. Sistem akan menyimpan <i>variable</i> jumlah.</p> <p>7. Sistem akan memindahkan data ke tabel penjualan.</p> <p>8. Sistem akan menyimpan data penjualan di <i>database</i> dan mengeluarkan konfirmasi transaksi.</p>
<i>Exception Condition</i>	-	

Tabel 4.6 menjelaskan alur *usecase* pemasukkan data transaksi penjualan baru.

Proses pembelian barang untuk stok koperasi akan dijelaskan oleh *usecase* pembelian. *Usecase* pembelian akan dijelaskan oleh tabel 4.7.

Tabel 4.7 Deskripsi *Usecase* menambahkan data pembelian.

<i>Usecase Name</i>	Pembelian
Skenario	<i>Admin</i> memasukkan data pembelian baru.
<i>Brief Description</i>	<ul style="list-style-type: none"> - <i>Admin</i> menekan tombol Pembelian. - Sistem akan menampilkan <i>form</i> pembelian. - <i>Admin</i> memilih barang yang dibeli dengan cara menekan tombol cari. - <i>Admin</i> memilih barang yang akan dibeli yang ada di dalam tabel.

	<p>Apabila barang yang dibeli belum terdapat di tabel, maka <i>Admin</i> menekan tombol tambah barang.</p> <ul style="list-style-type: none"> - <i>Admin</i> memasukkan data barang baru di <i>form</i> yang muncul dan menekan tombol tambah. - <i>Admin</i> memilih barang yang dibeli dan menekan tombol tambah. - Sistem menampilkan barang yang dipilih. - <i>Admin</i> memasukkan jumlah dari barang yang dibeli. - <i>Admin</i> memasukkan nama dari penjual dari barang yang dibeli. - <i>Admin</i> menekan tombol tambah. - Sistem akan menambahkan data di tabel pembelian. - <i>Admin</i> menekan tombol proses untuk mengakhiri pembelian. - Sistem akan menyimpan data pembelian dan menampilkan konfirmasi pembelian. 	
<i>Actors</i>	<i>Admin, Pembantu</i>	
<i>Related Usecase</i>	<i>Login</i>	
<i>Stakeholder</i>	Anggota Koperasi Karya Audita	
<i>Precondition</i>	-	
<i>PostCondition</i>	<ul style="list-style-type: none"> - Data barang baru - Data transaksi pembelian baru. 	
<i>Flows of Events</i>	<i>Actors</i>	<i>System</i>
	<ol style="list-style-type: none"> 1. <i>Admin</i> menekan tombol pembelian. 2. <i>Admin</i> menekan tombol cari. 3. <i>Admin</i> menekan tombol tambah barang. 4. <i>Admin</i> mengisi <i>form</i> tambah barang, lalu <i>Admin</i> menekan tombol tambah. 	<ol style="list-style-type: none"> 1. Sistem menampilkan <i>form</i> pembelian. 2. Sistem menampilkan tabel barang. 3. Sistem menampilkan <i>form</i> tambah barang. 4. sistem akan menambahkan data barang baru di <i>database</i>, system akan memunculkan data barang baru di

	<p>5. <i>Admin</i> memilih barang yang ada di tabel barang, lalu <i>Admin</i> menekan tombol masukkan.</p> <p>6. <i>Admin</i> mengisi jumlah barang dan penjual, lalu menekan tombol tambah.</p> <p>7. <i>Admin</i> menekan tombol proses.</p>	<p>tabel barang.</p> <p>5. Sistem akan memilih dan menampilkan data barang yang dipilih ke dalam <i>form</i> pembelian.</p> <p>6. Sistem akan menyimpan variabel jumlah dan penjual, lalu menampilkan di tabel pembelian.</p> <p>7. Sistem akan menyimpan data pembelian dan menampilkan konfirmasi pembelian.</p>
<i>Exception Condition</i>	-	

Usecase Laporan akan dijelaskan oleh tabel 4.8.

Tabel 4.8 Deskripsi *Usecase* Laporan

<i>Usecase</i> Name	Laporan
Skenario	<ul style="list-style-type: none"> - <i>Admin</i> membuat laporan stok barang - <i>Admin</i> membuat laporan laporan penjualan tahunan. - <i>Admin</i> membuat laporan laporan penjualan harian. - <i>Admin</i> membuat laporan penjualan per anggota. - <i>Admin</i> membuat laporan tagihan per anggota.
<i>Brief Description</i>	<ul style="list-style-type: none"> - <i>Admin</i> membuka <i>form</i> laporan. - Untuk membuat laporan stok barang tekan tombol Stok Barang. - Sistem otomatis akan membuat laporan stok barang. - Untuk membuat laporan penjualan tahunan, tekan tombol Laporan Penjualan Tahunan.

	<ul style="list-style-type: none"> - Masukkan tahun yang diinginkan, tekan - Sistem akan menampilkan laporan tahunan untuk tahun yang telah ditentukan. - Untuk laporan penjualan harian, tekan tombol penjualan harian. - Masukkan tanggal awal dari periode yang diinginkan, masukkan tanggal akhir periode yang diinginkan, tekan tombol ok. - Sistem akan menampilkan laporan harian dari tanggal yang telah ditentukan. - Untuk laporan penjualan per anggota, tekan tombol Laporan Penjualan per Anggota. - Masukkan nomer anggota, tanggal awal periode, dan tanggal akhir periode, lalu tekan ok. - Sistem akan menampilkan seluruh penjualan dari anggota yang ditentukan dari awal periode yang ditentukan hingga akhir periode yang ditentukan. - Untuk laporan tagihan anggota, tekan tombol Laporan Tagihan Anggota. - Masukkan nomer anggota, tanggal awal periode, dan tanggal akhir periode, lalu tekan ok. - Sistem akan menampilkan seluruh tagihan kredit dari anggota yang ditentukan dari awal periode yang ditentukan hingga akhir periode yang ditentukan. 		
<i>Actors</i>	<i>Admin, Pembantu</i>		
<i>Related Usecase</i>	<i>Login</i>		
<i>Stakeholder</i>	Anggota Koperasi Karya Audita		
<i>Precondition</i>	Data dari seluruh transaksi yang diperlukan harus sudah ada		
<i>PostCondition</i>	Laporan untuk masing-masing kategori.		
<i>Flows of Events</i>	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td><i>Actors</i></td> <td><i>System</i></td> </tr> </table>	<i>Actors</i>	<i>System</i>
<i>Actors</i>	<i>System</i>		

	<p>1. <i>Admin</i> menekan tombol Laporan.</p> <p>2. <i>Admin</i> menekan tombol Laporan Stok Barang.</p> <p>3. <i>Admin</i> menekan tombol Laporan Penjualan Tahunan.</p> <p>4. <i>Admin</i> memasukkan tahun, menekan tombol <i>ok</i>.</p> <p>5. <i>Admin</i> menekan tombol Laporan Penjualan Harian.</p> <p>6. <i>Admin</i> memasukkan nomer anggota, tanggal awal dan tanggal akhir, lalu menekan tombol <i>ok</i>.</p> <p>7. <i>Admin</i> menekan tombol Laporan Tagihan Anggota.</p> <p>8. <i>Admin</i> memasukkan nomer anggota, tanggal awal dan tanggal akhir, lalu menekan tombol <i>ok</i>.</p>	<p>1. Sistem menampilkan <i>form</i> laporan.</p> <p>2. Sistem menampilkan laporan stok barang terakhir.</p> <p>3. Sistem menampilkan <i>form</i> tahun periode.</p> <p>4. Sistem menampilkan laporan sesuai tahun yang ditentukan.</p> <p>5. Sistem menampilkan <i>form input</i> nomer anggota, tanggal awal, tanggal akhir.</p> <p>6. Sistem menampilkan laporan penjualan harian anggota sesuai dengan nomer anggota, tanggal awal dan akhir sesuai dengan yang ditentukan oleh <i>Admin</i>.</p> <p>7. Sistem menampilkan <i>form input</i> nomer anggota, tanggal awal, tanggal akhir.</p> <p>8. Sistem menampilkan laporan penjualan harian anggota sesuai dengan nomer anggota, tanggal awal dan akhir sesuai</p>
--	--	---

		dengan yang ditentukan oleh <i>Admin</i>
<i>Exception Condition</i>	-	

Alur *usecase* untuk proses manajemen *user* akan dijelaskan oleh

Tabel 4.9.

Tabel 4.9 Deskripsi *Usecase* Manajemen *User*

<i>Usecase Name</i>	Manajemen <i>User</i>
<i>Skenario</i>	<ul style="list-style-type: none"> - <i>Admin</i> melihat data <i>User</i>. - <i>Admin</i> menambah data <i>User</i>. - <i>Admin</i> menghapus data <i>User</i>. - <i>Admin</i> mengubah data <i>User</i>.
<i>Brief Description</i>	<ul style="list-style-type: none"> - <i>Admin</i> menekan tombol <i>User</i>. - Sistem akan menampilkan <i>form</i> manajemen <i>user</i>. - <i>Admin</i> mengisi <i>form</i> yang kosong, lalu menekan tombol tambah. - system akan menambahkan data <i>user</i> baru ke dalam <i>database</i>. - <i>Admin</i> memilih data di tabel <i>user</i>. - <i>Admin</i> merubah data yang terisi di <i>form user</i>. - <i>Admin</i> menekan tombol ubah. - Sistem akan merubah data yang ada dalam <i>database</i> dengan data baru. - <i>Admin</i> memilih data di tabel <i>user</i>. - <i>Admin</i> menekan tombol hapus. - Sistem akan menghapus data yang berada di <i>database</i> sesuai dengan data yang dipilih.
<i>Actors</i>	<i>Admin</i>
<i>Related Usecase</i>	<i>Login</i>
<i>Stakeholder</i>	Anggota Koperasi Karya Audita
<i>Precondition</i>	Data <i>user</i> harus sudah ada, didalam sistem sudah tersimpan data <i>Admin</i> default yang tersimpan..
<i>PostCondition</i>	<ul style="list-style-type: none"> - <i>User</i> baru akan terbuat. - Data <i>user</i> lama akan berubah.

	- <i>User</i> terhapus.	
<i>Flows of Events</i>	Actors	System
	<ol style="list-style-type: none"> 1. <i>Admin</i> atau pembantu membuka <i>form user</i>. 2. <i>Admin</i> menambah data <i>user</i>. 3. <i>Admin</i> mengubah data <i>user</i>. 4. <i>Admin</i> menghapus data <i>user</i>. 	<ol style="list-style-type: none"> 1. Sistem menampilkan data <i>user</i> 2. Sistem menyimpan data <i>user</i> baru 3. Sistem menyimpan hasil perubahan data <i>user</i> 4. Sistem menghapus data <i>user</i>
<i>Exception Condition</i>	-	

Untuk merubah *password* lama menjadi *password* baru dijelaskan pada tabel 4.10. *Admin* harus memasukkan *username* dan *password* lama sebelum melakukan perubahan *password*.

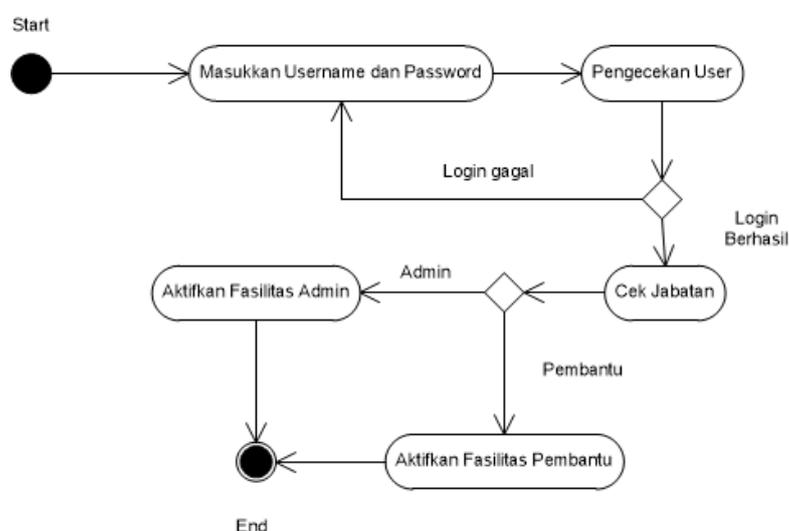
Tabel 4.10. Deskripsi *Usecase* Mengelola *password Admin*

<i>Usecase Name</i>	Mengelola <i>password Admin</i>
Skenario	- Mengubah <i>password</i>
<i>Brief Description</i>	<ul style="list-style-type: none"> - <i>Admin</i> membuka <i>form</i> ganti <i>password</i>. - Sistem akan menampilkan <i>form</i> ganti <i>password</i>. - <i>Admin</i> memasukkan <i>Username</i>, <i>password</i> lama, <i>password</i> baru dan konfirmasi <i>password</i> baru - Sistem menyimpan perubahan <i>password</i> pada <i>database</i>
<i>Actors</i>	<i>Admin</i>
<i>Related Usecase</i>	<i>Login</i>
<i>Stakeholder</i>	Anggota Koperasi Karya Audita
<i>Precondition</i>	Sistem menampilkan <i>form</i> ganti <i>password</i> .

<i>PostCondition</i>	Sistem menyimpan <i>password</i> baru <i>Admin</i>	
<i>Flows of Events</i>	Actors	System
	1. <i>Admin</i> membuka <i>form</i> ganti <i>password</i> . 2. <i>Admin</i> memasukkan <i>Username</i> , <i>password</i> lama, <i>password</i> baru, dan konfirmasi <i>password</i> baru	1. Sistem menampilkan <i>form</i> ganti <i>password</i> . 2. Sistem menyimpan <i>password</i> baru <i>Admin</i>
<i>Exception Condition</i>	-	

4.1.2 Activity Diagram

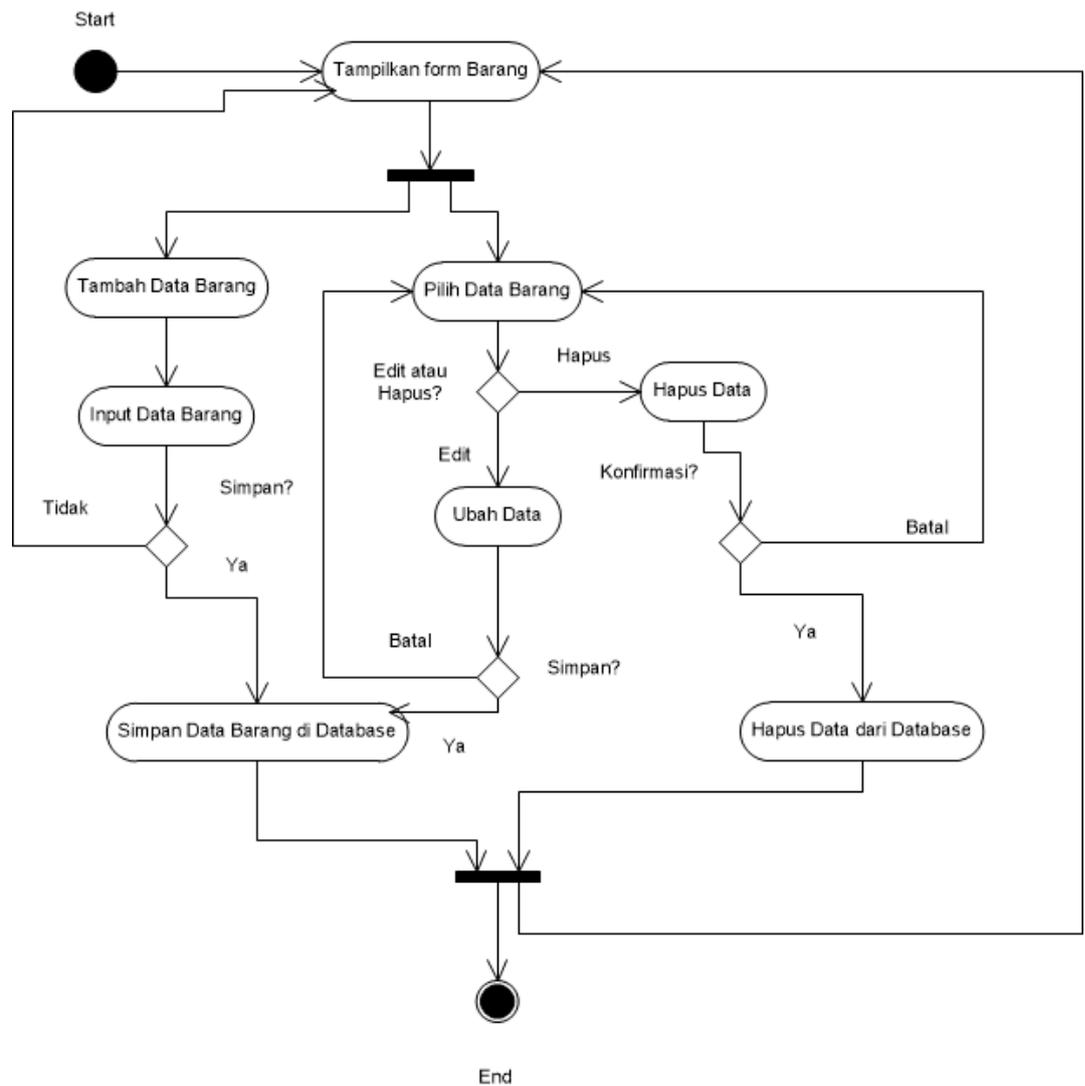
Activity diagram digunakan untuk menggambarkan *workflow* / aliran kerja proses dalam suatu sistem. *Activity* diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Activity dari proses *login* akan dijelaskan oleh gambar 4.4.



Gambar 4.4 Activity dari *Login*

Pada *Activity* ini, *Admin* memasukkan *username* dan *password*, jika *username* dan *password* cocok maka *Admin* akan berhasil *login*.

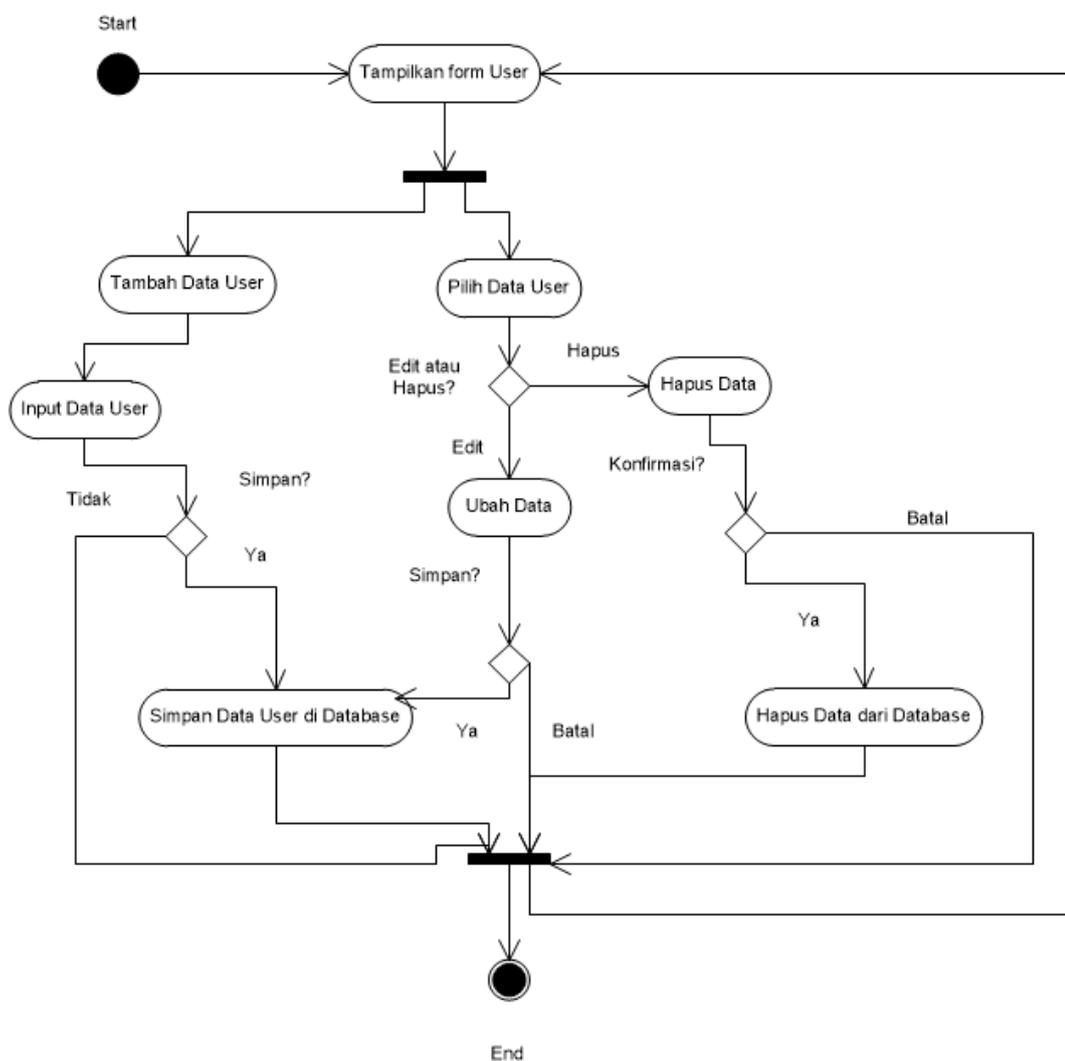
Activity dari manajemen data barang akan dijelaskan oleh gambar 4.5.



Gambar 4.5 *Activity* Manajemen Data Barang

Pada *Activity* manajemen data barang, *Admin* dapat menambah, mengubah, dan menghapus data barang.

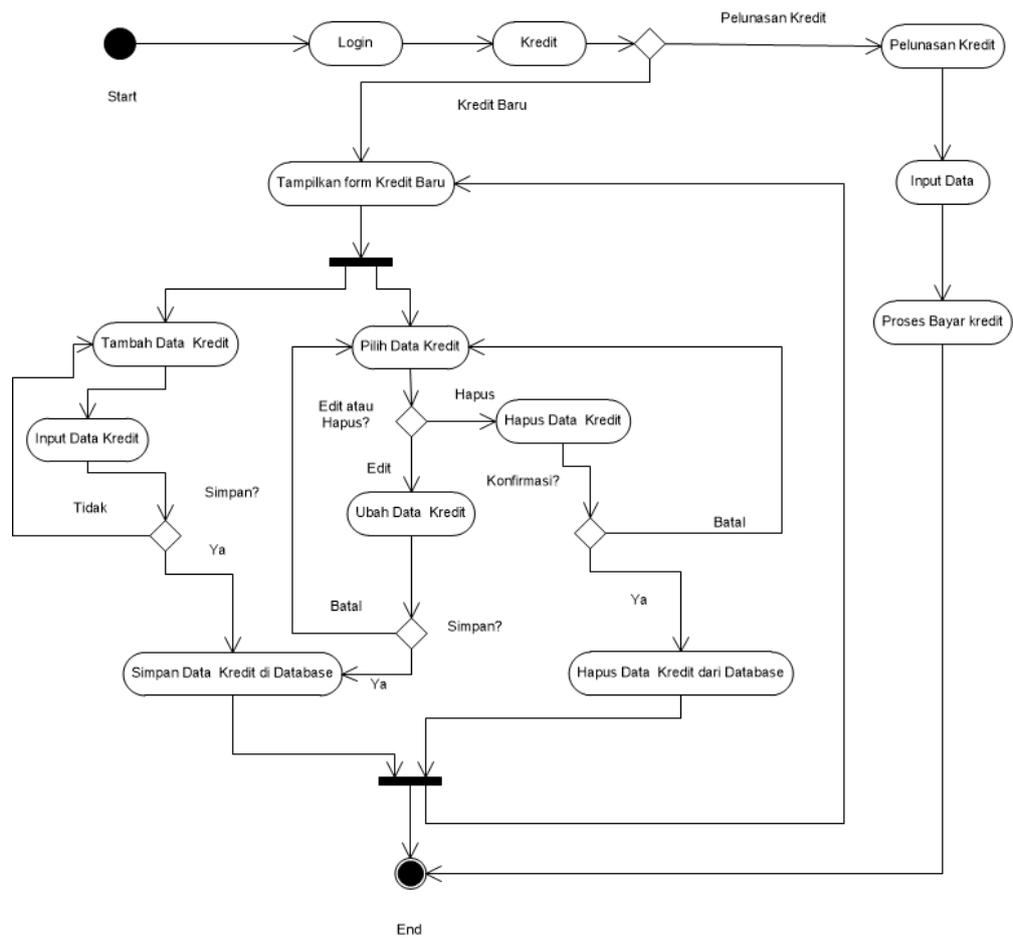
Activity dari manajemen Anggota akan dijelaskan oleh Gambar 4.6.



Gambar 4.6 Activity Manajemen Anggota

Pada *Activity* manajemen data anggota, *Admin* dapat menambah, mengubah, dan menghapus data anggota.

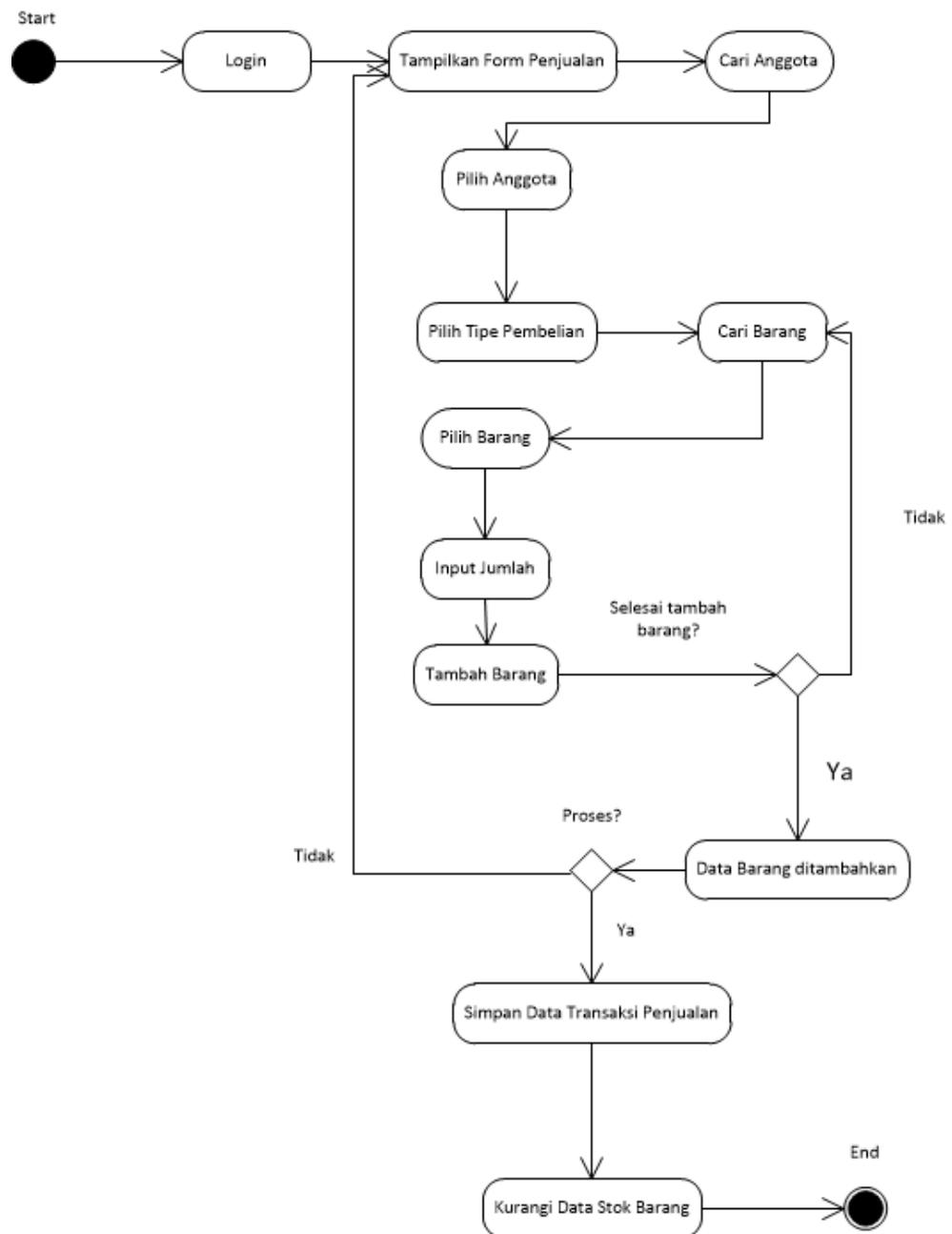
Activity dari pengelolaan kredit akan dijelaskan oleh gambar 4.7



Gambar 4.7 Activity Kredit

Pada *Activity* Kredit, *Admin* dapat menambahkan, mengubah, dan menghapus data kredit. Selain mengelola dari data kredit, *Admin* juga dapat menambahkan data pelunasan kredit.

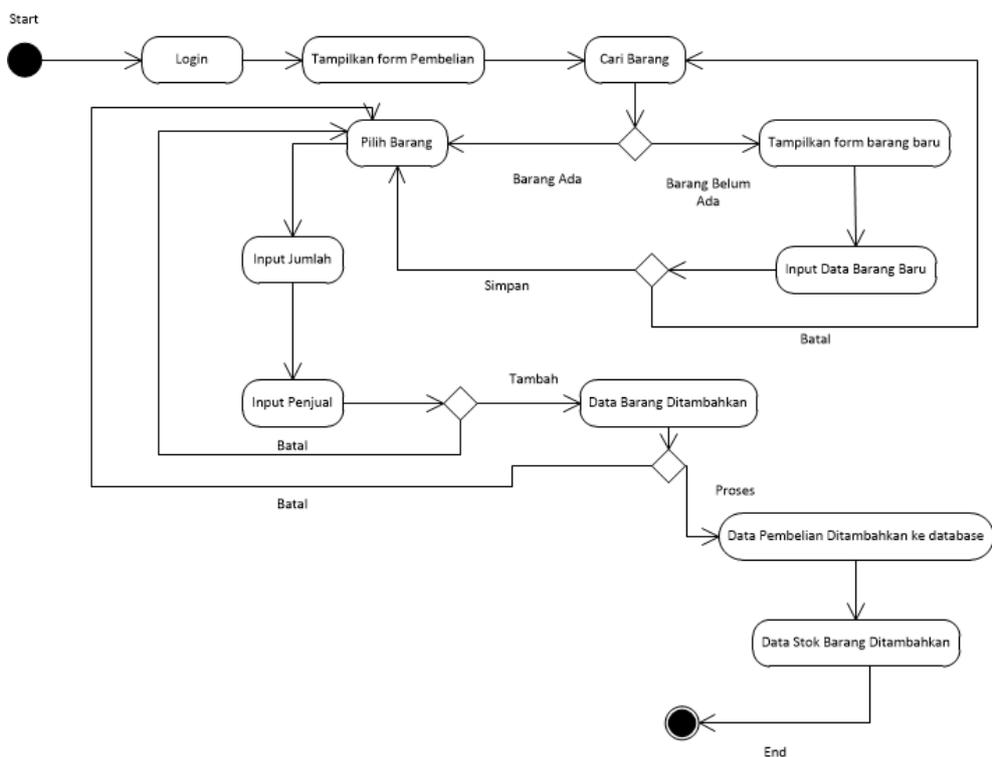
Activity dari penjualan akan dijelaskan oleh gambar 4.8.



Gambar 4.8 Activity Penjualan

Pada *Activity* Penjualan, *Admin* dapat menambahkan data transaksi penjualan baru, dan di dalam *Activity* ini, barang yang terkena transaksi akan dikurangi stok nya dari *database*.

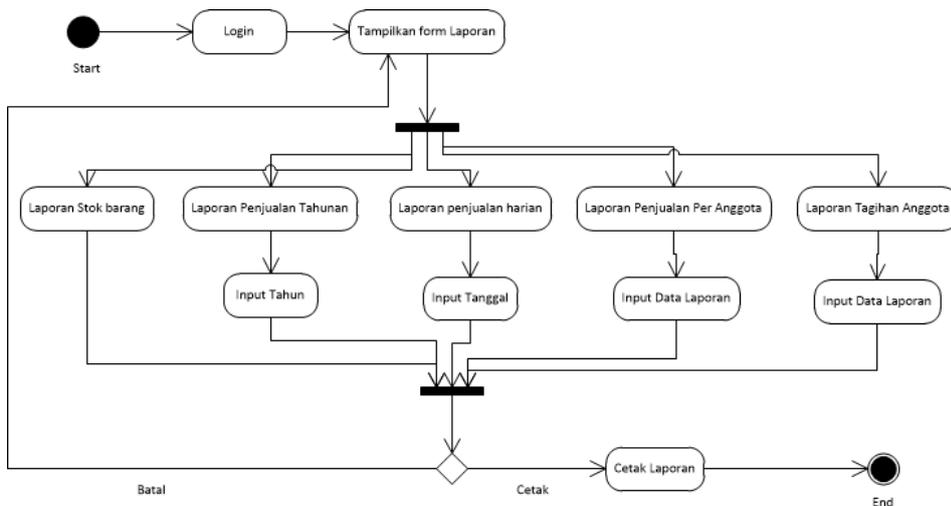
Activity dari transaksi pembelian akan dijelaskan oleh gambar 4.9



Gambar 4.9 Activity Pembelian

Pada *Activity* Pembelian, *Admin* dapat menambahkan data transaksi pembelian baru. Pada *Activity* ini, *Admin* juga dapat menambahkan data barang baru di *database*.

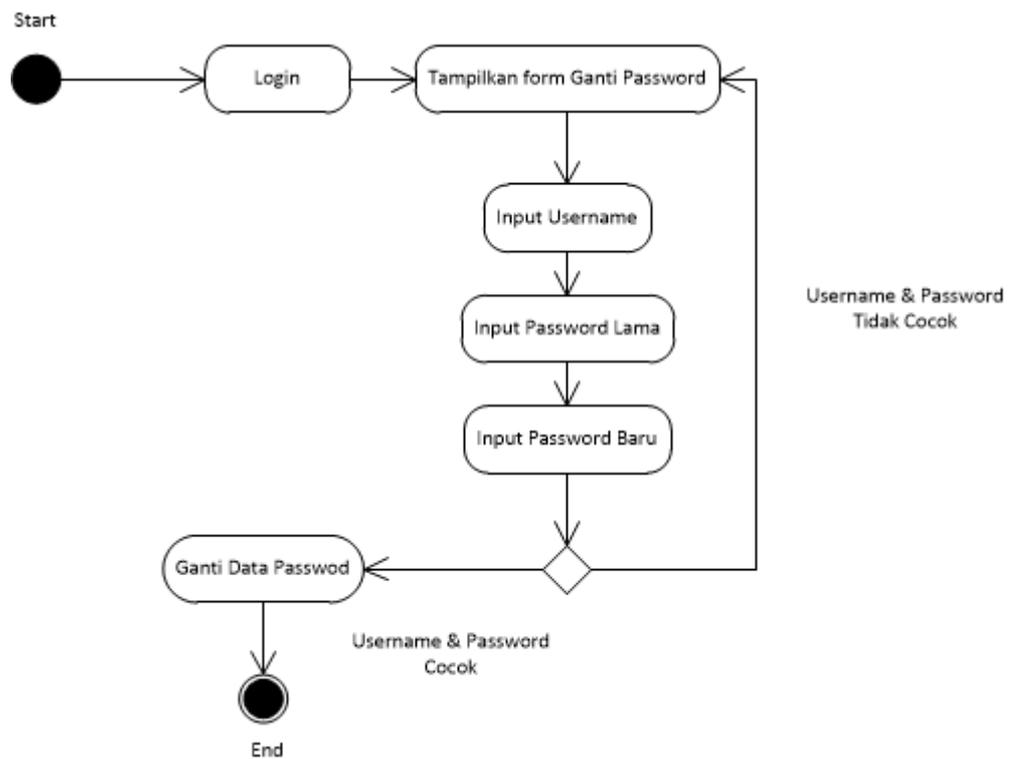
Activity dari Laporan akan dijelaskan oleh gambar 4.10.



Gambar 4.10 Activity Laporan

Pada Activity Laporan, *Admin* dapat membuat laporan dari beberapa bagian dari proses aplikasi ini.

Activity dari ganti *password* akan dijelaskan oleh gambar 4.11.



Gambar 4.11. *Activity* Ganti *Password*

4.1.3 Sequence Diagram

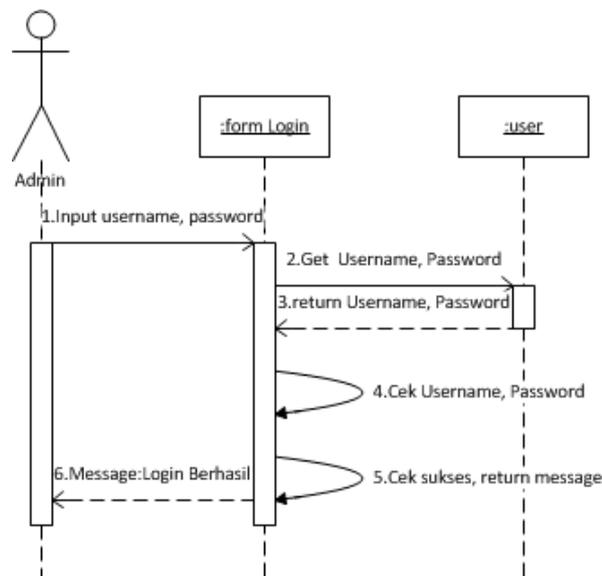
A. *Sequence Diagram Login*

Sequence Diagram proses *Login* menjelaskan alur untuk masuk bagi *Admin* ke *form* home. Proses yang terjadi antara lain:

1. *Admin* memasukkan *username* dan *password*
2. Sistem akan meminta data *user* dan *password* dari *database*.
3. *Database* akan mengirim balik *username* dan *password* yang ada.
4. Sistem melakukan verifikasi *login*.

5. Jika *login* berhasil, maka *login* return *true*.
6. Sistem akan menampilkan *login* berhasil

Sequence diagram *login* ditunjukkan pada gambar 4.12



Gambar 4.12. *Sequence* Diagram *Login*

B. *Sequence* Diagram Pengolahan Data Barang

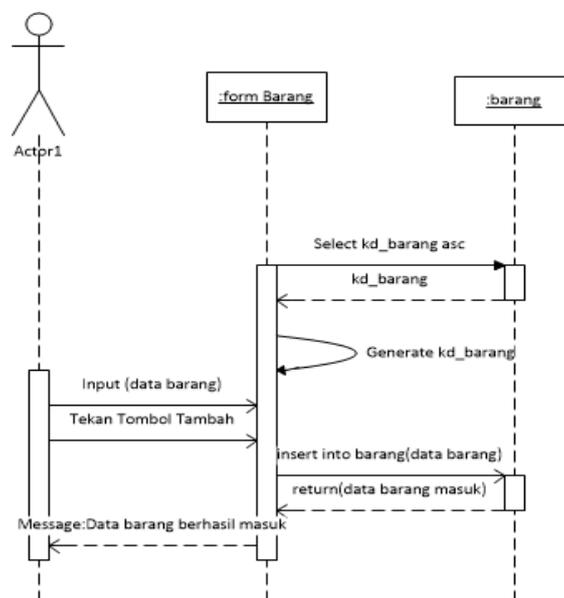
1. Menambah data barang

Sequence diagram menambah data barang menjelaskan alur memasukkan data-data barang ke dalam *database* sistem informasi Penjualan Koperasi. Berikut proses yang terjadi pada *Sequence* diagram menambah data barang:

1. Sistem akan meminta data *kd_barang*.
2. *Database* akan mengirim data *kd_barang*.
3. Sistem akan membuatkan *kd_barang* baru.

4. *Admin* memasukkan data barang baru.
5. *Admin* akan menekan tombol tambah.
6. Sistem mengirimkan data barang ke *database*.
7. *Database* mengirim *return true* jika data masuk.
8. Sistem akan mengirim *message* sukses ke *Admin*.

Sequence diagram menambah data barang ditunjukkan pada gambar 4.13



Gambar 4.13 *Sequence* Diagram menambah data barang

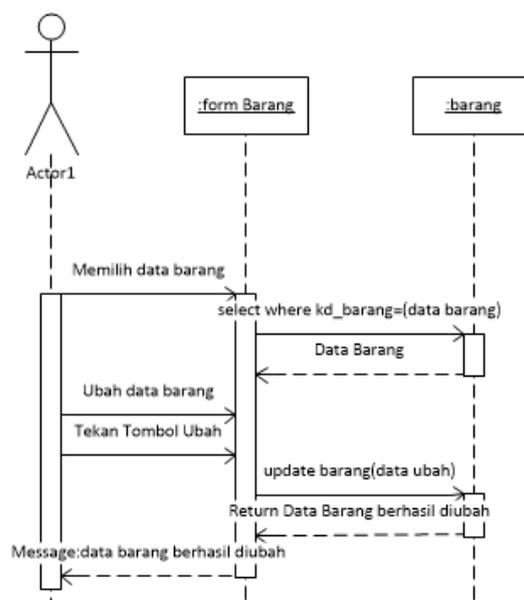
2. Mengubah data barang

Sequence diagram mengubah data barang menjelaskan mengenai alur memperbarui data barang. Berikut proses yang terjadi pada *Sequence* diagram mengubah data barang:

1. *Admin* memilih data barang
2. Sistem melakukan pemilihan data barang dari *database*.

3. *Database* mengirim data barang lengkap.
4. *Admin* mengubah data barang.
5. *Admin* Menekan tombol ubah.
6. Sistem mengirim perubahan data ke *database*.
7. Jika *database* berhasil diubah, *return* ubah=*true*.
8. Sistem mengirim *message* data barang berhasil diubah ke *Admin*.

Sequence diagram mengubah data barang ditunjukkan pada gambar 4.14



Gambar 4.14 *Sequence* diagram mengubah data barang

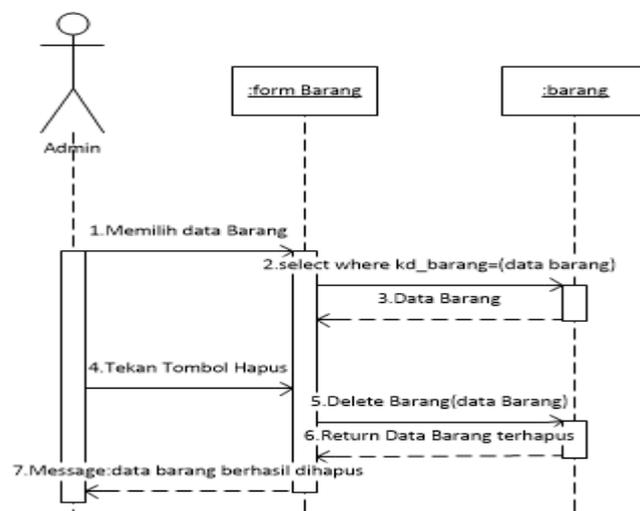
3.Menghapus data barang

Sequence diagram menghapus data barang menjelaskan mengenai alur menghapus data barang. Berikut proses yang terjadi pada *Sequence* diagram mengubah data barang:

1. *Admin* memilih data barang
2. Sistem mengirim permintaan data pilihan *Admin*.

3. *Database* mengirim data lengkap barang.
4. *Admin* menekan tombol hapus.
5. Sistem mengirim permintaan hapus barang ke *database*.
6. *Database* menghapus data barang.
7. Setelah proses hapus data yang berhasil, maka sistem akan mengirimkan konfirmasi hapus data berhasil.

Sequence diagram hapus barang ditunjukkan pada gambar 4.15



Gambar 4.15 *Sequence* diagram menghapus data barang

C. *Sequence* Diagram Pengolahan Data Anggota

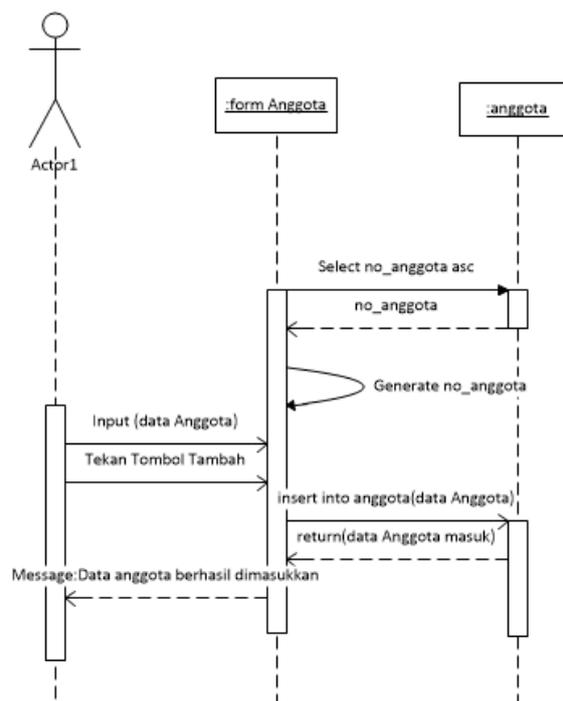
1. Menambah data Anggota

Sequence diagram menambah data anggota menjelaskan alur memasukkan data-data anggota ke dalam *database* sistem informasi penjualan koperasi. Berikut proses yang terjadi pada *Sequence* diagram menambah data anggota:

1. Sistem akan meminta data no_ang.
2. *Database* akan mengirim data no_ang.

3. Sistem akan membuat no_ang baru.
4. *Admin* memasukkan data anggota baru.
5. *Admin* akan menekan tombol tambah.
6. Sistem mengirimkan data anggota ke *database*.
7. *Database* akan mengirim *return true* jika data masuk.
8. Sistem akan mengirim *message* berhasil kepada *Admin*.

Sequence diagram menambah data anggota ditunjukkan pada gambar 4.16



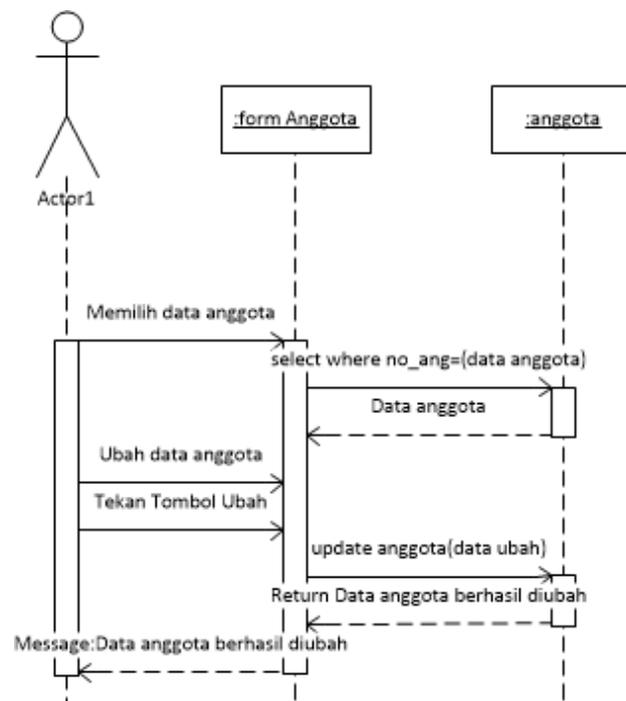
Gambar 4.16 *Sequence* Diagram menambah data anggota

2. Mengubah data anggota

Sequence diagram mengubah data anggota menjelaskan mengenai alur memperbarui data anggota. Berikut proses yang terjadi pada *Sequence* diagram mengubah data anggota:

1. *Admin* memilih data anggota
2. Sistem melakukan pemilihan data anggota dari *database*.
3. *Database* mengirim data anggota lengkap.
4. *Admin* mengubah data anggota.
5. *Admin* Menekan tombol ubah.
6. Sistem mengirim perubahan data ke *database*.
7. Jika *database* berhasil diubah, *return* *ubah=true*.
8. Sistem mengirim *message* data berhasil diubah ke *Admin*.

Sequence diagram mengubah data anggota ditunjukkan pada gambar 4.17



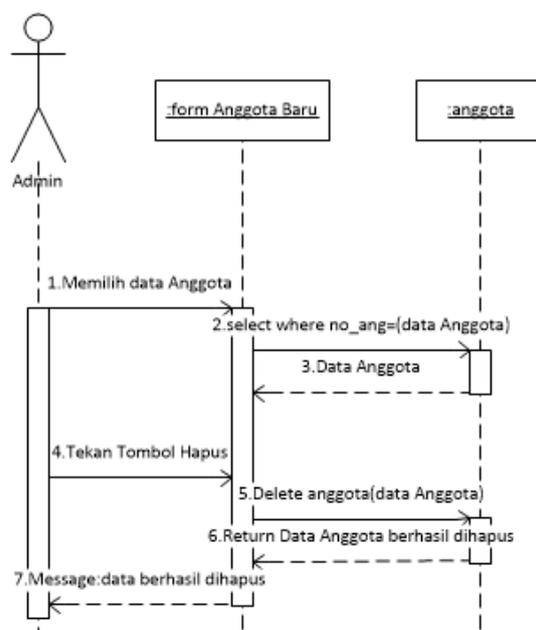
Gambar 4.17 *Sequence* diagram mengubah data anggota

3.Menghapus data anggota

Sequence diagram menghapus data anggota menjelaskan mengenai alur menghapus data anggota. Berikut proses yang terjadi pada *Sequence* diagram mengubah data anggota:

1. *Admin* memilih data anggota
2. Sistem mengirim permintaan data pilihan *Admin*.
3. *Database* mengirim data lengkap anggota.
4. *Admin* menekan tombol hapus.
5. Sistem mengirim permintaan hapus anggota ke *database*.
6. *Database* menghapus data anggota.
7. Setelah proses hapus data yang berhasil, maka sistem akan mengirimkan konfirmasi hapus data berhasil.

Sequence diagram hapus anggota dijelaskan pada gambar 4.12



Gambar 4.18 *Sequence* diagram menghapus data anggota

D. Sequence Diagram Pengolahan Data user

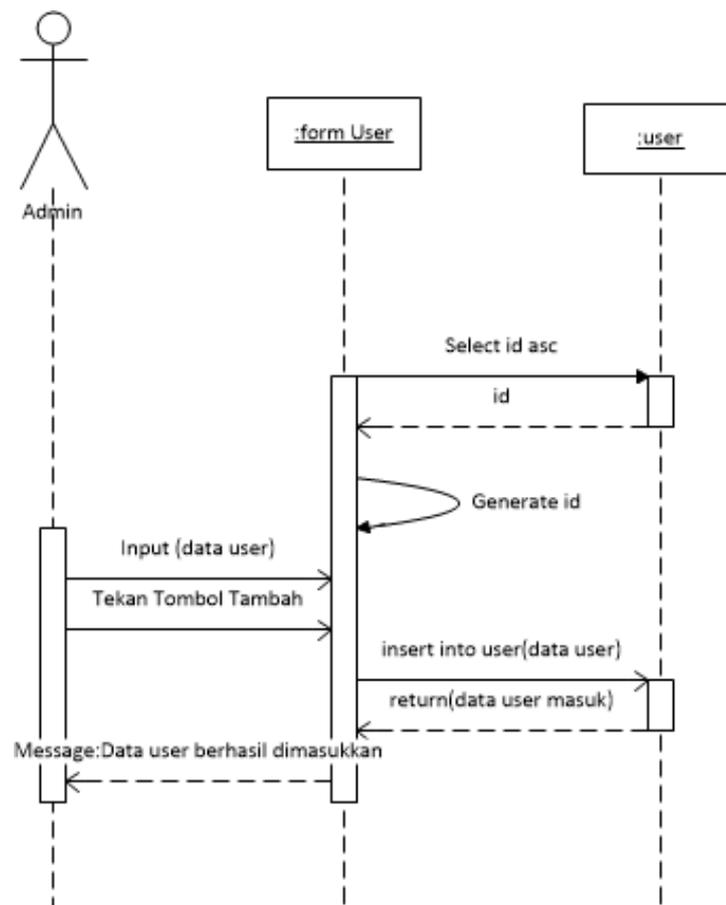
1. Menambah data user

Sequence diagram menambah data *user* menjelaskan alur memasukkan data-data *user* ke dalam *database* sistem informasi penjualan koperasi. Berikut proses yang terjadi pada *Sequence* diagram menambah data *user*:

1. Sistem akan meminta data id.
2. *Database* akan mengirim data id.
3. Sistem akan membuatkan id baru.
4. *Admin* memasukkan data *user* baru.
5. *Admin* akan menekan tombol tambah.
6. Sistem mengirimkan data *user* ke *database*.
7. *Database* akan mengirim *return true* jika data masuk.
8. Sistem akan mengirim *message* berhasil kepada *Admin*.

Sequence diagram tambah data *user* ditunjukkan pada gambar

4.19



Gambar 4.19 *Sequence Diagram* menambah data *user*

2. Mengubah data *user*

Sequence diagram mengubah data *user* menjelaskan mengenai alur memperbarui data *user*. Berikut proses yang terjadi pada *Sequence diagram* mengubah data *user*:

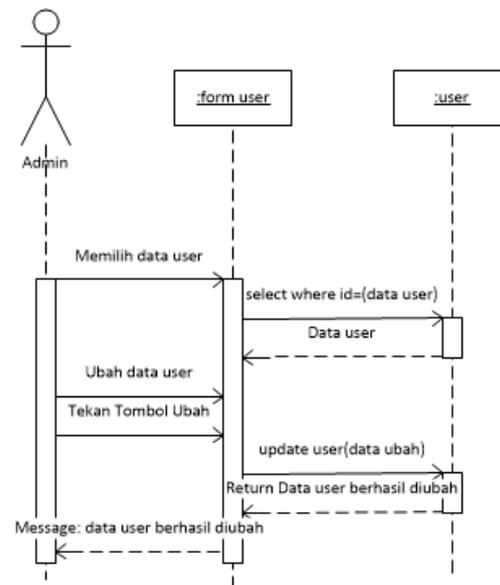
1. *Admin* memilih data *user*
2. Sistem melakukan pemilihan data *user* dari *database*.
3. *Database* mengirim data *user* lengkap.
4. *Admin* mengubah data *user*.
5. *Admin* Menekan tombol ubah.
6. Sistem mengirim perubahan data ke *database*.

7. Jika *database* berhasil diubah, *return* *ubah=true*.

8. Sistem mengirim *message* data *user* berhasil diubah ke *Admin*.

Sequence diagram ubah data *user* ditunjukkan pada gambar

4.20



Gambar 4.20 *Sequence* diagram mengubah data *user*

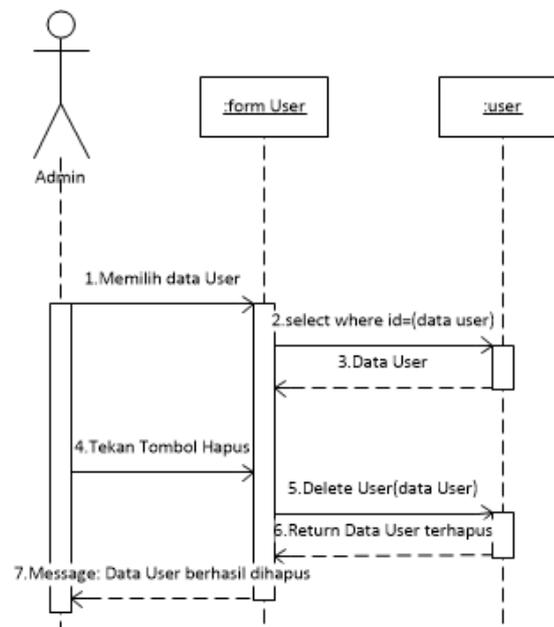
3. Menghapus data *user*

Sequence diagram menghapus data *user* menjelaskan mengenai alur menghapus data *user*. Berikut proses yang terjadi pada *Sequence* diagram mengubah data *user*:

1. *Admin* memilih data *user*
2. Sistem mengirim permintaan data pilihan *Admin*.
3. *Database* mengirim data lengkap *user*.
4. *Admin* menekan tombol hapus.
5. Sistem mengirim permintaan hapus *user* ke *database*.
6. *Database* menghapus data *user*.

7. Setelah proses hapus data yang berhasil, maka sistem akan mengirimkan konfirmasi hapus data berhasil.

Sequence diagram hapus *user* ditunjukkan pada gambar 4.21



Gambar 4.21 *Sequence* diagram menghapus data *user*

E. *Sequence* Diagram Pengolahan Data kredit

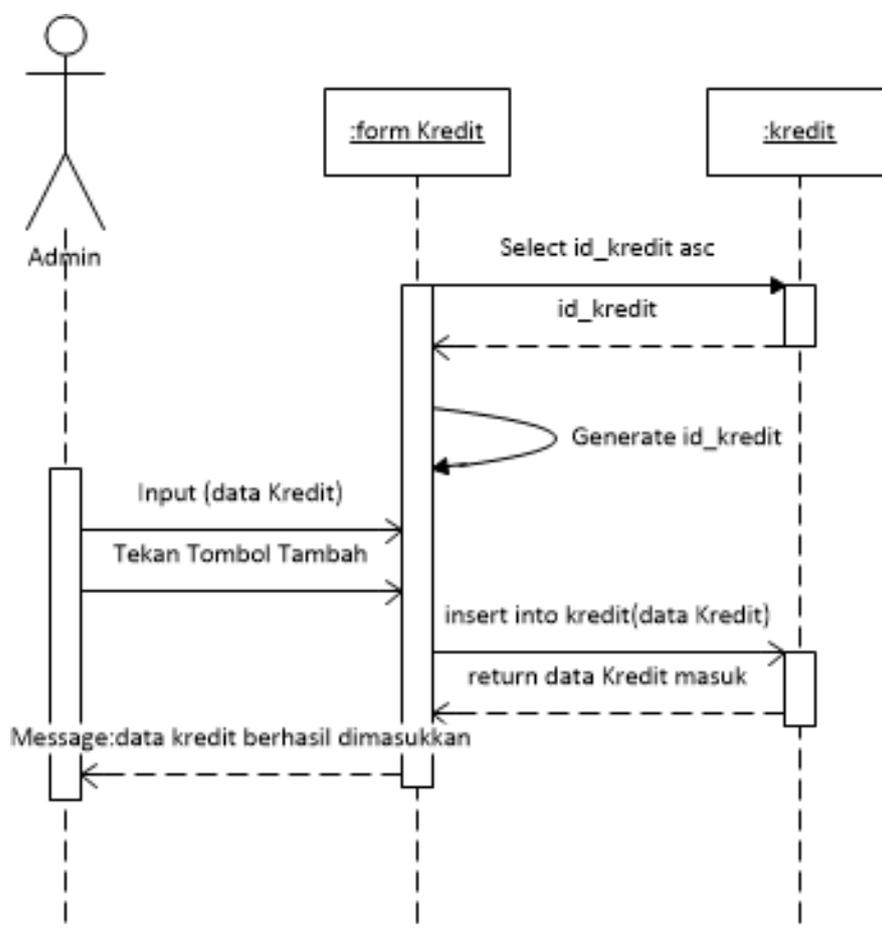
1. Menambah data kredit

Sequence diagram menambah data kredit menjelaskan alur memasukkan data-data kredit ke dalam *database* sistem informasi penjualan koperasi. Berikut proses yang terjadi pada *Sequence* diagram menambah data kredit:

1. Sistem akan meminta data `id_kredit`.
2. *Database* akan mengirim data `id_kredit`.
3. Sistem akan membuat `id_kredit` baru.

4. *Admin* memasukkan data kredit baru.
5. *Admin* akan menekan tombol tambah.
6. Sistem mengirimkan data kredit ke *database*.
7. *Database* akan mengirim *return true* jika data masuk.
8. Sistem akan mengirim *message* berhasil kepada *Admin*.

Sequence diagram tambah data kredit ditunjukkan pada gambar 4.22



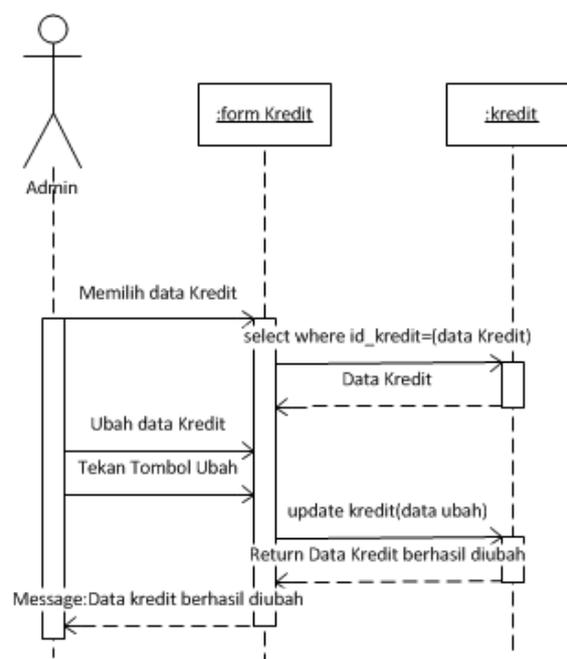
Gambar 4.22 *Sequence* Diagram menambah data kredit

2. Mengubah data kredit

Sequence diagram mengubah data kredit menjelaskan mengenai alur memperbarui data kredit. Berikut proses yang terjadi pada *Sequence* diagram mengubah data kredit:

1. *Admin* memilih data kredit
2. Sistem melakukan pemilihan data kredit dari *database*.
3. *Database* mengirim data kredit lengkap.
4. *Admin* mengubah data kredit.
5. *Admin* Menekan tombol ubah.
6. Sistem mengirim perubahan data ke *database*.
7. Jika *database* berhasil diubah, *return* *ubah=true*.
8. Sistem mengirim *message* data kredit berhasil diubah ke *Admin*.

Sequence diagram ubah data kredit ditunjukkan pada gambar 4.23



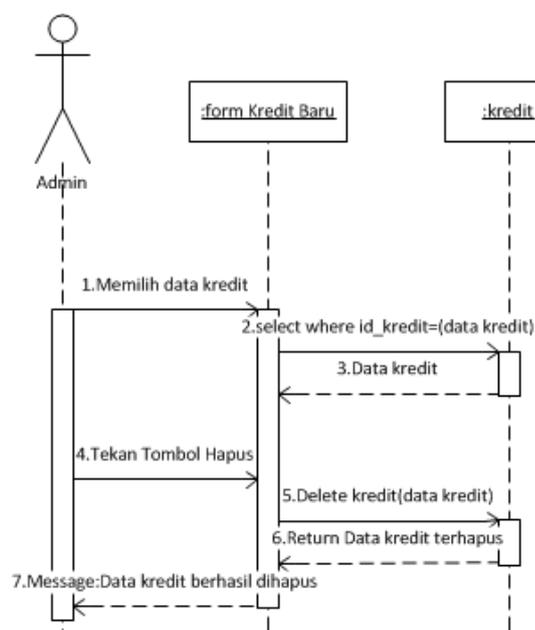
Gambar 4.23 *Sequence* diagram mengubah data kredit

3. Menghapus data kredit

Sequence diagram menghapus data kredit menjelaskan mengenai alur menghapus data kredit. Berikut proses yang terjadi pada *Sequence* diagram mengubah data kredit:

1. *Admin* memilih data kredit
2. Sistem mengirim permintaan data pilihan *Admin*.
3. *Database* mengirim data lengkap kredit.
4. *Admin* menekan tombol hapus.
5. Sistem mengirim permintaan hapus kredit ke *database*.
6. *Database* menghapus data kredit.
7. Setelah proses hapus data yang berhasil, maka sistem akan mengirimkan konfirmasi hapus data berhasil.

Sequence diagram hapus kredit dijelaskan pada gambar 4.24



Gambar 4.24 *Sequence* diagram menghapus data kredit

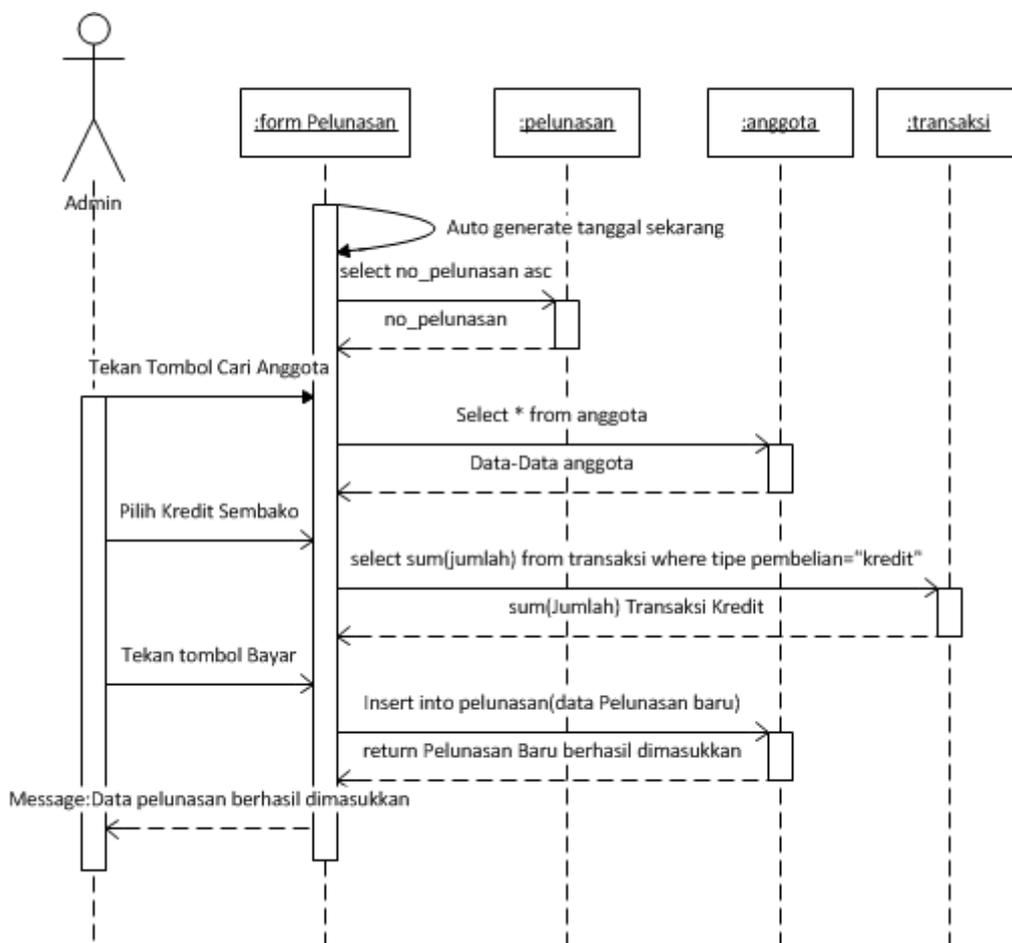
F. *Sequence Diagram* Pelunasan

1. *Sequence Diagram* Pelunasan Kredit Sembako

Sequence diagram pelunasan kredit menjelaskan alur memasukkan data-data pelunasan kredit ke dalam *database* sistem informasi penjualan koperasi. Berikut proses yang terjadi pada *Sequence diagram* menambah data kredit:

1. Sistem akan *auto generate* tanggal sekarang.
2. Sistem akan mengirim permintaan data *no_pelunasan*.
3. *Database* pelunasan akan mengirim data *no_pelunasan* ke sistem.
4. *Admin* menekan tombol cari anggota.
5. Sistem akan mengirim permintaan data anggota ke *database*..
6. *Database* mengirimkan data anggota ke sistem.
7. *Admin* memilih Kredit Sembako
8. Sistem akan mengirim permintaan data jumlah tanggungan ke *database*.
9. *Database* akan mengirim jumlah tanggungan sembako
10. *Admin* menekan tombol bayar.
11. Sistem akan mengirim data pelunasan kredit sembako ke *database*.
12. *Database* akan menyimpan data pelunasan kredit dan *return* masuk.
13. Sistem akan mengirim *message* berhasil kepada *Admin*.

Sequence diagram pelunasan kredit sembako ditunjukkan pada gambar 4.25



Gambar 4.25 *Sequence* Diagram Pelunasan kredit

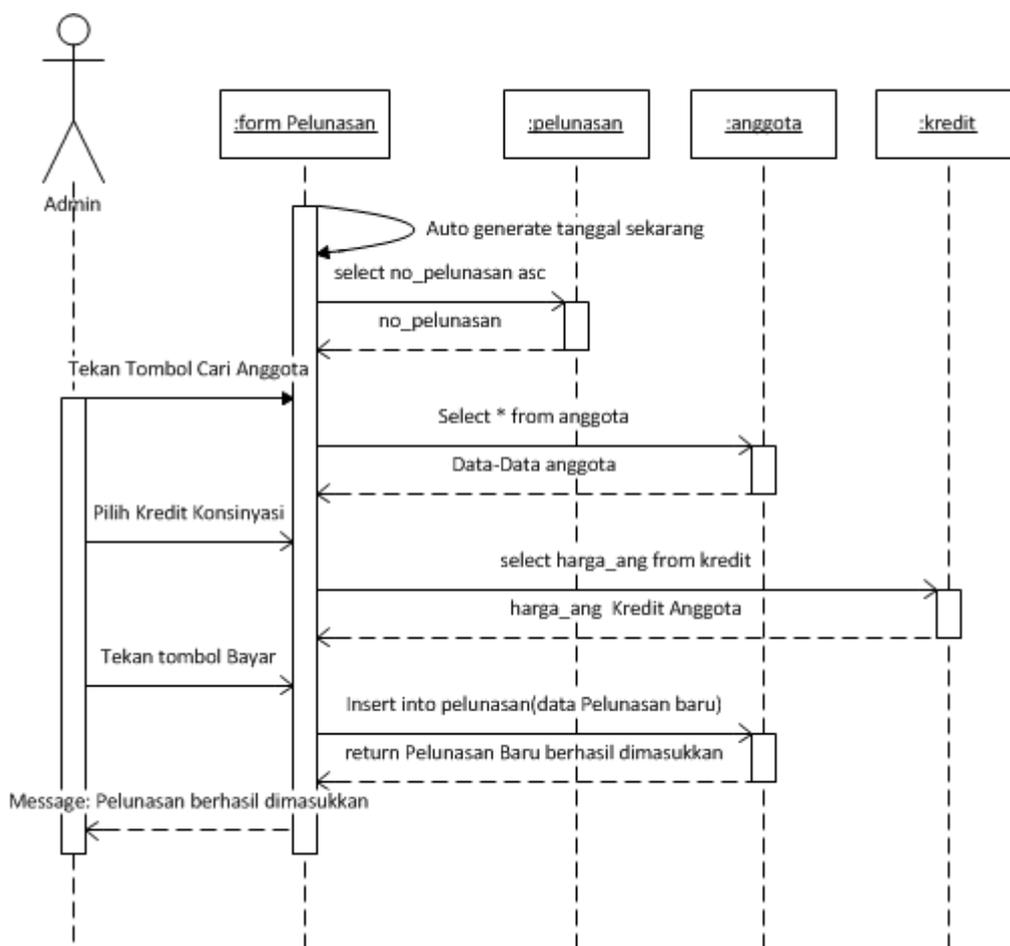
2. *Sequence* Diagram Pelunasan Kredit Konsinyasi

Sequence diagram pelunasan kredit konsinyasi menjelaskan alur memasukkan data-data pelunasan kredit konsinyasi ke dalam *database* sistem informasi penjualan koperasi. Berikut proses yang terjadi pada *Sequence* diagram menambah data kredit:

1. Sistem akan *auto generate* tanggal sekarang.

2. Sistem akan mengirim permintaan data no_pelunasan.
3. *Database* pelunasan akan mengirim data no_pelunasan ke sistem.
4. *Admin* menekan tombol cari anggota.
5. Sistem akan mengirim permintaan data anggota ke *database*..
6. *Database* mengirimkan data anggota ke sistem.
7. *Admin* memilih Kredit Konsinyasi.
8. Sistem akan mengirim permintaan data jumlah tanggungan kredit konsinyasi ke *database*.
9. *Database* akan mengirim jumlah tanggungan konsinyasi.
10. *Admin* menekan tombol bayar.
11. Sistem akan mengirim data pelunasan kredit konsinyasi ke *database*.
12. *Database* akan menyimpan data pelunasan kredit dan *return* masuk.
13. Sistem akan mengirim *message* berhasil kepada *Admin*.

Sequence diagram pelunasan kredit konsinyasi ditunjukkan pada gambar 4.26



Gambar 4.26 Sequence Diagram Pelunasan kredit Konsinyasi

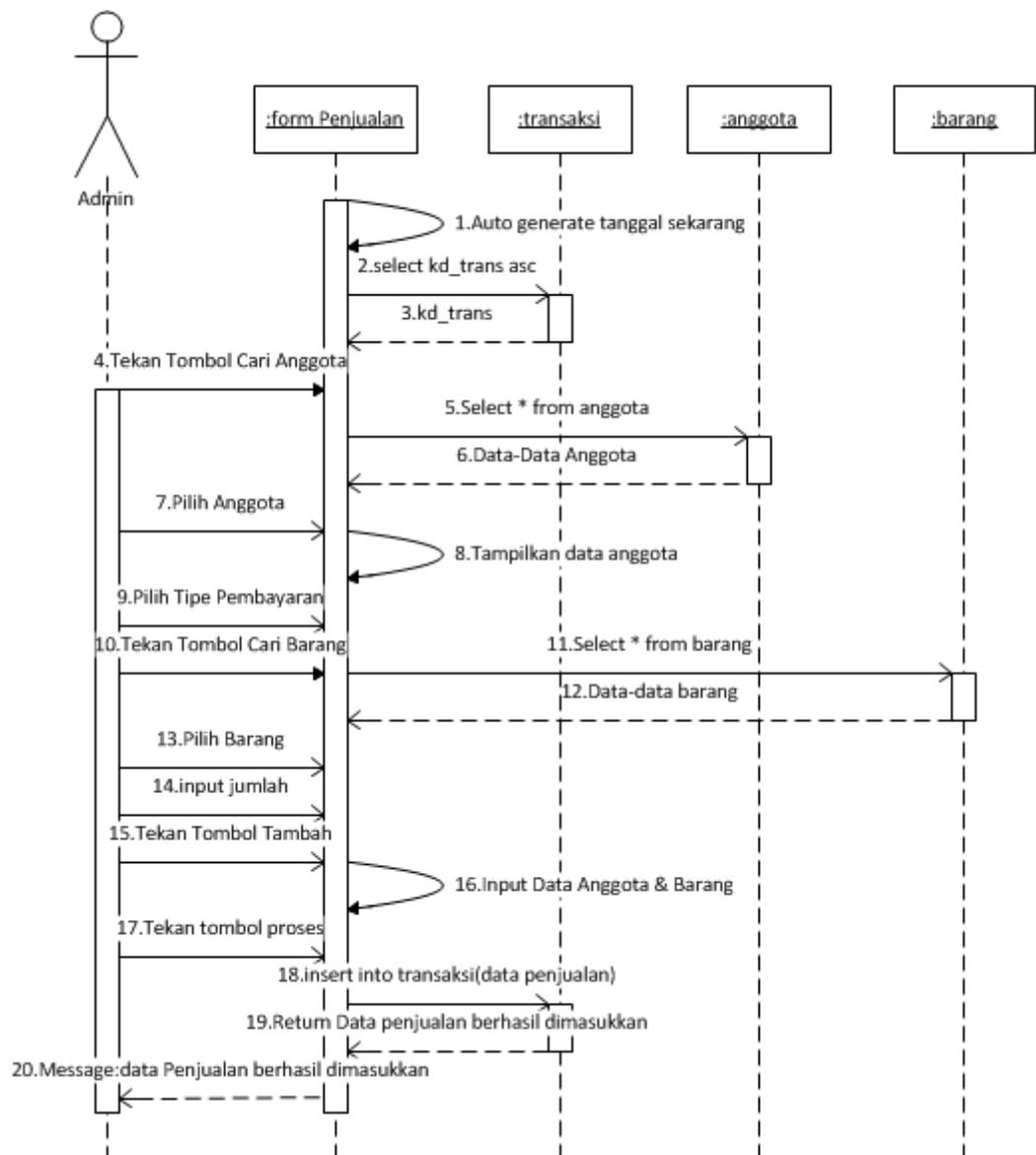
G. Sequence Diagram Penjualan

Sequence diagram penjualan menjelaskan alur memasukkan data-data penjualan ke dalam *database* sistem informasi Penjualan Koperasi. Berikut proses yang terjadi pada Sequence diagram Penjualan:

1. Sistem akan *auto generate* tanggal sekarang.
2. Sistem akan mengirim permintaan data kd_trans
3. *Database* akan mengirim data kd_trans ke system
4. *Admin* menekan tombol cari anggota.

5. Sistem akan mengirim permintaan data anggota ke *database*.
6. *Database* akan mengirim seluruh data anggota ke sistem.
7. *Admin* memilih anggota.
8. Sistem akan menampilkan data anggota yang dipilih.
9. *Admin* memilih tipe pembayaran.
10. *Admin* menekan tombol cari barang
11. Sistem akan mengirim permintaan data barang ke *database*.
12. *Database* akan mengirim seluruh data barang ke sistem.
13. *Admin* memilih barang.
14. *Admin* input jumlah barang
15. *Admin* menekan tombol tambah.
16. Data anggota dan barang akan disimpan di tabel sementara.
17. *Admin* menekan tombol proses.
18. Sistem akan memasukkan data transaksi ke *database*.
19. *Database* akan terisi oleh data transaksi baru dan *return true* masuk.
20. Setelah proses penyimpanan berhasil, maka sistem akan mengirimkan konfirmasi penyimpanan data berhasil.

Sequence diagram transaksi penjualan ditunjukkan pada gambar



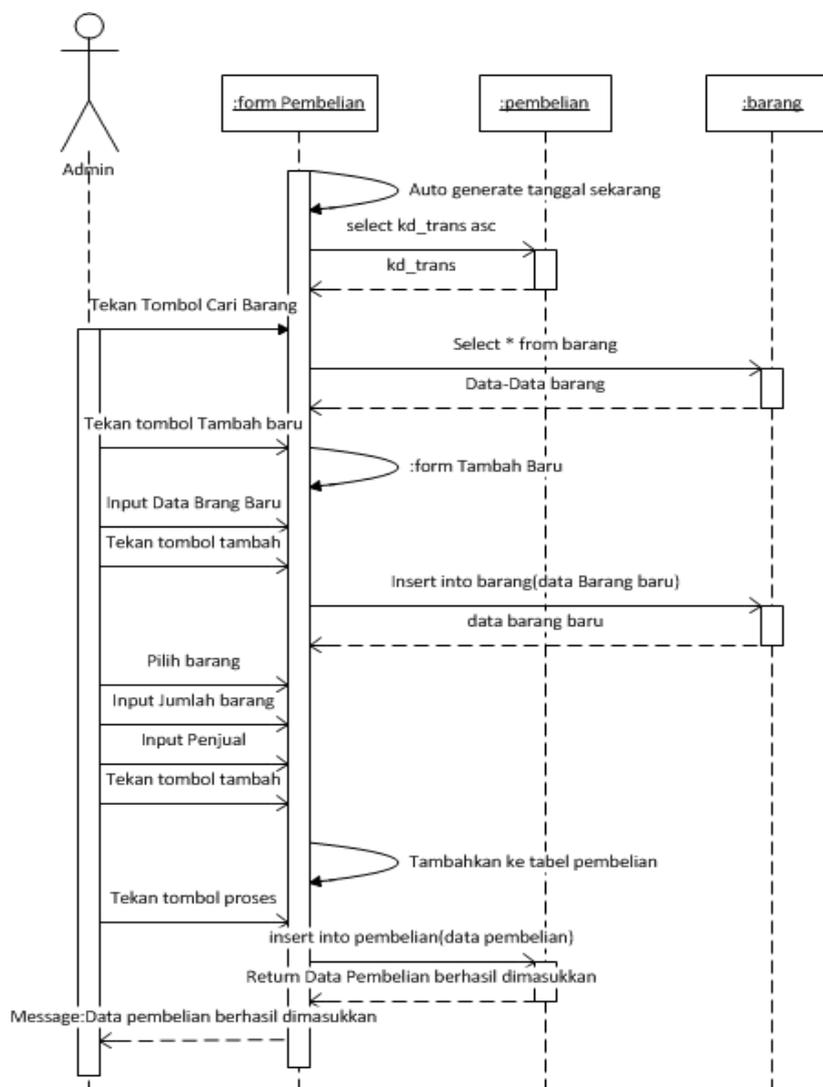
Gambar 4.27 Sequence diagram menambah data kategori

H. Sequence Diagram Pembelian

Sequence diagram penjualan menjelaskan alur memasukkan data-data pembelian ke dalam *database* sistem informasi Penjualan Koperasi. Berikut proses yang terjadi pada Sequence diagram Pembelian:

1. Sistem akan *auto generate* tanggal sekarang.
2. Sistem akan mengirim permintaan data *kd_trans*
3. *Database* akan mengirim data *kd_trans* ke system
4. *Admin* menekan tombol cari barang
5. Sistem akan mengirim permintaan data barang ke *database*.
6. *Database* akan mengirim seluruh data barang ke sistem.
7. *Admin* menekan tombol tambah barang.
8. Sistem akan membuka *form* tambah barang baru.
9. *Admin input* data barang baru.
10. *Admin* menekan tombol tambah.
11. Sistem akan memasukkan data barang baru ke *database*.
12. *Database* akan *ter-update* dan mengirim balik data *update*.
13. *Admin* memilih barang.
14. *Admin input* jumlah barang.
15. *Admin input* data penjual
16. *Admin* menekan tombol tambah.
17. Data barang akan disimpan di tabel sementara.
18. *Admin* menekan tombol proses.
19. Sistem akan memasukkan data transaksi pembelian ke *database*.
20. *Database* akan terisi oleh data transaksi baru dan *return true* masuk.
21. Setelah proses penyimpanan berhasil, maka sistem akan mengirimkan konfirmasi penyimpanan data berhasil.

Sequence diagram transaksi pembelian ditunjukkan pada gambar 4.28



Gambar 4.28 *Sequence* diagram menambah data kategori

I. *Sequence* Diagram Laporan

Sequence diagram laporan menjelaskan alur pembuatan laporan dari data-data yang ada dari dalam *database* sistem informasi Penjualan Koperasi. Berikut proses yang terjadi pada *Sequence* diagram Laporan:

1.1 *Admin* menekan tombol Laporan Stok Barang

1.2 Sistem akan mengirim permintaan sisa stok barang ke *database*.

1.3 *Database* akan mengirimkan data barang dan Sistem akan generate laporan

1.4 *Admin* memutuskan untuk mencetak.

1.5 Sistem akan mengirim perintah ke printer untuk mencetak laporan.

2.1 *Admin* membuka Laporan Penjualan Tahunan

2.2 *Admin input* tahun.

2.3 Sistem akan mengirim permintaan jumlah penjualan tahunan ke *database*.

2.4 *Database* akan mengirimkan data penjualan tahunan dan Sistem akan generate laporan

2.5 *Admin* memutuskan untuk mencetak.

2.6 Sistem akan mengirim perintah ke printer untuk mencetak

3.1 *Admin* membuka Laporan Penjualan Harian

3.2 *Admin input* tanggal periode.

3.3 Sistem akan meminta jumlah penjualan harian ke *database*.

3.4 Sistem akan generate laporan

3.5 *Admin* memutuskan untuk mencetak.

3.6 Sistem akan mengirim perintah ke printer

4.1 *Admin* membuka Laporan Penjualan per Anggota

4.2 *Admin input* nomer anggota & tanggal periode.

4.3 Sistem akan mengirim permintaan data ke *database*.

4.4 *Database* akan mengirimkan data penjualan

4.5 *Admin* memutuskan untuk mencetak.

4.6 Sistem akan mengirim perintah ke printer untuk mencetak

5.1 *Admin* membuka Laporan Tagihan Anggota

5.2 *Admin input* nomer anggota & tanggal periode.

5.3 Sistem akan mengirim permintaan data ke *database*.

5.4 *Database* akan mengirimkan data Tagihan Anggota dan Sistem akan generate laporan

5.5 *Admin* memutuskan untuk mencetak.

5.6 Sistem akan mengirim perintah ke printer untuk mencetak

6.1 *Admin* membuka Laporan Pelunasan Kredit

6.2 *Admin input* nomer anggota & tanggal periode.

6.3 Sistem akan mengirim permintaan data ke *database*.

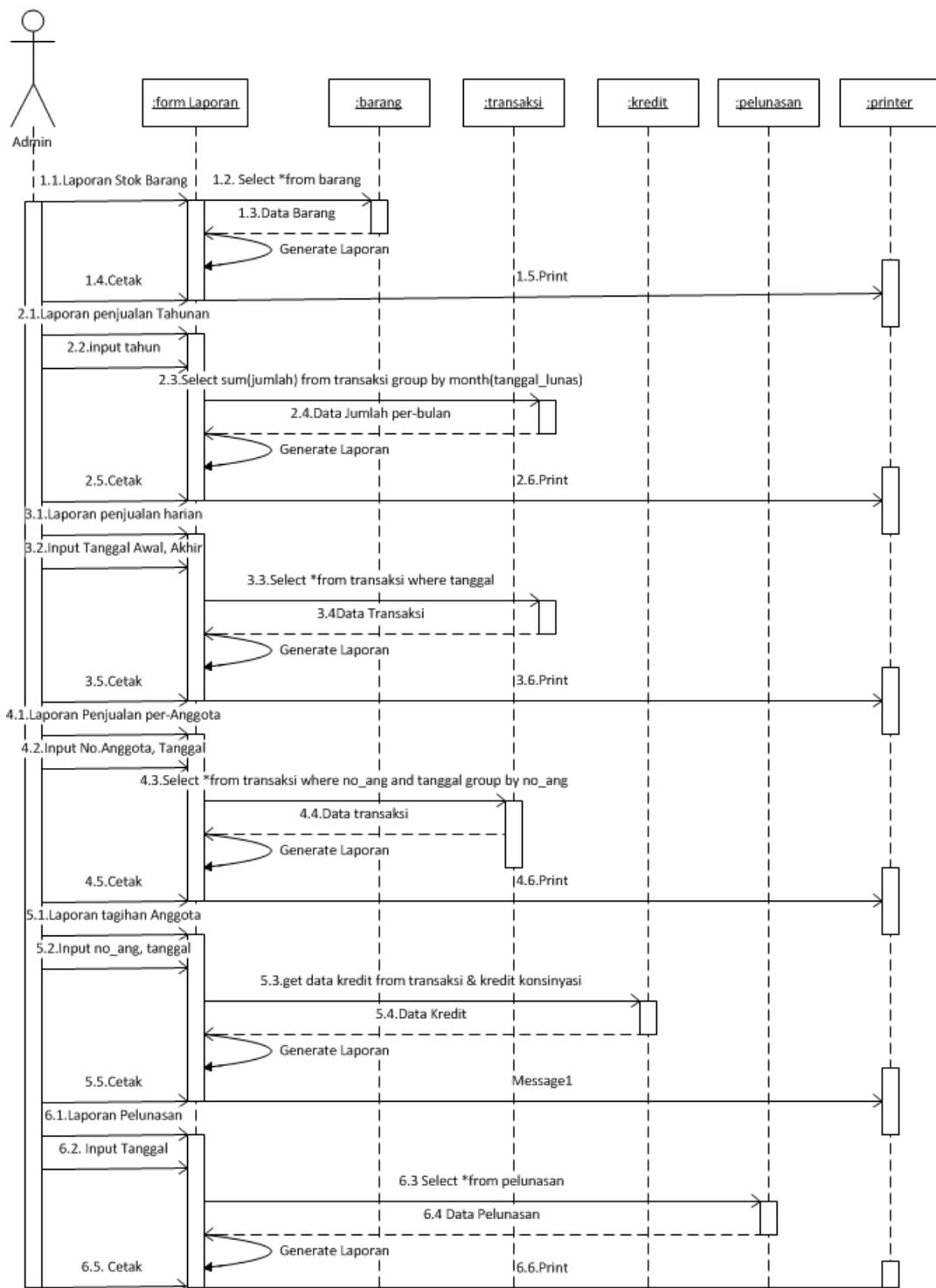
6.4 *Database* akan mengirimkan data Pelunasan Kredit dan Sistem akan generate laporan

6.5 *Admin* memutuskan untuk mencetak.

6.6 Sistem akan mengirim perintah ke printer untuk mencetak laporan.

Sequence diagram transaksi pembelian ditunjukkan pada gambar

4.29



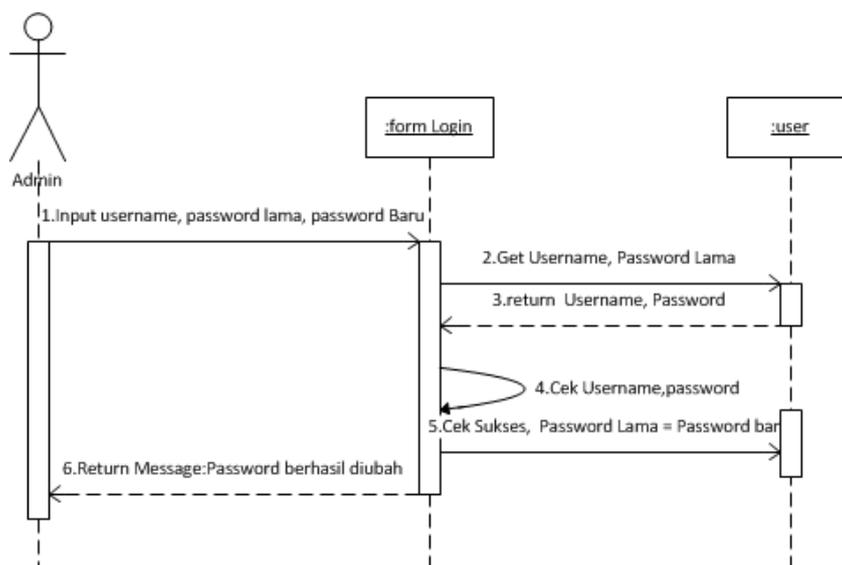
Gambar 4.29 Sequence diagram Laporan

J. Sequence Diagram Ganti Password

Sequence Diagram ganti *password* menjelaskan alur mengganti *password* lama dengan *password* baru. Berikut proses yang terjadi pada *Sequence* diagram ganti *password*:

1. Admin input *username*, *password* lama, *password* baru
2. Sistem mengirim permintaan data *user* dari *database*.
3. *Database* mengirim balik data *user*.
4. Sistem mengecek *username* dan *password* lama dengan yang ada dari *database*.
5. Jika cek sukses, *password* lama akan tergantikan dengan *password* baru.
6. Sistem menampilkan konfirmasi bahwa *password* berhasil dirubah.

Sequence diagram akun ditunjukkan pada gambar 4.31



Gambar 4.30 *Sequence* diagram akun

1. *Class User*

adalah *class* yang menyimpan data *Admin*, yaitu berisikan *username* serta *password Admin*. *Admin* dapat mengelola semua *class*. *Class* ini mempunyai method *insert()* untuk menambah data baru, *update()* untuk mengubah data, dan *delete()* untuk menghapus data.

2. *Class barang*

Class barang adalah *class* yang menyimpan data barang. Terdapat *attribute* *id_barang* sebagai *primary key*. *Class* ini mempunyai method *insert()* untuk menambah data baru, *update()* untuk mengubah data, dan *delete()* untuk menghapus data.

3. *Class Anggota*

Class Anggota adalah *class* yang menyimpan data Anggota. Terdapat beberapa *attribute* *no_ang* sebagai *primary key*. *Class* ini mempunyai method *insert()* untuk menambah data baru, *update()* untuk mengubah data, dan *delete()* untuk menghapus data.

4. *Class Penjualan*

Class penjualan merupakan *class* yang menyimpan data transaksi penjualan, pada *class* ini terdapat beberapa *attribute* seperti: *no_transaksi* sebagai *primary key*, *id_barang*, *id_kredit*, *no_ang* dan *id* sebagai *foreign key*. *Class* ini mempunyai method *insert()* untuk menambah data baru, *getAll()* untuk melihat semua data, dan *getByOne()* untuk mengambil satu data.

5. *Class* Pembelian

Class Pembelian merupakan *class* yang menyimpan data transaksi pembelian barang koperasi. *Class* pembelian terdapat beberapa *attribute* antara lain: no sebagai *primary key* dan id sebagai *foreign key*. *Class* ini mempunyai method *insert()* untuk menambahkan data baru dan *getAll()* untuk mengambil semua data.

6. *Class* Pelunasan

Class Pelunasan adalah *class* yang menyimpan data dari transaksi pelunasan kredit. *Class* ini mempunyai beberapa *attribute*, antara lain: no_pelunasan sebagai *primary key*, id dan no_ang sebagai *foreign key*. *Class* ini mempunyai beberapa method, antara lain *insert()* untuk menambah data.

7. *Class* Kredit

Class Kredit berfungsi untuk menyimpan data Kredit. *Class* ini mempunyai beberapa *attribute*, antara lain: id_kredit sebagai *primary key* dan no_anggota sebagai *foreign key*.

8. *Class* Laporan

Class Laporan berfungsi sebagai pembuat laporan dari fungsi-fungsi yang ada. *Class* ini mempunyai beberapa *attribute* antara lain: no_ang sebagai input nomer anggota, tanggalAwal dan tanggalAkhir sebagai penentu periode. *Class* ini mempunyai beberapa method, antara lain *getAll()* untuk mengambil seluruh data, *print()* untuk

mengirim perintah cetak ke printer, JasperCompiler() dan JasperView() sebagai pembuat report, LaporanStokBarang(), LaporanPenjualanThn(), LaporanPenjualanHarian(), LaporanPenjualanAng(), LaporanTagAng(), dan LaporanPelunasan() untuk method pembuat report berdasar kategori.

4.1.5 Desain Tabel

a. Tabel *user*

Tabel *user* merupakan tabel yang berisikan data-data *Admin* yaitu meliputi *username* dan *password* yang berfungsi untuk *login*. Penjelasan tabel *user* akan dijelaskan oleh tabel 4.10.

Nama Kolom	Tipe Data	Keterangan
id	varchar(4)	<i>Primary Key</i>
nip	varchar(14)	
nama	varchar(25)	
alamat	Text	
telepon	varchar(12)	
<i>username</i>	varchar(8)	
<i>password</i>	varchar(8)	
jabatan	varchar(12)	

Tabel 4.10. Tabel *user*

b. Tabel barang

Tabel barang merupakan tabel yang menyimpan data-data barang, dimana *id_barang* merupakan *primary key*. *nama_barang* merupakan *foreign key*. Penjelasan tabel barang akan dijelaskan oleh tabel 4.11.

Nama Kolom	Tipe Data	Keterangan
id_barang	int(6)	<i>Primary Key</i>
nama_barang	varchar(25)	<i>Foreign Key</i>
harga_beli	int(11)	
harga_jual	int(11)	
tipe_barang	varchar(12)	
stock	int(4)	

Tabel 4.11. Tabel barang

c. Tabel Anggota

Tabel anggota merupakan tabel yang menyimpan data anggota dimana no_ang merupakan *primary key*. Penjelasan tabel anggota akan dijelaskan oleh tabel 4.12.

Nama Kolom	Tipe Data	Keterangan
id_jenis	int(4)	<i>Primary Key</i>
id_anggota	varchar(12)	
nama_anggota	varchar(26)	

Tabel 4.12. Tabel Anggota

d. Tabel kredit

Tabel kredit merupakan tabel yang menyimpan data kredit. id_kredit merupakan *primary key*. No_anggota sebagai *foreign key*. Penjelasan tabel kredit dijelaskan oleh tabel 4.13.

Nama Kolom	Tipe Data	Keterangan
id_kredit	int(6)	<i>Primary Key</i>
nama_barang	varchar(26)	
harga_barang	int(11)	
bunga	int(11)	
jumlah_ang	int(11)	

harga_kredit	int(11)	
harga_ang	int(11)	
no_anggota	int(6)	<i>Foreign Key</i>

Tabel 4.13. Tabel barang

e. Tabel pelunasan

Tabel pelunasan merupakan tabel yang menyimpan data transaksi pelunasan kredit, baik sembako maupun konsinyasi. no_pelunasan merupakan *primary key*. id_kredit, id dan no_ang sebagai *foreign key*. Penjelasan tabel pelunasan akan dijelaskan oleh tabel 4.14.

Nama Kolom	Tipe Data	Keterangan
no_pelunasan	int(6)	<i>Primary Key</i>
tanggal	Date	
id_kredit	int(6)	<i>Foreign Key</i>
angsuran_ke	int(11)	
jumlah	int(11)	
id	int(4)	<i>Foreign Key</i>
no_anggota	int(6)	<i>Foreign Key</i>

Tabel 4.14. Tabel pelunasan

f. Tabel pembelian

Tabel pembelian merupakan tabel yang menyimpan data transaksi pembelian barang sembako. no merupakan *primary key*. id_kredit, id dan no_ang sebagai *foreign key*. Penjelasan tabel pembelian akan dijelaskan oleh tabel 4.15.

Nama Kolom	Tipe Data	Keterangan
No	int(11)	<i>Primary Key</i>
id_trans	int(8)	
nama_barang	varchar(25)	<i>Foreign Key</i>
tipe_barang	varchar(11)	

jumlah_barang	int(3)	
total_harga	int(11)	
tanggal	date	
nama_penjual	varchar(24)	
id	int(4)	<i>Foreign Key</i>

Tabel 4.15. Tabel pembelian

g. Tabel transaksi

Tabel transaksi merupakan tabel yang menyimpan data transaksi penjualan barang sembako. no_transaksi merupakan *primary key*. id_barang, id_kredit, id dan no_ang sebagai *foreign key*. Penjelasan tabel transaksi akan dijelaskan oleh tabel 4.16.

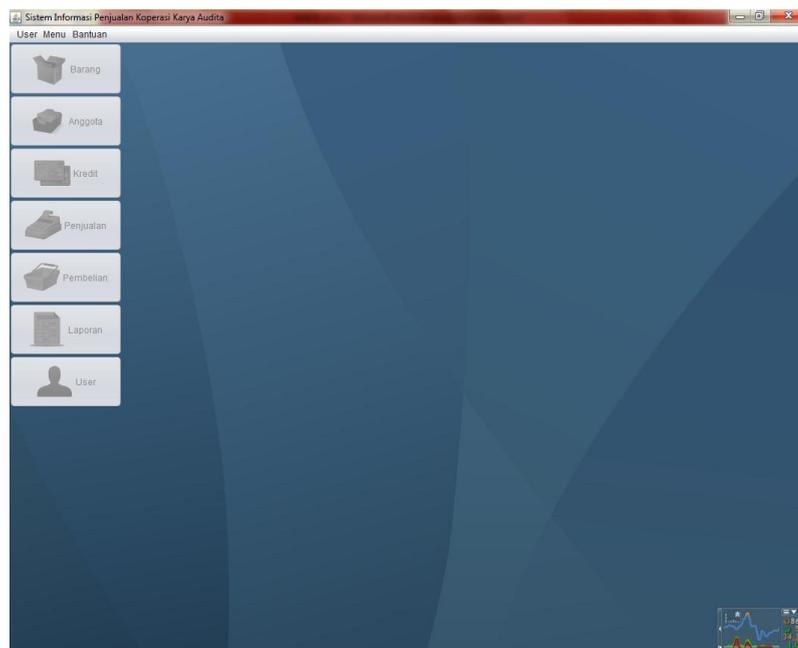
Nama Kolom	Tipe Data	Keterangan
no_transaksi	int(4)	<i>Primary Key</i>
id_transaksi	varchar(6)	
id_barang	int(6)	<i>Foreign Key</i>
jumlah_barang	int(4)	
total_harga	int(11)	
tipe_pembelian	varchar(8)	
tanggal_lunas	date	
id_kredit	int(6)	<i>Foreign Key</i>
no_ang	int(6)	<i>Foreign Key</i>
id	int(4)	<i>Foreign Key</i>

Tabel 4.16. Tabel transaksi

4.2 Implementasi Sistem

4.2.1. *Form Home*

Form Home adalah *form* yang ditampilkan pertama kali ketika aplikasi dijalankan, setelah *login* melalui *form* ini, *Admin* dapat berpindah ke *form* lainnya. *Form Home* akan dijelaskan oleh gambar 4.32.



Gambar 4.32 *Form Home*

4.2.2. *Form Login*

Form Login adalah *form* yang muncul bersamaan dengan *form home* saat aplikasi dijalankan pertama kali. *Admin* memasukkan *Username* beserta *password* yang valid. *Form Login* akan dijelaskan oleh gambar 4.33.

Gambar 4.33 *Form Login*

4.2.3. *Form Barang*

Form Barang adalah *Form* yang digunakan untuk *insert*, *update*,serta *delete* data barang. *Form Login* akan dijelaskan oleh gambar 4.34.

ID Barang	Nama Barang	Harga Beli	Harga Jual	TipeBarang	Stock
1	rins	500	600	Koperasi	6
3	daia	200	300	Konsinyasi	1
4	nuvo	2000	2250	Konsinyasi	1
5	molto 500m	500	750	Koperasi	8
6	pantene	500	750	Koperasi	2
7	pepsodent	2000	2250	Koperasi	2

Gambar 4.34 *Form barang*

4.2.4. *Form Anggota*

Form Anggota adalah *Form* yang digunakan untuk *insert*, *update*,serta *delete* data Anggota. *Form Login* akan dijelaskan oleh gambar 4.35.

Data Anggota

No. :

Id Anggota :

Nama Anggota :

Tanggungans Anggota

Status Tanggungan :

Jumlah Tanggungan :

Jumlah Angsuran :

Catatan

Untuk mengedit dan menghapus, pilihlah salah satu data yang terletak di tabel bawah ini.

Kontrol

Cari

Kategori : Keyword :

Tabel Anggota

Id Anggota	Nama Anggota	Status Tanggungan	Jumlah Tanggungan	Jumlah Angsuran
4532111	Tri Harjono	Tidak	0	0
1235412	Sri Mulyani	Ada	110	110
12312412	Emprah Handayani	Tidak	0	0
134234234	Nicolas Sugiyono	Ada	3600	0
123456	Jonas Hartanto	Ada	22800	0

Gambar 4.35 *Form Anggota*

4.2.5. *Form Pilihan Kredit*

Form Pilihan Kredit adalah *Form* yang digunakan untuk memilih pembuatan kredit baru atau pelunasan kredit lama. Untuk lebih jelasnya akan dijelaskan oleh gambar 4.36.

Gambar 4.36 *Form Pilihan Kredit*

4.2.6. Form Kredit

Form Kredit adalah *Form* yang digunakan untuk *insert*, *update*, serta *delete* data Kredit. *Form Login* akan dijelaskan oleh gambar 4.37.

ID	Nama Bara...	Harga Bara...	Bunga	Jumlah An...	Harga Kredit	Harga Ang
1	sembako	0	0	0	0	0
10	dora	10,000	15	3	11,500	3833
11	buku	200,000	10	6	220,000	36667
12	motor	16,000,000	15	10	17,250,000	1725000
21	motora	15,000,000	10	12	16,500,000	1375000
22	toyota	200,000,000	10	15	220,000,000	14666667
23	donat	200,000	12	4	224,000	56000

Gambar 4.37 *Form Kredit*

4.2.7. Form Pelunasan Kredit

Form Pelunasan Kredit digunakan untuk melakukan Pembayaran pelunasan kredit lama. *Form Login* akan dijelaskan oleh gambar 4.38.

The image shows a software window titled "Form Pelunasan Kredit" with the following fields and controls:

- Tanggal:** May 21, 2012 (with a calendar icon)
- No. Transaksi:** 12
- Data Anggota:**
 - No. Anggota:** 5 (with a "Cari" button)
 - Nama Anggota:** Nicolas Sugiyono
- Tipe Pembayaran:**
 - Kredit Sembako (id: 1)
 - Kredit Konsinyasi
- Data Pembayaran:**
 - Angsuran ke:** 1
 - Jumlah Tertagih:** 3600
 - Dari:** 1
 - Bayar** button

Gambar 4.38 *Form Pelunasan Kredit*

4.2.8. Form Penjualan

Form Penjualan digunakan untuk menambah data transaksi penjualan anggota. *Form Penjualan* akan dijelaskan oleh gambar 4.39.

ID	Nam...	Tipe...	Harga	Juml...	Total	No. A...	Tipe...
6	pante...	Kope...	750	2	1500	Nicol...	Tunai
5	molto...	Kope...	750	3	2250	Nicol...	Tunai
1	rinso	Kope...	600	4	2400	Nicol...	Tunai

Gambar 4.39 *Form Penjualan*

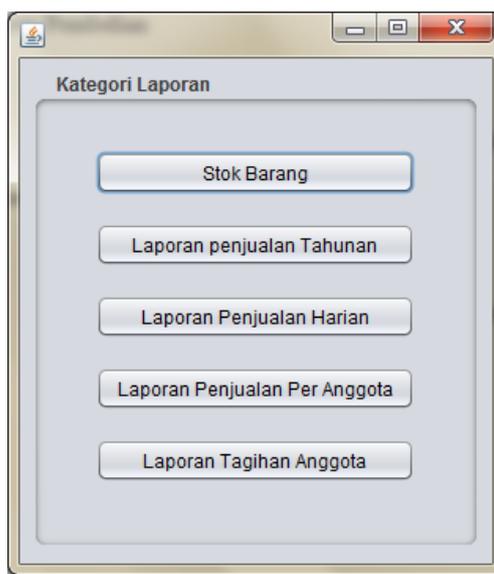
4.2.9. *Form Pembelian*

Form Pembelian merupakan *form* yang digunakan *Admin* untuk menambahkan data pembelian barang untuk koperasi. *Form Pembelian* akan dijelaskan oleh gambar 4.40.

Gambar 4.40 *Form Pembelian*

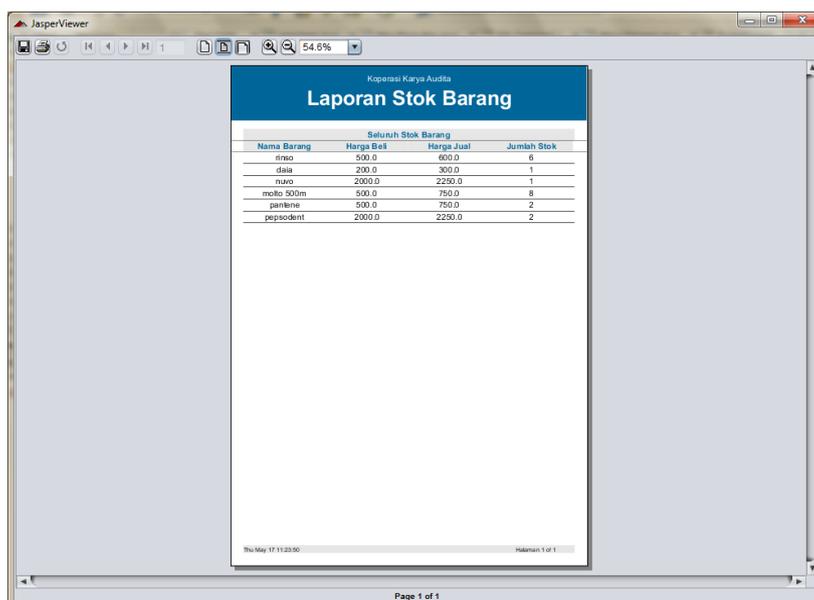
4.2.10. Form Laporan

Form Laporan merupakan *form* yang digunakan *Admin* untuk menampilkan laporan dari data yang ada. *Form* Penjualan akan dijelaskan oleh gambar 4.41.



Gambar 4.41 *Form* Laporan

Contoh Laporan yang ditampilkan akan digambarkan oleh gambar 4.42



The image shows a JasperViewer window displaying a report titled 'Laporan Stok Barang' for 'Koperasi Karya Audita'. The report contains a table with the following data:

Seluruh Stok Barang			
Nama Barang	Harga Beli	Harga Jual	Jumlah Stok
iriso	500.0	600.0	6
daia	200.0	300.0	1
rufo	2000.0	2250.0	1
mako 500ml	500.0	750.0	8
panlene	500.0	750.0	2
pepsodent	2000.0	2250.0	2

The report footer indicates the date and time: 'Thu May 17 11:23:00' and 'Halaman 1 of 1'.

Gambar 4.42 Contoh Laporan yang Ditampilkan

4.2.11. Form User

Form User adalah *Form* yang digunakan untuk *insert*, *update*, serta *delete* data *User*. *Form User* akan dijelaskan oleh gambar 4.43.

The screenshot shows a window titled "Menu Admin" with a light gray background. It is divided into several sections:

- Data Admin:** A vertical list of input fields:
 - Id:** 9
 - NIP:** 1002310412
 - Nama:** Mono
 - Alamat:** Warungboto
 - Telepon:** 0865526586
 - Username:** Mono
 - Password:** *
 - Jabatan:** Admin (dropdown menu)
- Kontrol:** A vertical stack of buttons:
 - Tambah (with a plus icon)
 - Ubah (with a pencil icon)
 - Hapus (with a red X icon)
 - Reset (with a circular arrow icon)
- Catatan:** A text box containing the instruction: "Untuk mengedit dan menghapus, pilihlah terlebih dahulu salah satu dari admin di bawah".
- Tabel Admin:** A table with 6 columns: ID, Nip, Nama, Telepon, Username, and Jabatan. The first row is highlighted in blue.

ID	Nip	Nama	Telepon	Username	Jabatan
9	1002310412	Mono	0865526586	Mono	Admin
10	111221441	Jonas	22001020	admin	Admin
11	223123123	Kezef	2312312	asd	Admin

Gambar 4.43 *Form User*

4.2.12. Form Cari Anggota

Form Cari Anggota adalah *Form* yang digunakan untuk mencari anggota. *Form Cari Anggota* ini digunakan di beberapa *form*, seperti *form* transaksi, *form* pembelian dan *form* laporan. *Form Cari Anggota* akan dijelaskan oleh gambar 4.44.

ID	Nama
4532111	Tri Harjono
1235412	Sri Mulyani
12312412	Emprah Handayani
134234234	Nicolas Sugiyono
123456	Jonas Hartanto

Gambar 4.44 Form Cari Anggota

4.2.13. Form Cari Barang

Form Cari Barang adalah Form yang digunakan untuk mencari anggota. Form Cari Barang ini digunakan di beberapa form, seperti form transaksi dan form pembelian. Form Cari Barang akan dijelaskan oleh gambar 4.45.

ID	Nama	Harga Beli	Harga Jual	Tipe Barang	Stock
1	rinsu	500	600	Koperasi	6
3	daia	200	300	Konsinyasi	1
4	nuvo	2000	2250	Konsinyasi	1
5	molto 500m	500	750	Koperasi	8
6	pantene	500	750	Koperasi	2
7	pepsodent	2000	2250	Koperasi	2

Gambar 4.45 Form Cari Barang

4.2.14. *Form Ganti password*

Form Ganti password adalah *Form* yang digunakan untuk mengganti password *user*. *Form Ganti password* ini digunakan oleh semua tipe *user*. *Form Ganti password* akan dijelaskan oleh gambar 4.46.



The image shows a screenshot of a web application window titled "Ganti Password". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. Inside the window, there is a form with three input fields and one button. The first input field is labeled "User Id :". The second input field is labeled "Masukkan Password Lama :". The third input field is labeled "Masukkan Password Baru :". Below the input fields is a button labeled "Ganti Password".

Gambar 4.46 *Form Ganti Password*

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

- a. Penyusun telah berhasil membangun aplikasi penjualan di Koperasi Karya Audita BPKP Yogyakarta.
- b. Sistem yang dibuat dalam Kerja Praktek ini dapat menyediakan data barang dan transaksi penjualan serta pembelian koperasi.
- c. Sistem yang dibuat dalam Kerja Praktek ini dapat membuat laporan untuk Koperasi Karya Audita secara terotomatisasi.

5.2 Saran

- a. Aplikasi Sistem Informasi Penjualan ini akan lebih baik jika dapat membuat semua jenis laporan yang lebih spesifik untuk Koperasi Karya Audita.
- b. Sistem Aplikasi ini akan lebih baik jika dapat terintegrasi dengan sistem simpan pinjam yang telah dimiliki oleh Koperasi Karya Audita.

DAFTAR PUSTAKA

- Kadir, Abdul. *Pengenalan Sistem Informasi*. Yogyakarta: ANDI, 2003.
- Pressman, Roger S, Ph.D. 2002. *Rekayasa Perangkat Lunak Pendekatan Praktisi (BUKU SATU)*. Penerbit ANDI : Yogyakarta.
- Putranta, H.D. 2004. *Pengantar Sistem dan Teknologi Informasi*. Yogyakarta: AMUS
- Suandy, Erly, Jessica. 2008. *Praktikum Akuntansi Manual dan Komputerisasi dengan MYOB*. Jakarta: Penerbit Salemba Empat
- Supriyanto, A. 2005. *Pengantar Teknologi Informasi*. Jakarta: Salemba Infotek
- Wahyudi, Bambang. 2008. *Konsep Sistem Informasi dari Bit Sampai Database*. Yogyakarta: Penerbit Andi
- Davis, Gordon B. *Kerangka Dasar Sistem Informasi Manajemen Bagian I : Pengantar*. Diterj. oleh Andreas S. Adiwardana. Cetakan kesebelas, PT Ikrar Mandiriabadi, 1999.
- Nugroho, Bunafit. 2004. *Database Relational dengan MySQL*. Yogyakarta : ANDI
- Miftakhul Huda, Bunafit Komputer. (2010). *Membuat Aplikasi Database dengan Java, MySQL dan Netbeans*. Jakarta: PT Elex Media Komputindo.