BAB 3

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Pada tahap analisis sistem dilakukan pengumpulan data dan menganalisa segala dokumen-dokumen yang digunakan pada sistem yang sedang berjalan. Ini dilakukan untuk memudahkan mengevaluasi kekurangan-kekurangan apa saja yang ada pada sistem tersebut yang selanjutnya merancang perbaikan-perbaikan pada sistem tersebut dan menyusun sistem baru. Selain itu juga dibahas analisis kebutuhan non-fungsional yang terdiri dari analisis perangkat keras, analisis perangkat lunak dan analisis *user.*

3.1.1 Analisis Sistem Yang Sedang Berjalan

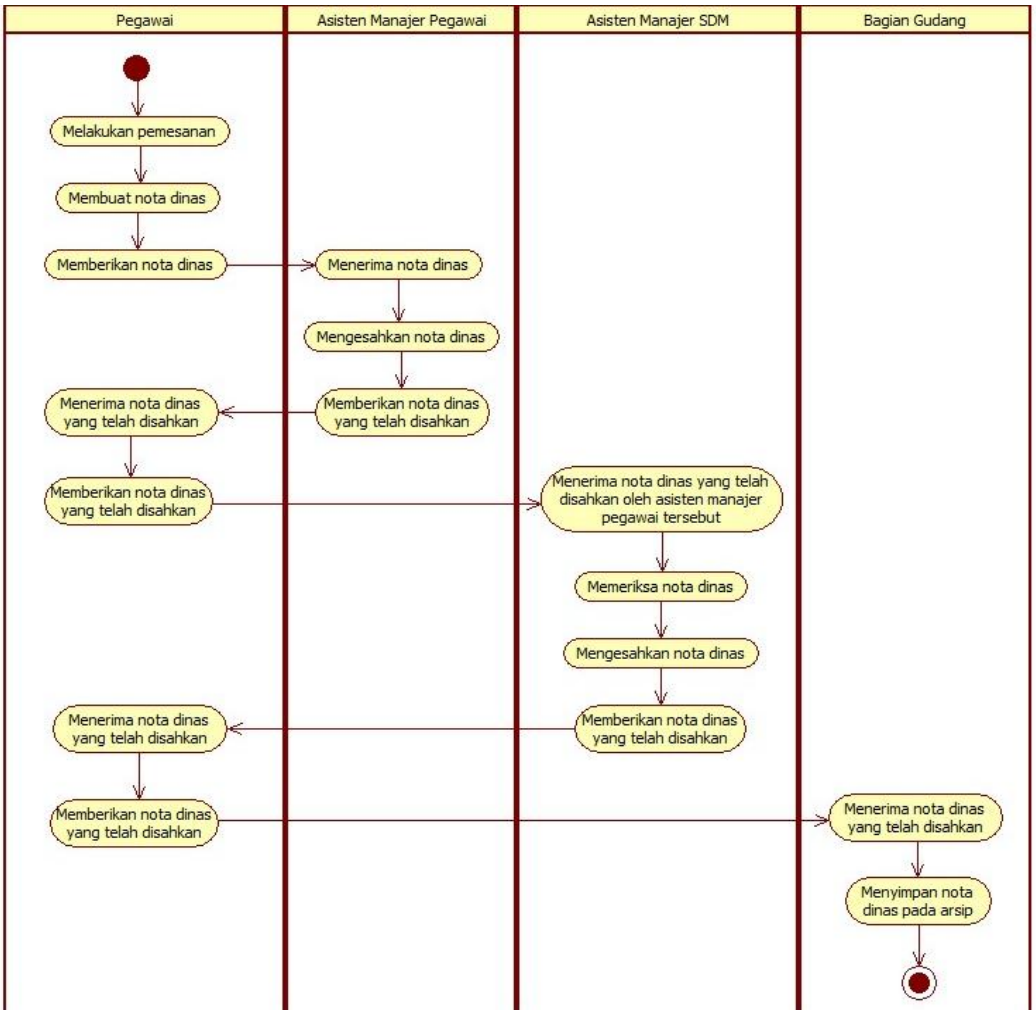
Analisis sistem yang sedang berjalan berisi tentang pemaparan sistem pengolahan data *inventory* yang ada saat ini. Analisis ini dimaksudkan agar perangkat lunak yang dibangun tidak keluar dari cakupan sistem pengolahan data *inventory control* yang ada.

Berdasarkan penelitian yang telah dilakukan, terdapat beberapa prosedur yang sedang berjalan, yaitu prosedur pemesanan, prosedur pengadaan barang, dan prosedur pengiriman barang.

3.1.1.1 Prosedur Pemesanan Barang

1. Pegawai yang akan melakukan pemesanan barang membuat nota dinas yang berisikan daftar pemesanan barang.

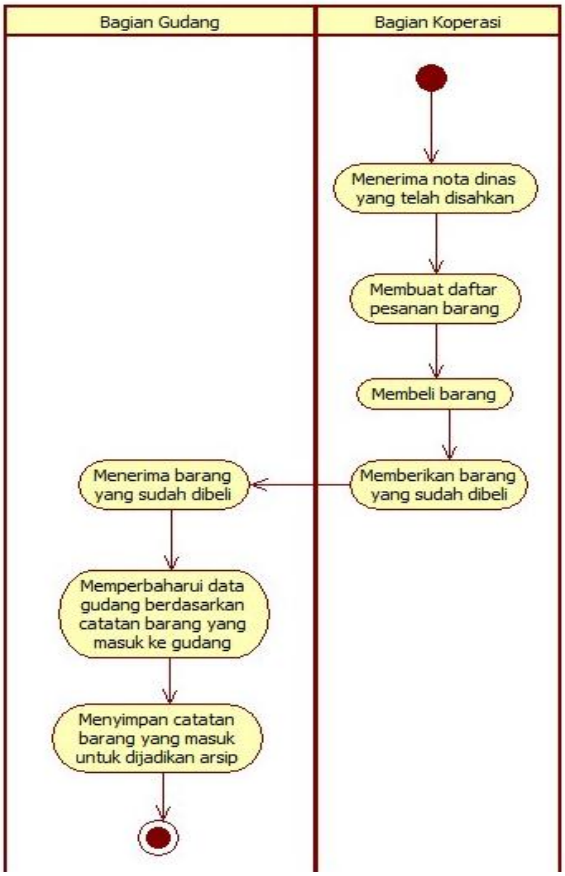
2. Pegawai menyerahkan nota dinas ke asisten manajer (asman) pegawai tersebut untuk meminta pengesahan nota dinas.
3. Nota dinas yang telah disetujui diserahkan ke asisten manajer (asman) bagian SDM. Asman SDM melakukan pengecekan permintaan yang akan disetujui.
4. Setelah disetujui oleh Asman SDM, nota dinas diserahkan ke bagian gudang untuk pengadaan barang.
5. Nota dinas disimpan untuk dijadikan arsip.



Gambar 3.1 Prosedur Pemesanan Barang

3.1.1.2 Prosedur Pengadaan Barang

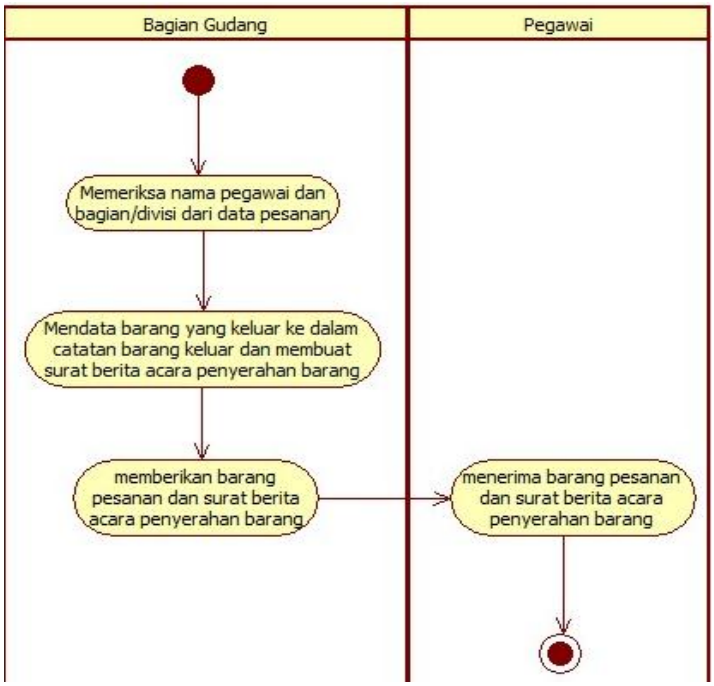
1. Bagian gudang akan memeriksa pada data gudang apakah barang yang diperlukan tersedia atau tidak. Jika barang ada maka bagian gudang akan menyerahkan langsung barang tersebut ke pegawai/pemohon. Bila barang tidak ada di gudang, bagian gudang akan memberikan daftar pesanan barang ke bagian koperasi untuk melakukan pembelian barang.
2. Bagian koperasi memberikan barang yang sudah dibeli ke bagian gudang untuk dicatat.
3. Bagian gudang memperbaharui data gudang berdasarkan catatan barang yang masuk ke gudang.
4. Bagian gudang menyimpan catatan barang yang masuk untuk dijadikan arsip.



Gambar 3.2 Prosedur Pengadaan Barang

3.1.1.3 Prosedur Pengiriman Barang

1. Bagian gudang memeriksa nama pegawai dan bagian/divisi dari data pesanan, untuk menyerahkan barang pesanan.
2. Barang yang akan diserahkan ke pemesan di data terlebih dahulu ke dalam catatan barang keluar dan dibuat surat berita acara penyerahan barang.
3. Bagian gudang memberikan barang pesanan dan surat berita acara penyerahan barang ke pegawai yang melakukan pemesanan barang.



Gambar 3.3 Prosedur Pengiriman Barang

3.1.2 Analisis Dokumen

Data dalam bentuk dokumen menghasilkan informasi yang digunakan dalam proses pada sistem yang dibangun. Dokumen yang terlibat dalam aplikasi pengolahan data *inventory control* tertera pada Tabel 3.1.

Tabel 3.1 Dokumen yang terlibat

No	Nama Dokumen	Deskripsi
1	Nota Dinas	Fungsi : menginformasikan permohonan permintaan barang Item Data : sumber, tujuan, tanggal pengajuan, data barang yang dibutuhkan.

3.1.3 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional menggambarkan kebutuhan luar sistem yang diperlukan untuk menjalankan aplikasi yang dibangun. Adapun kebutuhan non-fungsional untuk menjalankan aplikasi pengolahan data *inventory control*

meliputi kebutuhan perangkat keras, kebutuhan perangkat lunak dan pengguna sistem yang akan memakai aplikasi. Analisis kebutuhan non-fungsional bertujuan agar aplikasi yang dibangun dapat digunakan sesuai dengan kebutuhan bagian gudang dalam pengolahan data *inventory control*.

3.1.3.1 Analisis Perangkat Keras

Untuk menjalankan suatu aplikasi maka diperlukan perangkat keras yang dapat mendukung proses kerja dari sistem itu sendiri. Spesifikasi perangkat keras yang digunakan oleh bagian gudang dalam melakukan pengolahan data *inventory control* tertera dalam Tabel 3.2.

Tabel 3.2 Spesifikasi Perangkat Keras Bagian Gudang

No	Perangkat Keras	Spesifikasi
1	Processor	Intel Pentium 4 3.00 GHz
2	Monitor	Montor LCD 17 inch
3	VGA	VGA On-Board 224 MB
4	Memori	1024 MB
5	Optical Drive	DVD - ROM
6	Keyboard	Logitech
7	Mouse	Logitech
8	Printer	Printer hitam-putih

Untuk menjalankan aplikasi pengolahan data *inventory control* dibutuhkan perangkat keras pendukung sehingga aplikasi yang dibangun dapat berjalan dengan baik. Spesifikasi minimun perangkat keras untuk menjalankan aplikasi tersebut tertera pada Tabel 3.3.

Tabel 3.3 Spesifikasi Perangkat Keras Minimal

No	Perangkat Keras	Spesifikasi
1	Processor	Intel Pentium 4 1.6 GHz
2	Monitor	Montor 15 inch
3	Harddisk	320 GB
4	Memori	1 GB
6	Keyboard dan Mouser	Standar
7	Printer	Printer warna

Perangkat keras yang ada pada saat ini belum memenuhi kebutuhan untuk dapat menjalankan aplikasi yang akan dibangun.

3.1.3.2 Analisis Perangkat Lunak

Perangkat lunak yang digunakan oleh bagian gudang untuk membantu pengolahan data *inventory control* adalah :

1. Sistem Operasi Windows XP SP2
2. Microsoft Office Excel 2007
3. Microsoft Office Word 2007

Sedangkan untuk mengoptimalkan program yang dibangun, dibutuhkan beberapa perangkat lunak pendukung sebagai berikut :

1. Sistem Operasi Windows XP SP2.
2. Web Browser seperti Mozilla Firefox atau Google Chrome.
3. Macromedia Dreamweaver 8
4. MySQL 5.1
5. XAMPP atau WAMPServer

Berdasarkan perbandingan perangkat lunak yang ada di bagian gudang dengan perangkat lunak yang dibutuhkan untuk menjalankan aplikasi maka dapat disimpulkan bahwa dibutuhkan perangkat lunak tambahan untuk menjalankan aplikasi pengolahan data *inventory control* sehingga aplikasi yang dibangun dapat berjalan dengan optimal.

3.1.3.3 Pengguna Sistem

Suatu aplikasi akan berjalan optimal apabila ditunjang oleh perangkat pikir yang memiliki kemampuan dalam menjalankan aplikas yang bersangkutan. Pengguna sistem, yaitu pegawai yang membawahi bidang pengolahan data *inventory control* dijelaskan sebagai berikut :

Umur	:	30 tahun
Pendidikan Terakhir	:	Strata Satu (S1)
Kemampuan yang dimiliki	:	Mampu menggunakan perangkat lunak *Office* dan aplikasi lainnya yang terdapat di bagian gudang.

Untuk menjalankan aplikasi pengolahan data pengobatan dibutuhkan pengguna sistem sehingga aplikasi yang dibangun dapat berjalan dengan baik. Berikut ini penjelasan pengguna sistem untuk menjalankan aplikasi yang dibangun :

Umur	:	Minimal 23
Pendidikan Terakhir	:	Diploma Tiga (D3)

Kemampuan yang dimiliki : Mampu menggunakan aplikasi yang dibangun dalam pengolahan data *inventory control*.

Melihat perbandingan pengguna sistem yang ada dengan pengguna sistem yang dibutuhkan untuk menjalankan aplikasi yang dibangun maka dapat disimpulkan bahwa diperlukan adanya pelatihan dalam penggunaan sistem terhadap *user*. Pelatihan perangkat pikir yang ada bertujuan agar pengguna sistem dapat sebaik mungkin menggunakan aplikasi yang dibangun.

3.1.3.4 Analisis Metode Garis-Lurus (*Straight-Line Method*)

Pada Subbab ini akan dibahas mengenai analisis penggunaan metode yang digunakan untuk perhitungan biaya susut. Metode yang digunakan yaitu metode garis-lurus (*straight-line method*). Kelebihan dari metode ini adalah telah digunakan secara luas dalam praktek karena kemudahannya dan prosedur garis-lurus secara konseptual seringkali juga merupakan prosedur penyusutan yang paling sesuai. Keuntungan dari perhitungan biaya susut untuk sebuah perusahaan adalah supaya perusahaan bisa menjurnal pengurangan nilai aktiva (aset tetap), Penjurnalan merupakan proses penyusunan laporan keuangan perusahaan dan pada saat akan menjual aset akan diketahui apakah penjualan aset tersebut menguntungkan atau tidak.

Klasifikasi barang yang mempunyai biaya susut yaitu barang yang tidak habis pakai atau barang yang ada di kategori inventaris. Berikut ini adalah klasifikasi barang dari biaya susut :

Tabel 3.4 Klasifikasi barang yang mempunyai biaya susut

Ada Biaya Susut (kategori inventaris)	Tidak Ada Biaya Susut (kategori ATK)
1. Meja	1. Kertas A4
2. Lemari	2. Pulpen
3. Kursi	3. Buku
4. Filing Cabinet	4. Spidol
5. Papan Tulis	5. Lem
6. Peralatan Komputer	6. Tinta Printer
7. Dll	7. Dll

Contoh kasus :

Lemari dibeli tanggal 1 Januari 2011 seharga Rp. 2.000.000,- dengan nilai sisa Rp. 200.000, - (10% dari harga beli) dan umur ekonomis 5 tahun.

$$\frac{(2.000.000 - 200.000)}{5\,\text{tahun}} = 360.000$$

Beban Penyusutan sebesar Rp. 360.000,-

Tabel 3.5 Metode Garis Lurus

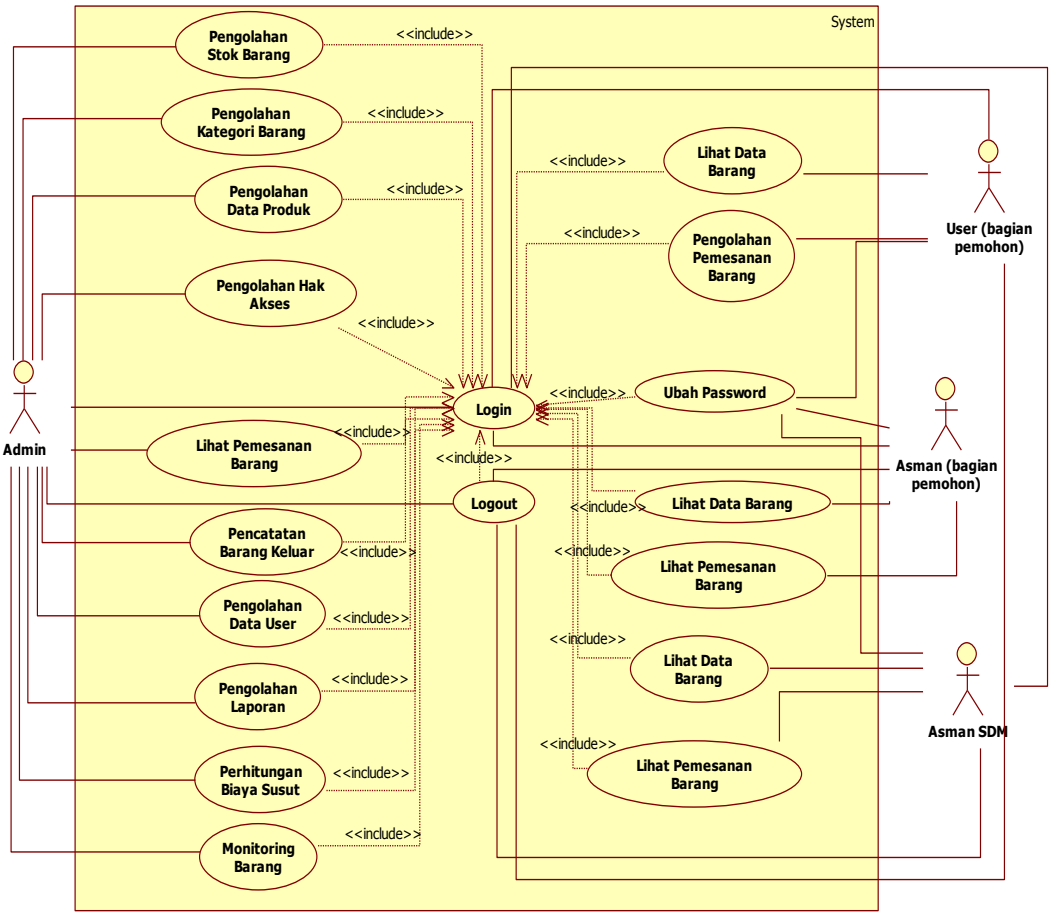
Tahun	Nilai Lemari Awal Tahun	Beban Penyusutan	Saldo Akhir Penyusutan	Nilai Lemari Akhir Tahun
1	Rp. 2.000.000	Rp. 72.000	Rp. 72.000	Rp. 1.928.000
2	Rp. 1.928.000	Rp. 72.000	Rp. 144.000	Rp. 1.856.000
3	Rp. 1.856.000	Rp. 72.000	Rp. 216.000	Rp. 1.784.000
4	Rp. 1.784.000	Rp. 72.000	Rp. 288.000	Rp. 1.712.000
5	Rp. 1.712.000	Rp. 72.000	Rp. 360.000	Rp. 1.640.000

3.1.4 Analisis Kebutuhan Fungsional

Analisis sistem yang dilakukan menggunakan tools UML. Adapun tahapan analisis sistem menggunakan uml adalah *use case diagaram, use case scenario, sequence diagram, class diagram* dan *deployment diagram.*

3.1.4.1 Use case Diagram

Use case diagram merupakan konstruksi untuk mendeskripsikan hubungan-hubungan yang terjadi antar aktor dengan aktivitas yang terdapat pada sistem. Sasaran pemodelan *use case* diantaranya adalah mendefinisikan kebutuhan fungsional dan operasional sistem dengan mendefinisikan skenario penggunaan yang disepakati antara pemakai dan pengembang. Dari analisis pengguna aplikasi yang ada maka *use case diagram* untuk aplikasi pengolahan data *inventory control* di PT. PLN Distribusi Jawa Barat dan Banten APJ Bandung dapat dilihat dalam Gambar 3.4.



Gambar 3.4 ***Use case Diagram***

3.1.4.2 *Use case* Skenario

Use case Skenario mendeskripsikan urutan langkah-langkah dalam proses bisnis, baik yang dilakukan aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor. Berikut ini penjelasan dari masing-masing skenario tersebut:

3.1.4.2.1 *Use case* Skenario Login

Interaksi antara aktor pengguna dengan *use case* login dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3.6 *Use case* Skenario Login

Identifikasi Masalah	
Nama	Login
Tujuan	Masuk ke dalam sistem sebagai administrator
Deskripsi	Proses login ini sebagai autentifikasi kewenangan sebagai administrator dalam sistem
Aktor	Admin, User, Asman pemohon, Asman sdm
Usecase Yang Berkaitan	-
Skenario Utama	
Kondisi Awal	*Form* Login di tampilkan
Aksi Aktor	**Reaksi Sistem**
1. Memasukkan *username* dan *password*	2. Mengautentifikasi data login dengan data admin pada basis data
	3. Bila valid akan menampilkan halaman utama untuk admin
Skenario Alternatif (Jika Gagal)	
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan pesan bahwa data login salah
2. Memasukkan lagi *username* dan *password*	
Kondisi Akhir	Admin dapat melakukan kegiatan pada sistem sesuai kewenangan sebagai adminitrator

3.1.4.2.2 *Use case* Skenario Logout

Interaksi antara aktor pengguna dengan *use case* logout dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3.7 *Use case* Skenario Logout

Identifikasi Masalah	
Nama	Logout
Tujuan	Keluar dari sistem
Deskripsi	Proses logout ini digunakan untuk keluar dari sistem
Aktor	Admin
Usecase Yang Berkaitan	Login
Skenario Utama	
Kondisi Awal	Admin berada pada sistem dengan kondisi login
Aksi Aktor	**Reaksi Sistem**
1. Admin menekan tombol logout	2. Sistem merespon dengan merubah status log user dan mengarahkan admin ke halaman login
Kondisi Akhir	Admin berhasil melakukan proses logout pada sistem

3.1.4.2.3 *Use case* Skenario Pengolahan Data *User*

Interaksi antara aktor pengguna, yaitu admin dengan *use case* pengolahan data *user* dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3.8 *Use case* Skenario Pengolahan Data *User*

Identifikasi			
Nama	Pengolahan Data User		
Tujuan	Mengelola data User pada sistem		
Deskripsi	Proses pengolahan data user merupakan proses untuk mengatur dan mengolah data user pada sistem yang mencakup tampil data, tambah data, ubah data, dan hapus data.		
Aktor	Admin		
***Use case* yang Berkaitan**	Login		
Skenario Utama			
Kondisi Awal	Menampilkan data user dan menu pengolahan user		
Aksi Aktor		**Reaksi Sistem**	
1	Menekan tombol tambah data user	2	Menampilkan form tambah data user
3	Mengisi inputan pada form tambah data user dan menekan tombol tambah	4	Memvalidasi data masukan, jika data valid maka data disimpan pada database dan mengarahkan ke menu pengolahan user, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form tambah data user

Skenario Alternatif (Ubah Data Kategori)			
5	Menekan tombol ubah pada data user yang akan diubah	6	Menampilkan form ubah data yang inputannya sudah terisi data dari database , berdasarkan data yang akan diubah.
7	Mengubah data user pada inputan form ubah data dan menekan tombol ubah	8	Memvalidasi data masukan, jika data valid maka data disimpan pada database dan mengarahkan ke menu pengolahan user, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form ubah data user
Skenario Alternatif (Hapus Data Kategori)			
9	Menekan tombol hapus pada data user yang akan dihapus	10	Menampilkan pesan konfirmasi apakah data akan dihapus
11	Menekan tombol ya pada pesan konfirmasi	12	Menghapus data user pada database dan mengarahkan kembali ke menu pengolahan user
13	Menekan tombol tidak pada pesan konfirmasi	14	Membatalkan proses penghapusan dan mengarahkan kembali ke menu pengolahan user
Kondisi Akhir	Data User telah berhasil diolah		

3.1.4.2.4 *Use case* Skenario Pengolahan Stok Barang

Interaksi antara aktor pengguna dengan *use case* pengolahan Stok barang dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3.9 *Use case* Skenario Pengolahan Stok Barang

Identifikasi			
Nama	Pengolahan Stok Barang		
Tujuan	Mengelola stok barang pada sistem		
Deskripsi	Proses pengolahan stok barang merupakan proses untuk mengatur dan mengolah stok barang pada sistem yang mencakup tampil data dan tambah data.		
Aktor	Admin		
***Use case* yang Berkaitan**	Login		
Skenario Utama			
Kondisi Awal	Menampilkan data stok barang dan menu pengolahan stok barang		
Aksi Aktor		**Reaksi Sistem**	
1	Menekan tombol tambah data barang	2	Menampilkan form tambah data barang
3	Memilih data pesanan yang akan ditambahkan dan menekan tombol tambah	4	Memvalidasi data masukan, jika data valid maka data disimpan pada database dan mengarahkan ke menu pengolahan stok barang, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form tambah data barang
Kondisi Akhir	Data barang telah berhasil diolah		

3.1.4.2.5 *Use case* Skenario Pengolahan Kategori Barang

Interaksi antara aktor pengguna dengan *use case* pengolahan kategori barang dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3.10 *Use case* Skenario Pengolahan Kategori Barang

Identifikasi			
Nama	Pengolahan Data Kategori		
Tujuan	Mengelola data kategori pada sistem		
Deskripsi	Proses pengolahan data kategori merupakan proses untuk mengatur dan mengolah data kategori pada sistem yang mencakup tampil data, tambah data, ubah data, dan hapus data.		
Aktor	Admin		
***Use case* yang Berkaitan**	Login		
Skenario Utama			
Kondisi Awal	Menampilkan data kategori dan menu pengolahan kategori		
Aksi Aktor		**Reaksi Sistem**	
1	Menekan tombol tambah data kategori	2	Menampilkan form tambah data kategori
3	Mengisi inputan pada form tambah data kategori dan menekan tombol tambah	4	Memvalidasi data masukan, jika data valid maka data disimpan pada database dan mengarahkan ke menu pengolahan kategori dan menampilkan pesan notifikasi, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form tambah data kategori
Skenario Alternatif (Ubah Data Kategori)			
5	Menekan tombol ubah pada data kategori yang akan diubah	6	Menampilkan form ubah data yang berisi data dari database yang akan diubah.
7	Mengubah data kategori pada inputan form ubah data dan menekan tombol ubah	8	Memvalidasi data masukan, jika data valid maka data disimpan pada database dan mengarahkan ke menu pengolahan kategori dan menampilkan pesan notifikasi, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form ubah data kategori
Skenario Alternatif (Hapus Data Kategori)			
9	Menekan tombol hapus pada data kategori yang akan dihapus	10	Menampilkan pesan konfirmasi apakah data akan dihapus
11	Menekan tombol ya pada pesan konfirmasi	12	Menghapus data kategori pada database dan mengarahkan kembali ke menu pengolahan kategori
13	Menekan tombol tidak pada pesan konfirmasi	14	Membatalkan proses penghapusan dan mengarahkan kembali ke menu pengolahan kategori
Kondisi Akhir	Data kategori telah berhasil diolah		

3.1.4.2.6 *Use case* Skenario Pengolahan Data Master

Interaksi antara aktor pengguna dengan *use case* pengolahan data master dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3.11 *User Case* Skenario Pengolahan Data Master

Identifikasi			
Nama	Pengolahan Data Master		
Tujuan	Mengelola data master pada sistem		
Deskripsi	Proses pengolahan data master merupakan proses untuk mengatur dan mengolah data master pada sistem yang mencakup tampil data, tambah data, ubah data, dan hapus data.		
Aktor	Admin		
***Use case* yang Berkaitan**	Login		
Skenario Utama			
Kondisi Awal	Menampilkan data master dan menu pengolahan data master		
Aksi Aktor		**Reaksi Sistem**	
1	Menekan tombol tambah data master	2	Menampilkan form tambah data master
3	Mengisi inputan pada form tambah data master dan menekan tombol tambah	4	Memvalidasi data masukan, jika data valid maka data disimpan pada database dan mengarahkan ke menu pengolahan data master dan menampilkan pesan notifikasi, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form (tambah produk/ tambah merek/ tambah keterangan)
Skenario Alternatif (Ubah Data Master)			
5	Menekan tombol ubah pada data master yang akan diubah	6	Menampilkan form ubah data yang berisi data dari database yang akan diubah.
7	Mengubah data master pada inputan form ubah data dan menekan tombol ubah	8	Memvalidasi data masukan, jika data valid maka data disimpan pada database dan mengarahkan ke menu pengolahan data master, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form ubah data (produk/merek/keterangan)
Skenario Alternatif (Hapus Data Master)			
9	Menekan tombol hapus pada data (produk/merek/keterangan) yang akan dihapus	10	Menampilkan pesan konfirmasi apakah data akan dihapus
11	Menekan tombol ya pada pesan konfirmasi	12	Menghapus data (produk/merek/keterangan) pada database dan mengarahkan kembali ke menu pengolahan data master
13	Menekan tombol tidak pada pesan konfirmasi	14	Membatalkan proses penghapusan dan mengarahkan kembali ke menu pengolahan data master
Kondisi Akhir	Data master berhasil diolah		

3.1.4.2.7 *Use case* Skenario Pengolahan Laporan

Interaksi antara aktor pengguna dengan *use case* pengolahan laporan dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3.12 *Use case* Skenario Pengolahan Laporan

Identifikasi Masalah	
Nama	Pengolahan Laporan
Tujuan	Untuk mengolah laporan
Deskripsi	Proses ini untuk mengelola laporan dari data yang telah dikelola sebelumnya oleh sistem.
Aktor	Admin
Usecase Yang Berkaitan	Login
Skenario Utama	
Kondisi Awal	Tampilan *Form* Pengolahan laporan
Aksi Aktor	**Reaksi Sistem**
1. admin memilih menu laporan	2. sistem akan menampilkan data laporan yang dipilih
3. admin memilih bulan dan tahun dan tekan tombol ok	4. sistem akan menampilkan laporan yang akan dicetak
5. admin menekan tombol cetak	6. sistem akan menampilkan laporan dalam bentuk pdf
Kondisi Akhir	Admin berhasil melakukan cari dan cetak laporan

3.1.4.2.8 *Use case* Skenario Pengolahan Barang Keluar

Interaksi antara aktor pengguna dengan *use case* pengolahan barang keluar dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3.13 *Case* Skenario Pengolahan Barang Keluar

Identifikasi			
Nama	Pengolahan Data Barang keluar		
Tujuan	Mengelola data barang keluar pada sistem		
Deskripsi	Proses pengolahan data barang keluar merupakan proses untuk mengatur dan mengolah data barang keluar pada sistem yang berdasarkan data pesanan yang mencakup tampil data, tambah data, dan hapus data.		
Aktor	Admin		
***Use case* yang Berkaitan**	Login		
Skenario Utama			
Kondisi Awal	Menampilkan data barang keluar dan menu pengolahan barang keluar		
Aksi Aktor		**Reaksi Sistem**	
1	Menekan tombol tambah data barang	2	Menampilkan form tambah data barang

	keluar		keluar
3	Memilih data pesanan yang akan ditambahkan dan menekan tombol tambah	4	Memvalidasi data masukan, jika data valid maka data disimpan pada database dan mengarahkan ke menu pengolahan barang keluar, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form tambah data barang keluar
Skenario Alternatif (Hapus Data Barang keluar)			
9	Menekan tombol hapus pada data barang keluar yang akan dihapus	10	Menampilkan pesan konfirmasi apakah data akan dihapus
11	Menekan tombol ya pada pesan konfirmasi	12	Menghapus data barang keluar pada database dan mengarahkan kembali ke menu pengolahan barang keluar
13	Menekan tombol tidak pada pesan konfirmasi	14	Membatalkan proses penghapusan dan mengarahkan kembali ke menu pengolahan barang keluar
Kondisi Akhir	Data barang keluar telah berhasil diolah		

3.1.4.2.9 *Use case* Skenario Pengolahan Pemesanan

Interaksi antara aktor pengguna dengan *use case* pengolahan pemesanan dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3.14 *Use case* Skenario Pengolahan Pemesanan

Identifikasi			
Nama	Pengolahan Data Pemesanan		
Tujuan	Mengelola data pemesanan pada sistem		
Deskripsi	Proses pengolahan data pemesanan merupakan proses untuk mengatur dan mengolah data pemesanan pada sistem yang mencakup tampil data, tambah data, ubah data, dan hapus data.		
Aktor	User		
***Use case* yang Berkaitan**	Login		
Skenario Utama			
Kondisi Awal	Menampilkan data pemesanan dan menu pengolahan pemesanan		
Aksi Aktor		**Reaksi Sistem**	
1	Menekan tombol tambah data pemesanan	2	Menampilkan form tambah data pemesanan
3	Mengisi inputan pada form tambah data pemesanan dan menekan tombol tambah	4	Memvalidasi data masukan, jika data valid maka data disimpan pada database dan mengarahkan ke halaman tambah pemesanan, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form tambah data pemesanan
Skenario Alternatif (Ubah Data Pemesanan)			
5	Menekan tombol ubah pada data detail pemesanan yang akan diubah	6	Menampilkan form ubah data yang berisi data dari database yang akan diubah.
7	Mengubah data detail pemesanan pada inputan form ubah data dan menekan tombol ubah	8	Memvalidasi data masukan, jika data valid maka data disimpan pada database dan mengarahkan ke halaman pemesanan, jika

			tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form ubah data pemesanan
Skenario Alternatif (Hapus Data Pemesanan)			
9	Menekan tombol hapus pada data detail pemesanan yang akan dihapus	10	Menampilkan pesan konfirmasi apakah data akan dihapus
11	Menekan tombol ya pada pesan konfirmasi	12	Menghapus data pemesanan pada database dan mengarahkan kembali ke menu pengolahan pemesanan
13	Menekan tombol tidak pada pesan konfirmasi	14	Membatalkan proses penghapusan dan mengarahkan kembali ke menu pengolahan pemesanan
Kondisi Akhir	Data pemesanan telah berhasil diolah		

3.1.4.2.10 *Use case* Skenario Pengolahan Monitoring Barang

Interaksi antara aktor pengguna dengan *use case* pengolahan monitoring barang dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3.15 *Use case* Skenario Pengolahan Monitoring Barang

Identifikasi			
Nama	Pengolahan Data Monitoring barang		
Tujuan	Mengelola data monitoring barang pada sistem		
Deskripsi	Proses pengolahan data monitoring barang merupakan proses untuk mengatur dan mengolah data monitoring barang pada sistem yang mencakup tampil data dan ubah data.		
Aktor	Admin		
***Use case* yang Berkaitan**	Login		
Skenario Utama			
Kondisi Awal	Menampilkan data monitoring barang dan menu pengolahan monitoring barang		
Aksi Aktor		**Reaksi Sistem**	
1	Menekan tombol ubah pada data monitoring barang yang akan diubah	2	Menampilkan form ubah data yang berisi data dari database yang akan diubah.
3	Mengubah data monitoring barang pada inputan form ubah data dan menekan tombol ubah	4	Memvalidasi data masukan, jika data valid maka data disimpan pada database dan mengarahkan ke menu pengolahan monitoring barang, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form ubah data monitoring barang
Kondisi Akhir	Data monitoring barang telah berhasil diolah		

3.1.4.2.11 *Use case* Skenario Pengolahan Hak Akses

Interaksi antara aktor pengguna dengan *use case* pengolahan hak akses dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3.16 *Use case* Skenario Pengolahan Hak Akses

Identifikasi			
Nama	Pengolahan Data Hak akses		
Tujuan	Mengelola data hak akses pada sistem		
Deskripsi	Proses pengolahan data hak akses merupakan proses untuk mengatur dan mengolah data hak akses pada sistem yang mencakup tampil data, tambah data, ubah data, dan hapus data.		
Aktor	Admin		
***Use case* yang Berkaitan**	Login		
Skenario Utama			
Kondisi Awal	Menampilkan data hak akses dan menu pengolahan hak akses		
Aksi Aktor		**Reaksi Sistem**	
1	Memilih tombol tambah data hak akses	2	Menampilkan form tambah data hak akses
3	Mengisi inputan pada form tambah data hak akses dan menekan tombol tambah	4	Memvalidasi data masukan, jika data valid maka data disimpan pada database dan mengarahkan ke menu pengolahan hak akses, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form tambah data hak akses
Skenario Alternatif (Ubah Data Hak akses)			
5	Menekan tombol ubah pada data hak akses yang akan diubah	6	Menampilkan form ubah data yang berisi data dari database yang akan diubah.
7	Mengubah data hak akses pada inputan form ubah data dan menekan tombol ubah	8	Memvalidasi data masukan, jika data valid maka data disimpan pada database dan mengarahkan ke menu pengolahan hak akses, jika tidak valid maka akan menampilkan pesan kesalahan dan menampilkan kembali form ubah data hak akses
Skenario Alternatif (Hapus Data Hak akses)			
9	Menekan tombol hapus pada data hak akses yang akan dihapus	10	Menampilkan pesan konfirmasi apakah data akan dihapus
11	Menekan tombol ya pada pesan konfirmasi	12	Menghapus data hak akses pada database dan mengarahkan kembali ke menu pengolahan hak akses
13	Menekan tombol tidak pada pesan konfirmasi	14	Membatalkan proses penghapusan dan mengarahkan kembali ke menu pengolahan hak akses
Kondisi Akhir	Data hak akses telah berhasil diolah		

3.1.4.2.12 *Use case* Skenario Ubah Password

Interaksi antara aktor pengguna dengan *use case* ubah password dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3.17 *Use case* Skenario Ubah Password

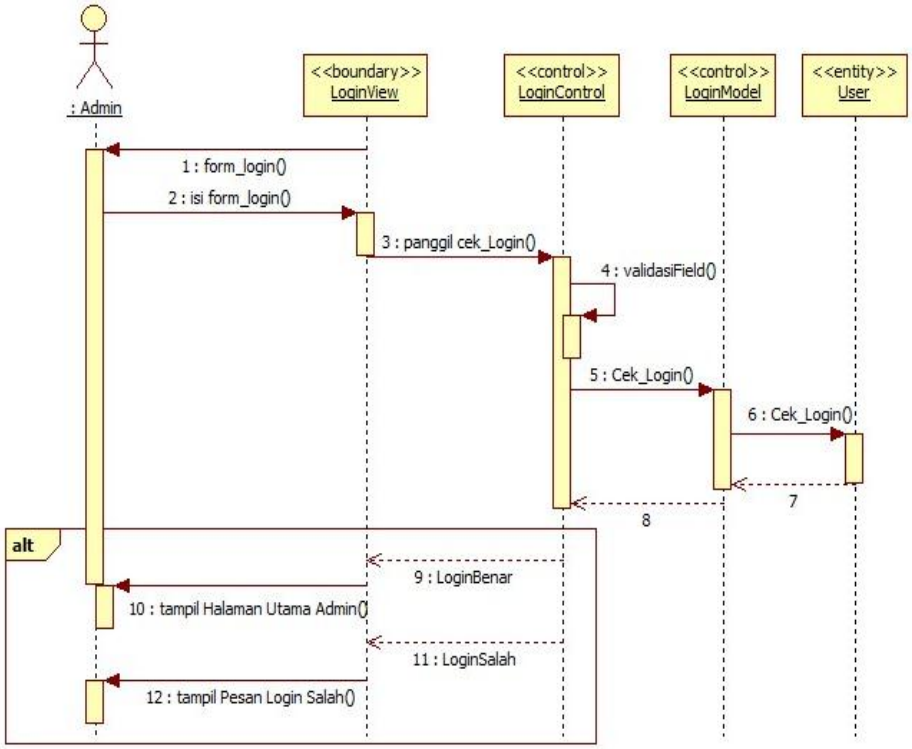
Identifikasi Masalah	
Nama	Ubah Password
Tujuan	Untuk merubah password
Deskripsi	Proses ini untuk merubah password
Aktor	User, Asman pemohon, Asman Sdm
Usecase Yang Berkaitan	Login
Skenario Utama	
Kondisi Awal	Tampilan *Form* Ubah Password
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu ubah password	2. Menampilkan *form* ubah password
2. mengisi inputan password lama yang ingin dirubah dengan password baru	
	3. Menyimpan hasil proses yang telah dilakukan
Kondisi Akhir	*User* berhasil mengubah password

3.1.4.3 Sequence Diagram

Sequence diagram menggambarkan interaksi antar masing-masing objek pada setiap *use case* dalam urutan waktu. Interaksi ini berupa pengiriman serangkaian data antar objek-objek yang saling berinteraksi. Berikut ini penjelasan dari masing-masing *sequence diagram* :

3.1.4.3.1 Sequence Diagram Login

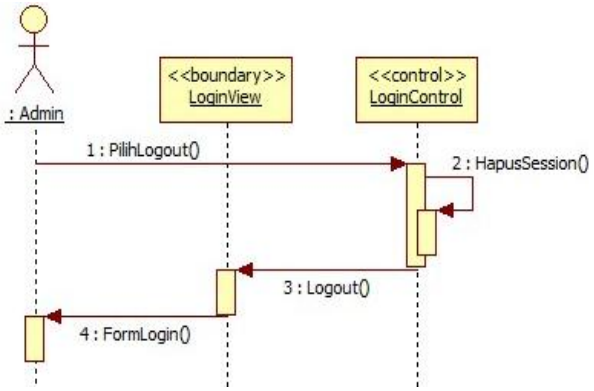
Sequence diagram login dapat dilihat pada gambar 3.5.



Gambar 3.5 *Sequence Diagram* Login

3.1.4.3.2 Sequence Diagram Logout

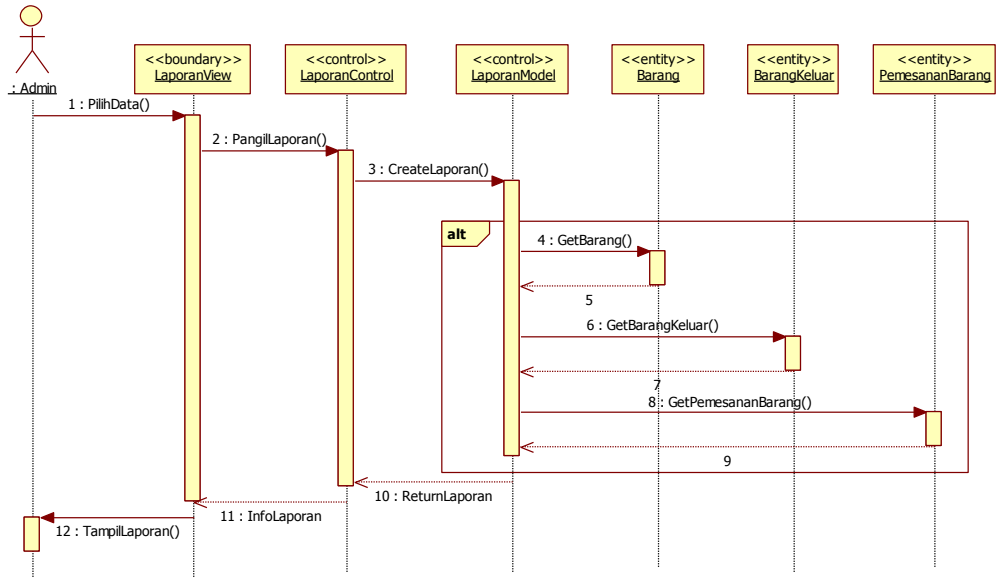
Sequence diagram logout dapat dilihat pada gambar 3.6.



Gambar 3.6 *Sequence Diagram* Logout

3.1.4.3.3 *Sequence Diagram* Pembuatan Laporan

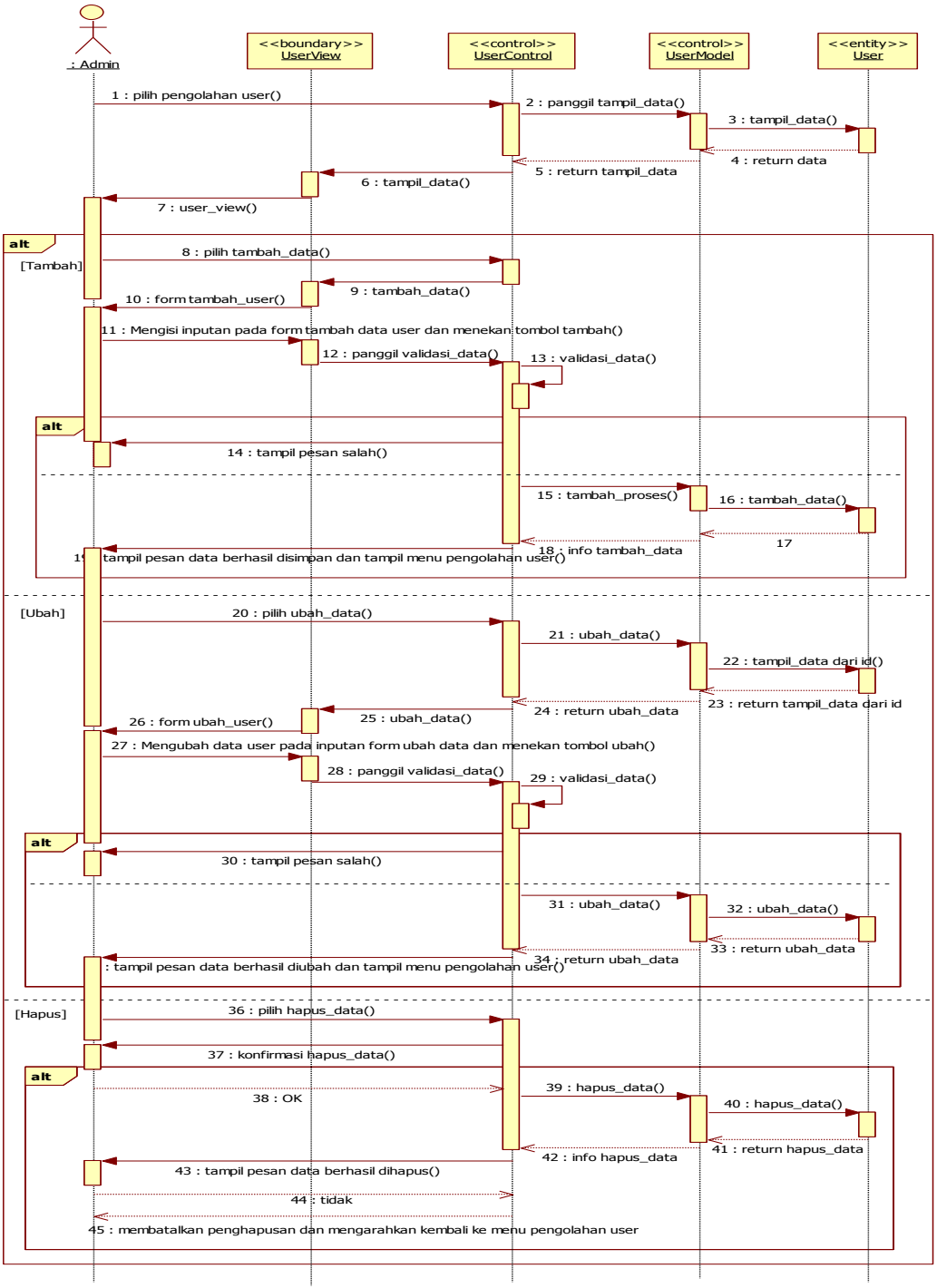
Sequence diagram pembuatan laporan dapat dilihat pada gambar 3.10.



Gambar 3.10 *Sequence Diagram* Pembuatan Laporan

3.1.4.3.4 Sequence Diagram Pengolahan Data User

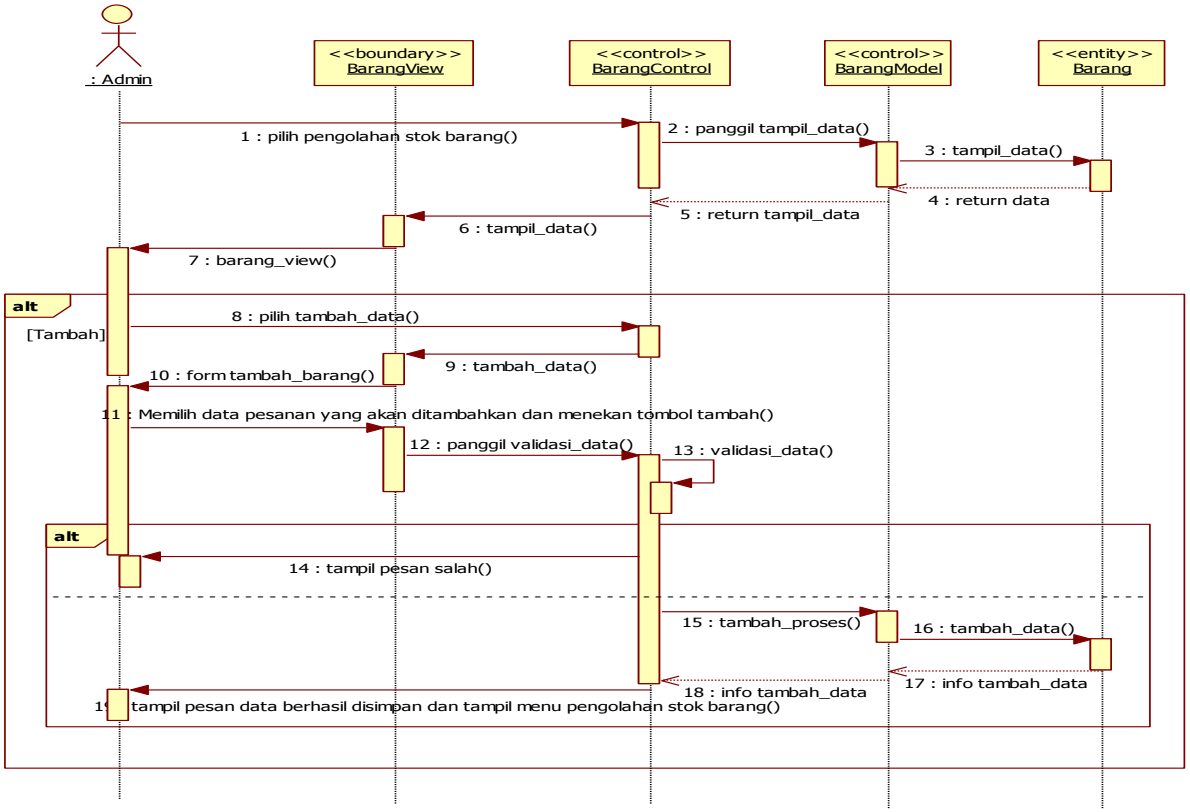
Sequence diagram pengolahan data *user* dapat dilihat pada gambar 3.7.



Gambar 3.7 *Sequence Diagram* Pengolahan Data User

3.1.4.3.5 Sequence Diagram Pengolahan Stok Barang

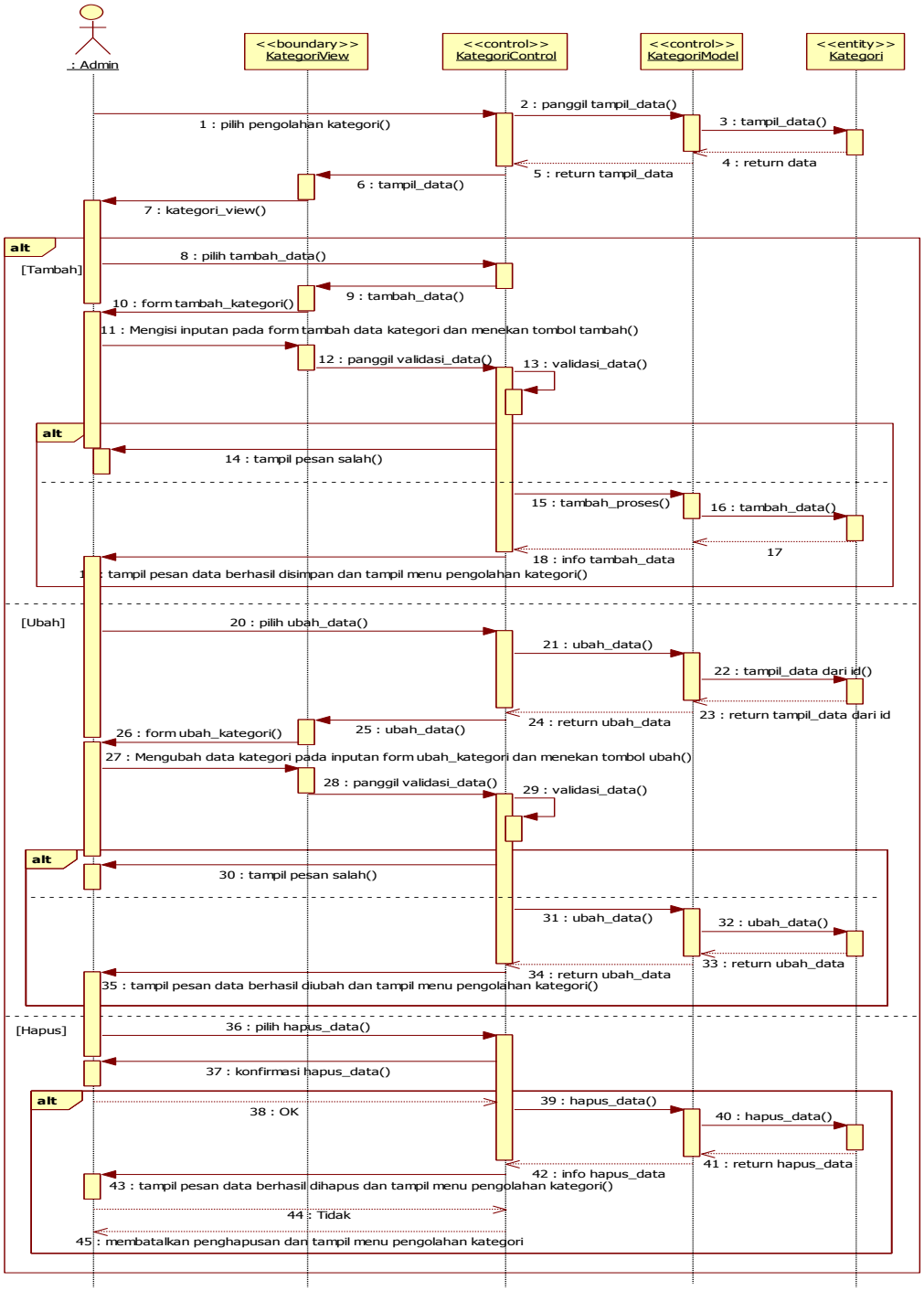
Sequence diagram pengolahan stok barang dapat dilihat pada gambar 3.8.



Gambar 3.8 *Sequence Diagram* Pengolahan Stok Barang

3.1.4.3.6 Sequence Diagram Pengolahan Kategori Barang

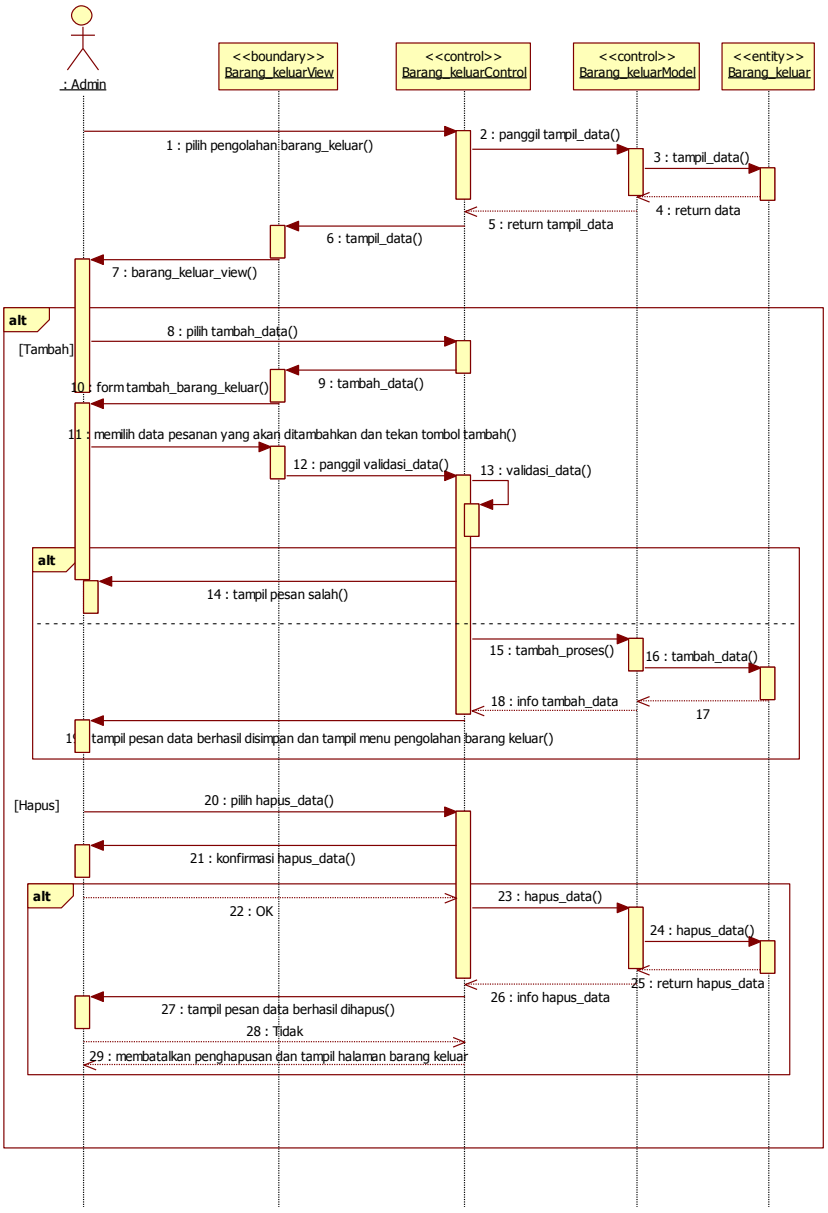
Sequence diagram pengolahan kategori barang dapat dilihat pada gambar 3.9.



Gambar 3.9 *Sequence Diagram* Pengolahan Kategori Barang

3.1.4.3.7 *Sequence Diagram* Barang Keluar

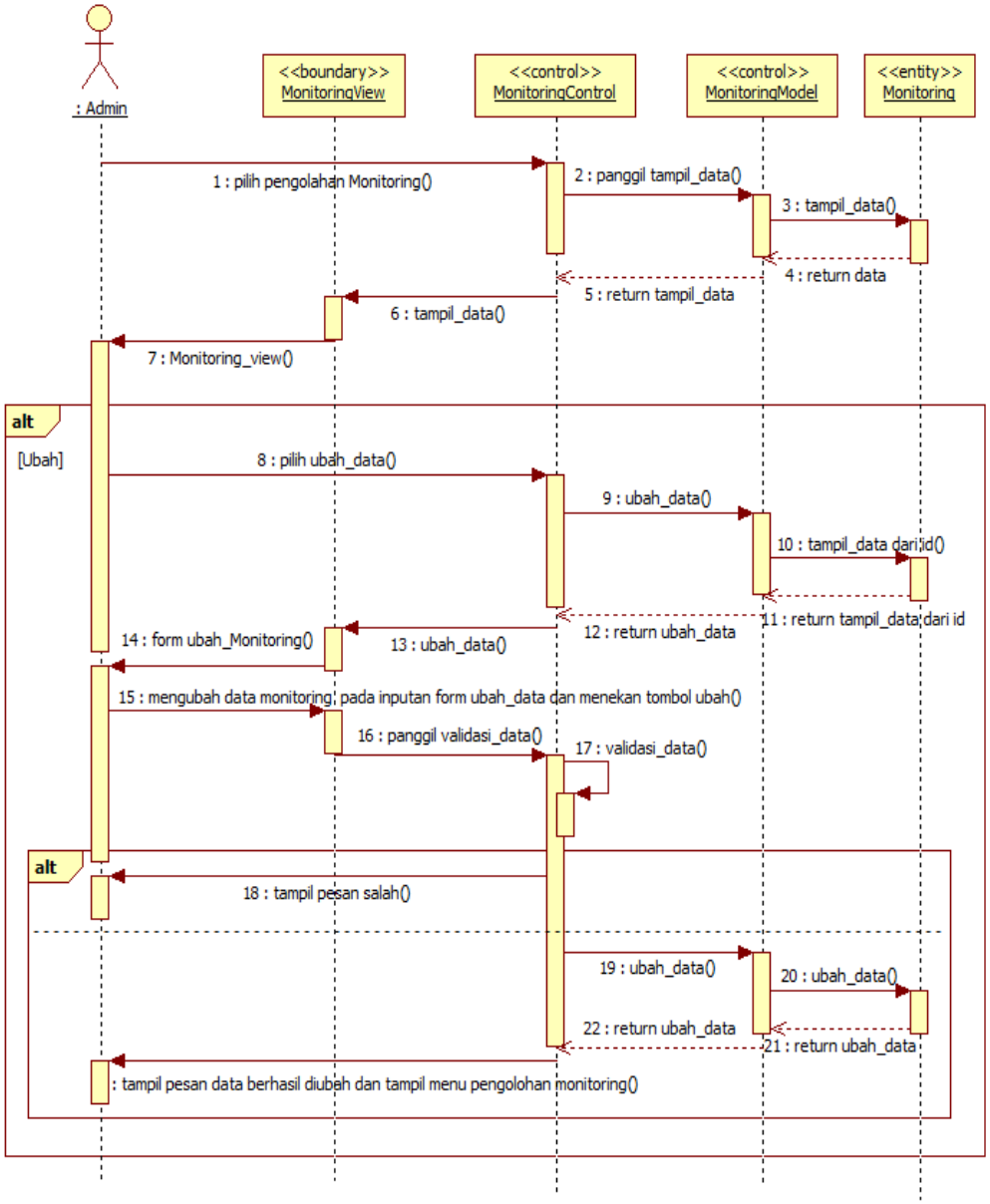
Sequence diagram pengolahan barang keluar dapat dilihat pada gambar 3.11.



Gambar 3.11 *Sequence Diagram* Barang Keluar

3.1.4.3.8 *Sequence Diagram* Monitoring Barang

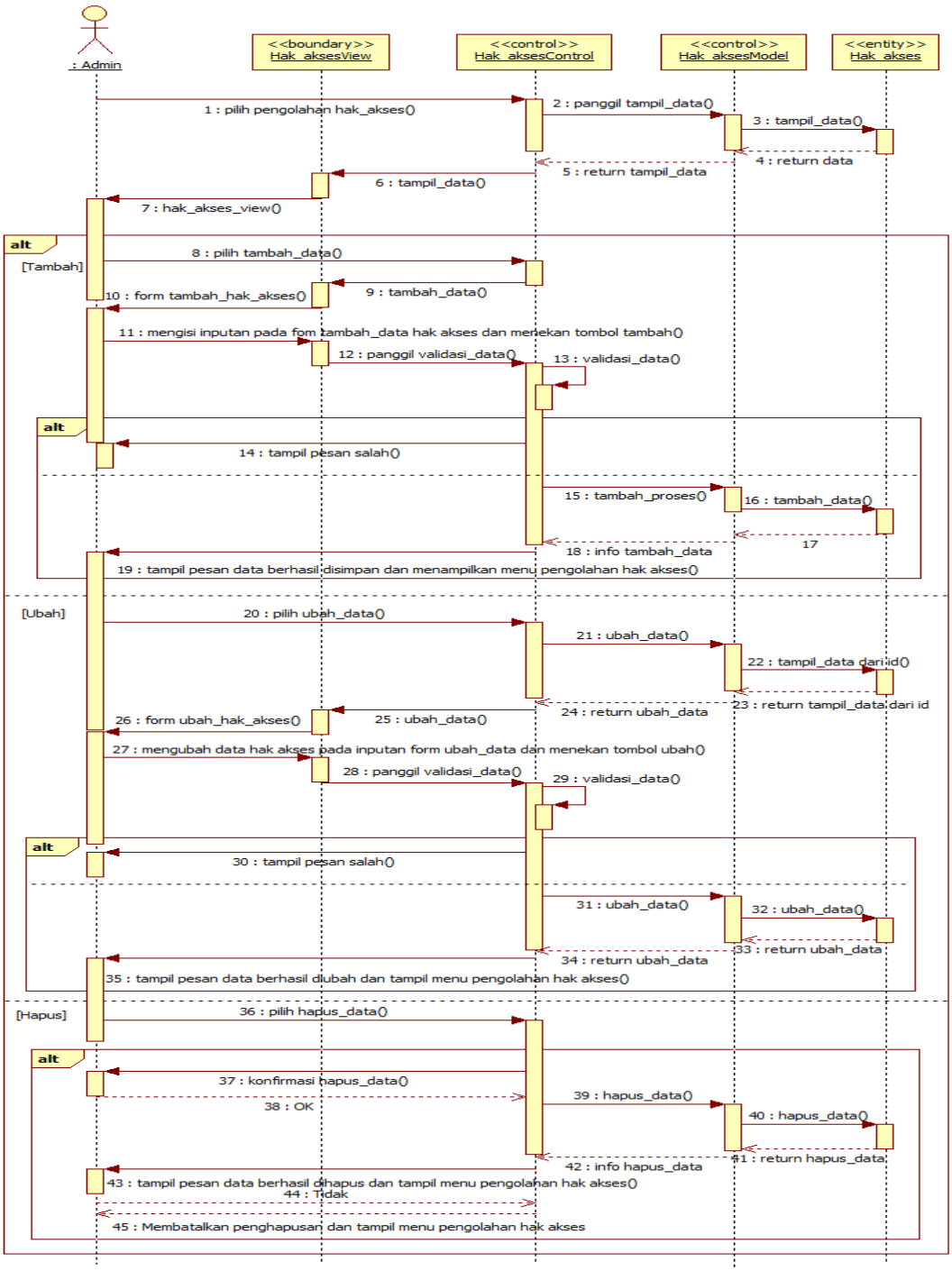
Sequence diagram pengolahan monitoring barang dapat dilihat pada gambar 3.13.



Gambar 3.13 *Sequence Diagram* Monitoring Barang

3.1.4.3.9 *Sequence Diagram* Pengolahan Hak Akses

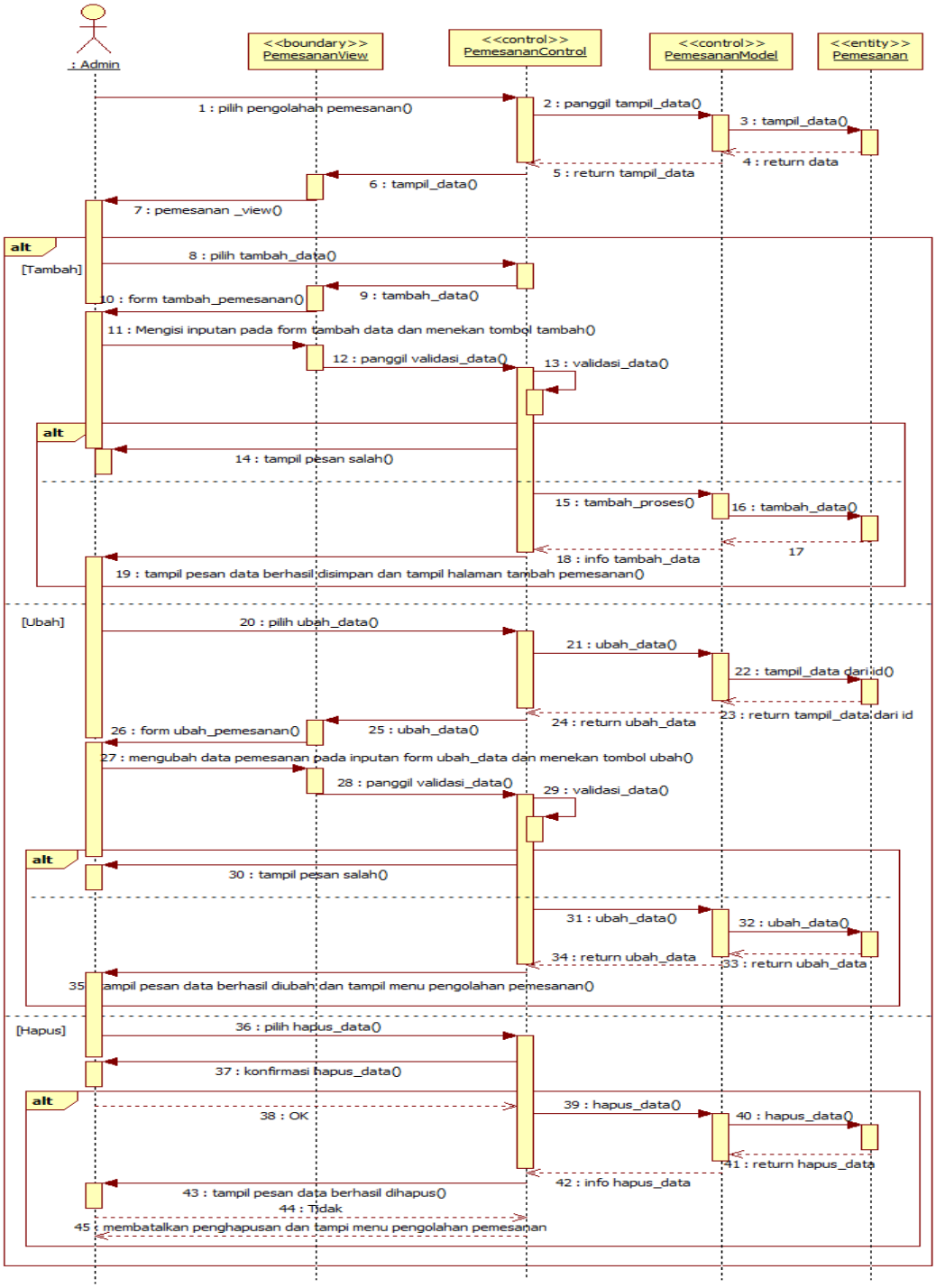
Sequence diagram pengolahan hak akses dapat dilihat pada gambar 3.14.



Gambar 3.14 *Sequence Diagram* Pengolahan Hak Akses

3.1.4.3.10 *Sequence Diagram* Pemesanan Barang

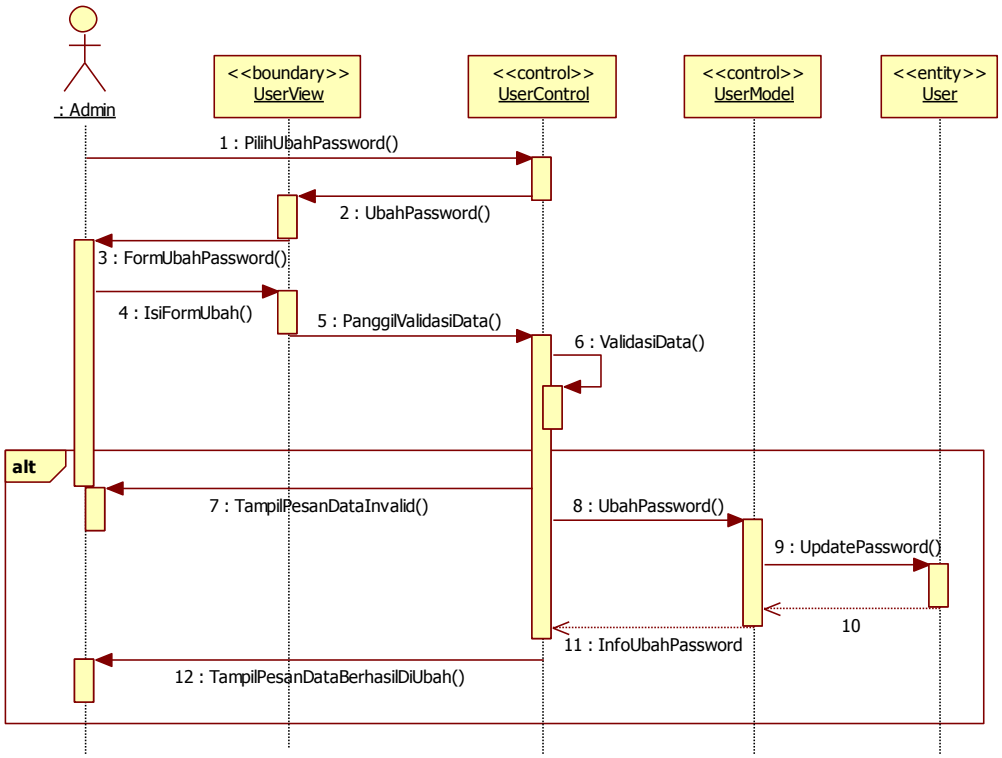
Sequence diagram pemesanan barang dapat dilihat pada gambar 3.15.



Gambar 3.15 *Sequence Diagram* Pemesanan Barang

3.1.4.3.11 *Sequence Diagram* Ubah Password

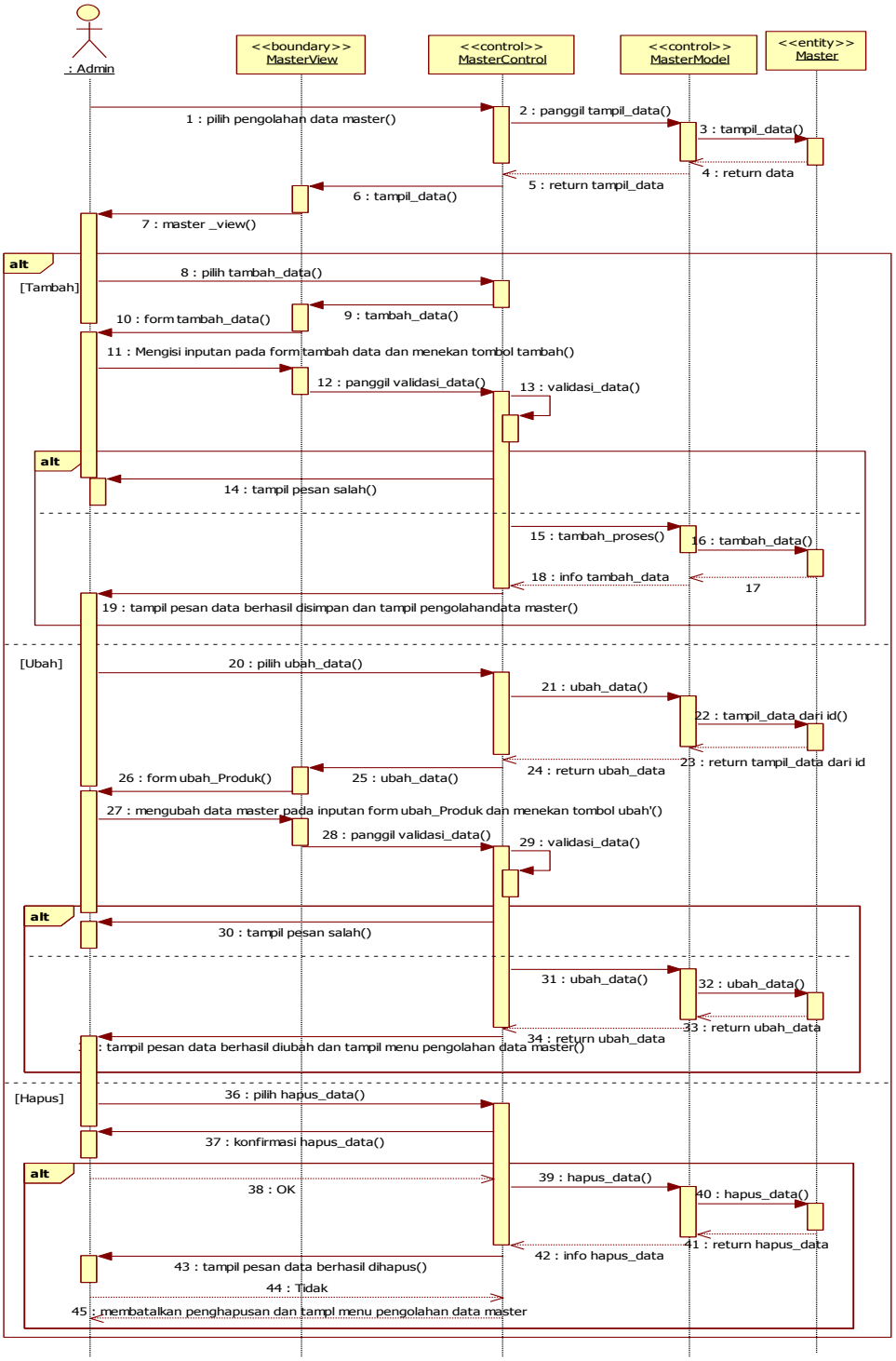
Sequence diagram ubah password dapat dilihat pada gambar 3.16.



Gambar 3.16 *Sequence Diagram* Ubah Password

3.1.4.3.12 Sequence Diagram Pengolahan Data Master

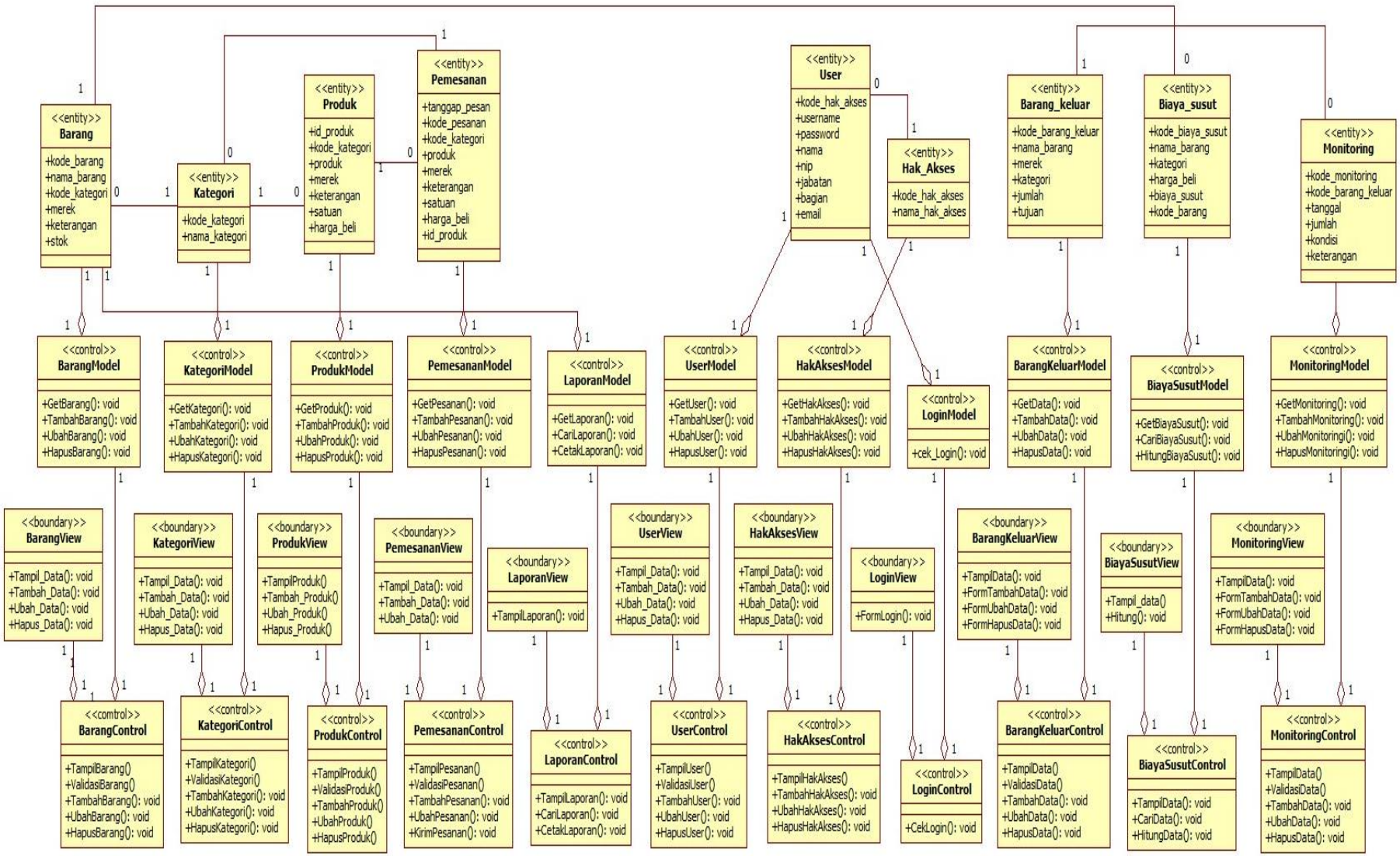
Sequence diagram pengolahan data master dapat dilihat pada gambar 3.17.



Gambar 3.17 *Sequence Diagram* Pengolahan Data Master

3.1.4.4 Class Diagram

Class diagram menggambarkan struktur dan hubungan antar objek-objek yang ada pada sistem. Struktur itu meliputi atribut-atribut dan method-method yang ada pada masing-masing kelas. Hubungannya dari masing-masing kelas yang ada tersebut digambarkan dengan menggunakan pewarisan dan generalilasi seperti terlihat pada gambar 3.18.



Gambar 3.18 ***Class Diagram*** **Sistem Informasi** ***Inventory Control***

Setiap kelas pada *class diagram* gambar 3.18 memiliki jenis kelasnya masing-masing seperti yang terlihat pada tabel 3.18.

Tabel 3.18 Deskripsi kelas

No.	Nama Kelas	Jenis Kelas
1.	LoginView	Boundary
2.	UserView	Boundary
3.	BarangView	Boundary
4.	KategoriView	Boundary
5.	PemesananView	Boundary
6.	BarangKeluarView	Boundary
7.	LaporanView	Boundary
8.	BiayaSusutView	Boundary
9.	MonitoringView	Boundary
10.	HakAksesView	Boundary
11.	ProdukView	Boundary
12.	LoginControl	Control
13.	LoginModel	Control
14.	UserControl	Control
15.	UserModel	Control
16.	BarangControl	Control
17.	BarangModel	Control
18.	KategoriControl	Control
19.	KategoriModel	Control
20.	PemesananControl	Control
21.	PemesananModel	Control
22.	BarangKeluarControl	Control
23.	BarangKeluarModel	Control
24.	LaporanControl	Control
25.	LaporanModel	Control
26.	BiayaSusutControl	Control
27.	BiayaSusutModel	Control
28.	MonitoringControl	Control
29.	MonitoringModel	Control
30.	HakAksesControl	Control
31.	HakAksesModel	Control
32.	ProdukControl	Control
33.	ProdukModel	Control
34.	User	Entity
35.	Barang	Entity
36.	Kategori	Entity
37.	Pemesanan	Entity
38.	BarangKeluar	Entity
39.	BiayaSusut	Entity
40.	Monitoring	Entity
41.	HakAkses	Entity
42.	Produk	Entity

Deskripsi dari setiap *class* yang ada dijelaskan sebagai berikut :

1. *Class* BarangControl

Class BarangControl merupakan *class* yang digunakan untuk mengatur proses pengolahan pada data barang dan bertugas untuk memanggil *class-class* yang lainnya.

Tabel 3.19 *Class* BarangControl

Nama Operasi	Keterangan
TampilBarang	Prosedur yang dipanggil ketika user meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *class* BarangModel untuk mengambil data pada database.
TambahBarang	Meminta kelas BarangModel untuk menambah data barang dan menyimpannya ke *database* dan meminta kelas BarangView untuk menampilkan form Tambah barang
UbahBarang	Meminta kelas BarangModel untuk Mengubah data barang dan meminta kelas BarangView untuk menampilkan form ubah barang
HapusBarang	Meminta kelas BarangModel untuk Menghapus data barang dan meminta kelas BarangView untuk menampilkan form hapus barang
ValidasiBarang	Prosedur yang dipanggil untuk memvalidasi data barang yang dimasukan pada form tambah atau ubah, akan menampilkan kesalahan jika data yang dimasukan tidak valid.

2. *Class* BarangModel

Class BarangModel merupakan *class* yang berisi prosedur-prosedur untuk mengolah data barang. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3.20 *Class* BarangModel

Nama Operasi	Keterangan
GetBarang	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data barang.
TambahBarang	Prosedur yang dipanggil ketika proses tambah data barang, melakukan perintah simpan data pada *database*.
UbahBarang	Prosedur yang dipanggil ketika proses ubah data barang,

	melakukan perintah update data pada *database*.
HapusBarang	Prosedur yang dipanggil ketika hapus data, menghapus data barang pada *database*.

3. *Class* BarangView

Class BarangView merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user.

Tabel 3.21 *Class* BarangView

Nama Operasi	**Keterangan**	
TampilBarang	Menampilkan data barang	
FormTambahBarang	menampilkan form tambah barang	
FormUbahBarang	menampilkan form ubah barang	
FormHapusBarang	menampilkan form hapus barang	
Nama Atribut	**Tipe**	**Keterangan**
kode_barang	int	Primary Key
nama_barang	varchar	
merk	varchar	
kategori	varchar	
keterangan	varchar	
stok	int	
tanggal	date	

4. *Class* LoginControl

Class LoginControl merupakan *class* yang digunakan untuk mengatur proses login dan bertugas untuk memanggil *class-class* yang lainnya.

Tabel 3.22 *Class* LoginControl

Nama Operasi	**Keterangan**
CekLogin	Prosedur yang dipanggil ketika user melakukan proses login, memvalidasi dan memanggil *class* LoginModel untuk mencari data user yang ada pada database.

5. *Class* LoginModel

Class LoginModel merupakan *class* yang berisi prosedur-prosedur untuk mengolah data login. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3.23 *Class* LoginModel

Nama Operasi	Keterangan
Login() : void	Prosedur ketika user akan login, mengecek apakah data login yang dimasukan terdaftar pada database.

6. *Class* LoginView

Class LoginView merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan *user*.

Tabel 3.24 *Class* LoginView

Nama Operasi	Keterangan
FormLogin	Prosedur untuk menampilkan form yang akan digunakan untuk proses login.

7. *Class* UserControl

Class UserControl merupakan *class* yang berisi prosedur-prosedur untuk mengolah data *user*. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3.25 *Class* UserControl

Nama Operasi	Keterangan
TampilUser	Prosedur yang dipanggil ketika user meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *class* UserModel untuk mengambil data pada database.
TambahUser	Meminta kelas UserModel untuk menambah data user dan menyimpannya ke *database* dan meminta kelas UserView untuk menampilkan form Tambah user

UbahUser	Meminta kelas UserModel untuk Mengubah data user dan meminta kelas UserView untuk menampilkan form ubah user
HapusUser	Meminta kelas UserModel untuk Menghapus data user dan meminta kelas UserView untuk menampilkan form hapus user
ValidasiUser	Prosedur yang dipanggil untuk memvalidasi data user yang dimasukan pada form tambah atau ubah, akan menampilkan kesalahan jika data yang dimasukan tidak valid.

8. *Class* UserModel

Class UserModel merupakan *class* yang berisi prosedur-prosedur untuk mengolah data user. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3.26 *Class* UserModel

Nama Operasi	Keterangan
GetUser	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data user.
TambahUser	Prosedur yang dipanggil ketika proses tambah data user, melakukan perintah simpan data pada *database*.
UbahUser	Prosedur yang dipanggil ketika proses ubah data user, melakukan perintah update data pada *database*.
HapusUser	Prosedur yang dipanggil ketika hapus data, menghapus data user pada *database*.

9. *Class* UserView

Class UserView merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user.

Tabel 3.27 *Class* UserView

Nama Operasi	Keterangan	
TampilUser	Menampilkan data user	
FormTambahUser	menampilkan form tambah user	
FormUbahUser	menampilkan form ubah user	
FormHapusUser	menampilkan form hapus user	
Nama Atribut	**Tipe**	**Keterangan**
username	varchar	Primary Key

password	varchar	
nama	varchar	
nip	varchar	
jabatan	varchar	
bagian	varchar	
email	varchar	
tanggal	date	

10. *Class* BarangKeluarControl

Class BarangKeluarControl merupakan *class* yang berisi prosedur-prosedur untuk mengolah data barang keluar. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3.28 *Class* BarangKeluarControl

Nama Operasi	Keterangan
TampilData	Prosedur yang dipanggil ketika user meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *class* BarangKeluarModel untuk mengambil data pada database.
TambahData	Meminta kelas BarangKeluarModel untuk menambah data barang keluar dan menyimpannya ke *database* dan meminta kelas BarangKeluarView untuk menampilkan form Tambah barang keluar
UbahData	Meminta kelas BarangKeluarModel untuk Mengubah data barang keluar dan meminta kelas BarangKeluarView untuk menampilkan form ubah barang keluar
HapusData	Meminta kelas BarangKeluarModel untuk Menghapus data barang keluar dan meminta kelas BarangKeluarView untuk menampilkan form hapus barang keluar
ValidasiData	Prosedur yang dipanggil untuk memvalidasi data barang keluar yang dimasukan pada form tambah atau ubah, akan menampilkan kesalahan jika data yang dimasukan tidak valid.

11. *Class* BarangKeluarModel

Class BarangKeluarModel merupakan *class* yang berisi prosedur-prosedur untuk mengolah data barang keluar. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3.29 *Class* BarangKeluarModel

Nama Operasi	Keterangan
GetData	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data barang keluar.
TambahData	Prosedur yang dipanggil ketika proses tambah data barang keluar, melakukan perintah simpan data pada *database*.
UbahData	Prosedur yang dipanggil ketika proses ubah data barang keluar, melakukan perintah update data pada *database*.
HapusData	Prosedur yang dipanggil ketika hapus data, menghapus data barang keluar pada *database*.

12. *Class* BarangKeluarView

Class BarangKeluarView merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user.

Tabel 3.30 *Class* BarangKeluarView

Nama Operasi	Keterangan	
TampilData	Menampilkan data barang keluar	
FormTambahData	menampilkan form tambah barang keluar	
FormUbahData	menampilkan form ubah barang keluar	
FormHapusData	menampilkan form hapus barang keluar	
Nama Atribut	**Tipe**	**Keterangan**
kodebarangkeluar	varchar	Primary Key
namabarang	varchar	
merk	varchar	
kategori	varchar	
keterangan	varchar	
jumlah	int	
tujuan	varchar	
tanggal	date	

13. *Class* KategoriControl

Class KategoriControl merupakan *class* yang berisi prosedur-prosedur untuk mengolah data kategori. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3.31 *Class* KategoriControl

Nama Operasi	Keterangan
TampilKategori	Prosedur yang dipanggil ketika user meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *class* KategoriModel untuk mengambil data pada database.
TambahKategori	Meminta kelas KategoriModel untuk menambah data kategori dan menyimpannya ke *database* dan meminta kelas KategoriView untuk menampilkan form Tambah kategori
UbahKategori	Meminta kelas KategoriModel untuk Mengubah data kategori dan meminta kelas KategoriView untuk menampilkan form ubah kategori
HapusKategori	Meminta kelas KategoriModel untuk Menghapus data kategori dan meminta kelas KategoriView untuk menampilkan form hapus kategori
ValidasiKategori	Prosedur yang dipanggil untuk memvalidasi data kategori yang dimasukan pada form tambah atau ubah, akan menampilkan kesalahan jika data yang dimasukan tidak valid.

14. *Class* KategoriModel

Class KategoriModel merupakan *class* yang berisi prosedur-prosedur untuk mengolah data kategori. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3.32 *Class* KategoriModel

Nama Operasi	Keterangan
GetKategori	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data kategori.
TambahKategori	Prosedur yang dipanggil ketika proses tambah data kategori, melakukan perintah simpan data pada

	database.
UbahKategori	Prosedur yang dipanggil ketika proses ubah data kategori, melakukan perintah update data pada *database*.
HapusKategori	Prosedur yang dipanggil ketika hapus data, menghapus data kategori pada *database*.

15. *Class* KategoriView

Class KategoriView merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user.

Tabel 3.33 *Class* KategoriView

Nama Operasi	**Keterangan**	
TampilData	Menampilkan data kategori	
FormTambahData	menampilkan form tambah kategori	
FormUbahData	menampilkan form ubah kategori	
FormHapusData	menampilkan form hapus kategori	
Nama Atribut	**Tipe**	**Keterangan**
kodekategori	varchar	Primary Key
namakategori	varchar	

16. *Class* PemesananControl

Class PemesananControl merupakan *class* yang berisi prosedur-prosedur untuk mengolah data pemesanan. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3.34 *Class* PemesananControl

Nama Operasi	**Keterangan**
TampilPesanan	Prosedur yang dipanggil ketika user meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *class* PemesananModel untuk mengambil data pada database.
TambahPesanan	Meminta kelas PemesananModel untuk menambah data pemesanan dan menyimpannya ke *database* dan meminta kelas PemesananView untuk menampilkan form Tambah pemesanan
UbahPesanan	Meminta kelas PemesananModel untuk Mengubah data pemesanan dan meminta kelas PemesananView untuk

	menampilkan form ubah pemesanan
HapusPesanan	Meminta kelas PemesananModel untuk Menghapus data pemesanan dan meminta kelas PemesananView untuk menampilkan form hapus pemesanan
ValidasiPesanan	Prosedur yang dipanggil untuk memvalidasi data pemesanan yang dimasukan pada form tambah atau ubah, akan menampilkan kesalahan jika data yang dimasukan tidak valid.

17. *Class* PemesananModel

Class PemesananModel merupakan *class* yang berisi prosedur-prosedur untuk mengolah data pemesanan. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3.35 *Class* PemesananModel

Nama Operasi	Keterangan
GetPesanan	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data pemesanan.
TambahPesanan	Prosedur yang dipanggil ketika proses tambah data pemesanan, melakukan perintah simpan data pada *database*.
UbahPesanan	Prosedur yang dipanggil ketika proses ubah data pemesanan, melakukan perintah update data pada *database*.
HapusPesanan	Prosedur yang dipanggil ketika hapus data, menghapus data pemesanan pada *database*.

18. *Class* PemesananView

Class PemesananView merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user.

Tabel 3.36 *Class* PemesananView

Nama Operasi	Keterangan
TampilData	Menampilkan data pemesanan
FormTambahData	menampilkan form tambah pemesanan
FormUbahData	menampilkan form ubah pemesanan
FormHapusData	menampilkan form hapus pemesanan

Nama Atribut	Tipe	Keterangan
kodepesanan	varchar	Primary Key
namabarang	varchar	
kodekategori	varchar	
merk	varchar	
keterangan	varchar	
jumlah	int	
tanggal	date	

19. *Class* HakAksesControl

Class HakAksesControl merupakan *class* yang berisi prosedur-prosedur untuk mengolah data hak akses. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3.37 *Class* HakAksesControl

Nama Operasi	Keterangan
TampilHakAkses	Prosedur yang dipanggil ketika user meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *class* HakAksesModel untuk mengambil data pada database.
TambahHakAkses	Meminta kelas HakAksesModel untuk menambah data hak akses dan menyimpannya ke *database* dan meminta kelas HakAksesView untuk menampilkan form Tambah hak akses
UbahHakAkses	Meminta kelas HakAksesModel untuk Mengubah data hak akses dan meminta kelas HakAksesView untuk menampilkan form ubah hak akses
HapusHakAkses	Meminta kelas HakAksesModel untuk Menghapus data hak akses dan meminta kelas HakAksesView untuk menampilkan form hapus hak akses

20. *Class* HakAksesModel

Class HakAksesModel merupakan *class* yang berisi prosedur-prosedur untuk mengolah data hak akses. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3.38 *Class* HakAksesModel

Nama Operasi	Keterangan
GetHakAkses	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data hak akses.
TambahHakAkses	Prosedur yang dipanggil ketika proses tambah data hak akses, melakukan perintah simpan data pada *database*.
UbahHakAkses	Prosedur yang dipanggil ketika proses ubah data hak akses, melakukan perintah update data pada *database*.
HapusHakAkses	Prosedur yang dipanggil ketika hapus data, menghapus data hak akses pada *database*.

21. *Class* HakAksesView

Class HakAksesView merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user.

Tabel 3.39 *Class* HakAksesView

Nama Operasi	Keterangan	
TampilData	Menampilkan data hak akses	
FormTambahData	menampilkan form tambah akses	
FormUbahData	menampilkan form ubah akses	
FormHapusData	menampilkan form hapus akses	
Nama Atribut	**Tipe**	**Keterangan**
namahakakses	varchar	Primary Key

22. *Class* LaporanControl

Class LaporanControl merupakan *class* yang berisi prosedur-prosedur untuk mengatur proses pembuatan laporan dan memanggil *class-class* lain yang berkaitan.

Tabel 3.40 *Class* LaporanControl

Nama Operasi	Keterangan
TampilLaporan	Prosedur untuk melakukan pembuatan laporan form yang sudah diisi sebelumnya dan kemudian menampilkannya.
CetakLaporan	Prosedur untuk mencetak laporan yang sudah dibuat.

23. *Class* LaporanModel

Class LaporanModel merupakan *class* yang berisi prosedur-prosedur untuk mengolah data laporan. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3.41 *Class* LaporanModel

Nama Operasi	Keterangan
GetLaporan	Prosedur mengambil data dari database, digunakan untuk menampilkan seluruh data berdasarkan jangka waktu yang dimasukan.
CetakLaporan	Prosedur yang dipanggil ketika proses cetak laporan.

24. *Class* LaporanView

Class LaporanView merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user.

Tabel 3.42 *Class* LaporanView

Nama Operasi	Keterangan
TampilLaporan	Prosedur untuk menampilkan data laporan yang hasil prosedur dari *class* LaporanModel dan LaporanControl.

25. *Class* MonitoringControl

Class MonitoringControl merupakan *class* yang berisi prosedur-prosedur untuk mengolah data Monitoring. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3.43 *Class* MonitoringControl

Nama Operasi	Keterangan
TampilData	Prosedur yang dipanggil ketika user meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *class* MonitoringModel untuk mengambil data pada database.
TambahData	Meminta kelas MonitoringModel untuk menambah data monitoring dan menyimpannya ke *database* dan meminta kelas MonitoringView untuk menampilkan form Tambah

	monitoring
UbahData	Meminta kelas Monitoring odel untuk Mengubah data monitoring dan meminta kelas MonitoringView untuk menampilkan form ubah monitoring
HapusData	Meminta kelas MonitoringModel untuk Menghapus data monitoring dan meminta kelas MonitoringView untuk menampilkan form hapus monitoring

26. *Class* MonitoringModel

Class MonitoringModel merupakan *class* yang berisi prosedur-prosedur untuk mengolah data monitoring. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3.44 *Class* MonitoringModel

Nama Operasi	Keterangan
GetMonitoring	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data monitoring.
TambahMonitoring	Prosedur yang dipanggil ketika proses tambah data monitoring, melakukan perintah simpan data pada *database*.
UbahMonitoring	Prosedur yang dipanggil ketika proses ubah data monitoring, melakukan perintah update data pada *database*.
HapusMonitoring	Prosedur yang dipanggil ketika hapus data, menghapus data monitoring pada *database*.

27. *Class* MonitoringView

Class MonitoringView merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user.

Tabel 3.45 *Class* MonitoringView

Nama Operasi	Keterangan	
TampilData	Menampilkan data monitoring	
FormTambahData	menampilkan form tambah monitoring	
FormUbahData	menampilkan form ubah monitoring	
FormHapusData	menampilkan form hapus monitoring	
Nama Atribut	**Tipe**	**Keterangan**
kodemonitoring	varchar	Primary Key

kodebarang	varchar	
tanggal	date	
jumlah	varchar	
kondisibarang	varchar	
keterangan	varchar	

28. *Class* BiayaSusutControl

Class BiayaSusutControl merupakan *class* yang berisi prosedur-prosedur untuk mengolah data biaya susut. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil data dan menghitung biaya susut barang.

Tabel 3.46 *Class* BiayaSusutControl

Nama Operasi	Keterangan
TampilData	Prosedur yang dipanggil ketika user meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *class* BiayaSusutModel untuk mengambil data pada database.
HitungData	Prosedur menghitung jumlah biaya susut barang

29. *Class* BiayaSusutModel

Class BiayaSusutModel merupakan *class* yang berisi prosedur-prosedur untuk mengolah data biaya susut. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil data dan menghitung biaya susut barang.

Tabel 3.47 *Class* BiayaSusutModel

Nama Operasi	Keterangan
GetBiayaSusut	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data monitoring.
HitungBiayaSusut	Prosedur yang dipanggil ketika proses hitung pada data barang

30. *Class* BiayaSusutView

Class BiayaSusutView merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user.

Tabel 3.48 *Class* BiayaSusutView

Nama Operasi	**Keterangan**	
TampilBiayaSusut	Menampilkan form biaya susut	
HitungBiayaSusut	Menampilkan form perhitungan	
Nama Atribut	**Tipe**	**Keterangan**
kodebiayasusut	varchar	Primary Key
kodebarang	varchar	
namabarang	varchar	
biayasusut	varchar	
tanggal	varchar	

31. *Class* ProdukControl

Class ProdukControl merupakan *class* yang berisi prosedur-prosedur untuk mengolah data produk. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3.49 *Class* ProdukControl

Nama Operasi	**Keterangan**
TampilProduk	Prosedur yang dipanggil ketika user meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *class* ProdukModel untuk mengambil data pada database.
TambahProduk	Meminta kelas ProdukModel untuk menambah data produk dan menyimpannya ke *database* dan meminta kelas ProdukView untuk menampilkan form Tambah Produk
UbahProduk	Meminta kelas ProdukModel untuk Mengubah data produk dan meminta kelas ProdukView untuk menampilkan form ubah produk
HapusProduk	Meminta kelas ProdukModel untuk Menghapus data produk dan meminta kelas ProdukView untuk menampilkan form hapus produk
ValidasiProduk	Prosedur yang dipanggil untuk memvalidasi data produk yang dimasukan pada form tambah atau ubah, akan menampilkan kesalahan jika data yang dimasukan tidak valid.

32. *Class* ProdukModel

Class ProdukModel merupakan *class* yang berisi prosedur-prosedur untuk mengolah data produk. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3.50 *Class* ProdukModel

Nama Operasi	Keterangan
GetProduk	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data produk.
TambahProduk	Prosedur yang dipanggil ketika proses tambah data produk, melakukan perintah simpan data pada *database*.
UbahProduk	Prosedur yang dipanggil ketika proses ubah data produk, melakukan perintah update data pada *database*.
HapusProduk	Prosedur yang dipanggil ketika hapus data, menghapus data produk pada *database*.

33. *Class* ProdukView

Class ProdukView merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user.

Tabel 3.51 *Class* ProdukView

Nama Operasi	Keterangan	
TampilProduk	Menampilkan data Produk	
FormTambahProduk	menampilkan form tambah Produk	
FormUbahProduk	menampilkan form ubah Produk	
FormHapusProduk	menampilkan form hapus Produk	
Nama Atribut	**Tipe**	**Keterangan**
idproduk	varchar	Primary Key
kodekategori	varchar	
produk	varchar	
satuan	varchar	

34. *Class* MerekControl

Class MerekControl merupakan *class* yang berisi prosedur-prosedur untuk mengolah data merek. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3.52 *Class* MerekControl

Nama Operasi	Keterangan
TampilMerek	Prosedur yang dipanggil ketika user meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *class* MerekModel untuk mengambil data pada database.
TambahMerek	Meminta kelas MerekModel untuk menambah data merek dan menyimpannya ke *database* dan meminta kelas MerekView untuk menampilkan form Tambah merek
UbahMerek	Meminta kelas MerekModel untuk Mengubah data Merek dan meminta kelas MerekView untuk menampilkan form ubah Merek
HapusMerek	Meminta kelas MerekModel untuk Menghapus data Merek dan meminta kelas MerekView untuk menampilkan form hapus Merek
ValidasiMerek	Prosedur yang dipanggil untuk memvalidasi data Merek yang dimasukan pada form tambah atau ubah, akan menampilkan kesalahan jika data yang dimasukan tidak valid.

35. *Class* MerekModel

Class MerekModel merupakan *class* yang berisi prosedur-prosedur untuk mengolah data merek. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3.53 *Class* MerekModel

Nama Operasi	Keterangan
GetMerek	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data Merek.
TambahMerek	Prosedur yang dipanggil ketika proses tambah data Merek, melakukan perintah simpan data pada *database*.
UbahMerek	Prosedur yang dipanggil ketika proses ubah data Merek,

	melakukan perintah update data pada *database*.
HapusMerek	Prosedur yang dipanggil ketika hapus data, menghapus data Merekpada *database*.

36. *Class* MerekView

Class MerekView merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user.

Tabel 3.54 *Class* MerekView

Nama Operasi	**Keterangan**	
TampilMerek	Menampilkan data Merek	
FormTambahMerek	menampilkan form tambah Merek	
FormUbahMerek	menampilkan form ubah Merek	
FormHapusMerek	menampilkan form hapus Merek	
Nama Atribut	**Tipe**	**Keterangan**
idmerek	varchar	Primary Key
idproduk	varchar	
merek	varchar	

37. *Class* KeteranganControl

Class KeteranganControl merupakan *class* yang berisi prosedur-prosedur untuk mengolah data keterangan. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3.55 *Class* KeteranganControl

Nama Operasi	**Keterangan**
TampilKeterangan	Prosedur yang dipanggil ketika user meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *class* KeteranganModel untuk mengambil data pada database.
TambahKeterangan	Meminta kelas KeteranganModel untuk menambah data Keterangan dan menyimpannya ke *database* dan meminta kelas KeteranganView untuk menampilkan form Tambah Keterangan
UbahKeterangan	Meminta kelas KeteranganModel untuk Mengubah data Keterangandan meminta kelas KeteranganView untuk menampilkan form ubah Keterangan
HapusKeterangan	Meminta kelas KeteranganModel untuk Menghapus data

	Keterangandan meminta kelas KeteranganView untuk menampilkan form hapus Keterangan
ValidasiKeterangan	Prosedur yang dipanggil untuk memvalidasi data Keterangan yang dimasukan pada form tambah atau ubah, akan menampilkan kesalahan jika data yang dimasukan tidak valid.

38. *Class* KeteranganModel

Class KeteranganModel merupakan *class* yang berisi prosedur-prosedur untuk mengolah data Keterangan. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3.56 *Class* KeteranganModel

Nama Operasi	Keterangan
GetKeterangan	Prosedur mengambil data dari *database,* digunakan untuk menampilkan seluruh data Keterangan.
TambahKeterangan	Prosedur yang dipanggil ketika proses tambah data Keterangan, melakukan perintah simpan data pada *database*.
UbahKeterangan	Prosedur yang dipanggil ketika proses ubah data Keterangan, melakukan perintah update data pada *database*.
HapusKeterangan	Prosedur yang dipanggil ketika hapus data, menghapus data Keterangan pada *database*.

39. *Class* KeteranganView

Class KeteranganView merupakan *class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan user.

Tabel 3.57 *Class* KeteranganView

Nama Operasi	Keterangan	
TampilKeterangan	Menampilkan data Keterangan	
FormTambahKeterangan	menampilkan form tambah Keterangan	
FormUbahKeterangan	menampilkan form ubah Keterangan	
FormHapusKeterangan	menampilkan form hapus Keterangan	
Nama Atribut	**Tipe**	**Keterangan**
idketerangan	varchar	Primary Key

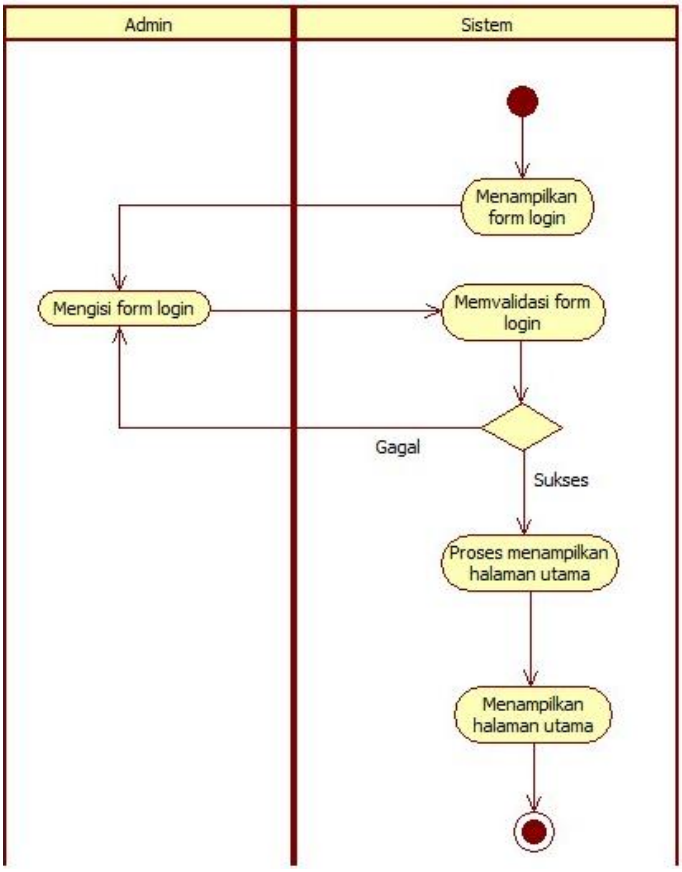
idproduk	varchar	
keterangan	varchar	

3.1.4.5 Activity Diagram

Activity diagram memodelkan aliran kerja atau *workflow* dari urutan aktifitas dalam suatu proses yang mengacu pada *use case diagram* yang ada. Berikut ini penjelasan dari masing-masing *activity diagram*:

3.1.4.5.1 Activity Diagram Login

Interaksi antara aktor pengguna dengan *use case* login dijelaskan dalam *activity diagram* sebagai berikut :



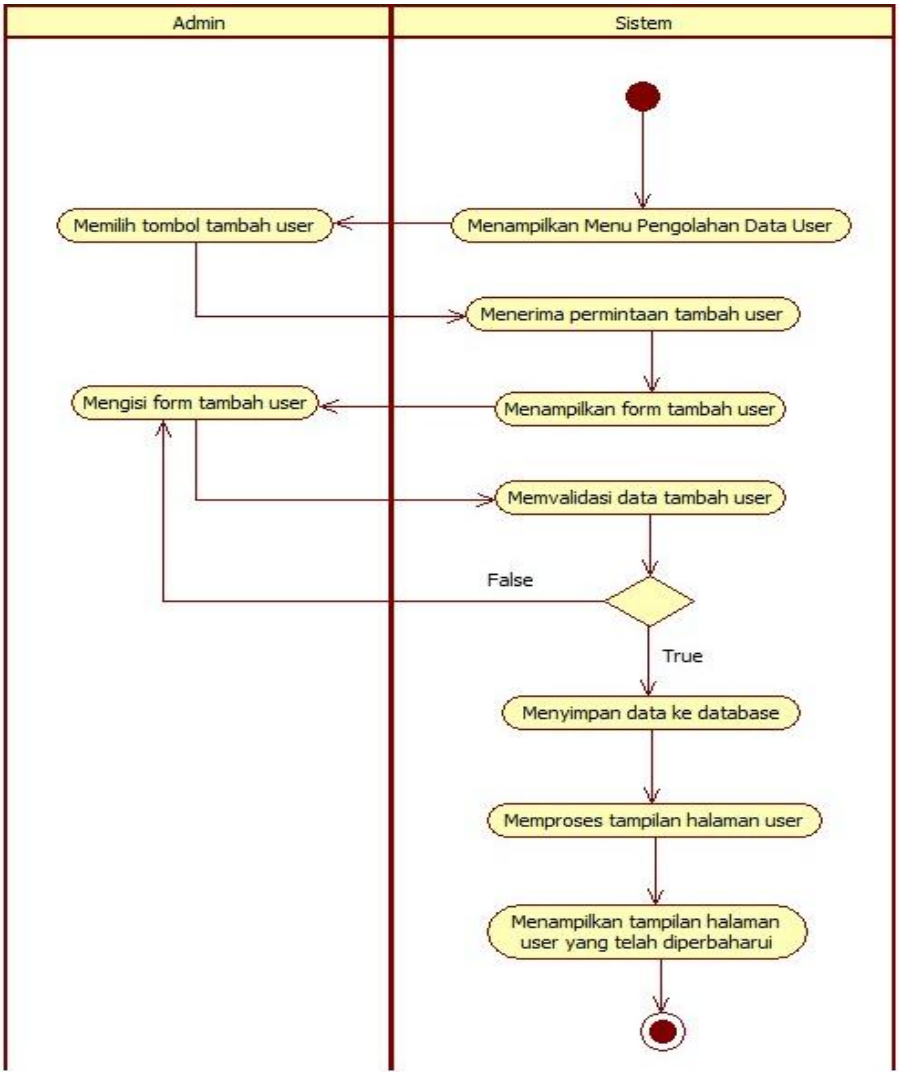
Gambar 3.19 *Activity Diagram* Login Admin

3.1.4.5.2 *Activity Diagram* Pengolahan Data User

Interaksi antara aktor pengguna dengan *use case* pengolahan data *user* dijelaskan dalam *activity diagram* tambah data *user*, ubah data *user* dan hapus data *user*. Berikut ini penjelasan dari masing-masing *activity diagram* tersebut:

3.1.4.5.3 *Activity Diagram* Tambah Data User

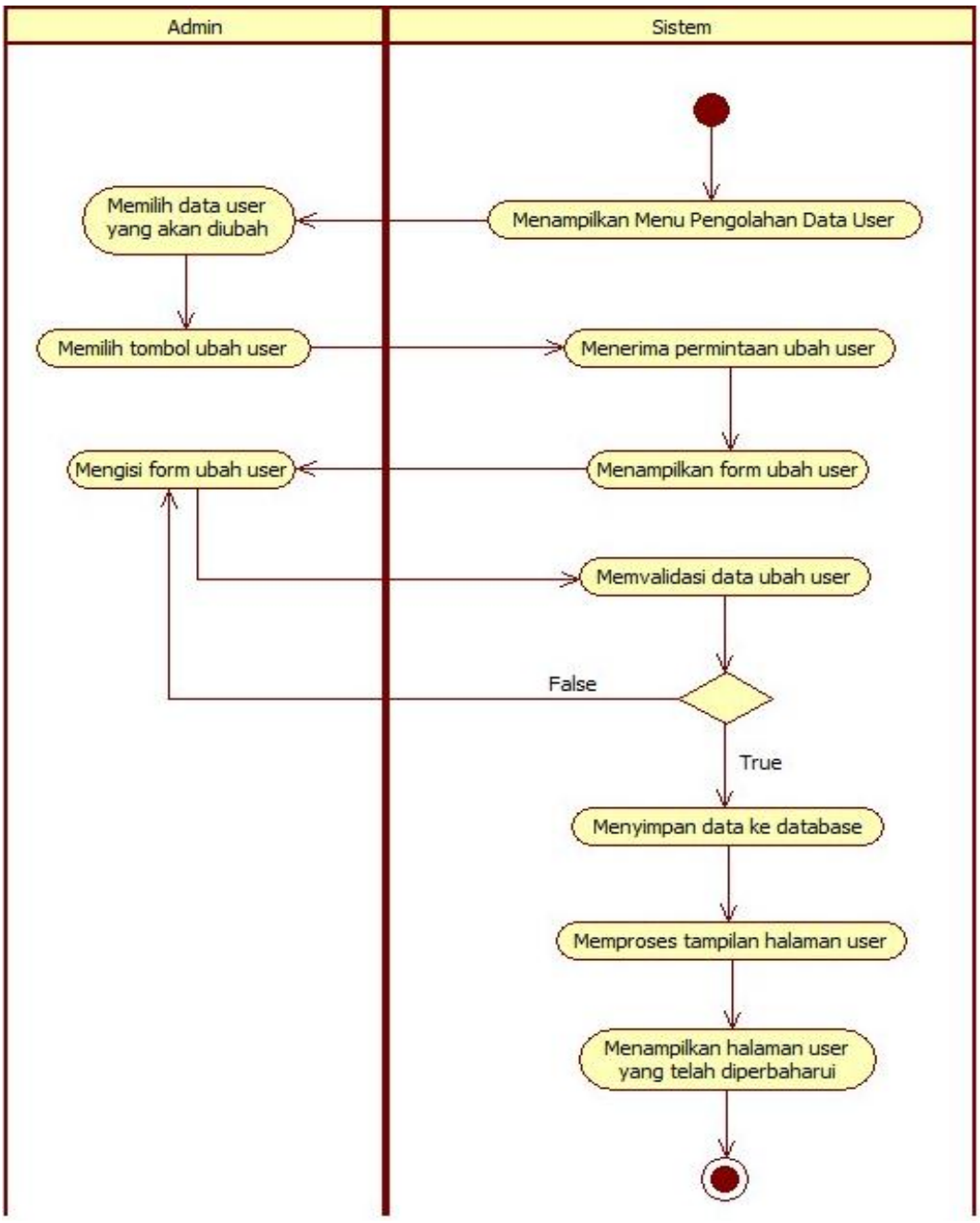
Interaksi antara aktor pengguna dengan *use case* tambah data *user* dijelaskan dalam *activity diagram* sebagai berikut :



Gambar 3.20 *Activity Diagram* Tambah Data User

3.1.4.5.4 *Activity Diagram* Ubah Data User

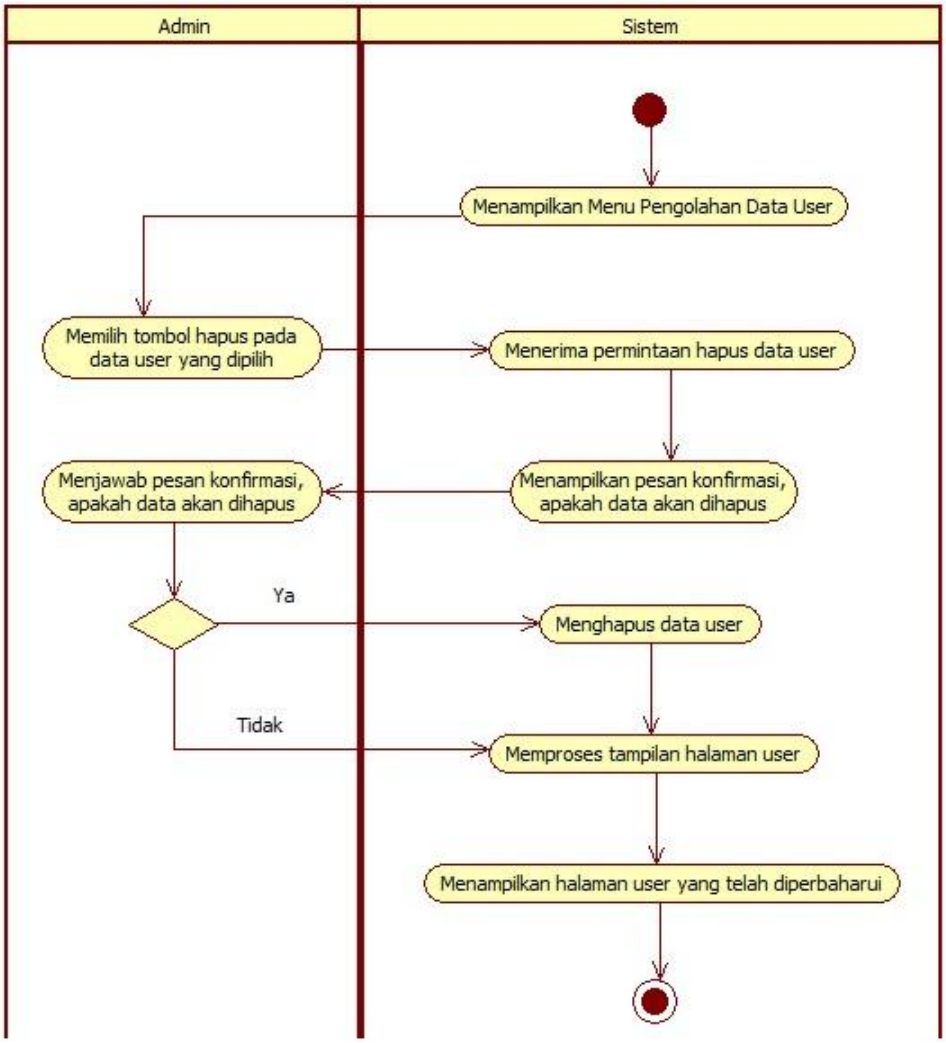
Interaksi antara aktor pengguna dengan *use case* ubah data *user* dijelaskan dalam *activity diagram* sebagai berikut :



Gambar 3.21 *Activity Diagram* Ubah Data User

3.1.4.5.5 *Activity Diagram* Hapus Data User

Interaksi antara aktor pengguna dengan *use case* hapus data *user* dijelaskan dalam *activity diagram* sebagai berikut :



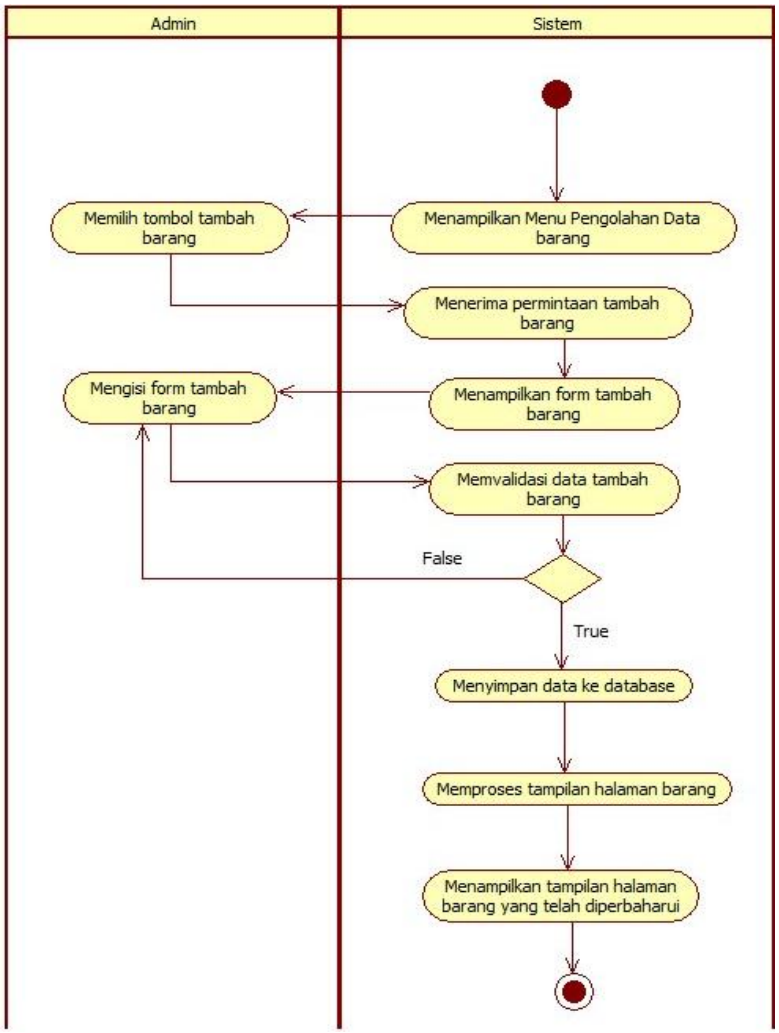
Gambar 3.22 *Activity Diagram* Hapus Data User

3.1.4.5.6 *Activity Diagram* Pengolahan Stok Barang

Interaksi antara aktor pengguna dengan *use case* pengolahan stok barang dijelaskan dalam *activity diagram* tambah stok barang, ubah stok barang dan hapus stok barang. Berikut ini penjelasan dari masing-masing *activity diagram* tersebut:

3.1.4.5.7 *Activity Diagram* Tambah Stok Barang

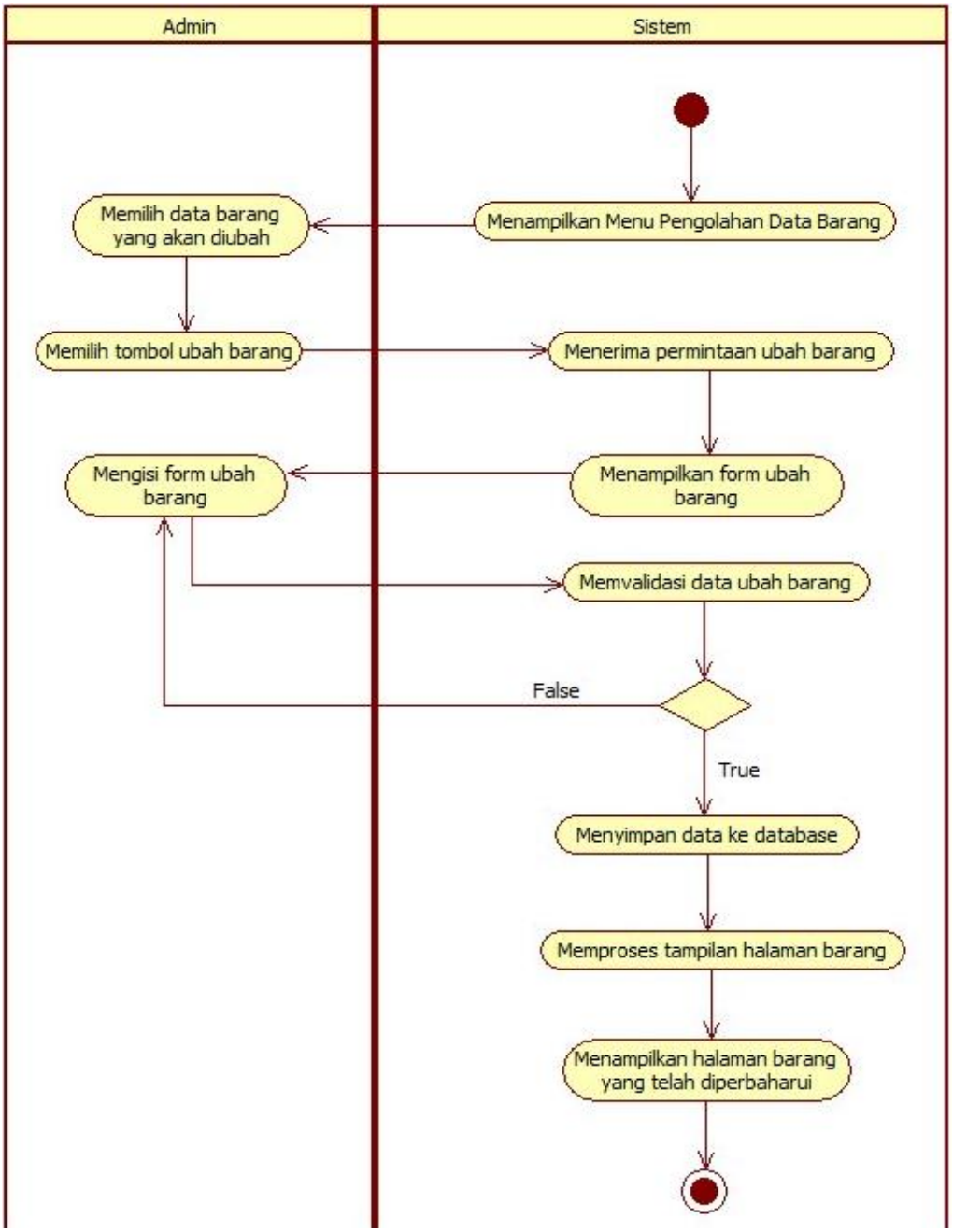
Interaksi antara aktor pengguna dengan *use case* tambah stok barang dijelaskan dalam *activity diagram* sebagai berikut :



Gambar 3.23 *Activity Diagram* Tambah Stok Barang

3.1.4.5.8 *Activity Diagram* Ubah Stok Barang

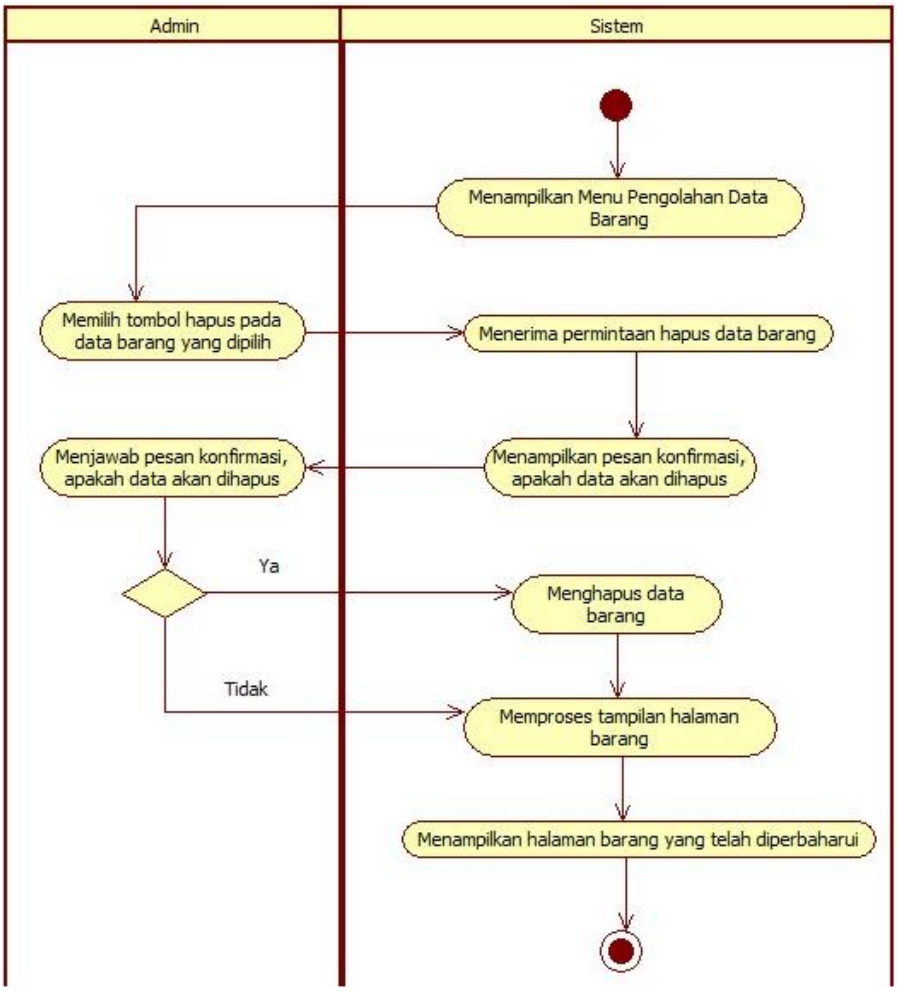
Interaksi antara aktor pengguna dengan *use case* ubah stok barang dijelaskan dalam *activity diagram* sebagai berikut :



Gambar 3.24 *Activity Diagram* Ubah Stok Barang

3.1.4.5.9 *Activity Diagram* Hapus Stok Barang

Interaksi antara aktor pengguna dengan *use case* hapus stok barang dijelaskan dalam *activity diagram* sebagai berikut :



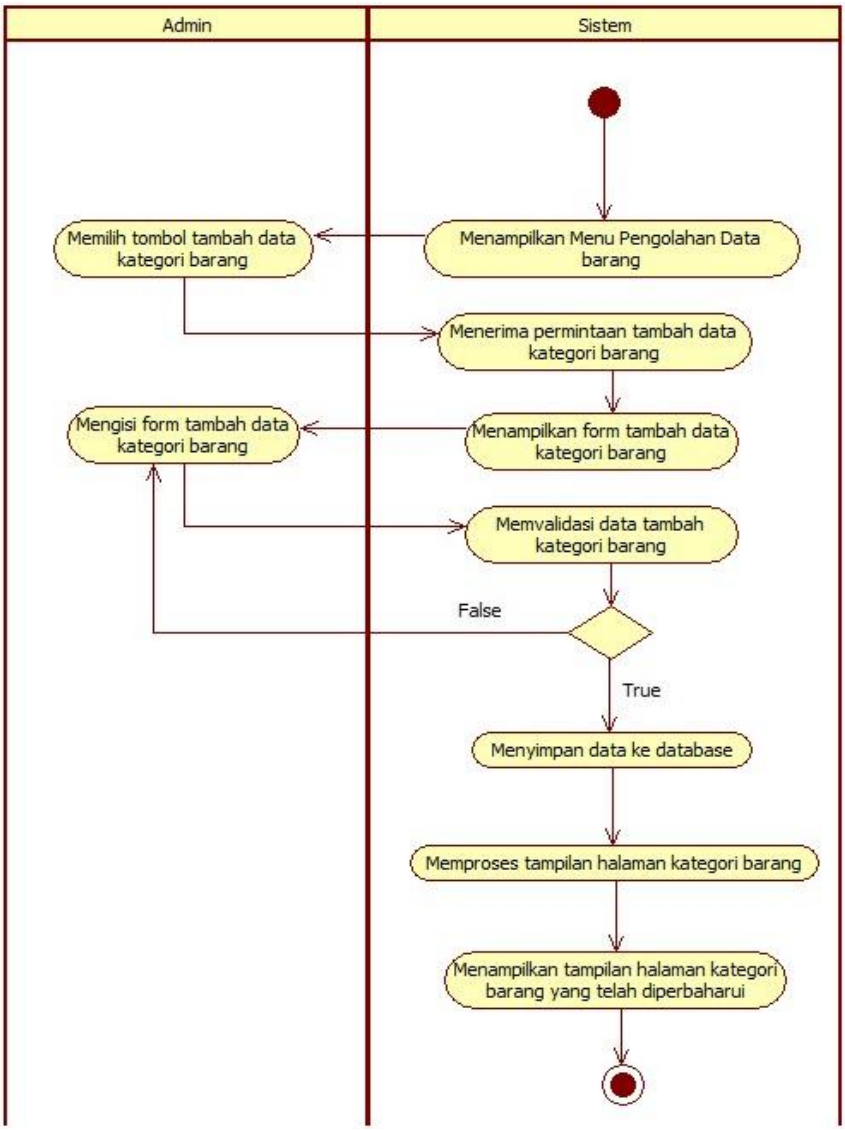
Gambar 3.25 *Activity Diagram* Hapus Stok Barang

3.1.4.5.10 *Activity Diagram* Pengolahan Kategori Barang

Interaksi antara aktor pengguna dengan *use case* pengolahan kategori barang dijelaskan dalam *activity diagram* tambah kategori barang, ubah kategori barang dan hapus kategori barang. Berikut ini penjelasan dari masing-masing *activity diagram* tersebut:

3.1.4.5.11 *Activity Diagram* Tambah Kategori Barang

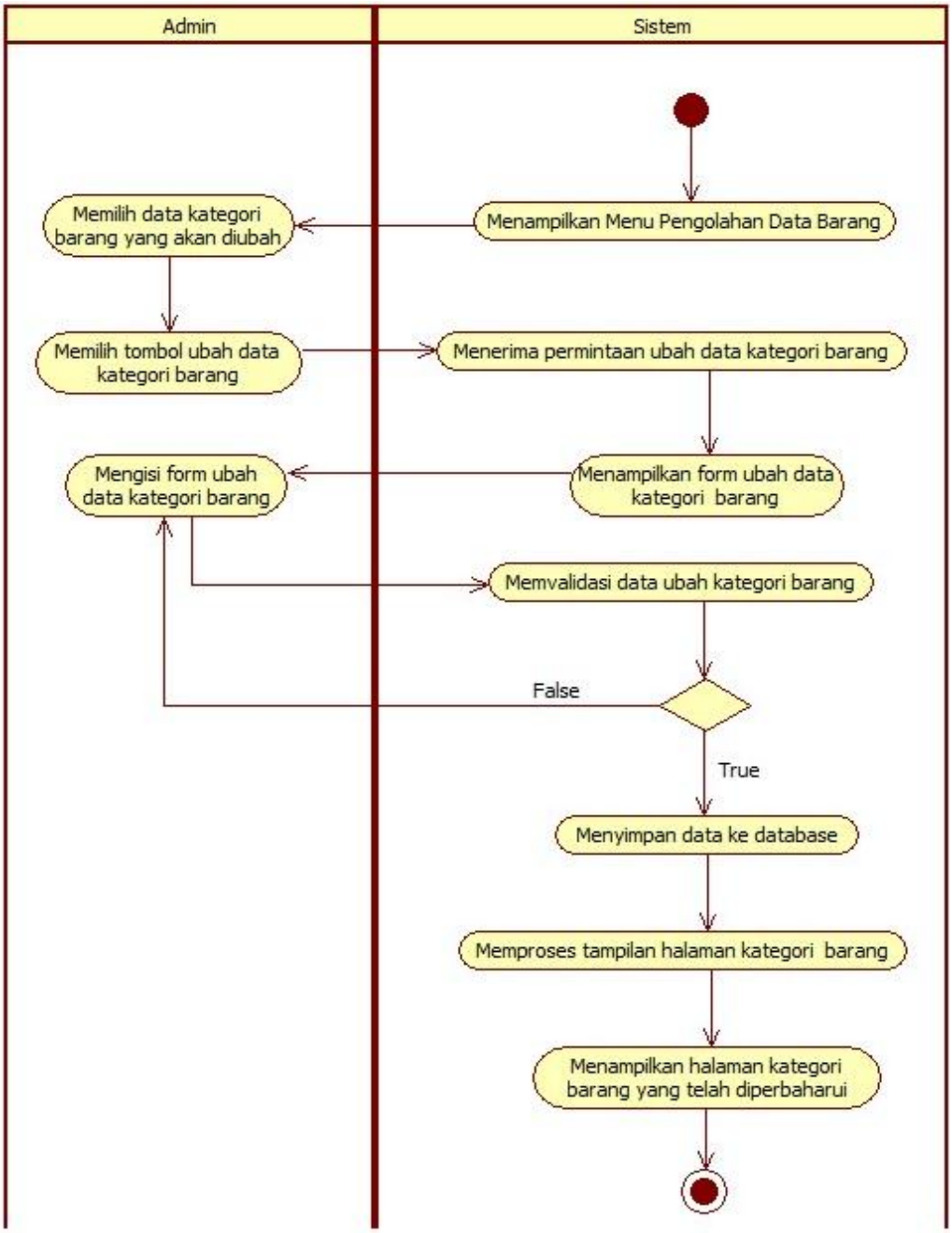
Interaksi antara aktor pengguna dengan *use case* tambah kategori barang dijelaskan dalam *activity diagram* sebagai berikut :



Gambar 3.26 *Activity Diagram* Tambah Kategori Barang

3.1.4.5.12 *Activity Diagram* Ubah Kategori Barang

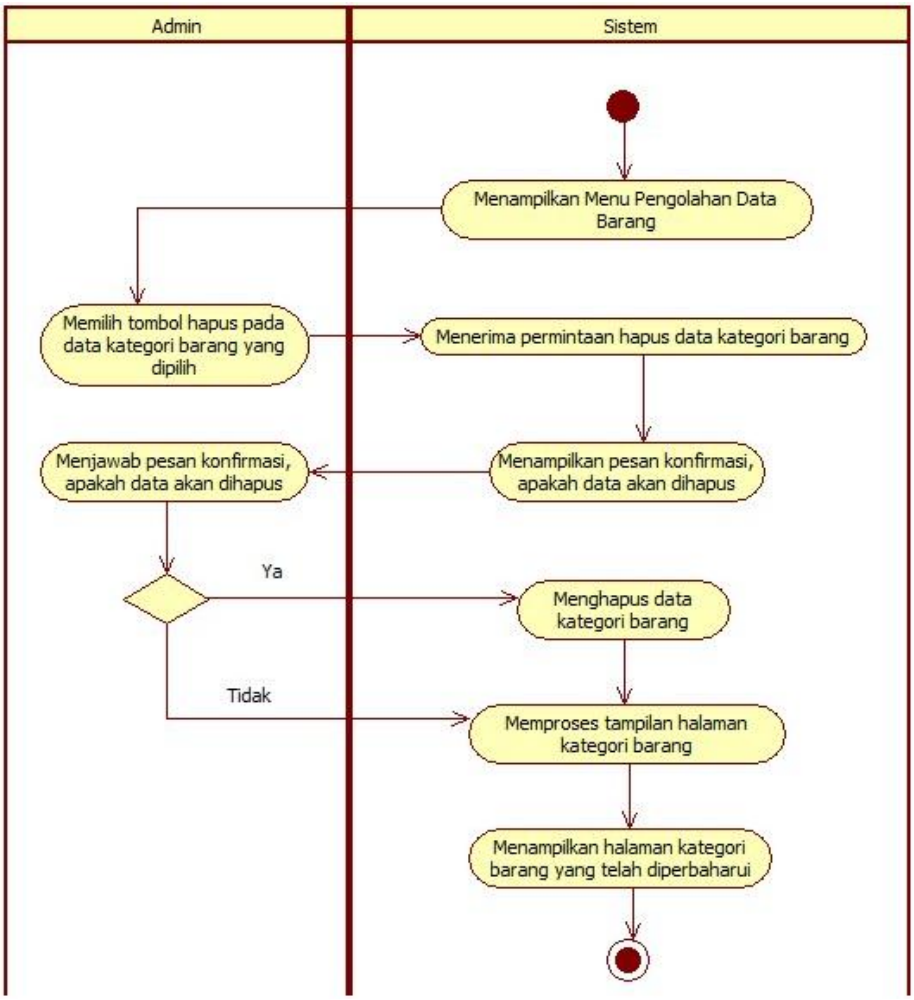
Interaksi antara aktor pengguna dengan *use case* ubah kategori barang dijelaskan dalam *activity diagram* sebagai berikut :



Gambar 3.27 *Activity Diagram* Ubah Kategori Barang

3.1.4.5.13 *Activity Diagram* Hapus Kategori Barang

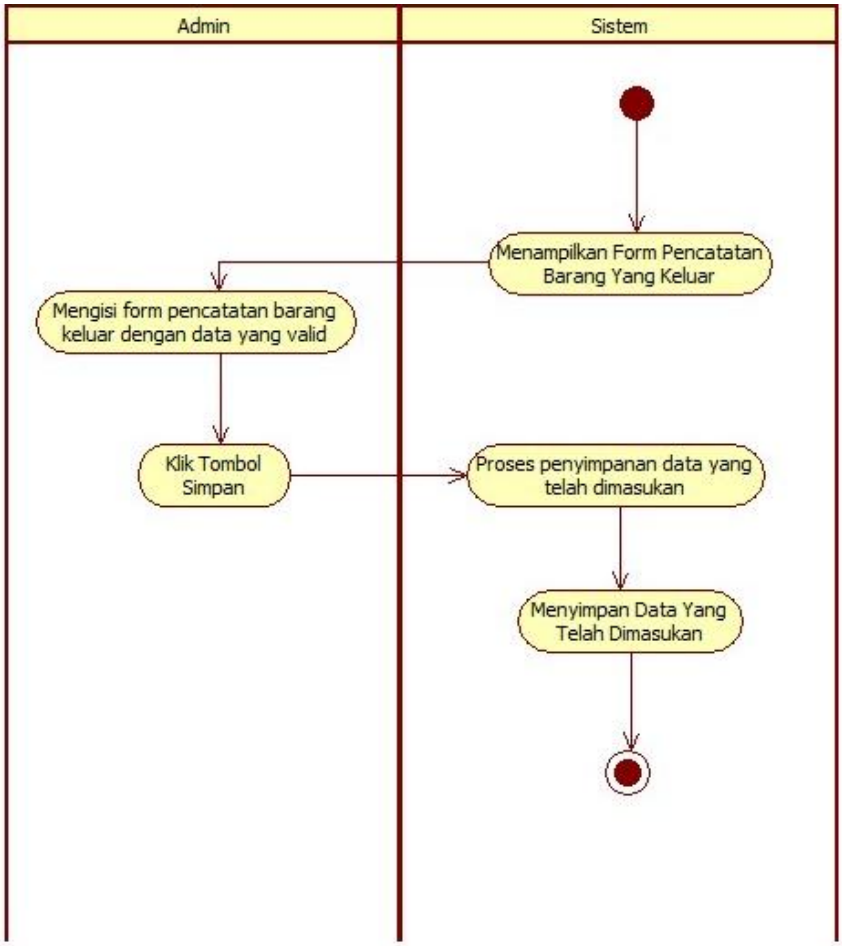
Interaksi antara aktor pengguna dengan *use case* hapus kategori barang dijelaskan dalam *activity diagram* sebagai berikut :



Gambar 3.28 *Activity Diagram* Hapus Kategori Barang

3.1.4.5.14 *Activity Diagram* Barang Keluar

Interaksi antara aktor pengguna dengan *use case* barang keluar dijelaskan dalam *activity diagram* sebagai berikut :



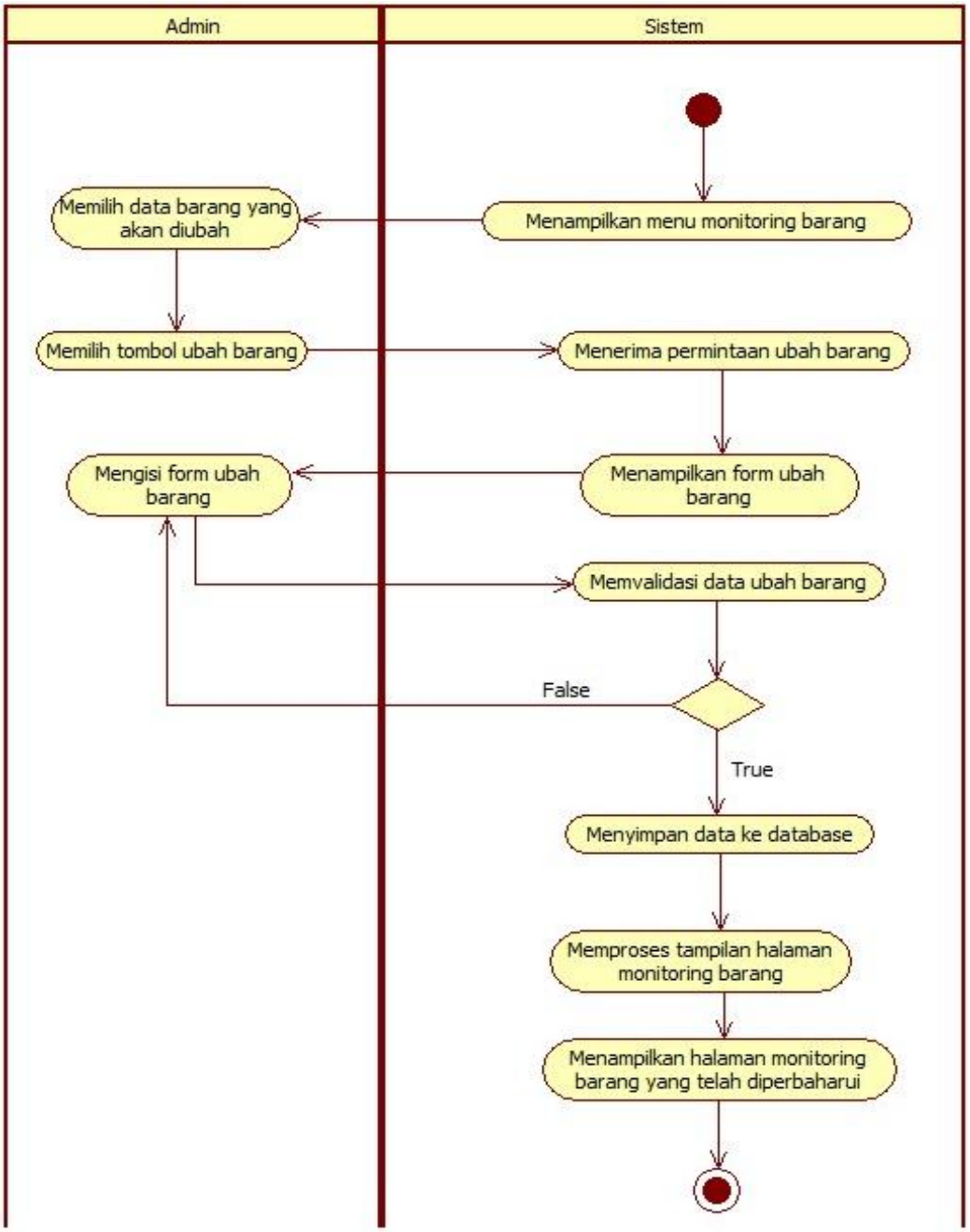
Gambar 3.29 *Activity Diagram* Barang Keluar

3.1.4.5.15 *Activity Diagram* Monitoring Barang

Interaksi antara aktor pengguna dengan *use case* monitoring barang dijelaskan dalam *activity diagram* tambah data barang, ubah data barang, hapus data barang dan lihat data barang. Berikut ini penjelasan dari masing-masing *activity diagram* tersebut:

3.1.4.5.16 *Activity Diagram* Ubah Data Barang

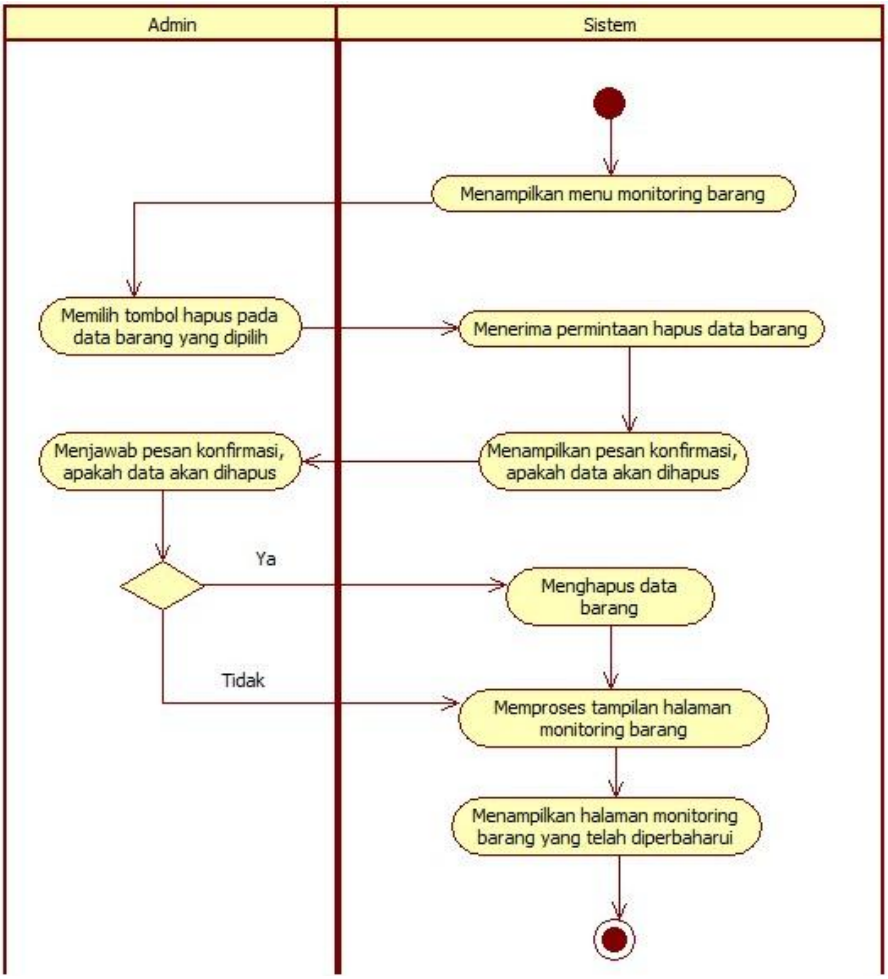
Interaksi antara aktor pengguna dengan *use case* ubah data barang dijelaskan dalam *activity diagram* sebagai berikut :



Gambar 3.30 *Activity Diagram* Ubah Data Barang

3.1.4.5.17 *Activity Diagram* Hapus Data Barang

Interaksi antara aktor pengguna dengan *use case* hapus data barang dijelaskan dalam *activity diagram* sebagai berikut :



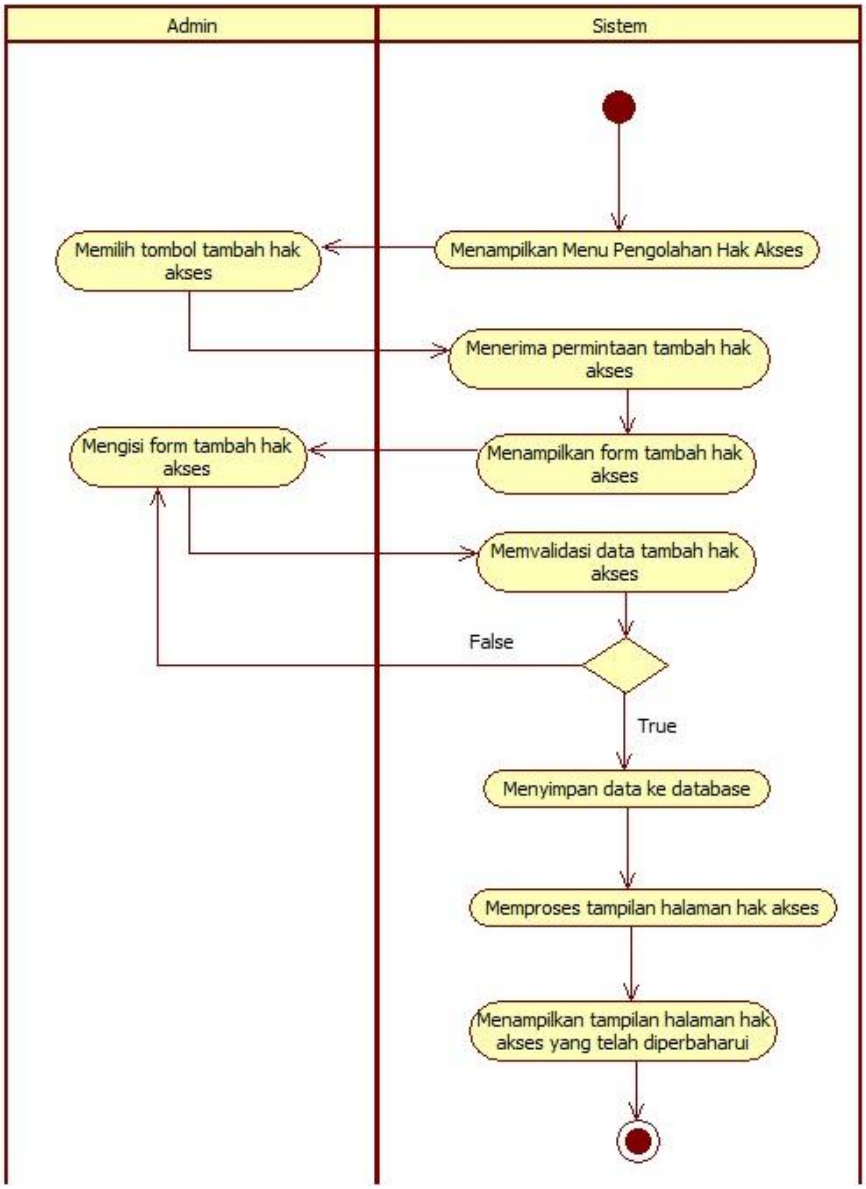
Gambar 3.31 *Activity Diagram* Hapus Data Barang

3.1.4.5.18 *Activity Diagram* Pengolahan Hak Akses

Interaksi antara aktor pengguna dengan *use case* pengolahan hak akses dijelaskan dalam *activity diagram* tambah hak akses, ubah hak akses dan hapus hak akses. Berikut ini penjelasan dari masing-masing *activity diagram* tersebut:

3.1.4.5.19 *Activity Diagram* Tambah Hak Akses

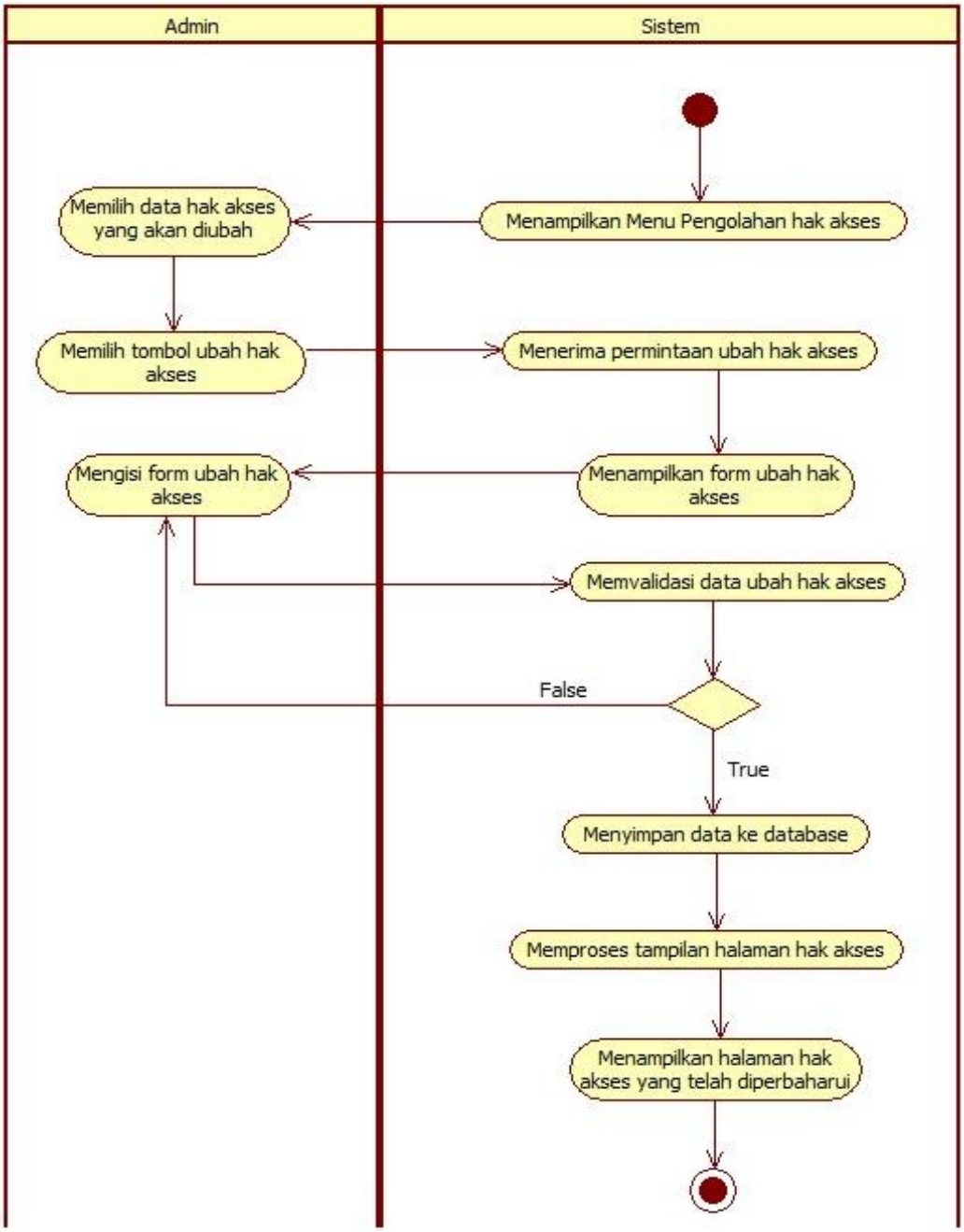
Interaksi antara aktor pengguna dengan *use case* tambah hak akses dijelaskan dalam *activity diagram* sebagai berikut :



Gambar 3.32 *Activity Diagram* Tambah Hak Akses

3.1.4.5.20 *Activity Diagram* Ubah Hak Akses

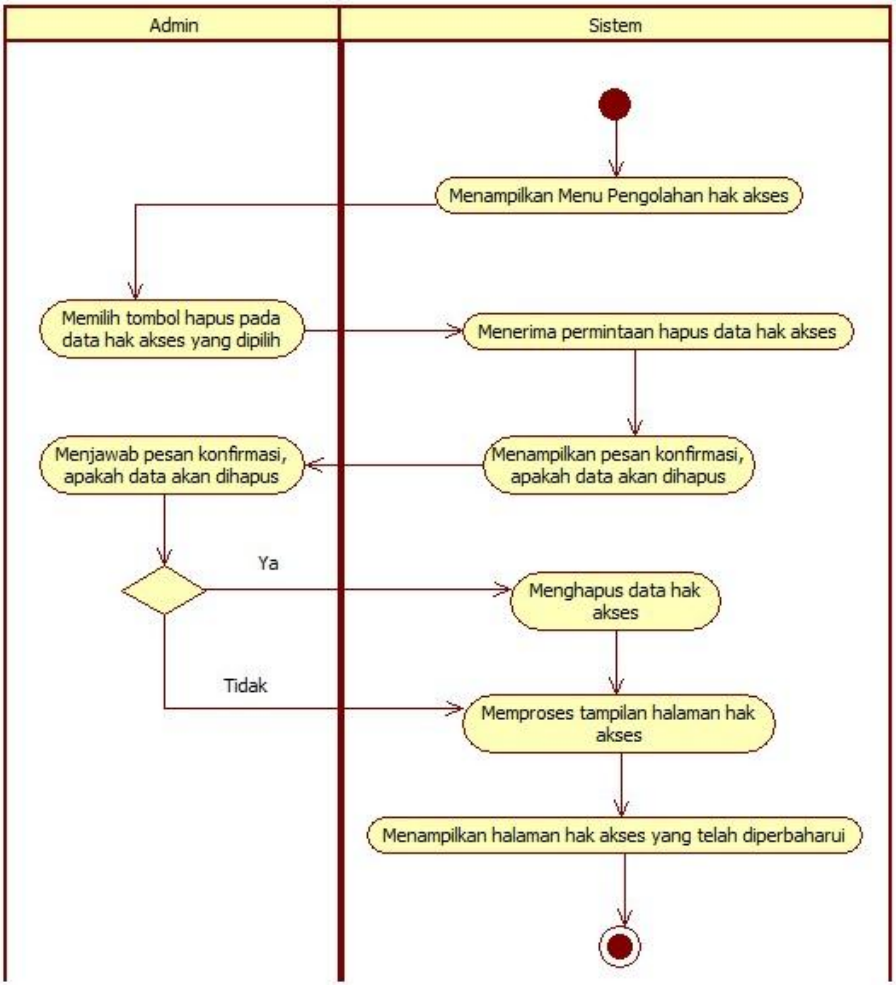
Interaksi antara aktor pengguna dengan *use case* ubah hak akses dijelaskan dalam *activity diagram* sebagai berikut :



Gambar 3.33 *Activity Diagram* Ubah Hak Akses

3.1.4.5.21 *Activity Diagram* Hapus Hak Akses

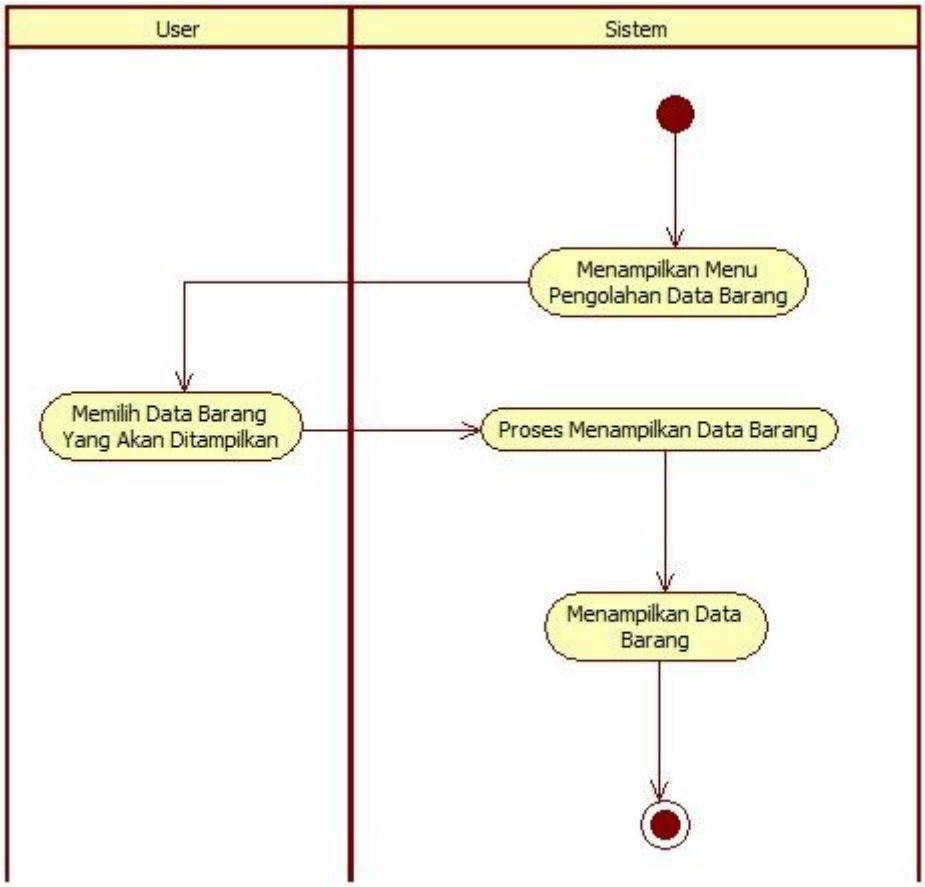
Interaksi antara aktor pengguna dengan *use case* hapus hak akses dijelaskan dalam *activity diagram* sebagai berikut :



Gambar 3.34 *Activity Diagram* Hapus Hak Akses

3.1.4.5.22 *Activity Diagram* Lihat Data Barang

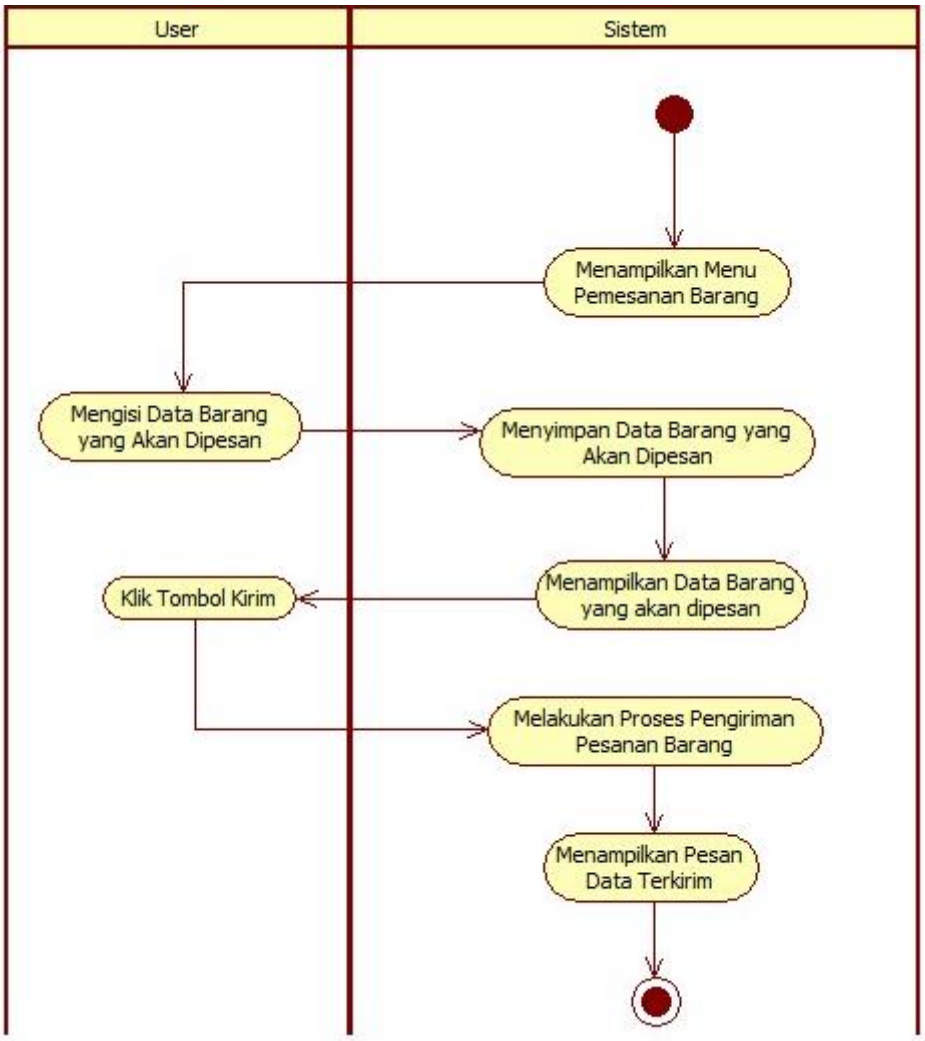
Interaksi antara aktor pengguna dengan *use case* lihat data barang dijelaskan dalam *activity diagram* sebagai berikut :



Gambar 3.35 *Activity Diagram* Lihat Data Barang

3.1.4.5.23 *Activity Diagram* Pemesanan Barang

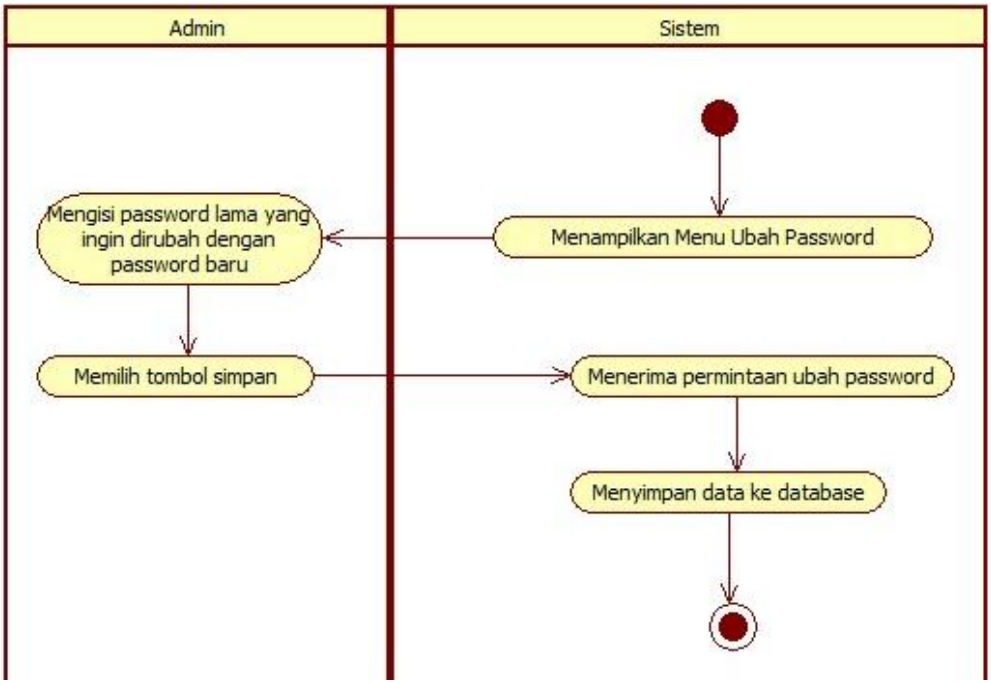
Interaksi antara aktor pengguna dengan *use case* pemesanan barang dijelaskan dalam *activity diagram* sebagai berikut :



Gambar 3.36 *Activity Diagram* Pemesanan Barang

3.1.4.5.24 *Activity Diagram* Ubah Password

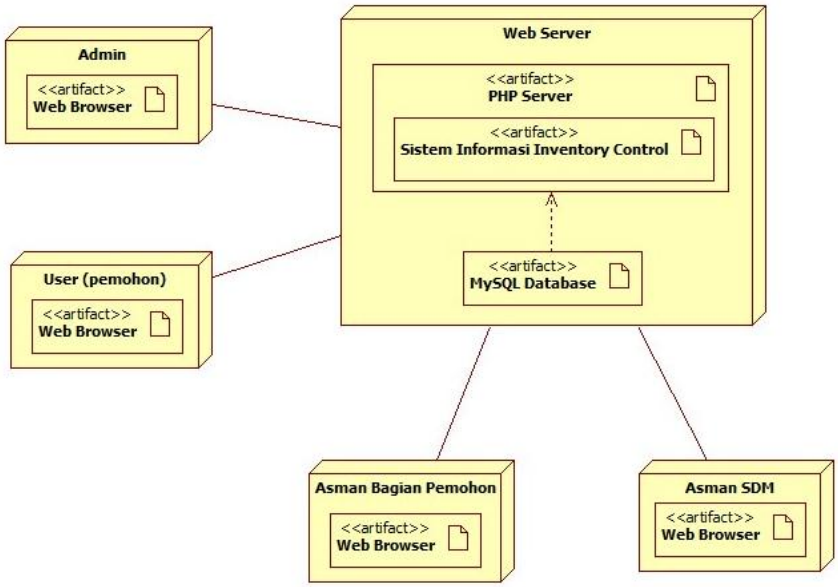
Interaksi antara aktor pengguna dengan *use case* ubah password dijelaskan dalam *activity diagram* sebagai berikut :



Gambar 3.37 *Activity Diagram* Ubah Password

3.1.4.6 Deployment Diagram

Deployment diagram adalah *diagram* yang mendeskripsikan arsitektur fisik dari *hardware* dan *software* didalam sistem. *Deployment diagram* pada sistem informasi *inventory control* dapat dilihat pada Gambar 3.38.



Gambar 3.38 Deployment *Diagram*

3.2 Perancangan Sistem

Perancangan merupakan penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh. Tahapan ini meliputi mengonfigurasi komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem. Adapun perancangan sistem informasi *inventory control* yang dibuat dijelaskan sebagai berikut.

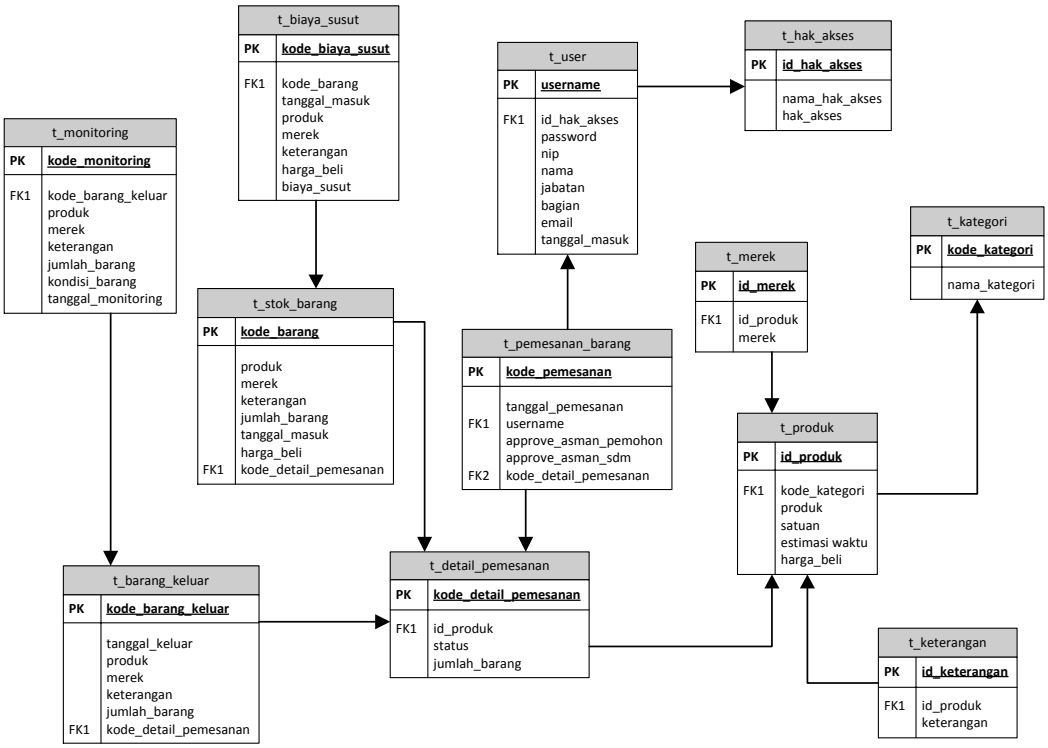
3.2.1 Perancangan Data

Perancangan data merupakan tahapan untuk memetakan model konseptual ke model basis data yang akan dipakai. Perancangan data terbagi menjadi skema relasi, *diagram* skema, dan perancangan struktur tabel. Berikut penjelasan detail perancangan data tersebut :

3.2.1.1 Skema Relasi

1. t_stok_barang (kode_barang, kode_detail_pemesanan, tanggal_masuk, produk, merk, harga_beli, stok, keterangan, jumlah_barang)
2. t_kategori (kode_kategori, nama_kategori)
3. t_barang_keluar (kode_barang_keluar, kode_detail_pemesanan, tanggal_keluar, produk, merk, jumlah_barang, keterangan)
4. t_user (username, id_hak_akses, password, nip, nama, jabatan, bagian, email, tanggal_masuk)
5. t_hak_akses (id_hak_akses, nama_hak_akses, hak_akses)
6. t_pemesanan_barang (kode_pemesanan, tanggal_pemesanan, username, approve_asman_pemohon, approve_asman_sdm, kode_detail_pemesanan)
7. t_detail_pemesanan (kode_detail_pemesanan, id_produk, status, jumlah_barang)
8. t_monitoring (kode_monitoring, kode_barang_keluar, produk, merek, kondisi_barang, keterangan, jumlah_barang, tanggal_monitoring)
9. t_biaya_susut (kode_biaya_susut, tanggal_masuk, kode_barang, produk, merek, keterangan, harga_beli, biaya_susut)
10. t_produk (id_produk, kode_kategori, produk, satuan, estimasi_waktu, harga_beli)
11. t_merek (id_merek, id_produk, merek)
12. t_keterangan (id_keterangan, id_produk, keterangan)

3.2.1.2 *Diagram* Relasi



Gambar 3.39 *Diagram* Relasi

3.2.1.3 Struktur File

Struktur tabel menggambarkan detail tabel yang berisi field, tipe data, panjang data, dan keterangan lainnya. Adapun tabel-tabel yang digunakan dalam database sistem informasi *inventory control* ini adalah sebagai berikut:

1. Tabel t_stok_barang

Tabel barang digunakan untuk menyimpan data barang yang memudahkan dalam proses pengolahan data barang. Berikut struktur dari tabel t_barang :

Tabel 3.58 Tabel t_stok_barang

Nama Field	Tipe Data	Panjang	Keterangan
kode_barang	Varchar	10	*Primary Key*
kode_detail_pemesanan	Varchar	10	*Foreign Key references* t_pemesanan(kode_ pemesanan)
tanggal_masuk	Date		
produk	Varchar	30	
merk	Varchar	30	
harga_beli	Int	10	
stok	Int	10	
keterangan	Varchar	40	
jumlah_barang	Int	5	

2. Tabel t_kategori

Tabel kategori digunakan untuk menyimpan data kategori barang yang ada di dalam gudang. Berikut struktur dari tabel t_kategori :

Tabel 3.59 Tabel t_kategori

Nama Field	Tipe Data	Panjang	Keterangan
kode_ kategori	Varchar	3	*Primary Key*
nama _kategori	Varchar	15	

3. Tabel t_barang_keluar

Tabel barang keluar digunakan untuk menyimpan data barang yang akan keluar dari gudang, memudahkan dalam proses pencatatan barang keluar. Berikut struktur dari tabel t_barang_keluar :

Tabel 3.60 Tabel t_barang_keluar

Nama Field	Tipe Data	Panjang	Keterangan
kode_barang_keluar	Varchar	10	*Primary Key*
kode_detail_pemesanan	Varchar	10	*Foreign Key references*

			t_stok_barang(kode_barang)
tanggal_keluar	Date		
produk	Varchar	30	
merk	Varchar	30	
jumlah_barang	Int	5	
keterangan	Varchar	40	

4. Tabel t_user

Tabel user digunakan untuk menyimpan data user. Berikut struktur dari tabel t_user :

Tabel 3.61 Tabel t_user

Nama Field	Tipe Data	Panjang	Keterangan
username	Varchar	45	*Primary Key*
id_ hak_akses	Tinyint	5	*Foreign Key references* t_hak akses(id_hak_akses)
password	Varchar	20	
nip	Varchar	10	
nama	Varchar	45	
jabatan	Varchar	25	
bagian	Varchar	20	
email	Varchar	45	
tanggal_masuk	Date		

5. Tabel t_hak_akses

Tabel hak akses digunakan untuk menyimpan data hak akses yang ada di dalam sistem informasi *inventory control*. Berikut struktur dari tabel t_hak_akses :

Tabel 3.62 Tabel t_hak_akses

Nama Field	Tipe Data	Panjang	Keterangan
id_ hak_akses	Varchar	3	*Primary Key*
nama _ hak_akses	Varchar	15	
Hak_akses	Varchar	100	

6. Tabel t_pemesanan_barang

Tabel pemesanan barang digunakan untuk menyimpan data pesanan barang yang dipesan oleh user. Berikut struktur dari tabel t_pemesanan_barang :

Tabel 3.63 Tabel t_pemesanan_barang

Nama Field	Tipe Data	Panjang	Keterangan
kode_pemesanan	Varchar	9	*Primary Key*
username	Varchar	45	
appprove_asman_pemohon	Tinyint	1	
appprove_asman_SDM	Tinyint	1	
tanggal_pemesanan	Date		
kode_detail_pemesanan	Varchar	9	*Foreign Key references* t_detail_pemesanan (kode_detail_pemesanan)

7. Tabel t_detail_pemesanan

Tabel detail pemesanan digunakan untuk menyimpan detail data pesanan barang yang dipesan oleh user. Berikut struktur dari tabel t_pemesanan_barang :

Tabel 3.64 Tabel t_detail_pemesanan

Nama Field	Tipe Data	Panjang	Keterangan
kode_detail_pemesanan	Int	9	*Primary Key*
id_produk	Varchar	3	*Foreign Key references* t_produk (id_produk)
jumlah	Int	10	
satuan	Varchar	15	
username	Varchar	45	
tanggal_pemesanan	Date		
keterangan	Text		

8. Tabel t_monitoring

Tabel monitoring digunakan untuk menyimpan data barang yang ada di tiap bagian/unit perusahaan. Berikut struktur dari tabel t_monitoring :

Tabel 3.65 Tabel t_monitoring

Nama Field	Tipe Data	Panjang	Keterangan
kode_monitoring	Varchar	10	*Primary Key*
kode_ barang_keluar	Varchar	10	*Foreign Key references t*_barang_keluar (kode_barang_keluar)
produk	Varchar	30	
merek	Varchar	30	
keterangan	Varchar	40	
jumlah_barang	Int	3	
kondisi_barang	Varchar	40	
tanggal_monitoring	Date		

9. Tabel t_biaya_susut

Tabel biaya susut digunakan untuk menyimpan data barang yang sudah melalui proses perhitungan biaya susut. Berikut struktur dari tabel t_biaya_susut :

Tabel 3.66 Tabel t_biaya_susut

Nama Field	Tipe Data	Panjang	Keterangan
kode_biaya_susut	Varchar	9	*Primary Key*
kode_ barang	Varchar	10	*Foreign Key references* t_stok_barang (kode_barang)
tanggal_masuk	Date		
produk	Varchar	30	
merek	Varchar	30	
keterangan	Varchar	40	
harga_beli	Int	10	
biaya_susut	Int	10	

10. Tabel t_produk

Tabel produk digunakan untuk menyimpan data produk. Berikut struktur dari tabel t_produk :

Tabel 3.67 Tabel t_produk

Nama Field	Tipe Data	Panjang	Keterangan
id_produk	Int	3	*Primary Key*
kode_ kategori	Varchar	3	*Foreign Key references* kategori (kode_kategori)
produk	Varchar	30	
satuan	Varchar	10	
estimasi_waktu	Int	2	
harga_beli	Int	10	

11. Tabel t_merek

Tabel merek digunakan untuk menyimpan data merek. Berikut struktur dari tabel t_merek :

Tabel 3.68 Tabel t_merek

Nama Field	Tipe Data	Panjang	Keterangan
id_merek	Int	3	*Primary Key*
id_produk	Int	3	*Foreign Key references* t_produk (id_produk)
merek	Varchar	30	

12. Tabel t_keterangan

Tabel keterangan digunakan untuk menyimpan data keterangan. Berikut struktur dari tabel t_ keterangan :

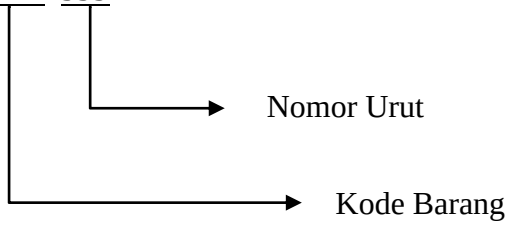
Tabel 3.69 Tabel t _keterangan

Nama Field	Tipe Data	Panjang	Keterangan
id_ keterangan	Int	3	*Primary Key*
id_produk	Int	3	*Foreign Key references* t_produk (id_produk)
keterangan	Varchar	40	

3.2.1.4 Perancangan Kode

1. **Kode Barang**

Format : AAA 999



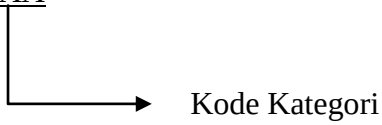
Contoh :

ATK 1, kode barang yang mempunyai kode kategori ATK dengan nomor urut 1

INV 2, kode barang yang mempunyai kode kategori INV dengan nomor urut 1

2. **Kode Kategori**

Format : AAA

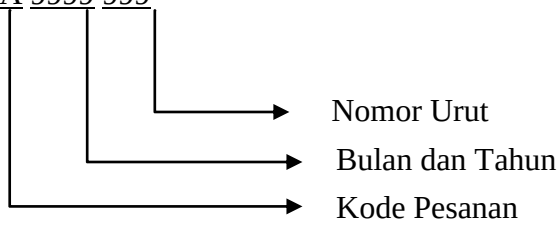


Contoh :

INV, kode kategori barang untuk barang tidak habis pakai

3. Kode Pesanan

Format : AA 9999 999

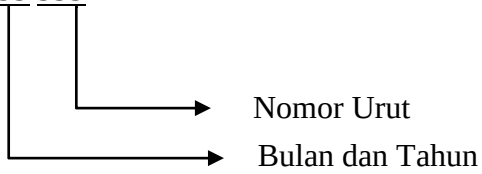


Contoh :

KP 0411 001, kode pesanan yang masuk pada bulan april 2011 dengan nomor urut 1

4. Kode Monitoring

Format : 9999 999

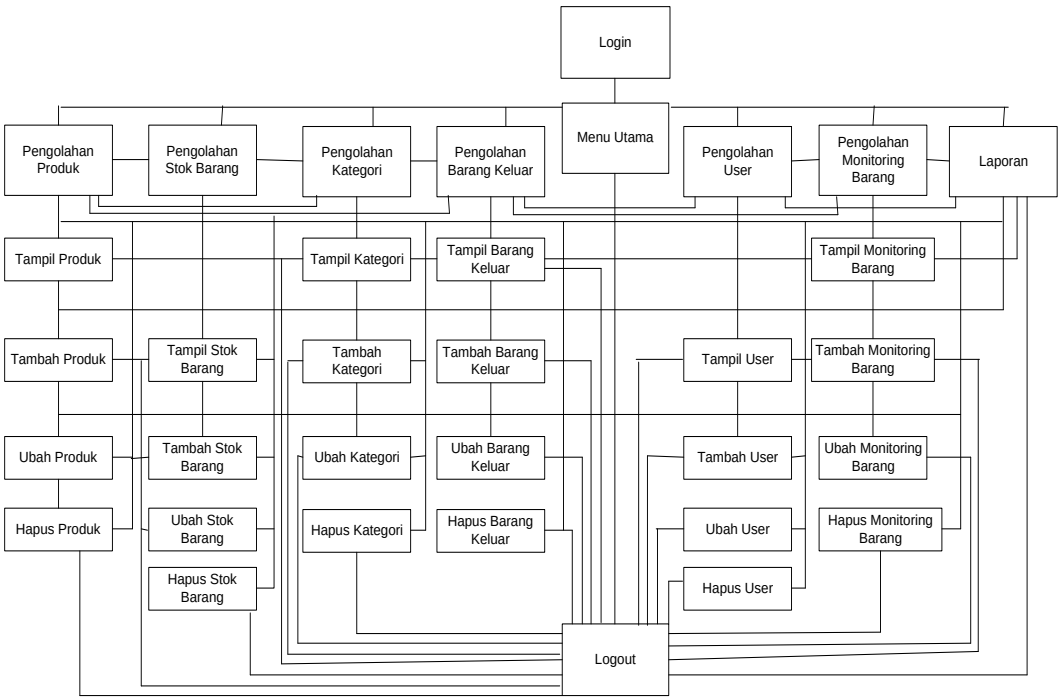


Contoh :

0411 001, kode monitoring yang dicatat pada bulan april 2011 dengan nomor urut 1

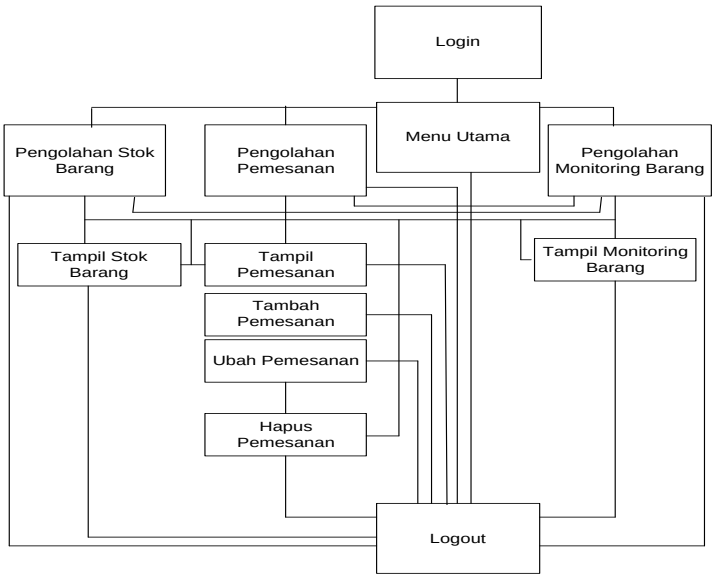
3.2.1.5 Perancangan Arsitektur

3.2.1.5.1 Struktur Menu Admin



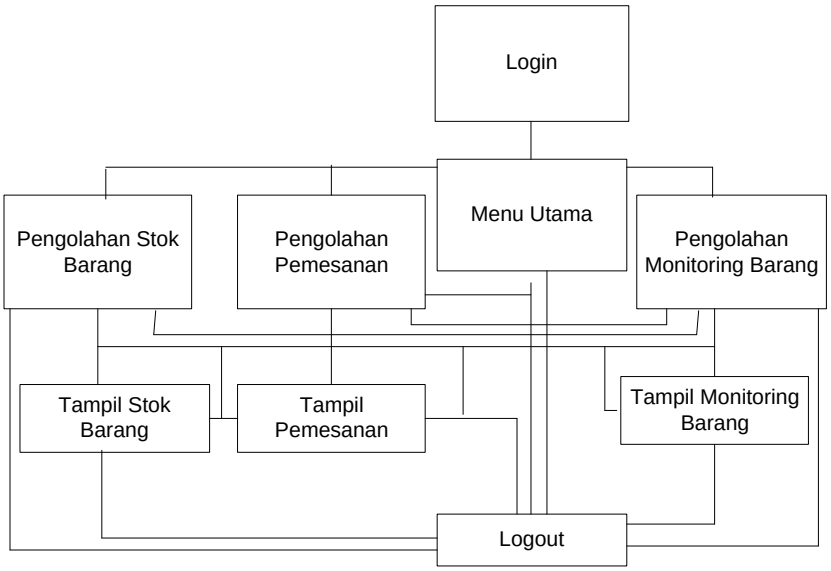
Gambar 3.40 Struktur Menu Admin

3.2.1.5.2 Struktur Menu User



Gambar 3.41 Struktur Menu User

3.2.1.5.3 Struktur Menu Asman

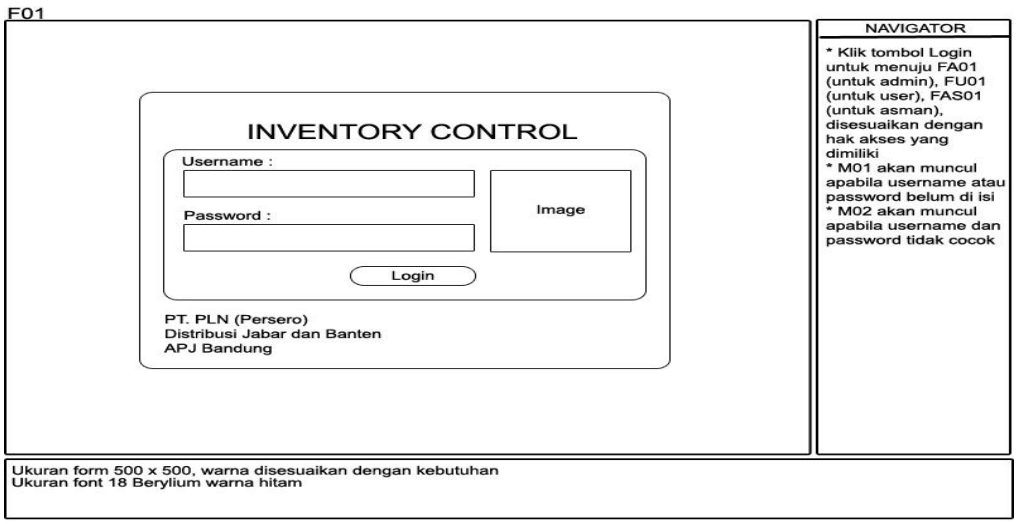


Gambar 3.42 Struktur Menu Asman

3.2.2 Perancangan Antarmuka

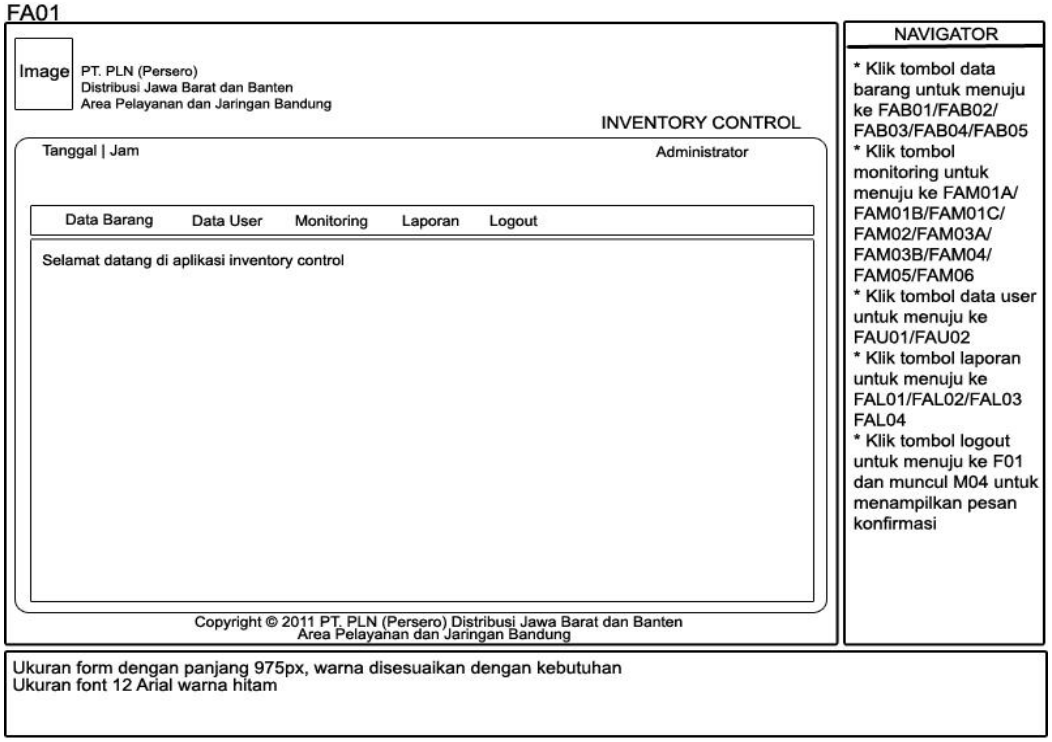
3.2.2.1 Perancangan Form

1. Form Login



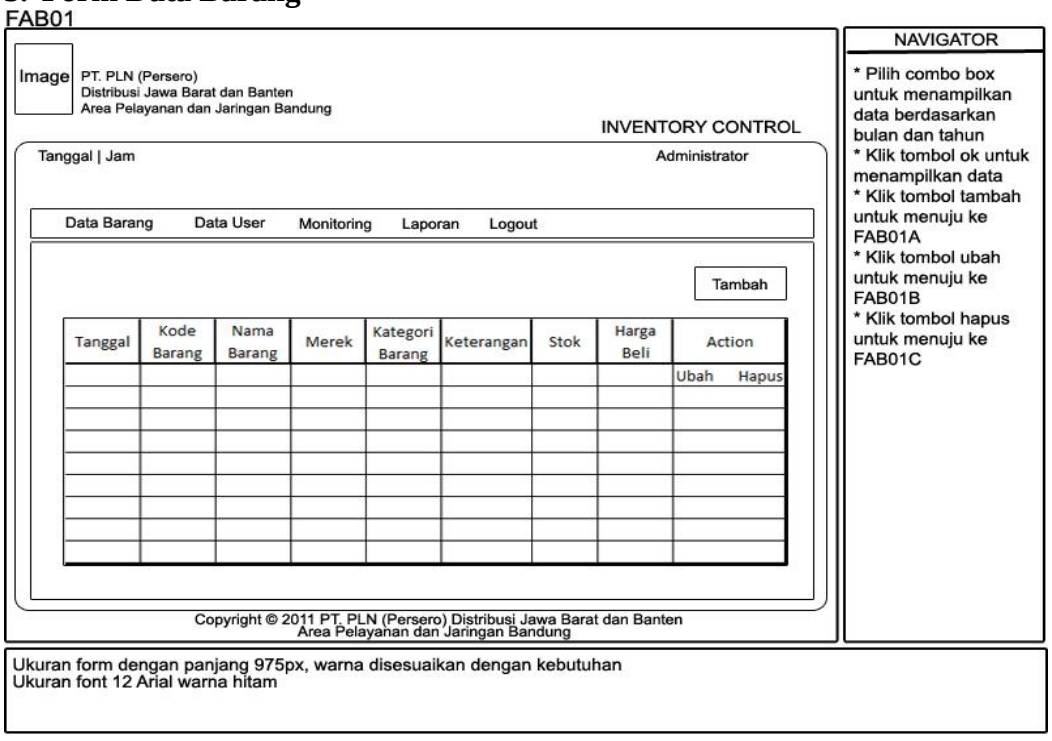
Gambar 3.43 Form Login

2. Form Menu Admin



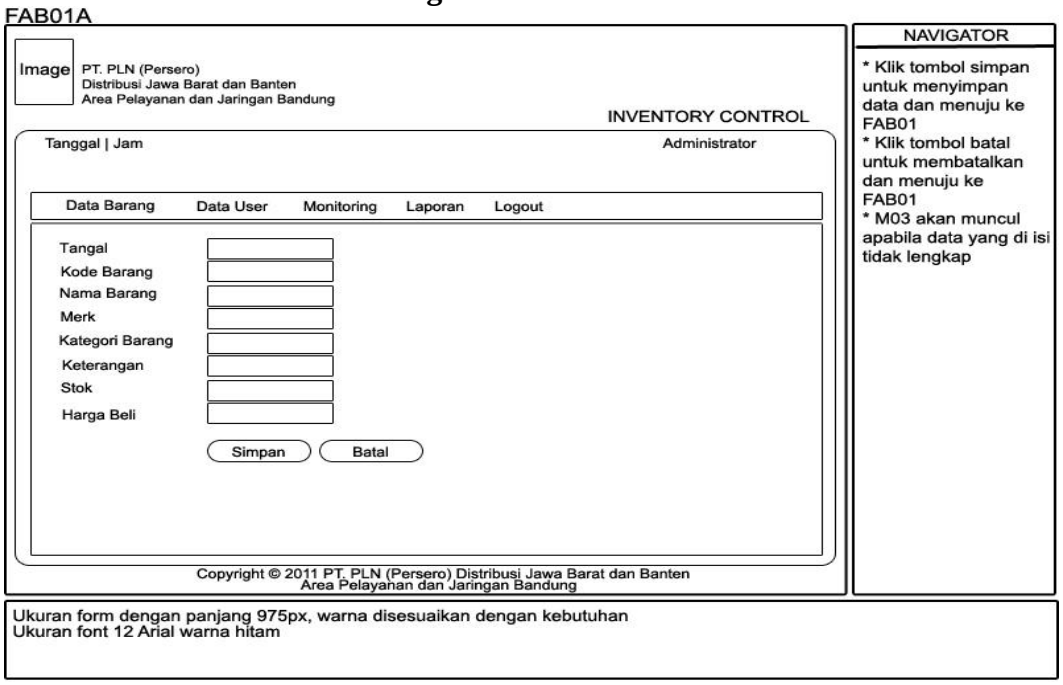
Gambar 3.44 Menu Admin

3. Form Data Barang



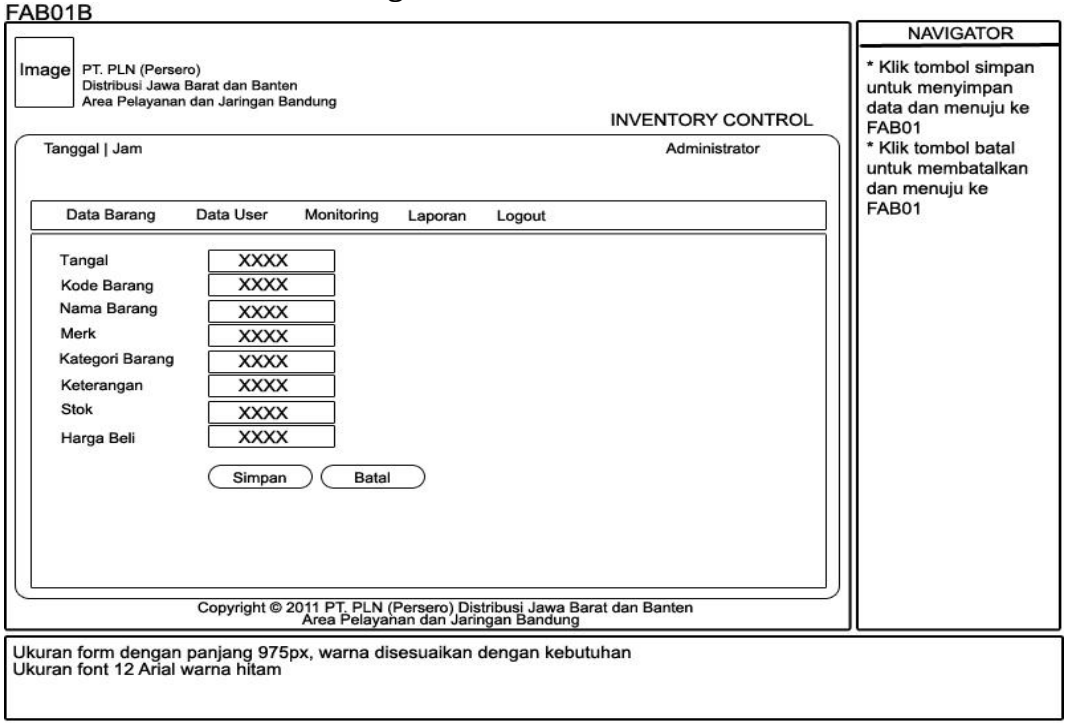
Gambar 3.45 Form Data Barang

4. Form Tambah Data Barang



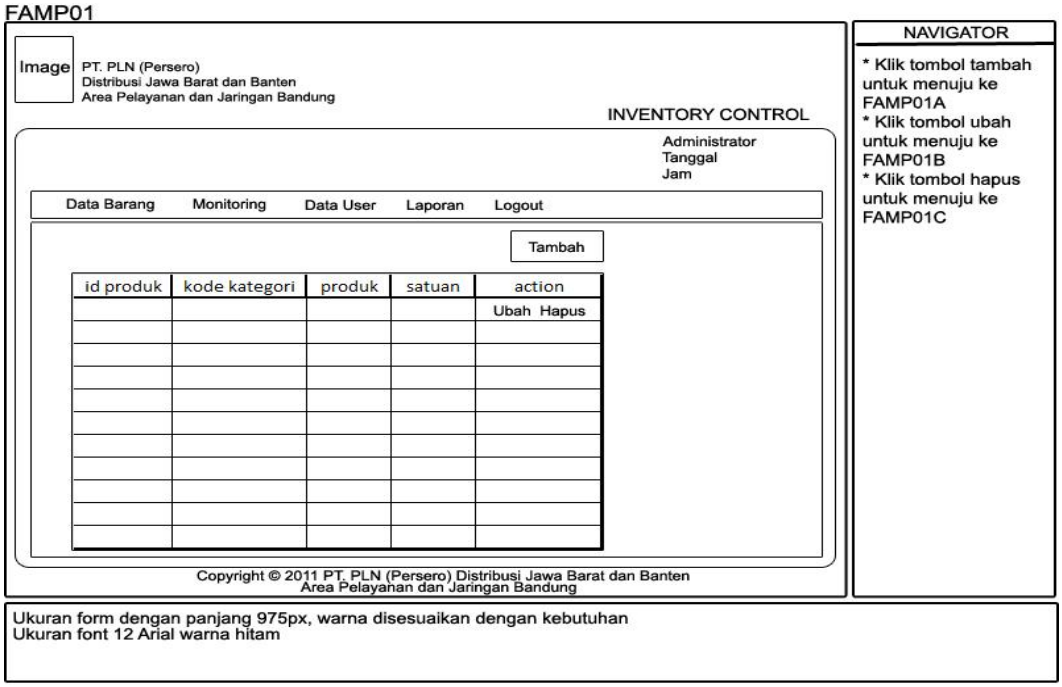
Gambar 3.46 Form Tambah Data Barang

5. Form Ubah Data Barang



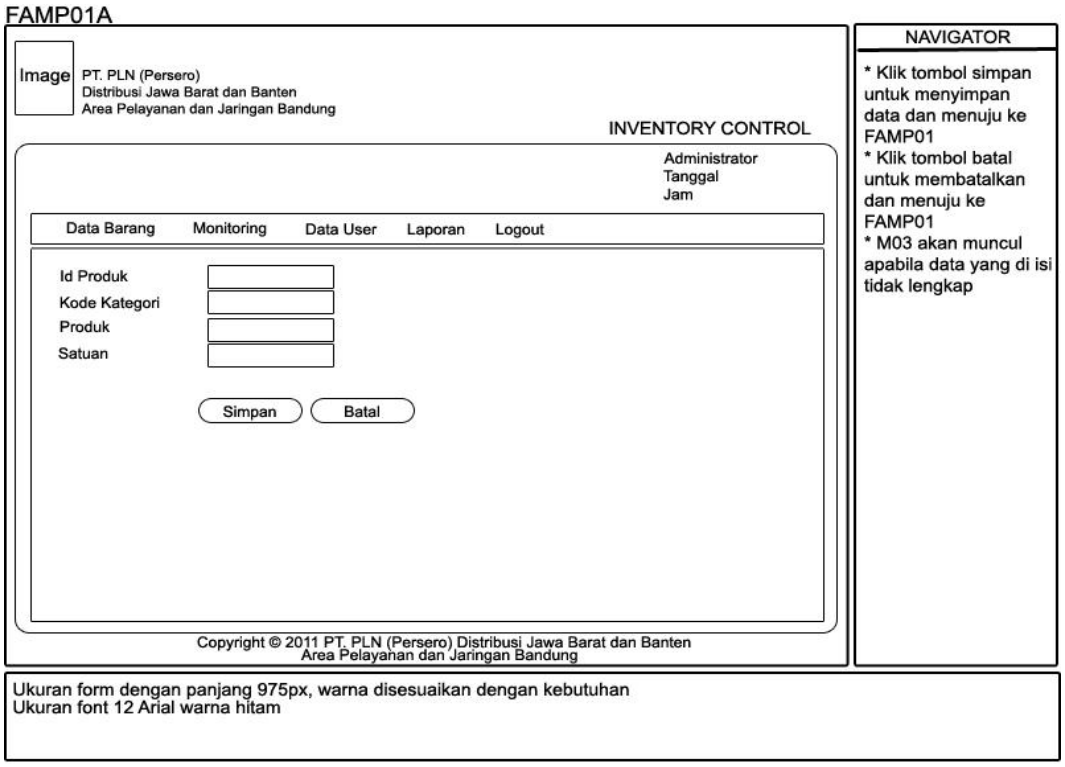
Gambar 3.47 Form Ubah Data Barang

6. Form Produk



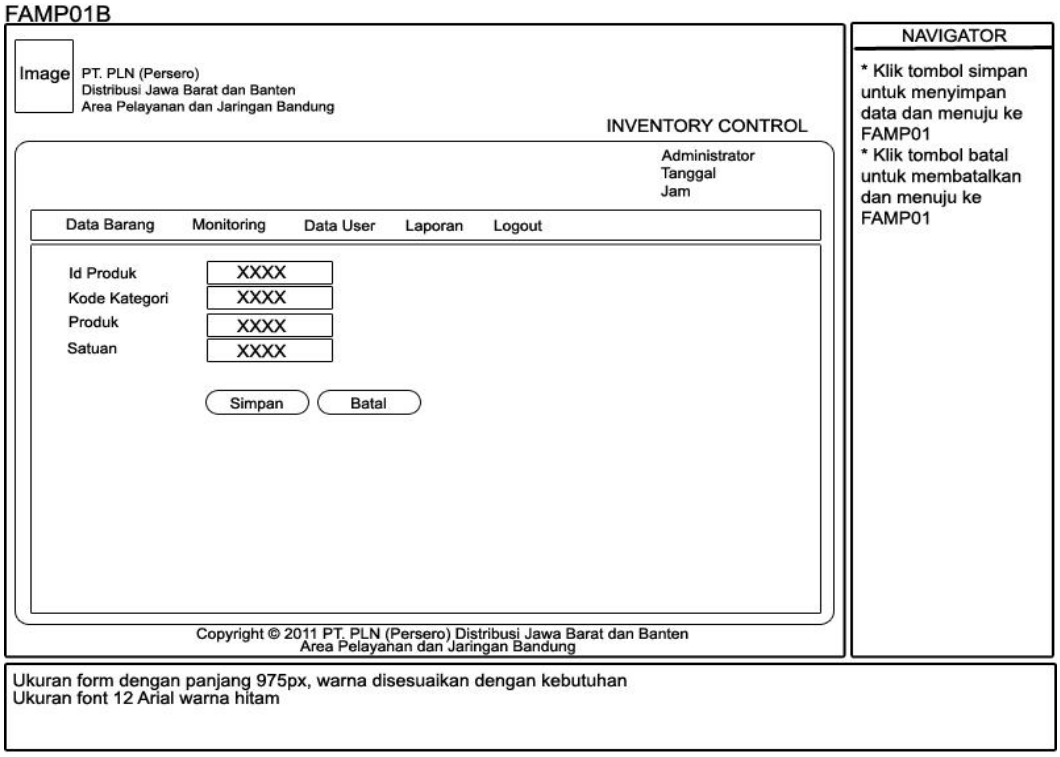
Gambar 3.48 Form Produk

7. Form Tambah Produk



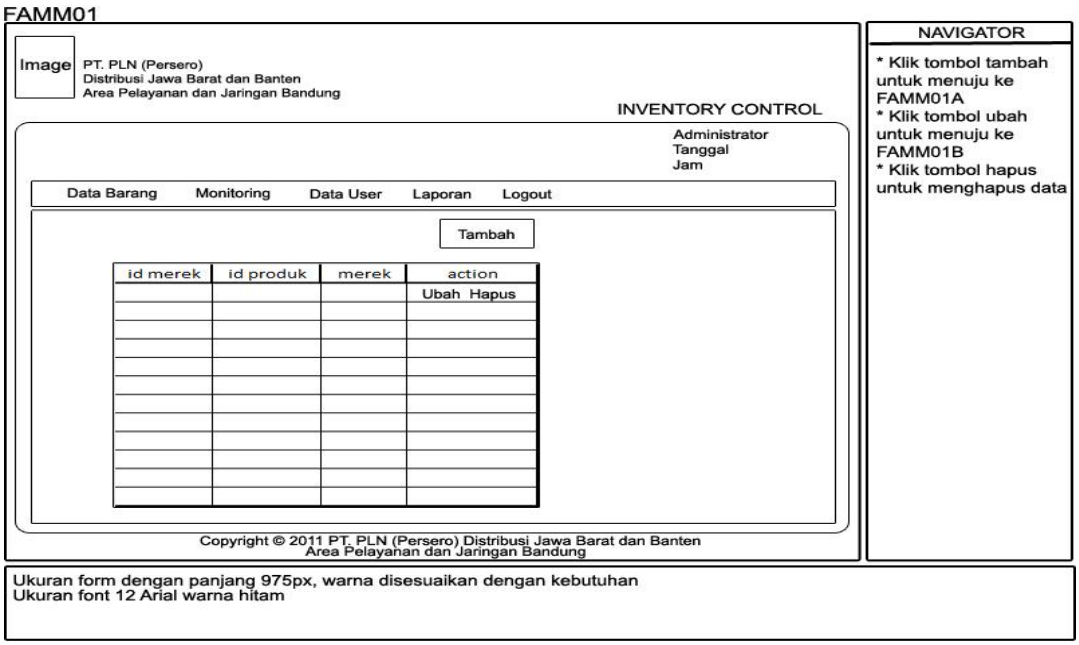
Gambar 3.49 Form Tambah Produk

8. Form Ubah Produk



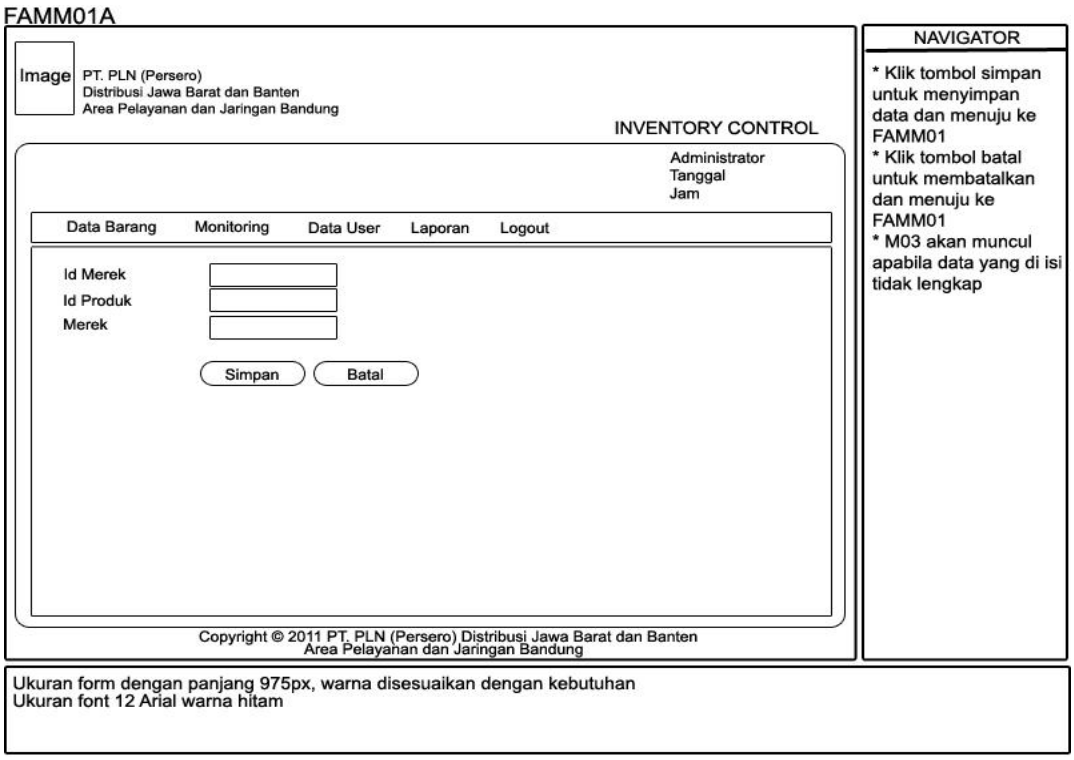
Gambar 3.50 Form Ubah Produk

9. Form Merek



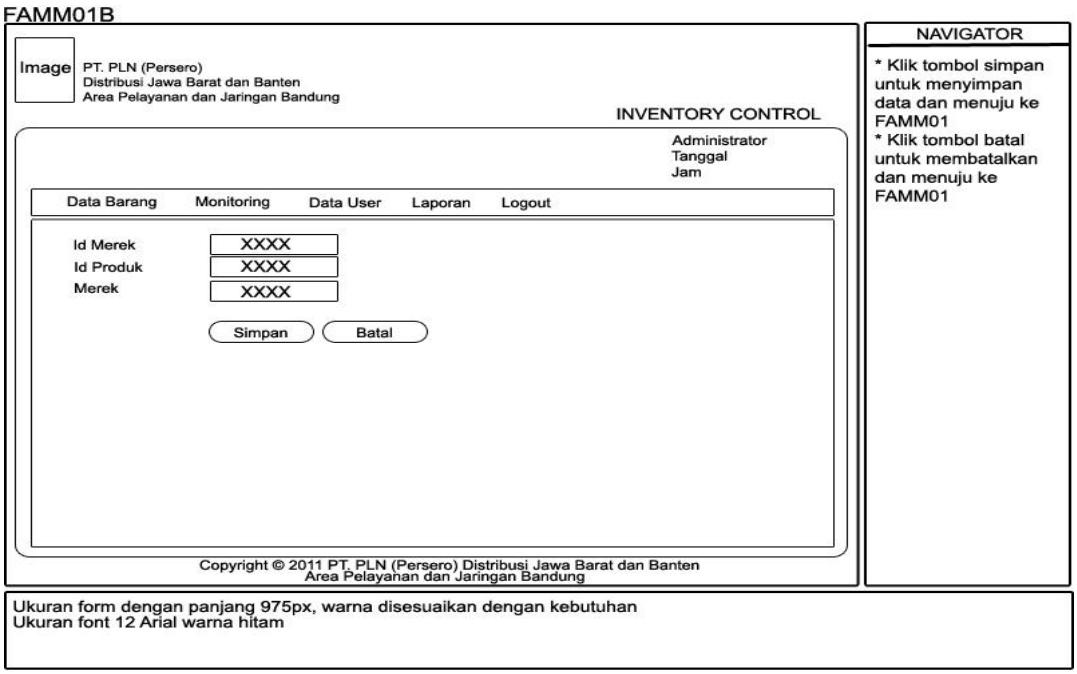
Gambar 3.51 Form Merek

10. Form Tambah Merek



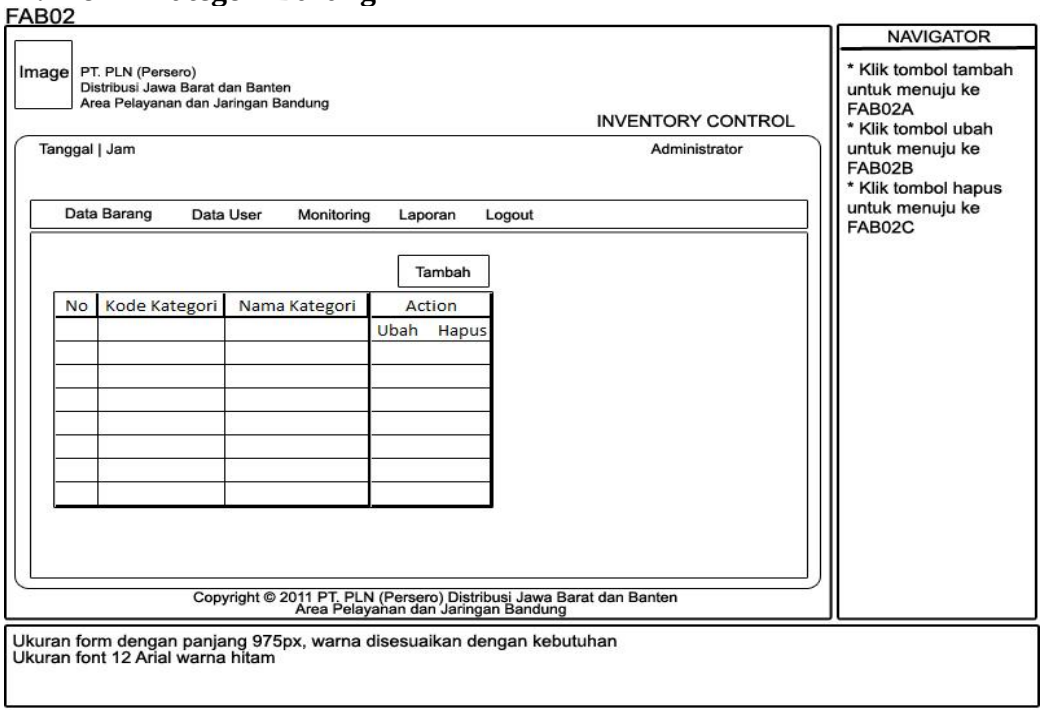
Gambar 3.52 Form Tambah Merek

11. Form Ubah Merek



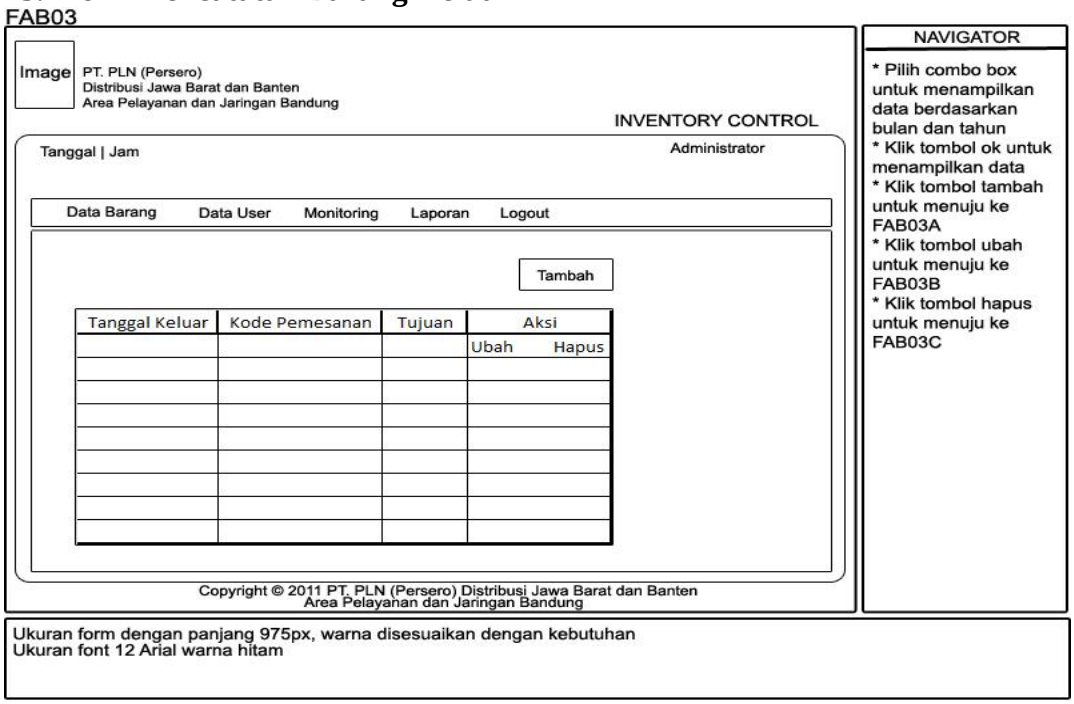
Gambar 3.53 Form Ubah Merek

12. Form Kategori Barang



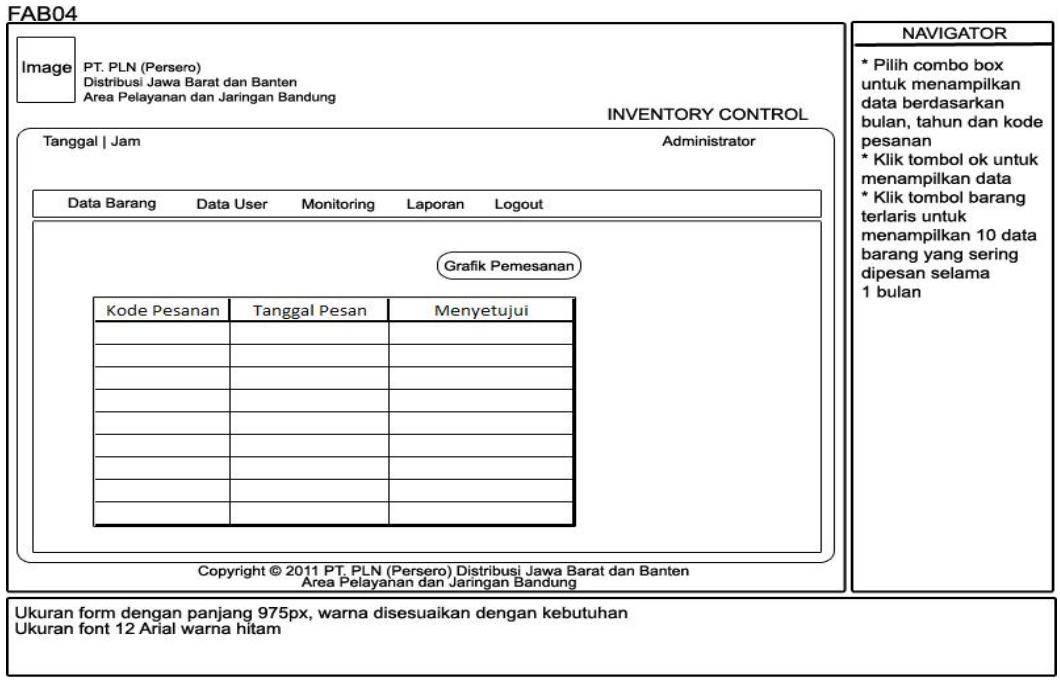
Gambar 3.54 Form Kategori Barang

13. Form Pencatatan Barang Keluar



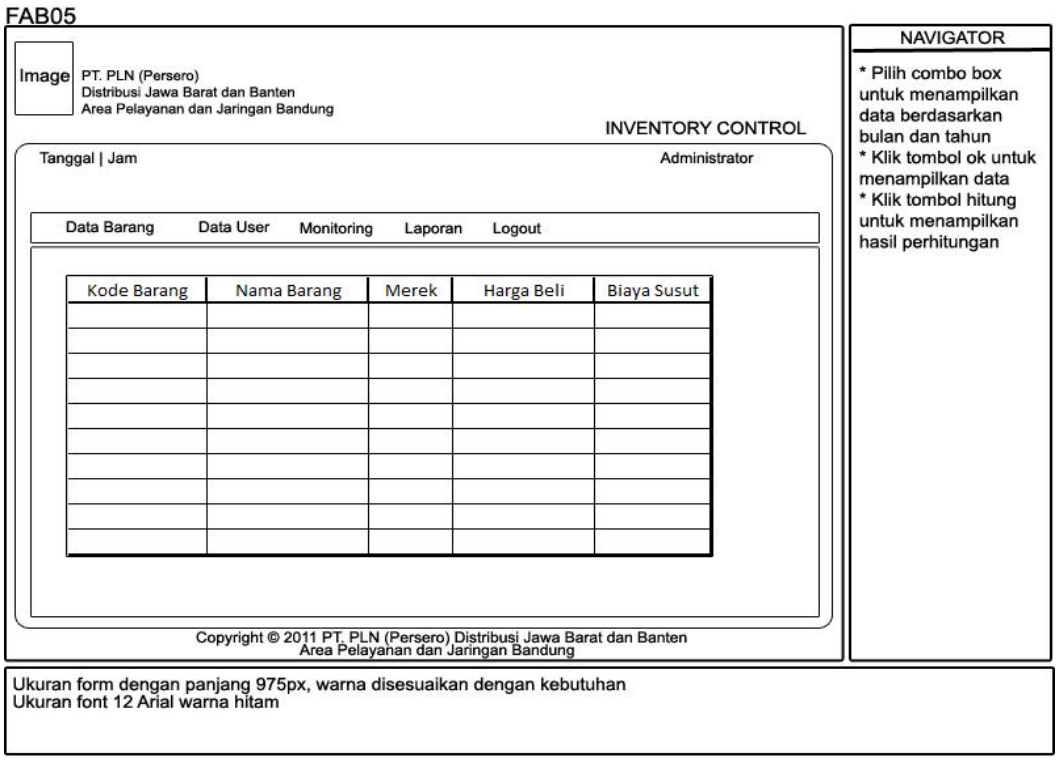
Gambar 3.55 Form Pencatatan Barang Keluar

14. Form Pemesanan Barang



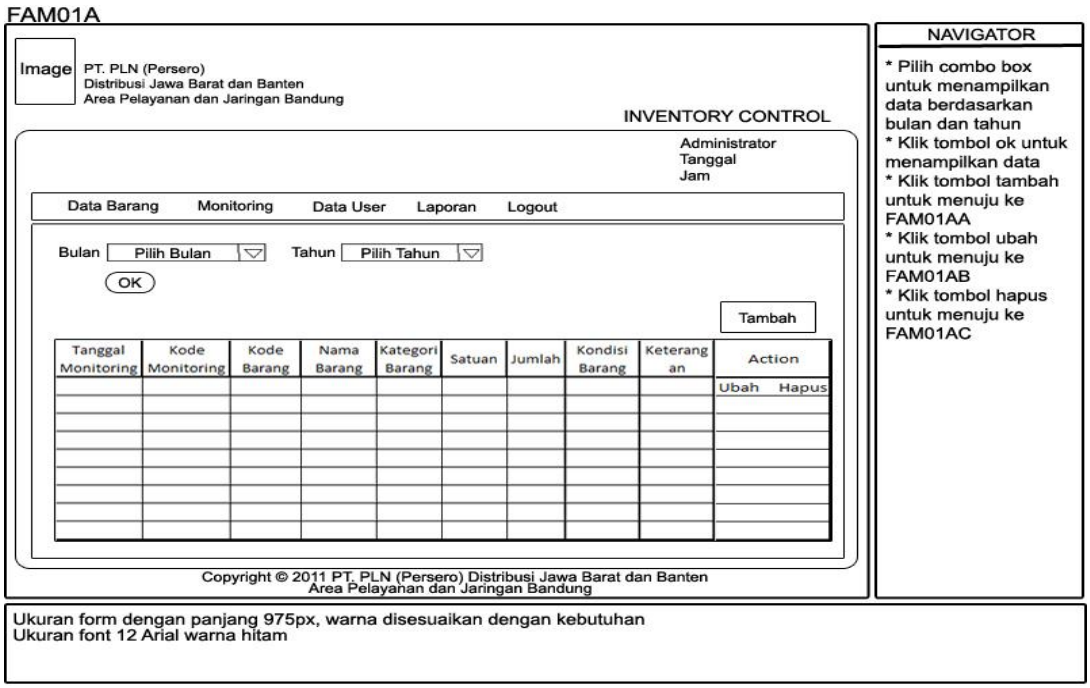
Gambar 3.56 Form Pemesanan Barang

15. Form Biaya Susut



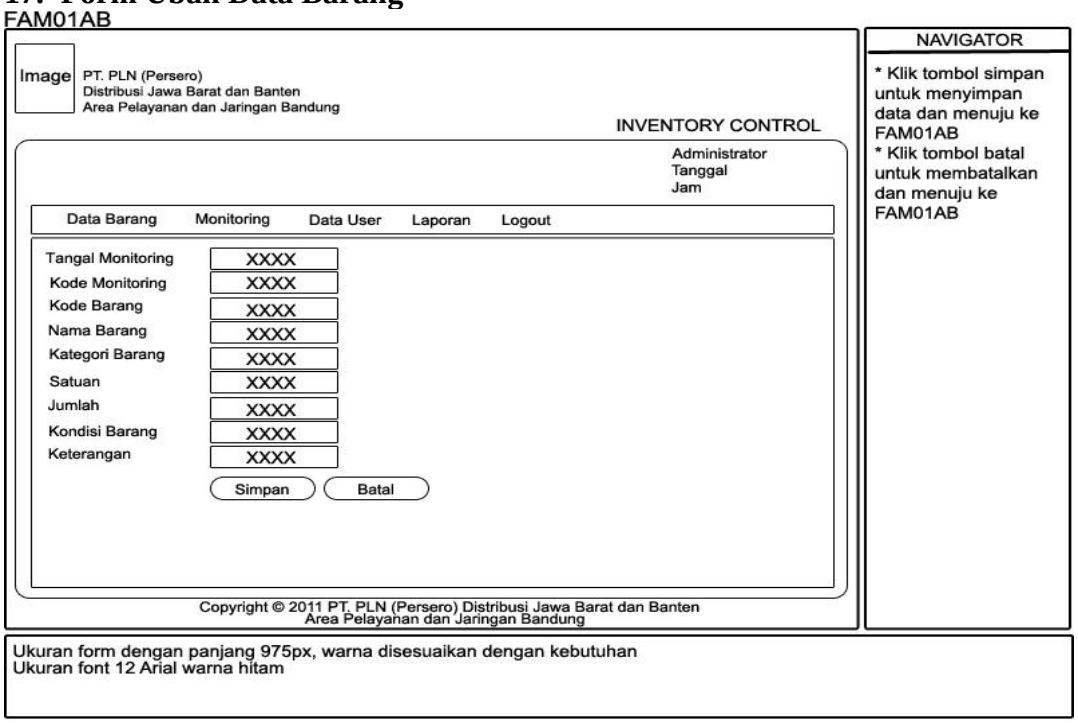
Gambar 3.57 Form Biaya Susut

16. Form Monitoring



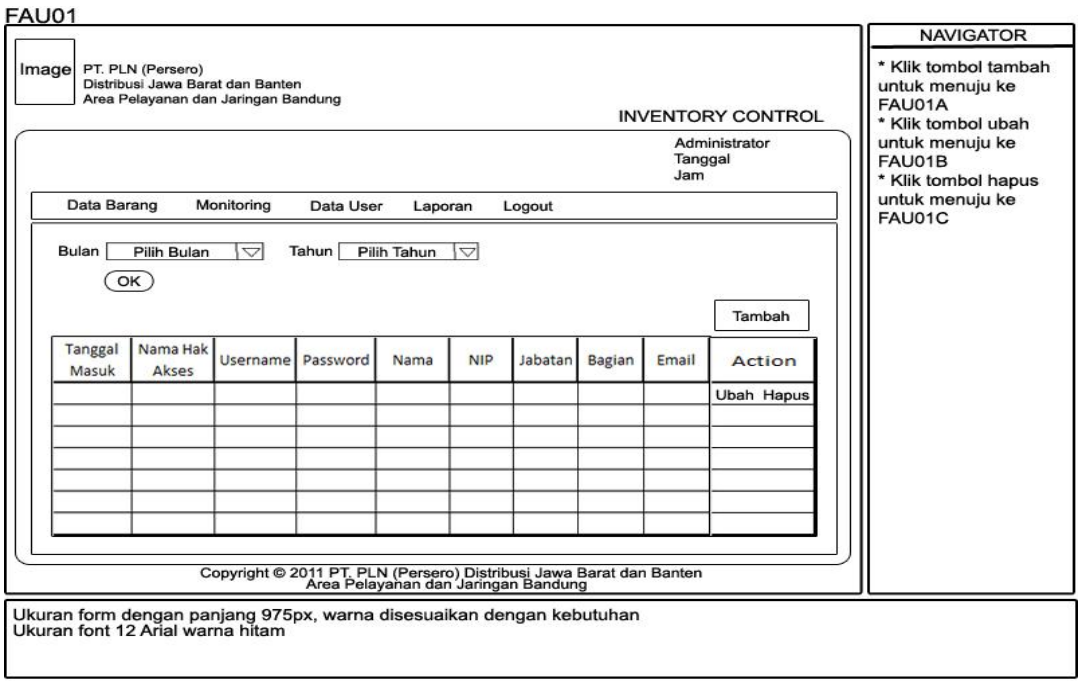
Gambar 3.58 Form Monitoring

17. Form Ubah Data Barang



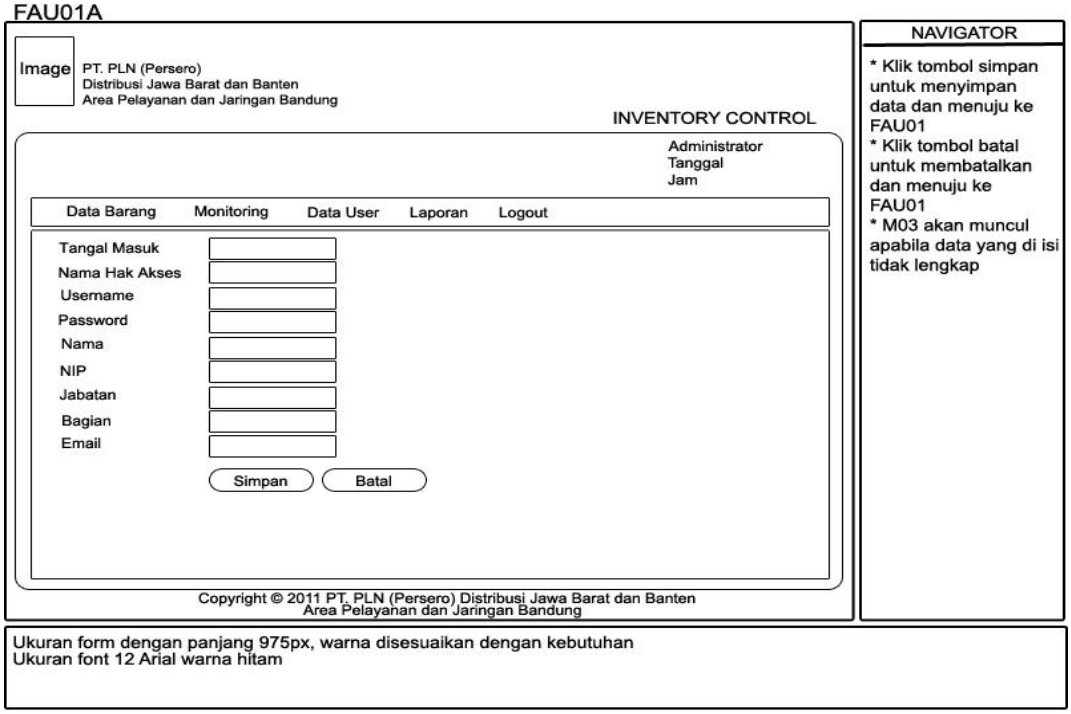
Gambar 3.59 Form Ubah Data Barang

18. Form Data User



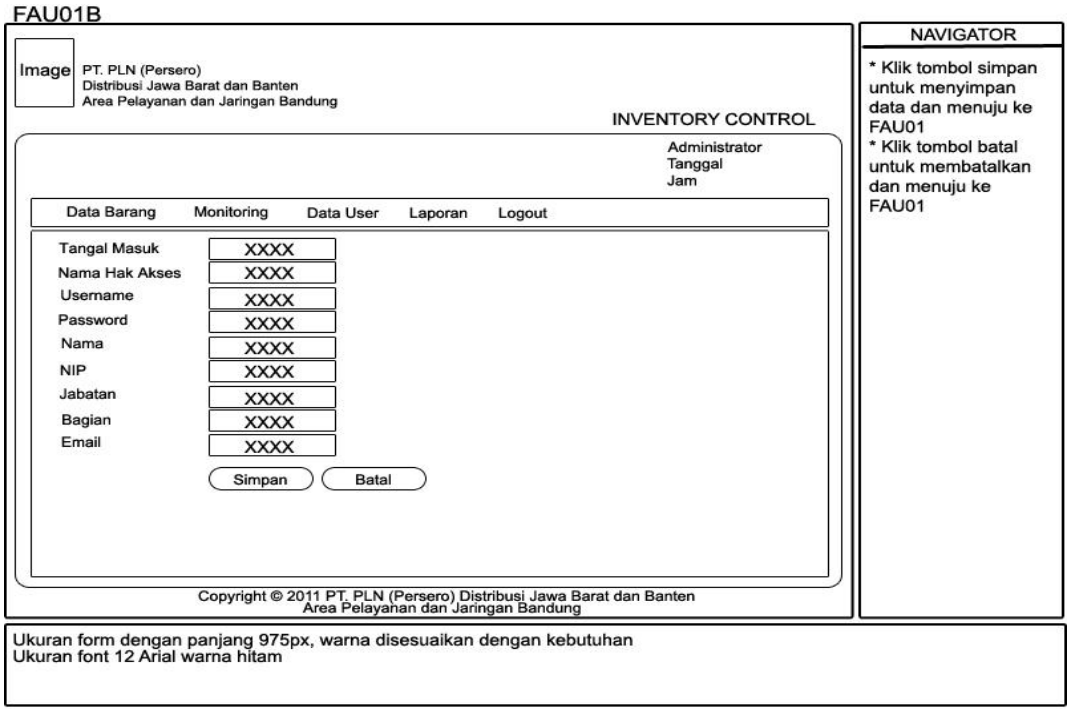
Gambar 3.60 Form Data User

19. Form Tambah Data User



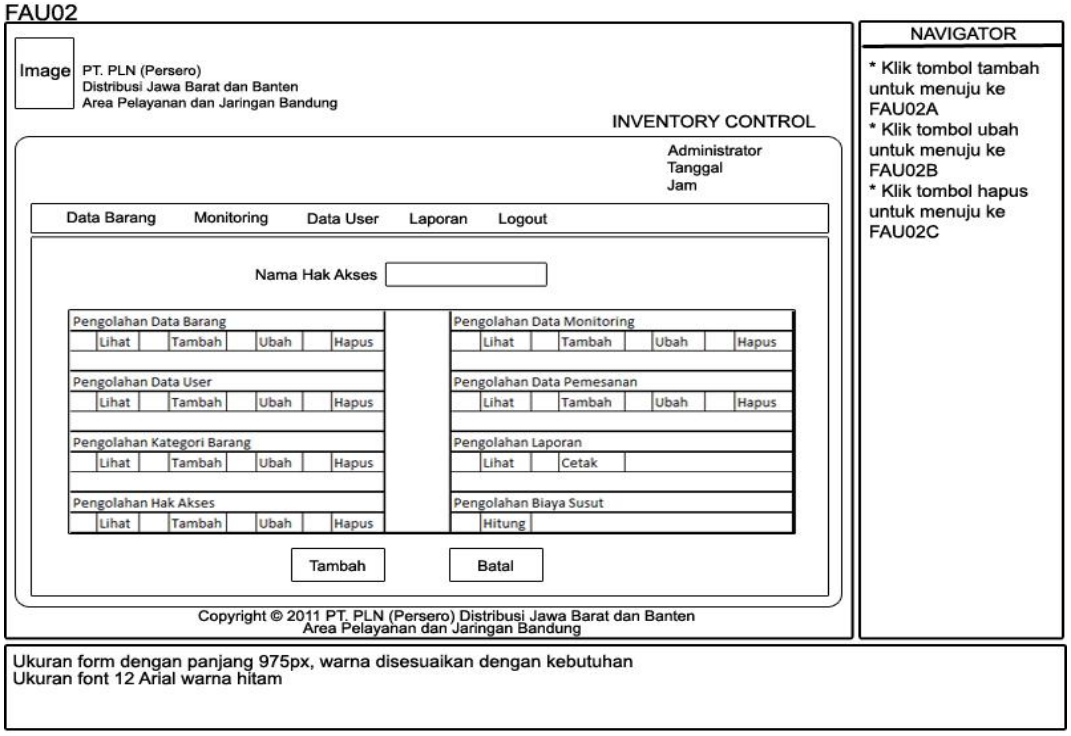
Gambar 3.61 Form Tambah Data User

20. Form Ubah Data User



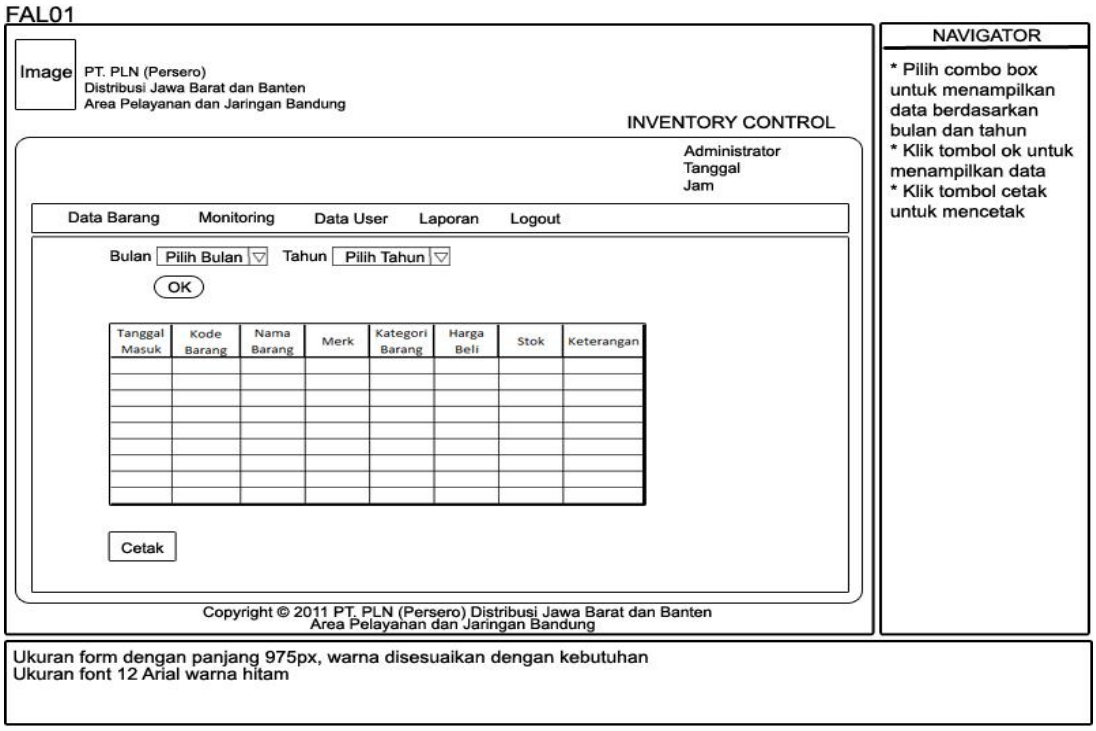
Gambar 3.62 Form Ubah Data User

21. Form Hak Akses



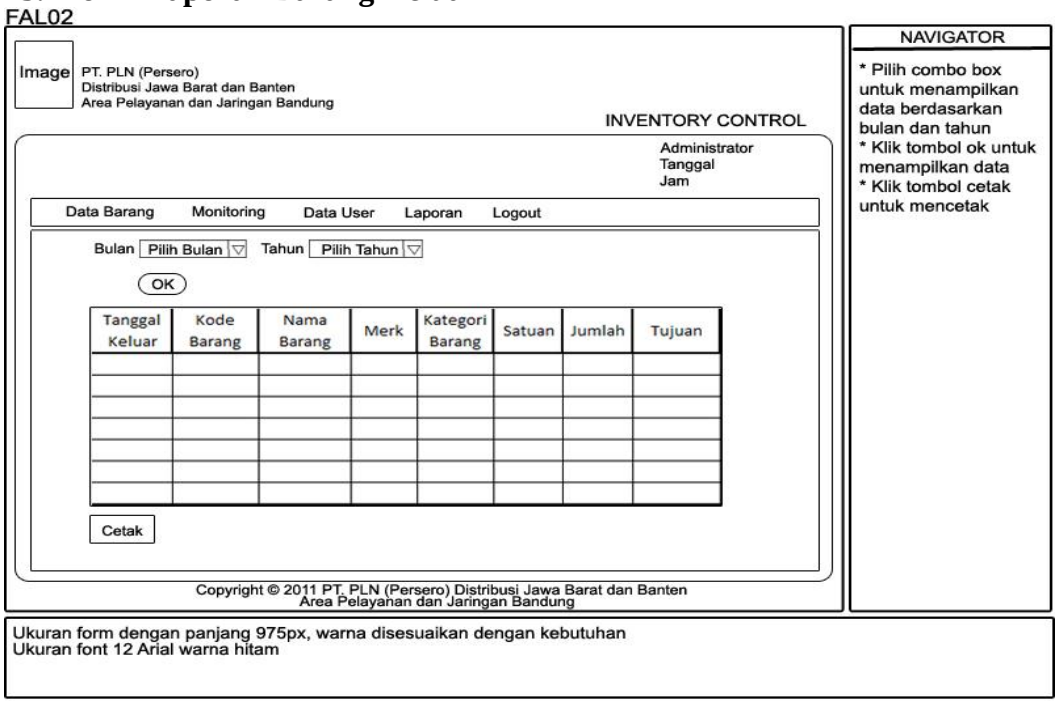
Gambar 3.63 Form Hak Akses

22. Form Laporan Barang Masuk



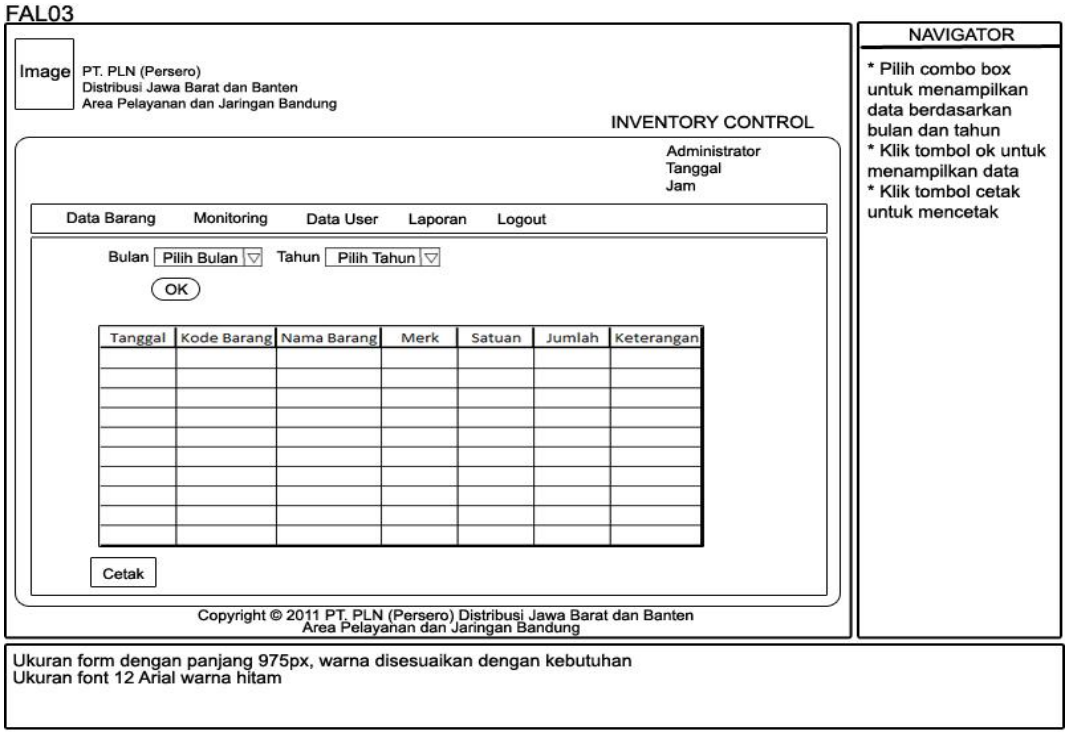
Gambar 3.64 Form Laporan Barang Masuk

23. Form Laporan Barang Keluar



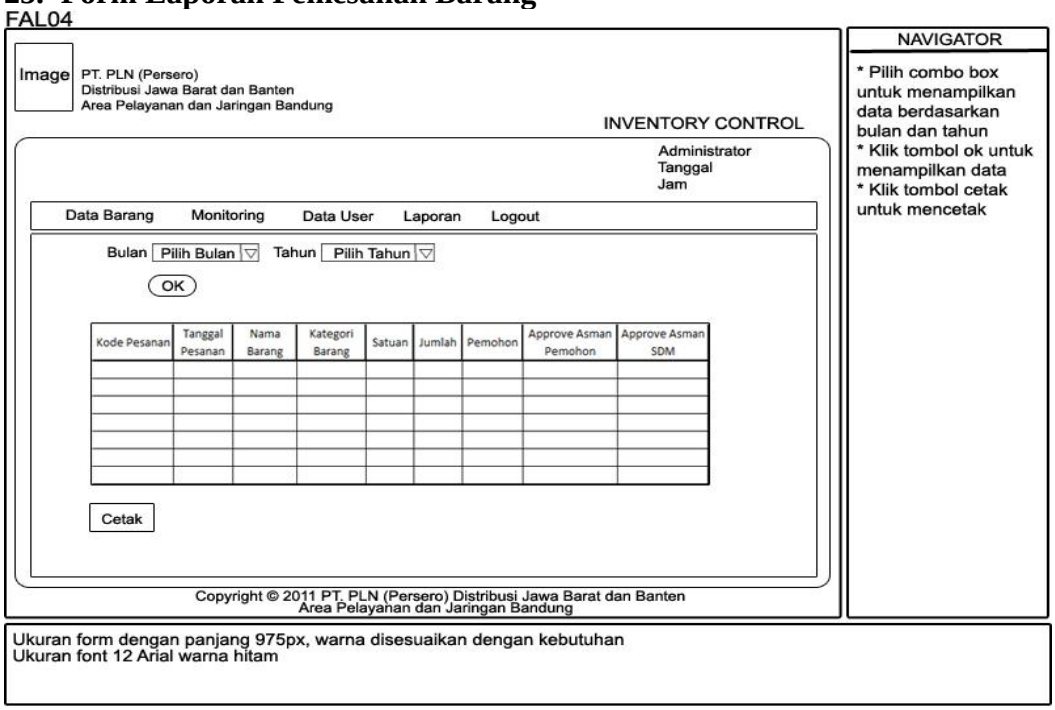
Gambar 3.65 Form Laporan Barang Keluar

24. Form Laporan ATK Bulanan



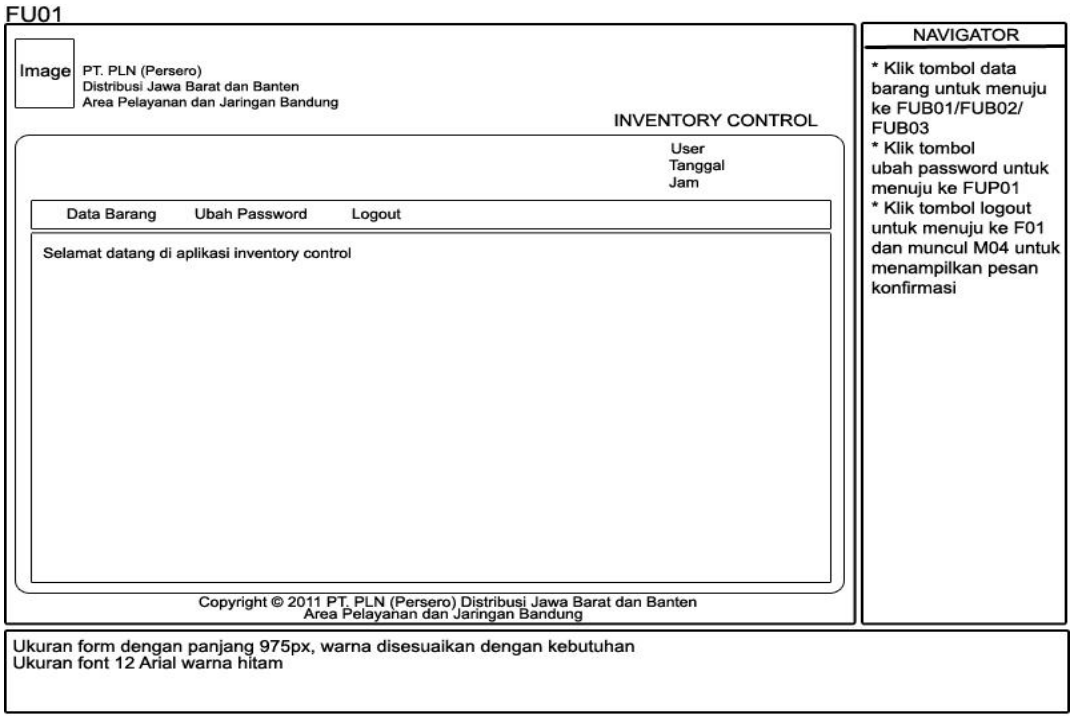
Gambar 3.66 Form Laporan ATK Bulanan

25. Form Laporan Pemesanan Barang



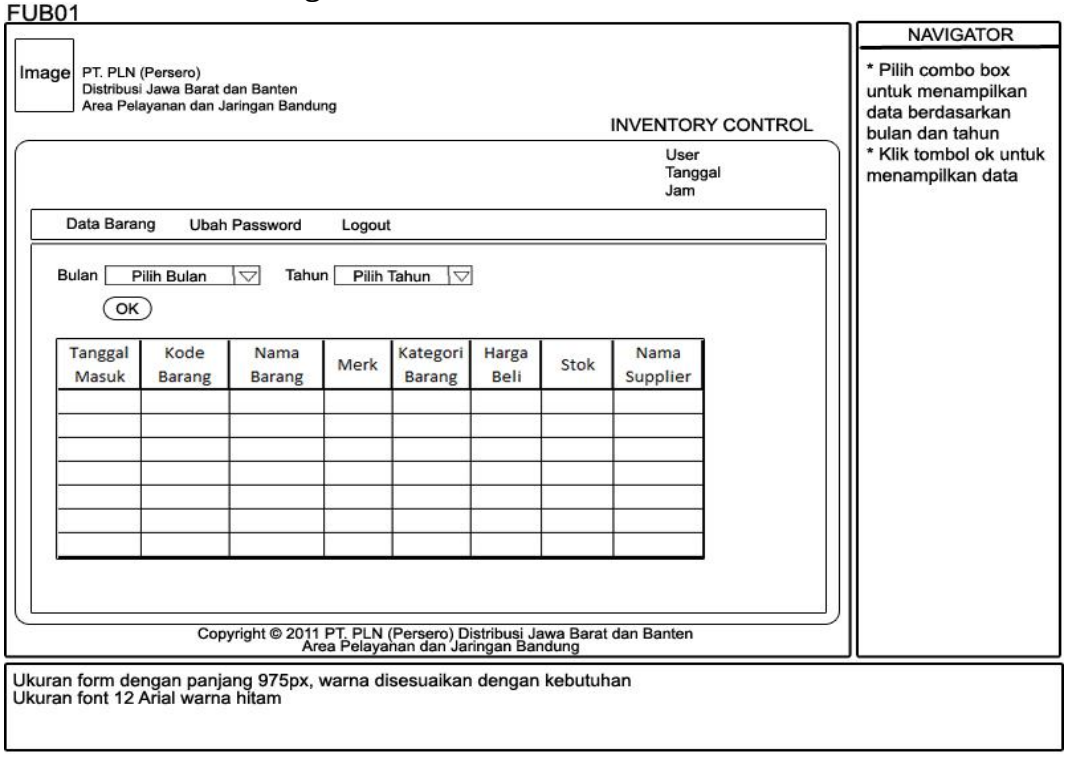
Gambar 3.67 Form Laporan Pemesanan Barang

26. Form Menu User



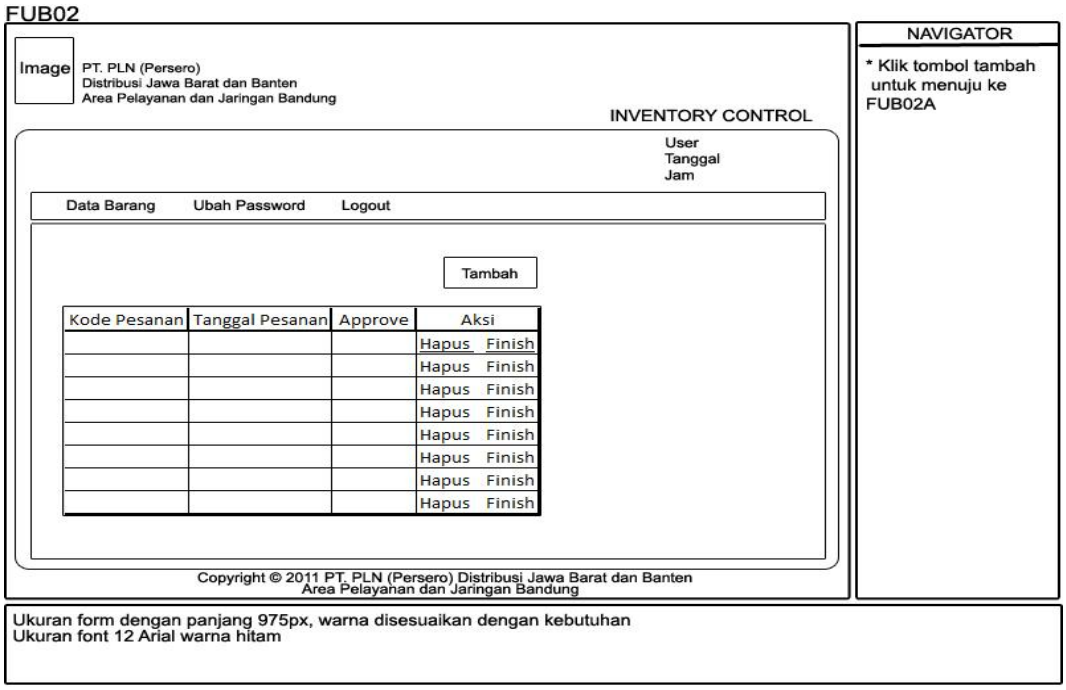
Gambar 3.68 Form Menu User

27. Form Data Barang



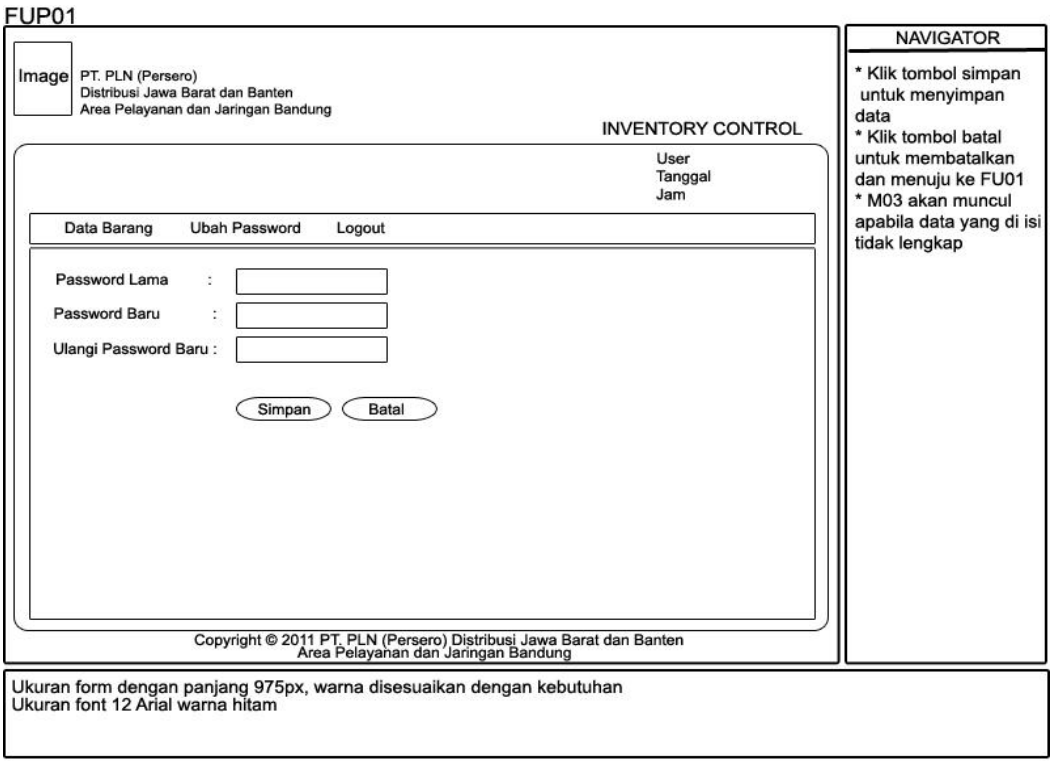
Gambar 3.69 Form Data Barang

28. Form Pemesanan Barang



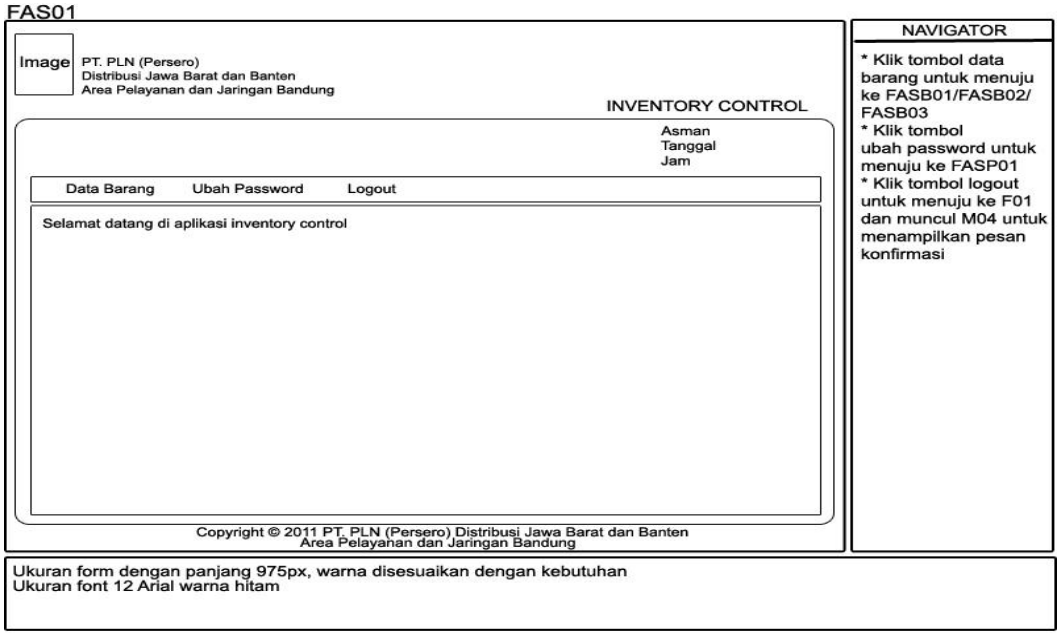
Gambar 3.70 Form Pemesanan Barang

29. Form Ubah Password



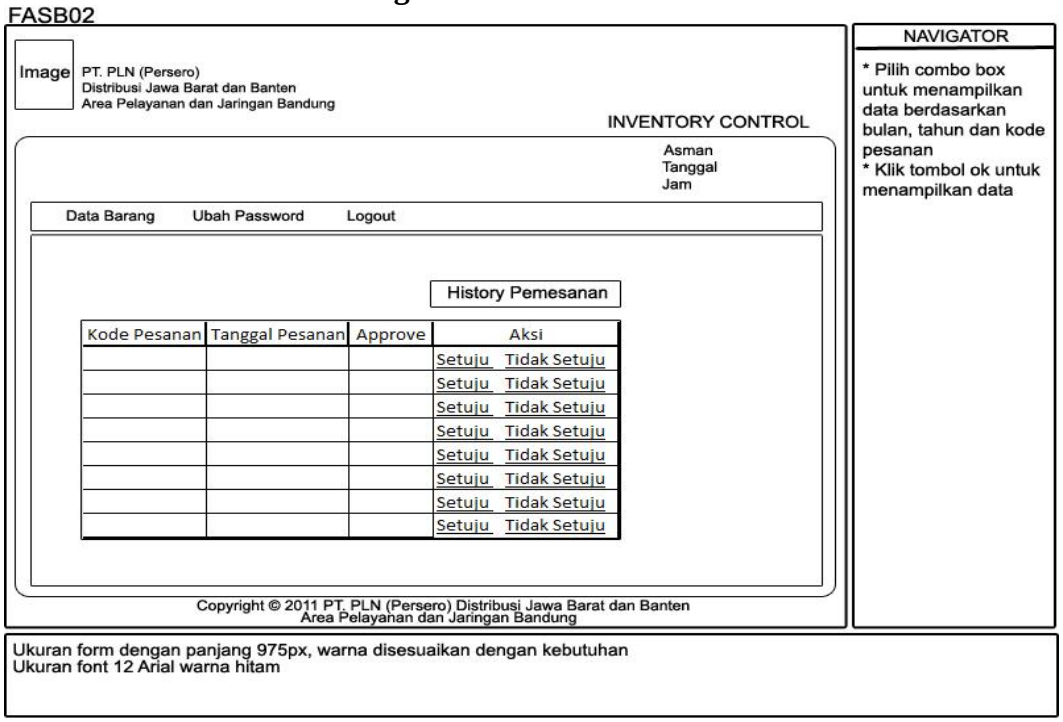
Gambar 3.71 Form Ubah Password

30. Form Menu Asman



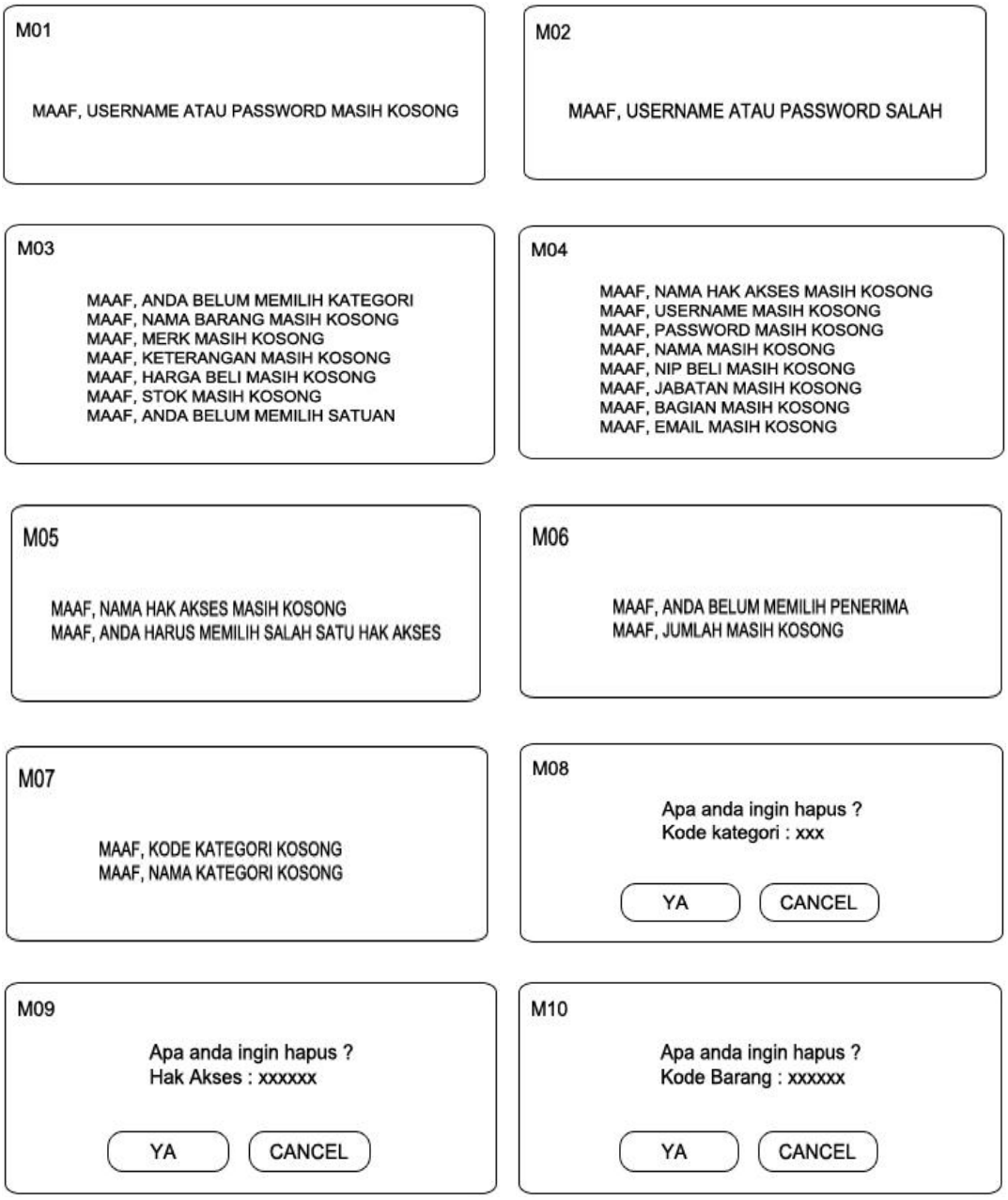
Gambar 3.72 Form Menu Asman

31. Form Pemesanan Barang



Gambar 3.73 Form Pemesanan Barang

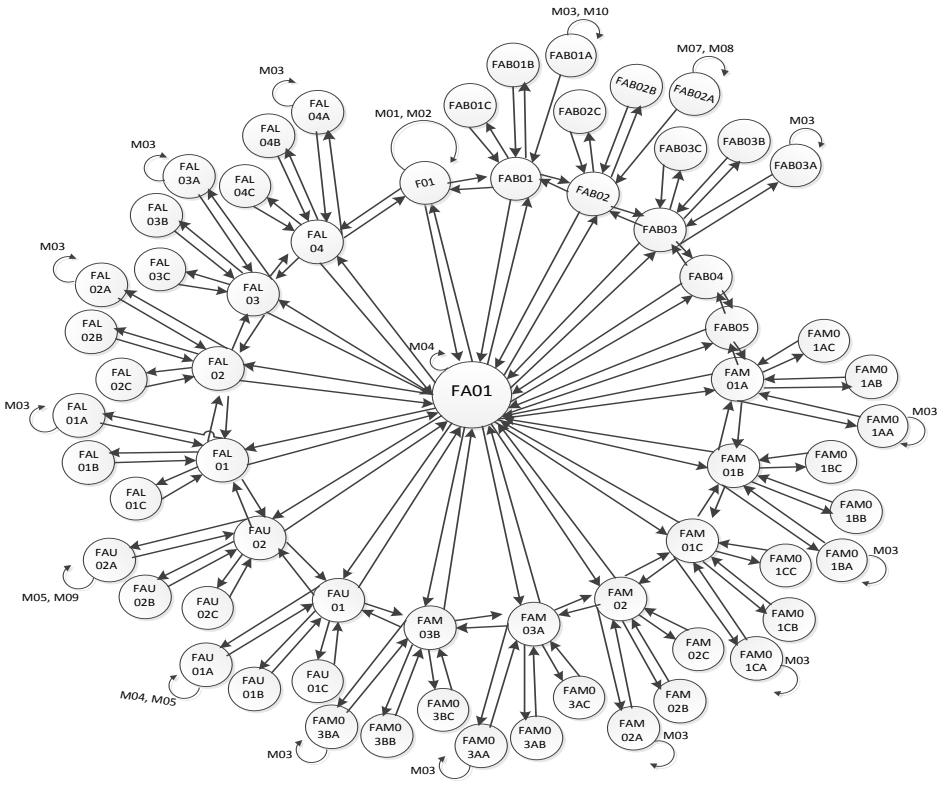
3.2.2.2 Perancangan Pesan



Gambar 3.74 Perancangan Pesan

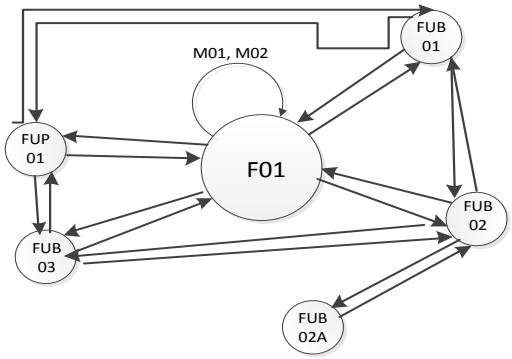
3.2.2.3 Jaringan Semantik

3.2.2.3.1 Jaringan Semantik Admin



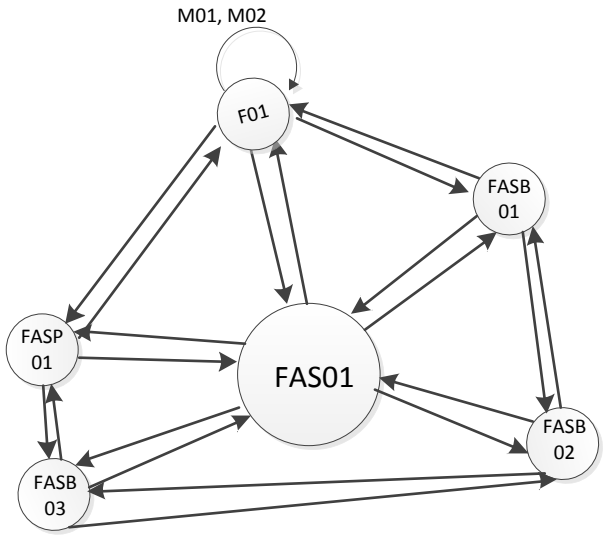
Gambar 3.75 Jaringan Semantik Admin

3.2.2.3.2 Jaringan Semantik User



Gambar 3.76 Jaringan Semantik User

3.2.2.3.3 Jaringan Semantik Asman



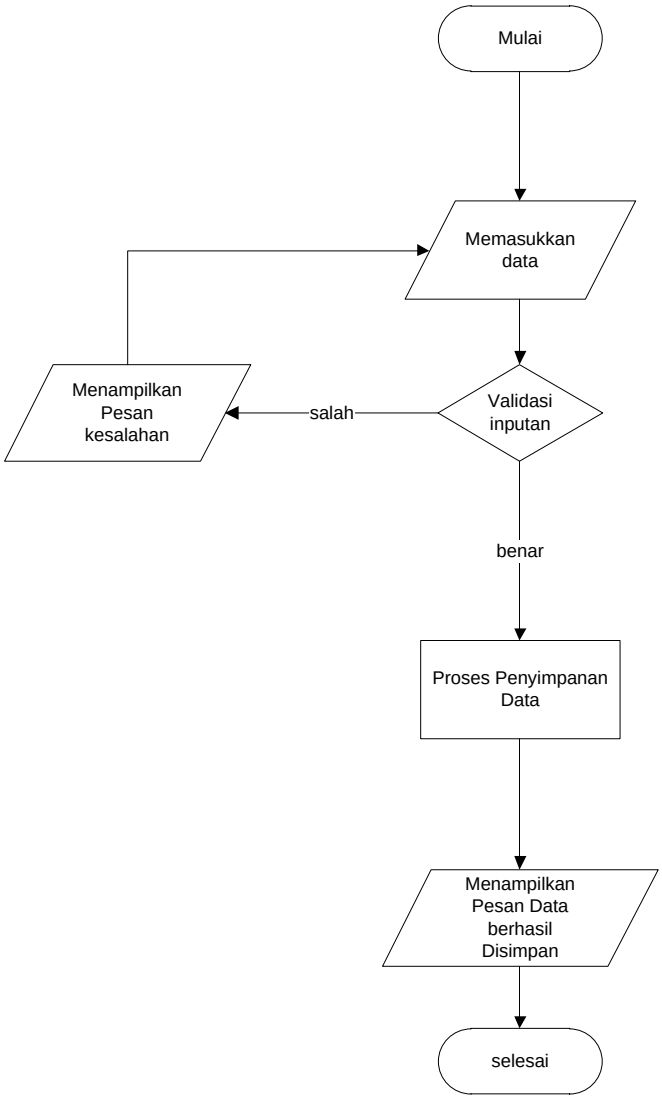
Gambar 3.77 Jaringan Semantik Asman

3.2.3 Perancangan Prosedural

Perancangan prosedural menggunakan flowchart merupakan representasi algoritma dari perangkat lunak sistem informasi *inventory control*. Detail flowchart sistem informasi *inventory control* adalah sebagai berikut :

1. *Flowchart* Penambahan Data

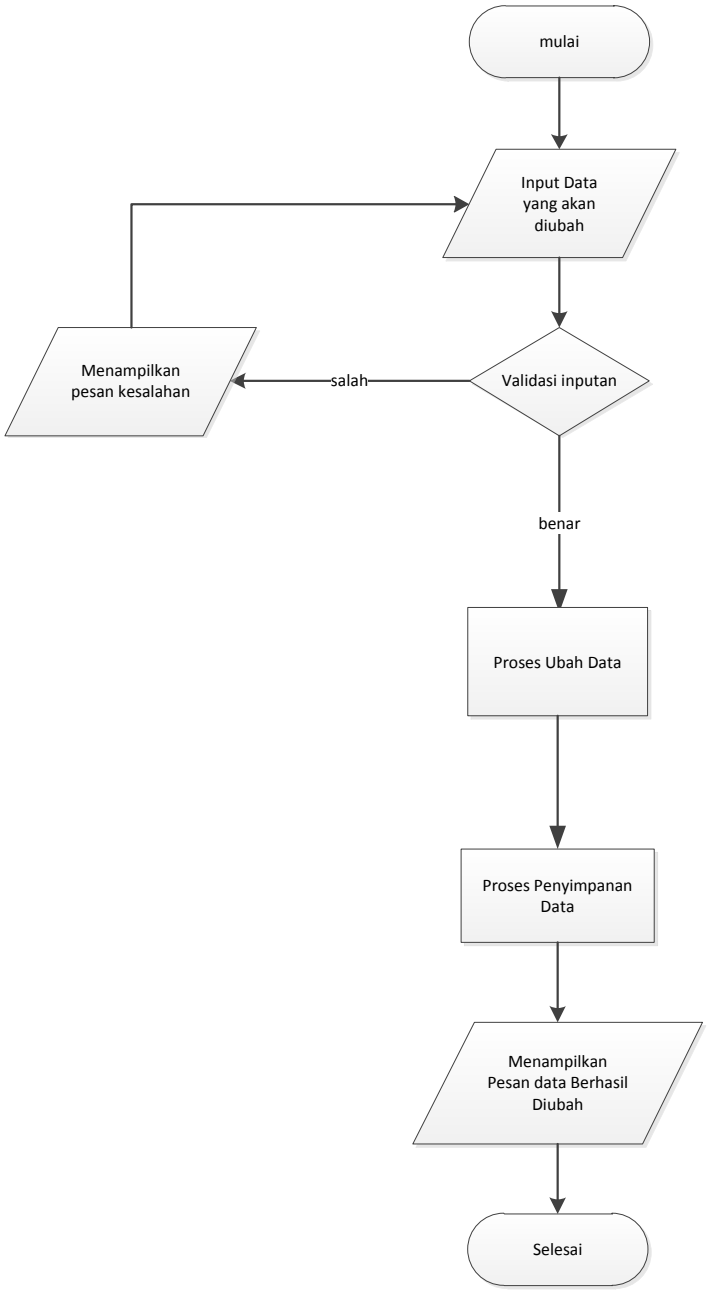
Flowchart penambahan data ini menggambarkan langkah-langkah yang dilakukan untuk penambahan data pada sistem yang dikembangkan.



Gambar 3.78 *Flowchart* Penambahan Data

2. *Flowchart* Perubahan Data

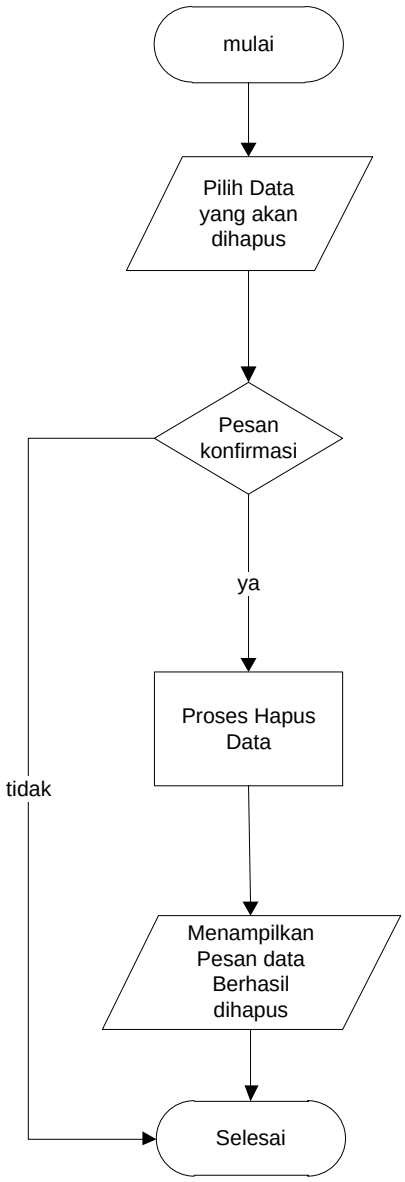
Flowchart pengubahan data ini menggambarkan langkah-langkah yang dilakukan untuk mengubah data pada sistem yang dikembangkan.



Gambar 3.79 *Flowchart* Perubahan Data

3. *Flowchart* Penghapusan Data

Flowchart penghapusan data ini menggambarkan langkah-langkah yang dilakukan untuk menghapus data pada sistem yang dikembangkan.



Gambar 3.80 *Flowchart* Penghapusan Data