BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Analisis sistem merupakan kegiatan penguraian suatu sistem yang utuh dan nyata ke dalam bagian-bagian atau komponen-komponen komputer yang bertujuan untuk mengidentifikasi serta mengevaluasi masalah dan hambatan yang mungkin terjadi pada sistem dan kebutuhan-kebutuhan yang diharapkan, sehingga dapat diusulkan perbaikan pada sistem tersebut dan menyusun sistem baru.

Tahapan analisis harus dilakukan dengan teliti agar diketahui detil yang ada dalam sistem yang berjalan saat ini. Beberapa hal yang akan dianalisis adalah sebagai di antaranya:

1. Analisis Prosedur yang Berjalan
2. Analisis Kebutuhan Non-Fungsional
3. Analisis Kebutuhan Fungsional

3.1.1 Analisis Prosedur yang Berjalan

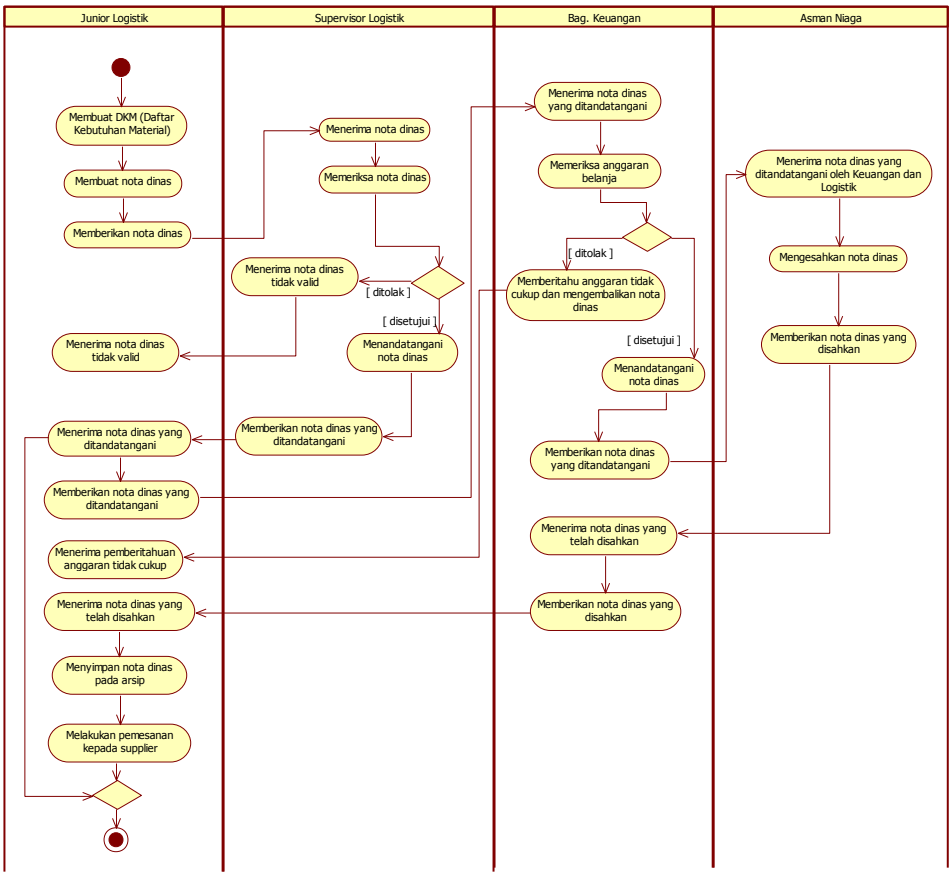
Prosedur merupakan urutan dari langkah-langkah yang terjadi atau yang dilakukan di dalam suatu sistem. Berdasarkan penelitian yang dilakukan, terdapat beberapa prosedur yang sedang berjalan di antaranya prosedur pemesanan barang, penerimaan barang dan prosedur pengeluaran barang.

3.1.1.1 Prosedur Pemesanan Barang

Prosedur pemesanan barang yang berjalan di PT. PLN Persero APJ Majalaya adalah sebagai berikut.

1. *Junior* logistik membuat daftar kebutuhan barang (DKM).
2. Setelah itu, *junior* logistik akan membuat nota dinas yang berisi daftar pemesanan barang.
3. Nota dinas yang telah dibuat akan diberikan kepada *supervisor* logistik untuk diverifikasi.
4. Apabila nota dinas tersebut tidak valid, maka nota dinas tersebut akan diberikan kembali kepada *junior* logistik untuk diperbaiki.

5. Sedangkan nota dinas yang valid akan ditandatangani oleh *supervisor* logistik yang berarti bagian telah menyetujui daftar kebutuhan tersebut dan memberikannya kepada *junior* logistik.
6. *Junior* logistik menerima nota dinas yang sudah ditandatangani oleh *supervisor* logistik, kemudian *Junior* logistik akan memberikan nota dinas tersebut ke bagian keuangan .
7. Setelah bagian itu, bagian keuangan akan memeriksa apakah dana yang diperlukan untuk pemesanan barang tersebut tersedia atau mencukupi. Bila tidak, bagian keuangan akan memberitahu bahwa anggaran untuk nota dinas tersebut tidak mencukupi dan nota dinas tersebut ditolak. Apabila anggaran cukup, maka nota dinas tersebut akan ditandatangani juga oleh bagian keuangan dan memberikannya kepada Asman Niaga.
8. Asman Niaga menerima nota dinas tersebut kemudian mengesahkannya.
9. Nota dinas yang telah disahkan oleh Asman Niaga akan diberikan ke bagian keuangan. Setelah itu bagian keuangan akan memberikan nota dinas yang disahkan kepada *junior* logistik.
10. *Junior* logistik menerima nota dinas yang telah disahkan kemudian menyimpanya sebagai arsip.
11. Setelah itu, *junior* logistik akan melakukan pemesanan barang kepada *supplier*.



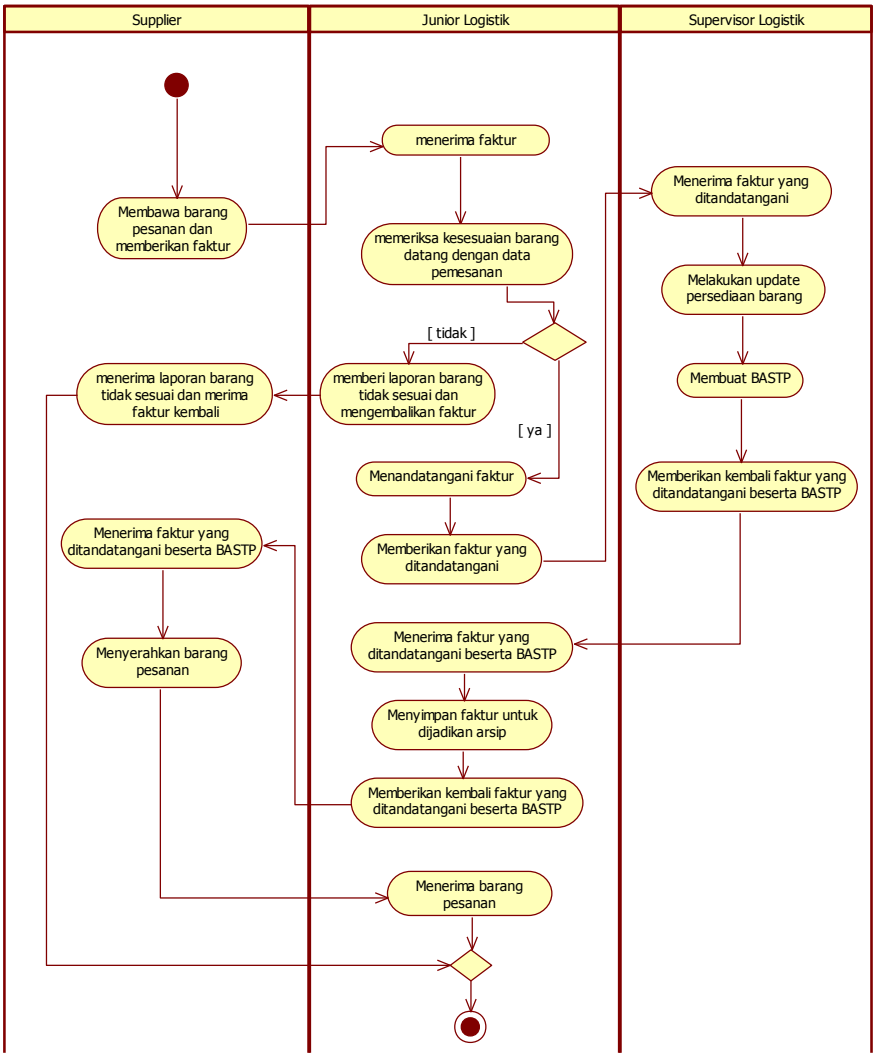
Gambar 3.1 *Activity Diagram* Pemesanan Barang

3.1.1.2 Prosedur Penerimaan Barang

Prosedur penerimaan barang yang berjalan di PT. PLN Persero APJ Majalaya adalah sebagai berikut:

1. *Supplier* membawa barang pesanan dan faktur, kemudian faktur tersebut diberikan kepada *junior* logistik (petugas gudang).
2. *Junior* logistik menerima faktur tersebut kemudian memeriksa apakah barang yang datang sesuai dengan data pemesanan. Bila tidak, petugas faktur tersebut akan diserahkan kembali pada pengirim (*supplier*) dan melaporkan bahwa barang yang datang tidak sesuai dengan pesanan. Bila sesuai, faktur tersebut akan ditandatangani kemudian diberikan kepada *supervisor* logistik.

3. Setelah menerima faktur tersebut, *supervisor* logistik akan memperbaharui (*update*) data persediaan barang. Kemudian *supervisor* logistik akan menadatangani faktur dan membuat BASTP (Berita Acara Serah Terima Pekerjaan) sebagai tanda bukti serah terima barang.
4. Faktur dan BASTP akan diberikan kembali kepada *junior* logistik.
5. Setelah menerima faktur tersebut, *junior* logistik akan menyimpan salah satu rangkap dari faktur tersebut untuk dijadikan arsip dan mencatat barang yang masuk.
6. *Junior* logistik akan memberikan rangkap faktur yang telah ditandatangani lainnya beserta BASTP kepada pengirim barang (*supplier*).
7. Pengirim barang menerima faktur yang telah ditandatangani dan BASTP sebagai bukti barang telah diterima. Selain itu BASTP juga digunakan *supplier* untuk melakukan penagihan kepada pihak PLN.
8. Pengirim akan memberikan barang pesanan kepada *junior* logistik.



Gambar 3.2 *Activity Diagram* Penerimaan Barang

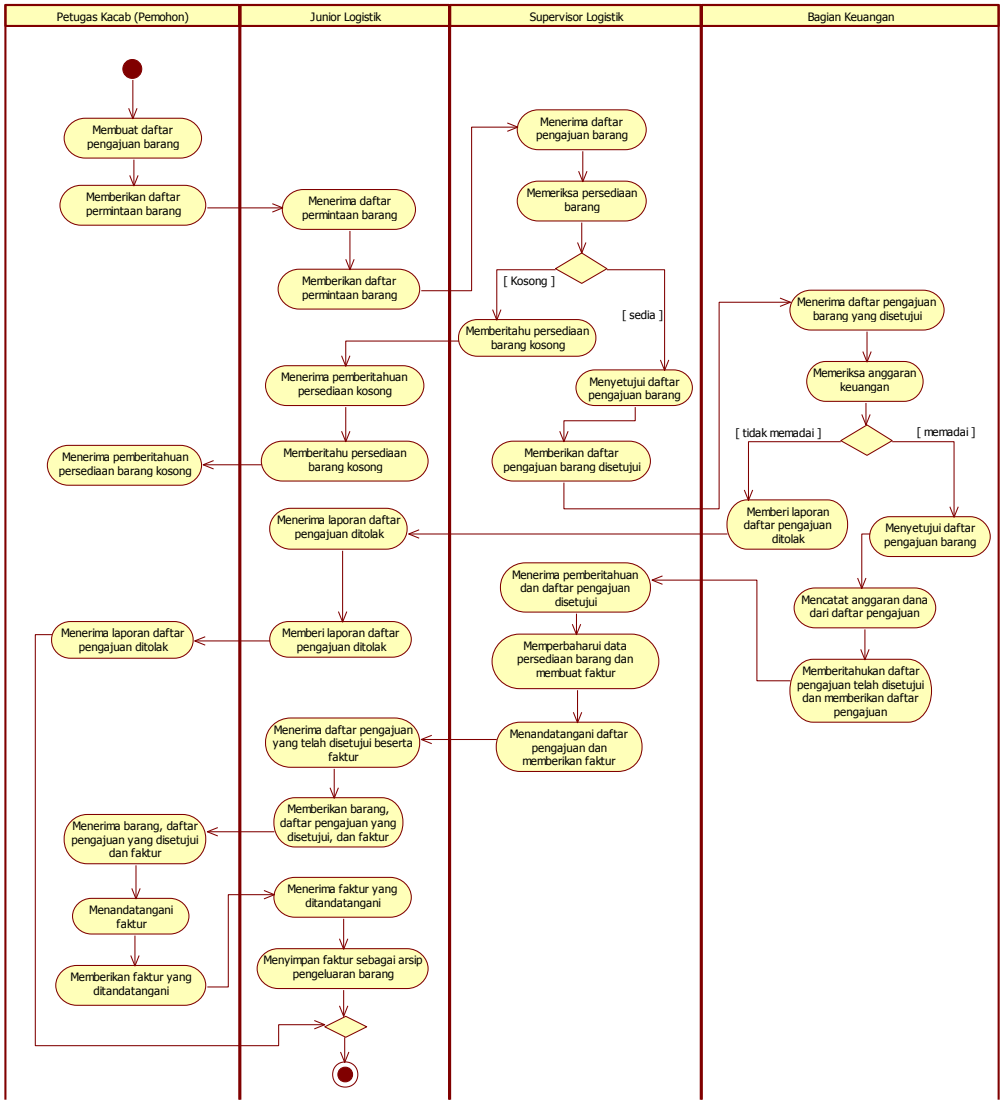
3.1.1.3 Prosedur Pengeluaran Barang

Prosedur pengeluaran barang yang berjalan di PT. PLN Persero APJ Majalaya adalah sebagai berikut:

1. Petugas kacab (pemohon) membawa daftar pengajuan barang dan memberikannya kepada petugas bagian gudang (*junior* logistik).
2. *Junior* logistik akan memberikan daftar pengajuan tersebut kepada *supervisor* logistik.

3. Setelah *supervisor* logistik menerima daftar pengajuan barang tersebut, mereka akan memeriksa apakah barang tersebut tersedia. Bila tidak, *supervisor* logistik akan memberitahu *junior* logistik bahwa material tersebut tidak tersedia (kosong). Kemudian *junior* logistik akan memberitahu pada pihak pemohon (petugas kacab) bahwa barang yang diminta tidak tersedia atau kosong.
4. Bila barang tersebut ada, *supervisor* logistik memberikannya kepada bagian keuangan.
5. Bagian keuangan akan memeriksa apakah anggaran untuk barang yang ada pada daftar pengajuan tersebut memadai. Bila tidak, bagian keuangan akan memberitahukan kepada *junior* logistik bahwa daftar pengajuan ditolak, kemudian *junior* logistik akan mengembalikan daftar pengajuan dan memberitahukan kepada petugas kacab bahwa daftar pengajuan ditolak.
6. Bila anggaran untuk daftar pengajuan tersedia, bagian keuangan akan menyetujui daftar tersebut kemudian akan mencatat anggaran dana dari daftar pengajuan. Setelah itu, bagian keuangan akan memberitahu kepada *supervisor* logistik bahwa daftar pengajuan disejui dan memberikan daftar tersebut kepada *supervisor* logistik.
7. Setelah *supervisor* logistik menerima daftar tersebut, mereka akan membuat faktur sebagai tanda pergerakan barang keluar. Setelah faktur tersebut dibuat, *supervisor* logistik akan memberikannya kepada *junior* logistik beserta daftar pengajuan yang disetujui.
8. *Junior* logistik akan memberikan barang, daftar pengajuan yang disetujui, dan faktur tersebut kepada pihak pemohon (petugas kacab).
9. Petugas kacab (pemohon) menerima barang, lalu faktur yang diberikan oleh *junior* logistik akan ditandatangani. Satu rangkap faktur untuk dibawa oleh petugas kacab, sedangkan rangkap lainnya akan diberikan kembali *junior* logistik.

10. *Junior* logistik yang sudah menerima faktur yang ditandatangani oleh petugas kacab akan menyimpannya sebagai arsip barang keluar (mutasi keluar).



Gambar 3.3 *Activity Diagram* Pengeluaran Barang

3.1.2 Analisis Kebutuhan Non-Fungsional

Analisis nonfungsional adalah sebuah tahap dimana seorang pembangun perangkat lunak menganalisis sumber daya yang akan menggunakan perangkat lunak yang dibangun. Analisis nonfungsional yang dilakukan dibagi dalam enam tahap yaitu:

1. Analisis Pengguna (*user*)
2. Analisis Perangkat keras (*hardware*)
3. Analisis Perangkat Lunak (*Software*)
4. Analisis Jaringan
5. Analisis Metode Q (*Continuous Review Method*)
6. Spesifikasi Kebutuhan Perangkat Lunak

3.1.2.1 Analisis Pengguna (*user*)

Suatu aplikasi dapat berjalan dengan baik apabila ditunjang oleh pengguna yang memiliki kemampuan dalam menjalankan aplikasi yang bersangkutan. Analisis pengguna dimaksudkan untuk mengetahui siapa saja pengguna yang terlibat beserta karakteristiknya sehingga dapat diketahui tingkat pengalaman dan pemahaman pengguna terhadap komputer.

Petugas yang kini mengelola data di bagian logistik PT. PLN Persero APJ Majalaya yaitu petugas dengan jabatan *supervisor* logistik dan *junior* logistik:

1. *Supervisor* Logistik

Umur	:	53 Tahun
Tingkat Pendidikan	:	Strata 1
Kemampuan Komputer	:	Mampu menggunakan Sistem Operasi Windows, MS Word dan MS Excel

2. *Junior* Logistik

Umur	:	47 Tahun
Tingkat Pendidikan	:	SMP
Kemampuan Komputer	:	Mampu menggunakan Sistem Operasi Windows dan MS Word

Petugas dari divisi lain yang berkaitan dengan divisi logistik dalam proses bisnisnya adalah:

1. **Spv. Operasi Distribusi**

Umur	:	40 Tahun
Tingkat Pendidikan	:	STM
Kemampuan Komputer	:	Mampu menggunakan Sistem Operasi Windows, Web Browser, MS Word dan MS Excel

2. **Spv. Perencanaan & Kontruksi**

Umur	:	38 Tahun
Tingkat Pendidikan	:	SMA
Kemampuan Komputer	:	Mampu menggunakan Sistem Operasi Windows, Web Browser, MS Word dan MS Excel

3. **Spv. Pemeliharaan Jaringan**

Umur	:	42 Tahun
Tingkat Pendidikan	:	STM
Kemampuan Komputer	:	Mampu menggunakan Sistem Operasi Windows, Web Browser, MS Word dan MS Excel

4. **Spv. Pengendalian Pengukuran**

Umur	:	39 Tahun
Tingkat Pendidikan	:	STM
Kemampuan Komputer	:	Mampu menggunakan Sistem Operasi Windows, Web Browser, MS Word dan MS Excel

Pengguna (*user*) yang dibutuhkan untuk mengoperasikan aplikasi yang dibagun adalah *Supervisor* Logistik yang akan berperan sebagai aktor Admin di dalam sistem, *Junior* Logistik yang akan berperan sebagai Petugas, sedangkan

petugas divisi lain seperti *Supervisor* Operasi Distribusi, *Supervisor* Perencanaan & Kontruksi, *Supervisor* Pemeliharaan Jaringan, dan *Supervisor* Pengendalian Pengukuran akan berperan sebagai Divisi Lain. Spesifikasi kebutuhan petugas untuk menjalankan aplikasi dari sistem yang dibangun adalah sebagai berikut:

1. **Admin**

Jabatan	:	Spv. Logisitik
Umur	:	Minimal 20 tahun
Tingkat Pendidikan	:	Strata 1
Kemampuan Komputer	:	Mampu menggunakan sistem *inventory control* yang dibangun

2. **Petugas**

Jabatan	:	Jr. Logisitik
Umur	:	Minimal 20 Tahun
Tingkat Pendidikan	:	Minimal SMA/SMK/ sederajat
Kemampuan Komputer	:	Mampu menggunakan sistem *inventory control* yang dibangun

3. **Divisi Lain**

Jabatan	:	Spv. Operasi Distribusi, Spv. Perencanaan & Kontruksi, Spv. Pemeliharaan Jaringan, Spv. Pengendalian Pengukuran
Umur	:	Minimal 20 Tahun
Tingkat Pendidikan	:	Minimal SMA/SMK/ sederajat
Kemampuan Komputer	:	Mampu menggunakan sistem *inventory control* yang dibangun

Pengguna sistem yang sedang berjalan saat ini belum memenuhi sefesifikasi kebutuhan pengguna untuk aplikasi sistem yang dibangun, maka dapat disimpulkan bahwa perlu adanya pelatihan penggunaan sistem kepada petugas (*user*). Pelatihan perangkat pikir (*brainware*) yang ada ditujukan agar pengguna sistem bisa sebaik mungkin menggunakan aplikasi yang dibangun.

3.1.2.2 Analisis Perangkat Keras (*hardware*)

Perangkat keras yang diperlukan untuk menjalankan aplikasi sistem yang dibangun serta untuk mendukung proses kerja sistem tersebut. Spesifikasi perangkat keras yang biasa digunakan di divisi logistik untuk melakukan pengendalian persediaan dan spesifikasi perangkat keras di divisi lainnya saat ini tertera pada Tabel 3.1 .

Tabel 3.1 Analisis Perangkat Keras

No	Perangkat Keras (*hardware*)	Spesifikasi
1	Prosesor	Intel Pentium 4 2.8 Ghz
2	Monitor	CRT 15 inch
3	VGA	On-Board 224 MB
4	Memori (RAM)	512 MB
5	Mouse	Standar
6	Keyboard	Standar
7	Optical Drive	DVD-ROM
8	Harddisk	80 GB
9	Printer	warna

Untuk menjalankan aplikasi pengolahan data *inventory control* diperlukan satu komputer sebagai *server* dan komputer lainnya sebagai *client*. Ada pun spesifikasi minimum perangkat keras untuk menjalankan aplikasi yang dibangun adalah sebagai berikut.

1. Komputer *Server*

   Spesifikasi minimum perangkat keras sebuah *server* yang dibutuhkan tertera pada Tabel 3.2.

Tabel 3.2 Speseifikasi Minimum Perangkat Keras *Server*

No	Perangkat Keras (*hardware*)	Spesifikasi
1	Prosesor	Intel Pentium 4 2.8 Ghz
2	Monitor	Monitor 15 inch
3	VGA	On-Board 224 MB
4	Memori (RAM)	2 GB
5	Mouse	Standar
6	Keyboard	Standar
7	Harddisk	250 GB
9	Ethernet Card	RTL813 PCI 10/100 Mbps
10	Switch	24 ports, 10/100Mbps
11	Printer	Printer warna

2. Komputer *Client*

Spesifikasi minimum perangkat keras komputer *client* yang dibutuhkan tertera pada Tabel 3.2.

Tabel 3.3 Speseifikasi Minimum Perangkat Keras *Client*

No	Perangkat Keras (*hardware*)	Spesifikasi
1	Prosesor	Intel Pentium 4 1GHz
2	Monitor	Monitor 15 inch
3	VGA	On-Board 224 MB
4	Memori (RAM)	1024 MB
5	Mouse	Standar
6	Keyboard	Standar
7	Harddisk	250 GB
9	Ethernet Card	RTL813 PCI 10/100 Mbps
10	Printer	Printer warna

Perangkat keras yang ada di divisi logistik saat ini belum memenuhi kebutuhan untuk menjalankan aplikasi yang akan dibangun. Penggantian atau *upgrade* perangkat keras diperlukan agar aplikasi yang dibangun bisa berjalan dengan baik.

3.1.2.3 Analisis Perangkat Lunak (*software*)

Spesifikasi perangkat lunak yang ada di bagian logistik PT. PLN Persero APJ Majalaya saat ini di antaranya:

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

Spesifikasi kebutuhan perangkat lunak untuk mengoptimalkan aplikasi yang akan dibangun diantaranya:

A. Komputer *Server*
   1. Sistem Operasi Windows XP SP2
   2. Web Browser seperti Mozilla Firefox atau Google Chrome
   3. MySQL 5.5
   4. WAMPServer atau XAMPP

B. Komputer *Client*
   1. Sistem Operasi Windows XP SP2
   2. Web Browser seperti Mozilla Firefox atau Google Chrome

Perangkat lunak yang digunakan oleh bagian logistik saat ini tidak memenuhi spesifikasi kebutuhan untuk menjalankan aplikasi yang dibangun dengan baik, maka dapat disimpulkan perlunya instalasi perangkat lunak yang sesuai dengan spesifikasi kebutuhan agar aplikasi yang dibangun dapat bejalan dengan baik.

3.1.2.4 Analisis Jaringan

Setiap bagian dari PT. PLN Persero APJ Majalaya menggunakan komputer sebagai alat bantu dalam melakukan pengolahan data. Tetapi komputer yang ada di setiap bagian atau divisi tidak terhubung dalam jaringan komputer sehingga tidak adanya *sharing* atau sistem berbagi data antar divisi khususnya dengan bagian logisitik. Untuk mencari informasi tentang persediaan barang, divisi lainnya harus mencari tahu dan menanyakan kepada petugas logisitik. Oleh karena itu diperlukan sebuah jaringan komputer untuk aplikasi yang dibangun. Topologi jaringan komputer yang akan digunakan adalah topologi *star* karena topologi ini

dianggap paling efektif untuk jaringan pada sebuah gedung. Teknik *subnetting*, alamat IP yang akan digunakan adalah IP kelas A, karena jaringan yang dibangun tidak membutuhkan jumlah *network* maksimum yang besar. Berikut adalah tahapan dari *subnetting*:

1. Network Address = 10.0.0.0/16
2. Subnetmask = 11111111.11111111.00000000.00000000
   (255.255.0.0)

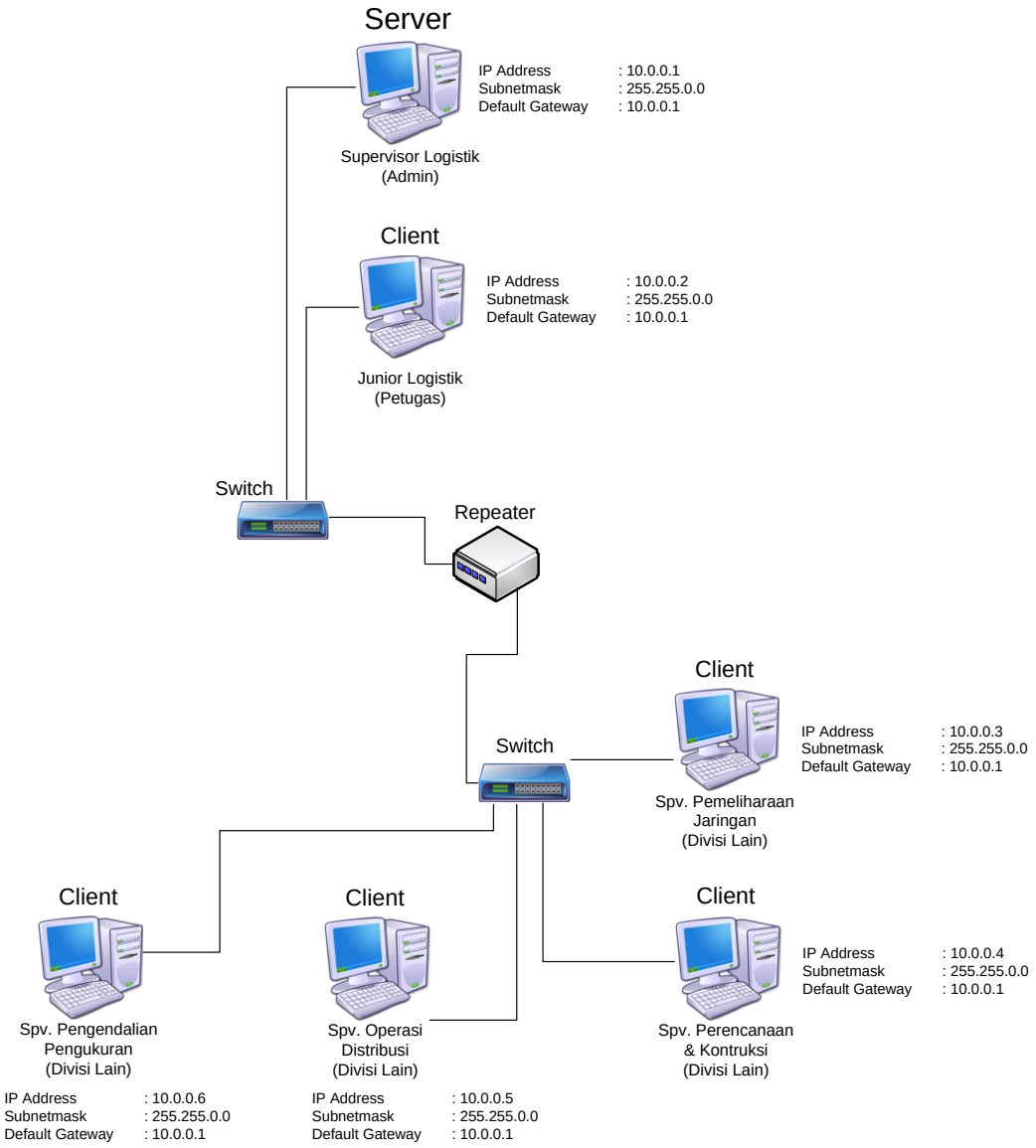
Perhitungan:

1. Jumlah *Subnet* = 2^x, dimana x adalah banyaknya binari 1 pada 2 oktet terakhir. Jumlah *Subnet* = 2^8 = 256 *subnet*
2. Jumlah *Host/Subnet* adalah 2^y-2, di mana y adalah banyaknya binary 0 pada 2 oktet terakhir. Jumlah *Host/Subnet* = 2^{16} – 2 = 65534 *host*
3. Blok *Subnet* = 256-255(nilai octet terakhir *subnet mask*) = 1. Jadi subnet lengkapnya: 1,2,3,4,…255
4. Alamat *host* valid adalah seperti yang tercantum pada table di bawah ini:

Tabel 3.4 Tabel *Subnetiting*

Subnet	10.0.0.0	10.1.0.0	...	10.254.0.0	10.255.0.0
Host Pertama	10.0.0.1	10.1.0.1	…	10.254.0.1	10.255.0.1
Host Terakhir	10.0.255.254	10.1.255.254	…	10.254.255.254	10.254.255.254

Arsitektur jaringan yang akan dibuat untuk sistem informasi yang dibangun dapat dilihat pada Gambar 3.4.



Gambar 3.4 Arsitektur Jaringan Komputer PT. PLN Persero APJ Majalaya

3.1.2.5 Analisis Metode *Continuous Review Method* (Metode Q)

Pada sub bab ini akan dibahas mengenai analisis penggunaan metode yang digunakan untuk persediaan kebutuhan barang atau material. Metode yang digunakan yaitu *Continuous Review System* (Metode Q).

Sebagaimana model probabilistik sederahana, permasalahan kebijakan *inventory* yang dapat dibantu dengan metode atau model Q berkaitan dengan penetuan besarnya stok operasi (*operating stock*) dan cadangan pengamananya (*safety stock*). Kelebihan dari metode ini yaitu bisa digunakan untuk mencari tingkat persediaan yang optimal berdasarkan pada stok operasi dan stok cadangan sehingga tingkat pelayanan terhadap permintaan pengguna barang dapat dikendalikan dengan lebih baik. Selain itu juga, kemungkinan terjadinya persediaan habis (*stock out*) lebih bisa terhindar mengingat aturan pengeluaran barang di PT. PLN tidak terlalu condong terhadap permintaan sehingga peminta barang bisa mencari di tempat lain atau kantor cabang yang lainnya.

Contoh kasus :

PT. PLN Persero APJ Majalaya mengeluarkan alat pengukur KWH dengan nama material **MTR; kWH M; 1p; 220V; 5-20A; 2; ; 2W** dalam satu tahun yang direkap menajdi 6 periode seperti yang tertera pada tabel berikut ini.

Tabel 3.5 Pengeluaran Barang

Periode	Pengeluaran
1	3567
2	3781
3	2628
4	2340
5	2945
6	1812
Total	17073

Pengadaan barang dipasok dari *supplier* dengan ongkos pesan sebesar Rp. 1.000.000,- per pesan dengan waktu estimasi selama 2 bulan. Harga barang Rp. 100.000 per unit. Bila ongkos kekurangan barang sebesar Rp. 5000 per unit barang dan ongkos simpan sebesar 5% dari harga barang per unit untuk setahun.

Tentukan:

1. Kebijakan pengadaan barang/material yang optimal?

Penyelesaian :

Diketahui :

D = 17703 unit/tahun

L = 2 bulan = 1/6 tahun

A = Rp. 1.000.000,- /pesan

P = Rp. 100.000 /unit

h = 5 % x Rp. 100.000 = Rp. 5000 /tahun

C_u = Rp. 5.000/unit

Ditanyakan :

1. Kebijakan pengadaan persediaan (inventori) yang optimal?

Jawab :

Untuk kebijakan persediaan berarti ada 3 hal yaitu :

1. Berapa q_0 (besarnya lot pemesanan)?
2. Berapa *ss* (*safety stock* atau stok aman)?
3. Berepa *r* (*reorder point* atau titik pemesanan kembali)?

Pertama, mencari nilai Standar Deviasi (simpangan baku) terlebih dahulu dengan menghitung deviasi (simpangan) dan varians

Simpangan (deviasi) = $xi - \bar{x}$

Varians (rata-rata simpangan kuadrat) = $\sigma = \frac{\Sigma(xi-\bar{x})}{n}$

Dimana :

xi = pengeluaran ke-i

$\bar{x}$ = rata-rata pengeluaran

σ = varians

n = banyaknya sampel

Tabel 3.6 Perhitungan Standar Deviasi

Periode	Pengeluaran	Simpangan	Simpangan Kuadrat
1	3567	721,5	520562,25
2	3781	935,5	875160,25
3	2628	-217,5	47306,25
4	2340	-505,5	255530,25
5	2945	99,5	9900,25
6	1812	-1033,5	1068122,25
Total	**17073**	**0**	**2776581,5**
Rata-rata	**2845,5**		
Varians			**462763,5833**
Standar Deviasi			**680,2672882**

Dari data tabel di atas ditemukan hasil bahwa :

Standar deviasi (S) = 680, 267 dibulatkan jadi **681 unit.**

Keterangan :

Bila satuan bilangan berupa unit maka angka di belakang koma (,) dibulatkan menjadi bernilai 1 yang ditambahkan pada angka sebelum tanda koma (,)

Standar deviasi selama waktu estimasi (S_L) = S x L

S_L = 681 x 1/6 = 113,5 unit dibulatkan **114** unit

a. Hitung q_0^* dengan menggunakan formula Wilson:

$$q_0^* = \sqrt{\frac{2\,A.D}{h}}$$

$$= \sqrt{\frac{2\,(1.000.000)(17.073)}{5000}}$$

$q_0^* =$ 2613,273 unit dibulatkan jadi **2.614** unit

b. Hitung α (jumlah kemungkinan terjadinya kekurangan persediaan) dan r^* (titik pemesanan kembali).

$$\alpha = \frac{hq_0^*}{c_u D + hq_0^*}$$

$$= \frac{(5000)(2614)}{[(5000)(17703)] + [(5000)(2614)]}$$

$\alpha = 0{,}133$

Berdasarkan tabel distribusi normal standar untuk $\alpha = 0{,}133$ diperoleh z_α (faktor pengamanan) = 1,11

$$r = DL + z_\alpha S\sqrt{L}$$
$$r = (17703)\,{}^{1}/_{6} + 1{,}11(681\sqrt{1/6})$$
$$r = 2950{,}5 + 311{,}342$$
$$r = \mathbf{3.155}\ \text{unit}$$

c. Hitung q_{01}^* dengan menggunakan formula :

$$q_{01}^* = \sqrt{\frac{2D\left[A + c_u \int_{r_1^*}^{\infty}(x - r_1^*)f(x)dx\right]}{h}}$$

$$\int_{r_1^*}^{\infty}(x - r_1^*)f(x)dx = N = S_L[f(z_\alpha) - z_\alpha\psi(z_\alpha)]$$

Sebelum mencari , cari terlebih dahulu nilai N (ekspektasi permintaan yang tidak terpenuhi). Untuk menghitung N, cari nilai variabel ordinat atau $f(z_\alpha)$ dan ekspektasi parsial $\psi(z_\alpha)$. Dari Tabel Distribusi Standar, untuk $z_\alpha = 1{,}11$ diperoleh $f(z_\alpha) = 0{,}2155$ dan $\psi(z_\alpha) = 0{,}06486$ sehingga ekspektasi permintaan yang tak terpenuhi (N) adalah:

$$N = S_L[f(z\alpha) - z\alpha\psi(z\alpha)]$$
$$N = 114[0{,}2155 - 1{,}11(0{,}06486)]$$
$$N = 114[0{,}2155 - 0{,}071995]$$
$$N = 16{,}3596 \text{ dibulatkan } \mathbf{17\ unit}$$

Setelah N diketahui, hitung q_{01}^* ;

$$q_{01}^* = \sqrt{\frac{2(17.073)(1.000.000 + (5000 \times 17))}{5000}}$$

$q_{01}^* = \mathbf{2.723\ unit}$ (dibulatkan)

d. Hitung kembali α dan r_1^*

$$\alpha = \frac{hq_{01}^*}{c_u D + hq_{01}^*} = \frac{5000(2723)}{[(5.000)(17.073)] + [(5.000)(2723)]}$$

$\alpha = 0{,}137 \longrightarrow z_\alpha = 1{,}1$

$= DL + z_\alpha S\sqrt{L}$

$= (17.073)\,{}^1/_6 + 1{,}1(681\sqrt{1/6})$

$= 2845{,}5 + 305{,}7826$

$= \mathbf{3.152}$ unit (dibulatkan)

Kebijakan inventori optimal beserta *safety stock* nya adalah:

1. Kebijakan inventori optimal

   $q_0^* = q_{01}^* =$ **2.723 unit** (besarnya lot pemesanan)

   $r_1^* = r_1^* =$ **3.152** unit (titik pemesanan kembali)

   $ss = z_a S\sqrt{L}$ (stok aman)

   $ss = 1{,}1(681\sqrt{1/6})$

   $ss = 305{,}818$ dibulatkan jadi **306 unit**

Rencana kebutuhan adalah (q_{01}^* + ss) x jumlah periode yaitu (2723 + 306) x 6 = **18174** unit.

Evaluasi

Saat ini PT. PLN Persero APJ Majalaya masih menjadikan total jumlah pengeluaran barang pada tahun sebelumnya sebagai patokan untuk menentukan rencana kebutuhan persediaan tahun mendatang. Proses pengadaan barang bersifat konstan baik dari segi kuantitas pemesanan maupun dari segi jarak waktu pengadaan barang dengan ketentuan sebagai berikut:

a. Jumlah pemesanan/pengadaan awal = 5.000 unit
b. Jumlah pemesanan/pengadaan selanjutnya = 3.000 unit/pemesanan
c. Jarak waktu pengadaan = 2 bulan / 6 kali per tahun
d. Total Persediaan = 20.000 unit
e. Total Pengeluaran = 17.073 unit

f. Sisa (Barang tak laku) = **2.927 unit**

Sedangkan dengan metode Q, kebijakan inventori/persediaan untuk periode mendatang adalah sebagai berikut:

a. Rencana Kebutuhan = 18.174 unit
b. Lot Pemesanan = 2.723 unit/pemesanan
c. Stok Aman = 306 unit
d. Titik Pemesanan kembali = 3.152 unit

Pemesanan/pengadaan awal periode adalah sebesar jumlah Lot Pemesanan, sedangkan untuk periode mendatang jumlah pengadaan barang adalah sebesar Lot Pemesanan + Stok Aman (cadangan pengaman). Sedangkan Titik pemesanan kembali (*reorder point*) adalah indikator untuk melakukan pemesanan kembali apabila persediaan telah mencapai titik tersebut atau lebih kecil dari titik tersebut.

Pada Tabel 3.7 dijelaskan mengenai langkah-langkah penerapan metode Q apabila diterapkan pada contoh kasus seperti yang dipaparkan sebelumnya sekaligus untuk membandingkan antara kebijakan persediaan PT. PLN Persero APJ Majalaya saat ini dengan kebijakan persediaan metode Q.

Tabel 3.7 Evaluasi Metode Q

Periode	Lot Pemesanan	Stok Aman	Mutasi Masuk	Persediaan Awal	Pengeluaran	Persediaan Akhir
1	2723	0	2723	2723		
	2723	306	3029	5752	3567	2185
				2185		
2	2723	306	3029	5214	3781	1433
				1433		
3	2723	306	3029	4462	2628	1834
				1834		
4	2723	306	3029	4863	2340	2523
				2523		
5	2723	306	3029	5552	2945	2607
				2607		
6	0	0	0	2607	1812	795
				795		
Total			**17868**	**795**	**17073**	

Pada periode ke-6 yang tidak dilakukan pemesanan/pengadaan barang. Hal ini dimaksudkan untuk menghindari stok barang sisa yang terlalu berlebihan karena menurut kejadian di setiap akhir tahun, tingkat pengeluaran barang selalu berjumlah jauh lebih kecil dari periode sebelum akhir tahun. Sisa barang (tak terjual) metode Q adalah sejumlah **795 unit**, sedangkan PT. PLN Persero APJ Majalaya adalah sejumlah **2927 unit**. Hal ini disebabkan karena metode Q lebih fokus mencari optimasi tingkat pelayanan dengan mempertimbangkan perubahan angka pada tingkat permintaan terpenuhi (pengeluaran barang).

3.1.2.6 Spesifikasi Kebutuhan Perangkat Lunak

Berdasarkan hasil evaluasi prosedur yang berjalan, solusi yang ditawarkan adalah dengan membangun aplikasi sistem informasi *inventory control* berbasis *web* di PT.PLN Persero APJ Majalaya. Spesifikasi kebutuhan perangkat lunak berisikan deskripsi dari kebutuhan perangkat lunak yang akan dibangun, baik kebutuhan fungsional maupun non-fungsional.

1. Kebutuhan Non-Fungsional

Spesifikasi kebutuhan perangkat lunak nonfungsional dijelaskan pada Tabel 3.8 berikut.

Tabel 3.8 SKPL NonFungsional

No.	Kode	Kategori	Deskripsi Kebutuhan
1	SKPL-NF-01	*Efficiency (Product Requirement)*	Sistem dapat menghitung jumlah rencana kebutuhan persediaan dan jumlah pemesanan yang harus dilakukan.
3	SKPL-FN-02	*Dependability (Product Requirement)*	Sistem membutuhkan jaringan internet untuk menghubungkan *client* dan *server*.
4	SKPL-NF-03	*Enviromental (Organizatonal Requirement)*	Sistem yang dibangun menggunakan *database* MySQl sebagai sistem manajemen data (DBMS).
5	SKPL-NF-04	*Operational (Organizatonal Requirement)*	Sistem ini menampilkan pesan kesalahan kepada *user* saat terjadi kesalahan (*error*).
6	SKPL-NF-05	*Development (Organizatonal Requirement)*	Untuk menjalankan sistem yang dibangun maka dibutuhkan: a. Microsoft Windows sebagai sistem operasi. b. *Web server* seperti Wamp 2.2. c. *Web browser* untuk mengkases sistem ini.

2. Kebutuhan Fungsional

Spesifikasi kebutuhan perangkat lunak fungsional dijelaskan pada Tabel 3.9 dan Tabel 3.10.

Tabel 3.9 SKPL *User Requirement*

No.	Kode	Deskripsi Kebutuhan
1	SKPL-F-01	Sistem menyediakan layanan login untuk user
2	SKPL-F-02	Sistem dapat mengelola Data Kategori
3	SKPL-F-03	Sistem dapat mengelola Data Barang
4	SKPL-F-04	Sistem dapat mengelola Data Petugas
5	SKPL-F-05	Sistem dapat mengelola Data Supplier
6	SKPL-F-06	Sistem dapat mengelola Data Pemesanan
7	SKPL-F-07	Sistem dapat mengelola Data Penerimaan
8	SKPL-F-08	Sistem dapat mengelola Data Pengeluaran
9	SKPL-F-09	Sistem dapat mengelola Data Laporan Pemesanan
10	SKPL-F-10	Sistem dapat mengelola Data Laporan Penenerimaan
11	SKPL-F-11	Sistem dapat mengelola Data Laporan Pengeluaran
12	SKPL-F-12	Sistem dapat mencari Rencana Inventori

Tabel 3.10 SKPL *System Requirement*

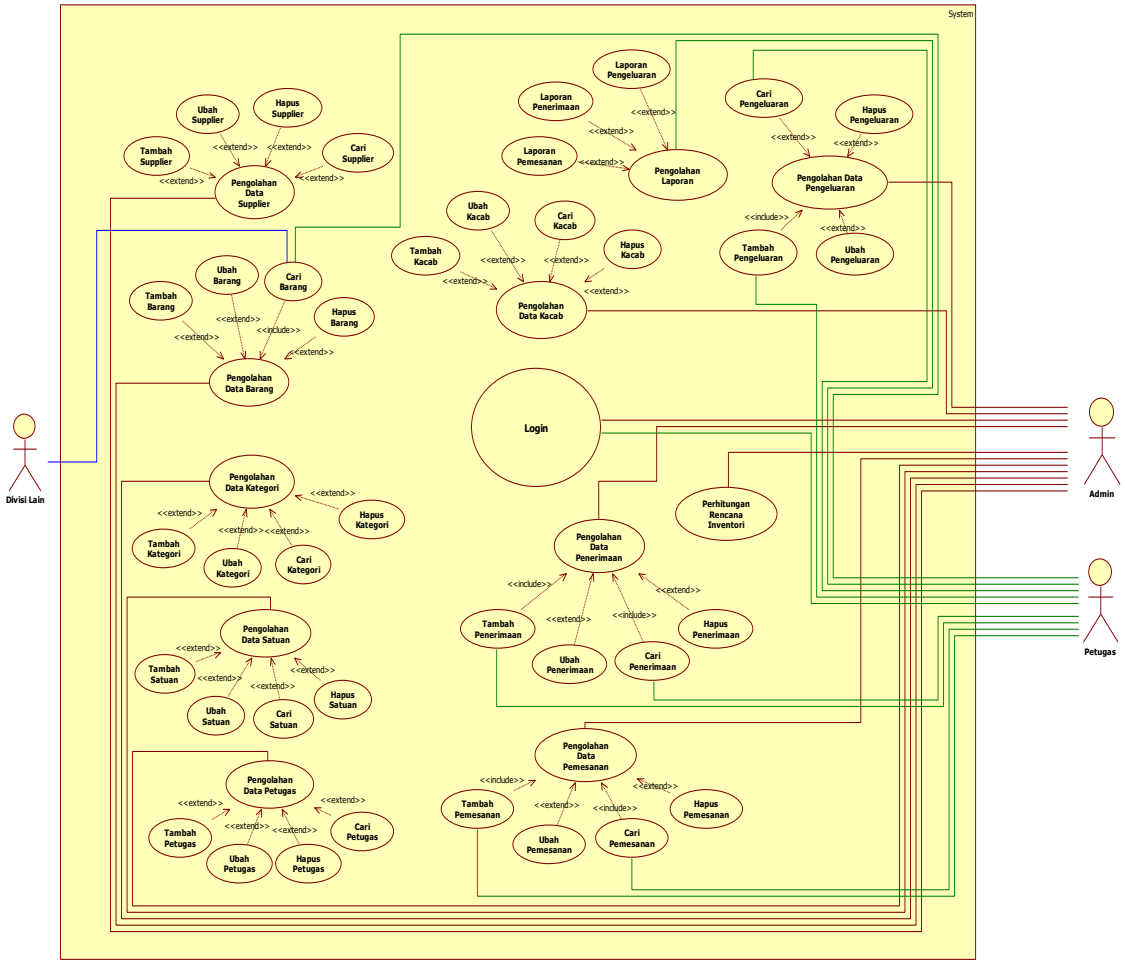
No.	Kode	Deskripsi Kebutuhan
1	SKPL-F-01	Data yang digunakan untuk *login* adalah username dan *password*
2	SKPL-F-02	Sistem dapat melakukan tambah, ubah, cari, dan hapus Data Kategori
3	SKPL-F-03	Sistem dapat melakukan tambah, ubah, cari, dan hapus Data Barang
4	SKPL-F-04	Sistem dapat melakukan tambah, ubah, cari, dan hapus Data Petugas
5	SKPL-F-05	Sistem dapat melakukan tambah, ubah, cari, dan hapus Data Supplier
6	SKPL-F-06	Sistem dapat melakukan tambah, ubah, cari, dan hapus Data Pemesanan
7	SKPL-F-07	Sistem dapat melakukan tambah, ubah, cari, dan hapus Data Penerimaan
8	SKPL-F-08	Sistem dapat melakukan tambah, ubah, cari, dan hapus Data Pengeluaran
9	SKPL-F-9	Sistem dapat mencetak Data Laporan Pemesanan
10	SKPL-F-10	Sistem dapat mencetak Data Laporan Penerimaan
11	SKPL-F-11	Sistem dapat mencetak Data Laporan Pengeluaran
12	SKPL-F-12	Sistem dapat mencetak Data Laporan Penerimaan
13	SKPL-F-13	Sistem dapat mencetak Data Laporan Pemakaian
14	SKPL-F-14	Sistem dapat mencetak Data Laporan Permintaan

3.1.2.7 Analisis Kebutuhan Fungsional

Analisis sistem yang dilakukan menggunakan tools UML. Tahapan analisis sistem menggunakan UML di antaranya adalah *use case diagaram, use case scenario, activity diagram, class diagram, sequence diagram,* dan *deployment diagram.*

3.1.2.8 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. Berdasarkan analisis pengguna aplikasi yang ada maka *use case diagram* untuk aplikasi pengolahan data *inventory control* di PT. PLN Persero APJ Majalaya dapat dilihat dalam Gambar 3.5.



Gambar 3.5 ***Use Case Diagram*** **Sistem** ***Inventory Control***

3.1.2.9 *Use case Scenario*

Use case scenario 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.

3.1.2.9.1 *Use Case Scenario Login*

Merupakan interaksi antara aktor dengan *use case Login*. Dalam melakukan proses *login* terdapat aturan yang tercantum dalam Requirement A.1.

Requirement A.1
Data login terdiri dari *username* dan *password*. *Username* untuk admin dan petugas adalah nama pengguna itu sendiri karena tidak ada ketentuan proses *login* dengan menggunakan NIP sebagai *username* terhadap kedua aktor tersebut.

Skenario *use case login* dapat dilihat pada Tabel 3.11 berikut.

Tabel 3.11 ***Use Case Scenario Login***

Use case name	***Login***	
Related Requirement	Requirement A.1	
Goal In Context	Masuk ke dalam sistem *inventory control*	
Precondition	*User* memiliki bukti identitas akun yang sesuai dengan data pada *database*	
Successful End Condition	*User* masuk ke dalam sistem *inventory control* dan tampil *form* beranda	
Failed End Condition	*User* tidak bisa masuk ke dalam sistem dan tampil pesan kesalahan *login*	
Primary Actor	Admin dan Petugas	
Trigger	*User* memilih tombol *login*	
Include Cases	-	
Main Flow	Step	Action
	1	Sistem menampilkan *form login*
	2	*User* mengisi data *login*
	3	*User* memilih tombol *login*
	4	Sistem melakukan validasi *field login*
	5	Sistem melakukan autentikasi data *login*
	6	*User* masuk ke dalam sistem *inventory control*
Extensions	Step	Branching Action
	4.1	*Field login* tidak diisi
	5.1	Data *login* salah
	5.1	*User* tidak dapat masuk ke dalam sistem

3.1.2.9.2 *Use Case Scenario* Pengolahan Data Petugas

Use case scenario pengolahan data petugas menjelaskan interaksi antara aktor Admin dengan sistem dalam mengolah data petugas di antaranya dengan *use case* tambah petugas, ubah, cari, dan hapus petugas. Dalam melakukan proses pengolahan data petugas terdapat aturan yang tercantum dalam Requirement A.2.

Requirement A.2
NIP dan nama petugas digunakan untuk mencari data yang ada di *database*. NIP, nama petugas, kategori petugas, harus diisi karena data ini dibutuhkan agar bisa digunakan saat melakukan *login*.

A. *Use Case Scenario* Tambah Petugas

Deskripsi dari skenario *use case* tambah petugas dapat dilihat pada Tabel 3.12 berikut.

Tabel 3.12 *Use case Scenario* Tambah Petugas

Use case name	**Tambah Petugas**	
Related Requirement	Requirement A.2	
Goal In Context	Mengelola tambah data petugas pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Tambah petugas berhasil dan tampil *form* lihat petugas	
Failed End Condition	Tambah petugas gagal dan tampil pesan kesalahan	
Primary Actor	Admin	
Secondary Actor	Petugas	
Trigger	*User* memilih tombol simpan	
Include Cases	Pengolahan Data Petugas	
Main Flow	Step	Action
	1	Sistem menampilkan *form* tambah petugas
	2	*User* memilih kategori petugas
	3	*User* mengisi *field input*
	4	*User* memilih tombol simpan
	5	Sistem melakukan validasi data *input*
	6	Data berhasil disimpan di *database*
Extensions	Step	Branching Action
	5.1	Data *input* tidak valid
	5.2	Data tidak tersimpan di *database*

B. *Use Case Scenario* Ubah Petugas

Deskripsi dari skenario *use case* ubah petugas dapat dilihat pada Tabel 3.13 berikut.

Tabel 3.13 *Use case Scenario* Ubah Petugas

Use case name	**Ubah Petugas**	
Related Requirement	Requirement A.2	
Goal In Context	Mengelola ubah data petugas pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Ubah petugas berhasil dan tampil *form* lihat petugas	
Failed End Condition	Ubah petugas gagal dan tampil pesan kesalahan	
Primary Actor	Admin	
Secondary Actor	Petugas	
Trigger	*User* memilih tombol simpan	
Include Cases	Pengolahan Data Petugas	
Main Flow	Step	Action
	1	Sistem menampilkan *form* pengaturan petugas
	2	*User* memilih data yang akan diubah
	3	*User* memilih tombol ubah
	4	Sistem menampilkan *form* ubah petugas
	5	*User* mengisi *field input*
	6	*User* memilih tombol Simpan
	7	Sistem melakukan validasi data *input*
	8	Sistem menyimpan data pada *database*
	9	Data berhasil disimpan
Extensions	Step	Branching Action
	7.1	Data *input* tidak valid
	7.2	Data tidak tersimpan di *database*

C. *Use Case Scenario* Cari Petugas

Deskripsi dari skenario *use case* cari petugas dapat dilihat pada Tabel 3.14 dan Tabel 3.15 berikut.

Tabel 3.14 *Use case Scenario* Cari Petugas

Use case name	**Cari Petugas**
Related Requirement	Requirement A.2
Goal In Context	Mencari data petugas pada sistem
Precondition	*User* sudah berada pada *session login*
Successful End Condition	Menampilkan data hasil pencarian
Failed End Condition	Pencarian data tidak berhasil dan sistem menampilkan pesan data tidak ditemukan
Primary Actor	Admin
Trigger	*User* menekan tombol Enter

Tabel 3.15 *Use case Scenario* Cari Petugas (Lanjutan)

Include Cases	Pengolahan Data Petugas	
Main Flow	Step	Action
	1	Sistem menampilkan *form* pengaturan petugas
	2	*User* mengisi kolom filter pecarian
	3	*User* menekan tombol Enter
	4	Sistem mencari data di *database* sesuai *input*-an *user*
	5	Sistem menampilkan data hasil pencarian
Extensions	Step	Branching Action
	4.1	Sistem tidak menemukan data cari di *database*
	4.2	Tidak ditemukan hasil

D. *Use Case Scenario* Hapus Petugas

Deskripsi dari skenario *use case* hapus petugas dapat dilihat pada Tabel 3.16 berikut.

Tabel 3.16 *Use case Scenario* Hapus Petugas

Use case name	**Hapus Petugas**	
Related Requirement	-	
Goal In Context	Menghapus data petugas pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Hapus petugas berhasil dan tampil *form* pengaturan petugas	
Failed End Condition	Ubah petugas gagal dan tampil pesan kesalahan	
Primary Actor	Admin	
Trigger	*User* memilih tombol hapus	
Include Cases	Pengolahan Data Petugas	
Main Flow	Step	Action
	1	Sistem menampilkan *form* pengaturan petugas
	2	*User* memililh data petugas
	3	*User* memilih tombol hapus
	4	Sistem menampilkan pesan konfirmasi
	5	*User* memilih tombol Ok
	6	Sistem melakukan validasi hapus data
	7	Data berhasil dihapus dari *database*
Extensions	Step	Branching Action
	5.1	*User* memilih tombol Cancel
	5.2	Sistem membatalkan penghapusan data
	6.1	Sistem membatalkan penghapusan data
	6.2	Data gagal dihapus

3.1.2.9.3 *Use Case Scenario* Pengolahan Data Kategori

Use case scenario pengolahan data kategori menjelaskan interaksi antara aktor Admin dalam mengolah data kategori di antaranya dengan *use case* tambah kategori, ubah, cari, dan hapus kategori. Dalam melakukan proses pengolahan data kategori terdapat aturan yang tercantum dalam Requrement A.3.

Requirement A.3
Nama kategori yang ada digunakan untuk atribut dari data barang dan nama kategori harus bersifat unik.

A. *Use Case Scenario* Tambah Kategori

Deskripsi dari skenario *use case* tambah kategori dapat dilihat pada Tabel 3.17 berikut.

Tabel 3.17 *Use case Scenario* Tambah Kategori

Use case name	**Tambah Kategori**	
Related Requirement	Requirement A.3	
Goal In Context	Mengelola tambah data kategori pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Tambah kategori berhasil dan tampil *form* lihat kategori	
Failed End Condition	Tambah kategori gagal dan tampil pesan kesalahan	
Primary Actor	Admin	
Trigger	*User* memilih tombol simpan	
Include Cases	Pengolahan Data Kategori	
Main Flow	Step	Action
	1	Sistem menampilkan *form* pengaturan kategori
	2	*User* memilih menu tambah kategori
	3	Sistem menampilkan *form* tambah kategori
	4	*User* mengisi *field input*
	5	*User* memilih tombol simpan
	6	Sistem melakukan validasi data *input*
	7	Sistem menyimpan data ke *database*
	8	Data berhasil disimpan
Extensions	Step	Branching Action
	6.1	Data *input* tidak valid
	6.2	Data tidak tersimpan di *database*

B. *Use Case Scenario* Ubah Kategori

Deskripsi dari skenario *use case* ubah kategori dapat dilihat pada Tabel 3.18 berikut.

Tabel 3.18 *Use case Scenario* Ubah Kategori

Use case name	**Ubah Kategori**	
Related Requirement	Requirement A.3	
Goal In Context	Mengelola ubah data kategori pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Ubah kategori berhasil dan tampil *form* lihat kategori	
Failed End Condition	Ubah kategori gagal dan tampil pesan kesalahan	
Primary Actor	Admin	
Trigger	*User* memilih tombol simpan	
Include Cases	Pengolahan Data Kategori	
Main Flow	Step	Action
	1	Sistem menampilkan *form* pengaturan kategori
	2	Memilih data kategori yang akan diubah
	3	User memilih menu ubah kategori
	4	*User* mengisi *field input*
	5	*User* memilih tombol simpan
	6	Sistem melakukan validasi data *input*
	7	Sistem menyimpan data ke *database*
	8	Data berhasil disimpan
Extensions	Step	Branching Action
	6.1	Data *input* tidak valid
	6.2	Data tidak tersimpan di *database*

C. *Use Case Scenario* Cari Kategori

Deskripsi dari skenario *use case* cari kategori dapat dilihat pada Tabel 3.19 dan Tabel 3.20 berikut.

Tabel 3.19 *Use case Scenario* Cari Kategori

Use case name	**Cari Kategori**
Related Requirement	
Goal In Context	Mencari data kategori pada sistem
Precondition	*User* sudah berada pada *session login*
Successful End Condition	Menampilkan data hasil pencarian
Failed End Condition	Pencarian data tidak berhasil dan sistem menampilkan pesan data tidak ditemukan
Primary Actor	Admin

Trigger	*User* menekan tombol Enter

Tabel 3.20 *Use case Scenario* Cari Kategori (Lanjutan)

Include Cases	Pengolahan Data Kategori	
Main Flow	Step	Action
	1	Sistem menampilkan *form* pengaturan kategori
	2	*User* mengisi kolom filter pecarian
	3	*User* menekan tombol Enter
	4	Sistem mencari data di *database*
	5	Sistem menampilkan data hasil pencarian
Extensions	Step	Branching Action
	4.1	Sistem tidak menemukan data cari di *database*
	4.2	Tidak ditemukan hasil

D. *Use Case Scenario* Hapus Kategori

Deskripsi dari skenario *use case* hapus kategori dapat dilihat pada Tabel 3.21 berikut.

Tabel 3.21 *Use case Scenario* Hapus Kategori

Use case name	**Hapus Kategori**	
Related Requirement	-	
Goal In Context	Menghapus data kategori pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Hapus kategori berhasil dan tampil *form* pengaturan kategori	
Failed End Condition	Hapus kategori gagal dan tampil pesan kesalahan	
Primary Actor	Admin	
Trigger	*User* memilih tombol hapus	
Include Cases	Pengolahan Data Kategori	
Main Flow	Step	Action
	1	Sistem menampilkan *form* pengaturan kategori
	2	*User* memililh data kategori
	3	*User* memilih tombol hapus
	4	Sistem menampilkan pesan konfirmasi
	5	*User* memilih tombol Ok
	6	Sistem melakukan validasi hapus data
	7	Data berhasil dihapus dari *database*
Extensions	Step	Branching Action
	5.1	*User* memilih tombol Cancel
	5.2	Sistem membatalkan penghapusan data
	6.1	Sistem membatalkan penghapusan data
	6.2	Data gagal dihapus

3.1.2.9.4 *Use Case Scenario* Pengolahan Data Barang

Use case scenario pengolahan data barang menjelaskan interaksi antara aktor Admin dalam mengolah data barang di antaranya dengan *use case* tambah barang, ubah, cari, dan hapus barang. Dalam melakukan proses pengolahan data barang terdapat aturan yang tercantum dalam Requrement A.5.

Requirement A.5
Data barang terdiri dari kode barang yang dibuat oleh sistem, nama barang, kategori, satuan, stok, dan stok minimal. Nama barang harus unik.

A. *Use Case Scenario* Tambah Barang

Deskripsi dari skenario *use case* tambah barang dapat dilihat pada Tabel 3.22 berikut.

Tabel 3.22 *Use case Scenario* Tambah Barang

Use case name	**Tambah Barang**	
Related Requirement	Requirement A.5	
Goal In Context	Mengelola tambah data barang pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Tambah barang berhasil dan tampil *form* lihat barang	
Failed End Condition	Tambah barang gagal dan tampil pesan kesalahan	
Primary Actor	Admin	
Trigger	*User* memilih tombol simpan	
Include Cases	Pengolahan Data Barang	
Main Flow	Step	Action
	1	Sistem menampilkan *form* pengaturan barang
	2	*User* memilih menu tambah barang
	3	Sistem menampilkan *form* tambah barang
	4	*User* mengisi *field input*
	5	*User* memilih tombol simpan
	6	Sistem melakukan validasi data *input*
	7	Sistem menyimpan data ke *database*
	5	Data berhasil disimpan
Extensions	Step	Branching Action
	4.1	Data *input* tidak valid
	4.2	Data tidak tersimpan di *database*

B. *Use Case Scenario* Ubah Barang

Deskripsi dari skenario *use case* ubah barang dapat dilihat pada Tabel 3.23 berikut.

Tabel 3.23 *Use case Scenario* Ubah Barang

Use case name	**Ubah Barang**	
Related Requirement	Requirement A.5	
Goal In Context	Mengelola ubah data barang pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Ubah barang berhasil dan tampil *form* lihat barang	
Failed End Condition	Ubah barang gagal dan tampil pesan kesalahan	
Primary Actor	Admin	
Trigger	*User* memilih tombol simpan	
Include Cases	Pengolahan Data Barang	
Main Flow	Step	Action
	1	Sistem menampilkan *form* pengaturan barang
	2	Memilih data barang yang akan diubah
	3	*User* memilih menu ubah barang
	4	Sistem menampilkan *form* ubah barang
	4	*User* mengisi *field input*
	5	*User* memilih tombol simpan
	6	Sistem melakukan validasi data *input*
	7	Sistem menyimpan data ke *database*
	8	Data berhasil disimpan
Extensions	Step	Branching Action
	6.1	Data *input* tidak valid
	6.2	Data tidak tersimpan di *database*

C. *Use Case Scenario* Cari Barang

Deskripsi dari skenario *use case* cari barang dapat dilihat pada Tabel 3.24 berikut.

Tabel 3.24 *Use case Scenario* Cari Barang

Use case name	**Cari Barang**	
Related Requirement	-	
Goal In Context	Mencari data barang pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Menampilkan data hasil pencarian	
Failed End Condition	Pencarian data tidak berhasil dan sistem menampilkan pesan data tidak ditemukan	
Primary Actor	Admin	
Trigger	*User* menekan tombol Enter	
Include Cases	Pengolahan Data Barang	
Main Flow	Step	Action
	1	Sistem menampilkan *form* pengaturan barang
	2	*User* mengisi kolom filter pecarian
	3	*User* menekan tombol Enter
	4	Sistem mencari data di *database*
	5	Sistem menampilkan data hasil pencarian
Extensions	Step	Branching Action
	4.1	Sistem tidak menemukan data cari di *database*
	4.2	Tidak ditemukan hasil

D. *Use Case Scenario* Hapus Barang

Deskripsi dari skenario *use case* hapus barang dapat dilihat pada Tabel 3.25 berikut.

Tabel 3.25 *Use case Scenario* Hapus Barang

Use case name	**Hapus Barang**	
Related Requirement	-	
Goal In Context	Menghapus data barang pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Hapus barang berhasil dan tampil *form* pengaturan barang	
Failed End Condition	Hapus barang gagal dan tampil pesan kesalahan	
Primary Actor	Admin	
Trigger	*User* memilih tombol hapus	
Include Cases	Pengolahan Data Barang	
Main Flow	Step	Action
	1	Sistem menampilkan *form* pengaturan barang
	2	*User* memililh data barang
	3	*User* memilih tombol hapus
	4	Sistem menampilkan pesan konfirmasi
	5	*User* memilih tombol Ok
	6	Sistem melakukan validasi hapus data
	7	Data berhasil dihapus dari *database*
Extensions	Step	Branching Action
	5.1	*User* memilih tombol Cancel
	5.2	Sistem membatalkan penghapusan data
	6.1	Sistem membatalkan penghapusan data
	6.2	Data gagal dihapus

3.1.2.9.5 Use Case Scenario Pengolahan Data Supplier

Use case scenario pengolahan data *supplier* menjelaskan interaksi antara aktor Admin dalam mengolah data barang di antaranya dengan *use case* tambah *supplier*, ubah, cari, dan hapus *supplier*. Dalam melakukan proses pengolahan data *supplier* terdapat aturan yang tercantum dalam Requirement A.6.

Requirement A.6
Data *supplier* terdiri dari nama *supplier*, no telp, dan alamat *supplier*. No telp maksimal 12 karakter dan unik serta alamat supplier harus bersifat unik.

A. *Use Case Scenario* Tambah *Supplier*

Deskripsi dari skenario *use case* tambah *supplier* dapat dilihat pada Tabel 3.26 berikut.

Tabel 3.26 *Use case Scenario* Tambah *Supplier*

Use case name	**Tambah *Supplier***	
Related Requirement	Requirement A.6	
Goal In Context	Mengelola tambah data *supplier* pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Tambah *supplier* berhasil dan tampil *form* lihat *supplier*	
Failed End Condition	Tambah *supplier* gagal dan tampil pesan kesalahan	
Primary Actor	Admin	
Trigger	*User* memilih tombol simpan	
Include Cases	Pengolahan Data *supplier*	
Main Flow	Step	Action
	1	Sistem menampilkan *form* pengaturan *supplier*
	2	*User* memilih menu tambah *supplier*
	3	Sistem menampilkan *form* tambah *supplier*
	4	*User* mengisi *field input*
	5	*User* memilih tombol simpan
	6	Sistem melakukan validasi data *input*
	7	Sistem menyimpan data ke *database*
	5	Data berhasil disimpan
Extensions	Step	Branching Action
	4.1	Data *input* tidak valid
	4.2	Data tidak tersimpan di *database*

B. *Use Case Scenario* Ubah *Supplier*

Deskripsi dari skenario *use case* ubah *supplier* dapat dilihat pada Tabel 3.27 berikut.

Tabel 3.27 *Use case Scenario* Ubah *Supplier*

Use case name	**Ubah *Supplier***	
Related Requirement	Requirement A.6	
Goal In Context	Mengelola ubah data *supplier* pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Ubah *supplier* berhasil dan tampil *form* lihat *supplier*	
Failed End Condition	Ubah *supplier* gagal dan tampil pesan kesalahan	
Primary Actor	Admin	
Trigger	*User* memilih tombol simpan	
Include Cases	Pengolahan Data *Supplier*	
Main Flow	Step	Action
	1	Sistem menampilkan *form* pengaturan *supplier*
	2	Memilih data *supplier* yang akan diubah
	3	*User* memilih menu ubah *supplier*
	4	Sistem menampilkan *form* ubah *supplier*
	4	*User* mengisi *field input*
	5	*User* memilih tombol simpan
	6	Sistem melakukan validasi data *input*
	7	Sistem menyimpan data ke *database*
	8	Data berhasil disimpan
Extensions	Step	Branching Action
	6.1	Data *input* tidak valid
	6.2	Data tidak tersimpan di *database*

C. *Use Case Scenario* Cari *Supplier*

Deskripsi dari skenario *use case* cari *supplier* dapat dilihat pada Tabel 3.28 berikut.

Tabel 3.28 *Use case Scenario* Cari *Supplier*

Use case name	**Cari *Supplier***	
Related Requirement	-	
Goal In Context	Mencari data *supplier* pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Menampilkan data hasil pencarian	
Failed End Condition	Pencarian data tidak berhasil dan sistem menampilkan pesan data tidak ditemukan	
Primary Actor	Admin	
Trigger	*User* menekan tombol Enter	
Include Cases	Pengolahan Data *Supplier*	
Main Flow	Step	Action
	1	Sistem menampilkan *form* pengaturan *supplier*
	2	*User* mengisi kolom filter pecarian
	3	*User* menekan tombol Enter
	4	Sistem mencari data di *database*
	5	Sistem menampilkan data hasil pencarian
Extensions	Step	Branching Action
	4.1	Sistem tidak menemukan data cari di *database*
	4.2	Tidak ditemukan hasil

D. *Use Case Scenario* Hapus *Supplier*

Deskripsi dari skenario *use case* hapus *supplier* dapat dilihat pada Tabel 3.29 berikut.

Tabel 3.29 *Use case Scenario* Hapus *Supplier*

Use case name	**Hapus *Supplier***	
Related Requirement	-	
Goal In Context	Menghapus data *supplier* pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Hapus *supplier* berhasil dan tampil *form* pengaturan *supplier*	
Failed End Condition	Hapus *supplier* gagal dan tampil pesan kesalahan	
Primary Actor	Admin	
Trigger	*User* memilih tombol hapus	
Include Cases	Pengolahan Data *Supplier*	
Main Flow	Step	Action
	1	Sistem menampilkan *form* pengaturan *supplier*
	2	*User* memililh data *supplier*
	3	*User* memilih tombol hapus
	4	Sistem menampilkan pesan konfirmasi
	5	*User* memilih tombol Ok
	6	Sistem melakukan validasi hapus data
	7	Data berhasil dihapus dari *database*
Extensions	Step	Branching Action
	5.1	*User* memilih tombol Cancel
	5.2	Sistem membatalkan penghapusan data
	6.1	Sistem membatalkan penghapusan data
	6.2	Data gagal dihapus

3.1.2.9.6 *Use Case Scenario* Pengolahan Data Pemesanan

Use case scenario pengolahan data pemesanan menjelaskan interaksi antara aktor Petugas dan Admin dalam mengolah data pemesanan di antaranya dengan *use case* tambah pemesanan, ubah, cari, dan hapus pemesanan. Dalam melakukan proses pengolahan data pemesanan terdapat aturan yang tercantum dalam Requirement A.8.

Requirement A.8
Data pemesanan terdiri dari no pemesanan yang otomatis dibuat oleh sistem, nama petugas, nama *supplier*, tanggal pemesanan, dan detail pemesanan, serta setiap *field input* tidak boleh kosong.

A. *Use Case Scenario* Tambah Pemesanan

Deskripsi dari skenario *use case* tambah pemesanan dapat dilihat pada Tabel 3.30 berikut.

Tabel 3.30 *Use case Scenario* Tambah Pemesanan

Use case name	**Tambah Pemesanan**	
Related Requirement	Requirement A.8	
Goal In Context	Mengelola tambah data pemesanan pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Tambah pemesanan berhasil dan tampil *form* lihat pemesanan	
Failed End Condition	Tambah pemesanan gagal dan tampil pesan kesalahan	
Primary Actor	Petugas	
Trigger	*User* memilih tombol simpan	
Include Cases	Pengolahan Data Pemesanan	
Main Flow	Step	Action
	1	Sistem menampilkan *form* tambah pemesanan
	2	*User* memilih tombol tambah barang
	3	Sistem menampilkan daftar barang
	4	*User* memilih barang yang akan dipesan
	5	Sistem menampilkan barang pada detil pemesanan
	6	*User* memilih tombol hapus barang detil pemesanan
	7	Sistem menampilkan pesan konfirmasi
	8	*User* memilih tombol Ok
	9	Sistem menghapus data barang pada detil pemesanan
	10	*User* mengisi *field input*

Tabel 3.31 *Use case Scenario* Tambah Pemesanan (Lanjutan)

Main Flow	Step	Action
	11	*User* memilih tombol simpan
	12	Sistem melakukan validasi data *input*
	13	Sistem menyimpan data ke *database*
	14	Data berhasil disimpan
Extensions	Step	Branching Action
	5.1	Data barang tidak dapat disisipkan pada detil pemesanan
	8.1	*User* memilih tombol Cancel
	8.2	Sistem membatalkan hapus barang detil pemesanan
	12.1	Data *input* tidak valid
	12.2	Data tidak tersimpan pada *database*

B. *Use Case Scenario* Ubah Pemesanan

Deskripsi dari skenario *use case* ubah pemesanan dapat dilihat pada Tabel 3.32 berikut.

Tabel 3.32 *Use case Scenario* Ubah Pemesanan

Use case name	**Ubah Pemesanan**	
Related Requirement	Requirement A.8	
Goal In Context	Mengelola ubah data pemesanan pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Ubah pemesanan berhasil dan tampil *form* lihat pemesanan	
Failed End Condition	Ubah pemesanan gagal dan tampil pesan kesalahan	
Primary Actor	Admin	
Trigger	*User* memilih tombol simpan	
Include Cases	Pengolahan Data Pemesanan	
Main Flow	Step	Action
	1	Sistem menampikan *form* pengaturan pemesanan
	2	*User* memilih data dan memilih tombol lihat
	3	Sistem menampilkan *form* lihat pemesanan
	4	*User* memilih tombol ubah pemesanan
	5	Sistem menampilkan *form* ubah pemesanan
	6	*User* mengisi *field input*
	7	*User* memilih tombol simpan
	8	Sistem melakukan validasi data *input*
	9	Sistem menyimpan data ke dalam *database*
	10	Data berhasil disimpan

Tabel 3.33 *Use case Scenario* Ubah Pemesanan (Lanjutan)

Extensions	Step	Branching Action
	4.1	Data *input* tidak valid
	4.2	Data tidak tersimpan pada *database*

C. *Use Case Scenario* Cari Pemesanan

Deskripsi dari skenario *use case* cari pemesanan dapat dilihat pada Tabel 3.4 berikut.

Tabel 3.34 *Use case Scenario* Cari Pemesanan

Use case name	**Cari Pemesanan**	
Related Requirement	-	
Goal In Context	Mencari data pemesanan pada sistem	
Precondition	*User* sudah berada pada *session login* dan berada pada *form* pengaturan pemesanan	
Successful End Condition	Menampilkan data hasil pencarian	
Failed End Condition	Pencarian data tidak berhasil dan sistem menampilkan pesan data tidak ditemukan	
Primary Actor	Admin	
Secondary Actor	Petugas	
Trigger	*User* menekan tombol Enter	
Include Cases	Pengolahan Data Pemesanan	
Main Flow	Step	Action
	1	Sistem menampilkan *form* pengaturan pemesanan
	2	*User* mengisi kolom filter pecarian
	3	*User* menekan tombol Enter
	4	Sistem mencari data di *database*
	5	Sistem menampilkan data hasil pencarian
	6	*User* memilih data dan memilih tombol lihat
	7	Sistem menampilkan data pemesanan pada *form* lihat pemesanan
Extensions	Step	Branching Action
	4.1	Sistem tidak menemukan data cari di *database*
	4.2	Tidak ditemukan hasil

D. *Use Case Scenario* Hapus Pemesanan

Deskripsi dari skenario *use case* hapus pemesanan dapat dilihat pada Tabel 3.35 berikut.

Tabel 3.35 *Use case Scenario* Hapus Pemesanan

Use case name	**Hapus Pemesanan**	
Related Requirement	-	
Goal In Context	Menghapus data pemesanan pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Penghapusan data pemesanan berhasil dan tampil *form* pengaturan pemesanan	
Failed End Condition	Penghapusan data pemesanan gagal dan tampil pesan kesalahan	
Primary Actor	Admin	
Trigger	*User* memilih tombol hapus	
Include Cases	Pengolahan Data Pemesanan	
Main Flow	Step	Action
	1	Sistem menampilkan *form* pengaturan pemesanan
	2	*User* memilih tombol lihat
	3	Sistem menampilkan *form* lihat pemesanan
	4	*User* memilih tombol hapus pemesanan
	5	Sistem menampilkan pesan konfirmasi
	6	*User* memilih tombol Ok
	7	Sistem melakukan validasi hapus data
	8	Sistem menghapus data dari *database*
	9	Data berhasil dihapus
Extensions	Step	Branching Action
	3.1	*User* memilih tombol Cancel
	3.2	Sistem membatalkan penghapusan data
	4.1	Data yang akan dihapus tidak valid
	4.2	Data gagal dihapus

3.1.2.9.7 *Use Case Scenario* Pengolahan Data Penerimaan

Use case scenario pengolahan data penerimaan menjelaskan interaksi antara aktor Admin atau Petugas dalam mengolah data penerimaan seperti *use case* tambah, ubah, cari dan hapus penerimaan. Dalam melakukan proses pengolahan data penerimaan terdapat aturan yang tercantum dalam Requirement A.9.

Requirement A.9
Pengguna diharus memilih data pemesanan yang akan dilakukan penerimaannya karena data penerimaan berkaitan dengan data pemesanan.

A. *Use Case Scenario* Tambah Penerimaan

Deskripsi dari skenario *use case* tambah penerimaan dapat dilihat pada Tabel 3.36 berikut.

Tabel 3.36 *Use case Scenario* Tambah Penerimaan

Use case name	**Tambah Penerimaan**	
Related Requirement	Requirement A.9	
Goal In Context	Mengelola tambah data penerimaan pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Tambah penerimaan berhasil dan tampil *form* lihat penerimaan	
Failed End Condition	Tambah penerimaan gagal dan tampil pesan kesalahan	
Primary Actor	Petugas	
Trigger	*User* memilih tombol simpan	
Include Cases	Pengolahan Data Penerimaan	
Main Flow	Step	Action
	1	Sistem menampilkan *form* pengaturan penerimaan
	2	*User* memilih menu tambah penerimaan
	3	Sistem menampilkan *form* tambah penerimaan
	4	*User* memilih tombol cari pemesanan
	3	Sistem menampilkan *list* data pemesanan
	4	*User* memilih data pemesanan
	5	Sistem menampilkan data detil pemesanan
	6	*User* memilih tombol pilih pada tabel detil pemesanan
	7	Sistem menampilkan barang pada detil penerimaan
	8	*User* memilih tombol hapus barang penerimaan
	9	Sistem menampilkan pesan konfirmasi
	10	*User* memilih tombol Ok

Tabel 3.37 *Use case Scenario* Tambah Penerimaan (Lanjutan)

Main Flow	Step	Action
	11	Sistem menghapus data barang pada detil penerimaan
	12	*User* mengisi *field input*
	13	*User* memilih tombol simpan
	14	Sistem melakukan validasi data *input*
	15	Sistem melakukan *update* stok barang
	16	Sistem melakukan *update* jumlah sisa pemesanan
	17	Sistem menyimpan data ke *database*
	18	Data berhasil disimpan
Extensions	Step	Branching Action
	6.1	Data barang tidak dapat disisipkan pada detil penerimaan
	10.1	*User* memilih tombol Cancel
	10.2	Sistem membatalkan penghapusan detil penerimaan
	14.1	Data *input* tidak valid
	14.2	Data tidak tersimpan ke dalam *database*

B. *Use Case Scenario* Ubah Penerimaan

Deskripsi dari skenario *use case* ubah penerimaan dapat dilihat pada Tabel 3.38 berikut.

Tabel 3.38 *Use case Scenario* Ubah Pemesanan

Use case name	**Ubah Pemesanan**	
Related Requirement	Requirement A.9	
Goal In Context	Mengelola ubah data penerimaan pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Ubah penerimaan berhasil dan tampil *form* lihat penerimaan	
Failed End Condition	Ubah penerimaan gagal dan tampil pesan kesalahan	
Primary Actor	Admin	
Trigger	*User* memilih tombol simpan	
Include Cases	Pengolahan Data Penerimaan	
Main Flow	Step	Action
	1	Sistem menampikan *form* pengaturan penerimaan
	2	*User* memilih data dan memilih tombol lihat
	3	Sistem menampilkan *form* lihat penerimaan
	4	*User* memilih tombol ubah penerimaan
	5	Sistem menampilkan *form* ubah penerimaan
	6	*User* mengisi *field input*
	7	*User* memilih tombol simpan
	8	Sistem melakukan validasi data *input*
	9	Sistem melakukan *update* stok barang
	10	Sistem melakukan *update* jumlah sisa pemesanan
	11	Sistem menyimpan data ke *database*
	12	Data berhasil disimpan
Extensions	Step	Branching Action
	8.1	Data *input* tidak valid
	8.2	Data tidak tersimpan pada *database*

C. *Use Case Scenario* Cari Penerimaan

Deskripsi dari skenario *use case* cari penerimaan dapat dilihat pada Tabel 3.39 berikut.

Tabel 3.39 *Use case Scenario* Cari Penerimaan

Use case name	**Cari Penerimaan**	
Related Requirement	-	
Goal In Context	Mencari data penerimaan pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Menampilkan data hasil pencarian	
Failed End Condition	Pencarian data tidak berhasil dan sistem menampilkan pesan data tidak ditemukan	
Primary Actor	Admin	
Secondary Actor	Petugas	
Trigger	*User* menekan tombol Enter	
Include Cases	Pengolahan Data Penerimaan	
Main Flow	Step	Action
	1	Sistem menampilkan *form* pengaturan penerimaan
	2	*User* mengisi kolom filter pecarian
	3	*User* menekan tombol Enter
	4	Sistem mencari data di *database*
	5	Sistem menampilkan data hasil pencarian
	6	*User* memilih data dan memilih tombol lihat
	7	Sistem menampilkan data penerimaan pada *form* lihat penerimaan
Extensions	Step	Branching Action
	4.1	Sistem tidak menemukan data cari di *database*
	4.2	Tidak ditemukan hasil

D. *Use Case Scenario* Hapus Penerimaan

Deskripsi dari skenario *use case* hapus penerimaan dapat dilihat pada Tabel 3.40 berikut.

Tabel 3.40 *Use case Scenario* Hapus Penerimaan

Use case name	**Hapus Penerimaan**	
Related Requirement	-	
Goal In Context	Menghapus data penerimaan pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Penghapusan data pemesanan berhasil dan tampil *form* pengaturan penerimaan	
Failed End Condition	-	
Primary Actor	Admin	
Trigger	*User* memilih tombol hapus	
Include Cases	Pengolahan Data Penerimaan	
Main Flow	Step	Action
	1	Sistem menampilkan *form* pengaturan penerimaan
	2	*User* memilih tombol lihat
	3	Sistem menampilkan *form* lihat penerimaan
	4	*User* memilih tombol hapus penerimaan
	5	Sistem menampilkan pesan konfirmasi
	6	*User* memilih tombol Ok
	7	Sistem melakukan *update* stok barang
	8	Sistem melakukan *update* jumlah sisa pemesanan
	9	Sistem menghapus data dari *database*
	10	Data berhasil dihapus
Extensions	Step	Branching Action
	3.1	*User* memilih tombol Cancel
	3.2	Sistem membatalkan penghapusan data

3.1.2.9.8 *Use Case Scenario* Pengolahan Data Pengeluaran

Use case scenario pengolahan data pengeluaran menjelaskan interaksi antara aktor Admin atau Petugas dalam mengolah data pengeluaran seperti *use case* tambah, ubah, cari dan hapus pengeluaran. Dalam melakukan proses pengolahan data pengeluaran terdapat aturan bisnis yang tercantum dalam Requirement A.10.

Requirement A.10
Data pengeluaran terdiri dari no faktur, no permintaan, nama petugas, nama kacab, nama petugas kacab, tanggal pengeluaran, dan detail pengeluaran

A. *Use Case Scenario* Tambah Pengeluaran

Deskripsi dari skenario *use case* tambah pengeluaran dapat dilihat pada Tabel 3.41 berikut.

Tabel 3.41 *Use case Scenario* Tambah Pengeluaran

Use case name	**Tambah Pengeluaran**	
Related Requirement	Requirement A.10	
Goal In Context	Mengelola tambah data pengeluaran pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Tambah pengeluaran berhasil dan tampil *form* lihat pengeluaran	
Failed End Condition	Tambah pengeluaran gagal dan tampil pesan kesalahan	
Primary Actor	Petugas	
Trigger	*User* memilih tombol simpan	
Include Cases	Pengolahan Data Pengeluaran	
Main Flow	Step	Action
	1	Sistem menampilkan *form* pengaturan pengeluaran
	2	*User* memilih menu tambah pengeluaran
	3	Sistem menampilkan *form* tambah pengeluaran
	4	*User* memilih tombol tambah barang
	5	Sistem menampilkan daftar barang
	6	*User* memilih barang yang akan dikeluarkan
	7	Sistem menampilkan barang pada detil pengeluaran
	8	*User* memilih tombol hapus barang detil pengeluaran
	9	Sistem menampilkan pesan konfirmasi
	10	*User* memilih tombol Ok

Tabel 3.42 *Use case Scenario* Tambah Penerimaan (Lanjutan)

Main Flow	Step	Action
	11	Sistem menghapus data barang pada detil pengeluaran
	12	*User* mengisi *field input*
	13	*User* memilih tombol simpan
	14	Sistem melakukan validasi data *input*
	15	Sistem melakukan *update* stok barang
	16	Sistem menyimpan data ke *database*
	17	Data berhasil disimpan
Extensions	Step	Branching Action
	6.1	Data barang tidak dapat disisipkan pada detil pengeluaran
	10.1	*User* memilih tombol Cancel
	10.2	Sistem membatalkan penghapusan detil pengeluaran
	14.1	Data *input* tidak valid
	14.2	Data tidak tersimpan ke dalam *database*

B. *Use Case Scenario* Ubah Pengeluaran

Deskripsi dari skenario *use case* ubah pengeluaran dapat dilihat pada Tabel 3.43 berikut.

Tabel 3.43 *Use case Scenario* Ubah Pengeluaran

Use case name	**Ubah Pengeluaran**	
Related Requirement	Requirement A.10	
Goal In Context	Mengelola ubah data pengeluaran pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Ubah pengeluaran berhasil dan tampil *form* lihat pengeluaran	
Failed End Condition	Ubah pengeluaran gagal dan tampil pesan kesalahan	
Primary Actor	Admin	
Trigger	*User* memilih tombol simpan	
Include Cases	Pengolahan Data Pengeluaran	
Main Flow	Step	Action
	1	Sistem menampikan *form* pengaturan pengeluaran
	2	*User* memilih data dan memilih tombol lihat
	3	Sistem menampilkan *form* lihat pengeluaran
	4	*User* memilih tombol ubah pengeluaran
	5	Sistem menampilkan *form* ubah pengeluaran
	6	*User* mengisi *field input*
	7	*User* memilih tombol simpan
	8	Sistem melakukan validasi data *input*
	9	Sistem melakukan *update* stok barang
	10	Sistem menyimpan data ke *database*
	11	Data berhasil disimpan
Extensions	Step	Branching Action
	8.1	Data *input* tidak valid
	8.2	Data tidak tersimpan pada *database*

C. *Use Case Scenario* Cari Pengeluaran

Deskripsi dari skenario *use case* cari pengeluaran dapat dilihat pada Tabel 3.44 berikut.

Tabel 3.44 *Use case Scenario* Cari Pengeluaran

Use case name	**Cari Penerimaan**	
Related Requirement	-	
Goal In Context	Mencari data pengeluaran pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Menampilkan data hasil pencarian	
Failed End Condition	Pencarian data tidak berhasil dan sistem menampilkan pesan data tidak ditemukan	
Primary Actor	Admin	
Secondary Actor	Petugas	
Trigger	*User* menekan tombol Enter	
Include Cases	Pengolahan Data Pengeluaran	
Main Flow	Step	Action
	1	Sistem menampilkan *form* pengaturan pengeluaran
	2	*User* mengisi kolom filter pecarian
	3	*User* menekan tombol Enter
	4	Sistem mencari data di *database*
	5	Sistem menampilkan data hasil pencarian
	6	*User* memilih data dan memilih tombol lihat
	7	Sistem menampilkan data pengeluaran pada *form* lihat pengeluaran
Extensions	Step	Branching Action
	4.1	Sistem tidak menemukan data cari di *database*
	4.2	Tidak ditemukan hasil

D. *Use Case Scenario* Hapus Pengeluaran

Deskripsi dari skenario *use case* hapus pengeluaran dapat dilihat Tabel 3.45 berikut.

Tabel 3.45 *Use case Scenario* Hapus Pengeluaran

Use case name	**Hapus Pengeluaran**	
Related Requirement	-	
Goal In Context	Menghapus data pengeluaran pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Penghapusan data pengeluaran berhasil dan tampil *form* pengaturan pengeluaran	
Failed End Condition	-	
Primary Actor	Admin	
Trigger	*User* memilih tombol hapus	
Include Cases	Pengolahan Data Pengeluaran	
Main Flow	Step	Action
	1	Sistem menampilkan *form* pengaturan pengeluaran
	2	*User* memilih tombol lihat
	3	Sistem menampilkan *form* lihat pengeluaran
	4	*User* memilih tombol hapus pengeluaran
	5	Sistem menampilkan pesan konfirmasi
	6	*User* memilih tombol Ok
	7	Sistem melakukan *update* stok barang
	8	Sistem menghapus data dari *database*
	9	Data berhasil dihapus
Extensions	Step	Branching Action
	6.1	*User* memilih tombol Cancel
	6.2	Sistem membatalkan penghapusan data

3.1.2.9.9 *Use Case Scenario* Pengolahan Laporan

Use case scenario pengolahan laporan menjelaskan interaksi antara aktor Petugas dalam mengolah laporan di antaranyadengan *use case* laporan pemesanan, penerimaan, dan laporan pengeluaran.

A. *Use Case Scenario* Laporan Pemesanan

Deskripsi dari skenario *use case* laporan pemesanan dapat dilihat pada Tabel 3.46 berikut.

Tabel 3.46 *Use Case Scenario* Laporan Pemesanan

Use case name	**Laporan Pemesanan**	
Related Requirement	-	
Goal In Context	Mengelola laporan pemesanan pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Berhasil mencari data laporan pemesanan dan mencetak laporan	
Failed End Condition	Pencarian data laporan pemesanan dan mencetak tidak berhasil	
Primary Actor	Petugas	
Trigger	*User* memilih tombol cari	
Include Cases	Pengolahan Laporan	
Main Flow	Step	Action
	1	*User* menentukan tanggal pencarian data pemesanan
	2	*User* memilih tombol cari barang
	3	Sistem menampilkan daftar/*list* barang
	4	*User* memilih barang yang akan dicari data pemesanannya
	5	*User* menentukan kategori barang yang akan dicari data pemesananya
	6	*User* memilih tombol cari
	7	Sistem melakukan pencarian data di *database*
	8	Sistem menampilkan hasil pencarian
	9	*User* memilih tombol cetak
	10	Sistem menampilkan halaman *print preview* dan pesan dialog *printer*
Extensions	Step	Branching Action
	7.1	Sistem tidak berhasil melakukan pencarian data
	7.2	Hasil pencarian tidak ada
	7.3	Cetak laporan tidak bisa diproses

B. *Use Case Scenario* Laporan Penerimaan

Deskripsi dari skenario *use case* laporan penerimaan dapat dilihat pada Tabel 3.47.

Tabel 3.47 *Use Case Scenario* Laporan Penerimaan

Use case name	**Laporan Penerimaan**	
Related Requirement	-	
Goal In Context	Mengelola laporan penerimaan pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Berhasil mencari data laporan penerimaan dan mencetak laporan	
Failed End Condition	Pencarian data laporan penerimaan dan mencetak tidak berhasil	
Primary Actor	Petugas	
Trigger	*User* memilih tombol cari	
Include Cases	Pengolahan Laporan	
Main Flow	Step	Action
	1	*User* menentukan tanggal pencarian data penerimaan
	2	*User* memilih tombol cari barang
	3	Sistem menampilkan daftar/*list* barang
	4	*User* memilih barang yang akan dicari data penerimaannya
	5	*User* menentukan kategori barang yang akan dicari data penerimaannya
	6	*User* memilih tombol cari
	7	Sistem melakukan pencarian data di *database*
	8	Sistem menampilkan hasil pencarian
	9	*User* memilih tombol cetak
	10	Sistem menampilkan halaman *print preview* dan pesan dialog *printer*
Extensions	Step	Branching Action
	7.1	Sistem tidak berhasil melakukan pencarian data
	7.2	Hasil pencarian tidak ada
	7.3	Cetak laporan tidak dapat diproses

C. *Use Case Scenario* Laporan Pengeluaran

Deskripsi dari skenario *use case* laporan pengeluaran dapat dilihat pada Tabel 3.48.

Tabel 3.48 *Use Case Scenario* Laporan Pengeluaran

Use case name	**Laporan Pengeluaran**	
Related Requirement	-	
Goal In Context	Mengelola laporan pengeluaran pada sistem	
Precondition	*User* sudah berada pada *session login*	
Successful End Condition	Berhasil mencari data laporan pengeluaran dan mencetak laporan	
Failed End Condition	Pencarian data laporan pengeluaran dan cetak laporan tidak berhasil	
Primary Actor	Petugas	
Trigger	*User* memilih tombol cari	
Include Cases	Pengolahan Laporan	
Main Flow	Step	Action
	1	*User* menentukan tanggal pencarian data pengeluaran
	2	*User* memilih tombol cari barang
	3	Sistem menampilkan daftar/*list* barang
	4	*User* memilih barang yang akan dicari data pengeluarannya
	5	*User* menentukan kategori barang yang akan dicari data pengeluarannya
	6	*User* memilih tombol cari
	7	Sistem melakukan pencarian data di *database*
	8	Sistem menampilkan hasil pencarian
	9	*User* memilih tombol cetak
	10	Sistem menampilkan halaman *print preview* dan pesan dialog *printer*
Extensions	Step	Branching Action
	7.1	Sistem tidak berhasil melakukan pencarian data
	7.2	Hasil pencarian tidak ada
	7.3	Cetak laporan tidak dapat diproses

3.1.2.9.10 *Use Case Scenario* Perhitungan Rencana Inventori

Use case scenario perhitungan rencana inventori menjelaskan interaksi antara aktor Petugas dengan *use case* Pengolahan Rencana Inventori. *Use case scenario* perhitungan rencana inventori dijelaskan pada Tabel 3.49.

Tabel 3.49 *Use case Scenario* Perhitungan Rencana Inventori

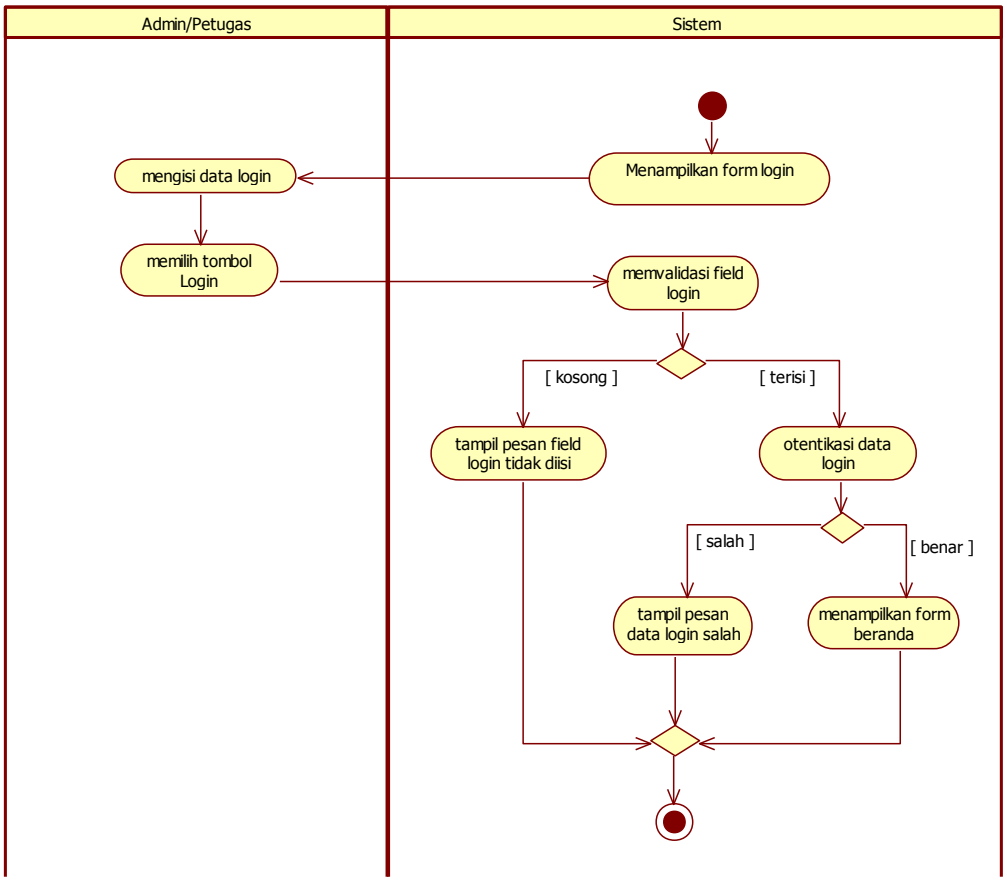
Use case name	**Perhitungan Rencana Inventori**	
Related Requirement	-	
Goal In Context	Mencari jumlah rencana inventori	
Precondition	Hanya *user* dengan kategori Admin yang diotorisasi melakukan pengolahan rencana inventori	
Successful End Condition	Tampil hasil perhitungan rencana inventori	
Failed End Condition	Tampil pesan kesalahan pencarian rencana inventori	
Primary Actor	Admin	
Trigger	*User* memilih tombol rencana	
Include Cases	*Login*	
Main Flow	Step	Action
	1	Sistem menampilkan *form* rencana inventori
	2	*User* memilih tombol cari barang
	3	Sistem menampilkan daftar barang
	4	*User* memilih data barang
	5	Sistem menampilkan nama barang
	6	*User* mengisi *field input* kriteria rencana inventori
	7	*User* memilih tombol Rencana
	8	Sistem melakukan validasi data *input*
	9	Sistem melakukan perhitungan
	10	Sistem menampilkan data pengeluaran dan hasil perhitungan
Extensions	Step	Branching Action
	5.1	Data input tidak valid
	5.2	Sistem tidak dapat melanjutkan proses perhitungan

3.1.2.10 *Activity Diagram*

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

3.1.2.10.1 *Activity Diagram Login*

Interaksi antara aktor Admin atau Petugas dengan sistem dalam proses *login* dijelaskan dalam *activity diagram* pada Gambar 3.6.



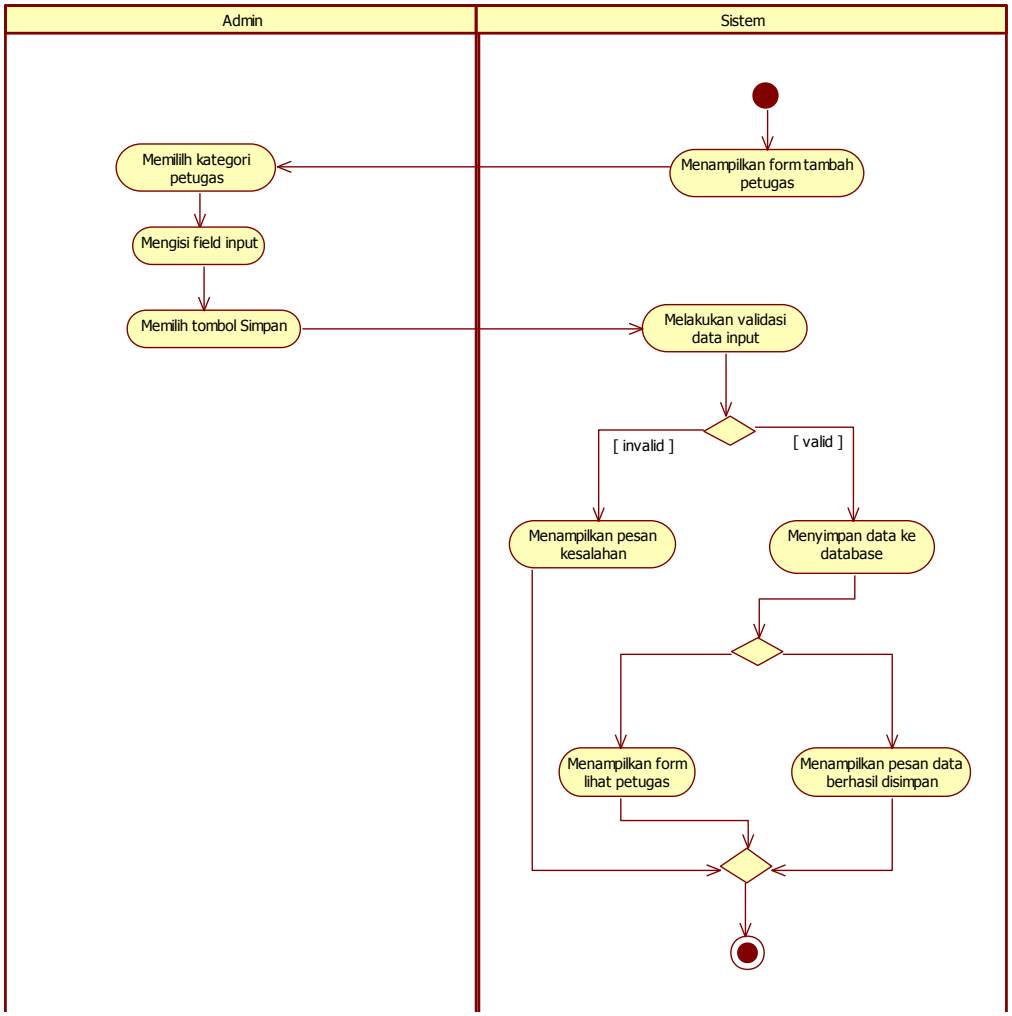
Gambar 3.6 *Activity Diagram Login*

3.1.2.10.2 *Activity Diagram* Pengolahan Data Petugas

Activity diagram pengolahan data petugas menjelaskan tentang aliran kerja aktor Admin dalam melakukan pengolahan data petugas yang digambarkan pada *activity diagram* tambah, ubah, dan hapus data petugas. Berikut penjelasan dari masing-masing *activity diagram* tersebut.

1. *Activity Diagram* Tambah Petugas

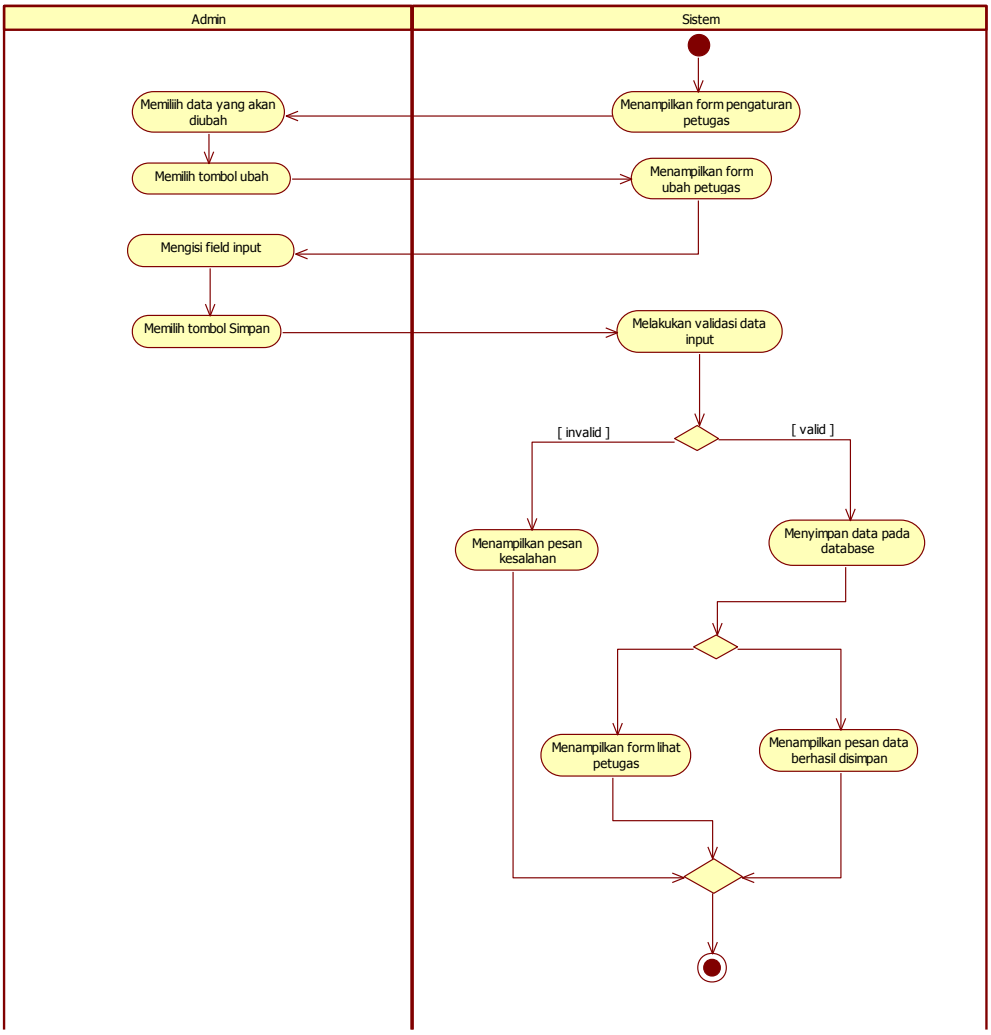
Interaksi antara aktor Admin dengan sistem dalam proses tambah data petugas dijelaskan dalam *activity diagram* pada Gambar 3.7.



Gambar 3.7 *Activity Diagram* Tambah Data Petugas

2. *Activity Diagram* Ubah Petugas

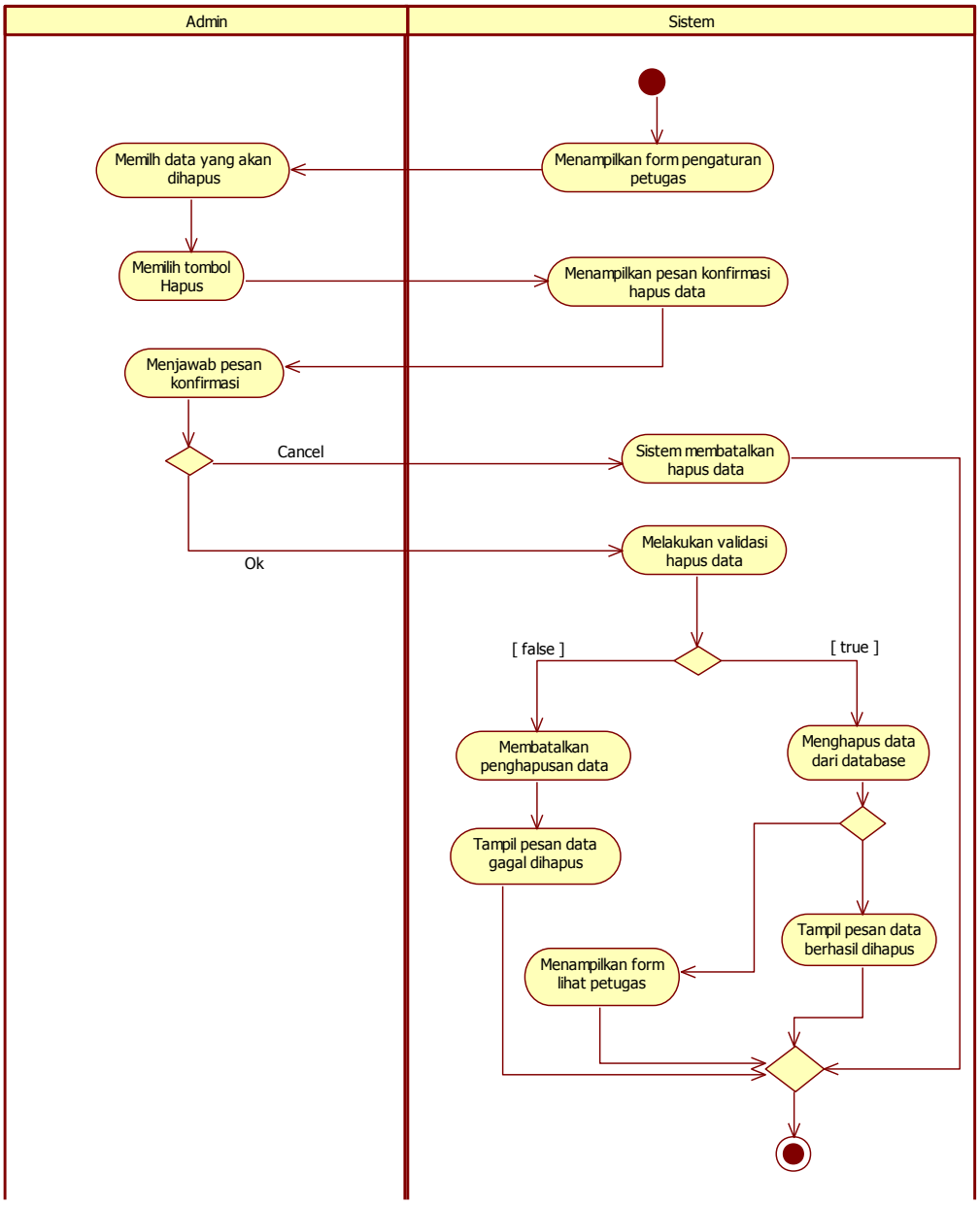
Interaksi antara aktor Admin atau petugas dengan sistem dalam proses ubah data petugas dijelaskan dalam *activity diagram* pada Gambar 3.8.



Gambar 3.8 *Activity Diagram* Ubah Petugas

3. *Activity Diagram* Hapus Petugas

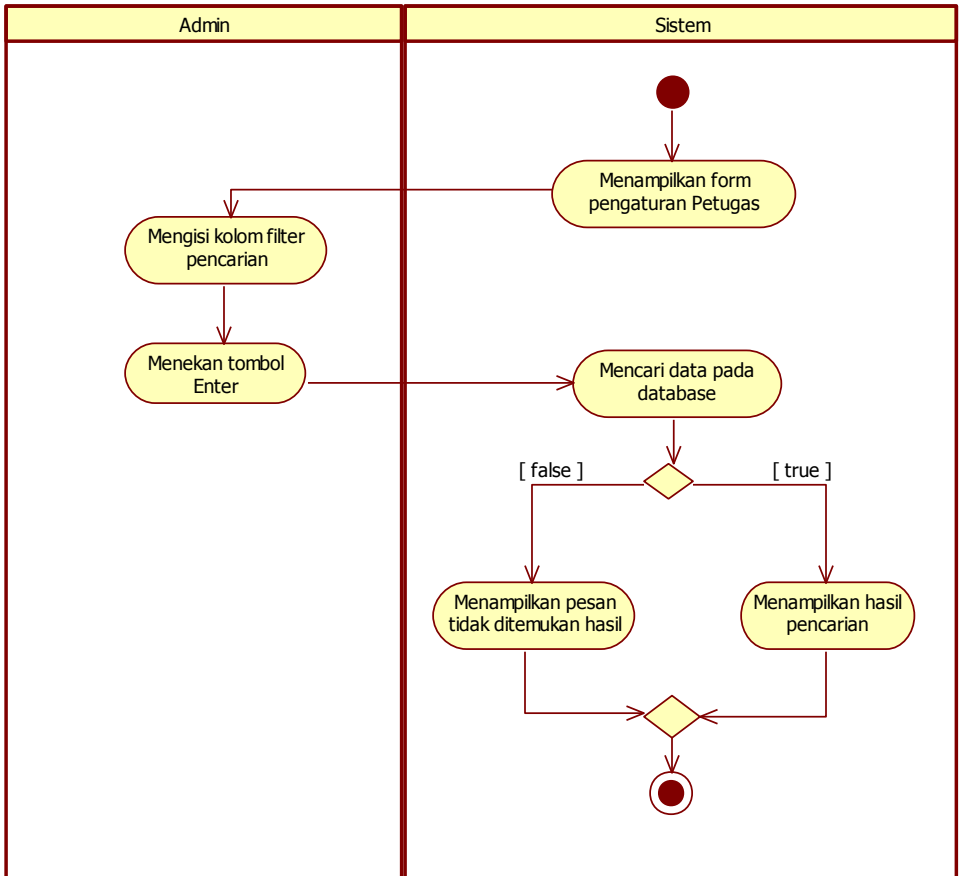
Interaksi antara aktor Admin atau petugas dengan sistem dalam proses hapus data petugas dijelaskan dalam *activity diagram* pada Gambar 3.9.



Gambar 3.9 *Activity Diagram* Hapus Petugas

4. *Activity Diagram* Cari Petugas

Interaksi antara aktor Admin dengan sistem dalam proses cari data petugas dijelaskan dalam *activity diagram* pada Gambar 3.10.



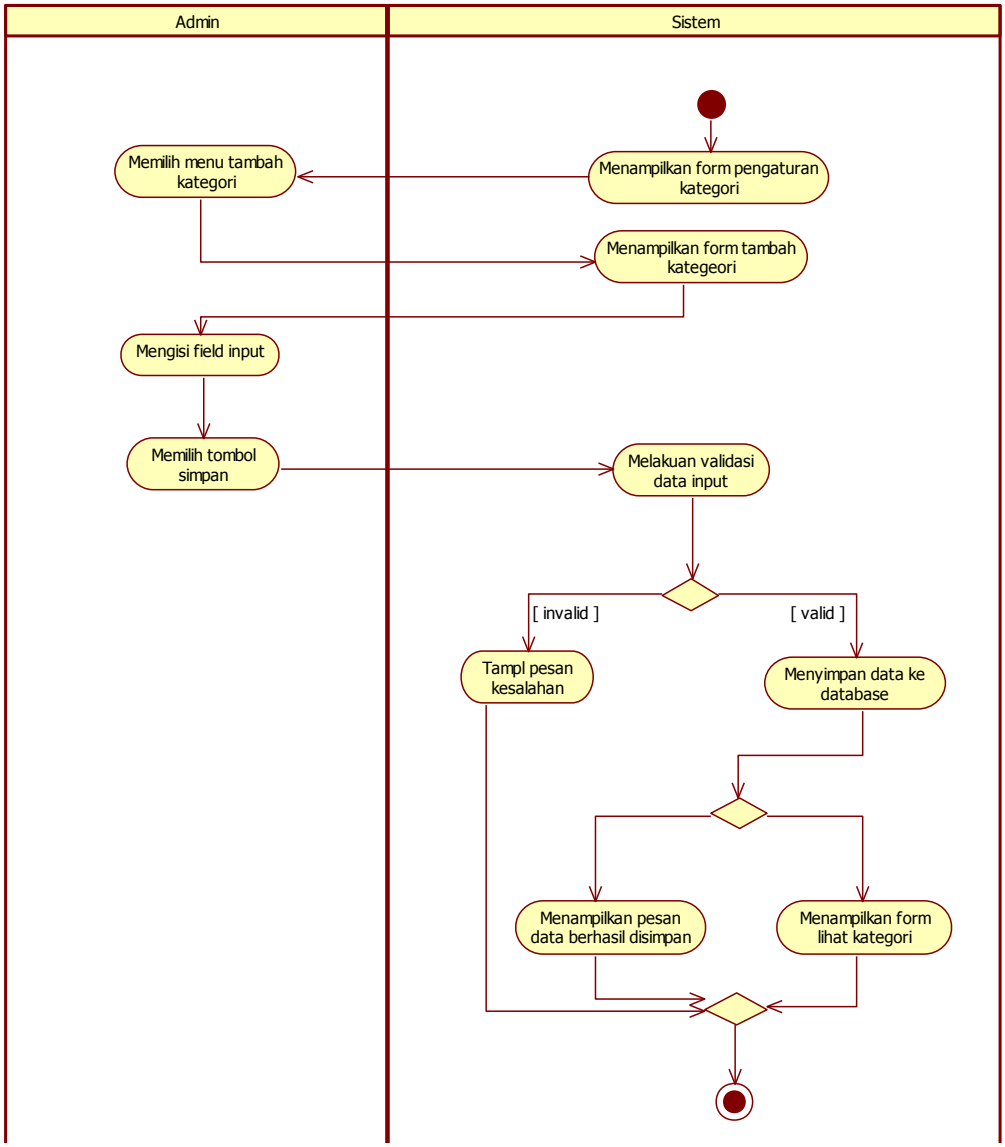
Gambar 3.10 *Activity Diagram* Cari Petugas

3.1.2.10.3 *Activity Diagram* Pengolahan Data Kategori

Activity diagram pengolahan data kategori menjelaskan tentang aliran kerja aktor Admin dalam melakukan pengolahan data kategori yang digambarkan pada *activity diagram* tambah, ubah, dan hapus data kategori.

1. *Activity Diagram* Tambah Kategori

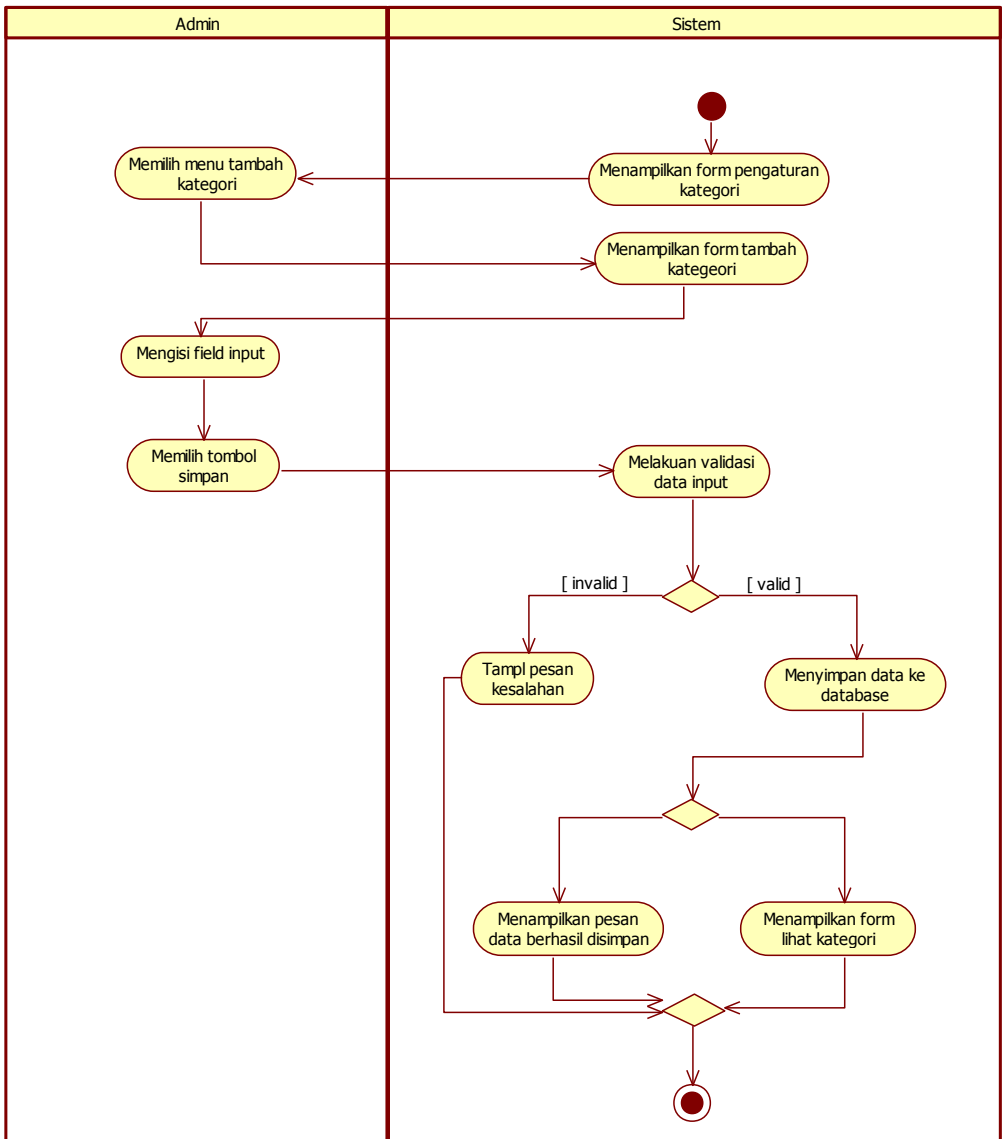
Interaksi antara aktor Admin dengan sistem dalam proses tambah data kategori dijelaskan dalam *activity diagram* pada Gambar 3.11.



Gambar 3.11 *Activity Diagram* Tambah Kategori

2. *Activity Diagram* Ubah Kategori

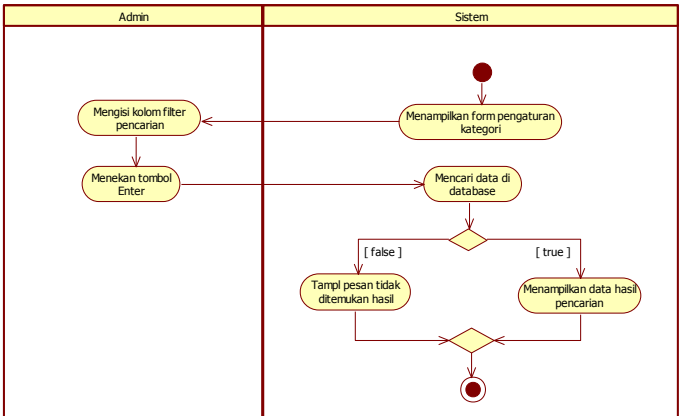
Interaksi antara aktor Admin dengan sistem dalam proses ubah data kategori dijelaskan dalam *activity diagram* pada Gambar 3.12.



Gambar 3.12 *Activity Diagram* Ubah Kategori

3. *Activity Diagram* Cari Kategori

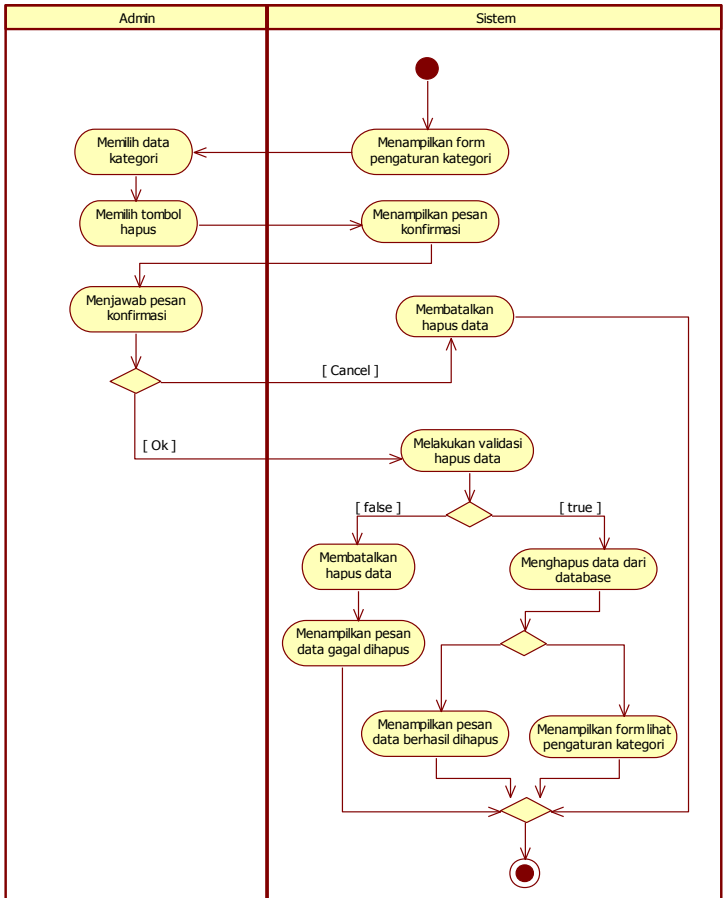
Interaksi antara aktor Admin dengan sistem dalam proses cari data kategori dijelaskan dalam *activity diagram* pada Gambar 3.13.



Gambar 3.13 *Activity Diagram* Cari Kategori

4. *Activity Diagram* Hapus Kategori

Interaksi antara aktor Admin dengan sistem dalam proses hapus data kategori dijelaskan dalam *activity diagram* pada Gambar 3.14.



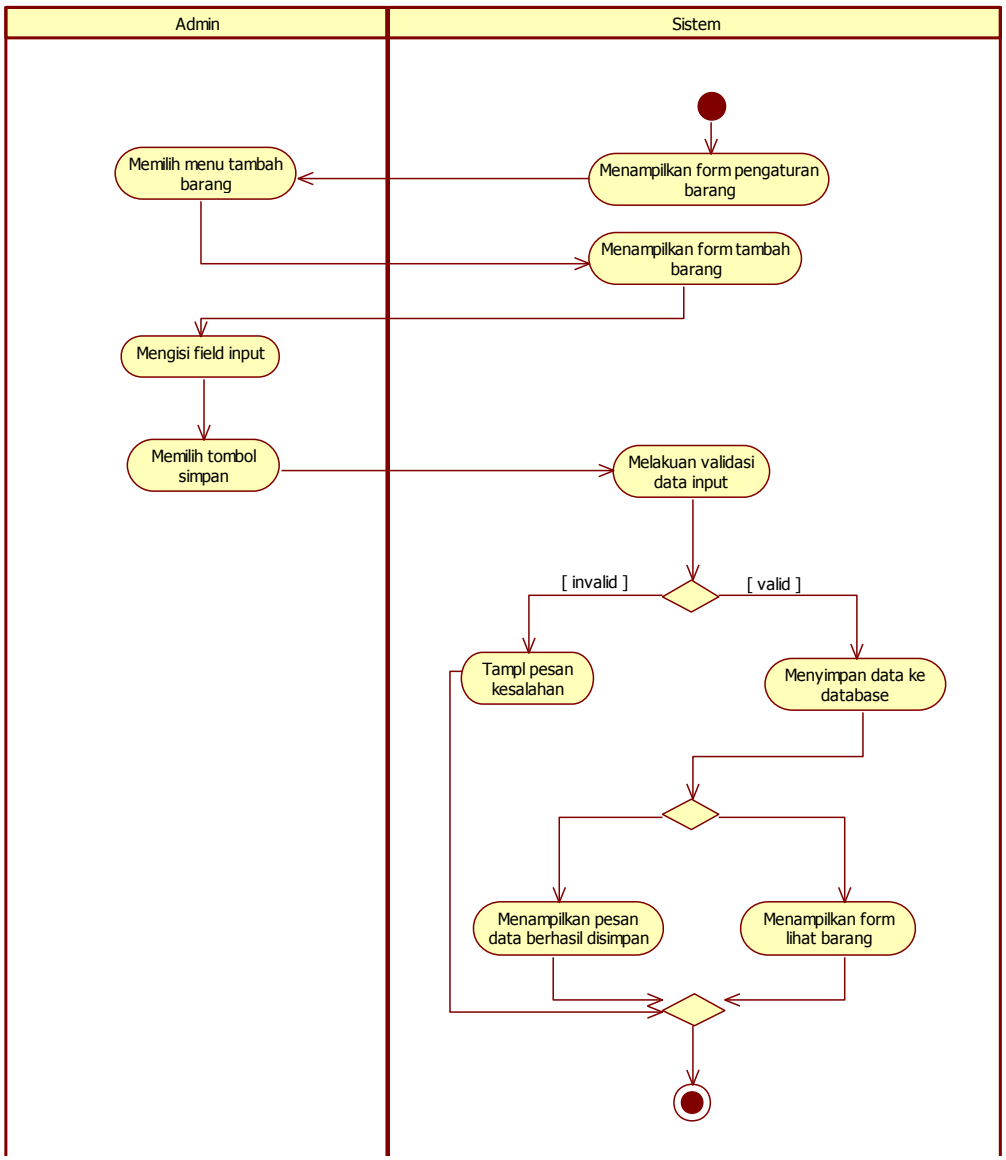
Gambar 3.14 *Activity Diagram* Hapus Kategori

3.1.2.10.4 *Activity Diagram* Pengolahan Data Barang

Activity diagram pengolahan data barang menjelaskan tentang aliran kerja aktor Admin dalam melakukan pengolahan data barang yang digambarkan pada *activity diagram* tambah, ubah, dan hapus data barang.

1. *Activity Diagram* Tambah Barang

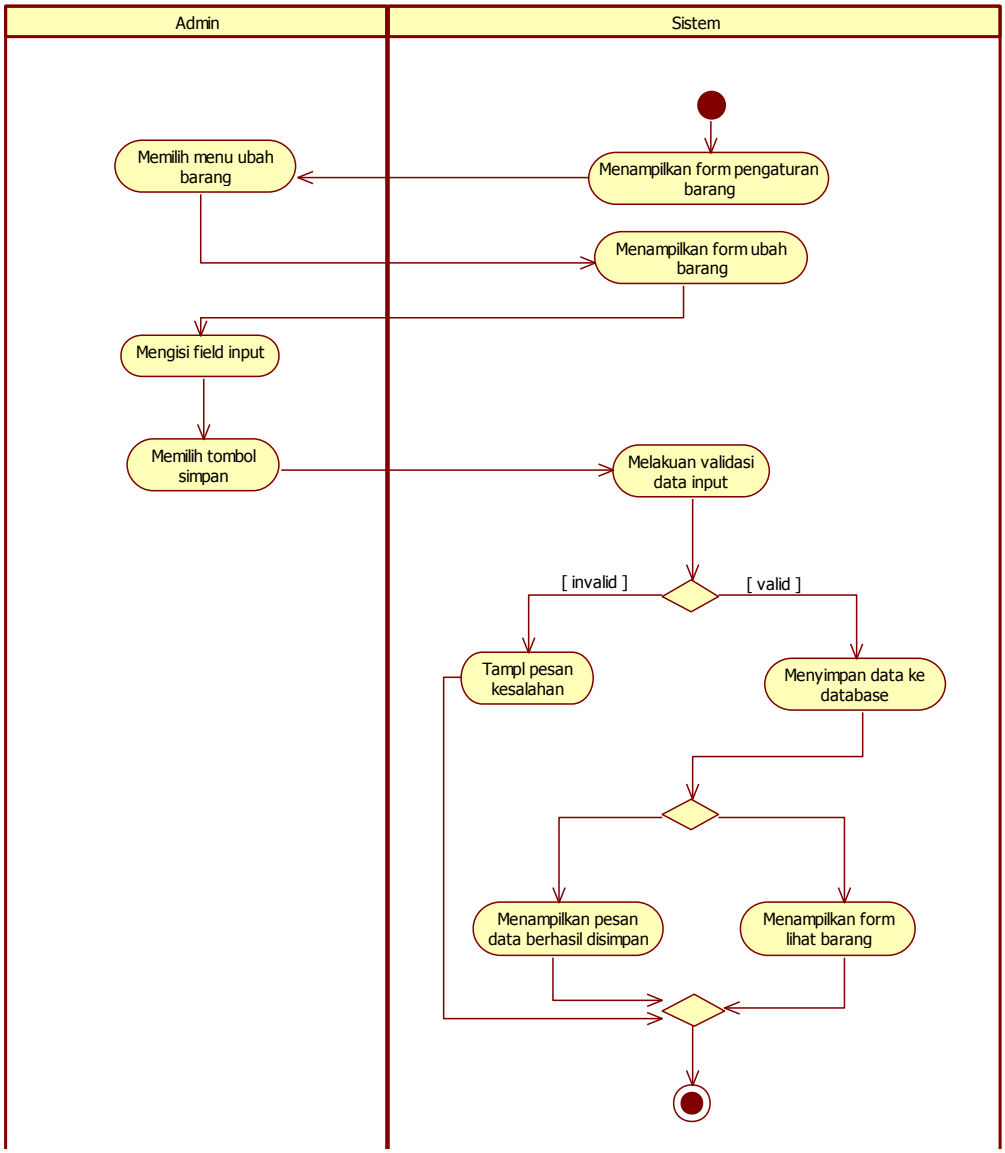
Interaksi antara aktor Admin dengan sistem dalam proses tambah data barang dijelaskan dalam *activity diagram* pada Gambar 3.15.



Gambar 3.15 *Activity Diagram* Tambah Barang

2. *Activity Diagram* Ubah Barang

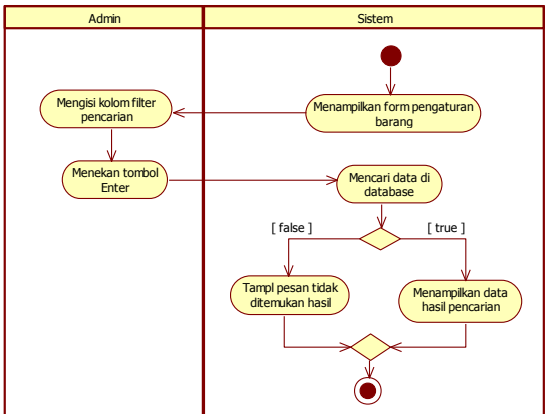
Interaksi antara aktor Admin dengan sistem dalam proses ubah data barang dijelaskan dalam *activity diagram* pada Gambar 3.16.



Gambar 3.16 *Activity Diagram* Ubah Barang

3. *Activity Diagram* Cari Barang

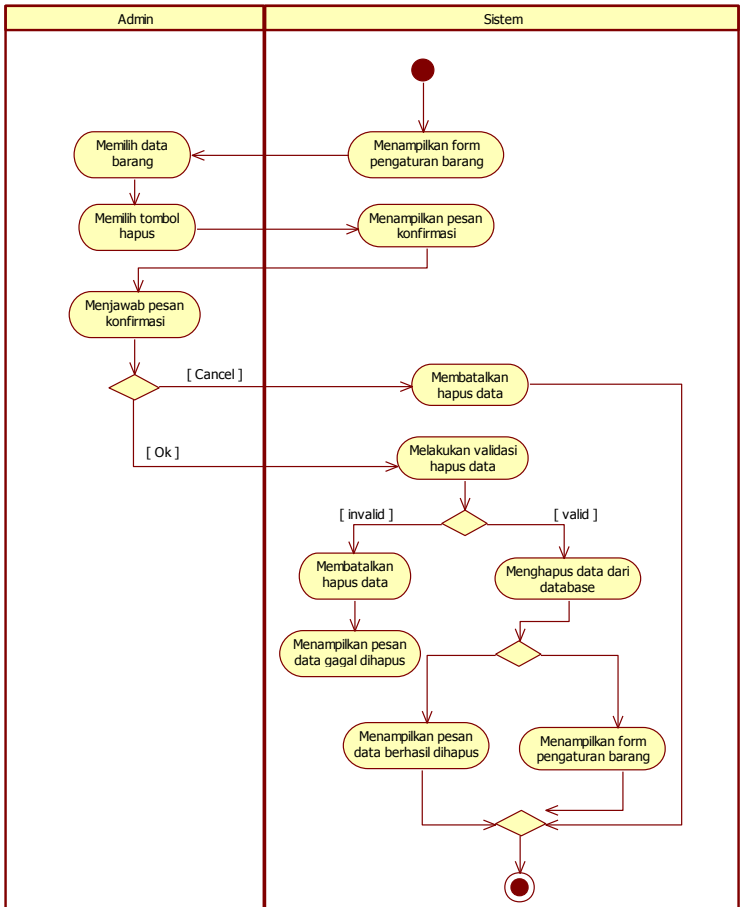
Interaksi antara aktor Admin dengan sistem dalam proses cari data barang dijelaskan dalam *activity diagram* pada Gambar 3.17.



Gambar 3.17 *Activity Diagram* Cari Barang

4. *Activity Diagram* Hapus Barang

Interaksi antara aktor Admin dengan sistem dalam proses hapus data barang dijelaskan dalam *activity diagram* pada Gambar 3.18.



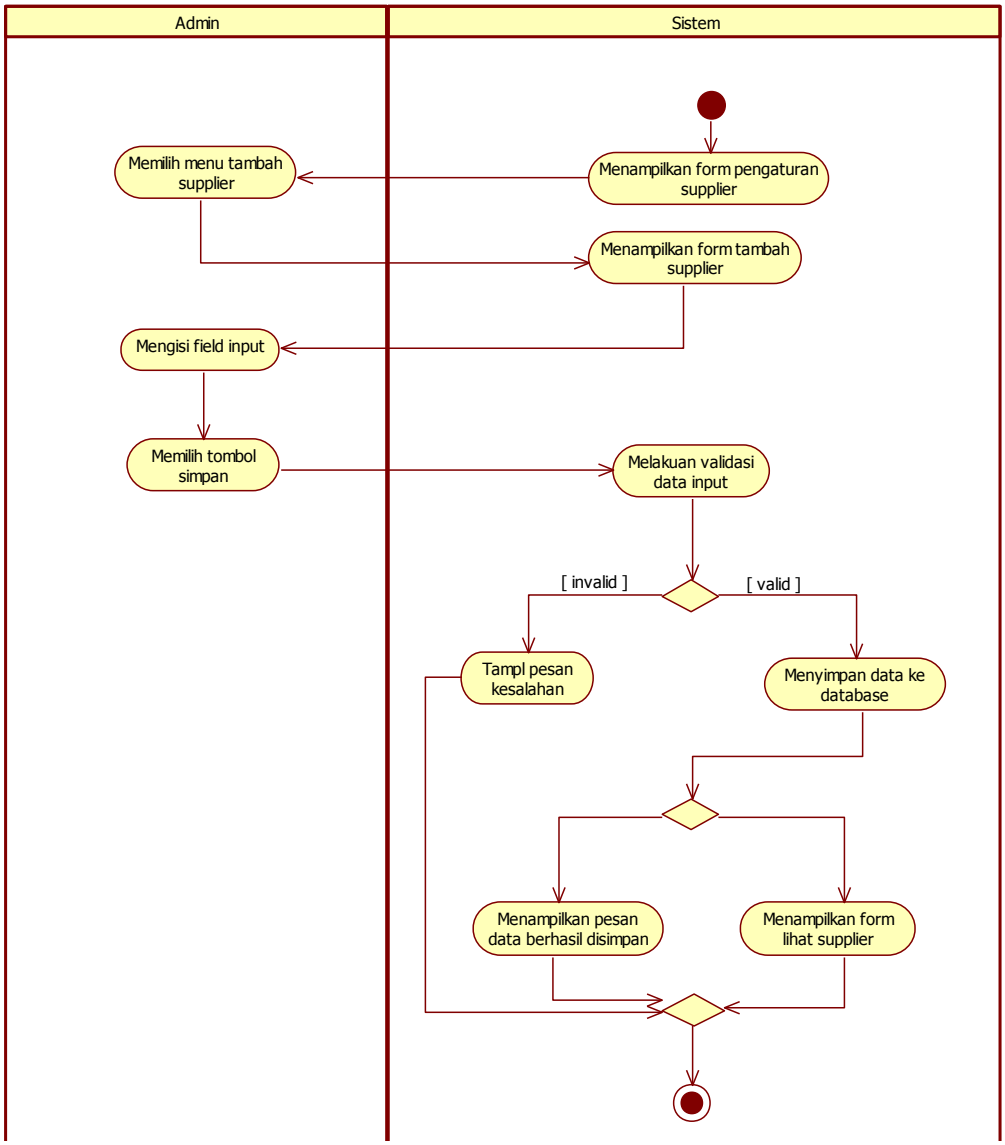
Gambar 3.18 *Activity Diagram* Hapus Barang

3.1.2.10.5 *Activity Diagram* Pengolahan Data *Supplier*

Activity diagram pengolahan data *supplier* menjelaskan tentang aliran kerja aktor Admin dalam melakukan pengolahan data *supplier* yang digambarkan pada *activity diagram* tambah, ubah, dan hapus data *supplier*.

1. *Activity Diagram* Tambah *Supplier*

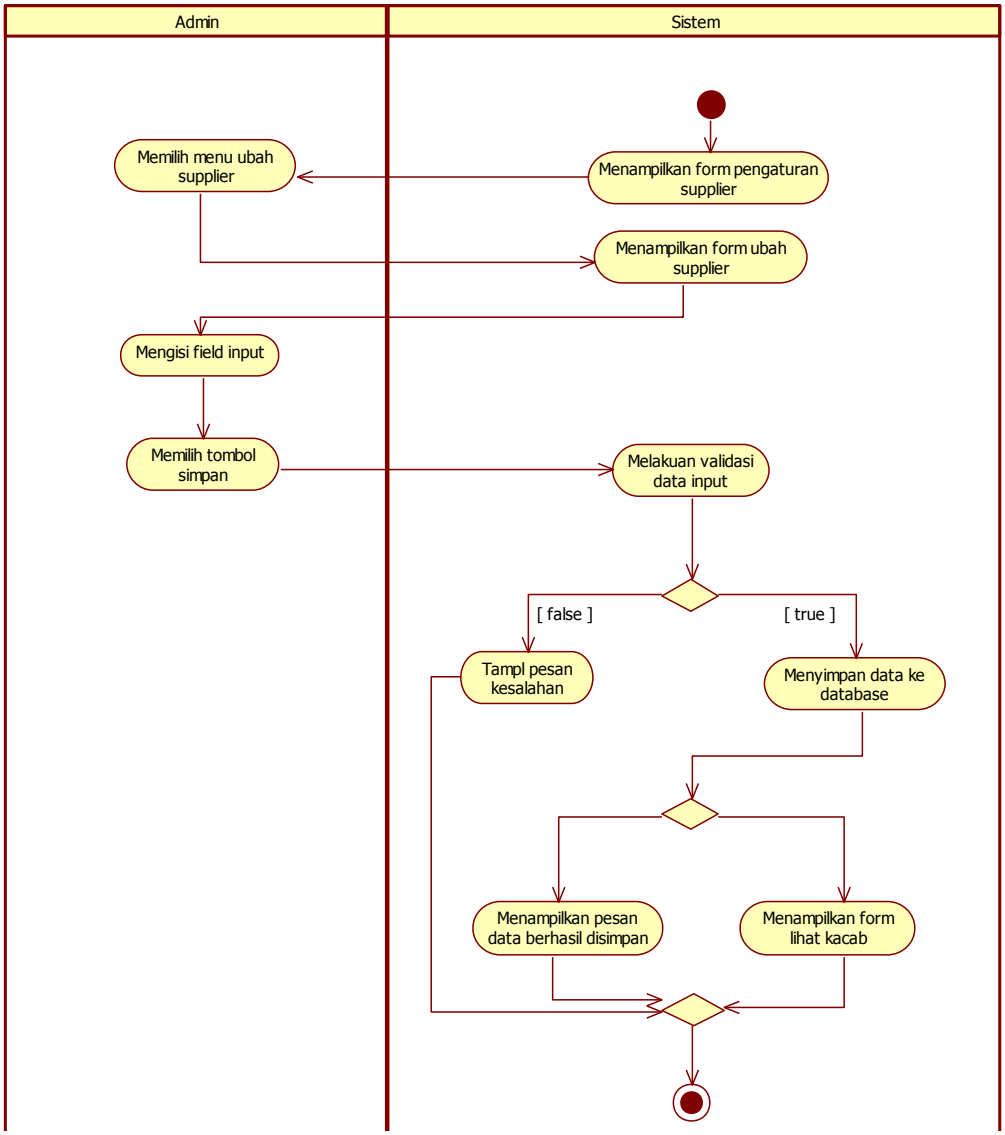
Interaksi antara aktor Admin dengan sistem dalam proses tambah data *supplier* dijelaskan dalam *activity diagram* pada Gambar 3.19.



Gambar 3.19 *Activity Diagram* Tambah *Supplier*

2. *Activity Diagram* Ubah *Supplier*

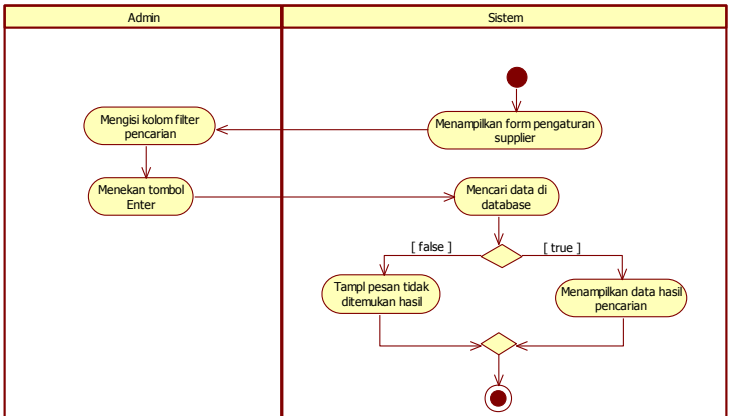
Interaksi antara aktor Admin dengan sistem dalam proses ubah data *supplier* dijelaskan dalam *activity diagram* pada Gambar 3.20.



Gambar 3.20 *Activity Diagram* Ubah *Supplier*

3. *Activity Diagram* Cari *Supplier*

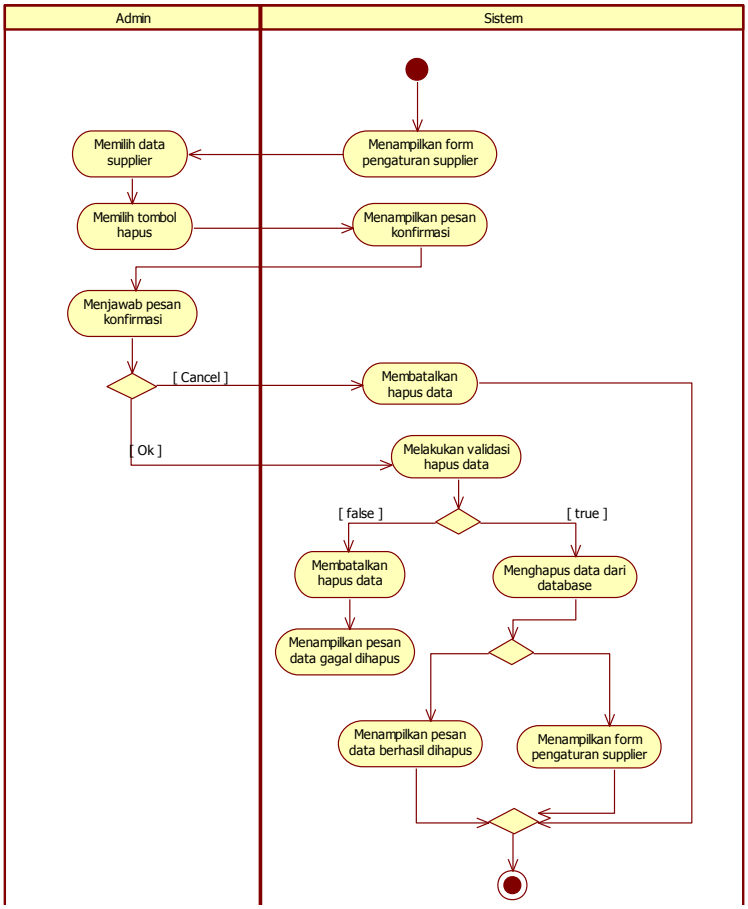
Interaksi antara aktor Admin dengan sistem dalam proses cari data *supplier* dijelaskan dalam *activity diagram* pada Gambar 3.21.



Gambar 3.21 *Activity Diagram* Hapus *Supplier*

4. *Activity Diagram* Hapus *Supplier*

Interaksi antara aktor Admin dengan sistem dalam proses hapus data *supplier* dijelaskan pada dalam *activity diagram* pada Gambar 3.22.



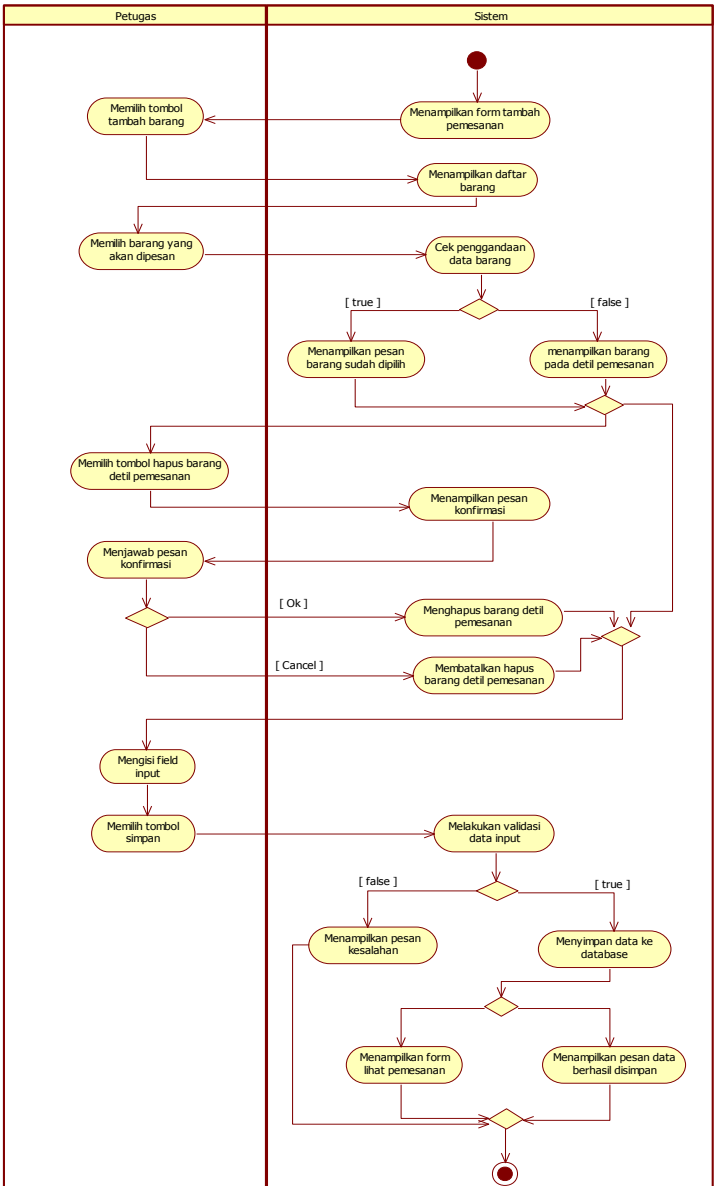
Gambar 3.22 Activity Diagram Hapus *Supplier*

3.1.2.10.6 *Activity Diagram* Pengolahan Data Pemesanan

Activity diagram pengolahan data pemesanan menjelaskan tentang aliran kerja aktor Admin atau Petugas dalam melakukan pengolahan data pemesanan yang digambarkan pada *activity diagram* tambah, ubah, dan hapus data pemesanan.

1. *Activity Diagram* Tambah Pemesanan

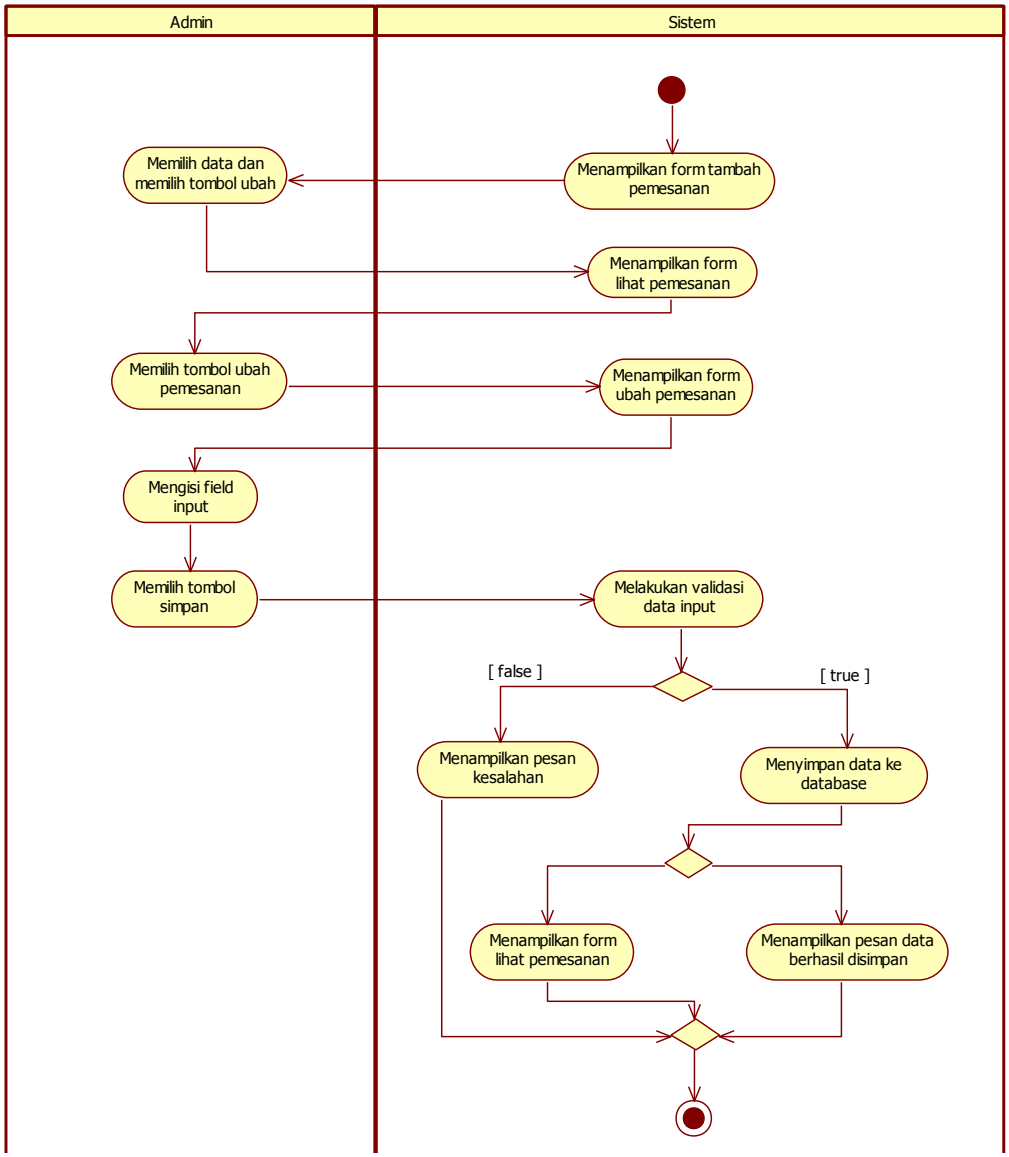
Interaksi antara aktor Petugas dengan sistem dalam proses tambah data pemesanan dijelaskan pada dalam *activity diagram* pada Gambar 3.23.



Gambar 3.23 *Activity Diagram* Tambah Pemesanan

2. *Activity Diagram* Ubah Pemesanan

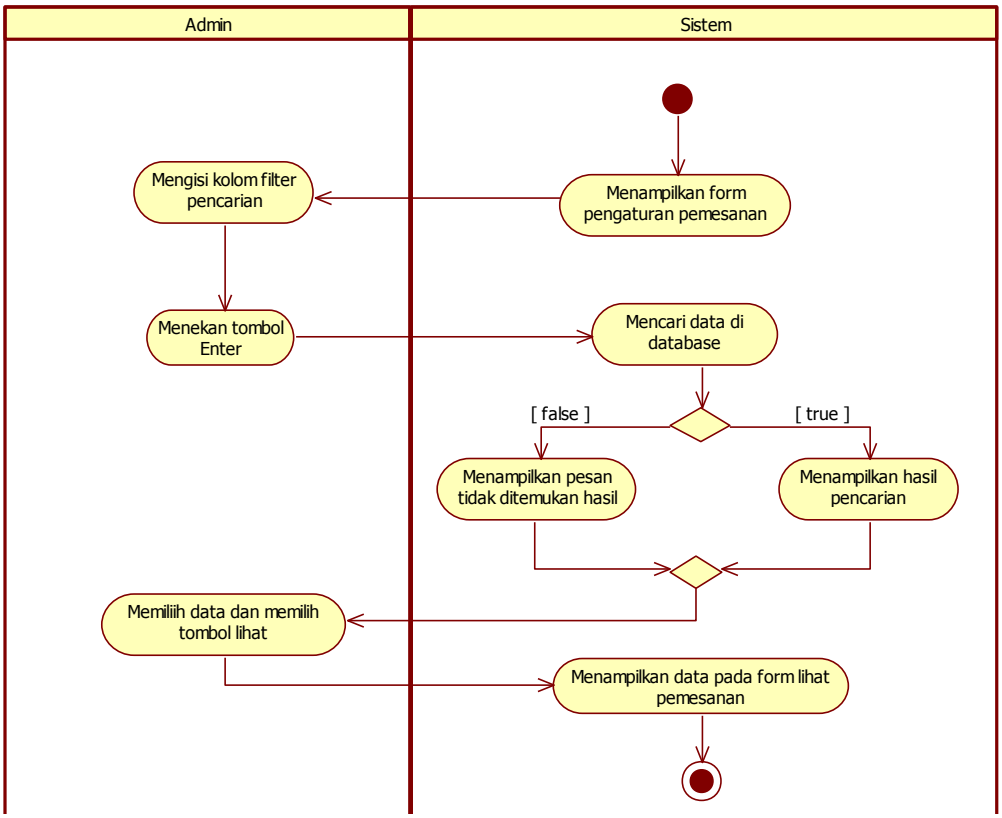
Interaksi antara aktor Admin dengan sistem dalam proses ubah data pemesanan dijelaskan dalam *activity diagram* pada Gambar 3.24.



Gambar 3.24 *Activity Diagram* Ubah Pemesanan

3. *Activity Diagram* Cari Pemesanan

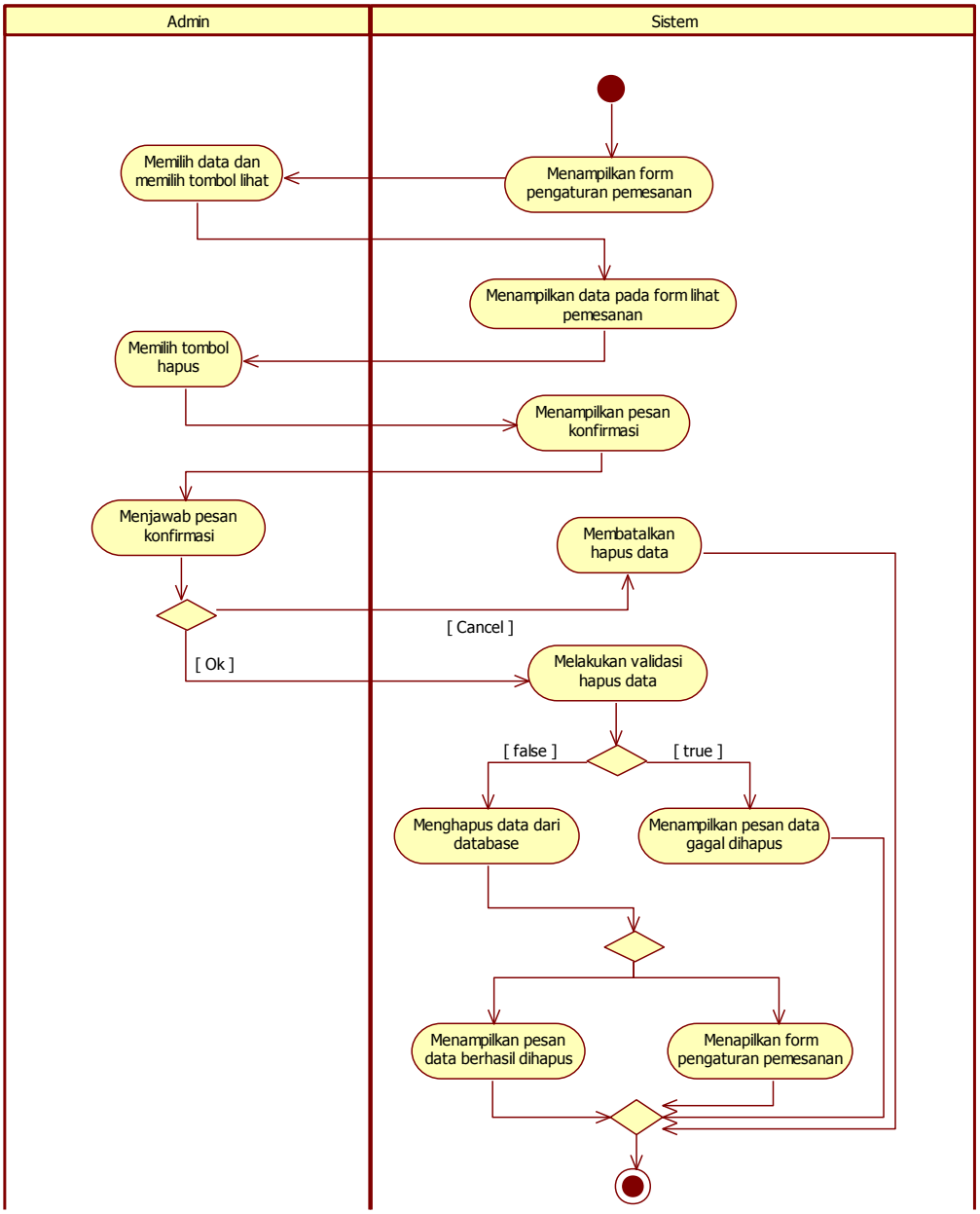
Interaksi antara aktor Admin dengan sistem dalam proses cari data pemesanan dijelaskan dalam *activity diagram* pada Gambar 3.25.



Gambar 3.25 *Activity Diagram* Cari Pemesanan

4. *Activity Diagram* Hapus Pemesanan

Interaksi antara aktor Admin dengan sistem dalam proses hapus data pemesanan dijelaskan dalam *activity diagram* pada Gambar 3.26.



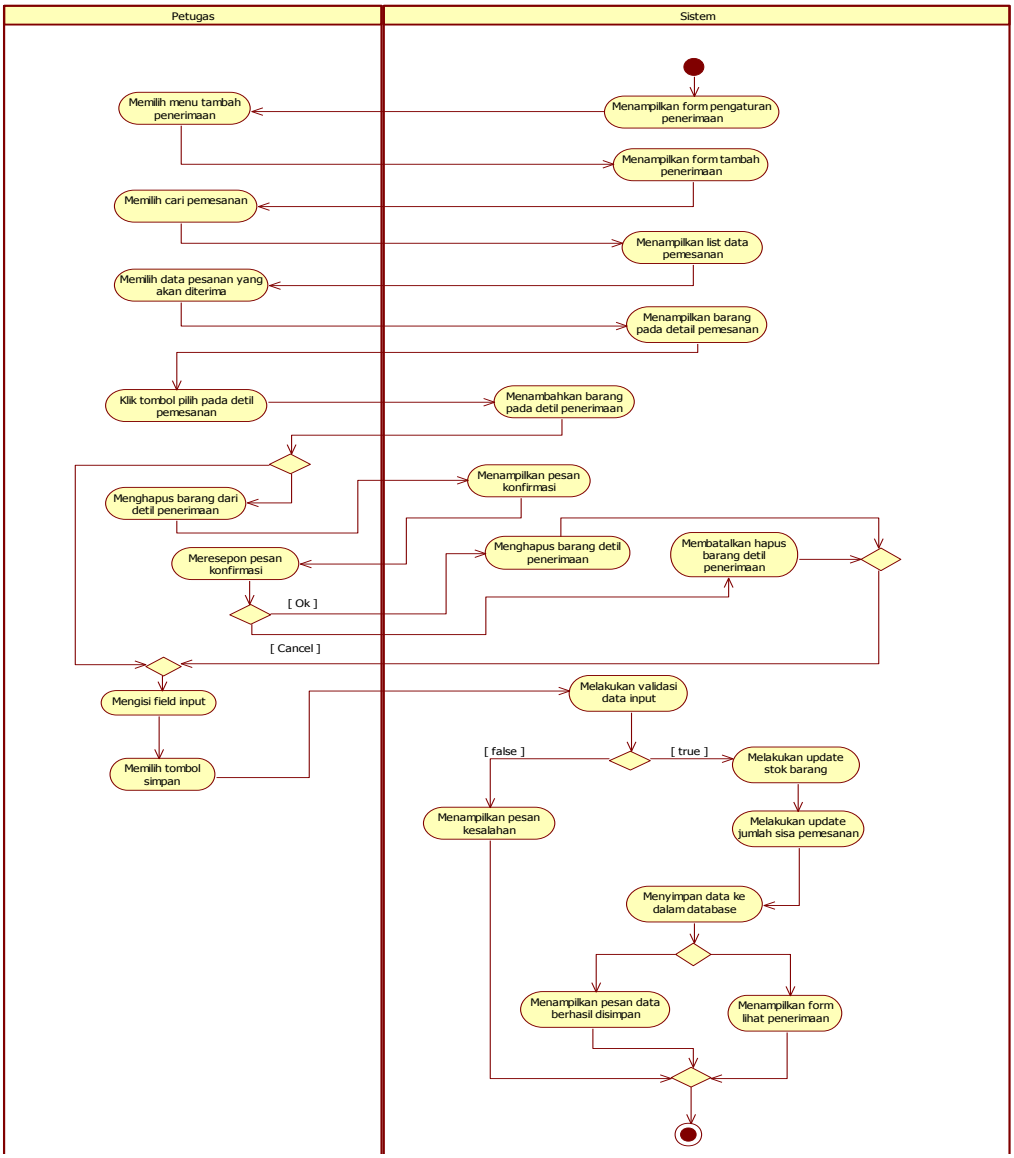
Gambar 3.26 *Activity Diagram* Hapus Pemesanan

3.1.2.10.7 *Activity Diagram* Pengolahan Data Penerimaan

Activity diagram pengolahan data penerimaan menjelaskan tentang aliran kerja aktor Admin atau Petugas dalam melakukan pengolahan data penerimaan yang digambarkan pada *activity diagram* tambah, ubah, dan hapus data penerimaan.

1. *Activity Diagram* Tambah Penerimaan

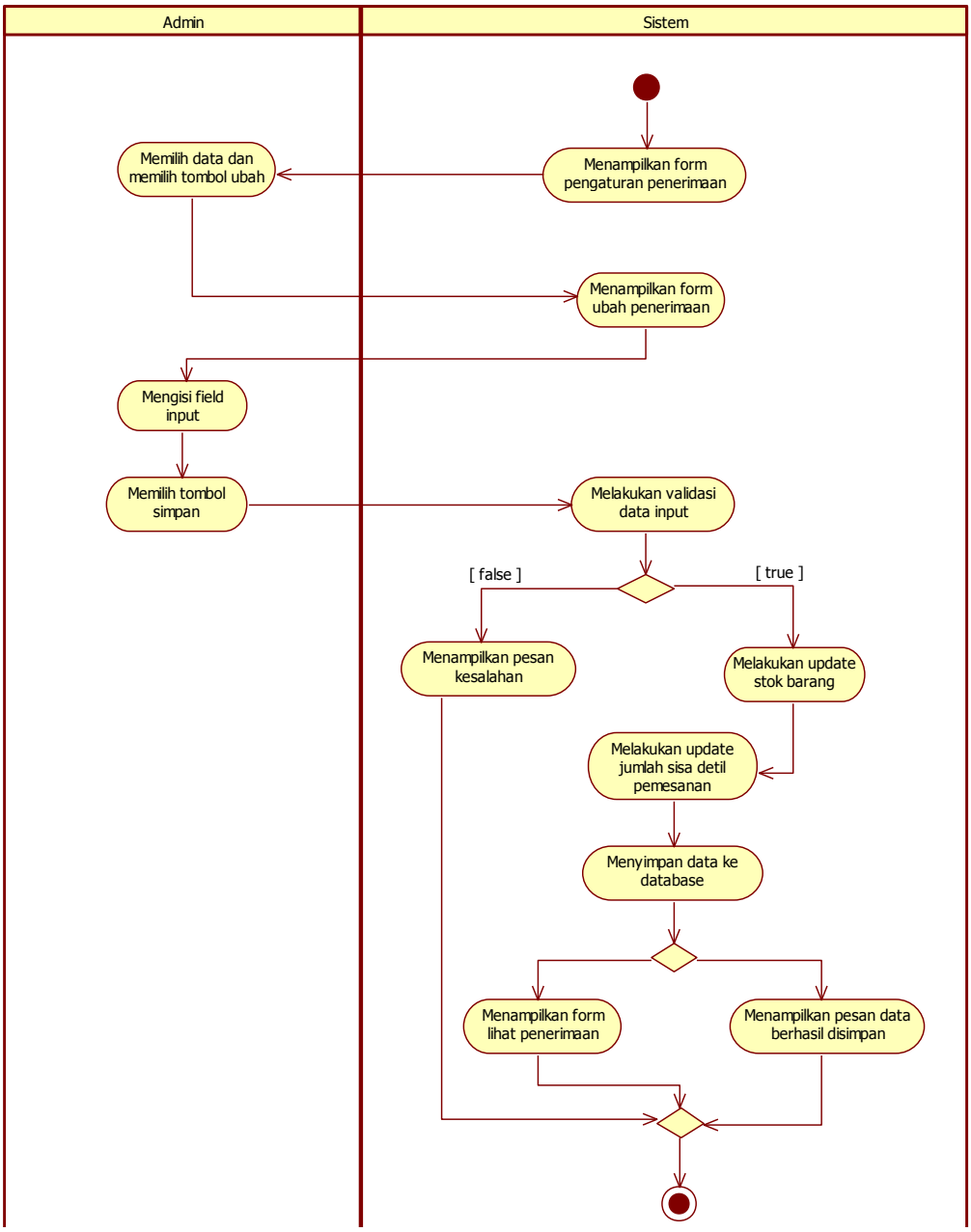
Interaksi antara aktor Petugas dengan sistem dalam proses tambah data pemesanan dijelaskan dalam *activity diagram* pada Gambar 3.27.



Gambar 3.27 *Activity Diagram* Tambah Penerimaan

2. *Activity Diagram* Ubah Penerimaan

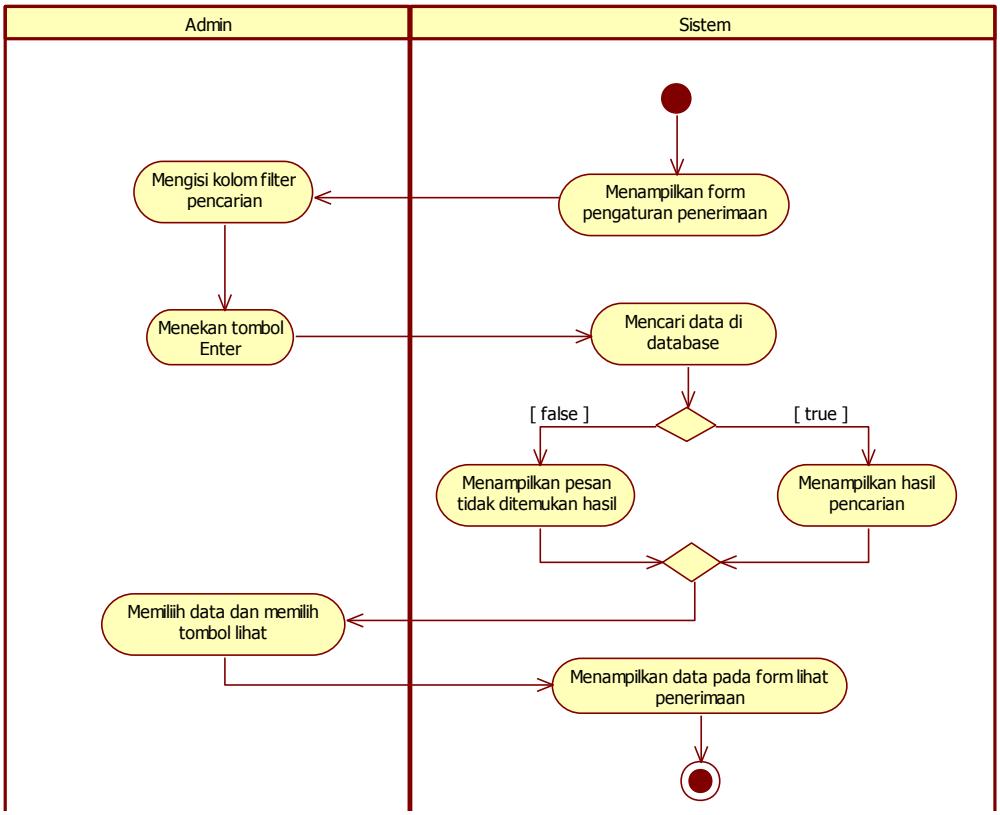
Interaksi antara aktor Admin dengan sistem dalam proses ubah data penerimaan dijelaskan dalam *activity diagram* pada Gambar 3.28.



Gambar 3.28 *Activity Diagram* Ubah Penerimaan

3. *Activity Diagram* Cari Penerimaan

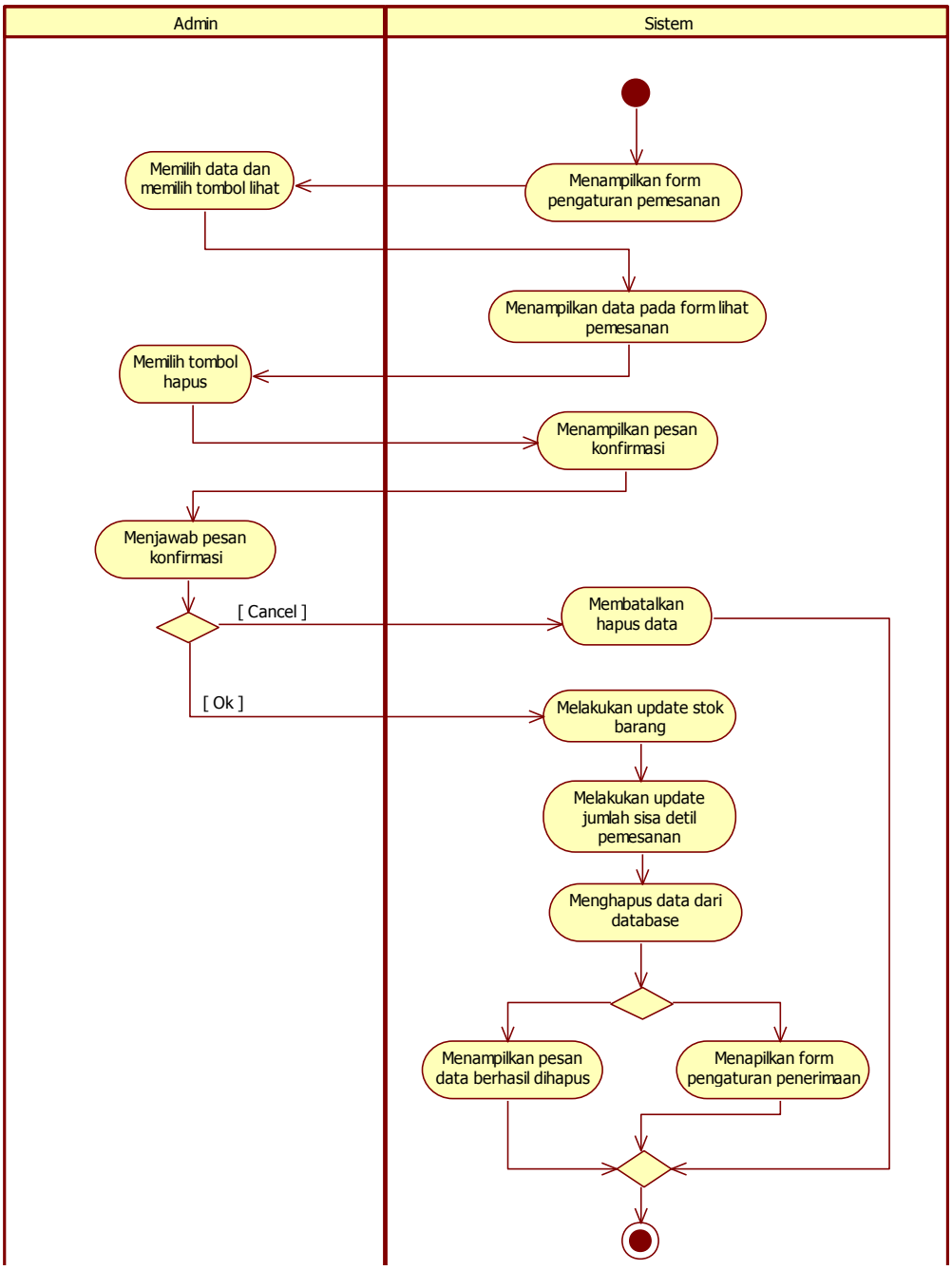
Interaksi antara aktor Admin dengan sistem dalam proses cari data penerimaan dijelaskan dalam *activity diagram* pada Gambar 3.29.



Gambar 3.29 *Activity Diagram* Cari Penerimaan

4. *Activity Diagram* Hapus Penerimaan

Interaksi antara aktor Admin dengan sistem dalam proses hapus data penerimaan dijelaskan dalam *activity diagram* pada Gambar 3.30.



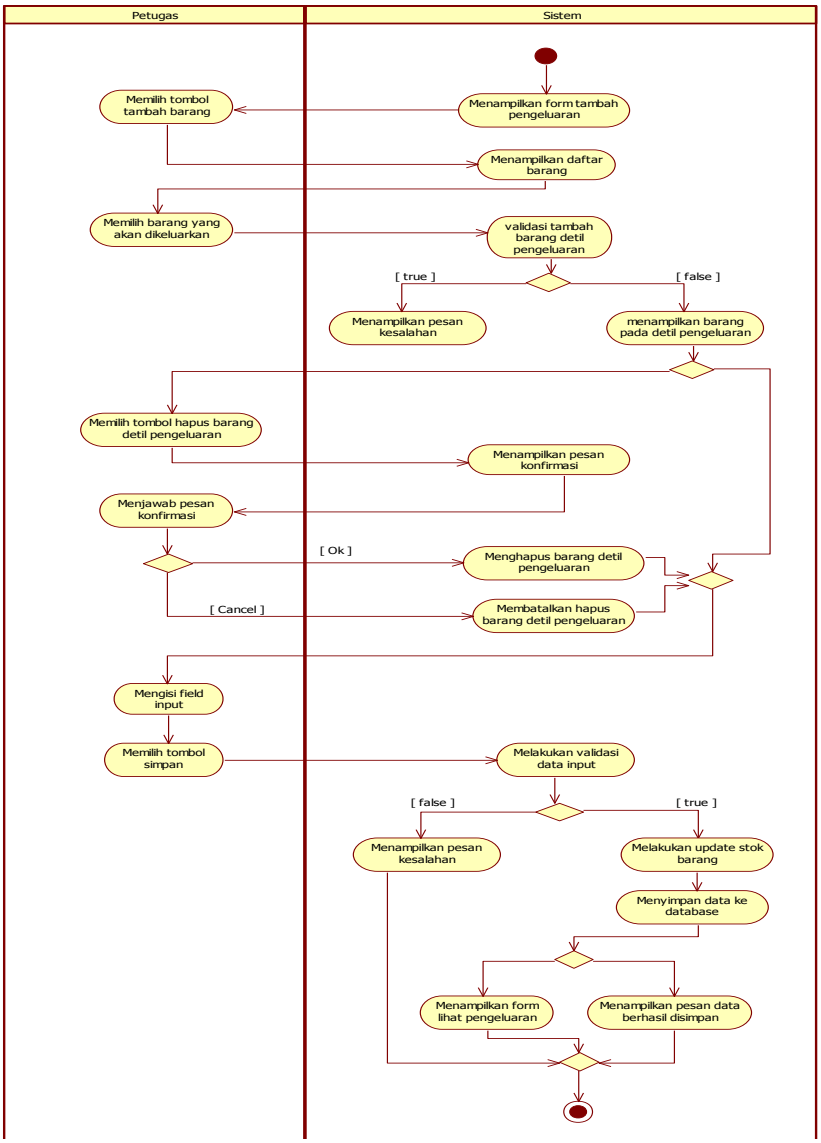
Gambar 3.30 *Activity Diagram* Hapus Penerimaan

3.1.2.10.8 *Activity Diagram* Pengolahan Data Pengeluaran

Activity diagram pengolahan data pengeluaran menjelaskan tentang aliran kerja aktor Admin atau Petugas dalam melakukan pengolahan data pengeluaran seperti tambah, ubah, dan hapus data pengeluaran. Berikut penjelasan dari masing-masing *activity diagram* tersebut.

1. *Activity Diagram* Tambah Pengeluaran

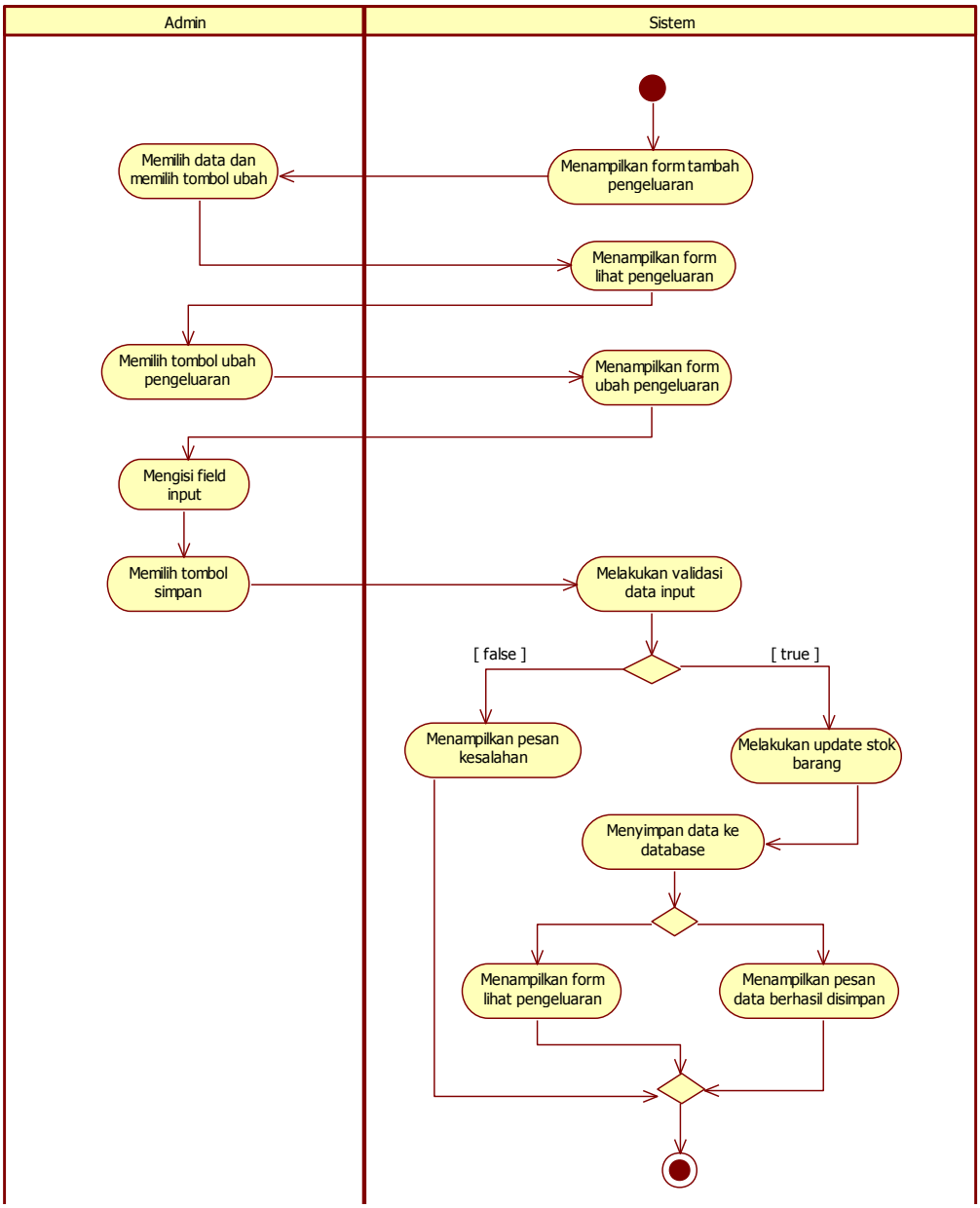
Interaksi antara aktor Petugas dengan sistem dalam proses tambah data pengeluaran dijelaskan dalam *activity diagram* pada Gambar 3.31.



Gambar 3.31 *Activity Diagram* Tambah Pengeluaran

2. *Activity Diagram* Ubah Pengeluaran

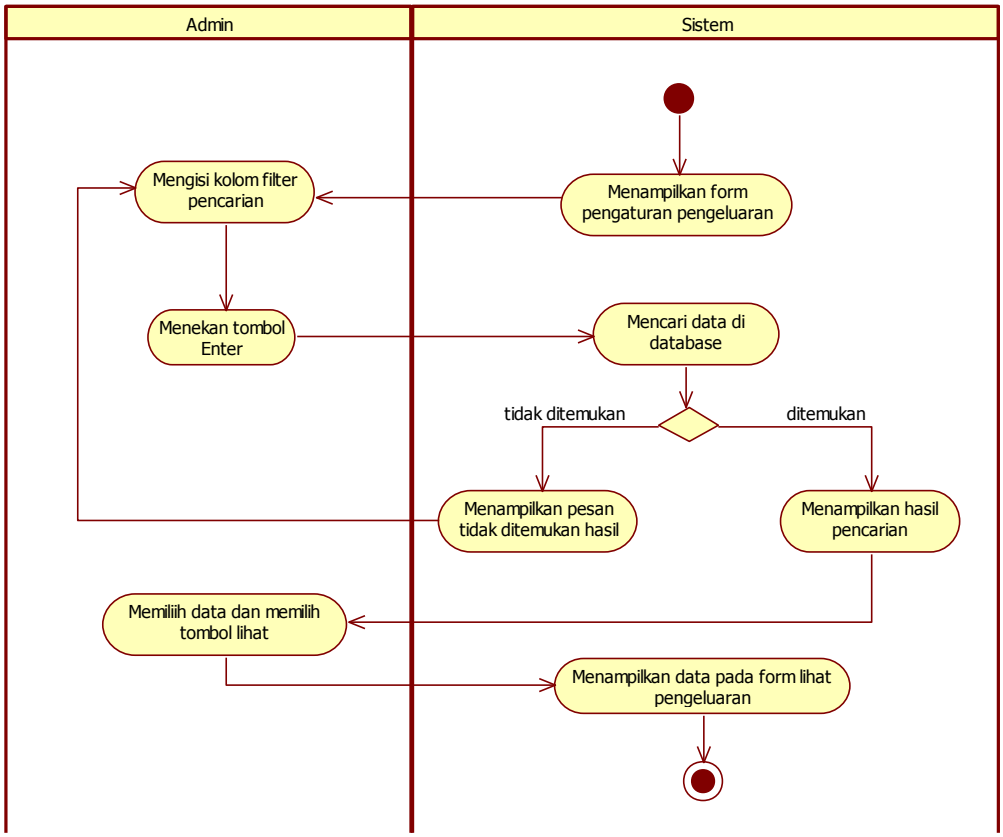
Interaksi antara aktor Admin dengan sistem dalam proses ubah data pengeluaran dijelaskan dalam *activity diagram* pada Gambar 3.32.



Gambar 3.32 *Activity Diagram* Ubah Pengeluaran

3. *Activity Diagram* Cari Pengeluaran

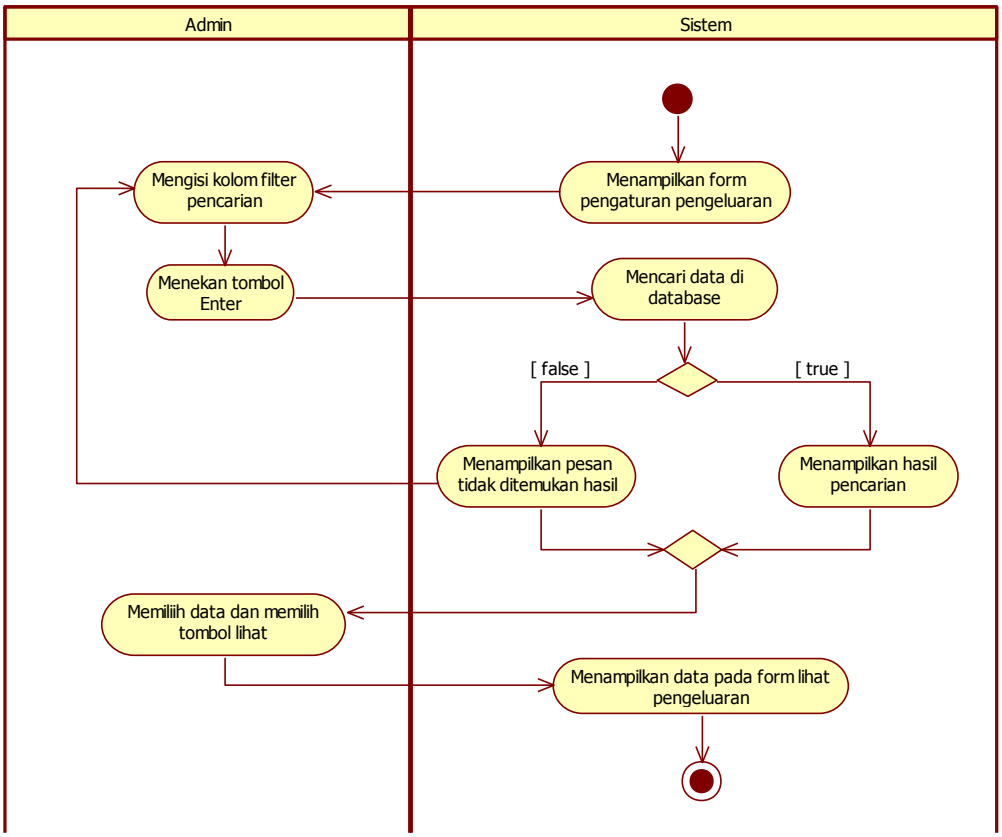
Interaksi antara aktor Admin dengan sistem dalam proses cari data pengeluaran dijelaskan dalam *activity diagram* pada Gambar 3.33.



Gambar 3.33 *Activity Diagram* Cari Pengeluaran

4. *Activity Diagram* Hapus Pengeluaran

Interaksi antara aktor Admin dengan sistem dalam proses hapus data pengeluaran dijelaskan dalam *activity diagram* pada Gambar 3.34.



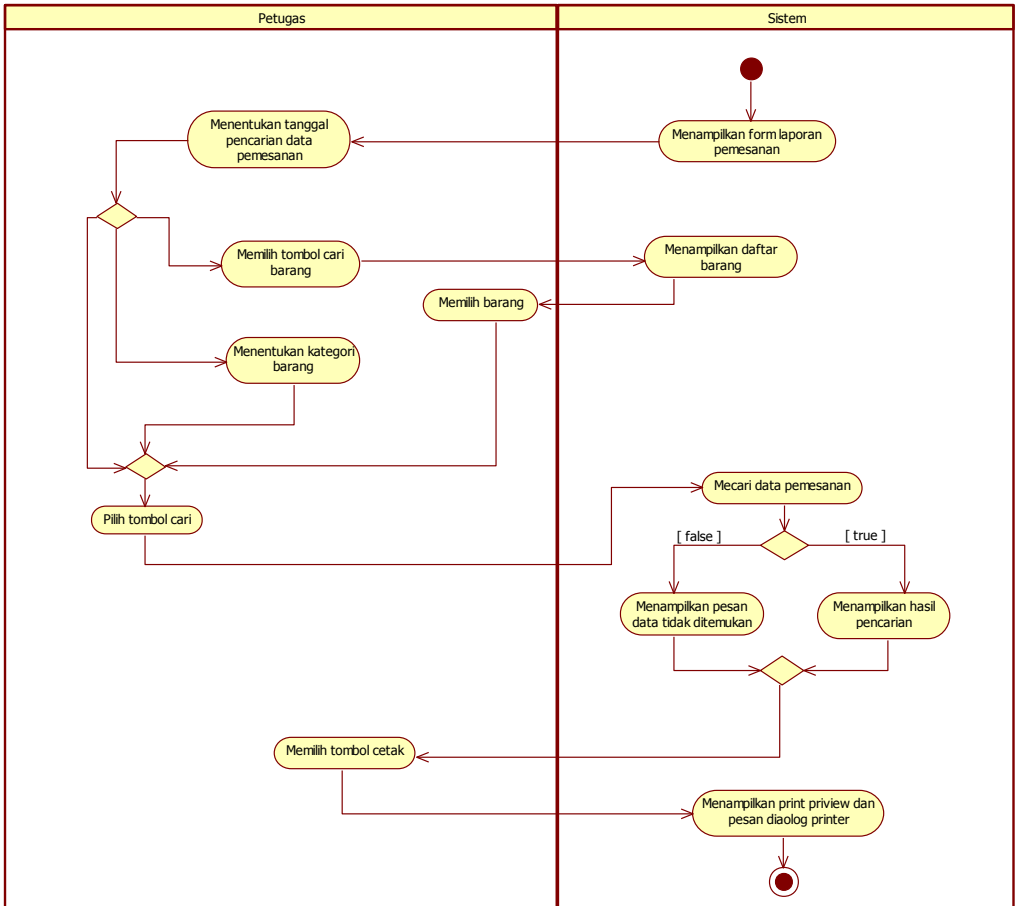
Gambar 3.34 *Activity Diagram* Hapus Pengeluaran

3.1.2.10.9 *Acitvity Diagram* Pengolahan Laporan

Interaksi antara aktor Petugas dengan sistem dalam proses pengolahan laporan dijelaskan pada *activity diagram* laporan pemesanan, penerimaan, dan pengeluaran.

1. *Activity Diagram* Laporan Pemesanan

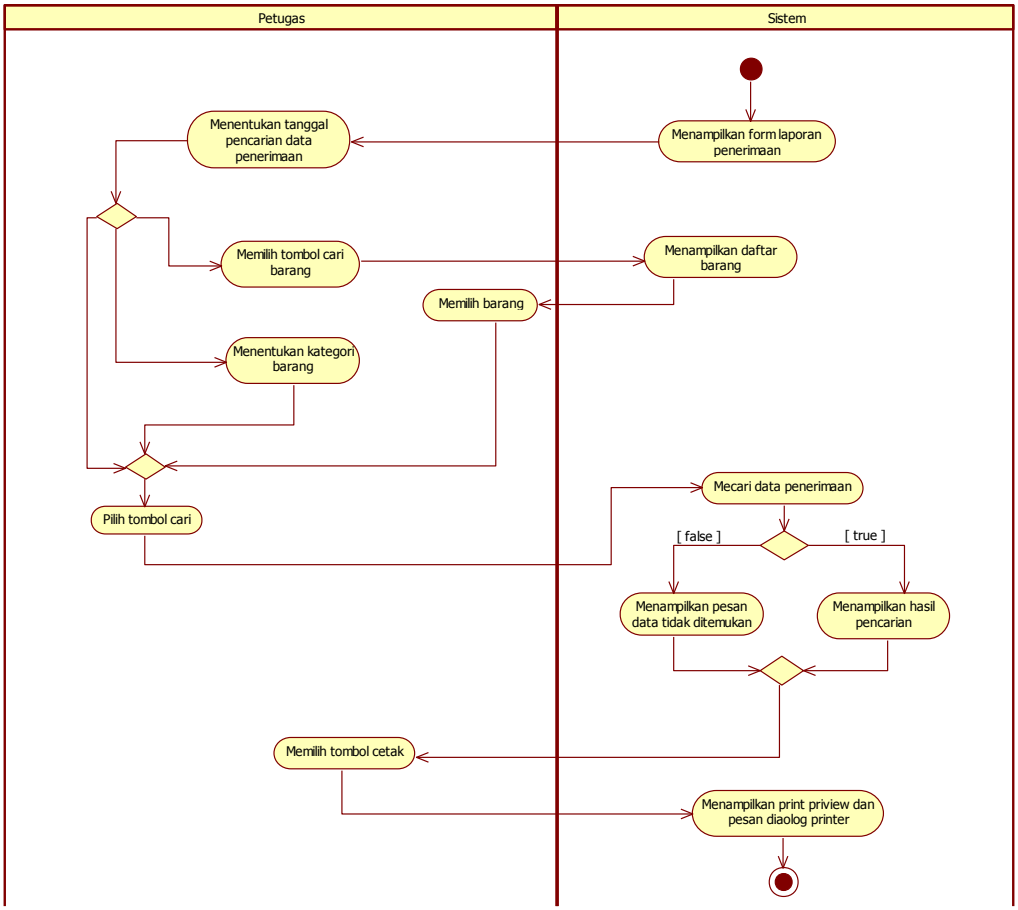
Interaksi antara aktor Petugas dengan sistem dalam proses pengolahan laporan pemesanan dijelaskan pada dalam *activity diagram* pada Gambar 3.35.



Gambar 3.35 *Activiy Diagram* Laporan Pemesanan

2. *Activity Diagram* Laporan Penerimaan

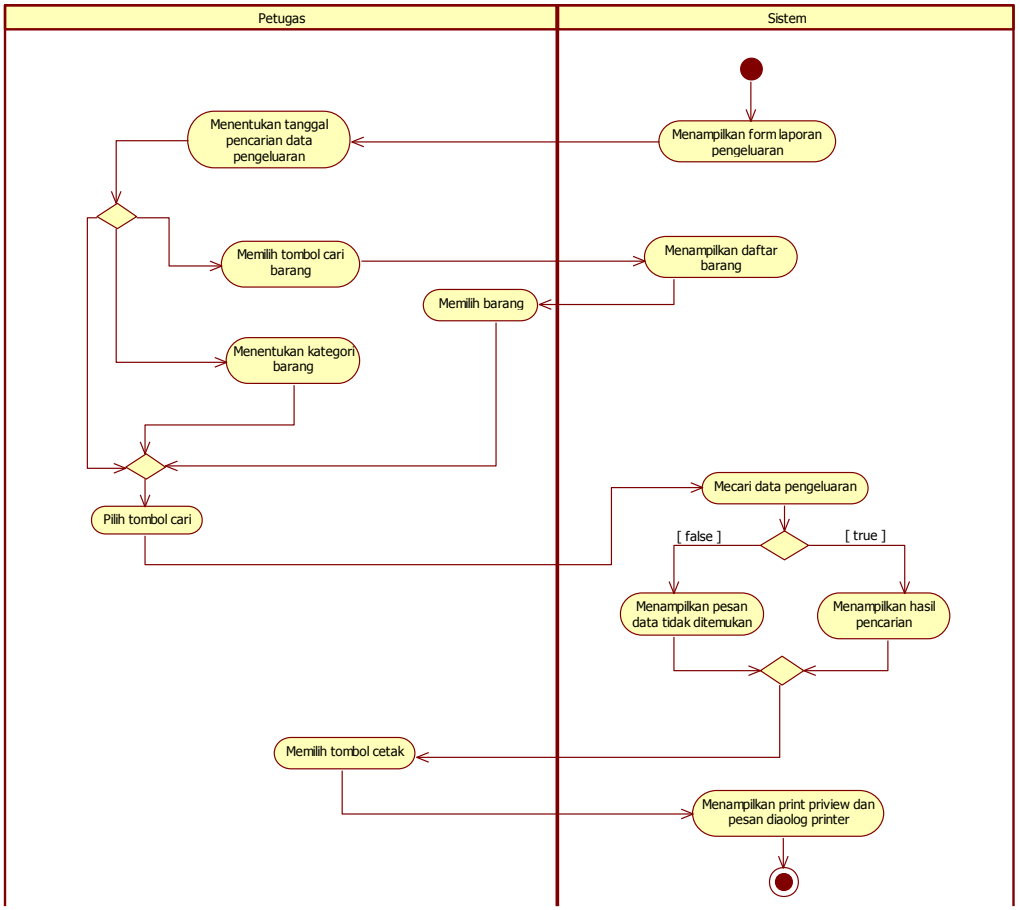
Interaksi antara aktor Petugas dengan sistem dalam proses pengolahan laporan penerimaan dijelaskan dalam *activity diagram* pada Gambar 3.36.



Gambar 3.36 *Activity Diagram* Laporan Penerimaan

3. *Activity Diagram* Laporan Pengeluaran

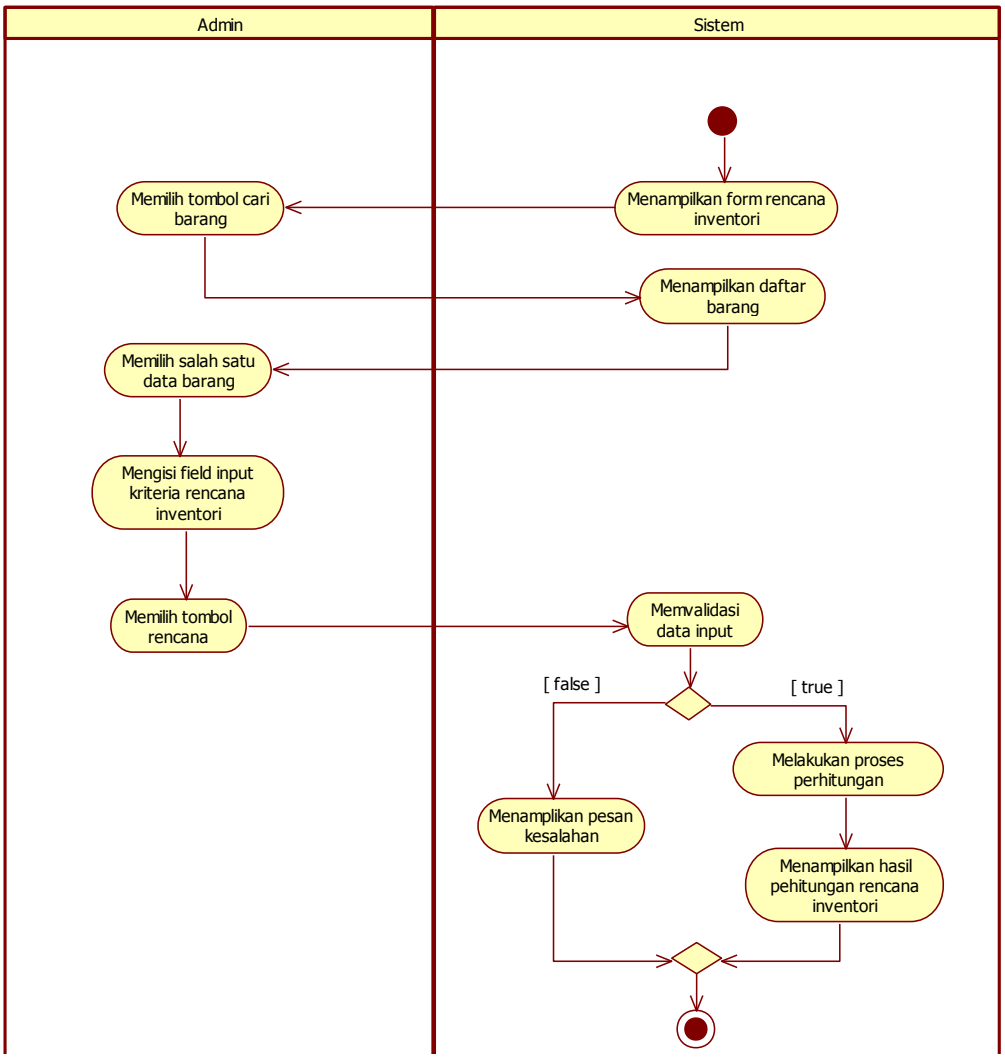
Interaksi antara aktor Petugas dengan sistem dalam proses pengolahan laporan pengeluaran dijelaskan dalam *activity diagram* pada Gambar 3.37.



Gambar 3.37 *Activity Diagram* Laporan Pengeluaran

3.1.2.10.10 *Activity Diagram* Perhitungan Rencana Inventori

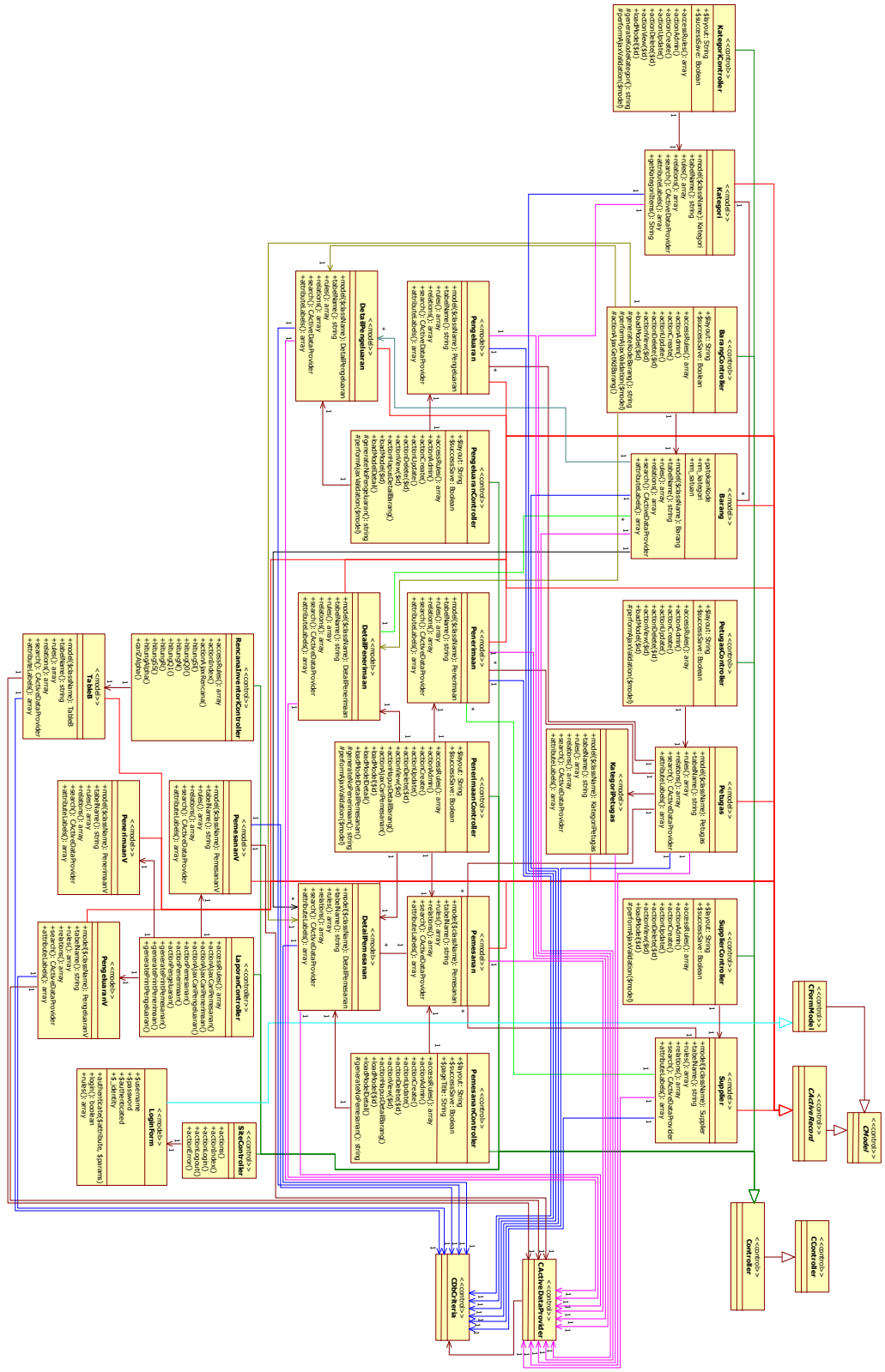
Interaksi antara aktor Petugas dengan sistem dalam proses perhitungan rencana inventori dijelaskan dalam *activity diagram* pada Gambar 3.38.



Gambar 3.38 *Activity Diagram* Perhitungan Rencana Inventori

3.1.2.11 *Class Diagram*

Class Diagram menggambarkan struktur dan hubungan antar objek-objek yang ada pada sistem. Struktur itu meliputi atribut-atribut dan metode-metode yang ada pada masing-masing kelas. Adapun gambaran *class diagram* dari aplikasi sistem informasi *inventory control* yang dibangun dapat dilihat pada Gambar 3.39.



Gambar 3.39 *Class Diagram* Sistem Informasi *Inventory Control*

3.1.2.11.1 Daftar Kelas

Kelas diagram aplikasi sistem informasi *inventory control* yang dibangun terdiri dari dua jenis kelas yaitu *class control* dan *class model.* Daftar setiap kelas pada *class diagram* sistem informasi *inventory control* berdasarkan jenisnya dapat dilihat pada tabel berikut.

1. Kelas *Model*

Tabel 3.50 Deskripsi Kelas *Model*

No	Nama Kelas	Jenis Kelas
1	LoginForm	*Model*
2	Barang	*Model*
3	Satuan	*Model*
4	Kategori	*Model*
5	Petugas	*Model*
6	Kacab	*Model*
7	Supplier	*Model*
8	Pemesanan	*Model*
9	Penerimaan	*Model*
10	Pengeluaran	*Model*
11	DetailPemesanan	*Model*
12	DetailPenerimaan	*Model*
13	DetailPengeluaran	*Model*
14	KategoriPetugas	*Model*
15	TabelB	*Model*
16	PemesananV	*Model*
17	PenerimaanV	*Model*
18	PengeluaranV	*Model*

2. Kelas *Control*

Tabel 3.51 Deskripsi Kelas *Control*

No	Nama Kelas	Jenis Kelas
1	BarangController	*Control*
2	KategoriController	*Control*
3	SatuanController	*Control*
4	PetugasController	*Control*
5	KacabController	*Control*
6	SupplierController	*Control*
7	SiteController	*Control*
8	PemesananController	*Control*
9	PenerimaanController	*Control*
10	PengeluaranController	*Control*
11	LaporanController	*Control*
12	RencanaInventoriController	*Control*

Spesifikasi kelas dalam sistem informasi *inventory control* yang dibangun di antaranya adalah sebagai berikut.

3.1.2.11.2 Kategori

Kelas yang digunakan dalam pengolahan data kategori dalam sistem informasi yang dibangun di antaranya adalah kelas KategoriController dan Kategori.

1. Kelas KategoriController

Kelas KategoriController merupakan kelas yang berfungsi untuk mengatur proses pengolahan data kategori dan mengatur hubungan antara *view* dengan kelas *model*. Deskripsi dari setiap metode dalam kelas KategoriController dapat dilihat pada Tabel 3.52.

Tabel 3.52 Deskripsi Kelas KategoriController

Nama	Keterangan
$layout	Parameter yang digunakan untuk mendeklarasikan templet halaman yang akan digunakan untuk pengolahan data kategori
$successSave	Parameter yang digunakan untuk acuan keberhasilan pengolahan data
accessRules()	Digunakan untuk menetapkan aturan akses *user* terhadap aksi yang ada pada *controller*
actionAdmin()	Untuk menampilkan *form* pengaturan kategori dan melakukan aksi cari data
actionCreate()	Untuk menampilkan *form* tambah kategori dan melakukan aksi tambah data
actionUpdate()	Untuk menampilkan *form* ubah kategori dan melakukan aksi ubah data
actionView()	Untuk menampilkan data pada halaman lihat kategori
actionDelete()	Untuk melakukan aksi hapus data kategori
loadModel()	Untuk memanggil data pada kelas *model*
generateKodeKategori()	Fungsi untuk menghasilkan kode kategori
performAjaxValidation()	Digunakan untuk validasi data masukan dari *user*

2. Kelas Kategori

Kelas Kategori merupakan kelas model yang mengeksekusi perintah atau aksi dari kelas *controller* terhadap *database* seperti aksi tambah, ubah, *load* data, cari dan hapus data. Penjelasan dari kelas Kategori dapat dilihat pada Tabel 3.53.

Tabel 3.53 Deskrisi Kelas Kategori

Nama	Keterangan
model()	Digunakan untuk menampung aksi dari *controller* terhadap *database* dengan me-*return* nama kelasnya sendiri
tableName()	Mendeklarasikan tabel dari *database* yang digunakan
rules()	Mendeklarasikan atribut entitas untuk diajdikan acuan validasi
relations()	Untuk mengatur relasi kelas Kategori dengan kelas lainnya
search()	Aturan cari data dan menampilkan hasil sesuai *input*-an *user*
attributeLabels()	Inisialisasi label untuk tiap atribut pada tabel data
getKategoriItems()	Fungsi untuk mengambil semua data dari tabel kategori

3.1.2.11.3 Barang

Kelas yang digunakan dalam pengolahan data barang dalam sistem informasi yang dibangun di antaranya adalah *class* BarangController dan *class* Barang.

1. Kelas BarangController

Kelas BarangController merupakan kelas yang berfungsi untuk mengatur proses pengolahan data barang dan mengatur hubungan antara *view* dengan kelas *model*. **P**enjelasan dari setiap metode dari kelas BarangController dapat dilihat pada Tabel 3.54.

Tabel 3.54 Deskripsi Kelas BarangController

Nama	Keterangan
$layout	Parameter yang digunakan untuk mendeklarasikan templet halaman yang akan digunakan untuk pengolahan data kategori
$successSave	Parameter yang digunakan untuk acuan keberhasilan pengolahan data
accessRules()	Digunakan untuk menetapkan aturan akses *user* terhadap aksi yang ada pada *controller*
actionAdmin()	Untuk menampilkan halaman pengaturan barang dan melakukan aksi cari data
actionCreate()	Untuk menampilkan halaman tambah barang dan melakukan aksi tambah data
actionUpdate()	Untuk menampilkan halaman ubah barang dan melakukan aksi ubah data
actionView()	Untuk menampilkan data pada halaman lihat barang
actionDelete()	Digunakan untuk melakukan aksi hapus data
loadModel()	Digunakan untuk memanggil data pada kelas *model*
generateKodeBarang()	Fungsi untuk menghasilkan kode barang
performAjaxValidation()	Digunakan untuk validasi data masukan dari *user*
actionAjaxGetKdBarang()	Digunakan untuk mengambil hasil generate kode barang

2. Kelas Barang

Kelas Barang merupakan kelas model yang mengeksekusi perintah atau aksi dari kelas *controller* terhadap *database* seperti aksi tambah, ubah, ambil data, cari data dan hapus data. Penjelasan dari setiap metode pada kelas Barang dapat dilihat pada Tabel 3.55.

Tabel 3.55 Deskrisi Kelas Barang

Nama	Keterangan
model()	Digunakan untuk menampung aksi dari *controller* terhadap *database* dengan me-*return* nama kelasnya sendiri
tableName()	Mendeklarasikan tabel dari *database* yang digunakan
rules()	Mendeklarasikan atribut entitas untuk dijadikan acuan validasi
relations()	Untuk mengatur relasi kelas Barang dengan kelas lainnya
search()	Fungsi untuk mencari data berdasarkan masukan dari *user*
attributeLabels()	Inisialisasi label untuk tiap atribut pada tabel data

3.1.2.11.4 Petugas

Kelas yang digunakan dalam pengolahan data petugas dalam sistem informasi yang dibangun di antaranya adalah kelas PetugasController, kelas Petugas, dan kelas KategoriPetugas.

1. Kelas PetugasController

Kelas PetugasController merupakan kelas yang berfungsi untuk mengatur proses pengolahan data petugas dan mengatur hubungan antara *view* dengan kelas *model*. Penjelasan dari kelas PetugasController dapat dilihat pada Tabel 3.56.

Tabel 3.56 Deskripsi Kelas PetugasController

Nama	Keterangan
$layout	Parameter yang digunakan untuk mendeklarasikan templet halaman yang akan digunakan untuk pengolahan data kategori
$successSave	Parameter yang digunakan untuk acuan keberhasilan pengolahan data
accessRules()	Digunakan untuk menetapkan aturan akses *user* terhadap aksi yang ada pada *controller*
actionAdmin()	Untuk menampilkan halaman pengaturan petugas dan melakukan aksi cari data
actionCreate()	Untuk menampilkan halaman tambah petugas dan melakukan aksi tambah data
actionUpdate()	Untuk menampilkan halaman ubah petugas dan melakukan aksi ubah data
actionView()	Untuk menampilkan data pada halaman lihat petugas
actionDelete()	Digunakan untuk melakukan aksi hapus data
loadModel()	Digunakan untuk memanggil data pada kelas *model*
performAjaxValidation()	Digunakan untuk validasi data masukan dari *user*

2. Kelas Petugas

Kelas Petugas merupakan kelas model yang mengeksekusi perintah atau aksi dari kelas *controller* terhadap *database* seperti aksi tambah, ubah, ambil data, cari data dan hapus data. Penjelasan kelas Petugas dapat dilihat pada Tabel 3.57.

Tabel 3.57 Deskrisi Kelas Petugas

Nama	Keterangan
model()	Digunakan untuk menampung aksi dari *controller* terhadap *database* dengan me-*return* nama kelasnya sendiri
tableName()	Mendeklarasikan tabel dari *database* yang digunakan
rules()	Mendeklarasikan atribut entitas untuk dijadikan acuan validasi
relations()	Untuk mengatur relasi kelas Petugas dengan kelas lainnya
search()	Fungsi untuk mencari data berdasarkan masukan dari *user*
attributeLabels()	Inisialisasi label untuk tiap atribut pada tabel data

3. Kelas Kategori Petugas

Kelas Petugas merupakan kelas model yang mengeksekusi perintah atau aksi dari kelas *controller* terhadap *database* seperti aksi tambah, ubah, ambil data, cari data dan hapus data. Penjelasan dari kelas Petugas dapat dilihat pada Tabel 3.58.

Tabel 3.58 Deskrisi Kelas Petugas

Nama	Keterangan
model()	Digunakan untuk menampung aksi dari *controller* terhadap *database* dengan me-*return* nama kelasnya sendiri
tableName()	Mendeklarasikan tabel dari *database* yang digunakan
rules()	Mendeklarasikan atribut entitas untuk dijadikan acuan validasi
relations()	Untuk mengatur relasi kelas KatgoriPetugas dengan kelas lainnya
search()	Fungsi untuk mencari data berdasarkan masukan dari *user*
attributeLabels()	Inisialisasi label untuk tiap atribut pada tabel data

3.1.2.11.5 *Supplier*

Kelas yang digunakan dalam pengolahan data *supplier* dalam sistem informasi yang dibangun di antaranya adalah kelas SupplierController, dan *class* Supplier.

1. Kelas SupplierController

Kelas SupplierController merupakan kelas yang berfungsi untuk mengatur proses pengolahan data *supplier* dan mengatur hubungan antara *view* dengan kelas *model*. Penjelasan dari kelas SupplierController dapat dilihat pada Tabel 3.59.

Tabel 3.59 Deskripsi Kelas SupplierController

Nama	Keterangan
$layout	Parameter yang digunakan untuk mendeklarasikan templet halaman yang akan digunakan untuk pengolahan data kategori
$successSave	Parameter yang digunakan untuk acuan keberhasilan pengolahan data
accessRules()	Digunakan untuk menetapkan aturan akses *user* terhadap aksi yang ada pada *controller*
actionAdmin()	Untuk menampilkan halaman pengaturan *supplier* dan melakukan aksi cari data
actionCreate()	Untuk menampilkan halaman tambah *supplier* dan melakukan aksi tambah data
actionUpdate()	Untuk menampilkan halaman ubah *supplier* dan melakukan aksi ubah data
actionView()	Untuk menampilkan data pada halaman lihat *supplier*
actionDelete()	Digunakan untuk melakukan aksi hapus data
loadModel()	Digunakan untuk memanggil data pada kelas *model*
performAjaxValidation()	Digunakan untuk validasi data masukan dari *user*

1. Kelas Supplier

Kelas Supplier merupakan kelas model yang mengeksekusi perintah atau aksi dari kelas *controller* terhadap *database* seperti aksi tambah, ubah, ambil data, cari data dan hapus data. Penjelasan dari setiap metode dari kelas Supplier dapat dilihat pada tabel Tabel 3.60.

Tabel 3.60 Deskrisi Kelas Supplier

Nama	Keterangan
model()	Digunakan untuk menampung aksi dari *controller* terhadap *database* dengan me-*return* nama kelasnya sendiri
tableName()	Mendeklarasikan tabel dari *database* yang digunakan
rules()	Mendeklarasikan atribut entitas untuk dijadikan acuan validasi
relations()	Untuk mengatur relasi kelas Supplier dengan kelas lainnya
search()	Fungsi untuk mencari data berdasarkan masukan dari *user*
attributeLabels()	Inisialisasi label untuk tiap atribut pada tabel data

3.1.2.11.6 Pemesanan

Kelas yang digunakan dalam pengolahan data pemesanan dalam sistem informasi yang dibangun di antaranya adalah *class* PemesananController, *class* Pemesanan, dan *class* DetailPemesanan.

1. Kelas PemesananController

Class PemesananController merupakan kelas yang berfungsi untuk mengatur proses pengolahan data pemesanan dan mengatur hubungan antara *view* dengan kelas *model*. Penjelasan dari setiap metode dari kelas PemesananController dapat dilihat pada Tabel 3.61.

Tabel 3.61 Deskripsi Kelas PemesananController

Nama	Keterangan
$layout	Parameter yang digunakan untuk mendeklarasikan templet halaman yang akan digunakan untuk pengolahan data kategori
$successSave	Parameter yang digunakan untuk acuan keberhasilan pengolahan data
accessRules()	Digunakan untuk menetapkan aturan akses *user* terhadap aksi yang ada pada *controller*
actionAdmin()	Untuk memanggil halaman pengaturan pemesanan dan melakukan aksi cari data
actionCreate()	Untuk menampilkan halaman tambah pemesanan dan melakukan aksi tambah data
actionUpdate()	Untuk menampilkan halaman ubah pemesanan dan melakukan aksi ubah data
actionView()	Untuk menampilkan data pada halaman lihat pemesanan
actionDelete()	Digunakan untuk melakukan aksi hapus data
actionHapusDetailBarang()	Digunakan untuk melakukan aksi hapus data detill pemesanan
loadModel()	Digunakan untuk memanggil data pada kelas *model*
loadModelDetail()	Digunakan untuk memanggil data detill pemesanan pada kelas *model*
generateNoPemesanan()	Digunakan untuk menghasilkan no pemesanan

2. Kelas Pemesanan

Kelas Pemesanan merupakan kelas model yang mengeksekusi perintah atau aksi dari kelas *controller* terhadap *database* seperti aksi tambah, ubah, ambil data, cari data dan hapus data. Penjelasan dari dari kelas Pemesanan dapat dilihat pada Tabel 3.62.

Tabel 3.62 Deskrisi Kelas Pemesanan

Nama	Keterangan
$tglAwal	Atribut yang digunakan sebagai parameter tanggal awal pencarian data pemesanan
$tglAkhir	Atribut yang digunakan sebagai parameter tanggal akhir pencarian data pemesanan
model()	Digunakan untuk menampung aksi dari *controller* terhadap *database* dengan me-*return* nama kelasnya sendiri
tableName()	Mendeklarasikan tabel dari *database* yang digunakan
rules()	Mendeklarasikan atribut entitas untuk dijadikan acuan validasi
relations()	Untuk mengatur relasi kelas Pemesanan dengan kelas lainnya
search()	Fungsi untuk mencari data berdasarkan masukan dari *user*
attributeLabels()	Inisialisasi label untuk tiap atribut pada tabel data

3. Kelas DetailPemesanan

Kelas DetailPemesanan merupakan kelas model yang mengeksekusi perintah atau aksi dari kelas *controller* terhadap *database* seperti aksi tambah, ubah, ambil data, cari dan hapus data detil pemesanan. Penjelasan dari kelas DetailPemesanan dapat dilihat pada Tabel 3.63.

Tabel 3.63 Deskrisi Kelas DetailPemesanan

Nama	Keterangan
model()	Digunakan untuk menampung aksi dari *controller* terhadap *database* dengan me-*return* nama kelasnya sendiri
tableName()	Mendeklarasikan tabel dari *database* yang digunakan
rules()	Mendeklarasikan atribut entitas untuk dijadikan acuan validasi
relations()	Untuk mengatur relasi kelas DeteailPemesanan dengan kelas lainnya
search()	Fungsi untuk mencari data berdasarkan masukan dari *user*
attributeLabels()	Inisialisasi label untuk tiap atribut pada tabel data

3.1.2.11.7 Penerimaan

Kelas yang digunakan dalam pengolahan data penerimaan dalam sistem informasi yang dibangun di antaranya adalah kelas PenerimaanController, Penerimaan, dan kelas DetailPenerimaan.

1. Kelas PenerimaanController

Kelas PenerimaanController merupakan kelas yang berfungsi untuk mengatur proses pengolahan data penerimaan dan mengatur hubungan antara *view* dengan kelas *model*. Penjelasan dari kelas PemesananController dapat dilihat pada Tabel 3.64 berikut.

Tabel 3.64 Deskripsi Kelas PenerimaanController

Nama	Keterangan
$layout	Parameter yang digunakan untuk mendeklarasikan templet halaman yang akan digunakan untuk pengolahan data kategori
$successSave	Parameter yang digunakan untuk acuan keberhasilan pengolahan data
accessRules()	Digunakan untuk menetapkan aturan akses *user* terhadap aksi yang ada pada *controller*
actionAdmin()	Digunakann untuk menuju halaman pengaturan penerimaan dan melakukan aksi cari data
actionCreate()	Digunakan untuk menuju halaman tambah data dan melakukan aksi tambah data
actionUpdate()	Digunakan untuk menuju halaman ubah data dan melakukan aksi ubah data
actionView()	Digunakan untuk menuju halaman lihat data
actionDelete()	Digunakan untuk melakukan aksi hapus data
actionHapusDetailBarang()	Digunakan untuk melakukan aksi hapus data detil penerimaan
loadModel()	Digunakan untuk memanggil data pada kelas *model*
loadModelDetail()	Digunakan untuk memanggil data detil penerimaan pada kelas *model*
loadModelDetailPemesanan()	Digunakan untuk memanggil data detil pemesanan pada kelas *model*
actionAjaxCariPemesanan()	Digunakan untuk mencari data pemesanan
generateNoPenerimaan()	Digunakan untuk menghasilkan no penerimaan

2. Kelas Penerimaan

Kelas Penerimaan merupakan kelas model yang mengeksekusi perintah atau aksi dari kelas *controller* terhadap *database* seperti aksi tambah, ubah, ambil data, cari data dan hapus data. Penjelasan kelas ini dapat dilihat pada Tabel 3.65.

Tabel 3.65 Deskrisi Kelas Penerimaan

Nama	Keterangan
$tglAwal	Atribut yang digunakan sebagai parameter tanggal awal pencarian data pemesanan
$tglAkhir	Atribut yang digunakan sebagai parameter tanggal akhir pencarian data pemesanan
model()	Digunakan untuk menampung aksi dari *controller* terhadap *database* dengan me-*return* nama kelasnya sendiri
tableName()	Mendeklarasikan tabel dari *database* yang digunakan
rules()	Mendeklarasikan atribut entitas untuk diajdikan acuan validasi
relations()	Untuk mengatur relasi kelas Penerimaan dengan kelas lainnya
search()	Fungsi untuk mencari data berdasarkan masukan dari *user*
attributeLabels()	Inisialisasi label untuk tiap atribut pada tabel data

3. *Class* DetailPenerimaan

Class DetailPemesanan merupakan kelas model yang mengeksekusi perintah atau aksi dari kelas *controller* terhadap *database* seperti aksi tambah, ubah, ambil data, cari data dan hapus data. Penjelasan kelas tersebut dapat dilihat pada Tabel 3.66 berikut.

Tabel 3.66 Deskrisi Kelas DetailPenerimaan

Nama	Keterangan
model()	Digunakan untuk menampung aksi dari *controller* terhadap *database* dengan me-*return* nama kelasnya sendiri
tableName()	Mendeklarasikan tabel dari *database* yang digunakan
rules()	Mendeklarasikan atribut entitas untuk dijadikan acuan validasi
relations()	Untuk mengatur relasi kelas DetailPenerimaan dengan kelas lainnya
search()	Fungsi untuk mencari data berdasarkan masukan dari *user*
attributeLabels()	Inisialisasi label untuk tiap atribut pada tabel data

3.1.2.11.8 Pengeluaran

Kelas yang digunakan dalam pengolahan data pengluaran dalam sistem informasi yang dibangun di antaranya adalah *class* PengeluaranController, kelas Pengeluaran, dan kelas DetailPengeluaran.

1. Kelas PengeluaranController

Kelas PengeluaranController merupakan kelas yang berfungsi untuk mengatur proses pengolahan data pengeluaran dan mengatur hubungan antara *view* dengan kelas *model*. Penjelasan dari kelas PengeluaranController dapat dilihat pada Tabel 3.67 berikut.

Tabel 3.67 Deskripsi Kelas PengeluaranController

Nama	Keterangan
$layout	Parameter yang digunakan untuk mendeklarasikan templet halaman yang akan digunakan untuk pengolahan data kategori
$successSave	Parameter yang digunakan untuk acuan keberhasilan pengolahan data
accessRules()	Digunakan untuk menetapkan aturan akses *user* terhadap aksi yang ada pada *controller*
actionAdmin()	Digunakann untuk menuju halaman pengaturan penerimaan dan melakukan aksi cari data
actionCreate()	Digunakan untuk menuju halaman tambah penerimaan dan melakukan aksi tambah data
actionUpdate()	Digunakan untuk menuju halaman ubah penerimaan dan melakukan aksi ubah data
actionView()	Digunakan untuk menuju halaman lihat penerimaan
actionDelete()	Digunakan untuk melakukan aksi hapus data
actionHapusDetailBarang()	Digunakan untuk melakukan aksi hapus data detil penerimaan
loadModel()	Digunakan untuk memanggil data pada kelas *model*
loadModelDetail()	Digunakan untuk memanggil data detail pengeluaran pada kelas *model*
generateNoPengeluaran()	Digunakan untuk menghasilkan no pengeluaran

2. Kelas Pengeluaran

Kelas Pengeluaran merupakan kelas model yang mengeksekusi perintah atau aksi dari kelas *controller* terhadap *database* seperti aksi tambah, ubah, ambil data, cari data dan hapus data. Penjelasan dari kelas tersebut dapat dilihat pada Tabel 3.68.

Tabel 3.68 Deskrisi Kelas Pengeluaran

Nama	Keterangan
$tglAwal	Atribut yang digunakan sebagai parameter tanggal awal pencarian data pemesanan
$tglAkhir	Atribut yang digunakan sebagai parameter tanggal akhir pencarian data pemesanan
model()	Digunakan untuk menampung aksi dari *controller* terhadap *database* dengan me-*return* nama kelasnya sendiri
tableName()	Mendeklarasikan tabel dari *database* yang digunakan
rules()	Mendeklarasikan atribut entitas untuk dijadikan acuan validasi
relations()	Untuk mengatur relasi kelas Pengeluaran dengan kelas lainnya
search()	Fungsi untuk mencari data berdasarkan masukan dari *user*
attributeLabels()	Inisialisasi label untuk tiap atribut pada tabel data

3. Kelas DetailPengeluaran

Kelas DetailPengeluaran merupakan kelas model yang mengeksekusi perintah atau aksi dari kelas *controller* terhadap *database* seperti aksi tambah, ubah, ambil data, cari data dan hapus data. Penjelasan kelas tersebut dapat dilihat pada pada Tabel 3.69.

Tabel 3.69 Deskrisi Kelas DetailPengeluaran

Nama	Keterangan
model()	Digunakan untuk menampung aksi dari *controller* terhadap *database* dengan me-*return* nama kelasnya sendiri
tableName()	Mendeklarasikan tabel dari *database* yang digunakan
rules()	Mendeklarasikan atribut entitas untuk dijadikan acuan validasi
relations()	Untuk mengatur relasi kelas DetailPengeluaran dengan kelas lainnya
search()	Fungsi untuk mencari data berdasarkan masukan dari *user*
attributeLabels()	Inisialisasi label untuk tiap atribut pada tabel data

3.1.2.11.9 Laporan

Kelas yang digunakan dalam pengolahan data laporan dalam sistem informasi yang dibangun di antaranya adalah kelas LaporanController, PemesananV, PenerimaanV, dan kelas PengeluaranV.

1. Kelas LaporanController

Class LaporanController merupakan kelas yang berfungsi untuk mengatur proses pengolahan data laporan dan mengatur hubungan antara *view* dengan kelas *model*. Deskripsi kelas LaporanController dapat dilihat pada tabel berikut.

Tabel 3.70 Deskripsi Kelas LaporanController

Nama	Keterangan
accessRules()	Digunakan untuk menetapkan aturan akses *user* terhadap aksi yang ada pada *controller*
actionAjaxCariPemesanan()	Fungsi untuk mencari data pemesanan dan menampilkannya dalam bentuk tabel
actionAjaxCariPenerimaan()	Fungsi untuk mencari data penerimaan dan menampilkannya dalam bentuk tabel
actionAjaxCariPengeluaran()	Fungsi untuk mencari data pengeluaran dan menampilkannya dalam bentuk tabel
actionPemesanan()	Fungsi untuk menuju halaman laporan pemesanan
actionPenerimaan()	Fungsi untuk menuju halaman laporan penerimaan
actionPengeluaran()	Fungsi untuk menuju halaman laporan pengeluaran
generatePrintPemesanan()	Fungsi untuk melakukan proses cetak laporan pemesanan
generatePrintPenerimaan()	Fungsi untuk melakukan proses cetak laporan penerimaan
generatePrintPengeluaran()	Fungsi untuk melakukan proses cetak laporan pengeluaran

2. Kelas PemesananV

Kelas PemesananV merupakan kelas model yang mengeksekusi perintah atau aksi dari kelas LaporanController untuk mengambil data pada *database*. Penjelasan dari kelas ini dapat dilihat pada Tabel 3.71.

Tabel 3.71 Dekripsi Kelas PemesananV

Nama	Keterangan
model()	Digunakan untuk menampung aksi dari *controller* terhadap *database* dengan me-*return* nama kelasnya sendiri
tableName()	Mendeklarasikan tabel dari *database* yang digunakan
rules()	Mendeklarasikan acuan validasi pada setiap atribut
relations()	Untuk mengatur relasi kelas PemesananV dengan kelas lainnya
search()	Fungsi untuk mencari data berdasarkan masukan dari *user*
attributeLabels()	Inisialisasi label untuk setiap atribut pada tabel data

3. Kelas PenerimaanV

Kelas PenerimaanV merupakan kelas model yang mengeksekusi perintah atau aksi dari kelas LaporanController untuk mengambil data pada *database*. Penjelasan dari kelas ini dapat dilihat pada Tabel 3.72.

Tabel 3.72 Dekripsi Kelas PenerimaanV

Nama	Keterangan
model()	Digunakan untuk menampung aksi dari *controller* terhadap *database* dengan me-*return* nama kelasnya sendiri
tableName()	Mendeklarasikan tabel dari *database* yang digunakan
rules()	Mendeklarasikan acuan validasi pada setiap atribut
relations()	Untuk mengatur relasi kelas PenerimaanV dengan kelas lainnya
search()	Fungsi untuk mencari data berdasarkan masukan dari *user*
attributeLabels()	Inisialisasi label untuk setiap atribut pada tabel data

4. Kelas PengeluaranV

Kelas PengeluaranV merupakan kelas model yang mengeksekusi perintah atau aksi dari kelas LaporanController untuk mengambil data pada *database*. Penjelasan dari kelas ini dapat dilihat pada Tabel 3.73.

Tabel 3.73 Dekripsi Kelas PengeluaranV

Nama	Keterangan
model()	Digunakan untuk menampung aksi dari *controller* terhadap *database* dengan me-*return* nama kelasnya sendiri
tableName()	Mendeklarasikan tabel dari *database* yang digunakan
rules()	Mendeklarasikan acuan validasi pada setiap atribut
relations()	Untuk mengatur relasi kelas PenerimaanV dengan kelas lainnya
search()	Fungsi untuk mencari data berdasarkan masukan dari *user*
attributeLabels()	Inisialisasi label untuk setiap atribut pada tabel data

3.1.2.11.10 Rencana Inventori

Kelas yang digunakan dalam pengolahan data rencana inventori dalam sistem informasi yang dibangun di antaranya adalah kelas RencanaInventoriController dan kelas TabelB

1. Kelas RencanaInventoriController

Kelas RencanaInventoriController merupakan kelas yang berfungsi untuk mengatur proses perhitugan rencana persediaan dan mengatur hubungan antara *view* dengan kelas *model*. Deskripsi dari kelas tersebut dapat dilihat pada pada Tabel 3.74.

Tabel 3.74 Deskripsi Kelas RecnanaInventoriController

Nama	Keterangan
accessRules()	Digunakan untuk menetapkan aturan akses *user* terhadap aksi yang ada pada *controller*
actionIndex()	Digunakan untuk menampilkan halaman rencana inventori
actionAjaxRencana()	Digunakann untuk menghitung rencana inventori
hitungSl()	Untuk menghitung standar deviasi pengeluaran
hitungQ0()	Fungsi mencari nilai lot pemesanan tahap 1
hitungN()	Fungsi meghitung nilai permintaan tidak terpenuhi
hitungQ1()	Fungsi meghitung nilai lot pemesanan tahap 2
hitungR()	Fungsi meghitung *reorder point*
hitungSS()	Fungsi meghitung *safety stock*
HitungAlpha()	Untuk mencari nilai *alpha*
cariZAlpha()	Untuk mencari nilai zα, ordinat, dan ekspektasi

2. Kelas TabelB

Kelas TabelB merupakan kelas model yang mengeksekusi perintah atau aksi dari kelas RencanaInventoriController terhadap *database*. Deskripsi mengenai kelas ini dapat dilihat pada Tabel 3.75.

Tabel 3.75 Deskrisi Kelas TabelB

Nama	Keterangan
model()	Digunakan untuk menampung aksi dari *controller* terhadap *database*
tableName()	Mendeklarasikan tabel dari *database* yang digunakan
rules()	Mendeklarasikan atribut entitas untuk diajdikan acuan validasi
relations()	Untuk mengatur relasi kelas TabelB dengan kelas lainnya
search()	Fungsi untuk mencari data pada tabel_b sesuai yang diminta kelas RencanaInventoriController
attributeLabels()	Mengatur label untuk proses perhitungan rencana inventori

3.1.2.11.11 *Login*

Kelas yang digunakan dalam proses *login* dalam sistem informasi yang dibangun di antaranya adalah kelas SiteController dan kelas LoginForm.

1. Kelas SiteController

Kelas SiteController merupakan kelas yang berfungsi untuk mengatur proses *login* dan mengatur hubungan antara *view* dengan kelas *model*. Deskripsi mengenai kelas ini dapat dilihat pada tabel berikut. Deskripsi mengenai kelas ini dapat dilihat pada Tabel 3.76.

Tabel 3.76 Deskripsi Kelas SiteController

Nama	Keterangan
actionIndex()	Digunakan untuk menampilkan halaman rencana inventori
actionLogin()	Digunakan untuk menampilkan halaman *login* dan melakukan aksi *login*
actionLogout()	Prosedur untuk keluar dari sistem dan tampil halaman awal sistem *inventory control*
actionError()	Digunakan untuk menangani kesalahan *login*

2. Kelas LoginForm

Kelas LoginForm merupakan kelas model yang mengeksekusi perintah atau aksi dari kelas SiteController terhadap *database.* Deskripsi mengenai kelas ini dapat dilihat pada Tabel 3.77.

Tabel 3.77 Deskripsi Kelas LoginForm

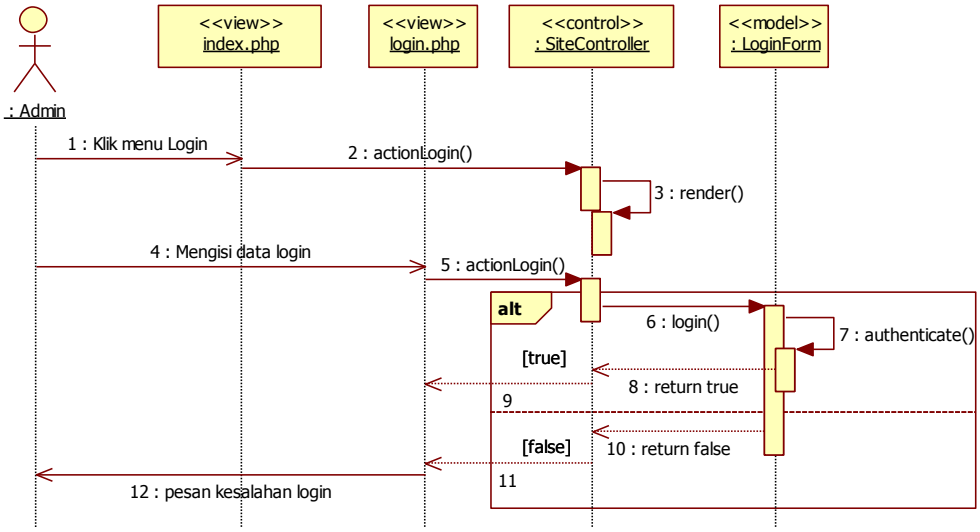
Nama	Keterangan
username	Atribut yang digunakan sebagai nama identitas *login*
password	Atribut yang digunakan sebagai kata kunci identitas *login*
authenticated	Atribut yang digunakan sebagai autentikasi
_identity	Atribut yang digunakan sebagai identitas *user*
authenticate()	Digunakan untuk outentifikasi data *login*
login()	Digunakan untuk validasi login dan mengembalikan nilai *Boolean* yang akan digunakan oleh fungsi *authenticate()*
rules()	Mendeklarasikan atribut yang dijadian acuan validasi

3.1.2.12 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 yang saling berinteraksi. Berikut ini penjelasan dari masing-masing *sequence diagram* tersebut:

3.1.2.12.1 *Sequence Diagram Login*

Sequence diagram login dapat dilihat sesuai dengan yang tertera pada gambar berikut.

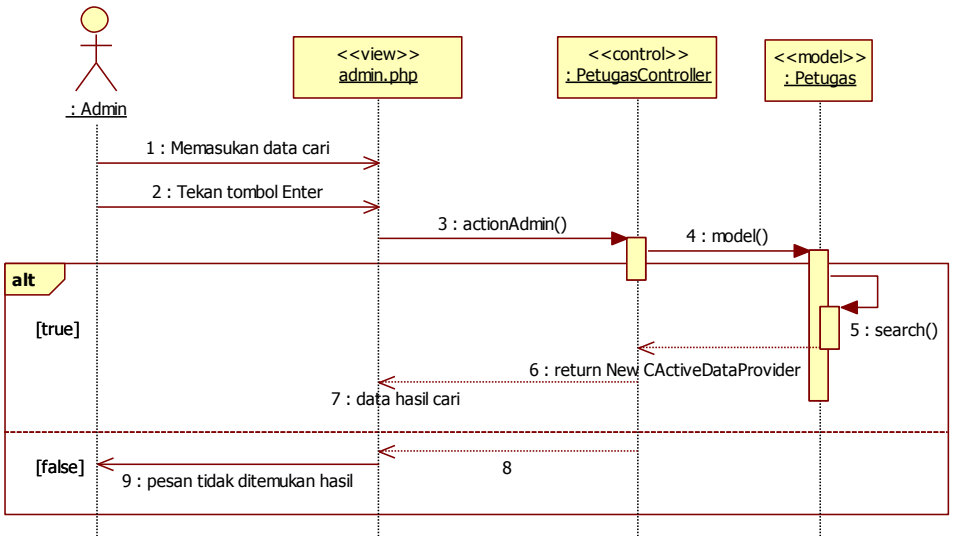


Gambar 3.40 *Sequence Diagram Login*

3.1.2.12.2 *Sequence Diagram* Pengolahan Data Petugas

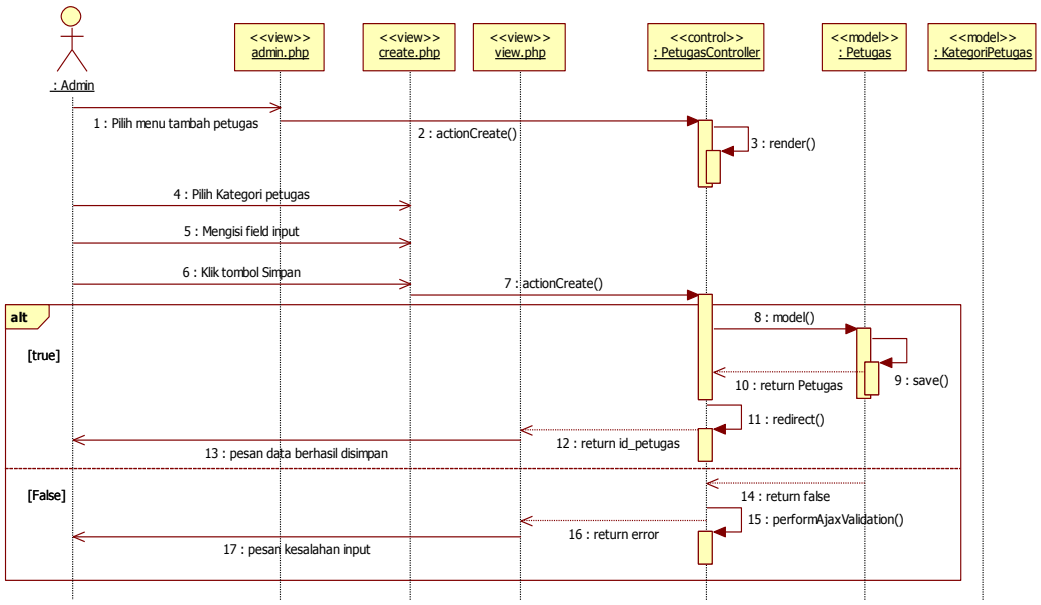
Sequence diagram pengolahan data petugas terdiri dari tambah, ubah, hapus dan cari data petugas. S*quence diagram* dari masing-masing *case* tersebut di antaranya:

1. *Sequence Diagram* Cari Petugas



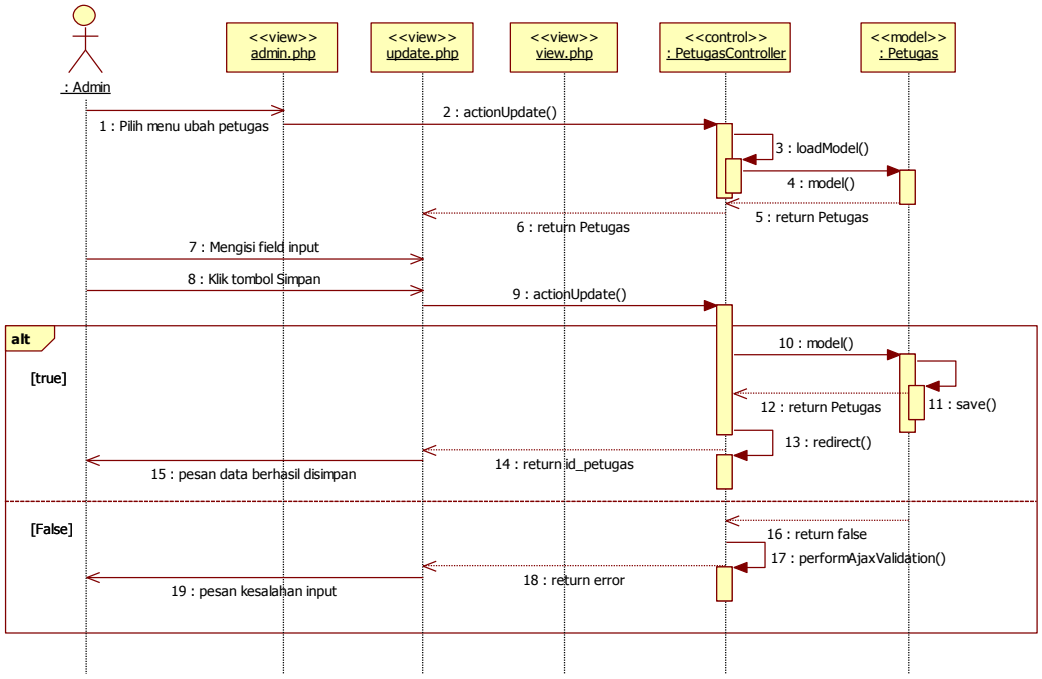
Gambar 3.41 *Sequence Diagram* Cari Petugas

2. *Sequence Diagram* Tambah Petugas



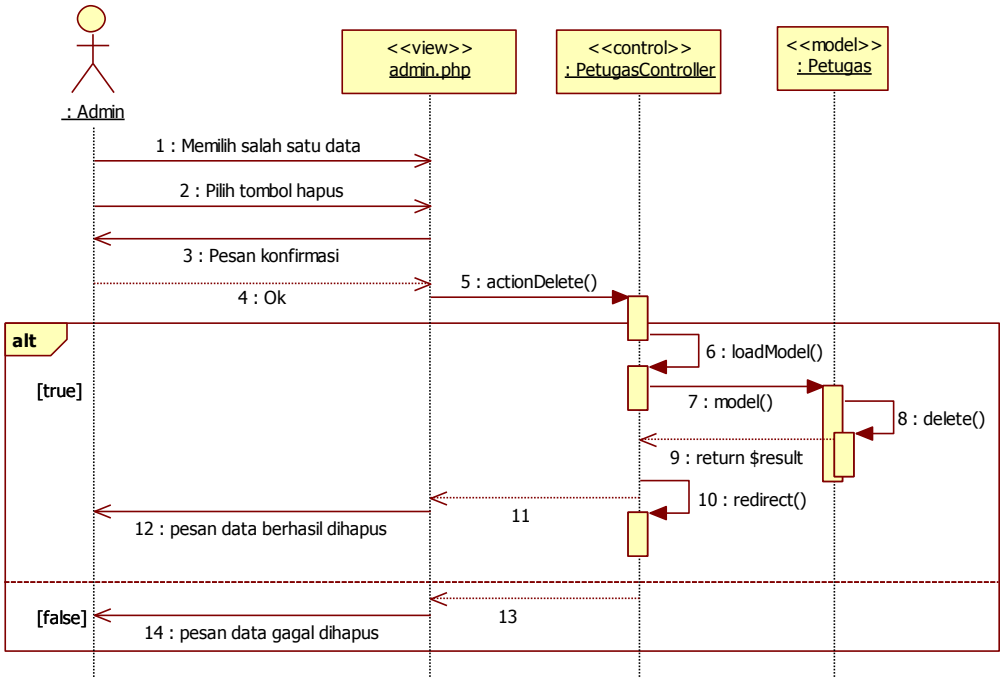
Gambar 3.42 *Sequence Diagram* Tambah Petugas

3. *Sequence Diagram* Ubah Petugas



Gambar 3.43 *Sequence Diagram* Ubah Petugas

4. *Sequence Diagram* Hapus Petugas

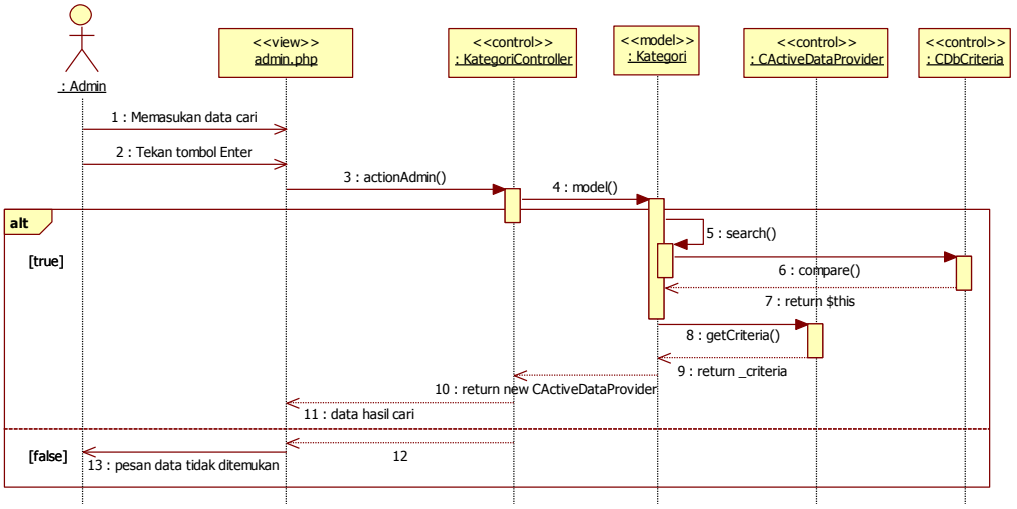


Gambar 3.44 *Sequence Diagram* Hapus Petugas

3.1.2.12.3 *Sequence Diagram* Pengolahan Data Kategori

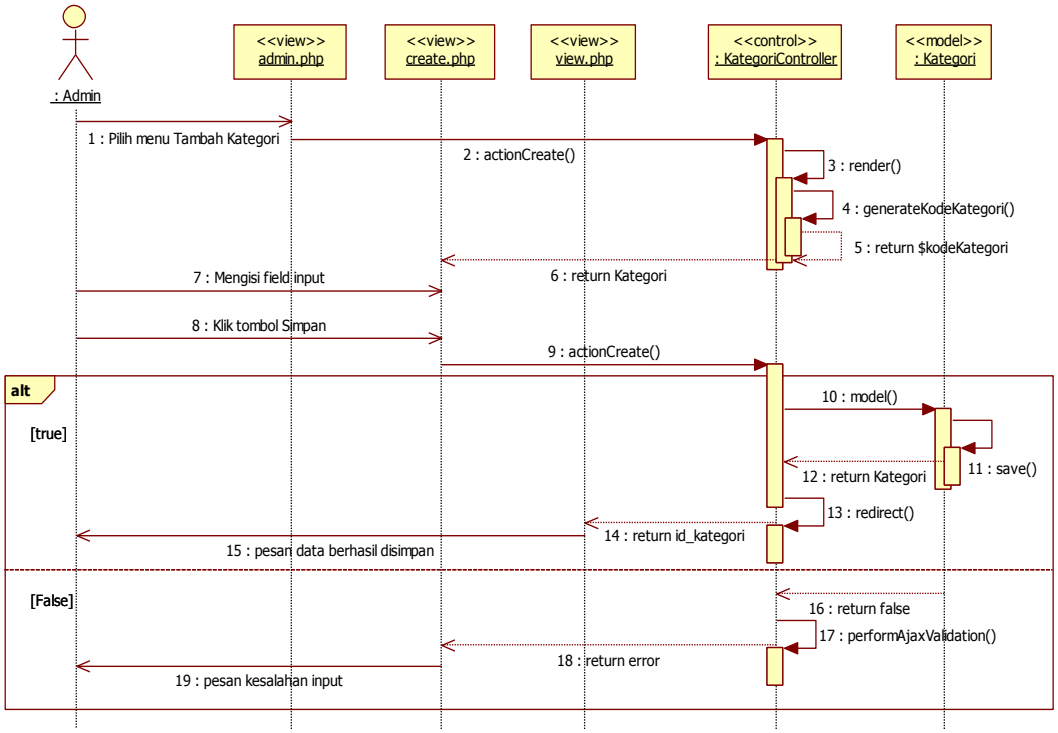
Sequence diagram pengolahan data kategori terdiri dari cari, hapus, tambah, dan ubah. S*quence diagram* dari masing-masing *case* tersebut di antaranya:

1. *Sequence Diagram* Cari Kategori



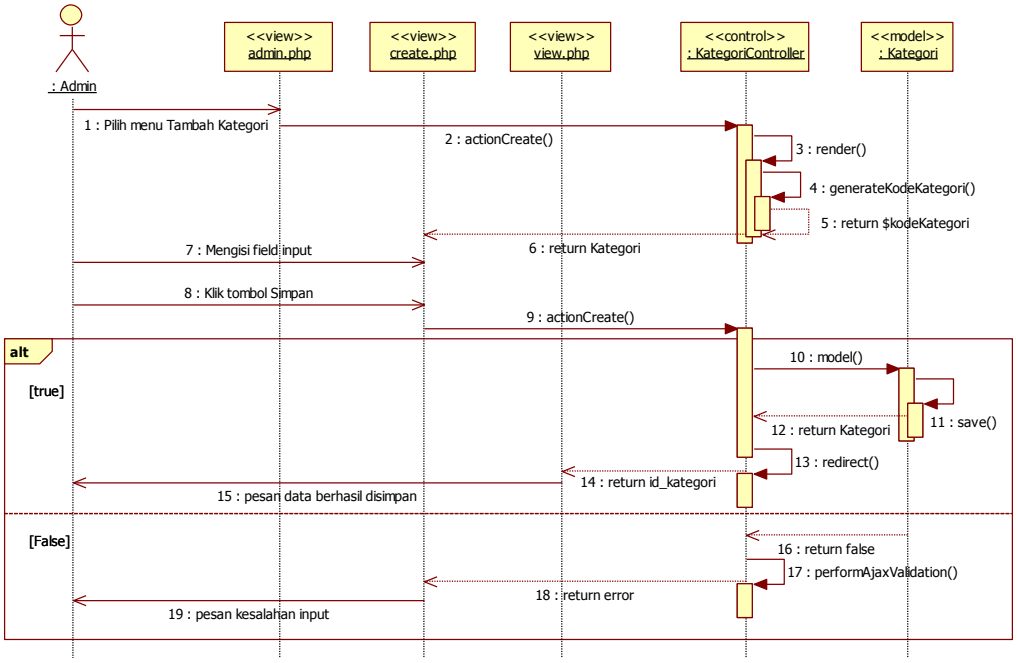
Gambar 3.45 *Sequence Diagram* Cari Kategori

2. *Sequence Diagram* Tambah Kategori



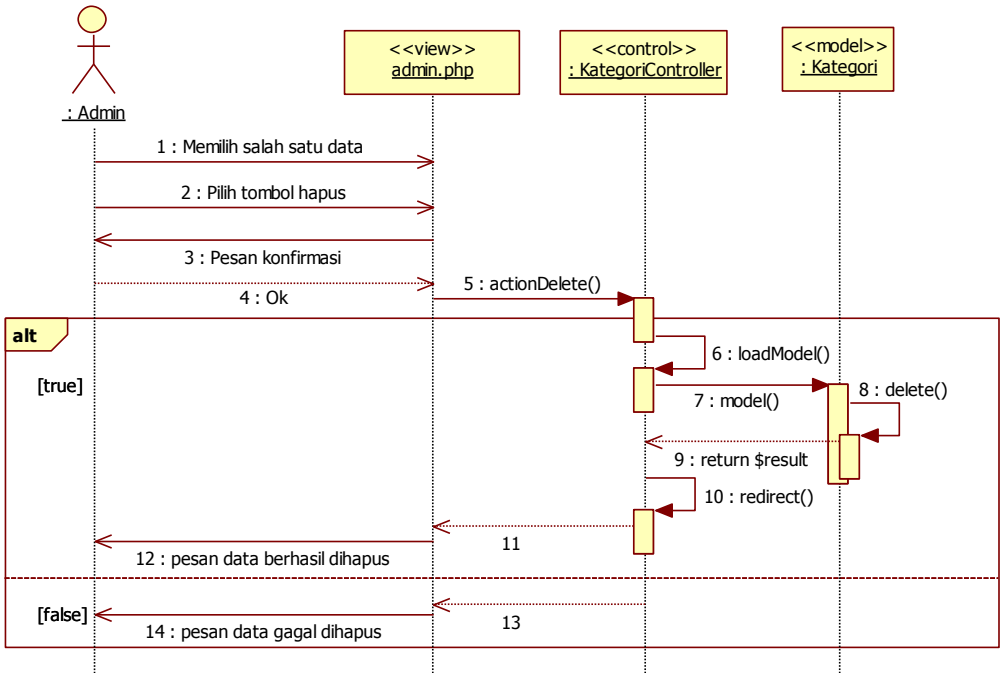
Gambar 3.46 *Sequence Diagram* Tambah Kategori

3. *Sequence Diagram* Ubah Kategori



Gambar 3.47 *Sequence Diagram* Ubah Kategori

4. *Sequence Diagram* Hapus Kategori

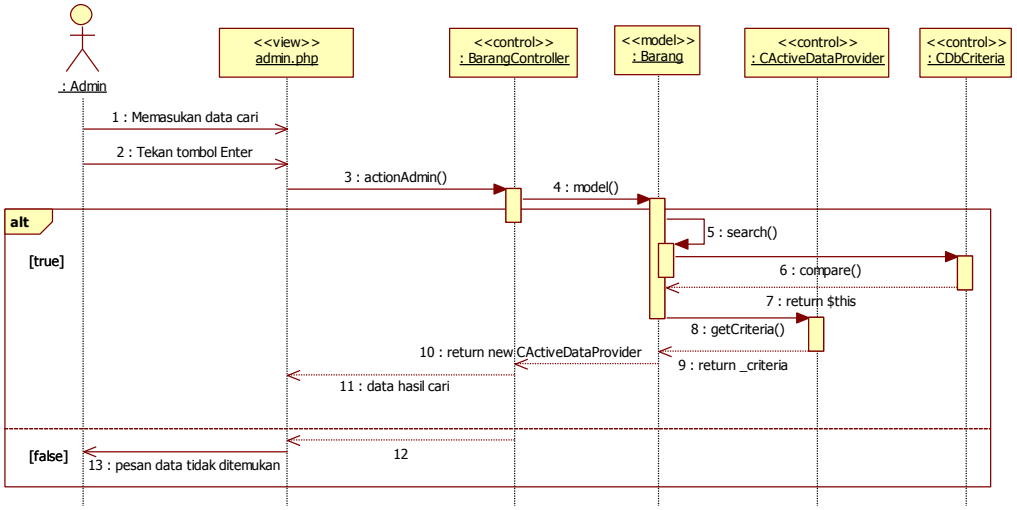


Gambar 3.48 *Sequence Diagram* Hapus Kategori

3.1.2.12.4 *Sequence Diagram* Pengolahan Data Barang

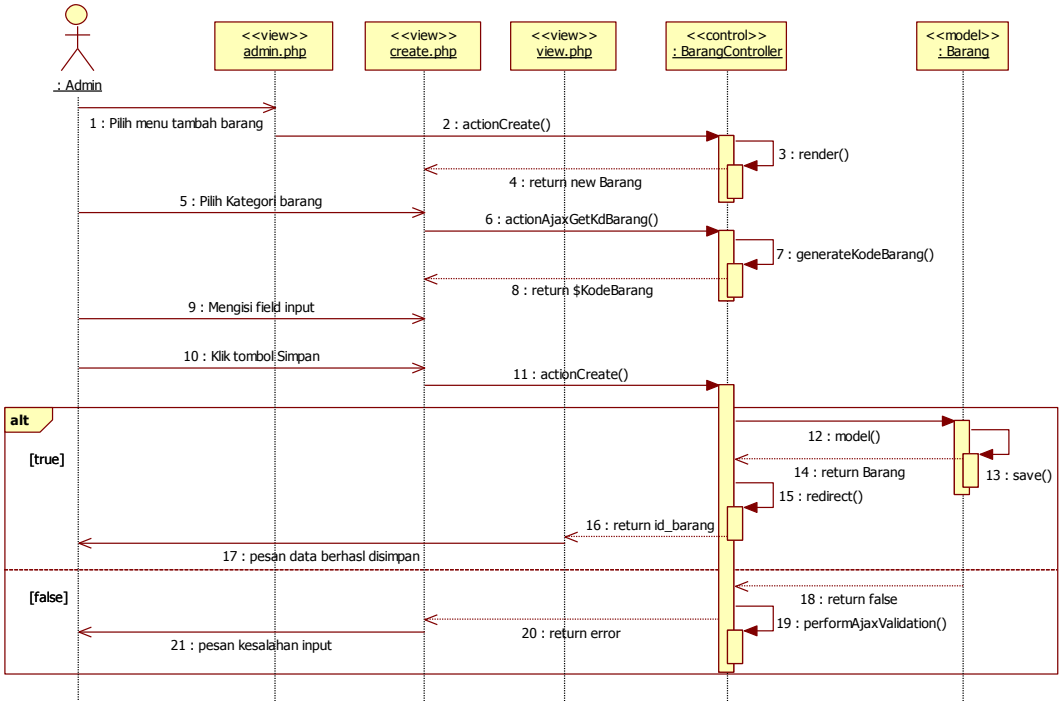
Squence diagram pengolahan data barang terdiri dari cari, hapus, tambah, dan ubah data. S*quence diagram* dari masing-masing *case* tersebut di antaranya:

1. *Sequence Diagram* Cari Barang



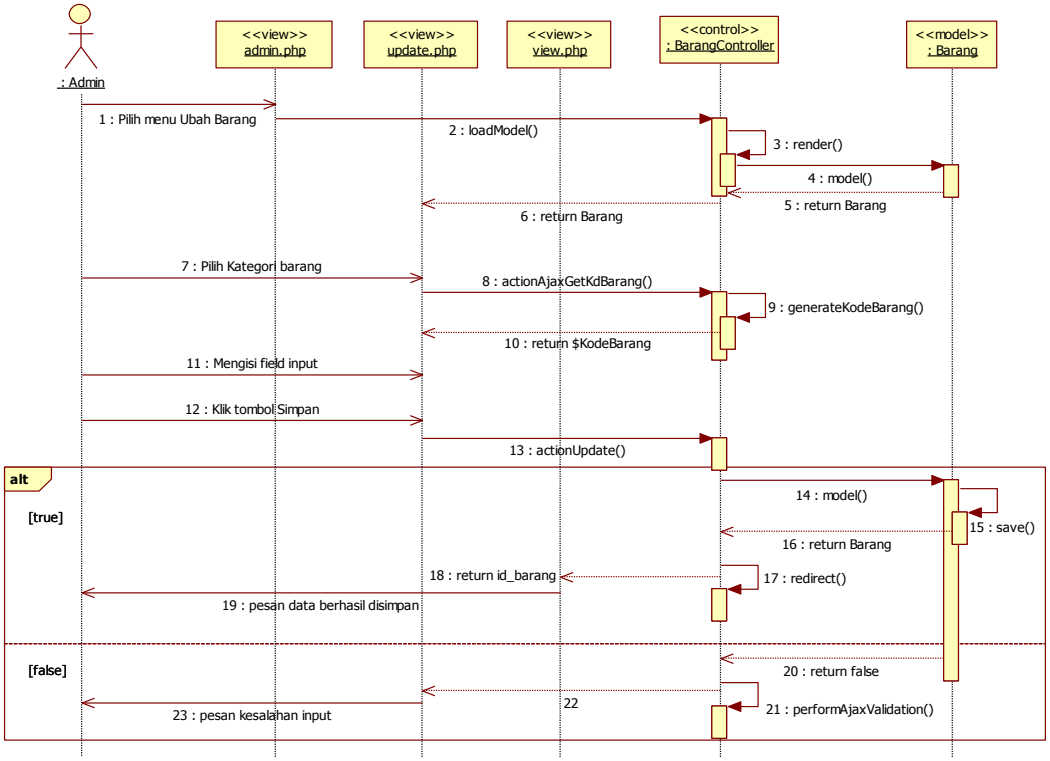
Gambar 3.49 *Sequence Diagram* Cari Barang

2. *Sequence Diagram* Tambah Barang



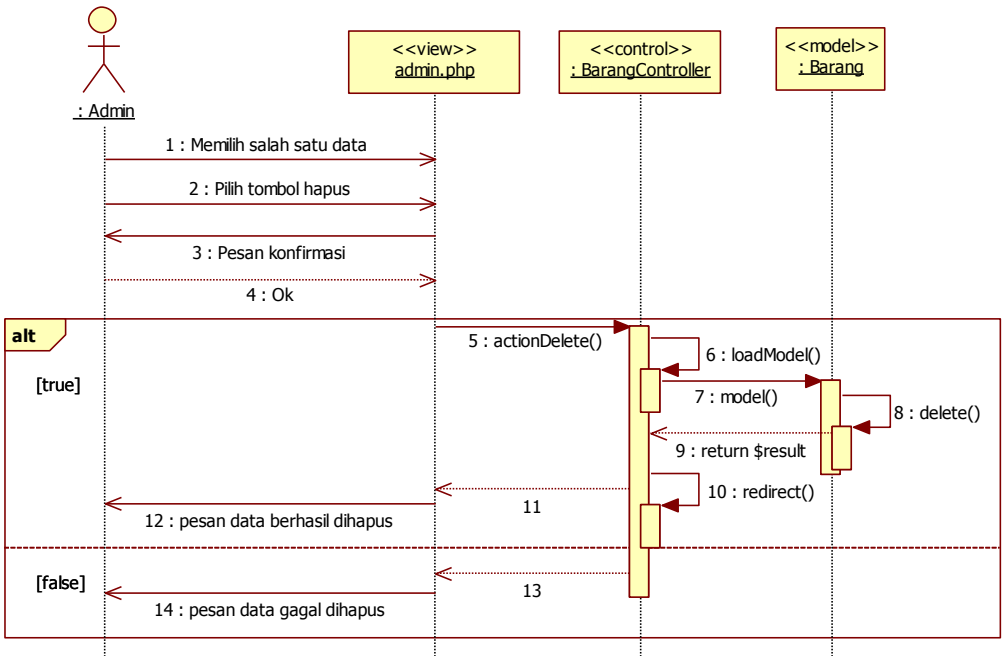
Gambar 3.50 *Sequence Diagram* Tambah Barang

3. *Sequence Diagram* Ubah Barang



Gambar 3.51 *Sequence Diagram* Ubah Barang

4. *Sequence Diagram* Hapus Barang

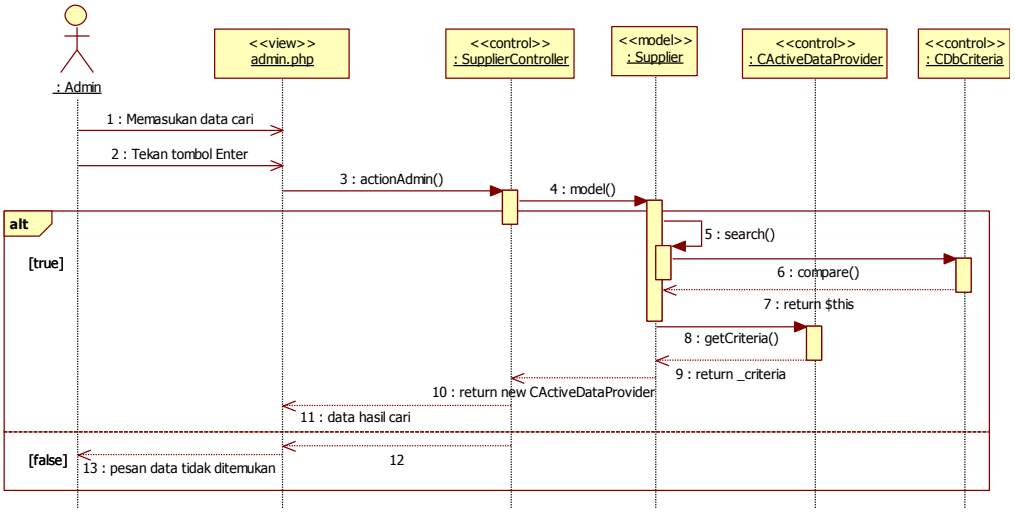


Gambar 3.52 *Sequence Diagram* Hapus Data Barang

3.1.2.12.5 *Sequence Diagaram* Pengolahan Data *Supplier*

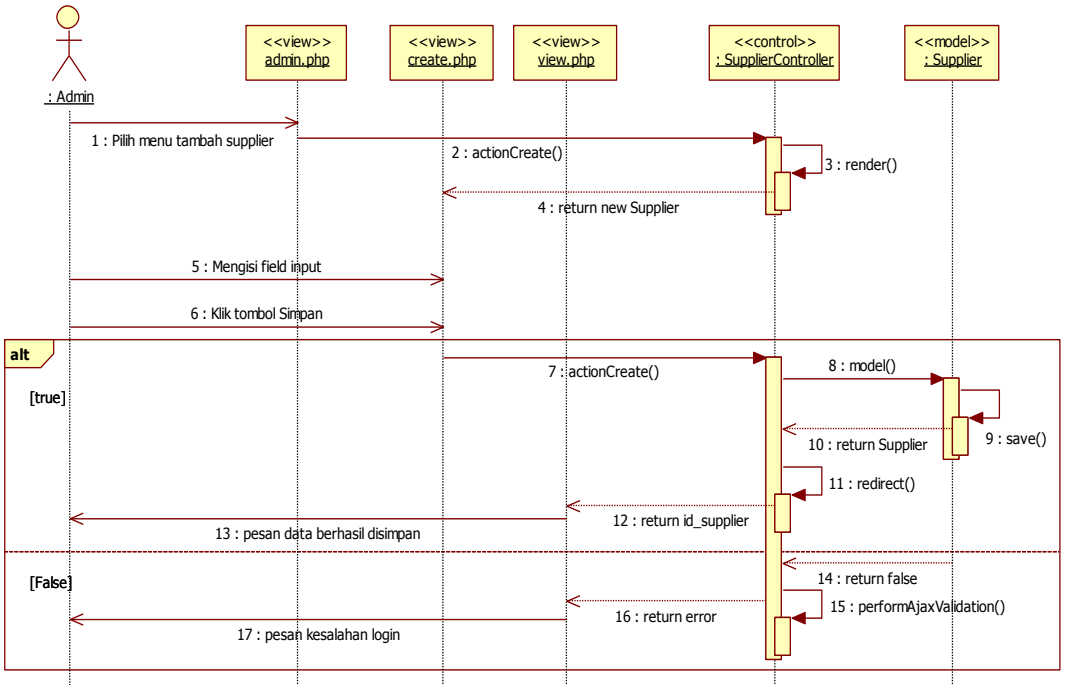
Sequence diagram pengolahan data *supplier* terdiri dari cari, hapus, tambah dan ubah data. S*quence diagram* dari masing-masing *case* tersebut di antaranya:

1. *Sequence Diagram* Cari *Supplier*



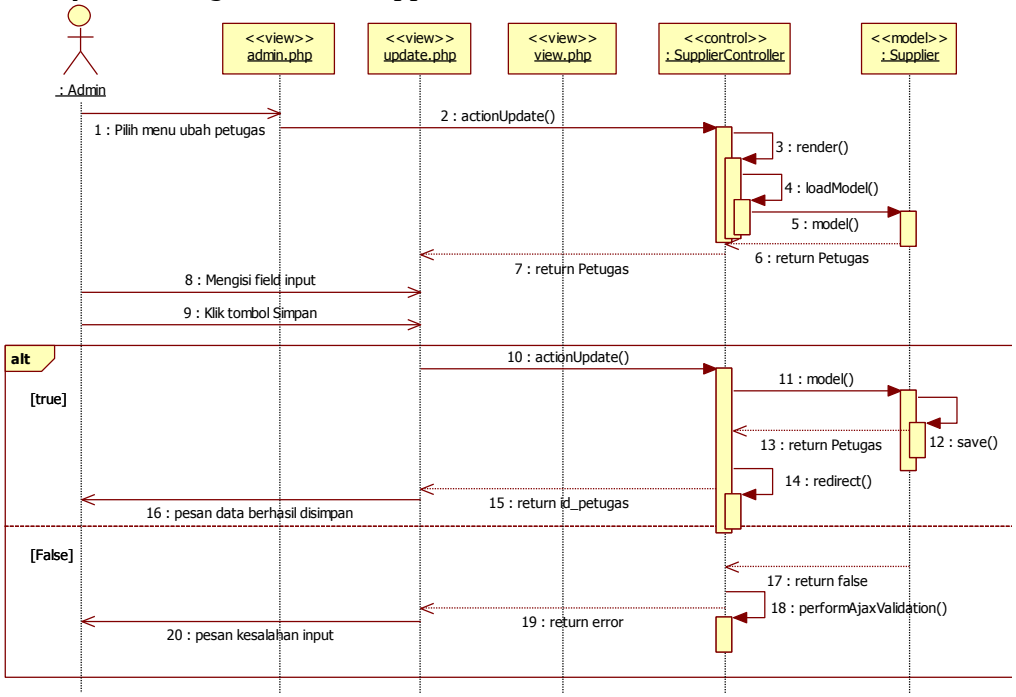
Gambar 3.53 *Sequence Diagram* Cari *Supplier*

2. *Sequence Diagram* Tambah *Supplier*



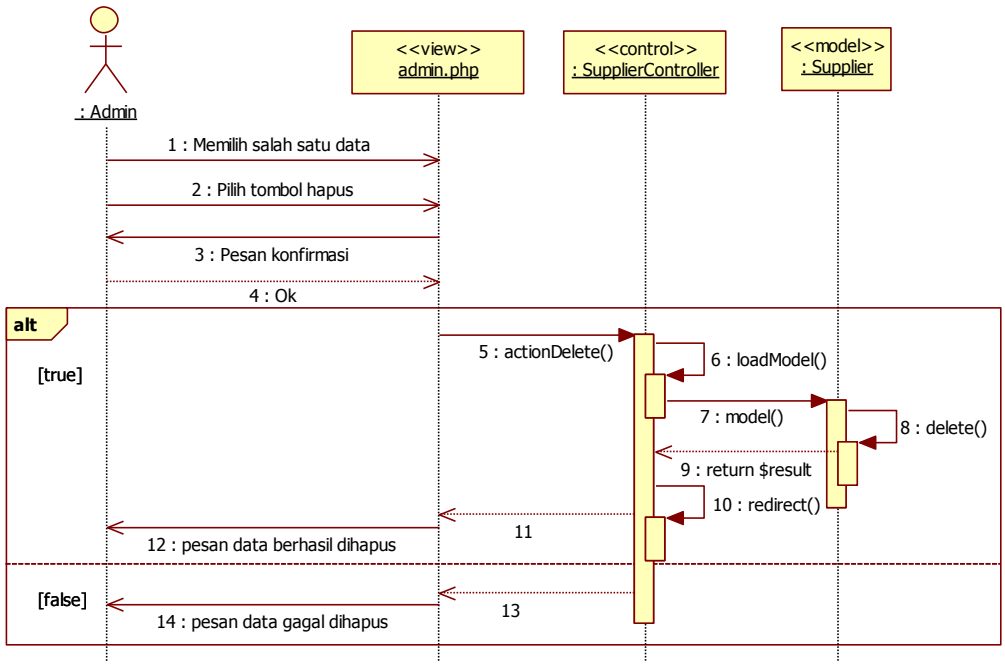
Gambar 3.54 *Sequence Diagram* Tambah *Supplier*

3. *Sequence Diagram* Ubah *Supplier*



Gambar 3.55 *Sequence Diagram* Ubah *Supplier*

4. *Sequence Diagram* Hapus *Supplier*

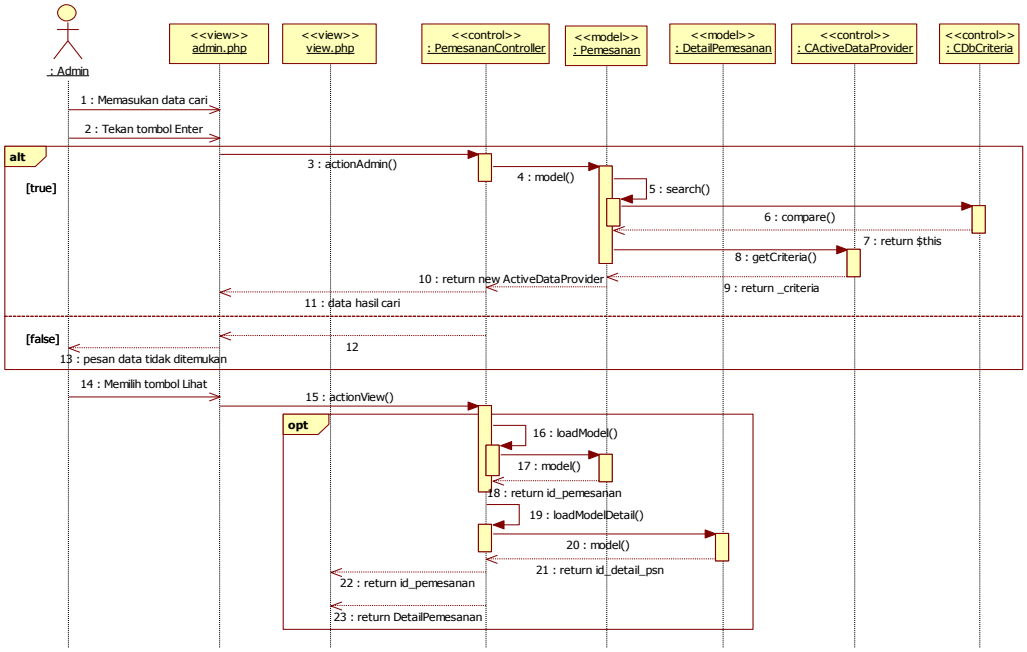


Gambar 3.56 *Sequence Diagram* Hapus *Supplier*

3.1.2.12.6 *Sequence Diagram* Pengolahan Data Pemesanan

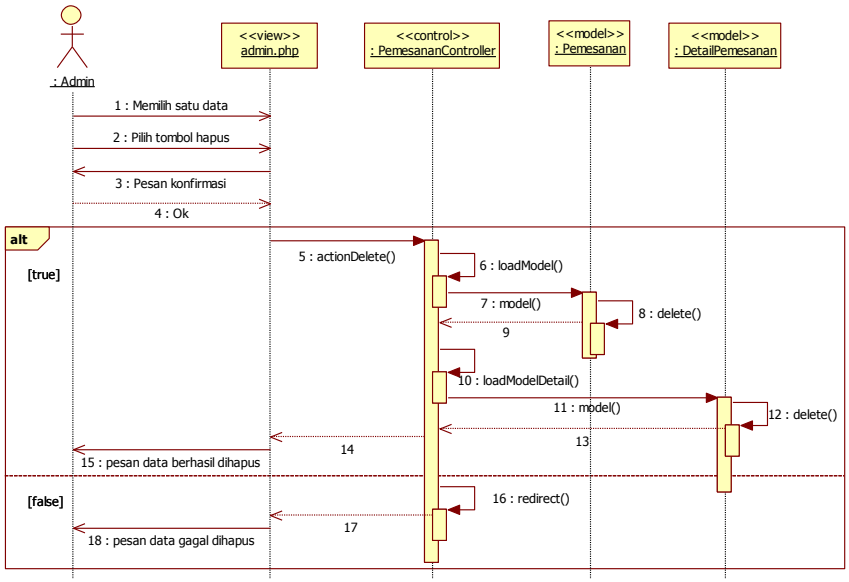
Sequence diagram pengolahan data pemesanan terdiri dari cari, hapus, tambah dan ubah. S*quence diagram* dari masing-masing *case* tersebut di antaranya:

1. *Sequence Diagram* Cari Pemesanan



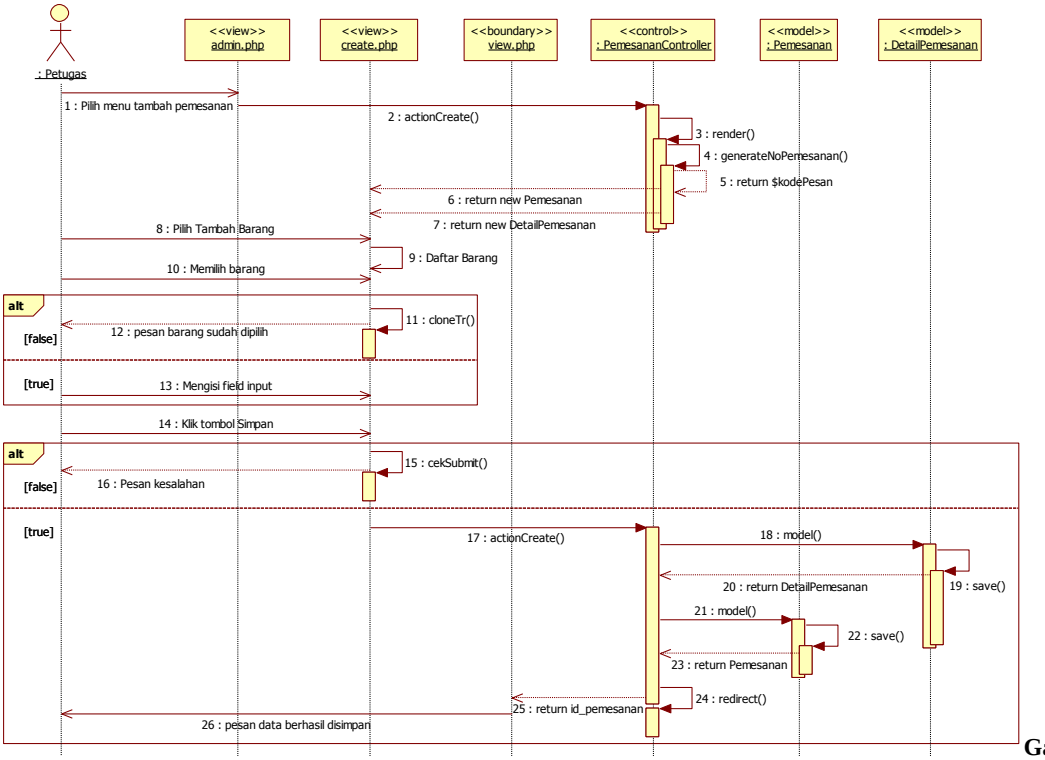
Gambar 3.57 *Sequence Diagram* Cari Pemesanan

2. *Sequence Diagram* Hapus Pemesanan



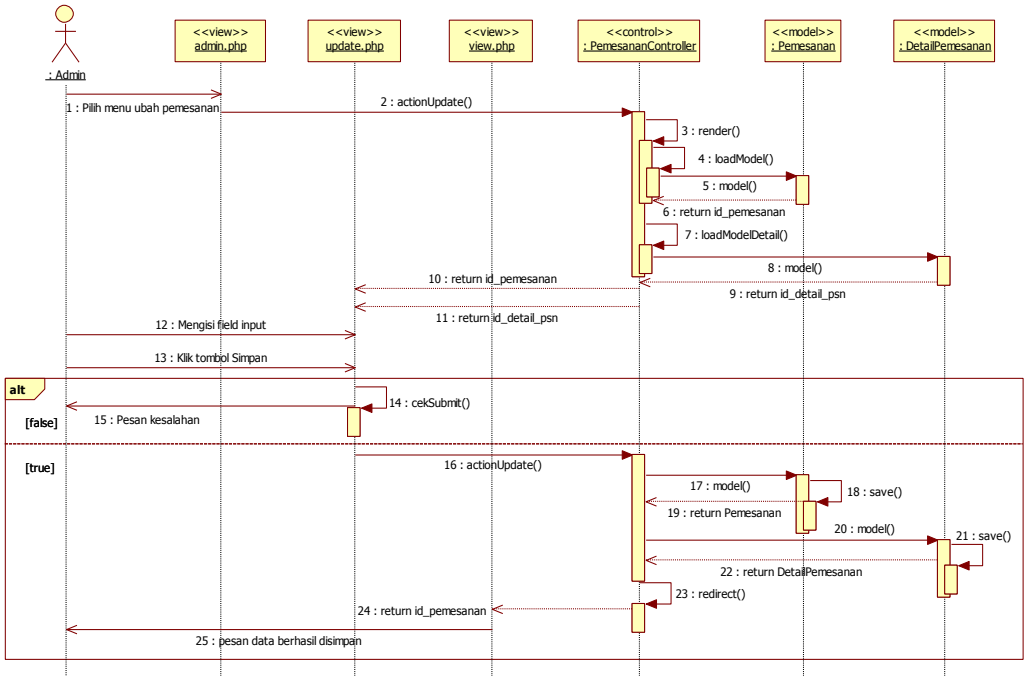
Gambar 3.58 *Sequence Diagram* Hapus Pemesanan

3. *Sequence Diagram* Tambah Pemesanan



Gambar 3.59 *Sequence Diagram* Tambah Pemesanan

4. *Sequence Diagram* Ubah Pemesanan

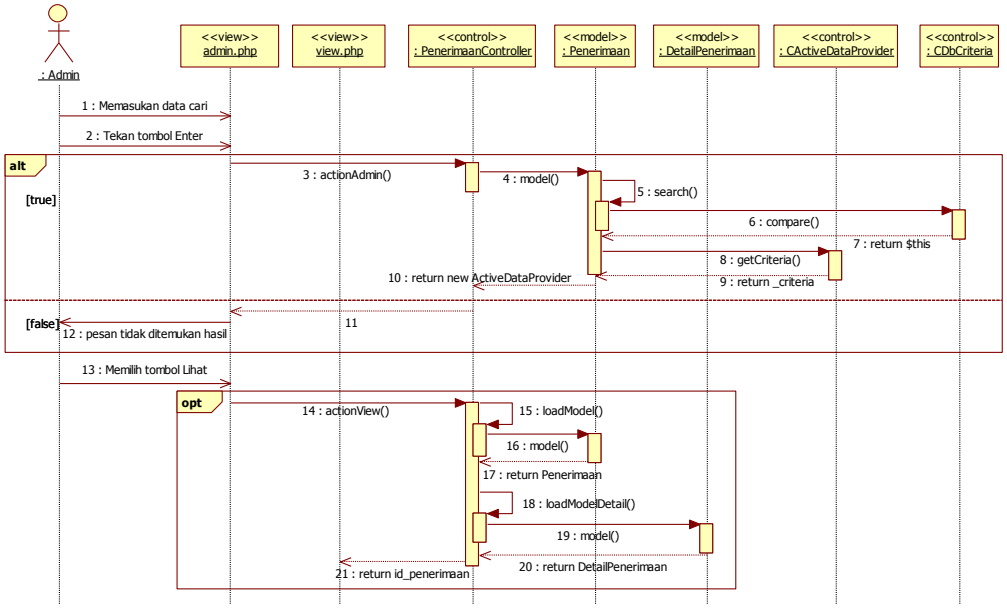


Gambar 3.60 *Sequence Diagram* Ubah Pemesanan

3.1.2.12.7 *Sequence Diagram* Pengolahan Data Penerimaan

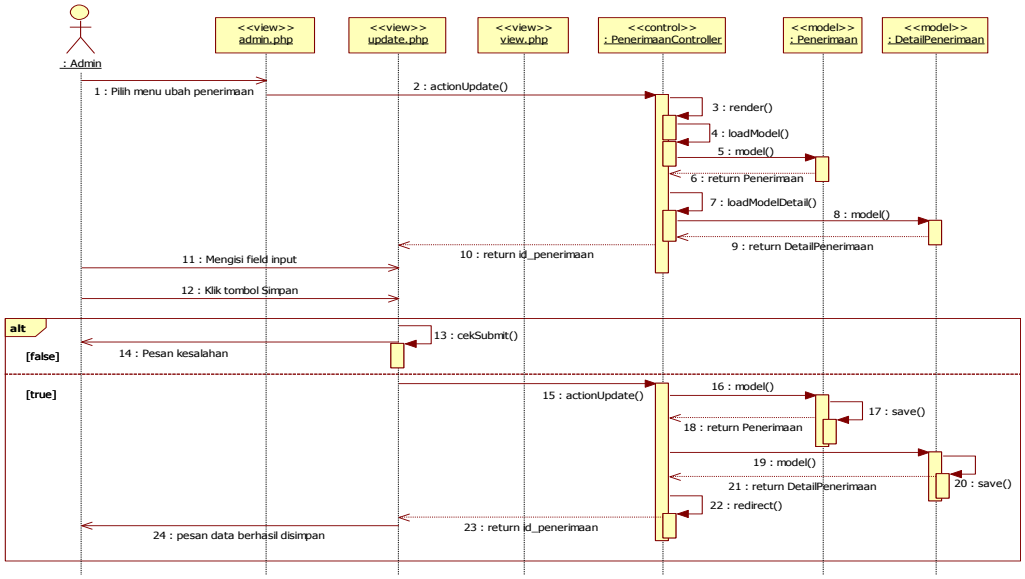
Sequence diagram pengolahan data penerimaan barang terdiri dari cari data, hapus data, tambah data dan ubah data.

1. *Sequence Diagram* Cari Penerimaan



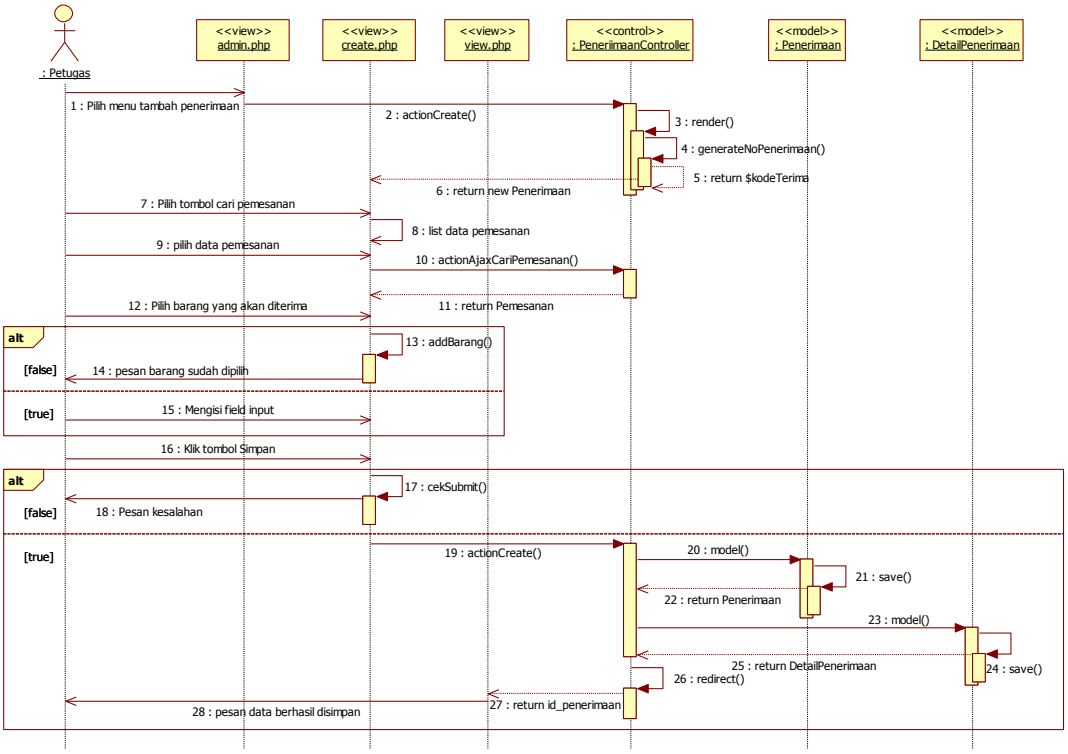
Gambar 3.61 *Sequence Diagram* Cari Penerimaan

2. *Sequence Diagram* Ubah Penerimaan



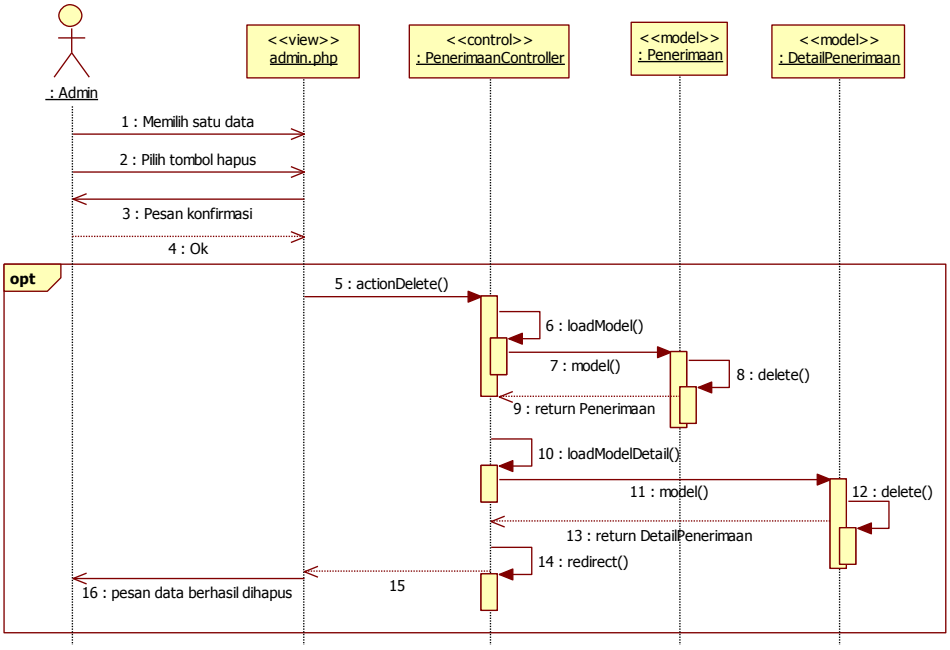
Gambar 3.62 *Sequence Diagram* Ubah Penerimaan

3. *Sequence Diagram* Tambah Penerimaan



Gambar 3.63 *Sequence Diagram* Tambah Penerimaan

4. *Sequence Diagram* Hapus Penerimaan

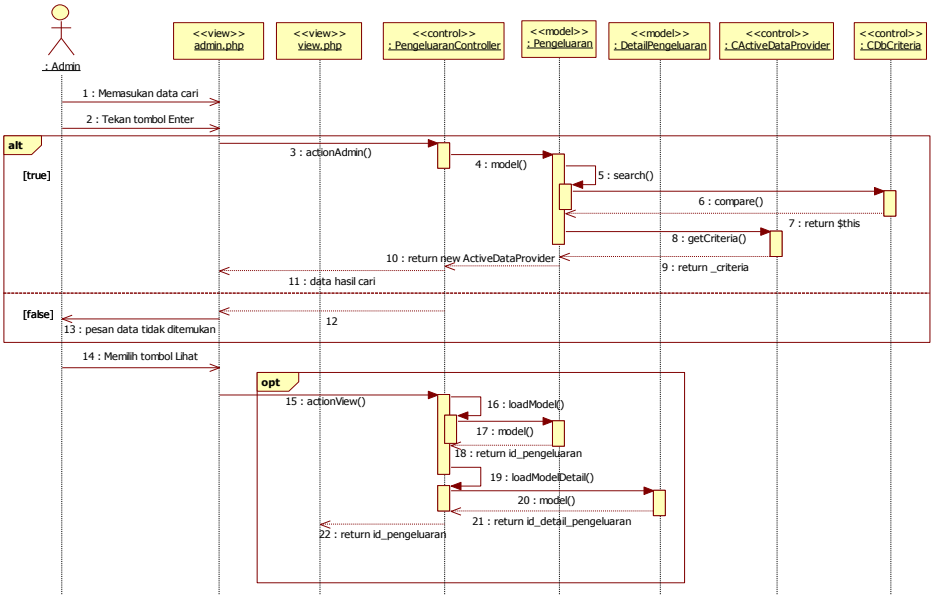


Gambar 3.64 *Sequence Diagram* Hapus Penerimaan

3.1.2.12.8 *Sequence Diagram* Pengolahan Data Pengeluaran

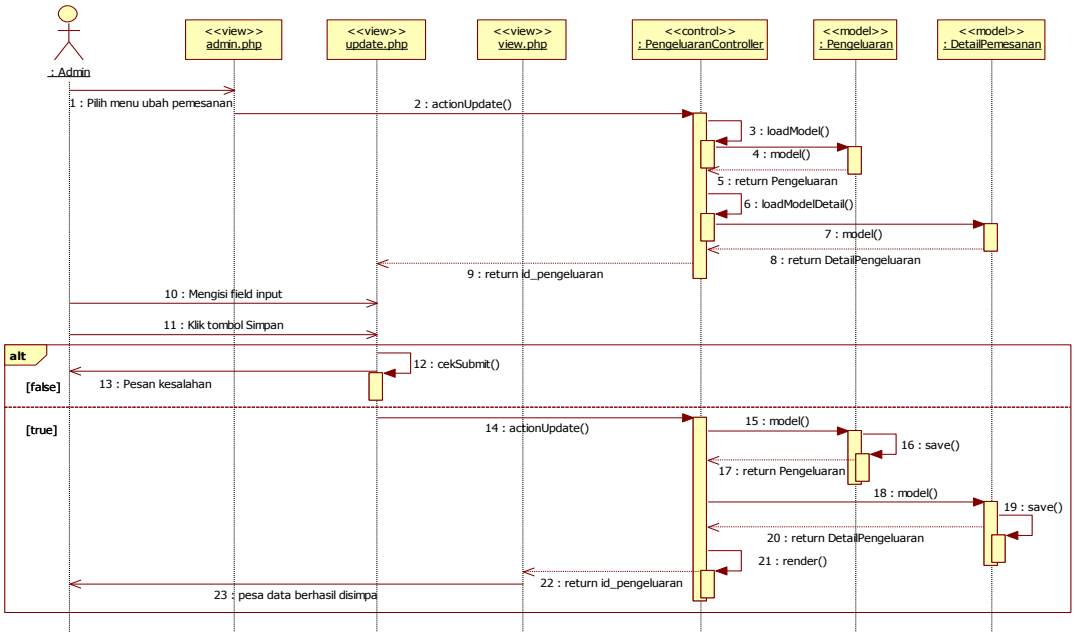
Sequence diagram pengolahan data pengeluaran terdiri dari cari data, hapus data, tambah data, dan ubah data.

1. *Sequence Diagram* Cari Pengeluaran



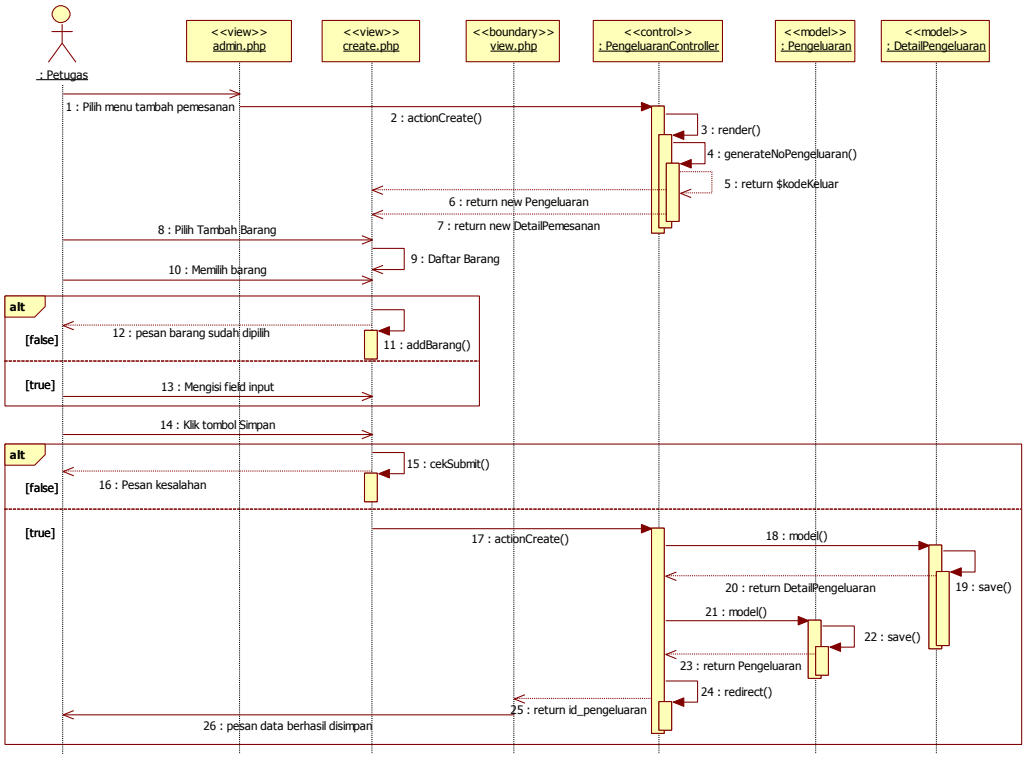
Gambar 3.65 *Sequence Diagram* Cari Pengeluaran

2. *Sequence Diagram* Ubah Pengeluaran



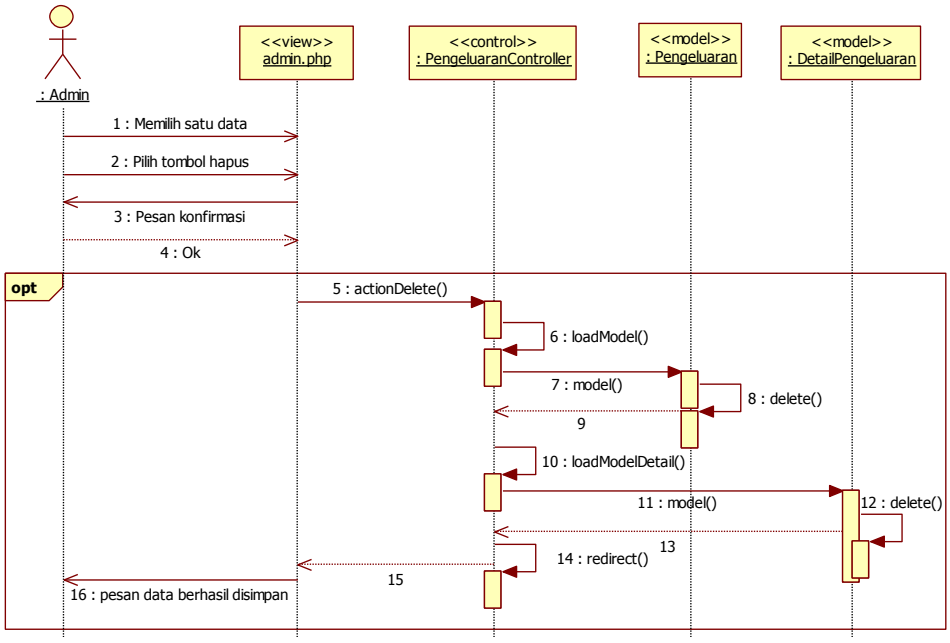
Gambar 3.66 *Sequence Diagram* Ubah Pengeluaran

3. *Sequence Diagram* Tambah Pengeluaran



Gambar 3.67 *Sequence Diagram* Tambah Pengeluaran

4. *Sequence Diagram* Hapus Pengeluaran

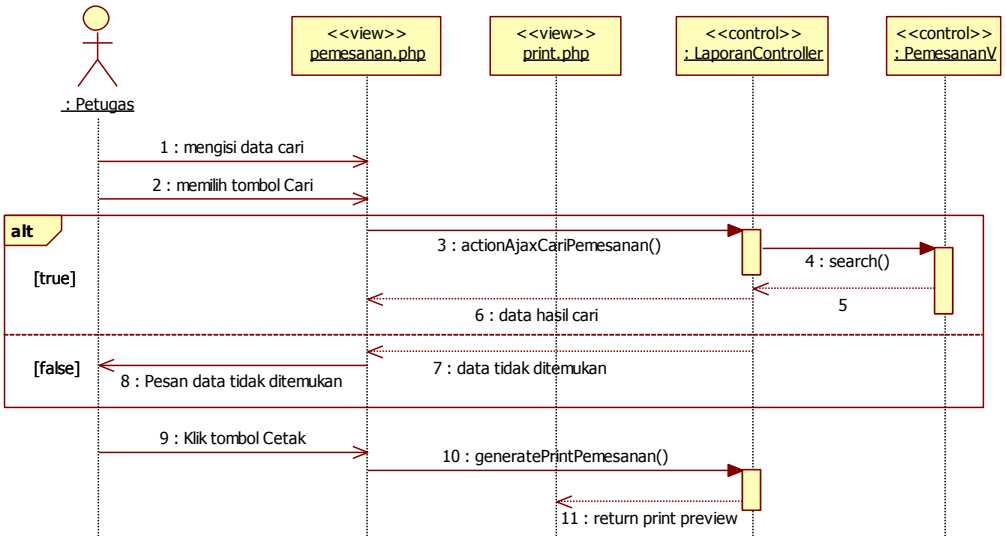


Gambar 3.68 *Sequence Diagram* Hapus Pengeluaran

3.1.2.12.9 *Sequence Diagram* Pengolahan Laporan

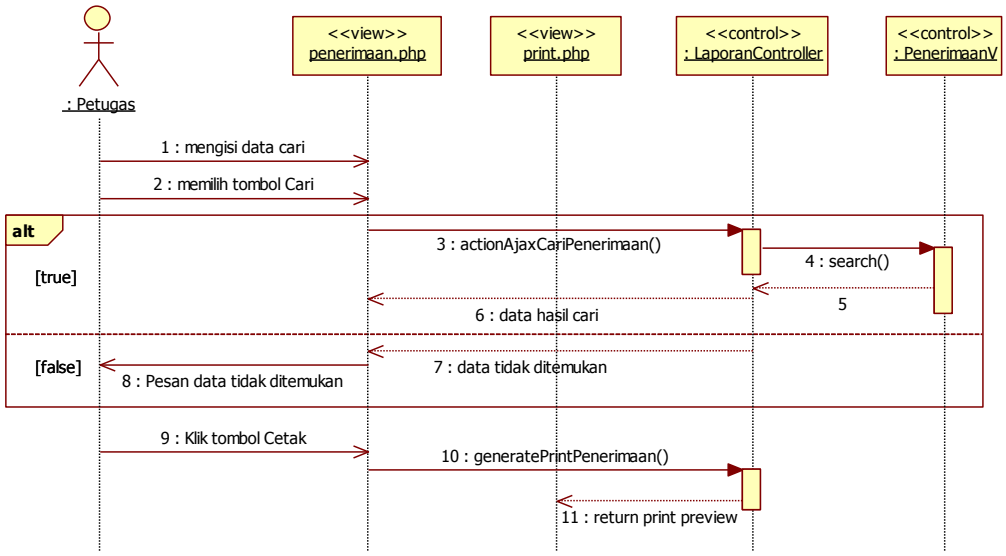
Sequence diagram pengolahan laporan terdiri dari laporan pemesanan, penerimaan dan laporan pengeluaran.

1. *Sequence Diagram* Laporan Pemesanan



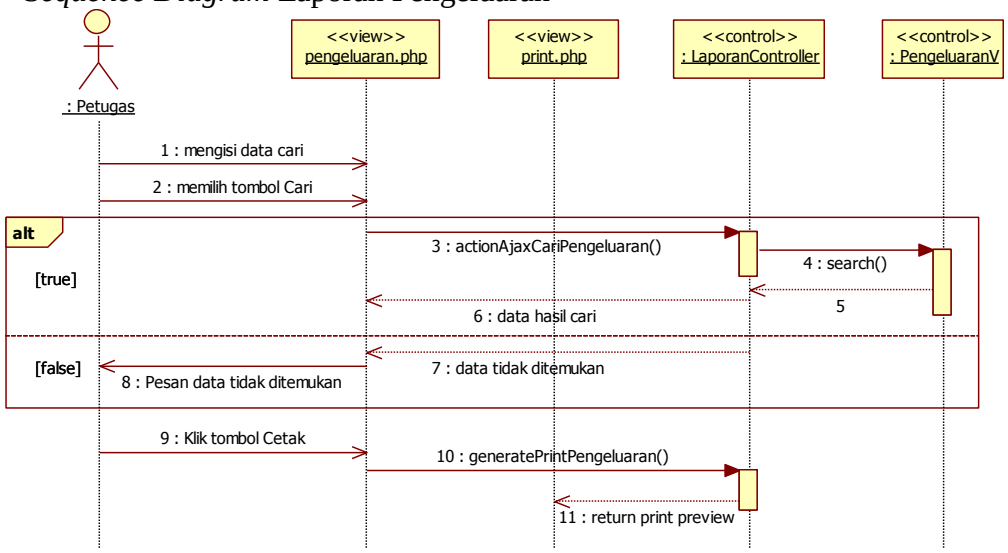
Gambar 3.69 *Sequence Diagram* Laporan Pemesanan

2. *Sequence Diagram* Laporan Penerimaan



Gambar 3.70 *Sequence Diagram* Laporan Penerimaan

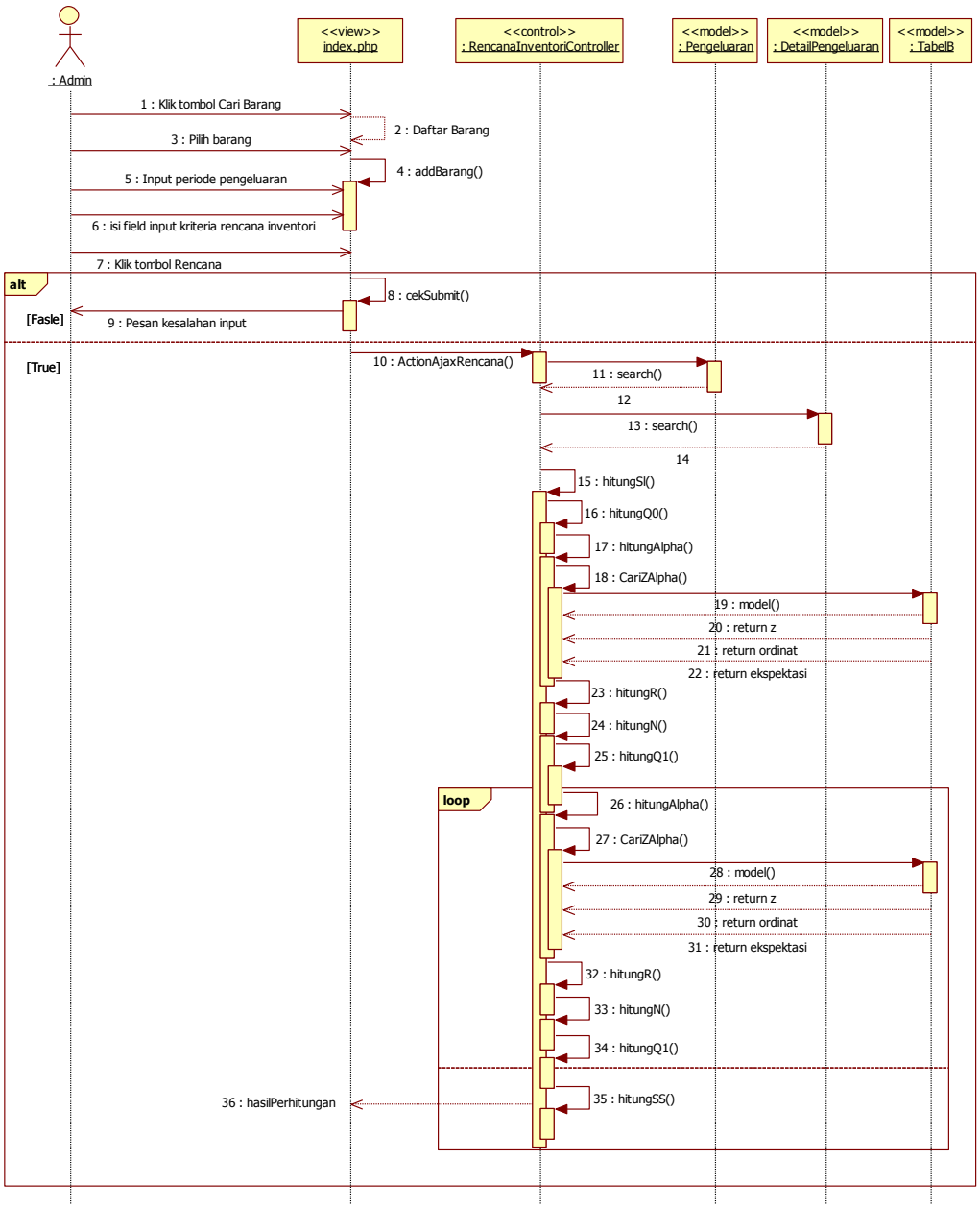
3. *Sequence Diagram* Laporan Pengeluaran



Gambar 3.71 *Sequence Diagram* Laporan Pengeluaran

3.1.2.12.10 *Sequence Diagram* Perhitungan Rencana Inventori

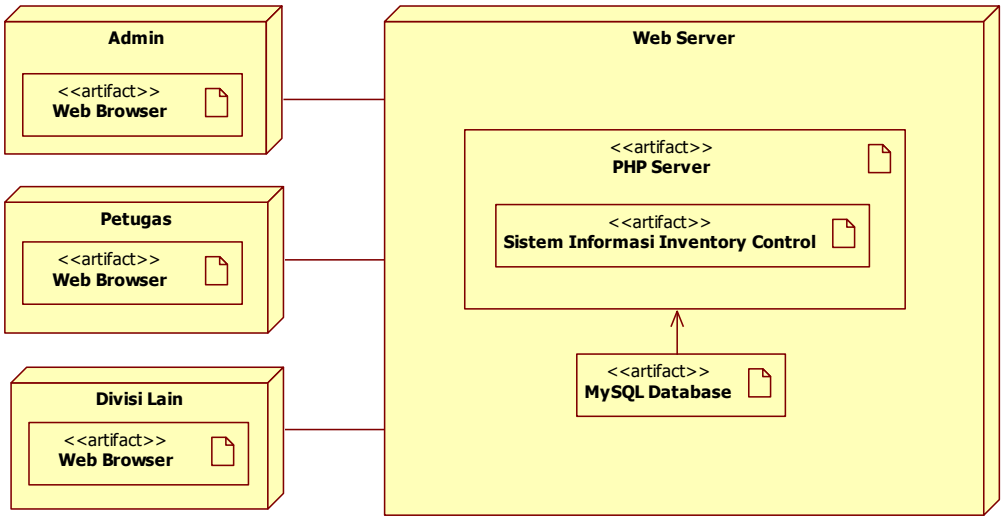
Sequence diagram pengolahan data rencana inventori (rencana persediaan) dapat dilihat seperti yang tertera pada Gambar 3.72.



Gambar 3.72 *Squence Diagram* Pengolahan Data Rencana Inventori

3.1.2.13 Deployment Diagram

Deployment diagram adalah diagram yang mendeskripsikan arsitektur fisik dari *hardware* dan *software* di dalam sistem. *Deployment diagram* pada sistem informasi *inventory control* dapat dilihat pada Gambar 3.73.



Gambar 3.73 *Deployment Diagram* Sistem *Inventory Control*

3.2 Perancangan Sistem

Perancangan sistem merupakan bagian dari metedologi penelitian dan pengembangan suatu perangkat lunak yang dilakukan setelah tahap analisis yang bertujuan untuk memberikan gambaran secara terperinci.

3.2.1 Perancangan Basis Data

Perancangan basis data adalah perancangan data yang yang tersimpan dalam media penyimpanan dan perangkat lunak yang akan memanipulasinya. Perancangan basis data terdiri dari skema relasi, diagram relasi, dan struktur tabel.

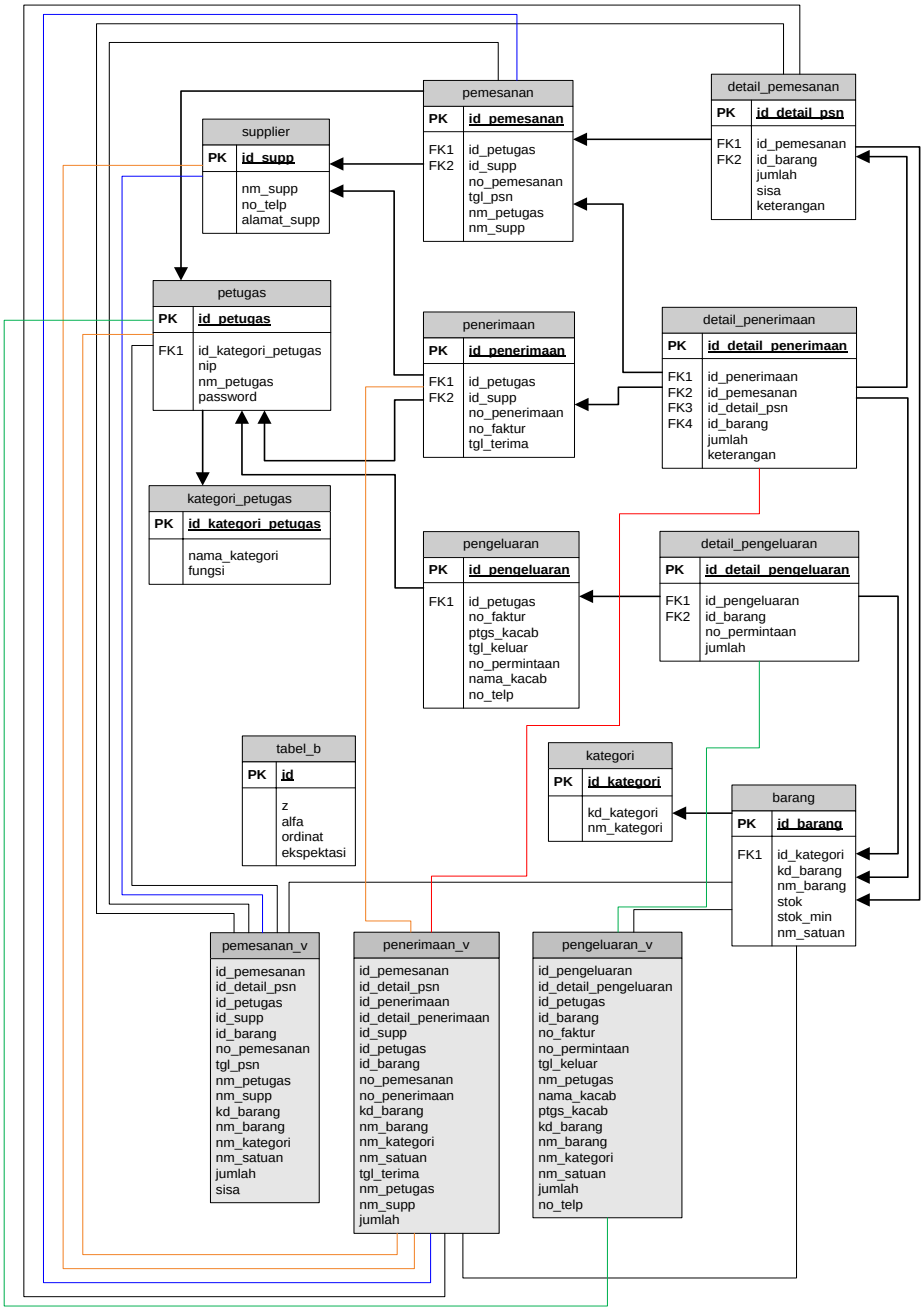
3.2.1.1 Skema Relasi

Skema relasi adalah relasi yang memiliki nama, didefinisikan oleh himpunan pasangan atribut dan domainnya. Skema relasi dari basis data yang terdapat di dalam sistem informasi *inventory control* adalah sebagai berikut.

kategori	(id_kategori, kd_kategori, nm_kategori),
barang	(id_barang, kd_barang, nm_barang, stok, stok_min, id_kategori, nm_satuan),
petugas	(id_petugas, nip, nm_petugas, password, id_kategori_petugas),
kategori_petugas	(id_kategori_petugas, nama_kategori, fungsi),
supplier	(id_supp, nm_supp, no_tlp, alamat_supp),
pemesanan	(id_pemesanan, no_pemesanan, tgl_psn, id_petugas, id_supp),
penerimaan	(id_penerimaan, no_penerimaan, no_faktur, tgl_terima, id_petugas, id_supp),
pengeluaran	(id_pengeluaran, no_faktur, no_permintaan, tgl_keluar, id_petugas, nm_petugas, nama_kacab, ptgs_kacab, no_telp),
detail_pemesanan	(id_detail_psn, id_pemesanan, id_barang, jumlah, sisa, keterangan),
detail_penerimaan	(id_detail_penerimaan, id_detail_psn, id_pemesanan, id_penerimaan, id_barang, jumlah, keterangan),
detail_pengeluaran	(id_detail_pengeluaran, id_pengeluaran, id_barang, jumlah, sisa),
tabel_b	(id, z, alfa, ordinat, ekspektasi),
pemesanan_v	(id_detail_psn, id_pemesanan, id_petugas, id_supp, no_pemsesanan,tgl_psn,id_barang, kd_barang, nm_barang, nm_kategori, nm_satuan, jumlah, sisa),
penerimaan_v	(id_detail_penerimaan, id_pemesanan, id_pemesanan, id_petugas, id_supp, no_pemsesanan, no_penerimaan, tgl_terima, id_barang, kd_barang, nm_barang, nm_kategori, nm_satuan, jumlah),
pengeluaran_v	(id_detail_pengeluaran, id_pengeluaran, id_petugas, nama_kacab, ptgs_kacab, no_telp, no_permintaan, no_pengeluaran, tgl_keluar, id_barang, kd_barang, nm_barang, nm_kategori, nm_satuan, jumlah, sisa)

3.2.1.2 Diagram Relasi

Diagram relasi adalah diagram yang menggambarkan hubungan antar data dalam suatu basis data berdasarkan objek-objek dasar data. Diagram relasi pada basis data sistem informasi *inventory control* dapat dilihat pada Gambar 3.74.



Gambar 3.74 Diagram Relasi

3.2.1.3 Struktur Tabel

Tabel yang terdapat pada sistem yang akan dibangun terdiri dari 15 tabel. Struktur dari tabel-tabel tersebut adalah sebagai berikut:

1. Tabel kategori

Tabel kategori digunakan untuk menyimpan data kategori. Struktur dari tabel kategori dijelaskan pada Tabel 3.78.

Tabel 3.78 Struktur Tabel kategori

Nama Field	Tipe	Panjang	Kunci	Keterangan
id_kategori	Int	2	*Primary Key*	*Not Null, Autoincrement*
kd_kategori	Varchar	15		*Not Null*
nm_kategori	Varchar	50		*Not Null*

2. Tabel barang

Tabel barang digunakan untuk menyimpan data barang. Struktur dari tabel barang dijelaskan pada Tabel 3.79.

Tabel 3.79 Struktur Tabel barang

Nama Field	Tipe	Panjang	Kunci	Keterangan
id_barang	Int	10	*Primary Key*	*NotNull, Autoincrement*
id_kategori	Int	2	*Foreign Key references* kategori (id_kategori)	*Null*
kd_barang	Varchar	15		*Not Null*
nm_barang	Varchar	50		*Not Null*
stok	Int	11		*Not Null*
stok_min	Int	11		*Not Null*
nm_satuan	Varchar	20		*Not Null*

3. Tabel supplier

Tabel supplier digunakan untuk menyimpan data *supplier*. Struktur dari tabel supplier dijelaskan pada Tabel 3.80.

Tabel 3.80 Struktur Tabel supplier

Nama Field	Tipe	Panjang	Kunci	Keterangan
id_supp	Int	5	*Primary Key*	*NotNull, Autoincrement*
nm_supp	Varchar	50		*Not Null*
no_tlp	Varchar	15		*Not Null*
alamat_supp	Text			*Not Null*

4. Tabel petugas

Tabel petugas digunakan untuk menyimpan data petugas. Struktur dari tabel petugas dijelaskan pada Tabel 3.81.

Tabel 3.81 Struktur Tabel petugas

Nama Field	Tipe	Panjang	Kunci	Keterangan
id_petugas	Int	5	*Primary Key*	*NotNull, Autoincrement*
id_kategori_petugas	Int	5	*Foreign Key references* kategori_petugas (id_kategori_petugas)	*Null*
nip	Varchar	10		*Not Null*
nm_petugas	Varchar	20		*Not Null*
password	Varchar	50		*Not Null*

5. Tabel kategori_petugas

Struktur dari tabel kategori_petugas dapat dijelaskan pada Tabel 3.82 berikut.

Tabel 3.82 Struktur Tabel kategori_petugas

Nama Field	Tipe	Panjang	Kunci	Keterangan
id_kategori_petugas	Int	5	*Primary Key*	*NotNull, autoincrement*
nama_kategori	Varchar	15		*Not Null*
fungsi	Text			*Not Null*

6. Tabel pemesanan

Tabel pemesanan digunakan untuk menyimpan data pemesanan. Struktur dari tabel pemesanan Tabel 3.83 berikut.

Tabel 3.83 Struktur Tabel pemesanan

Nama Field	Tipe	Panjang	Kunci	Keterangan
id_pemesanan	Int	10	*Primary Key*	*NotNull, Autoincrement*
no_pemesanan	Varchar	20		*Not Null*
id_petugas	Int	5	*Foreign Key references* petugas (id_petugas)	*Null*
tgl_psn	Date			*Not Null*
id_supp	Int	5	*Foreign Key references* supplier (id_supp)	*Null*

7. Tabel detail_pemesanan

Tabel detail pemesanan digunakan untuk menyimpan data detail pemesanan. Struktur dari tabel detail_pemesanan Tabel 3.84 berikut.

Tabel 3.84 Struktur Tabel detail _pemesanan

Nama Field	Tipe	Panjang	Kunci	Keterangan
id_detail_psn	Int	10	*Primary Key*	*NotNull, autoincrement*
id_pemesanan	Int	10	*Foreign Key references* pemesanan (id_pemesanan)	*Null*
id_barang	Int	10	*Foreign Key references* barang (id_barang)	*Null*
ketarangan	Varchar	25		*Null*
jumlah	Int	11		*Null*
sisa	Int	11		*Null*

8. Tabel penerimaan

Tabel penerimaan digunakan untuk menyimpan data penerimaan yang dilakukakan oleh petugas. Struktur dari tabel pemesanan dijelaskan pada Tabel 3.85.

Tabel 3.85 Struktur Tabel penerimaan

Nama Field	Tipe	Panjang	Kunci	Keterangan
id_penerimaan	Int	10	*Primary Key*	*NotNull, autoincrement*
no_penerimaan	Varchar	20		
no_faktur	Varchar	20		
tgl_terima	Date			*Not Null*
id_petugas	Int	5	*Foreign Key references* petugas (id_petugas)	*Not Null*
id_supp	Int	5	*Foreign Key references* supplier (id_ supp)	*Not Null*

9. Tabel detail_penerimaan

Tabel detail_penerimaan digunakan untuk menyimpan data detail penerimaan. Struktur dari tabel detail_penerimaan dijelaskan Tabel 3.86.

Tabel 3.86 Struktur Tabel detail_penerimaan

Nama Field	Tipe	Panjang	Kunci	Keterangan
id_detail_penerimaan	Int	10	*Primary Key*	*NotNull, autoincrement*
id_pemesanan	Int	10	*Foreign Key references* pemesanan (id_pemesanan)	*Null*
id_detail_psn	Int	10	*Foreign Key references* detail_pemesanan (id_detail_psn)	*Null*
id_penerimaan	Int	10	*Foreign Key references* penerimaan (id_penerimaan)	*Null*
id_barang	Int	10	Foreign Key references barang (id_barang)	*Null*
jumlah	Int	11		*Null*
keterangan	Varchar	30		*Null*

10. Tabel pengeluaran

Tabel pengeluaran digunakan untuk menyimpan data pengeluaran. Struktur dari tabel pengeluaran dijelaskan Tabel 3.87.

Tabel 3.87 Struktur Tabel pengeluaran

Nama Field	Tipe	Panjang	Kunci	Keterangan
id_pengeluaran	Int	10	*Primary Key*	*NotNull, Autoincrement*
no_faktur	Varchar	20		*Null*
no_permintaan	Varchar	20		*Null*
tgl_keluar	Date			*Null*
id_petugas	Int	5	*Foreign Key references* petugas (id_petugas)	*NotNull, Autoincrement*
ptgs_kacab	Varchar	25		*Null*
nm_kacab	Varchar	50		*NotNull*
no_telp	Varchar	15		*NotNull*

11. Tabel detail_pengeluaran

Tabel detail_pengeluaran digunakan untuk menyimpan data detail pengeluaran. Struktur dari tabel detail_pengeluaran dijelaskan Tabel 3.88.

Tabel 3.88 Struktur Tabel detail_pengeluaran

Nama Field	Tipe	Panjang	Kunci	Keterangan
id_detail_keluar	Int	10	*Primary Key*	*NotNull, Autoincrement*
id_pengeluaran	Int	10	*Foreign Key references* pengeluaran (id_pengeluaran)	*NotNull, Autoincrement*
id_barang	Int	10	*Foreign Key references* barang (id_barang)	*NotNull, Autoincrement*
jumlah	Int	11		*Not Null*

12. Tabel tabel_b

Tabel tabel_b digunakan untuk membantu dalam perhitungan rencana inventori. Struktur dari tabel_b dijelaskan Tabel 3.89.

Tabel 3.89 Struktur tabel_b

Nama Field	Tipe	Panjang	Kunci	Keterangan
id	Int	5	*Primary Key*	*NotNull, autoincrement*
z	Varchar	10		*Not Null*
alfa	Varchar	10		*Not Null*
ordinat	Varchar	10		*Not Null*
ekspektasi	Varchar	10		*Not Null*

13. Tabel pemesanan_v

Tabel pemesanan_v digunakan untuk membantu dalam pencarian data laporan pemesanan barang. Struktur dari tabel pemesanan_v dijelaskan Tabel 3.90.

Tabel 3.90 Struktur Tabel pemesanan_v

Nama Field	Tipe	Panjang	Kunci	Keterangan
id_pemesanan	Int	10		*Null*
id_detail_psn	Int	10		*Null*
id_supp	Int	5		*Null*
id_petugas	Int	5		*Null*
id_barang	Int	10		*Null*
no_pemesanan	Varchar	20		*Null*
kd_barang	Varchar	15		*Null*
nm_barang	Varchar	50		*Null*
nm_kategori	Varchar	50		*Null*
nm_satuan	Varchar	25		*Null*
nm_supplier	Varchar	50		*Null*
nm_petugas	Varchar	20		*Null*
tgl_psn	Date	-		*Null*
jumlah	Int	11		*Null*
sisa	Int	11		*Null*

14. Tabel penerimaan_v

Tabel penerimaan_v digunakan untuk membantu dalam pencarian data laporan penerimaan barang. Struktur dari tabel penerimaan_v dijelaskan Tabel 3.91.

Tabel 3.91 Struktur Tabel penerimaan_v

Nama Field	Tipe	Panjang	Kunci	Keterangan
id_pemesanan	Int	10		*Null*
id_detail_psn	Int	10		*Null*
id_penerimaan	Int	10		*Null*
id_detail_penerimaan	Int	10		*Null*
id_supp	Int	5		*Null*
id_petugas	Int	5		*Null*
id_barang	Int	10		*Null*
no_pemesanan	Varchar	20		*Null*
kd_barang	Varchar	15		*Null*
nm_barang	Varchar	50		*Null*
nm_kategori	Varchar	50		*Null*
no_penerimaan	Varchar	20		*Null*
nm_satuan	Varchar	25		*Null*
nm_supplier	Varchar	50		*Null*
nm_petugas	Varchar	20		*Null*
tgl_terima	Date	-		*Null*
jumlah	Int	11		*Null*

15. Tabel pengeluaran_v

Tabel pengeluaran_v digunakan untuk membantu dalam pencarian data laporan pengeluaran barang. Struktur dari tabel pengeluaran_v dijelaskan Tabel 3.92.

Tabel 3.92 Struktur Tabel pengeluaran_v

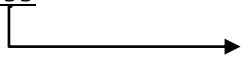
Nama Field	Tipe	Panjang	Kunci	Keterangan
id_pengeluaran	Int	10		*Null*
id_detail_pengeluaran	Int	10		*Null*
id_kacab	Int	5		*Null*
id_petugas	Int	5		*Null*
id_barang	Int	10		*Null*
no_faktur	Varchar	20		*Null*
kd_barang	Varchar	15		*Null*
nm_barang	Varchar	50		*Null*
nm_kategori	Varchar	50		*Null*
nm_satuan	Varchar	25		*Null*
nm_kacab	Varchar	50		*Null*
ptgs_kacab	Varchar	25		*Null*
nm_petugas	Varchar	20		*Null*
tgl_keluar	Date	-		*Null*
jumlah	Int	11		*Null*

3.2.2 Perancangan Kode

Kode yang akan digunakan dalam sistem yang akan dibangun adalah kode barang, kode kategori dan kode transaksi. Jenis kode transaksi yang akan digunakan adalah kode pemesanan, penerimaan, dan kode pengeluaran. Perancangan kode yang akan digunakan tersebut adalah sebagai berikut.

1. Kode Kategori

   *Form*at : 999

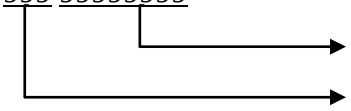
   

   3 digit nomor urut kategori

   Contoh : 001 sebagai nomor urut untuk nama kategori *transformator*.

2. Kode Barang

   Format : 999 99999999

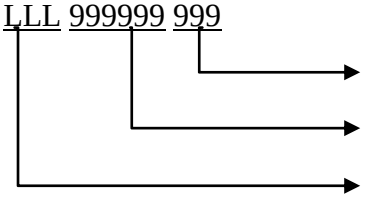
   8 digit nomor urut barang

   3 digit kode kategori

   Contoh : 00100000001 yang berarti tiga digit paling awal menandakan barang tersebut tergolong dalam kategori *transformator* dan sisanya adalah nomor urut barang tersebut.

3. Kode Pemesanan

   Format : LLL 999999 999

   3 digit nomor urut pemesanan

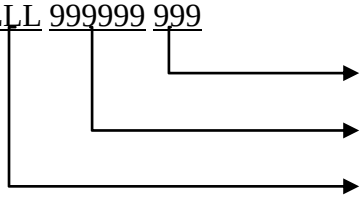
   6 digit menunjukan tahun dan bulan

   3 huruf label pemesanan

   Contoh kode yang digunakan adalah PSN110603001 yang menunjukan transaksi pemesanan yang pertama pada tanggal 03-06-2011.

4. Kode Pengeluaran

   Format : LLL 999999 999

   3 digit nomor urut pengeluaran

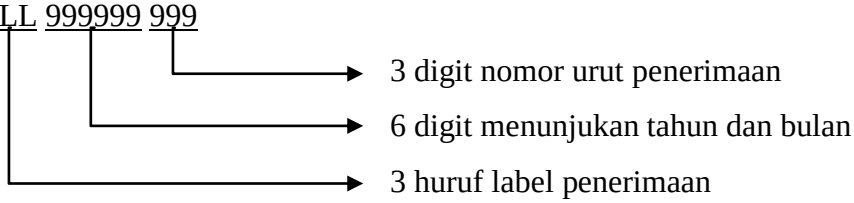
   6 digit menunjukan tahun dan bulan

   3 huruf label pengeluaran

   Contoh kode yang digunakan adalah KLR110301001 yang menunjukan transaksi pengeluaran dengan nomor urut satu pada bulan Maret 2011.

5. Kode Penerimaan

   Format : LLL 999999 999



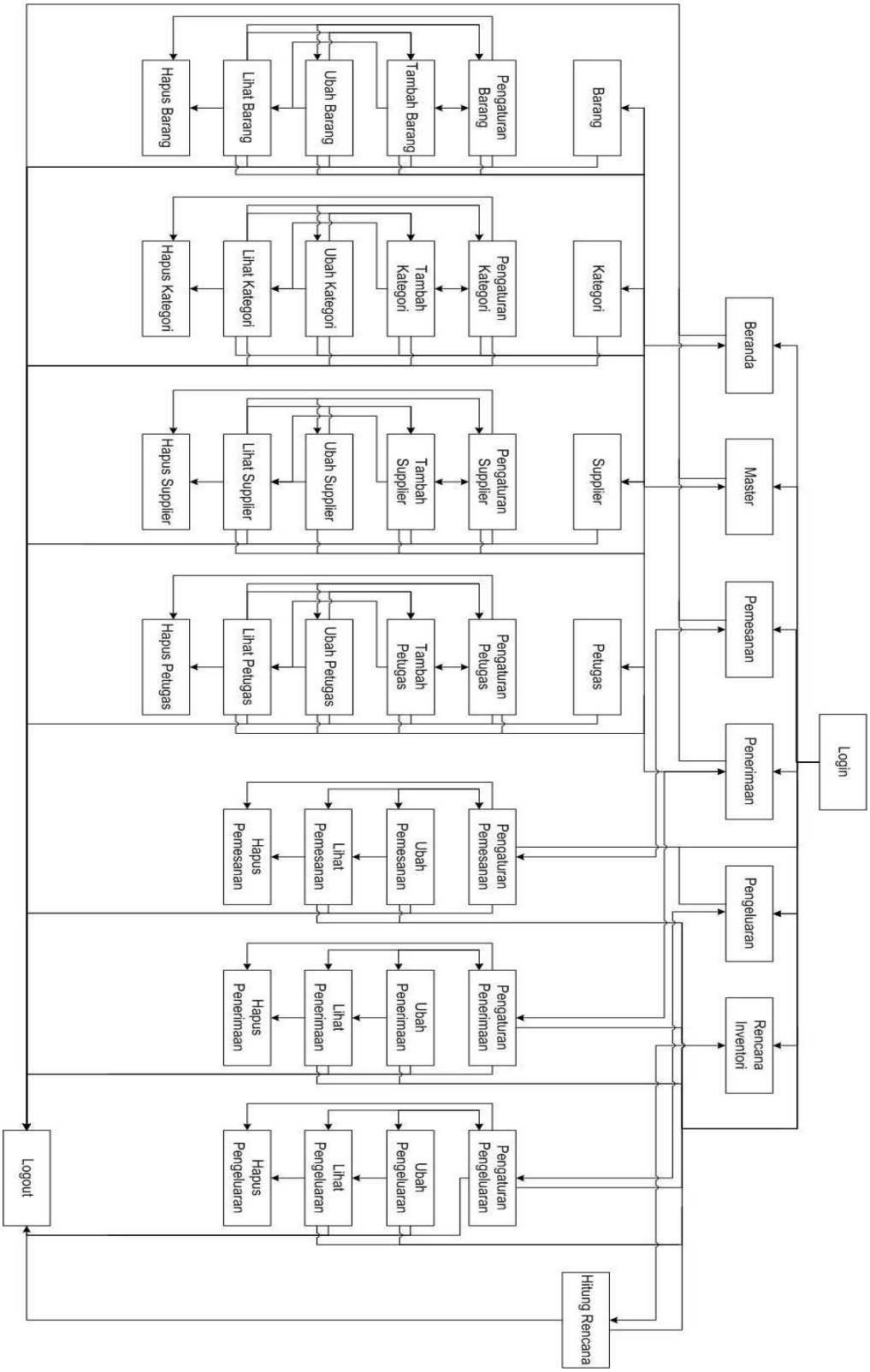
   Contoh kode yang digunakan adalah TRM110301001 yang menunjukan transaksi penerimaan barang dengan nomor urut satu pada bulan Maret 2011.

3.2.3 Struktur Menu

Struktur menu dari aplikasi yang akan dibangun terdiri dari struktur menu untuk hak akses admin dan petugas.

3.2.3.1 Struktur Menu Admin

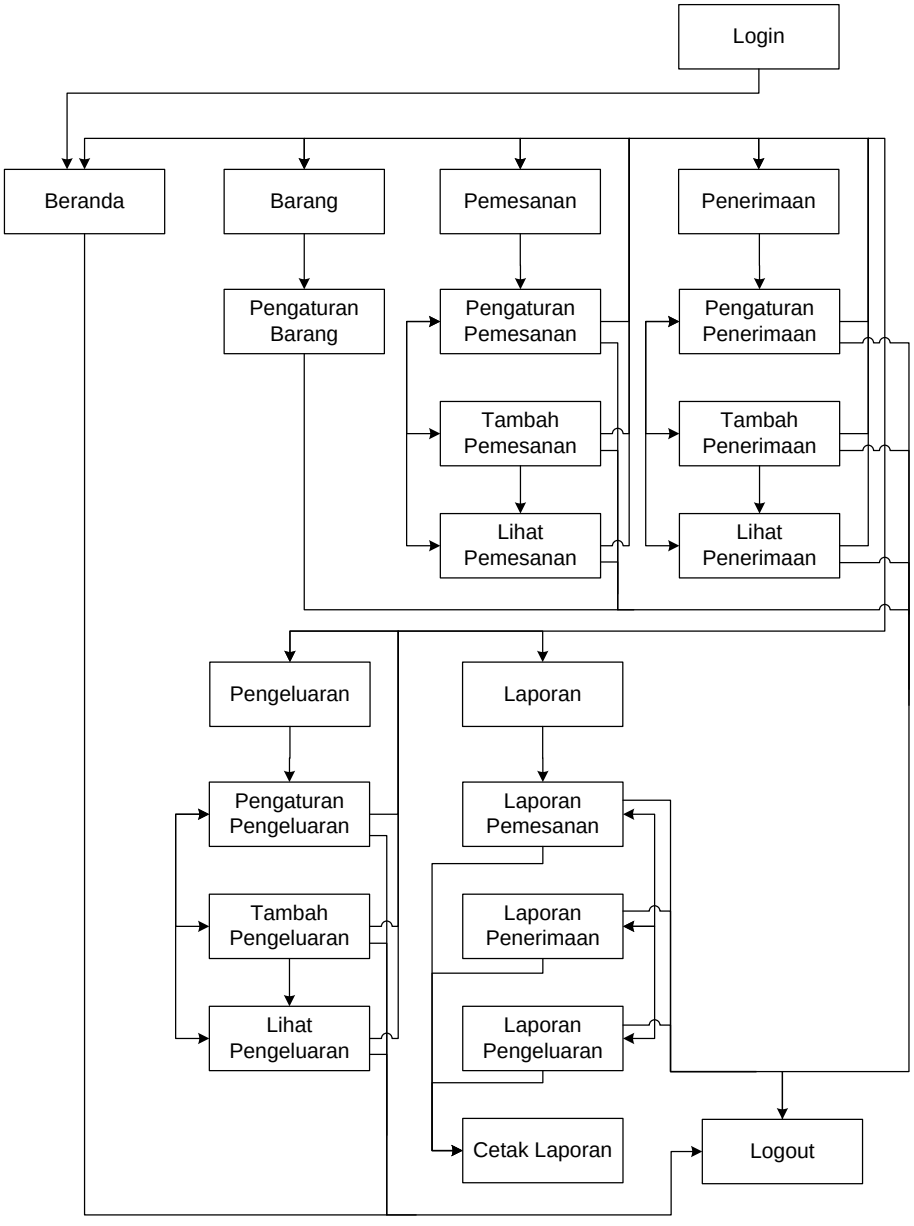
Struktur menu untuk admin menggambarkan proses pengolahan data yang dilakukan oleh admin sesuai yang tertera pada Gambar 3.75.



Gambar 3.75 Struktur Menu Admin

3.2.3.2 Struktur Menu Petugas

Struktur menu petugas menggambarkan proses berbagai pengolahan data yang dilakukan oleh petugas bagian gudang yaitu pengolahan pemesanan, penerimaan, pengeluaran, dan lain sebagainya seperti yang tertera pada Gambar 3.76.



Gambar 3.76 Struktur Menu Petugas

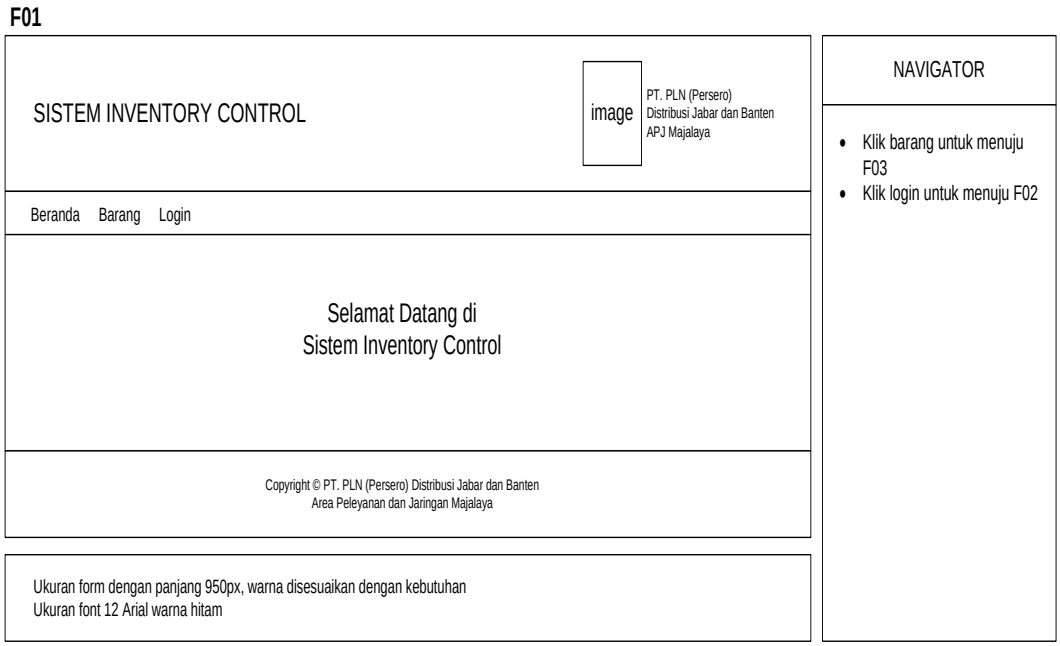
3.2.4 Perancangan Antarmuka

Perancangan antarmuka dibuat untuk memberi konsep pada implementasi pembuatan suatu antarmuka pada sistem. Berikut adalah gambaran dari perancangan antarmuka di sistem informasi *inventory control*.

3.2.4.1 Perancangan *Form*

Perancangan *form* dibuat untuk memberi gambaran mengenai halaman pada aplikasi yang dibangun. Gambaran dari masing dari masing-masing form adalah sebagai berikut:

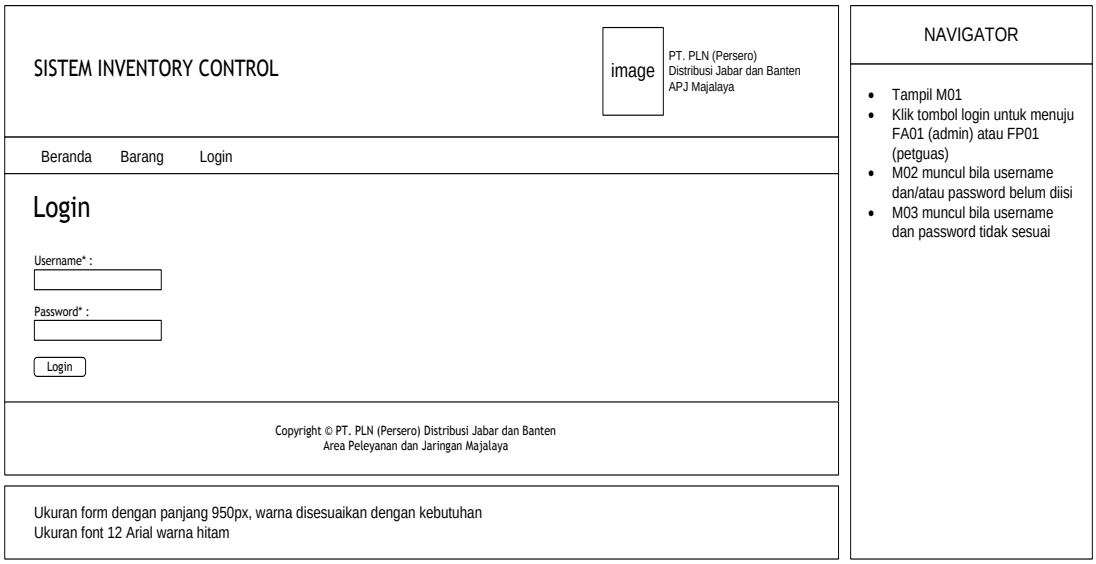
1. *Form* Beranda



Gambar 3.77 *Form* Beranda

2. *Form* Login

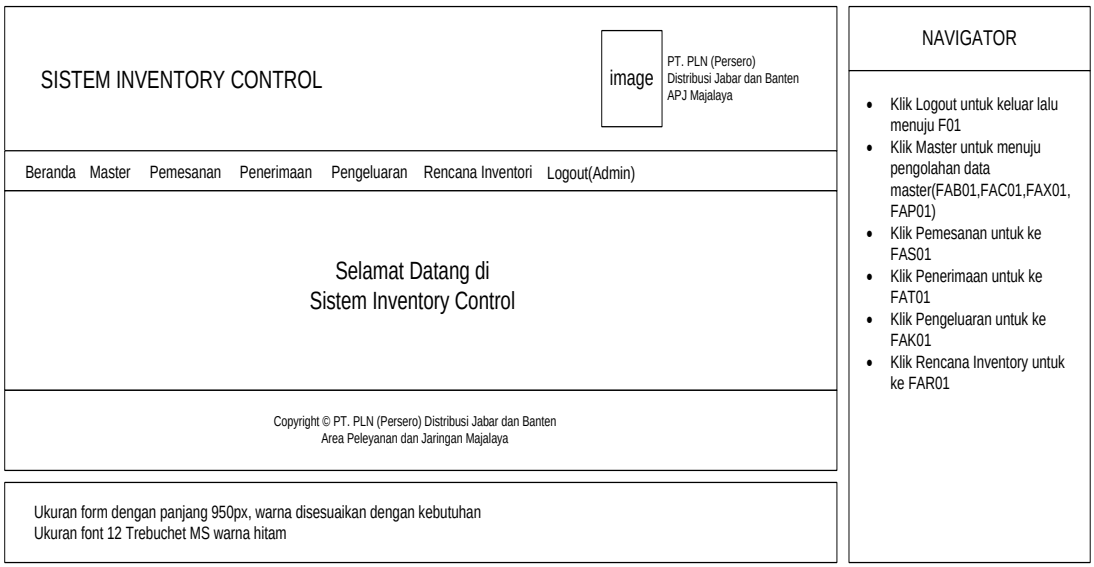
F02



Gambar 3.78 *Form* Login

3. Halaman Utama Admin

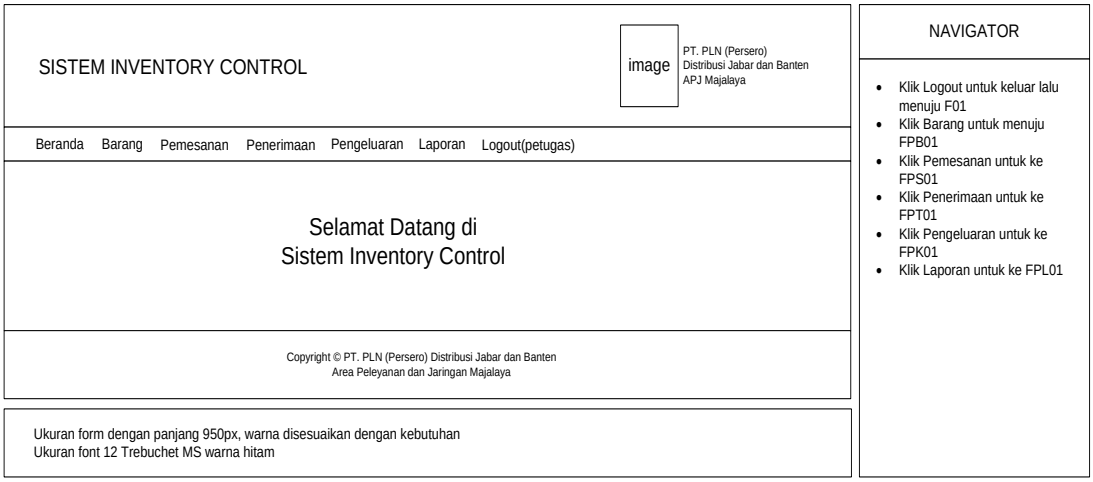
FA01



Gambar 3.79 *Form* Halaman Utama Admin

4. Halaman Utama Petugas

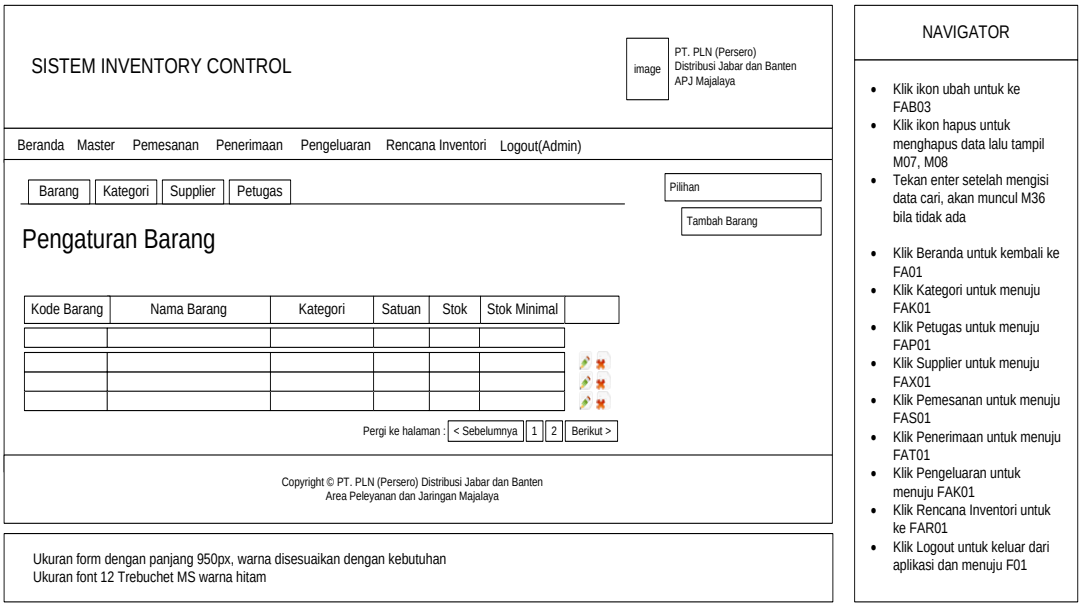
FP01



Gambar 3.80 Halaman Utama Petugas

5. *Form* Pengaturan Barang

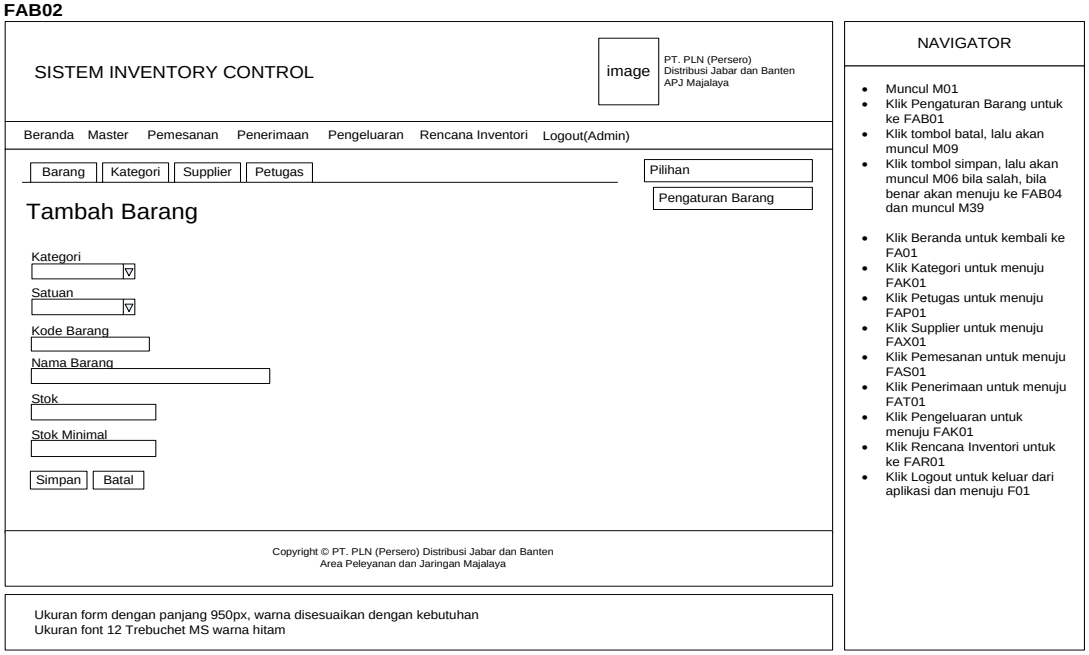
FAB01



Gambar 3.81 *Form* Pengaturan Barang

6. *Form* Tambah Barang

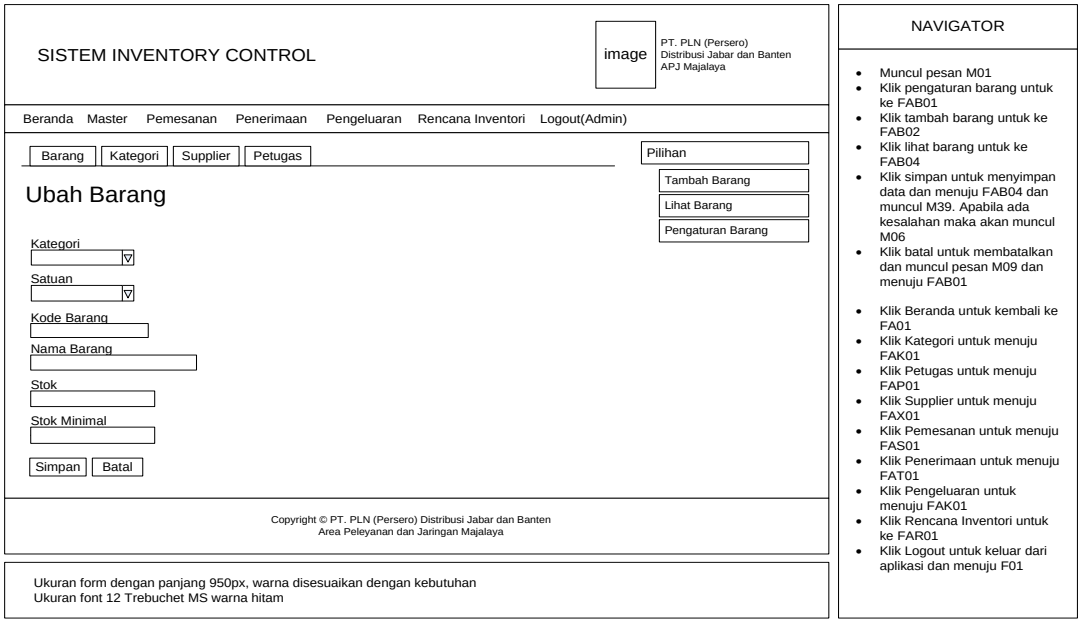
FAB02



Gambar 3.82 *Form* Tambah Barang

7. *Form* Ubah Barang

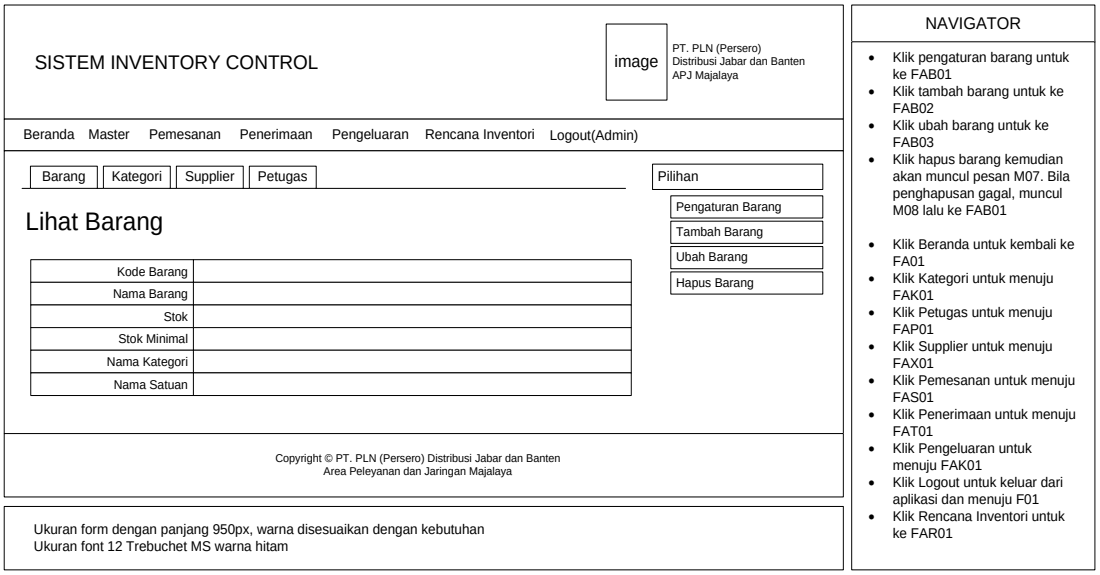
FAB03



Gambar 3.83 *Form* Ubah Barang

8. *Form* Lihat Barang

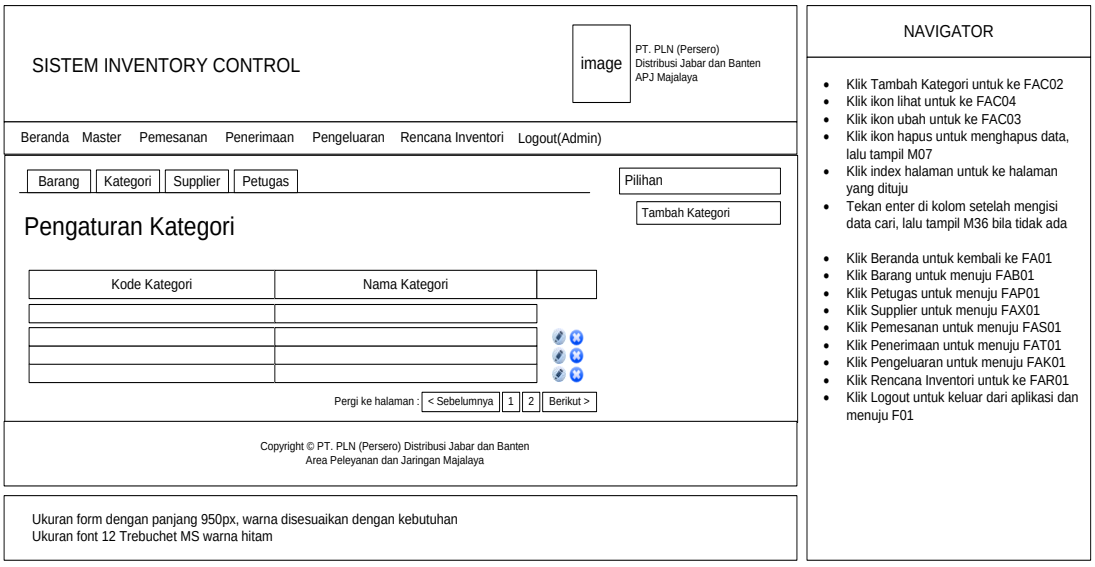
FAB04



Gambar 3.84 *Form* Lihat Barang

9. *Form* Pengaturan Kategori

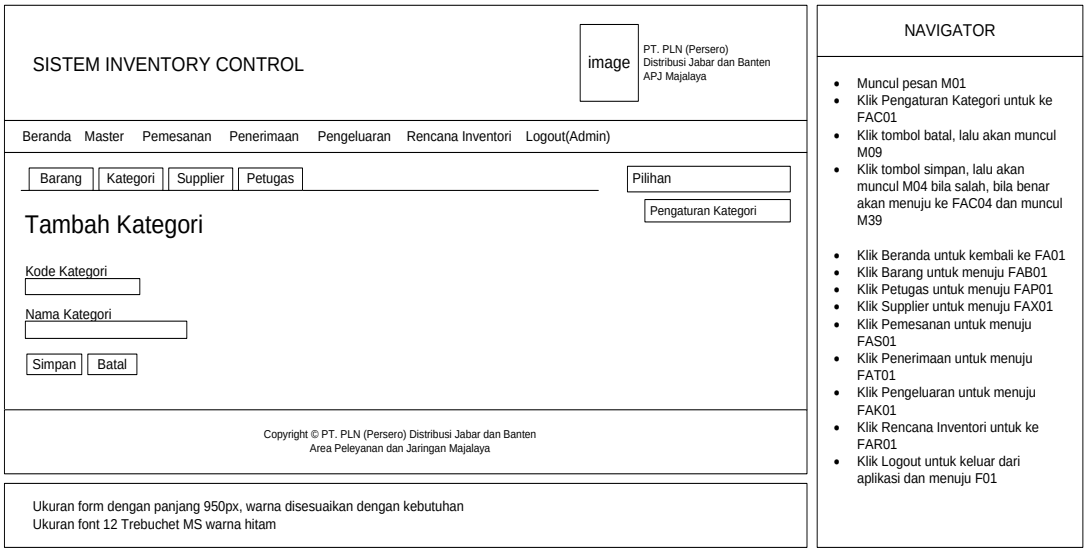
FAC01



Gambar 3.85 *Form* Pengturan Kategori

Form Tambah Kategori

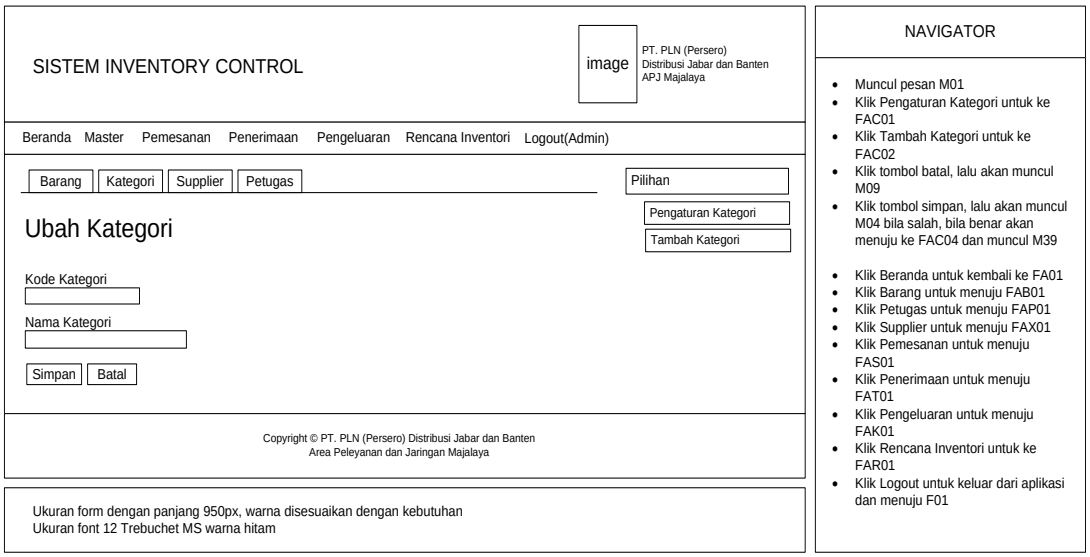
FAC02



Gambar 3.86 *Form* Tambah Kategori

10. *Form* Ubah Kategori

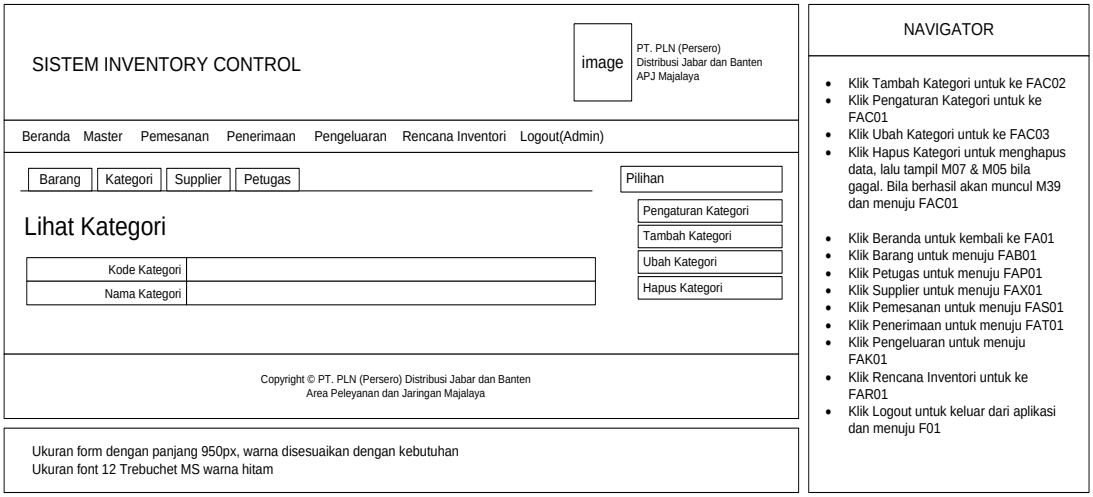
FAC03



Gambar 3.87 *Form* Ubah Kategori

11. *Form* Lihat Kategori

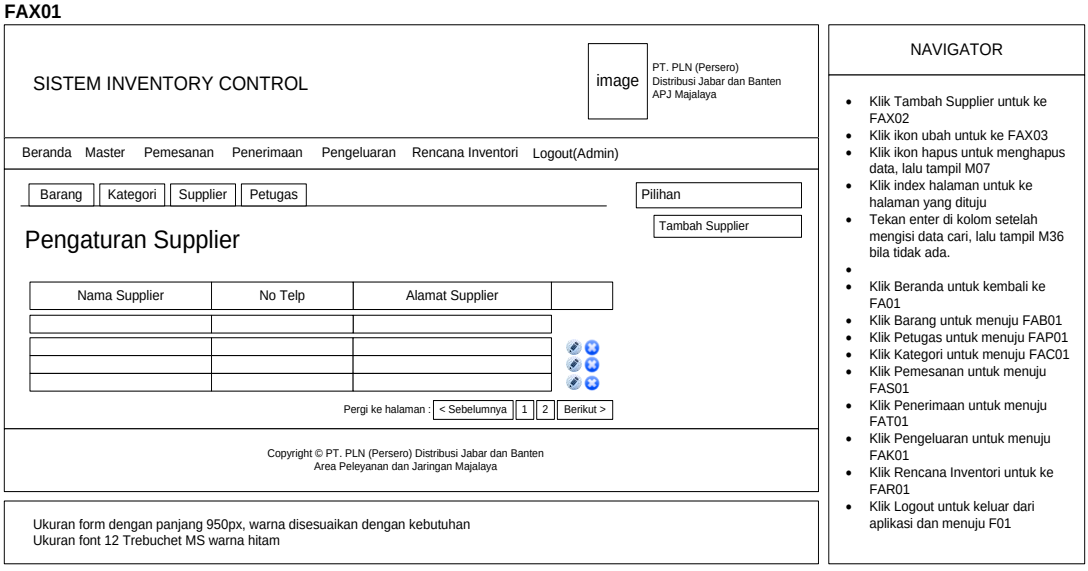
FAC04



Gambar 3.88 *Form* Lihat Kategori

12. *Form* Pengaturan Supplier

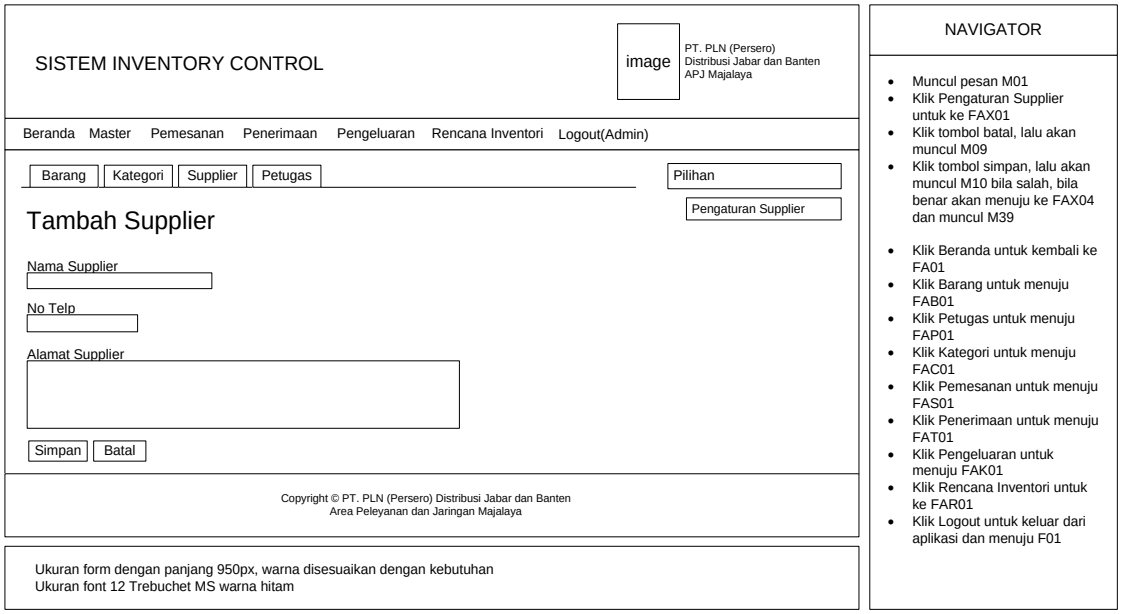
FAX01



Gambar 3.89 *Form* Pengaturan Supplier

Form Tambah Supplier

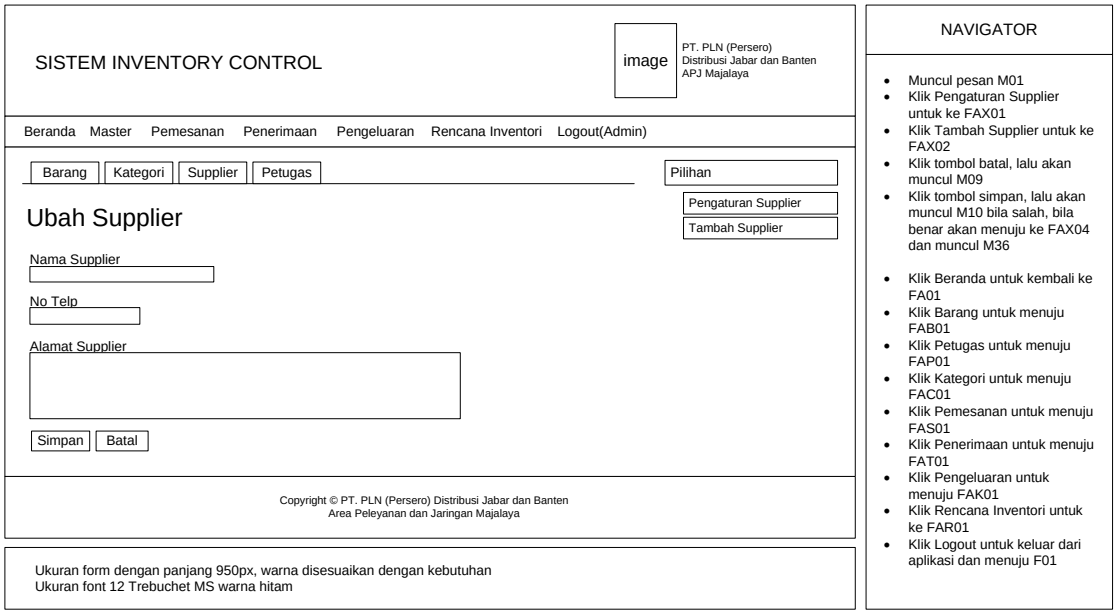
FAX02



Gambar 3.90 *Form* Tambah Supplier

13. *Form* Ubah Supplier

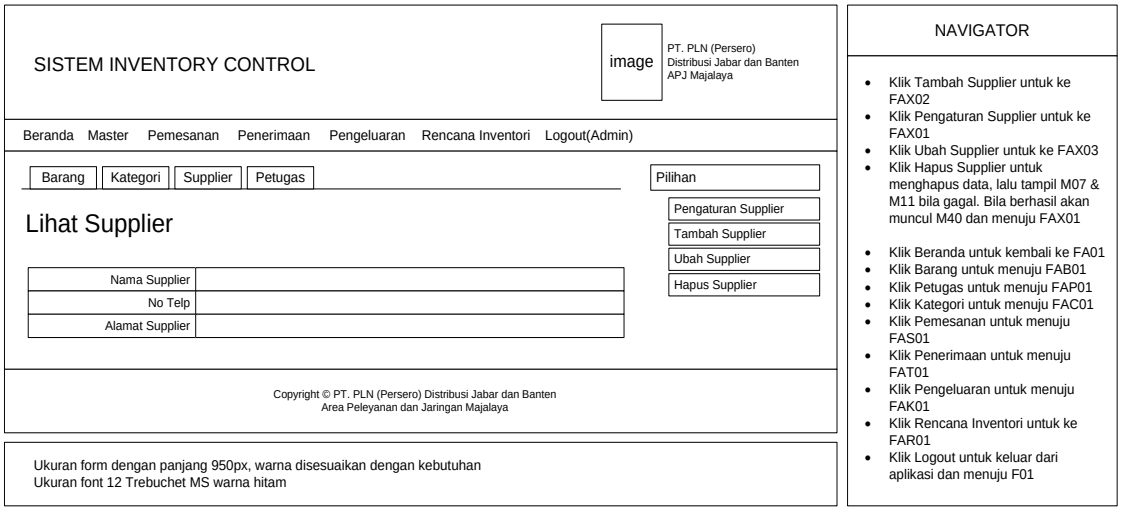
FAX03



Gambar 3.91 *Form* Ubah Supplier

14. *Form* Lihat Supplier

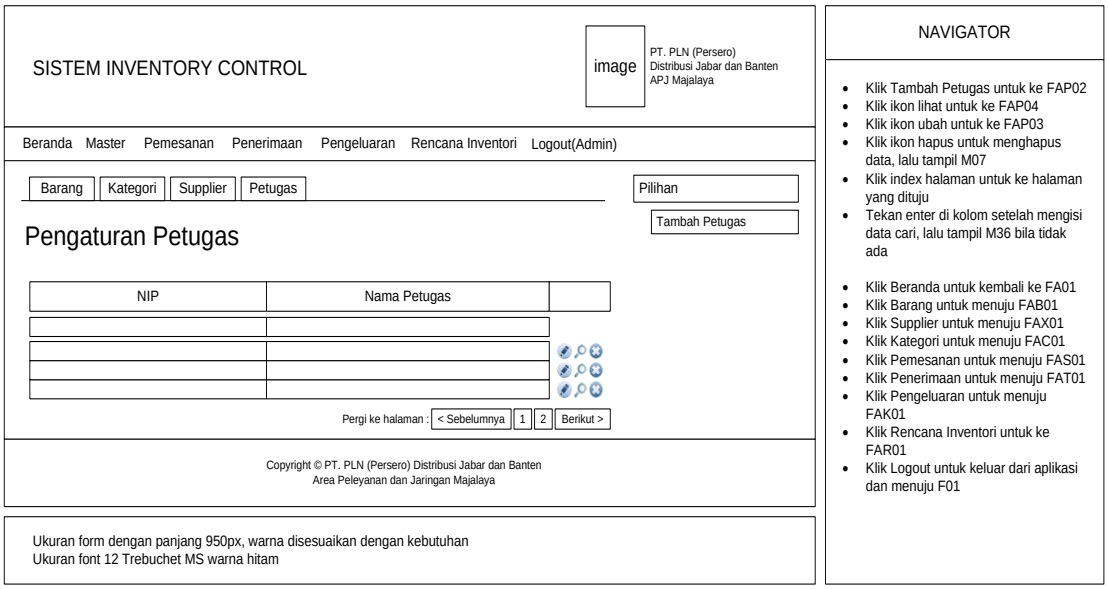
FAX04



Gambar 3.92 *Form* Lihat *Supplier*

15. *Form* Pengturan Petugas

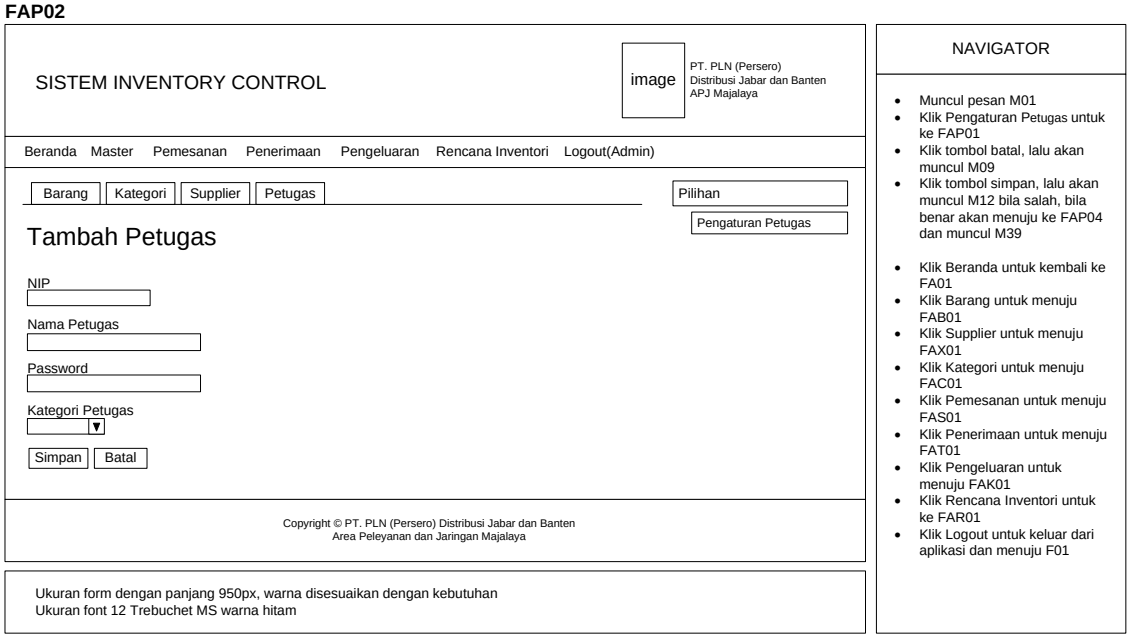
FAP01



Gambar 3.93 *Form* Pengaturan Petugas

16. *Form* Tambah Petugas

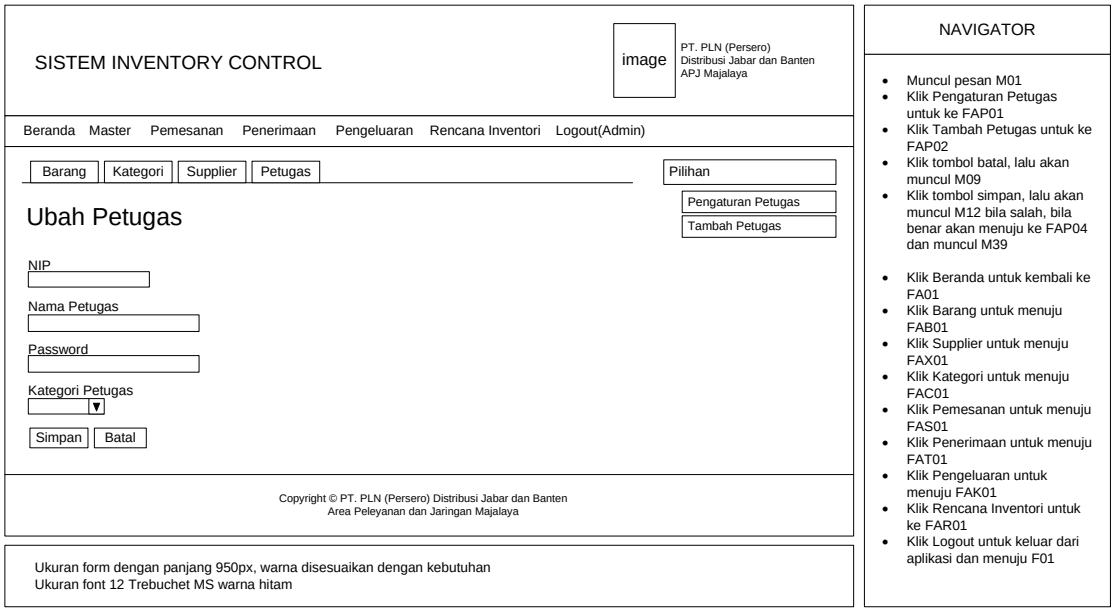
FAP02



Gambar 3.94 *Form* Tambah Petugas

17. *Form* Ubah Petugas

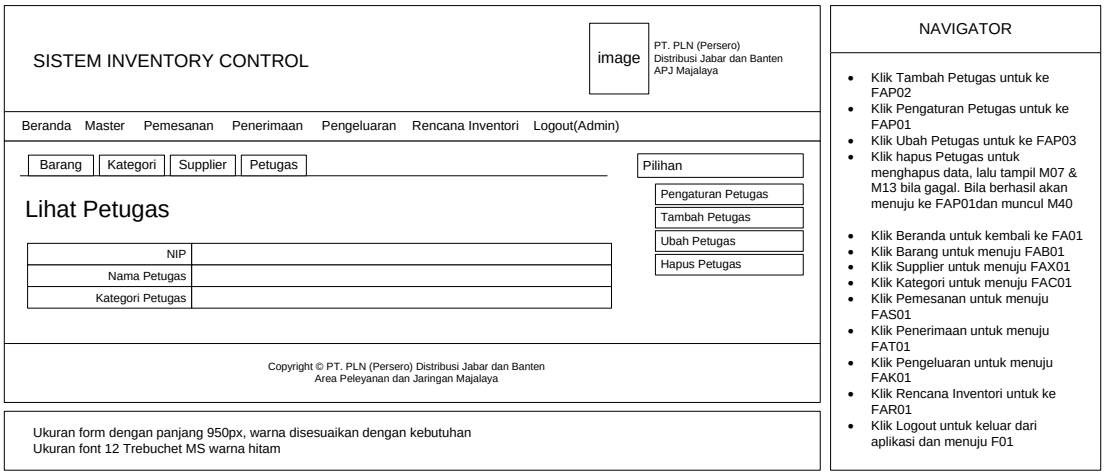
FAP03



Gambar 3.95 *Form* Ubah Petugas

18. *Form* Lihat Petugas

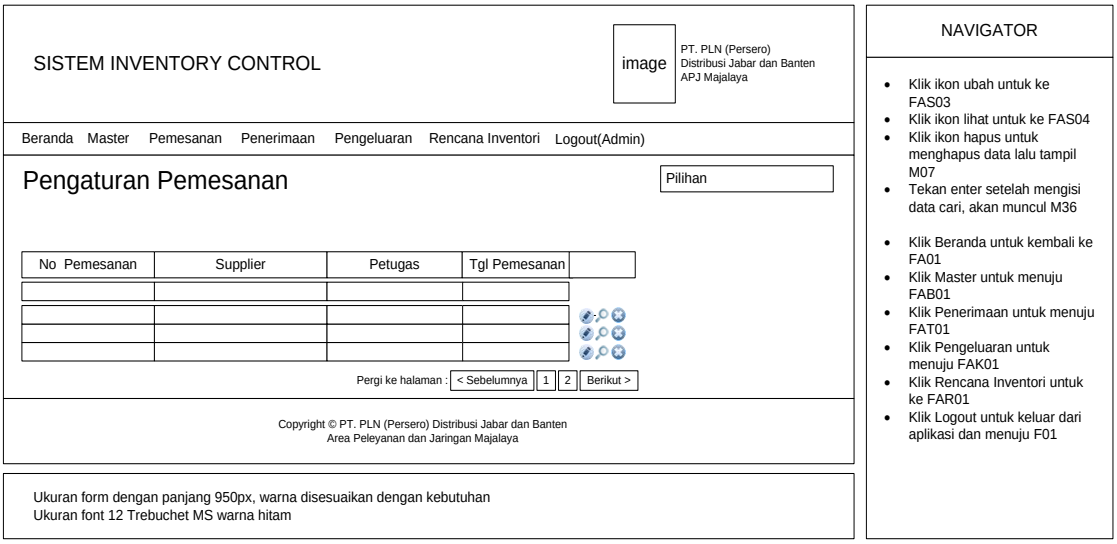
FAP04



Gambar 3.96 *Form* Lihat Petugas

19. *Form* Pengaturan Pemesanan

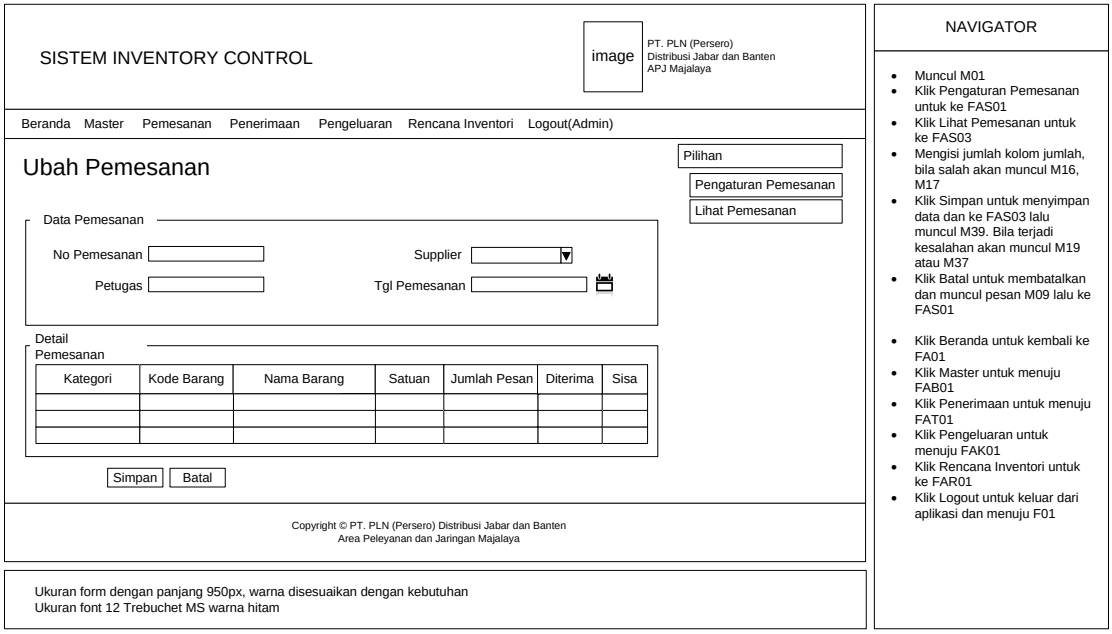
FAS01



Gambar 3.97 *Form* Pengaturan Pemesanan

20. *Form* Ubah Pemesanan

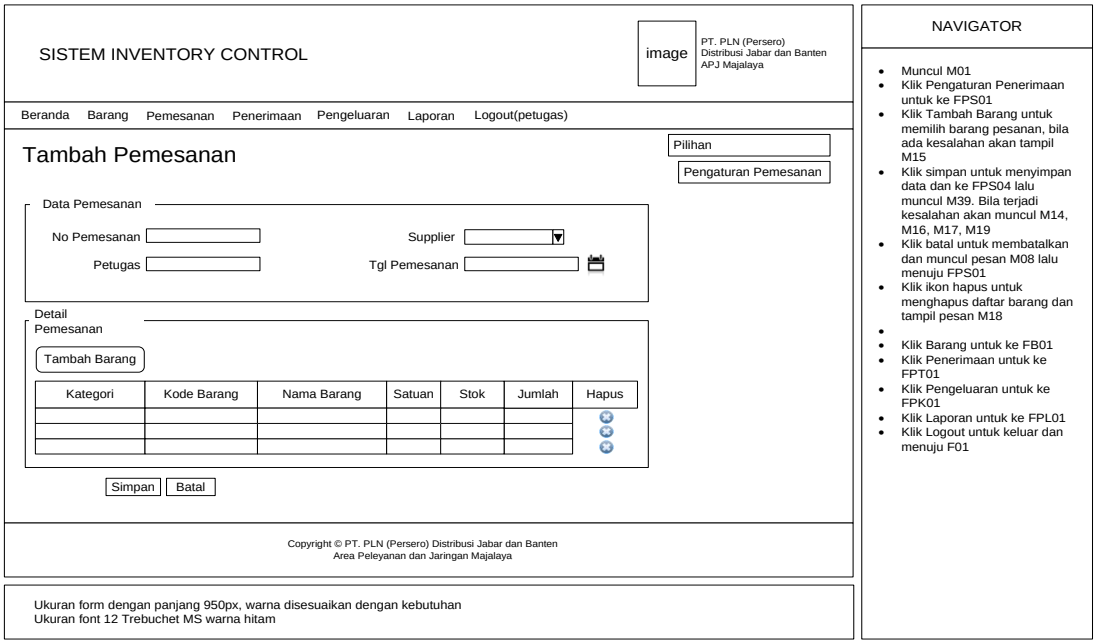
FAS02



Gambar 3.98 *Form* Ubah Pemesanan

21. *Form* Tambah Pemesanan

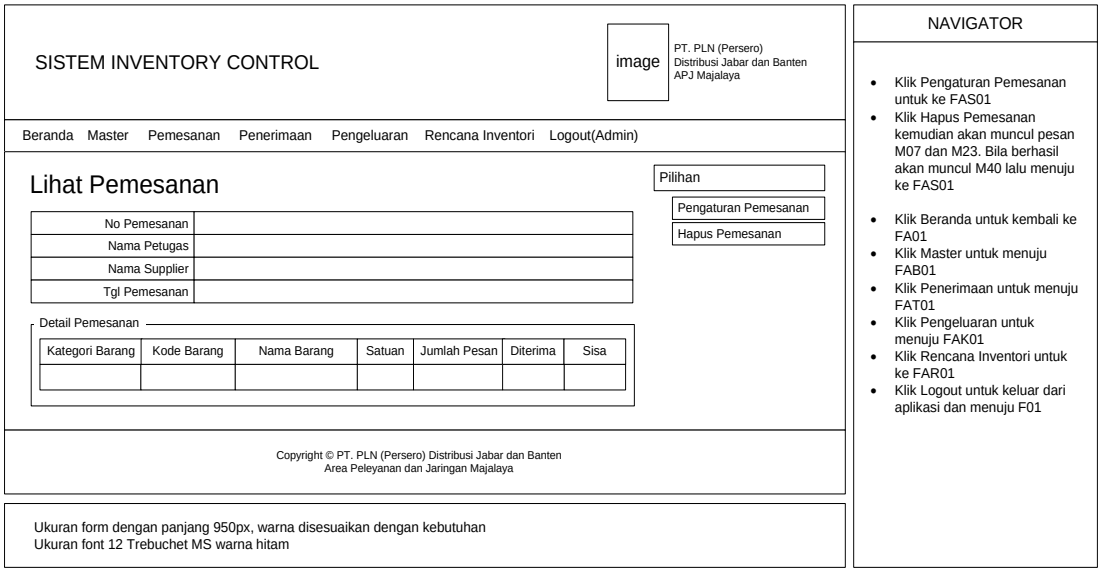
FPS02



Gambar 3.99 *Form* Tambah Pemesanan

22. *Form* Lihat Pemesanan

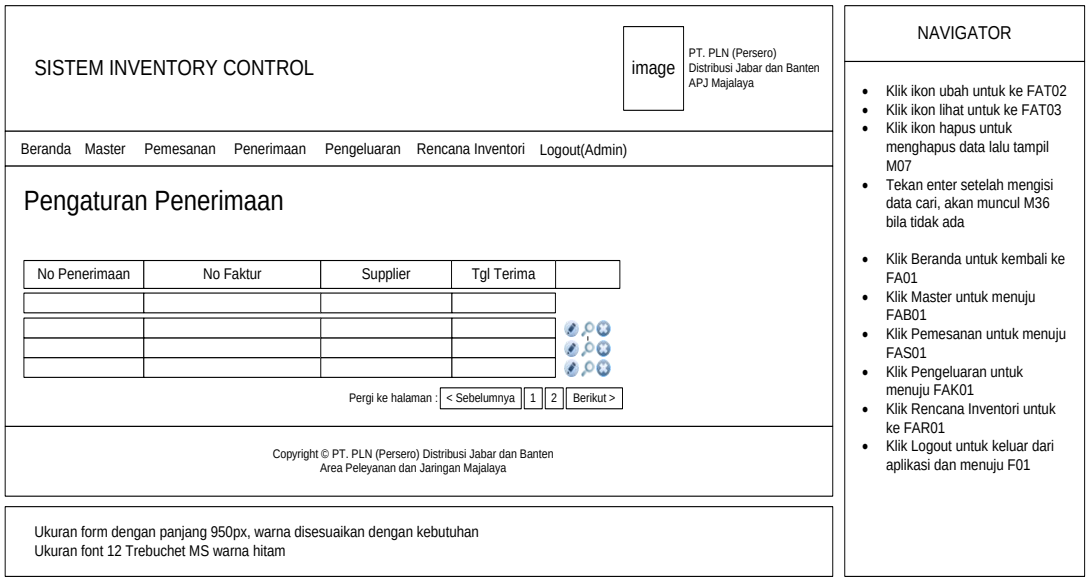
FAS03



Gambar 3.100 *Form* Lihat Pemesanan

23. *Form* Pengaturan Penerimaan

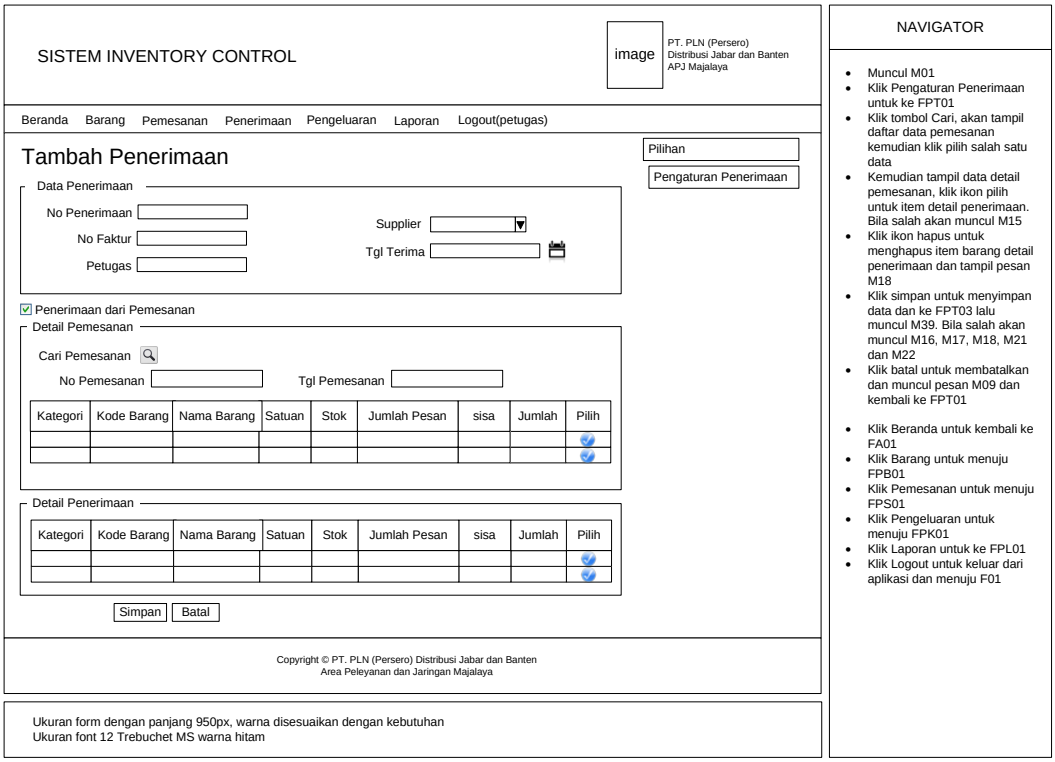
FAT01



Gambar 3.101 *Form* Pengaturan Penerimaan

24. *Form* Tambah Penerimaan

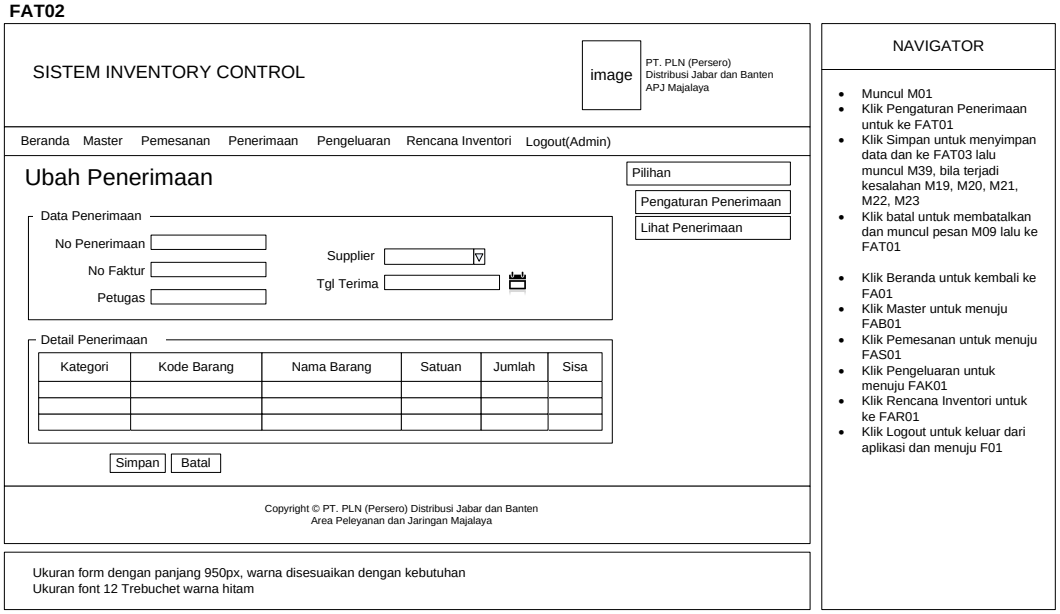
FPT02



Gambar 3.102 *Form* Tambah Penerimaan

25. *Form* Ubah Penerimaan

FAT02



Gambar 3.103 *Form* Ubah Penerimaan

26. *Form* Lihat Penerimaan

FAT03

SISTEM INVENTORY CONTROL

image

PT. PLN (Persero)
Distribusi Jabar dan Banten
APJ Majalaya

Beranda Master Pemesanan Penerimaan Pengeluaran Rencana Inventori Logout(Admin)

Lihat Penerimaan

Pilihan
Ubah Penerimaan
Hapus Penerimaan
Pengaturan Penerimaan

ID	
No Faktur	
No Pemesanan	
Nama Supplier	
Nama Petugas	
Tgl Terima	

Detail Peneriman

Kategori	Kode Barang	Nama Barang	Satuan	Jumlah

Copyright © PT. PLN (Persero) Distribusi Jabar dan Banten
Area Peleyanan dan Jaringan Majalaya

Ukuran form dengan panjang 950px, warna disesuaikan dengan kebutuhan
Ukuran font 12 Trebuchet warna hitam

NAVIGATOR

- Klik Ubah Penerimaan untuk ke FAT02
- Klik Pengaturan Penerimaan untuk ke FAT01
- Klik Hapus Penerimaan kemudian akan muncul pesan M07. Bila berhasil akan muncul M40 lalu menuju FAT01

Gambar 3.104 *Form* Lihat Penerimaan

27. *Form* Pengaturan Pengeluaran

FAK01

SISTEM INVENTORY CONTROL

image

PT. PLN (Persero)
Distribusi Jabar dan Banten
APJ Majalaya

Beranda Master Pemesanan Penerimaan Pengeluaran Rencana Inventori Logout(Admin)

Pengaturan Pengeluaran

Petugas	No Faktur	No Penerimaan	Kantor Cabang	Petugas Kantor Cabang	Tgl Pengeluaran	

Pergi ke halaman : < Sebelumnya 1 2 Berikut >

Copyright © PT. PLN (Persero) Distribusi Jabar dan Banten
Area Peleyanan dan Jaringan Majalaya

Ukuran form dengan panjang 950px, warna disesuaikan dengan kebutuhan
Ukuran font 12 Trebuchet MS warna hitam

NAVIGATOR

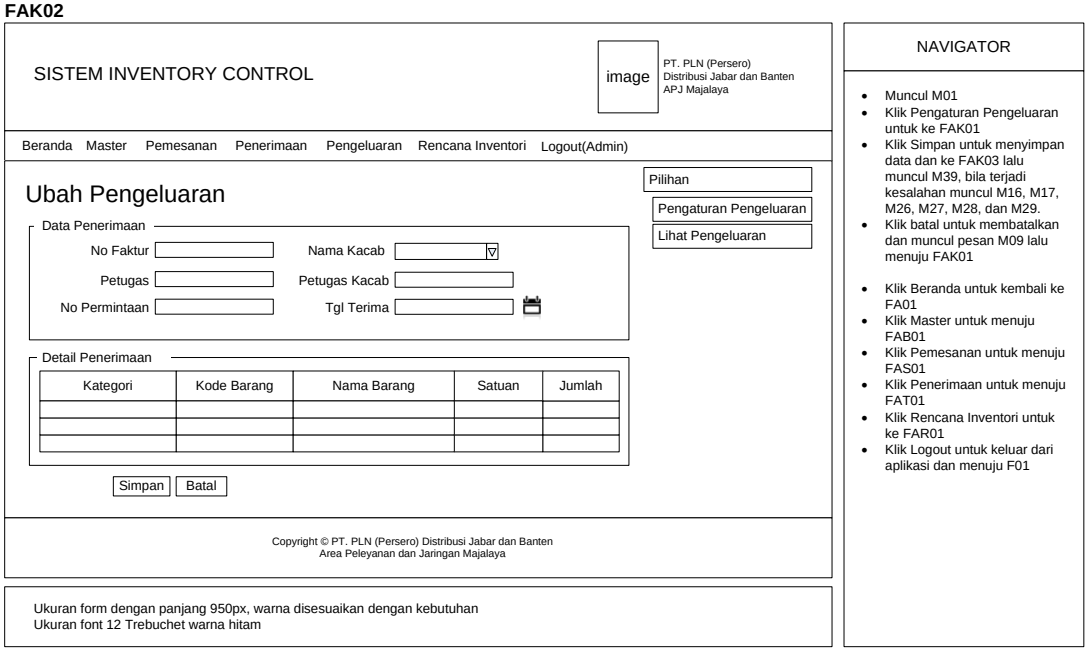
- Klik ikon ubah untuk ke FAT02
- Klik ikon lihat untuk ke FAT03
- Klik ikon hapus untuk menghapus data lalu tampil M07
- Tekan Enter setelah mengisi data cari, akan muncul M36 bila tidak ada

- Klik Beranda untuk kembali ke FA01
- Klik Master untuk menuju FAB01
- Klik Pemesanan untuk menuju FAS01
- Klik Penerimaan untuk menuju FAT01
- Klik Rencana Inventori untuk ke FAR01
- Klik Logout untuk keluar dari aplikasi dan menuju F01

Gambar 3.105 *Form* Pengaturan Pengeluaran

28. *Form* Ubah Pengeluaran

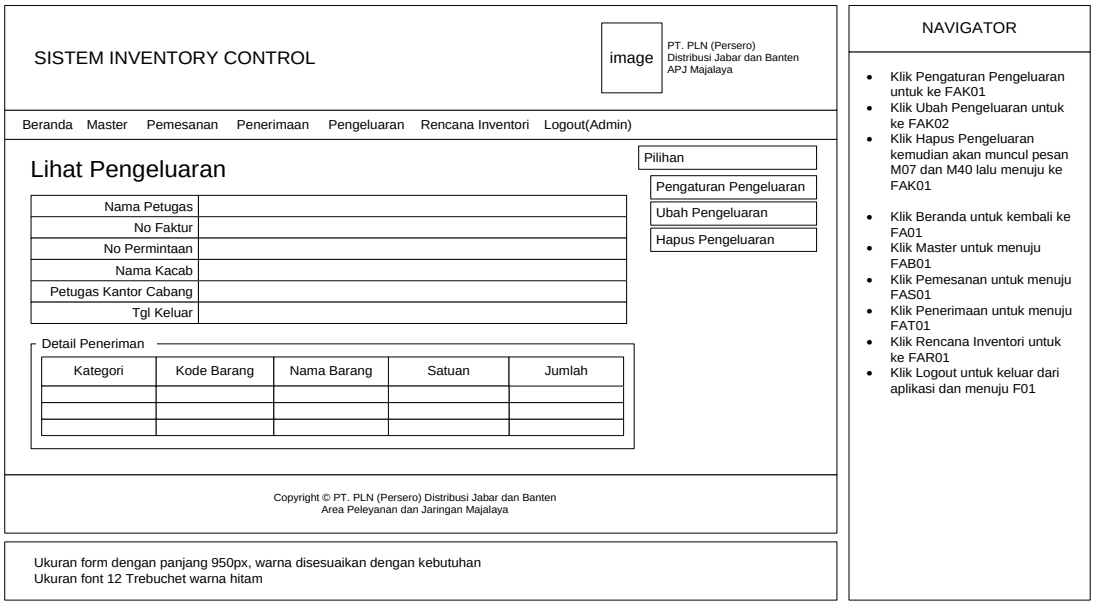
FAK02



Gambar 3.106 *Form* Ubah Pengeluaran

29. *Form* Lihat Pengeluaran

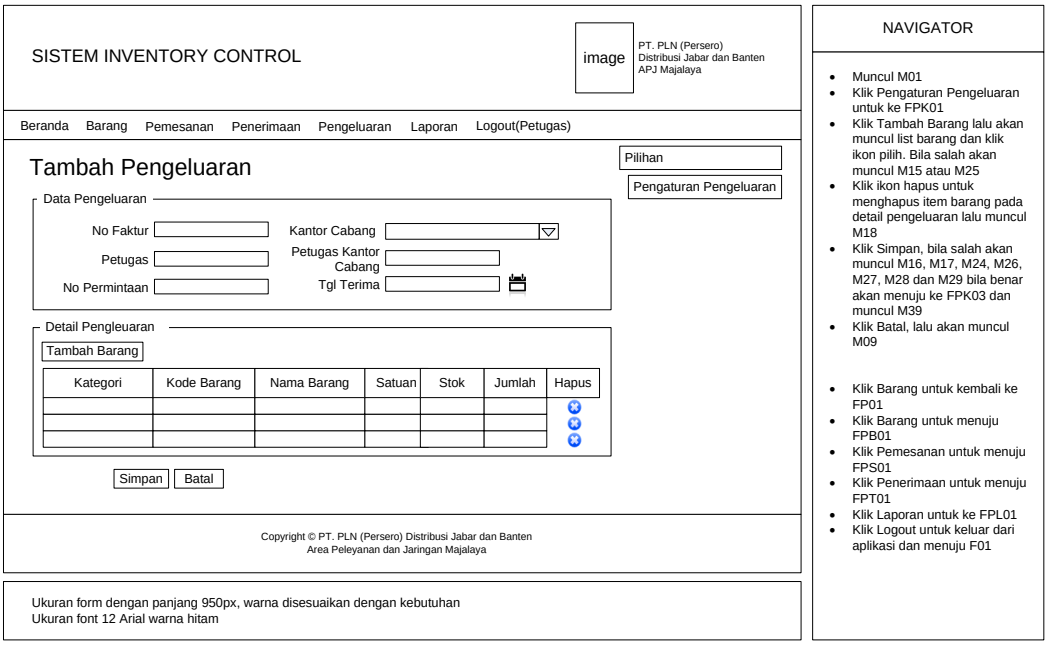
FAK03



Gambar 3.107 *Form* Lihat Pengeluaran

30. *Form* Tambah Pengeluaran

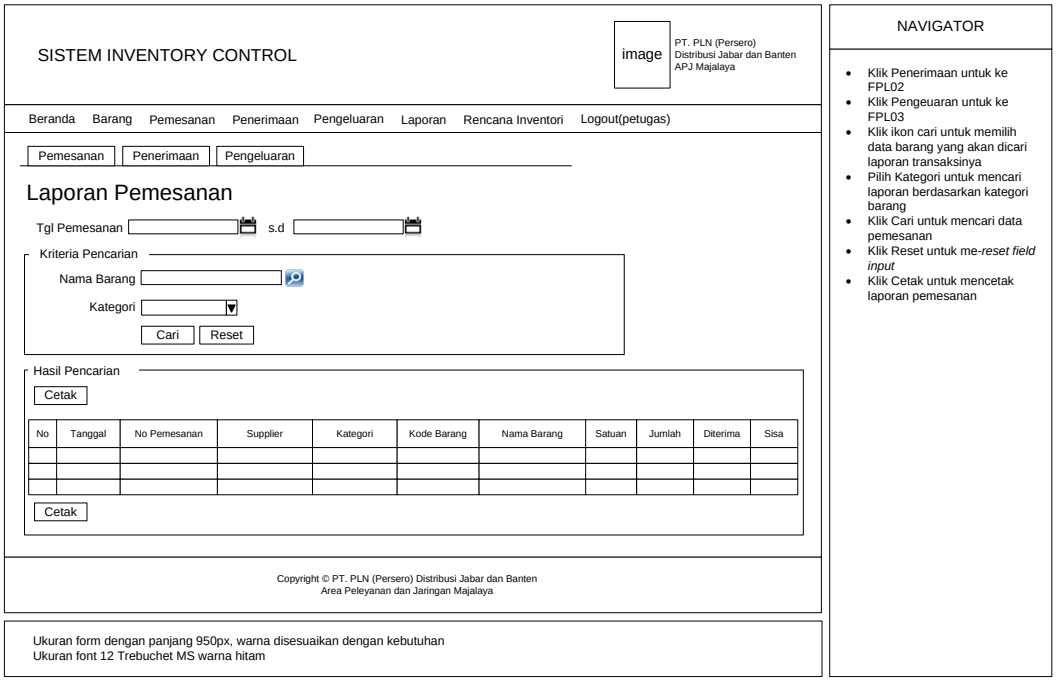
FPK02



Gambar 3.108 *Form* Tambah Pengeluaran

31. *Form* Laporan Pemesanan

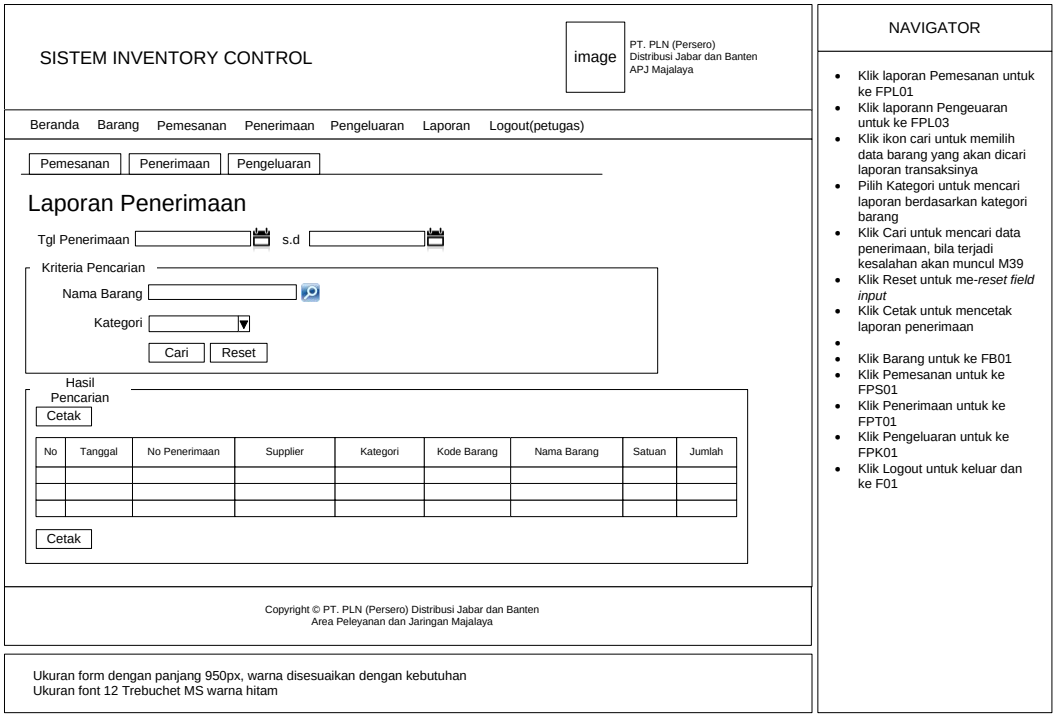
FPL01



Gambar 3.109 *Form* Laporan Pemesanan

32. *Form* Laporan Penerimaan

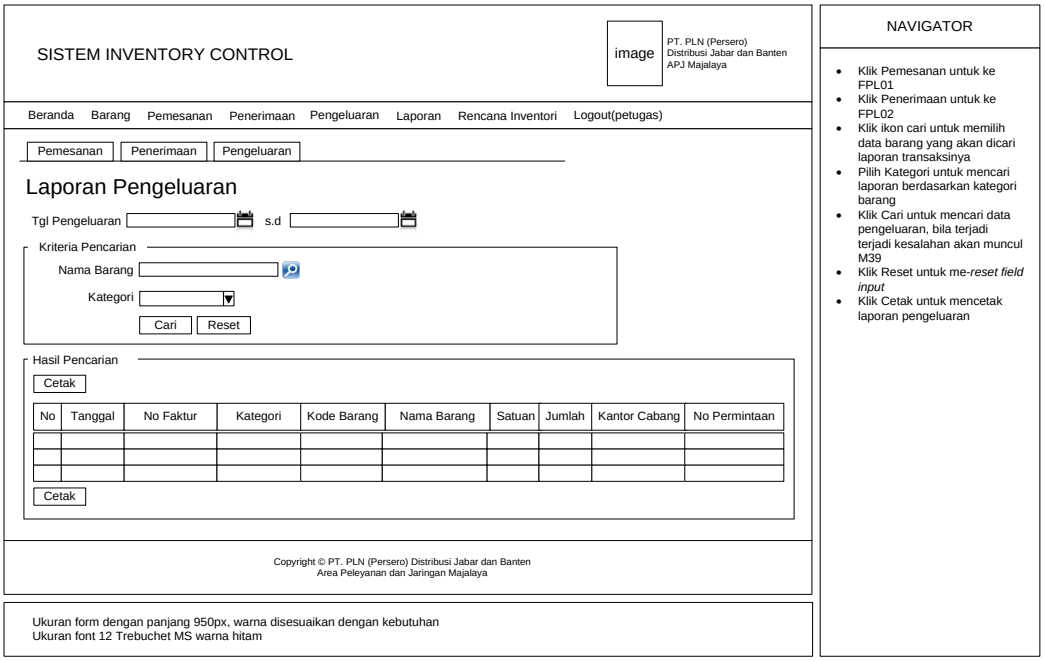
FPL02



Gambar 3.110 *Form* Laporan Penerimaan

33. *Form* Laporan Pengeluaran

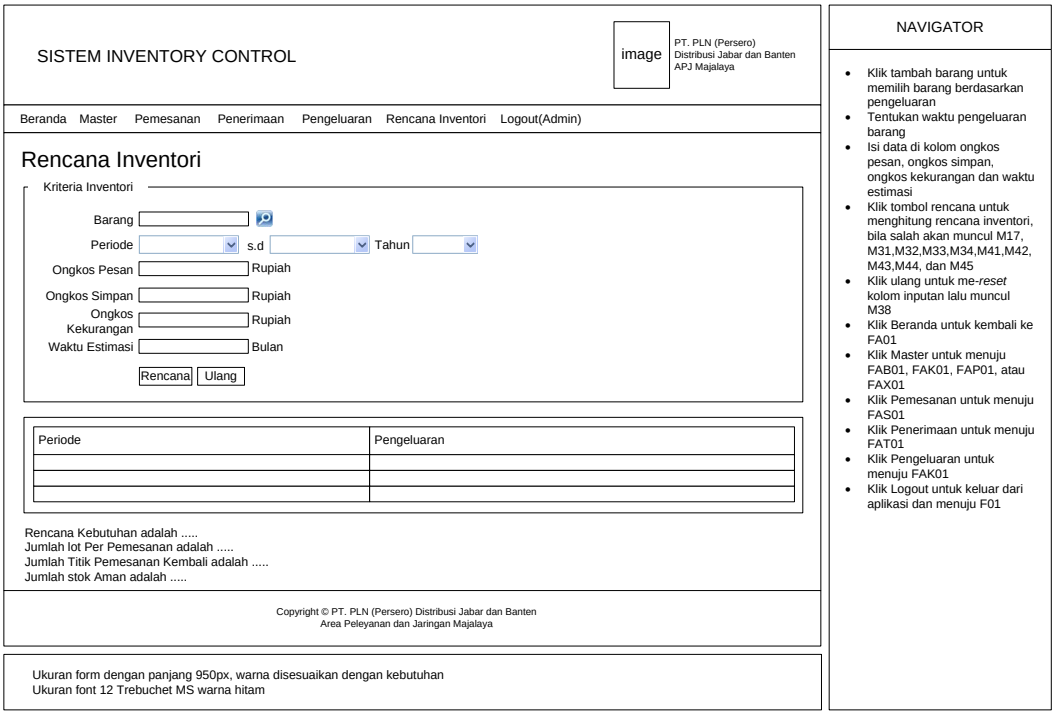
FPL03



Gambar 3.111 *Form* Laporan Pengeluaran

34. *Form* Rencana Inventori

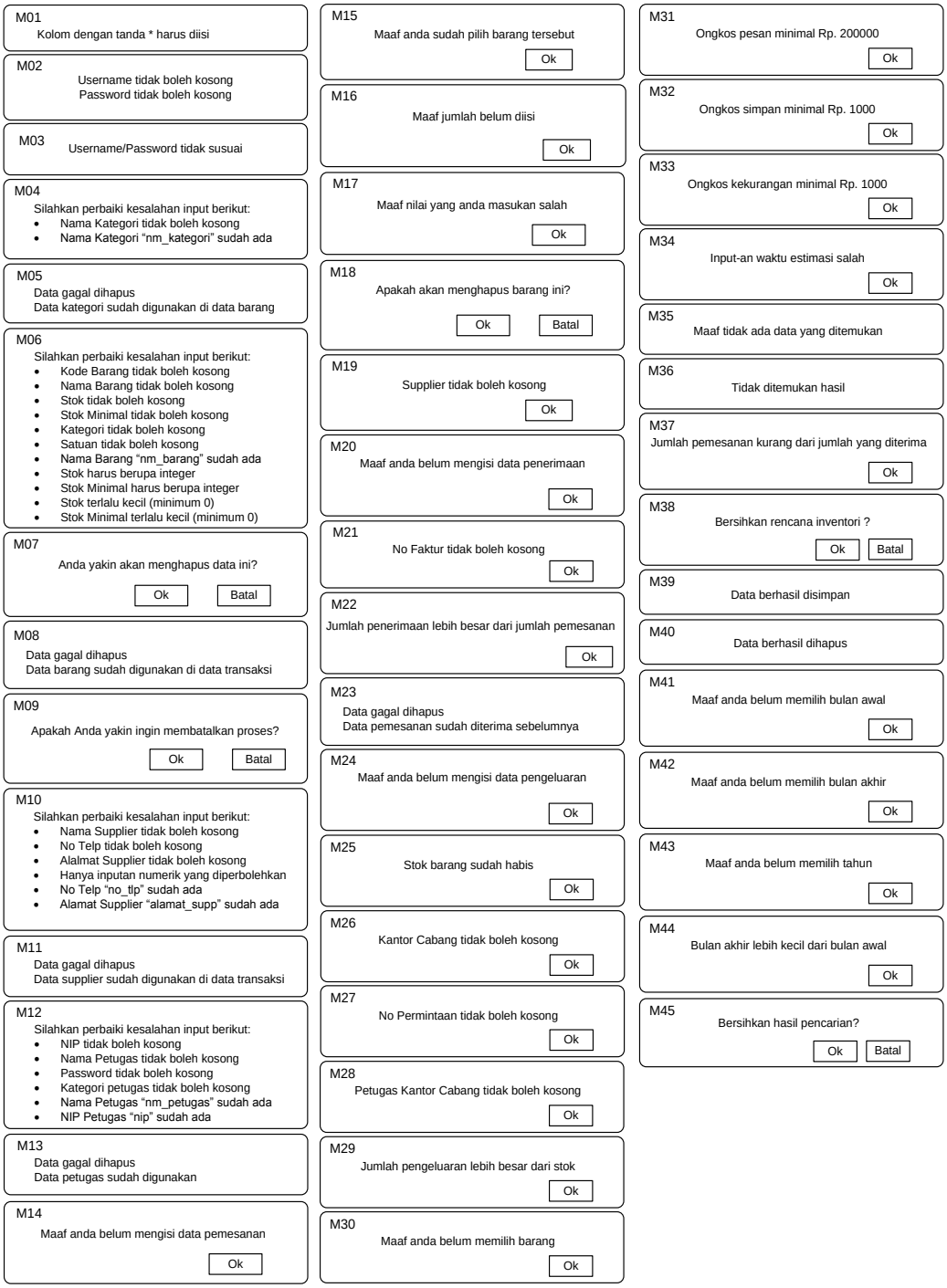
FAR01



Gambar 3.112 *Form* Rencana Inventori

3.2.4.2 Perancangan Pesan

Perancangan pesan yang dibuat dalam aplikasi sistem *inventory control* tertera pada Gambar 3.113.



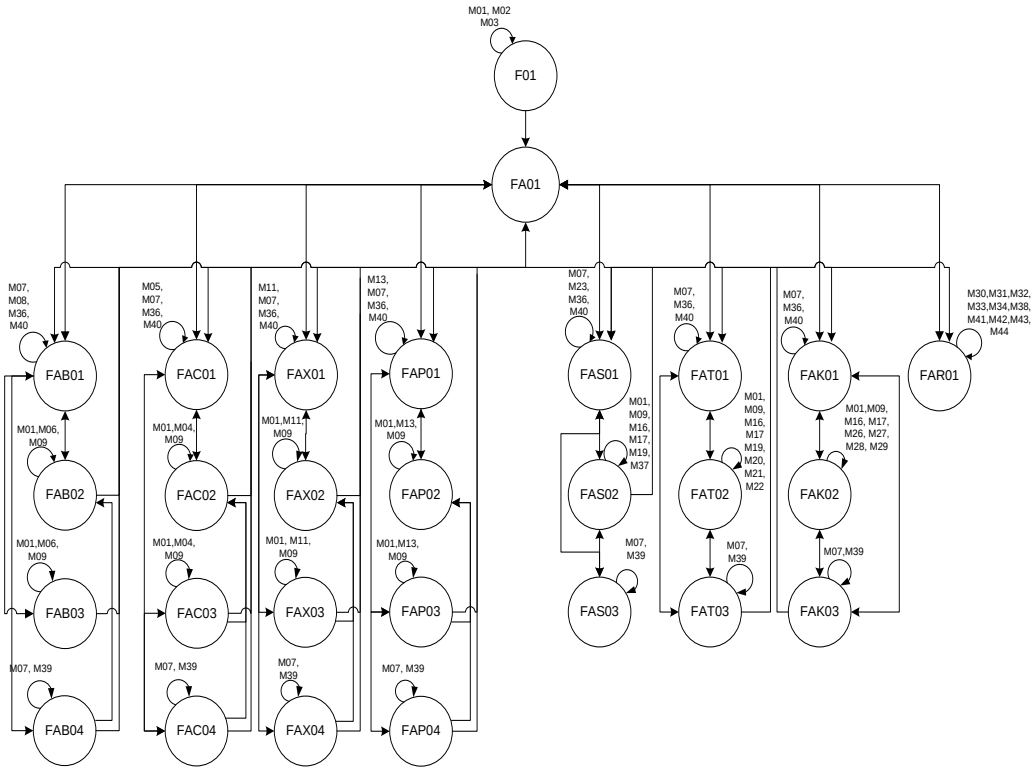
Gambar 3.113 Perancangan Pesan

3.2.4.3 Jaringan Semantik

Jaringan semantik adalah gambaran pengetahuan grafis yang menunjukkan hubungan antar berbagai objek, terdiri dari lingkaran-lingkaran yang dihubungkan dengan anak panah yang menunjukkan objek dan informasi tentang objek-objek tersebut. Dalam pembangunan sistem informasi *inventory control* yang dimaksud objek di sini adalah *form* pengolahan data yang diberikan kepada aktor Admin dan Petugas.

3.2.4.3.1 Jaringan Semantik Admin

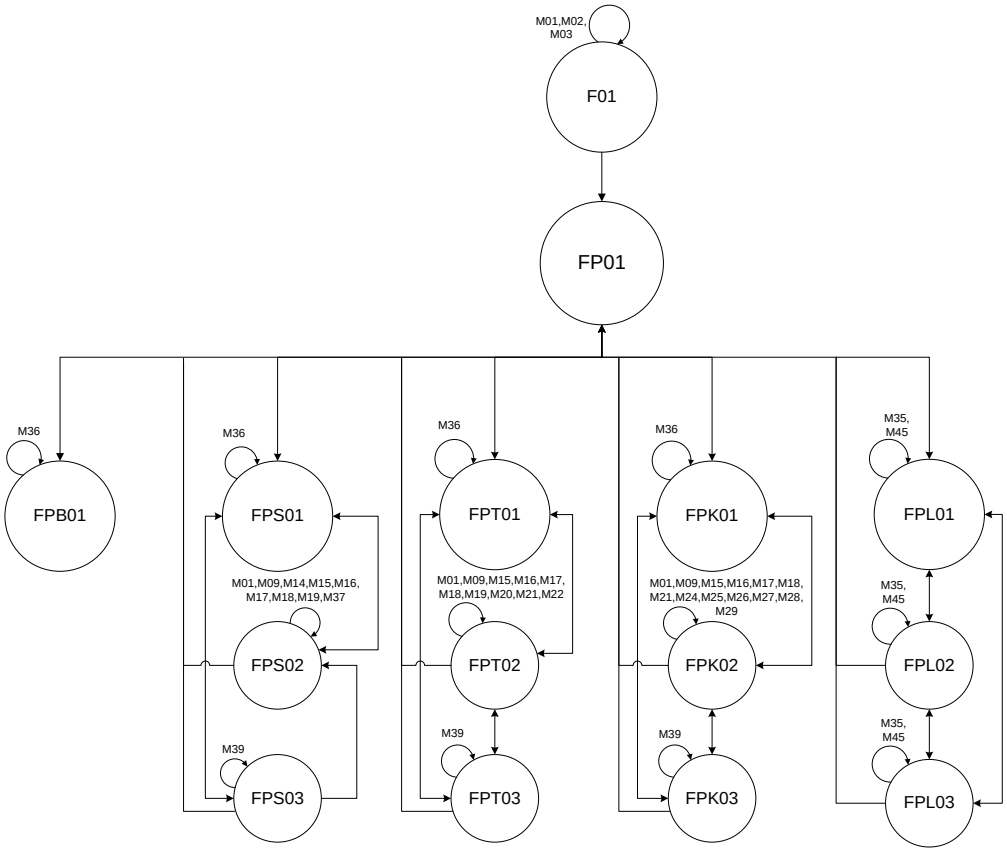
Jaringan semantik dengan aktor Admin dalam melakukan pengolahan data dapat dilihat pada Gambar 3.114.



Gambar 3.114 Jaringan Semantik Admin

3.2.4.3.2 Jaringan Semantik Petugas

Jaringan semantik dengan aktor Petugas dalam melakukan pengolahan data dapat dilihat pada Gambar 3.115.



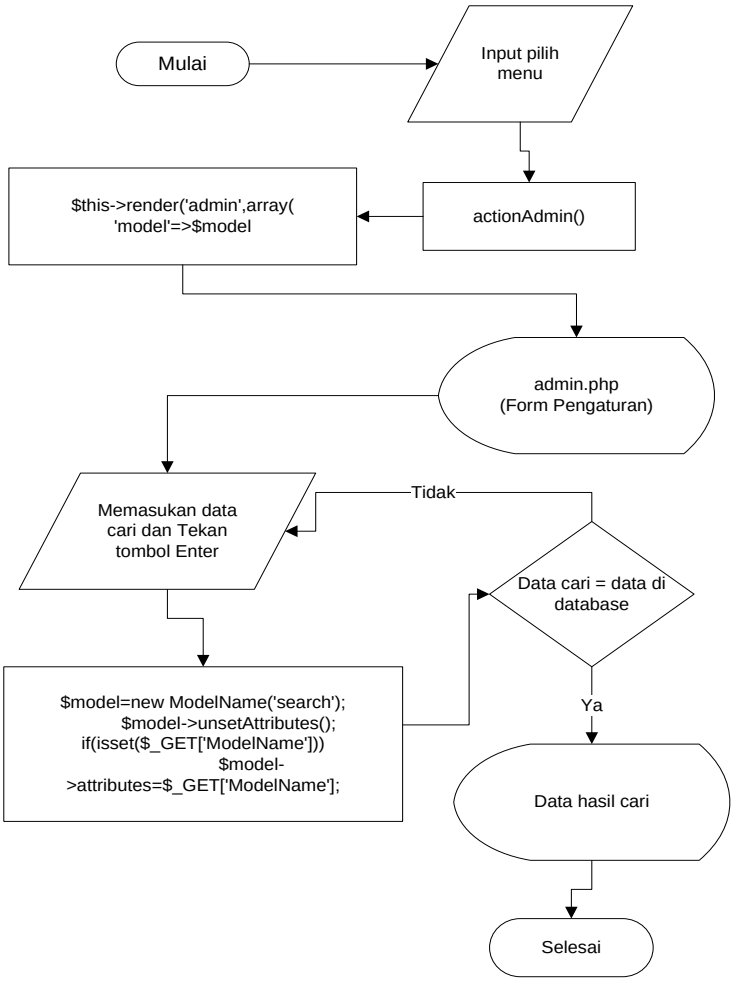
Gambar 3.115 Jaringan Semantik Petugas

3.2.5 Perancangan *Method*

Perancangan *method* merupakan kumpulan metode yang ada pada sistem yang dibangun dan berfungsi untuk menjalankan sistem. Perancangan *method* yang ada pada sistem informasi *inventory control* di antaranya adalah sebagai berikut.

1. ***Method* actionAdmin**

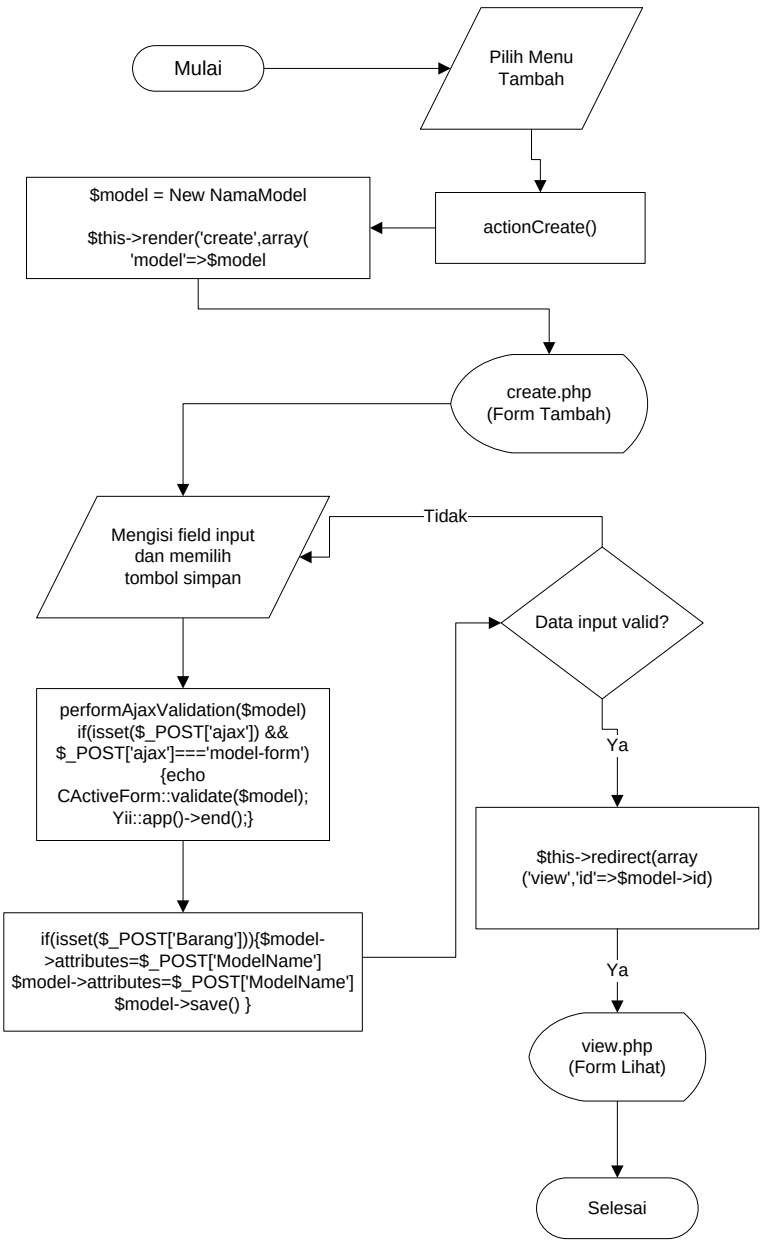
   Method ini berada pada setiap *class controller* pengolahan data dan dijalankan untuk menampilkan halaman pengaturan serta melakukan aksi cari data.



Gambar 3.116 *Method* actionAdmin

2. *Method* actionCreate

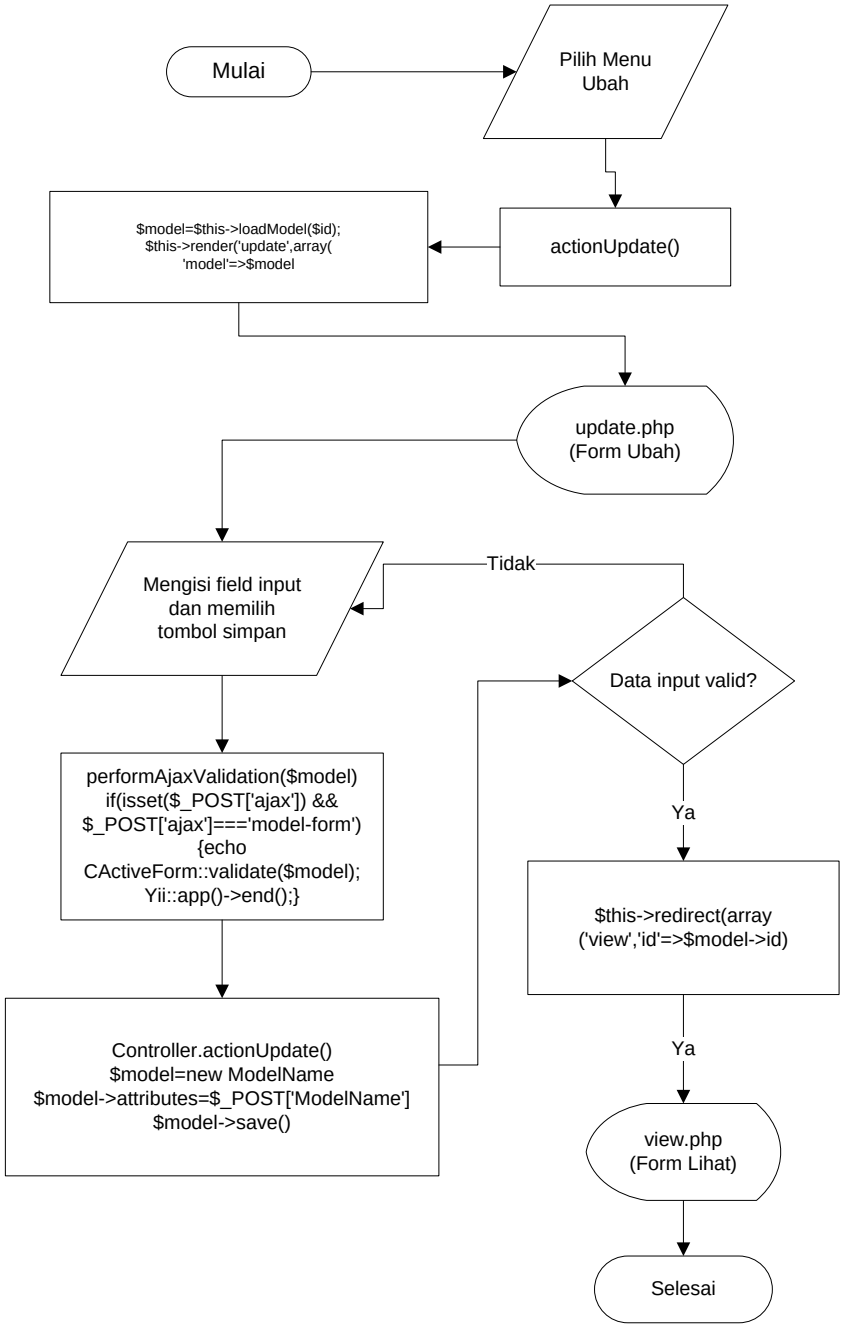
Method ini berada pada setiap *class controller* pengolahan data dan dijalankan untuk menampilkan halaman atau *form* tambah serta melakukan aksi tambah data.



Gambar 3.117 *Method* actionCreate

3. *Method* actionUpdate

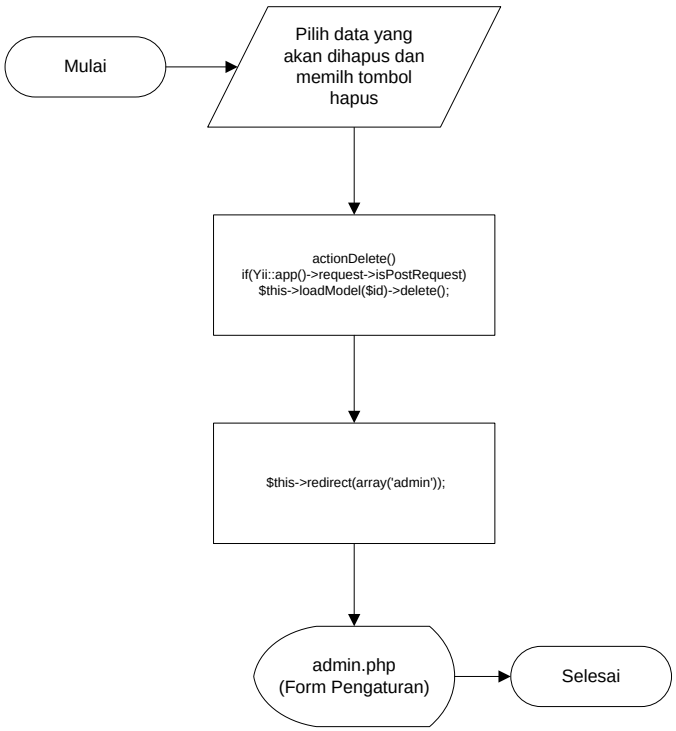
Method ini berada pada setiap *class controller* pengolahan data serta digunakan untuk menampilkan halaman atau *form* ubah dan melakukan aksi ubah data.



Gambar 3.118 *Method* actionUpdate

4. *Method* actionDelete

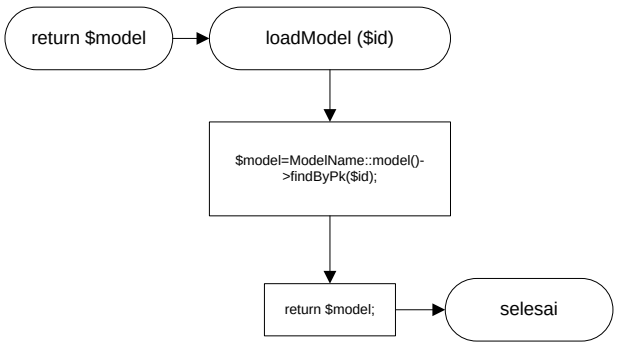
Method ini berada pada setiap *class controller* pengolahan data serta digunakan untuk melakukan aksi hapus data.



Gambar 3.119 *Method* actionDelete

5. *Method* loadModel

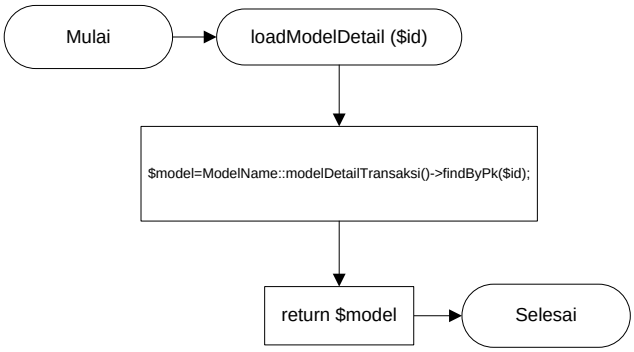
Method ini berada pada setiap *class controller* pengolahan data serta digunakan untuk mengambil data.



Gambar 3.120 *Method* loadModel

6. *Method* loadModelDetail

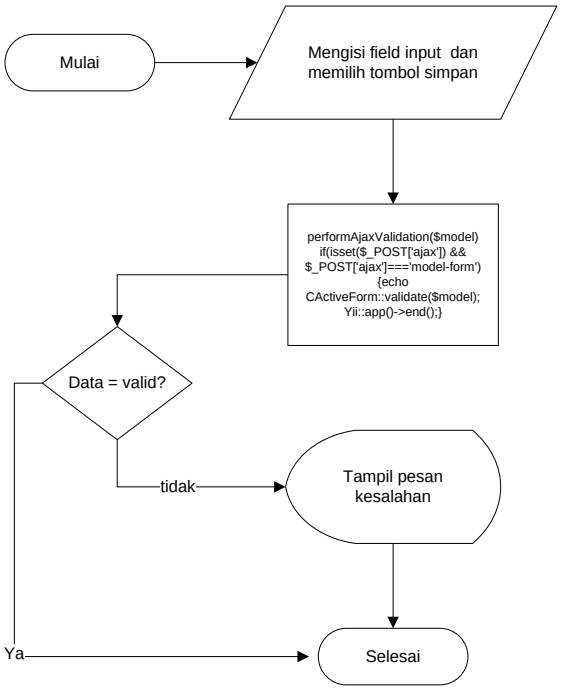
Method ini dijalankan untuk melakukan aksi *load* data detil transaksi seperti detil pemesanan, penerimaan, dan pengeluaran. *Method* ini berada pada setiap kelas *controller* pengolahan data transaksi.



Gambar 3.121 *Method* loadModelDetail

7. *Method* performAjaxValidation

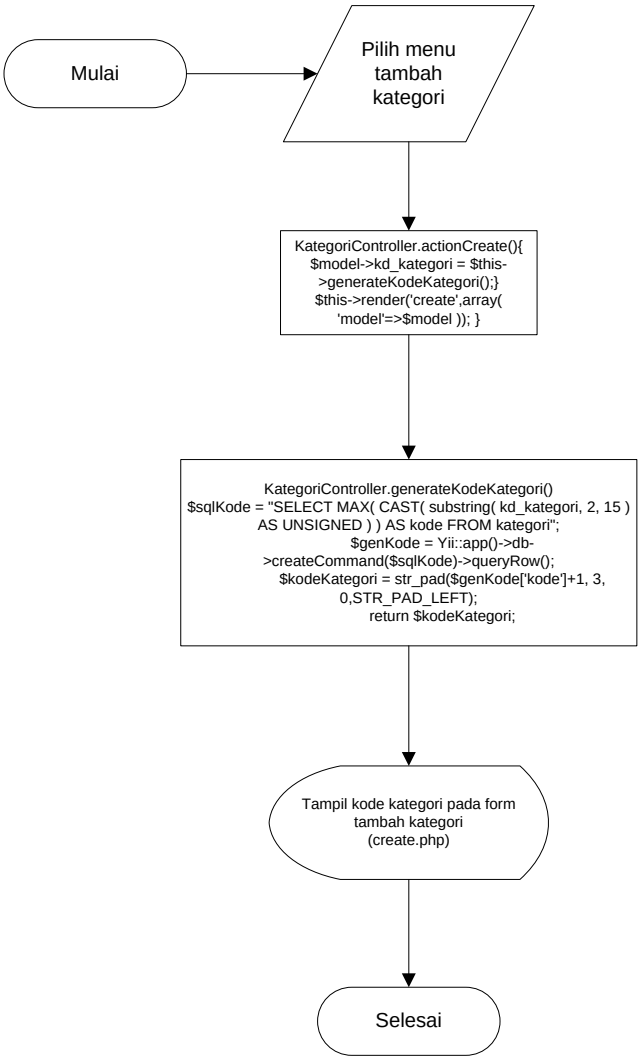
Method berada pada setiap kelas *controller* pengolahan data dan digunakan untuk melakukan aksi validasi data.



Gambar 3.122 *Method* performAjaxValidation

8. *Method* KategoriController.generateKodeKategori

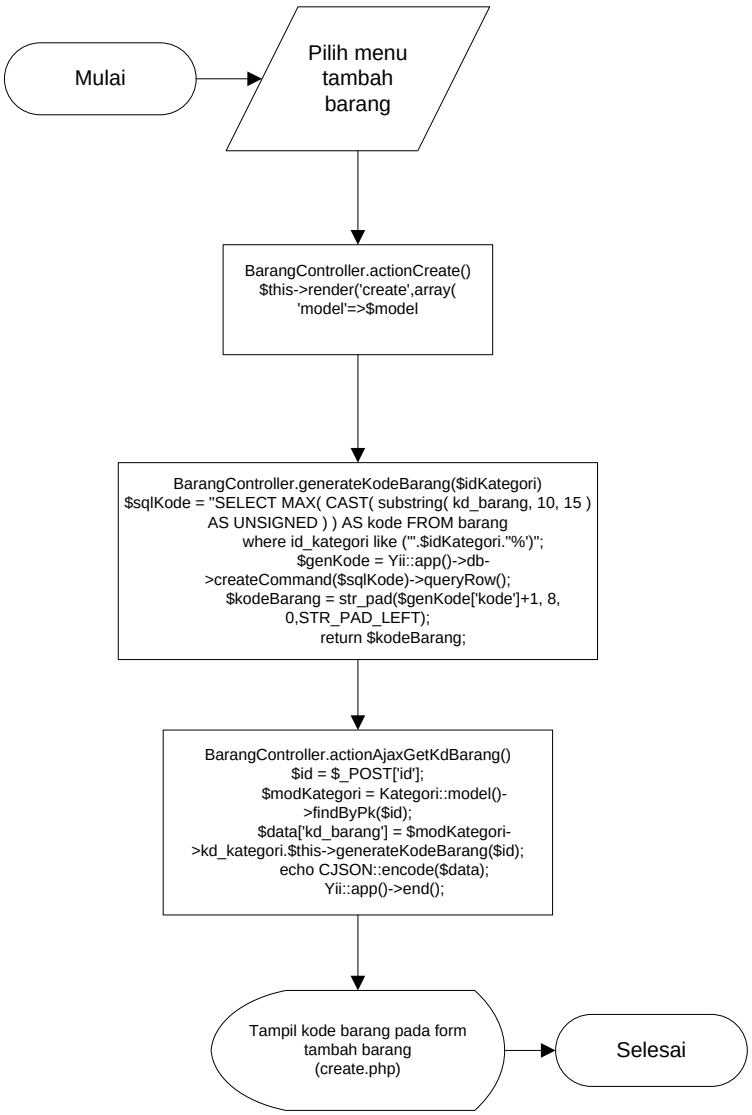
Method ini dijalankan untuk menghasilkan kode kategori. Penjelasan tentang *method* generateKodeKategori dapat dilihat Gambar 3.123.



Gambar 3.123 *Method* KategoriController.generateKodeKategori

9. *Method* BarangController.generateKodeBarang

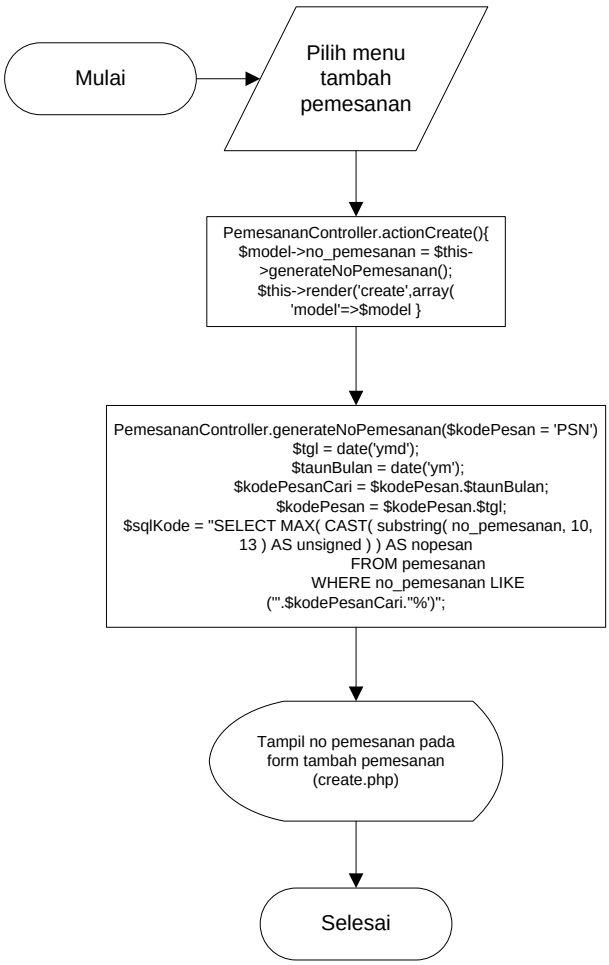
Method generateKodeBarang dijalankan untuk menghasilkan kode barang seperti yang dijelaskan pada Gambar 3.124.



Gambar 3.124 *Method* BarangController.generateKodeBarang

10. *Method* Pemesanan.generateNoPemesanan

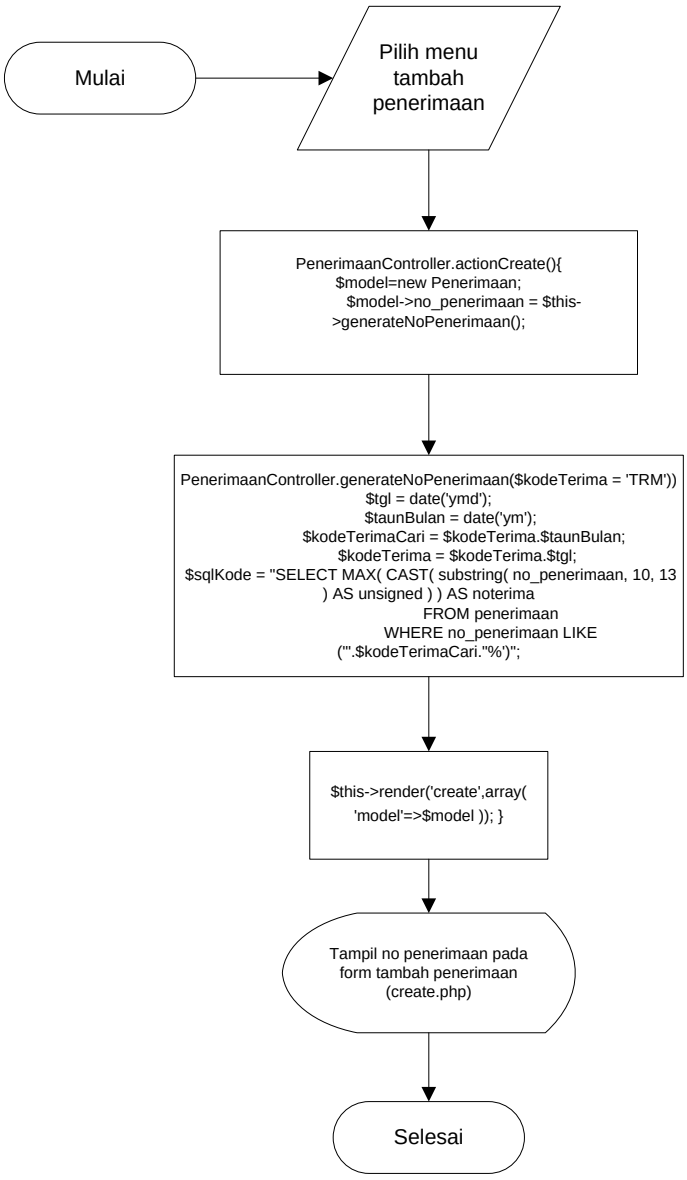
Method ini dijalankan untuk melakukan menghasilkan no pemesanan pada kelas PemesananController seperti yang dijelaskan pada Gambar 3.125.



Gambar 3.125 *Method* Pemesanan.generateNoPemesanan

11. *Method* PenerimaanController.generateNoPenerimaan

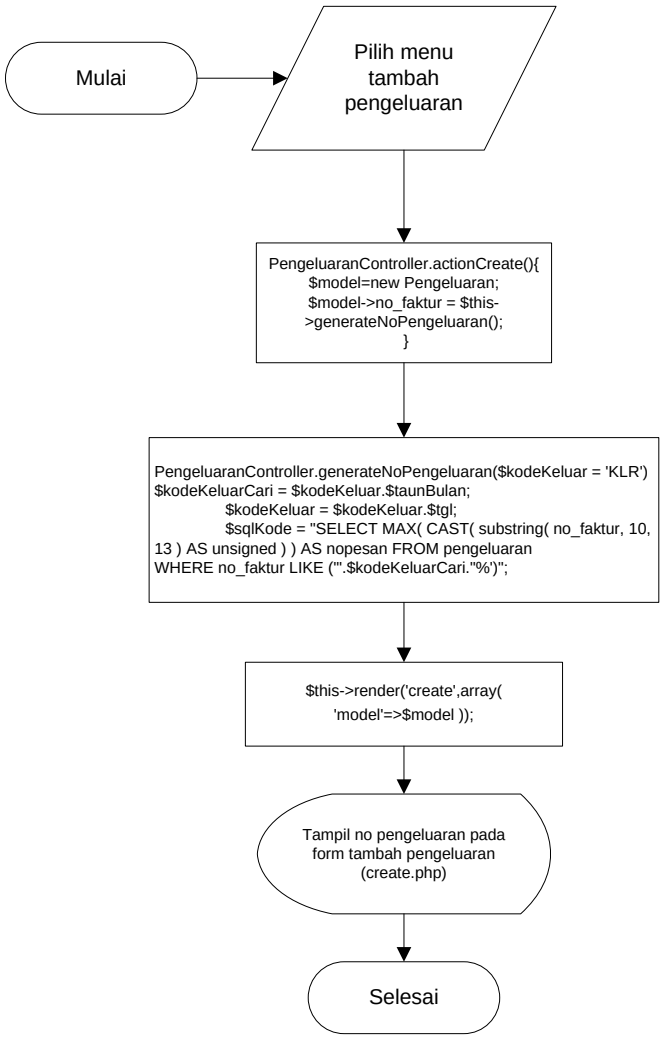
Method ini digunakan untuk menghasilkan no penerimaan yang akan digunakan pada proses tambah penerimaan. Penjelasan mengenai perancangan *method* ini dapat dilihat pada Gambar 3.126.



Gambar 3.126 *Method* PenerimaanController.generateNoPenerimaan

12. *Method* PengeluaranController.generateNoPengeluaran

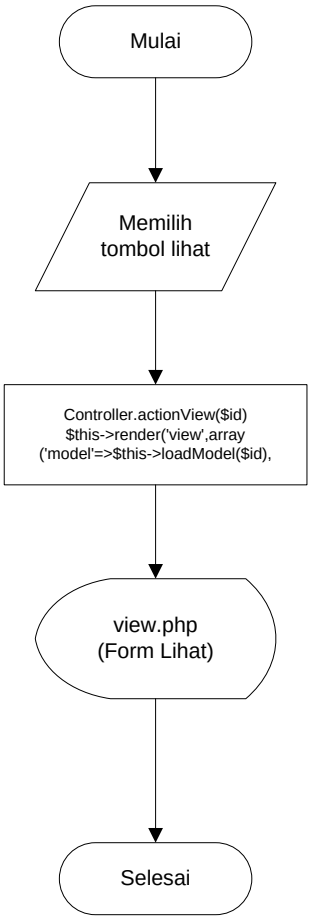
Method ini digunakan untuk melakukan menghasilkan no faktur pengeluaran digunakan pada proses tambah pengeluaran. Penjelasan mengenai perancangan *method* ini dapat dilihat pada Gambar 3.127.



Gambar 3.127 *Method* PengeluaranController.generateNoPengeluaran

13. *Method* actionView

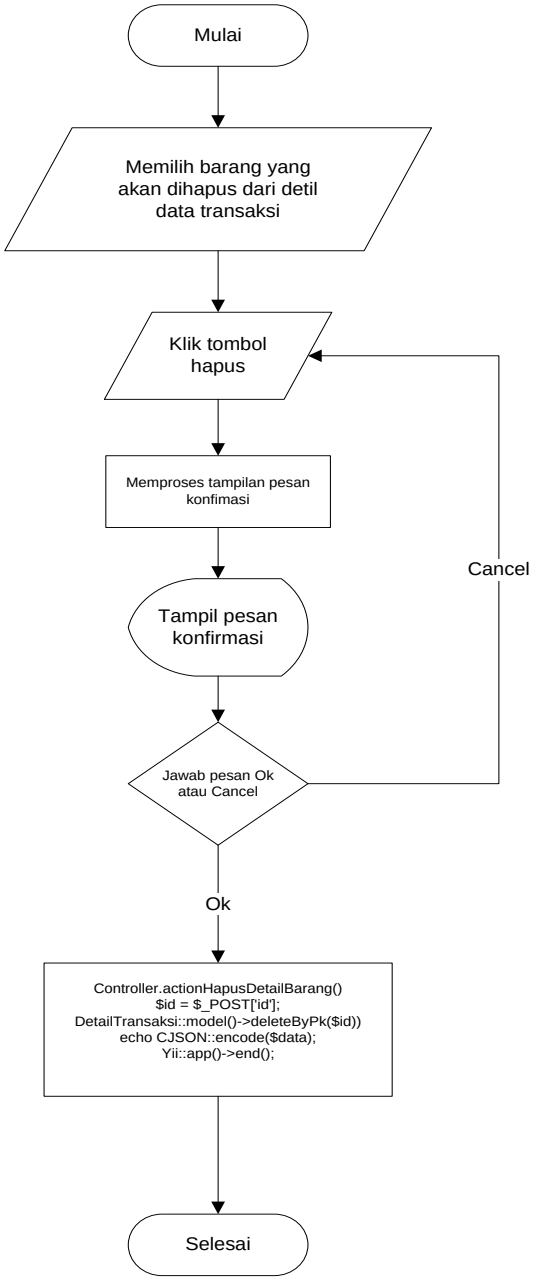
Method ini berada pada setiap kelas *controller* pengolahan data dan dijalankan untuk menampilkan data pada halaman lihat.



Gambar 3.128 *Method* actionView

14. *Method* actionHapusDetailBarang

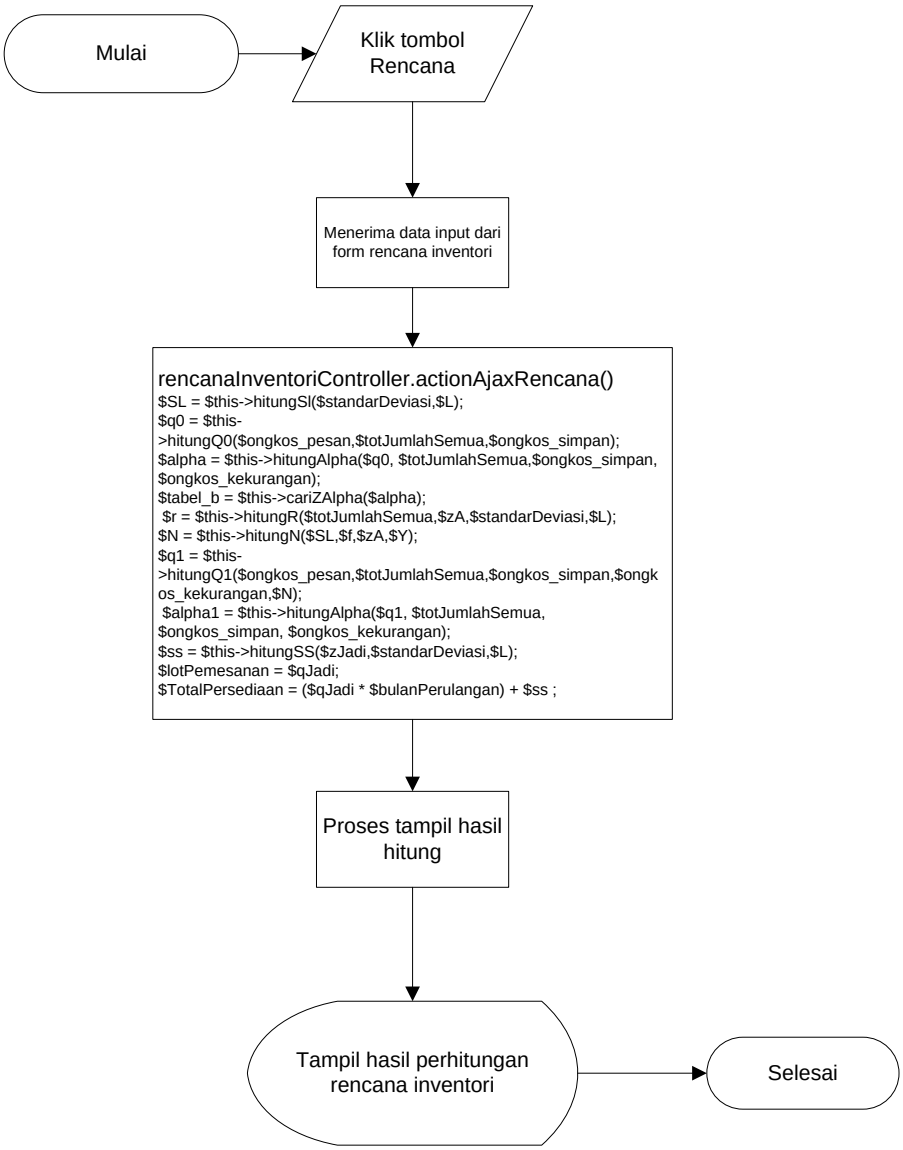
Method ini berada pada setiap kelas controller pengolahan data transaksi seperti pemesanan, penerimaan, dan pengeluaran, serta digunakan untuk menghapus data detil transaksi dari *database*.



Gambar 3.129 *Method* actionHapusDetailBarang

15. *Method* RencanaInventoriController.actionAjaxRencana

Method ini dijalankan untuk melakukan proses perhitungan rencana inventori. Penjelasan dari *method* ini dapat dilihat pada Gambar 3.130.



Gambar 3.130 *Method* RencanaInventoriController.actionAjaxRencana