

**RANCANG BANGUN APLIKASI *INVENTORY* DENGAN  
METODE *EOQ* DAN *ROP* BERBASIS *WEB* PADA GROSIR  
SEMBAKO**

**SKRIPSI**

Karya Tulis sebagai syarat memperoleh  
Gelar Sarjana Komputer dari Fakultas Teknologi Informasi  
Universitas Bale Bandung

Disusun oleh:

PRAMUDITA CAHYA FIRDAUS

NIM. C1A150037



PROGRAM STRATA 1  
PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BALE BANDUNG  
BANDUNG

2019

**LEMBAR PERSETUJUAN PEMBIMBING**

**RANCANG BANGUN APLIKASI *INVENTORY* DENGAN METODE *EOQ*  
DAN *ROP* BERBASIS *WEB* PADA GROSIR SEMBAKO**

Disusun oleh:

**PRAMUDITA CAHYA FIRDAUS**  
NIM. C1A150037

Skripsi ini Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

**SARJANA KOMPUTER**

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BALE BANDUNG**

Baleendah,     Agustus 2019

Disetujui oleh:

Pembimbing 1

Pembimbing 2

Rustiyana, S.T., M.T.  
NIK. 04104808015

Eky M Mufthi, S.T., M.Fis.  
NIK. 04104808115

## **LEMBAR PERSETUJUAN PENGUJI**

**RANCANG BANGUN APLIKASI *INVENTORY* DENGAN METODE *EOQ*  
DAN *ROP* BERBASIS WEB PADA GROSIR SEMBAKO**

Disusun oleh:

**PRAMUDITA CAHYA FIRDAUS**

**NIM. C1A150037**

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

**SARJANA KOMPUTER**

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2019

Disetujui oleh:

Penguji 1

Penguji 2

Yaya Suharya, S.Kom.,M.T

NIDN. 407047706

Zen Munawar.S.T.,M.T

NIDN. 04022037002

**LEMBAR PENGESAHAN PROGRAM STUDI**

**RANCANG BANGUN APLIKASI *INVENTORY* DENGAN METODE *EOQ*  
DAN *ROP* BERBASIS *WEB* PADA GROSIR SEMBAKO**

Disusun oleh:

**PRAMUDITA CAHYA FIRDAUS**

**NIM. C1A150037**

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

**SARJANA KOMPUTER**

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS BALE BANDUNG**

Baleendah,    Agustus 2019

Mengetahui,  
Dekan,

Mengesahkan,  
Ketua Program Studi

Yudi Herdiana, S.T, M.T  
NIK. 0410480808

Yaya Suharya, S.Kom, M.T  
NIK. 01043170007

## LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan dibawah ini:

Nama : PRAMUDITA CAHYA FIRDAUS

NIM : C1A150037

Judul Skripsi : **RANCANG BANGUN APLIKASI *INVENTORY* DENGAN METODE *EOQ* DAN *ROP* BERBASIS *WEB* PADA GROSIR SEMBAKO**

Menyatakan dengan sebenarnya bahwa penulisan skripsi ini berdasarkan hasil penelitian, pemikiran dan pemaparan asli dari saya sendiri, baik untuk naskah laporan maupun kegiatan *programming* yang tercantum sebagai bagian dari skripsi ini. Jika terdapat karya orang lain, saya mencantumkan sumber yang jelas.

Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidak benaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini dan sanksi lain sesuai dengan peraturan yang berlaku di FAKULTAS TEKNOLOGI INFORMASI UNIVERSITAS BALE BANDUNG.

Demikian surat pernyataan ini saya buat dalam keadaan sadar tanpa paksaan dari pihak manapun.

Baleendah, Agustus 2019

Yang membuat pernyataan,

**PRAMUDITA CAHYA FIRDAUS**

NIM. C1A150037

## **ABSTRACT**

*In doing this final project, the composer aims to design and build an inventory application for the asep Groceries groceries. Where the study aims to design and build an application through which the wholesaler owner can monitor and control the inventory that is at the intended wholesaler. Inventories are materials or goods (organization resources) that are stored, which will be used to meet certain objectives, for example : for the production or assembly process, for parts of equipment or for sale. EOQ is the most economical volume or amount of purchases to be carried out at each time of purchase. So by applying the EOQ model in purchasing , ordering costs and storage costs can be reduced. After the number of customer with the most economical costs is determined, then re-order is done. Re-order point explains when the order should be held again, so that the receipt of ordered materials is timely. Based on statement above, the author aims to create an application that can streamline the wholesale owner in managing inventory. In doing this research, the author uses MDD as the basis of research methodology, EOQ and ROP methods to control inventory, UML diagram to make system design. For the programming language used is Hyper text Preprocessor (PHP) with MySql as its database.*

*Keywords : Grocery Wholesale, inventory, MDD, EOQ & ROP, UML diagram, PHP, MySql*

## ABSTRAK

Dalam mengerjakan tugas akhir ini penyusun bertujuan untuk merancang dan membangun aplikasi *inventory* (persediaan barang) pada grosir sembako pak asepi. Dimana penelitian ini bertujuan untuk merancang dan membangun suatu aplikasi yang mana melalui aplikasi tersebut pemilik grosir dapat memantau dan mengendalikan *inventory* (persediaan barang) yang ada pada grosir yang dimaksud. Persediaan (*Inventory*) adalah bahan-bahan atau barang (sumberdaya-sumberdaya organisasi) yang di simpan, yang akan di pergunakan untuk memenuhi tujuan tertentu, misalnya: untuk proses produksi atau perakitan, untuk suku cadang dari peralatan, maupun untuk dijual. *EOQ* merupakan volume atau jumlah pembelian yang paling ekonomis untuk di laksanakan pada setiap kali pembelian. Sehingga dengan menerapkan model *EOQ* dalam pembelian, biaya pemesanan dan biaya penyimpanan dapat ditekan, setelah jumlah pemesanan dengan biaya yang paling ekonomis di tentukan maka di lakukan lah *re-order*. *Reorder Point* menjelaskan kapan saat seharusnya diadakan pemesanan lagi, sehingga penerimaan bahan yang dipesan tepat pada waktunya. Berdasarkan pernyataan di atas penyusun bertujuan untuk membuat suatu aplikasi yang dapat mengefisiensikan pemilik grosir dalam mengelola *inventory*. Dalam mengerjakan penelitian ini penyusun menggunakan *MDD* sebagai dasar metodologi penelitian, metode *EOQ* dan *ROP* untuk mengendalikan persediaan, *UML Diagram* untuk membuat perancangan sistem. Untuk bahasa pemrograman yang di gunakan adalah *Hypertext Preprocessor (PHP)* dengan *MySQL* sebagai *database*-nya.

Kata kunci : Grosir Sembako, *inventory*, *MDD*, *EOQ* & *ROP* ,*UML Diagram*, *PHP*, *MySQL*.

## KATA PENGANTAR

Puji dan syukur penulis panjatkan atas kehadiran Allah SWT yang mana berkat Rahmat dan Karunia-NYA penulis dapat menyelesaikan Laporan Penelitian ini dengan judul **“RANCANG BANGUN APLIKASI *INVENTORY* DENGAN METODE *EOQ* DAN *ROP* BERBASIS *WEB* PADA GROSIR SEMBAKO”** dengan baik dan tepat pada waktunya. Penelitian ini disusun untuk menyelesaikan tugas akhir guna memperoleh gelar Sarjana Komputer.

Laporan ini dibuat dengan berbagai observasi, wawancara dan beberapa bantuan dari berbagai pihak yang membantu menyelesaikan proses pengerjaan Penelitian ini. Oleh karena itu, penulis ucapkan terima kasih sebesar-besarnya kepada:

1. Yudi Herdiana, S.T., M.T, selaku Dekan Fakultas Teknologi Informasi Universitas Bale Bandung.
2. Yaya Suharya, S.Kom., M.T, selaku Ketua Prodi Teknik Informatika Fakultas Teknologi Informasi Universitas Bale Bandung.
3. Rustiyana, S.T, M.T dan Eky M Mufthi, S.T, M.Fis, selaku Dosen Pembimbing di Prodi Teknik Informatika Fakultas Teknologi Informasi Universitas Bale Bandung.
4. Keluarga yang memberikan dukungan secara moril dan materil serta do'a yang tak pernah putus dalam proses pengerjaan laporan.
5. Rekan-rekan seperjuangan yang saling membantu dalam proses pengerjaan laporan penelitian.

Penulis menyadari sepenuhnya bahwa laporan ini masih jauh dari kesempurnaan, oleh karena itu kritik dan saran dari semua pihak yang bersifat membangun selalu penulis harapkan demi kesempurnaan laporan ini.

Bandung,      Agustus 2019

Penulis



## DAFTAR ISI

<i>ABSTRACT</i> .....	v
ABSTRAK .....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL .....	xiii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan.....	3
1.5 Manfaat Penelitian .....	3
1.6 Metodologi Penelitian .....	3
1.7 Sistematika Penulisan .....	4
BAB II TINJAUAN PUSTAKA.....	6
2.1 Landasan Teori .....	6
2.1.1 Jurnal 1 .....	6
2.1.2 Jurnal 2 .....	7
2.1.3 Jurnal 3.....	7
2.2 Dasar Teori .....	8
2.2.1 <i>Inventory</i> (Persediaan Barang).....	8
2.2.2 Pengendalian Persediaan.....	9
2.2.3 <i>Economic Order Quantity</i> .....	10
2.2.4 <i>Reorder Point</i> .....	11
2.2.5 Rancang.....	12
2.2.6 Bangun .....	12
2.2.7 Aplikasi .....	13
2.2.8 <i>Website</i> .....	13
2.2.9 <i>My Structured Query Language (MySQL)</i> .....	14
2.2.10 <i>Hypertexy Preprocessor (PHP)</i> .....	16

2.2.11	<i>Unified Modeling Language (UML)</i> .....	16
2.2.12	<i>Model Driven Developement (MDD)</i> .....	20
2.2.13	Persamaan Dengan Riset Lain.....	22
2.2.14	Perbedaan Dengan Riset Lain .....	22
BAB III METODOLOGI PENELITIAN.....		23
3.1	Kerangka Berfikir .....	23
3.2	Deskripsi.....	24
3.2.1	Investigasi Pendahuluan.....	24
3.2.2	Analisa Masalah.....	26
3.2.3	Analisis Kebutuhan .....	27
3.2.4	Desain.....	28
3.2.5	Kontruksi.....	30
3.2.6	Impementasi .....	31
3.2.7	Pembuatan Laporan.....	31
BAB IV ANALISIS DAN PERANCANGA SISTEM .....		32
4.1	Analisis .....	32
4.1.1	Analisis Masalah .....	32
4.1.2	Analisis Sistem.....	33
4.1.3	Analisis Kebutuhan .....	36
4.1.4	Analisis Pengguna.....	37
4.1.5	<i>User Interface</i> .....	37
4.1.6	Fitur-Fitur.....	38
4.2	Perancangan Sistem.....	38
4.2.1	<i>Usacase Diagram</i> .....	39
4.2.2	Activity Diagram.....	41
4.2.3	Class Diagram .....	51
4.2.4	Perancangan <i>Database</i> .....	51
4.2.5	Desain.....	55
BAB VIMPLEMENTASI DAN PENGUJIAN .....		65
5.1	Implementasi .....	65
5.1.1	Implementasi Perangkat Keras dan Perangkat Lunak.....	65
5.2	Pengujian Sistem .....	66

5.3 Menghitung <i>Economic Order Quantity</i> dan <i>Re-order Point</i> .....	77
BAB VI .....	78
KESIMPULAN DAN SARAN .....	78
1.1 Kesimpulan .....	78
1.2 Saran .....	78
DAFTAR PUSTAKA .....	79
LAMPIRAN .....	80

## DAFTAR GAMBAR

Gambar 2.1 <i>Model Driven Developement</i> .....	21
Gambar 3.2 Model MDD ( <i>Model Driven Developement</i> ) .....	23
Gambar 3.3 Buku Nota Belanja Grosir Sembako .....	25
Gambar 3.4 Nota Belaja Grosir Sembako.....	26
Gambar 4.5 <i>Flow Map</i> Diagram Sistem yang sedang berjalan.....	34
Gambar 4.6 <i>Flow Map</i> yang di Usulkan .....	35
Gambar 4.7 <i>Use case</i> diagram yang di usulkan .....	39
Gambar 4.8 <i>Activity</i> diagram <i>login</i> .....	41
Gambar 4.9 <i>Activity</i> diagram olah data barang .....	42
Gambar 4.10 <i>Activity</i> diagram olah data barang masuk.....	43
Gambar 4.11 <i>Activity</i> diagram olah data barang keluar .....	44
Gambar 4.12 <i>Activity</i> diagram olah laporan order .....	45
Gambar 4.13 <i>Activity</i> diagram <i>logout</i> .....	46
Gambar 4.14 <i>Activity</i> diagram lapoan data barang .....	47
Gambar 4.15 <i>Activity</i> diagram data barang masuk.....	48
Gambar 4.16 <i>Activity</i> diagram data barang keluar .....	49
Gambar 4.17 <i>Activity</i> diagram order barang .....	50
Gambar 4.18 <i>Class</i> diagram aplikasi pengendalian <i>inventory</i> .....	51
Gambar 4.19 Desain Halaman <i>login</i> aplikasi.....	55
Gambar 4.20 Desain Halaman awal aplikasi .....	56
Gambar 4.21 Desain Halaman data barang.....	57
Gambar 4.22 Desain Halaman <i>input</i> data barang.....	58
Gambar 4.23 Desain Halaman data barang masuk .....	60
Gambar 4.24 Desain Halaman input data barang masuk .....	61
Gambar 4.25 Desain Halaman data barang keluar.....	62
Gambar 4.26 Desain Halaman <i>input</i> data barang keluar.....	63
Gambar 5.27 Tampilan <i>Interface</i> Menu <i>Login</i> .....	66
Gambar 5.28 Tampilan <i>Interface</i> Dashboard Aplikasi .....	67

Gambar 5.29 Tampilan <i>Interface</i> Halaman Data Barang.....	69
Gambar 5.30 Tampilan <i>Interface</i> Halaman Input Data Barang .....	69
Gambar 5.31 Tampilan <i>Interface</i> Halaman Barang Masuk .....	71
Gambar 5.32 Tampilan <i>Interface</i> Halaman Input Data Barang Masuk .....	71
Gambar 5.33 Tampilan <i>Interface</i> Halaman Barang Keluar .....	73
Gambar 5.34 Tampilan <i>Interface</i> Halaman Barang Keluar .....	73
Gambar 5.35 Tampilan Halaman <i>Form User</i> .....	74
Gambar 5.36 Tampilan Halaman <i>Rules</i> .....	75
Gambar 5.37 Tampilan <i>Interface Form</i> isi data untuk <i>Rules</i> .....	76

## DAFTAR TABEL

Tabel 2.1 <i>Element-elemnt</i> pada <i>Use Case Diagram</i> .....	19
Tabel 3.2 Spesifikasi Minimum <i>Hardware</i> .....	28
Tabel 4.3 Kebutuhan Sistem .....	36
Tabel 4.4 Spesifikasi perangkat keras .....	36
Tabel 4.5 Spesifikasi perangkat lunak untuk implementasi .....	37
Tabel 4.6 Deskripsi <i>Usecase</i> .....	40
Tabel 4.7 Perancangan <i>Database</i> Data barang .....	52
Tabel 4.8 Perancangan <i>Database</i> Data Barang Masuk.....	53
Tabel 4.9 Perancangan <i>Database</i> Data Barang Keluar .....	53
Tabel 4.10 Perancangan <i>Database</i> Data <i>Rules</i> .....	54
Tabel 4.11 Perancangan <i>Database</i> Data <i>User</i> .....	54
Tabel 4.12 Keterangan tombol pada halaman <i>login user</i> .....	55
Tabel 4.13 Keterangan tombol pada halaman awal apliaksi.....	56
Tabel 4.14 Keterangan tombol pada halaman data barang .....	58
Tabel 4.15 keterangan tombol pada halaman <i>input</i> data barang .....	59
Tabel 4.16 Keterangan tombol pada halaman data barang masuk.....	60
Tabel 4.17 Keterangan tombol pada halaman <i>input</i> data barang masuk.....	61
Tabel 4.18 Keterangan tombol pada halaman data barang keluar .....	62
Tabel 4.19 Keterangan tombol pada halaman input data barang keluar .....	64
Tabel 5.20 Spesifikasi <i>Hardware</i> yang di gunakan .....	65
Tabel 5.21 Pengujian <i>Login</i> .....	67
Tabel 5.22 Pengujian <i>dashboard</i> aplikasi .....	68
Tabel 5.23 Pengujian halaman data barang.....	70
Tabel 5.24 Pengujian halaman barang masuk .....	72
Tabel 5.25 Pengujian Halaman barang keluar .....	74
Tabel 5.26 Pengujian <i>form user</i> .....	75
Tabel 5.27 Pengujian halaman <i>rules</i> .....	76

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Salah satu jenis usaha wiraswasta yang bergerak dalam bidang dagang (*retailer*), harus memiliki *inventory* (persediaan barang). Tujuan utama dari penyimpanan persediaan adalah untuk memenuhi kebutuhan konsumen, di sisi lain pemilik usaha diuntut untuk mampu mengendalikan *inventory* (persediaan barang) untuk menghindari terjadinya *over stock*, sehingga pemilik usaha mendapatkan keuntungan yang optimal.

Grosir Sembako adalah salah satu jenis usaha wiraswasta yang bergerak dalam bidang dagang (*retailer*) yang menjual kebutuhan pokok hidup seperti penjualan mie instan, kopi, beras, alat tulis dan kebutuhan hidup lainnya. Pemilik Grosir bertugas mengelola segala aktivitas yang menyangkut pada produksi produk. saat ini Grosir Sembako menyediakan lebih dari 100 jenis produk dengan rata-rata produksi per minggu mencapai 100 hingga 200 pcs untuk sebagian jenis produk.

Dalam operasional pengendalian *inventory* (persediaan barang), untuk menentukan jumlah produk yang akan disediakan ulang, Pemilik Grosir melihat dari penjualan perhari, sedangkan untuk menentukan kapan dilakukan pemesanan (*restock*), Pemilik Grosir melihat rata-rata penjualan produk dari hari sebelumnya, jika jumlah produk mencapai jumlah rata-rata penjualan dan produk barang yang di sediakan hampir habis terjual maka akan dilakukan penyediaan kembali. Dengan proses pengendalian persediaan yang dilakukan seperti saat ini, Pemilik Grosir Sembako mengalami berbagaimacam permasalahan. Seperti lupa mencatatbarang yang harus di sediakan kembali sehingga menimbulkan kosongnya

salah satu produk yang seharusnya di sediakan dan tidak mampu memenuhi kebutuhan konsumen.

Penelitian sebelumnya telah di lakukan oleh (Tomi lukmana, Diana Trivena Y, 2015) dengan judul “Penerapan Metode EOQ dan ROP (Studi Kasus: PD. BARU)”. Penelitian ini memiliki kelemahan yakni aplikasi ini hanya bisa di akses oleh satu *user* yaitu oleh bagian *inventory*, sehingga tidak dapat di akses oleh pemilik perusahaan dari luar kator.

Berdasarkan masalah yang ada di atas Grosir Sembako membutuhkan sebuah aplikasi *inventory* (persediaan barang) yang mampu menentukan berapa banyaknya jumlah item produk yang akan dipesan pada suatu waktu dan kapan dilakukan pemesanan ulang terhadap item produk tersebut.

## 1.2 Rumusan Masalah

Dari permasalahan diatas maka yang menjadi masalah pokok dalam penelitian ini dapat dirumuskan, adalah ;

- bagaimana cara merancang dan membangun aplikasi *inventory* (persediaan barang) pada Grosir Sembako ?
- Bagaimana cara mengefisiensikan pemilik grosir dalam mengelola *inventory* pada grosir sembako ?

## 1.3 Batasan Masalah

Berdasarkan rumusan masalah diatas, adapun batasan masalah pada penelitian ini sebagai berikut:

1. Dalam merancang aplikasi, penyusun menggunakan metode pemograman terstruktur dengan *Model Driven Developement* (MDD) dan UML Diagram untuk membuat perancangan sistem.



2. Aplikasi ini meliputi input data barang masuk, input data barang keluar dan perencanaan persediaan produk.
3. Aplikasi bisa di akses oleh *administrator* dan *owner* Grosir.
4. Aplikasi tidak membahas retur produk.

## 1.4 Tujuan

Berdasarkan rumusan masalah di atas, maka tujuan yang ingin dicapai adalah ;

- Membuat aplikasi inventory pada grosir sembako.
- Meng-efisiensikan pemilik grosir dalam mengelola *inventory* pada grosir sembako.

## 1.5 Manfaat Penelitian

Manfaat yang diharapkan penyusun melalui penelitian ini adalah :

1. Menambah wawasan penyusun tentang aplikasi *inventory*, bahasa pemrograman PHP, dan menambah pengalaman yang diharapkan dapat memberi manfaat di kemudian hari.
2. Menghasilkan aplikasi pengendalian *inventory* pada grosir sembako.
3. Meminimalisir terjadinya kesalahan dalam menentukan barang yang akan di sediakan kembali.

## 1.6 Metodologi Penelitian

Metode yang digunakan untuk mengembangkan aplikasi ini menggunakan *Model Driven Development*. Yang terdiri dari tahap-tahap di bawah ini:

- 1) Investigasi Pendahuluan

- 2) Analisis Masalah
- 3) Analisis Kebutuhan
- 4) Perancangan
- 5) Kontruksi
- 6) Implementasi

## **1.7 Sistematika Penulisan**

Dalam menyusun laporan tugas akhir ini secara sistematis diatur dan disusun dalam lima bab, yang masing-masing terdiri dari beberapa sub bab. Adapun urutan dari bab pertama sampai bab terakhir adalah sebagai berikut:

### **BAB I : PENDAHULUAN**

Bab ini membahas tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan, metodologi penelitian, dan sistematika penulisan.

### **BAB II : TINJAUAN PUSTAKA**

Bab ini membahas tentang ladsan teori, dan dasar teori yang akan penyusun gunakan dalam penyusunan tugas ahir ini.

### **BAB III : METODOLOGI PENELITIAN**

Bab ini berisi uraian tentang desain penelitian, operasional variabel dan pengukuran, populasi dan sampel penelitian, teknik pengumpulan data dan teknik analisis data yang digunakan, rancangan uji hipotesis serta jadwal penelitian.

### **BAB IV ANALISIS DAN PERANCANGAN**

Bab ini membahas tentang analisis lebih lanjut terhadap masalah yang ada, perangkat lunak, perangkat keras, dan berbagai macam analisis yang di butuhkan

untuk pembuatan aplikasi. Juga menjelaskan tentang perancangan aplikasi di mulai dari flowmap diagram, DFD dan lain sebagainya.

## **BAB V IMPLEMENTASI DAN PENGUJIAN**

Bab ini membahas tentang implementasi yang mengacu pada perancangan desain sistem yang telah dibuat dan berfokus pada pembangunan aplikasi pengendalian stok produk beserta pengujian-nya.

## **BAB VI KESIMPULAN DAN SARAN**

Bab ini membahas tentang kesimpulan yang diperoleh dari pembuatan sistem ini serta saran yang bertujuan untuk pengembangan sistem dimasa yang akan datang.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Landasan Teori**

Landasan Teori berisi referensi dari jurnal yang berkaitan dengan judul dan objek penelitian, berikut beberapa judul jurnal yang digunakan dalam proses penelitian ini:

##### **2.1.1 Penerapan Metode *EOQ* dan *ROP* (Studi Kasus: PD. BARU). (Tomi Lukmana, Diana Trivena, Jurnal Teknik Informatika dan Sistem Informasi Volume 1 Nomor 3 Desember 2015)**

**Abstrak :** Pd. Baru tidak menggunakan teknologi yang dapat mengelola sistem di komputerisasi, masih menyimpan catatan data secara manual. Sehingga risiko sangat mudah untuk kehilangan data transaksi dan memerlukan perhitungan optimal alat ukur pembelian tahun berikutnya dan untuk mengetahui titik persediaan aman saham untuk membuat reservasi kembali. Oleh karena itu, membuat aplikasi desktop yang memiliki fitur untuk mengelola penjualan, pembelian dan inventaris menggunakan ekonomi urutan kuantitas dan Re order Point. Aplikasi yang dibuat menggunakan bahasa pemrograman C# dan menggunakan database SQL Server 2008. Aplikasi ini sistem pengujian menggunakan metode kotak hitam, dengan menggunakan metode kotak hitam ini dapat menunjukkan bahwa aplikasi telah mampu memenuhi hasil yang diharapkan. Diharapkan setelah PD. Baru untuk mengimplementasikan aplikasi ini, PD. Baru dapat mengelola dengan baik.

**2.1.2 Aplikasi Manajemen Persediaan Barang Berbasis *Economic Order Quantity (EOQ)* (Taufik Iqbal, Daniel Aprizal, Muhammad Wali, JTIK : Jurnal Teknologi Informasi dan Komunikasi, 7 juni 2017)**

**Abstrak :** Manajemen persediaan barang merupakan kegiatan yang dilakukan oleh suatu perusahaan yang diperlukan dalam membuat keputusan sehingga kebutuhan akan bahan ataupun barang untuk keperluan kegiatan perusahaan baik produksi maupun penjualan dapat terpenuhi secara optimal dengan resiko yang sekecil mungkin. Persediaan yang terlalu besar (over stock) merupakan pemborosan karena menyebabkan terlalu tingginya beban-beban biaya guna penyimpanan dan pemeliharaan selama penyimpanan di gudang. Penelitian ini mencoba membangun sebuah aplikasi persediaan dengan menerapkan metode EOQ (Economic Order Quantity) sebagai dasar pengembangan. Metode penelitian ini dibagi dalam tiga tahapan, yaitu pengumpulan data pra pengembangan, pengembangan serta implementasi, dan pengumpulan data pasca pengembangan. Hasil dari penelitian ini merupakan aplikasi persediaan dengan metode EOQ Probabilistik yang dengan menitikberatkan hasil dengan bentuk grafik untuk mempermudah pemilik perusahaan dalam mengambil keputusan. Sistem manajemen persediaan yang dibangun dengan menggunakan bahasa pemrograman HTML, CSS, JQuery, JavaScript, JSON, AJAX, Bootstrap sebagai media dalam perancangan antar muka. Sedangkan PHP sebagai server side dan MySQL sebagai database.

**2.1.3 Aplikasi Optimasi dan Efisiensi Persediaan Menggunakan Metode *EOQ, Safety Stock* dan *ROP* di Toko Batik Nusantara (Turmudi, Artikel Teknik Informatika Universitas Nusantara PGRI Kediri 2017).**

**Abstrak :** Sistem persediaan pada Toko Batik Nusantara saat ini masih menggunakan sistem manual. Yang perlu diketahui bahwa persediaan sangat dipengaruhi 2 hal penting yaitu penjualan dan pengadaan sehingga persediaan berhubungan erat dengan keduanya. Sistem manual dalam hal ini menjadi permasalahan pokok sehingga sering terjadi kesalahan terutama dalam aktifitas kerja

seperti pencatatan, penghitungan data barang dan laporan-laporan. Sistem manual juga menyebabkan proses pelayanan menjadi tidak praktis dan tidak efisien. Untuk mengatasi permasalahan tersebut, maka dibutuhkan suatu sistem baru yang dapat mengganti sistem yang lama yaitu mengubah sistem manual menjadi sistem komputer. Selanjutnya untuk mengendalikan persediaan dan tidak terjadi kehabisan stok, maka dalam aplikasi yang dibuat menggunakan metode *Economic Order Quantity*, *Safety Stock* dan *Re-order Point*. Perancangan dan pembuatan aplikasi penjualan, dan persediaan pada sistem yang baru adalah menggunakan Bahasa pemrograman Java untuk desain tampilannya dan MySQL untuk desain dan pembuatan databasenya. Pada tahap awal yang perlu dilakukan dalam merancang sistem adalah pembuatan *Flowchart System* dan *Entity Relation Diagram* (ERD). Hasil dari pembuatan sistem ini adalah tercipta sistem yang membantu pihak Toko Batik Nusantara Kediri dalam pengelolaan persediaan antara lain : Pengadaan, Penjualan, Pelaporan dan lain-lain. Dasar Teori

## 2.2 Dasar Teori

Landasan Teori berisi referensi dari jurnal yang berkaitan dengan judul dan objek penelitian, berikut beberapa judul jurnal yang digunakan dalam proses penelitian ini:

### 2.2.1 Inventory (Persediaan Barang)

Persediaan (*Inventory*) adalah bahan-bahan atau barang (sumberdaya-sumberdaya organisasi) yang disimpan, yang akan digunakan untuk memenuhi tujuan tertentu, misalnya: untuk proses produksi atau perakitan, untuk suku cadang dari peralatan, maupun untuk dijual. Walaupun persediaan hanya merupakan suatu sumber dan yang menganggur, akan tetapi dapat dikatakan tidak ada perusahaan yang beroperasi tanpa persediaan. (Hengki Fitrayco dan Habibi, 2009)

Persediaan memiliki banyak sekali manfaat bagi dunia usaha. Beberapa manfaat persediaan dalam dunia usaha yaitu :

- Sebagai antisipasi kemungkinan terjadinya keterlambatan kedatangan barang ataupun barang-barang yang di butuhkan perusahaan dalam aktifitas usahanya.
- Sebagai antisipasi kemungkinan terjadinya cacat pada barang yang dipesan sehingga harus di retur kembali ke perusahaan asal.
- Sebagai antisipasi terjadinya kelangkaan barang-barang tertentu yang tidak dapat di produksi sepanjang musim.
- Untuk mempertahankan dan menjaga aktifitas operasional perusahaan sekaligus menjamin keberlangsungan aktifitas produksi dalam perusahaan.
- Untuk mengantisipasi terjadinya kehabisan barang yang seharusnya di sediakan.

### **2.2.2 Pengendalian Persediaan**

Pengendalian persediaan adalah suatu usaha yang di lakukan suatu perusahaan untuk menjaga stabilitas produksi, sehingga dapat memenuhi kebutuhan kosumen. Dengan dilakukannya pengendalian persediaan jumlah produk yang di sediakan akan stabil sehingga dapat meghemat biaya produksi dan memaksimalkan keuntungan perusahaan.

Tujuan utama dari pengendalian persediaan adalah agar perusahaan selalu mempunyai persediaan dalam jumlah yang tepat, pada waktu yang tepat dan dalam spesifikasi atau mutu yang telah di tentukan, sehingga kontinuitas usaha dapat terjamin (tidak terganggu) (Hengki Fitrayco dan Habibi, 2009).

Jika suatu perusahaan tidak mampu mengendalikan persediaan, maka dapat di pastikan perusahaan tersebut tidak akan bertahan lama. Berikut beberapa kemungkinan yang akan terjadi apabila persediaan tidak terkendali (beberapa item produk kosong):

- Konsumen menagguhkan pembelian (jika kebutuhannya tidak mendesak). Hal ini dapat mengakibatkan tertundanya kesempatan memperoleh keuntungan.
- Konsumen membeli dari pesaing, dan kembali ke perusahaan (jika keadaan mendesak dan masih setia). Hal ini akan menimbulkan kehilangan kesempatan memperoleh keuntungan selama persediaan tidak ada.
- Yang terparah apabila pelanggan membeli dari pesaing dan kemudian pindah menjadi pelanggan pesaing, artinya perusahaan kehilangan konsumen.

### 2.2.3 *Economic Order Quantity*

*Economic Order Quantity* (EOQ) atau *Economic Lot Size* (ELS) merupakan suatu metode manajemen persediaan paling terkenal dan paling tua. Diperkenalkan oleh FW. Harris sejak tahun 1914. Model ini dapat dipergunakan baik untuk persediaan yang dibeli maupun yang dibuat sendiri, dan banyak digunakan sampai saat ini karena penggunaannya relatif mudah. Model ini mampu untuk menjawab pertanyaan tentang kapan pemesanan/pembelian harus dilakukan dan berapa banyak jumlah yang harus dipesan agar biaya total (penjumlahan antara biaya pemesanan dengan biaya penyimpanan) menjadi minimum.

Pengertian EOQ sebenarnya merupakan volume atau jumlah pembelian yang paling ekonomis untuk di laksanakan pada setiap kali pembelian. Sehingga dengan menerapkan model EOQ dalam pembelian, biaya pemesanan dan biaya penyimpanan dapat ditekan (Tri Muhamad Hani, 2011). EOQ dapat di tentukan dengan rumus sebagai berikut :

$$EOQ = \sqrt{(2D \times S) / H}$$

Keterangan :

D : Kebutuhan barang per periode



Adalah jumlah barang yang di butuhkan dalam satu periode (hari, minggu, bulan).

S : Biaya order pesanan

Adalah biaya tranfortasi, telepon dan lain-lain yang di kieluarkan ketika hendak melakukan peyediaan kembali (*Re-Stock*).

H : Biaya penyimpanan

Adalah biaya yang di butuhkan untuk penyimpanan barang atau persediaan yang di butuhkan, Contoh : untuk menyimpan *Ice Cream* di butuhkan *Freezer* dalam kasus ini biaya untuk membayar listrik termasuk kedalam biaya penyimpanan.

#### 2.2.4 *Reorder Point*

Setelah jumlah bahan yang di beli dengan biaya yang minimal di tentukan, masalah selanjutnya adalah kapan perusahaan harus memesan kembali agar perusahaan tidak kehabisan bahan atau persediaan. Titik dimana perusahaan harus memesan kembali agar kedatangan bahan baku yang di pesan tepat pada saat persediaan bahan diatas *safety stock* sama dengan nol disebut *Reorder Point*.

*Reorder Point* menjelaskan kapan saat seharusnya diadakan pemesanan lagi, sehingga penerimaan bahan yang dipesan tepat pada waktunya. Ada dua faktor yang menentukan *Reorder Point*, yaitu :

1. Penggunaan bahan selama *Lead Time*

Lead Time adalah masa tunggu sejak pesanan barang atau bahan dilakukan sampai bahan tersebut tiba di perusahaan.

2. Safety Stock adalah persediaan minimal yang ada dalam perusahaan.

Untuk menghitung *Reorder Point*, dirumuskan sebagai berikut :

$$\text{Rumus ROP} = \text{Safety stock} + (\text{Penggunaan per Tahun} / 365) \times \text{Lead Time}$$

Jika permintaan atas produk tidak diketahui dengan pasti, maka terdapat kemungkinan terjadinya kehabisan persediaan. Untuk mengatasi hal tersebut pihak gudang sering sekali memilih untuk menyimpan persediaan pengaman (*Safety Stock*), (Haposan Naingolan, 2010).

### **2.2.5 Rancang**

Perancangan merupakan salah satu hal yang penting dalam membuat program adapun tujuan dari perancangan adalah, untuk memberikan gambaran yang jelas dan lengkap kepada pemrogram dan ahli teknik yang terlibat. Perancangan harus berguna dan mudah di pahami sehingga mudah di gunakan.

Perancangan adalah sebuah proses untuk menderfinisikan sesuatu yang akan di kerjakan dengan menggunakan teknik yang bervariasi serta di dalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan di alami dalam proses pengerjaan-nya.

Peracangan atau ancag merupakan serangkaian prosedur untuk menterjemahkan hasil analisa dari sebuah *system* ke dalam bahasa pemograman untuk endeskripsikan dengan detail bagaimana komponen-komponen *system* di implementasikan (R. S. Presman, 2010)

### **2.2.6 Bangun**

Pengertian Pembangunan atau bangun *system* adalah kegiatan menciptakan *sitem* baru maupun mengganti atau memperbaiki system yang telah ada secara keseluruhan (R. S. Presman, 2010)

Jadi dapat di simpulkan bahwa rancang bangun adalah penggambaran, perencanaan atau pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Dengan demiki pengertian

rancang bangun merupakan kegiatan menrjemahkan hasil analisa ke dalam betuk paket perangkat lunak kemudian menciptakan *system* tersebut atau memperbaiki *system* yang sudah ada sebelumnya.

### 2.2.7 Aplikasi

Aplikasi adalah program siap pakai yang dapat di gunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan pembuatan aplikasi tersebut, aplikasi adalah program komputer yang di tulis dalam suatu bahasa pemograman dan di pergunakan untuk menyelesaikan masalah tertentu (Robi Muhamad, 2009).

### 2.2.8 Website

Secara *basic*, *website* di gunakan untuk publikasi informasi. Adapun informasi yang akan di sediakan adalah beraneka ragam dari *profil* pribadi hingga *company profil*. *Website* sering juga di sebut *web*, dapat diartikan suatu kumpulan halaman-halaman yang menampilkan berbagai macam informasi teks, data gambar diam ataupun bergerak, data dinamis, suara maupun gabungan dari semuanya, baik itu yang bersifat statis maupun dinamis yang dimana membentuk suatu rangkaian bangunan yang saling berkaitan dimana masing-masing dihubungkan dengan jaringan halaman atau *hyperlink* (Nurhadi, 2017).

Definisi lainnya dari *website* adalah kumpulan dari berbagai macam halaman situs, yang terangkum dari sebuah *domain* atau juga *subdomain* yang lebih tepatnya berada di dalam WWW (*World Wide Web*) yang tentunya terdapat di dalam internet.

Halaman website bisanya berupa dokumen yang ditulis dalam format *Hyper Text Markup Language* (HTML), yang bisa di akses menggunakan HTTP, HTTP adalah suatu protokol yang menyampaikan berbagai inforamsi dari *server website*

untuk ditampilkan kepada para user atau pemakai melalui *web browser*. Pada dasarnya *website* dibagi menjadi dua bagian, yaitu :

➤ *Website Statis*

Merupakan *web* yang halamannya tidak berubah, biasanya untuk melakukan perubahan dilakukan secara manual dengan mengubah kode. *Website* statis informasinya merupakan informasi satu arah, yakni hanya berasal dari pemilik *software*nya saja, hanya bisa di *update* oleh pemiliknya saja, contoh dari pengertian *website* statis ini adalah *profil* perusahaan.

➤ *Website Dinamis*

Merupakan *web* yang halamannya selalu *update*, biasanya terdapat halaman *backend* (halaman administrator) yang digunakan untuk menambah atau merubah konten. *Web* dinamis membutuhkan *database* untuk menyimpan data. *Web* dinamis mempunyai arus informasi dua arah, yakni berasal dari pengguna dan pemilik, sehingga *update*-an dapat dilakukan oleh pengguna dan pemilik *website*. Contoh dari pengertian *website* dinamis ini yaitu *friendster*, *Multiplay*, *Facebook* dan lain-lain.

### 2.2.9 *My Structured Query Language (MySQL)*

MySQL adalah Sebuah program *database server* yang mampu menerima dan mengirimkan datanya sangat cepat, *multi user* serta menggunakan perintah dasar SQL (*Structured Query Language*). MySQL merupakan dua bentuk lisensi, yaitu *FreeSoftware* dan *Shareware*. MySQL yang biasa kita gunakan adalah MySQL *FreeSoftware* yang berada dibawah Lisensi GNU/GPL (*General Public License*), (Haris Spuro, 2012).

MySQL Merupakan sebuah *database server* yang free, artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya. MySQL pertama kali dirintis oleh seorang *programmer*

*database* bernama Michael Widenius, Selain *database server*, MySQL juga merupakan program yang dapat mengakses suatu *database* MySQL yang berposisi sebagai *Server*, yang berarti program kita berposisi sebagai *Client*. Jadi MySQL adalah sebuah *database* yang dapat digunakan sebagai *Client* maupun *server*.

*Database* MySQL merupakan suatu perangkat lunak *database* yang berbentuk *database relasional* atau disebut *Relational Database Management System* (RDBMS) yang menggunakan suatu bahasa permintaan yang bernama SQL (*Structured Query Language* ).

### **I. Kelebihan MySQL**

*Database* MySQL memiliki beberapa kelebihan dibanding *database* lain, diantaranya :

- MySQL merupakan *Database Management System* ( DBMS )
- MySQL sebagai *Relation Database Management System* (RDBMS) atau disebut dengan *database Relational*
- MySQL Merupakan sebuah *database server* yang *free*, artinya kita bebas menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya
- MySQL merupakan sebuah *database client*
- MySQL mampu menerima *query* yang bertumpuk dalam satu permintaan atau *Multi Threading*.
- MySQL merupakan *Database* yang mampu menyimpan data berkapasitas sangat besar hingga berukuran *Giga Byte* sekalipun.
- MySQL di dukung oleh *driver* ODBC, artinya *database* MySQL dapat diakses menggunakan aplikasi apa saja termasuk berupa visual seperti *visual Basic* dan *Delphi*.
- MySQL adalah *database* menggunakan *enkripsi password*, jadi *database* ini cukup aman karena memiliki *password* untuk mengakses

- MySQL merupakan *Database Server* yang *multi user*, artinya database ini tidak hanya digunakan oleh satu pihak orang akan tetapi dapat digunakan oleh banyak pengguna.
- MySQL mendukung *field* yang dijadikan sebagai kunci *premier* dan kunci unqi (*Unique*).
- MySQL memiliki kecepatan dalam pembuatan table maupun *update-an* table.

#### 2.2.10 *Hypertexy Preprocessor (PHP)*

Bahasa pemograman PHP merupakan bahasa pemograman yang cukup populer dan banyak di gunakan oleh para *programmer* di dunia. PHP atau yang memiliki kepanjangan *Hypertext Preprocessor*, merupakan suatu bahasa pemograman yang di fungsikan untuk membangun suatu website dinamis, PHP menyatu degan kode HTML. HTML digunakan seagai pembangun atau pondasi dari kerangka *layout web*, sedangkan PHP difungsikan sebagai prosesnya, sehingga dengan adanya PHP tersebut sebuah web akan mudah di *maintenance*, (Agus Saputra, 2013)

PHP berjalan pada sisi *server* sehingga PHP disebut juga sebagai bahasa *Sever Side Scripting*, artinya bahwa dalam setiap/unuk menjalankan PHP, akan mebutuhka web *server* untuk menjalankannya. PHP ini besifat *open source* sehingga dapat di pakai secara gratis, dan mampu lintas *platform*, yaitu dapat berjalan pada sistem operasi *widow* maupun *linux*.

#### 2.2.11 *Unified Modeling Language (UML)*

Menurut (Ivar Jacobson, 2010). *Unified Modeling Language (UML)* adalah bahasa standar untuk menulis cetak biru perangkat lunak. UML dapat digunakan

untuk memvisualisasikan, menentukan, membuat, dan mendokumentasikan artefak dari sistem intensif perangkat lunak. UML hanya bahasa dan hanya satu bagian dari metode pengembangan perangkat lunak.

Kosakata UML mencakup tiga jenis blok bangunan:

### 1. *Things*

Ada empat macam *thing* dalam UML:

#### a. *Structural things*

*Structural things* adalah kata benda dari model UML

#### b. *Behavioral things*

*Behavioral things* bagian dinamis dari model UML. Ini adalah kata kerja dari model, mewakili perilaku dari waktu ke waktu dan ruang.

#### c. *Grouping things*

*Grouping things* adalah bagian organisasi dari model UML. Ini adalah kotak di mana model dapat diuraikan.

#### d. *Annotational things*

*Annotational things* adalah bagian penjelasan dari model UML. Ini adalah komentar yang dapat Anda terapkan untuk menggambarkan, menerangi, dan berkomentar tentang elemen apa pun dalam model. *Relationships*

Ada empat macam *relationships* dalam UML:

#### a. *Dependency*

*Dependency* adalah hubungan semantik antara dua hal di mana perubahan ke satu hal (hal yang independen) dapat mempengaruhi semantik dari hal lain (hal yang tergantung).

#### b. *Association*

*Association* adalah hubungan struktural yang menggambarkan sekumpulan tautan, tautan yang menjadi koneksi antar objek.

#### c. *Generalization*

*Generalization* adalah hubungan spesialisasi / generalisasi di mana objek-objek elemen khusus (anak) dapat disubstitusikan untuk objek-objek elemen generalisasi (induk).

d. *Realization*

*Realization* adalah hubungan semantik antara pengklasifikasi, di mana satu pengklasifikasi menentukan kontrak yang dijamin untuk diklasifikasi oleh pengklasifikasi lain

2. *Diagrams*

*Diagram* adalah presentasi grafis dari sekumpulan elemen, paling sering disajikan sebagai grafik terhubung dari simpul (benda) dan busur (hubungan). Ada 3 *diagram* UML yang di gunakan:

a. *Class diagram*

*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

*Class* memiliki tiga area pokok :

- 1) Nama (dan *stereo type*)
- 2) Atribut
- 3) Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :





- *Private*, tidak dapat dipanggil dari luar class yang bersangkutan
- *Protected*, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya
- *Public*, dapat dipanggil oleh siapa saja



b. *Use case diagram*

*Use Case Diagram* adalah gambaran dari beberapa atau semua *actor*, *use case*, dan interaksi diantaranya yang memperkenalkan suatu sistem. Element - elemen pada *Use Case Diagram*:

Tabel 2.1 Element-elemnt pada *Use Case Diagram*

Elemen	Keterangan
	<i>Actor</i> : Mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem.
	<i>Use Case</i> : Adalah gambaran fungsionalitas dari suatu sistem, sehingga <i>customer</i> atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.
	<i>Association</i> : Menghubungkan link antar element.
	<i>&lt;&lt;Include&gt;&gt;</i> :Yaitu kelakuan yang harus terpenuhi agar sebuah <i>event</i> dapat terjadi, dimana pada kondisi ini sebuah <i>use case</i> adalah bagian dari <i>use case</i> lainnya

c. *Activity diagram*

*Activity Diagram* adalah diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Komponen yang ada pada activity diagram antara lain:

- *Activity* atau *state*: Menunjukan aktivitas yang dilakukan.

- *Initial activity* atau *initial state*: Menunjukkan awal aktivitas dimulai.
- *Final Activity* atau *final state*: Menunjukkan bagian akhir dari aktivitas.
- *Decission*: Digunakan untuk menggambarkan test kondisi untuk memastikan bahwa control flow atau object flow mengalir lebih ke satu jalur. Jumlah jalur sesuai yang diinginkan.
- *Merge*: Berfungsi menggabungkan flow yang dipecah oleh decission.
- *Synchronization*: dibagi menjadi 2 yaitu fork dan join. Fork digunakan untuk memecah behaviour menjadi activity atau action yang paralel, sedangkan join untuk menggabungkan kembali activity atau action yang paralel.
- *Swimlanes*: Memecah activity diagram menjadi baris dan kolom untuk membagi tanggung jawab obyek-obyek yang melakukan aktivitas.
- *Transition*: Menunjukkan aktivitas selanjutnya setelah aktivitas sebelumnya.

### 2.2.12 *Model Driven Developement (MDD)*

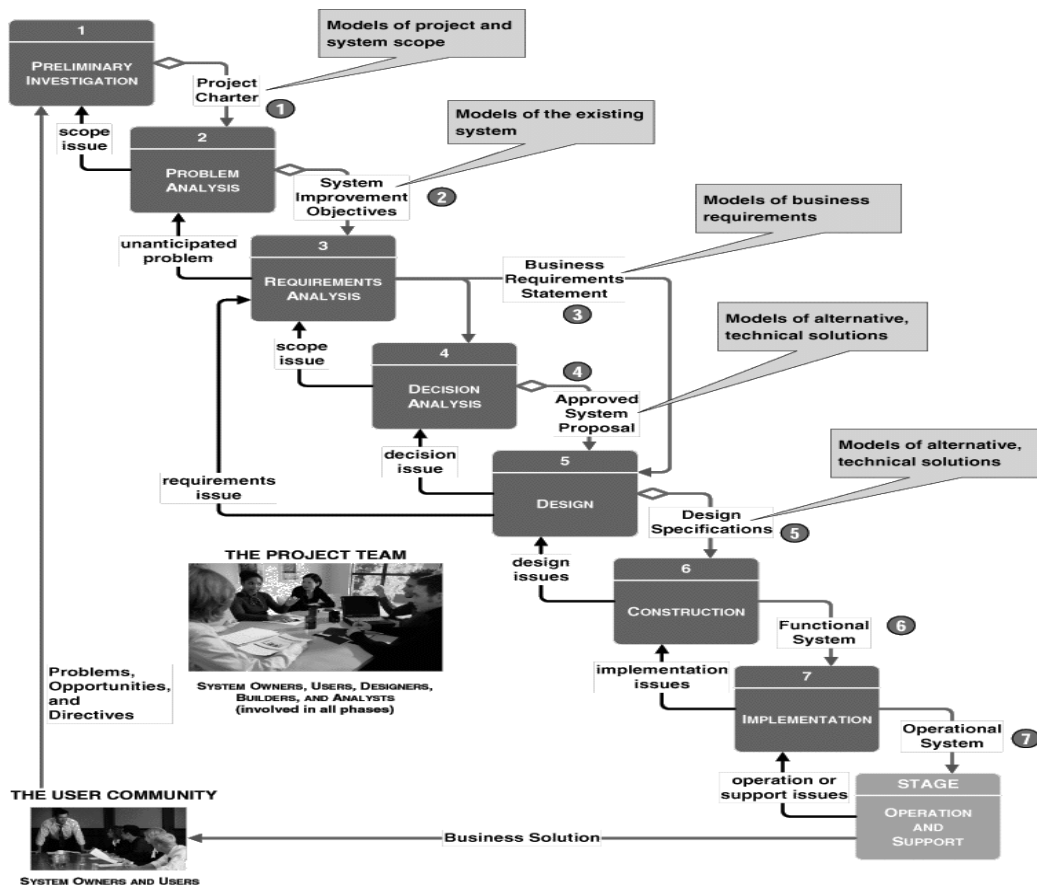
*Model Driven Development (MDD)* merupakan suatu teknik yang menekankan penggambaran model untuk memvisualisasikan dan menganalisis masalah, mendefinisikan proses bisnis, dan merancang sistem informasi (Wijaya, 2009). Ada beberapa pendekatan dalam teknik pemodelan:

- 1) *Structured Analysis Design* → berorientasi proses
- 2) *Information engineering (IE)* → berorientasi data

- 3) *Object-oriented analysis and design* (OOAD)→menggaungkan orientasi proses dan data ke dalam bentuk objek.

Keunggulan model-driven:

1. Spesifikasi persyaratan lebih menyeluruh dan didokumentasikan dengan baik
2. Persyaratan bisnis dan desain sistem lebih mudah divalidasi dengan gambar daripada dengan kata-kata
3. Lebih mudah mengidentifikasi, mengkonseptualkan, dan menganalisis solusi-solusi teknis alternatif
4. Spesifikasi desain cenderung solid, stabil, dapat beradaptasi, dan fleksibel karena berbasis model dan dianalisis lebih menyeluruh sebelum dibangun
5. Sistem dapat dikonstruksikan dengan lebih tepat pertama kali saat dibangun.



Gambar 2.1 Model Driven Development

### **2.2.13 Persamaan Dengan Riset Lain**

Adapun Persamaan penelitian ini dengan penelitian nomor 2 dan 3 adalah sama sama menggunakan metode EOQ dan ROP untuk metode pengendalian persediaan, dan persamaan dengan penelitian nomor 1 adalah sama-sama menggunakan metode EOQ.

### **2.2.14 Perbedaan Dengan Riset Lain**

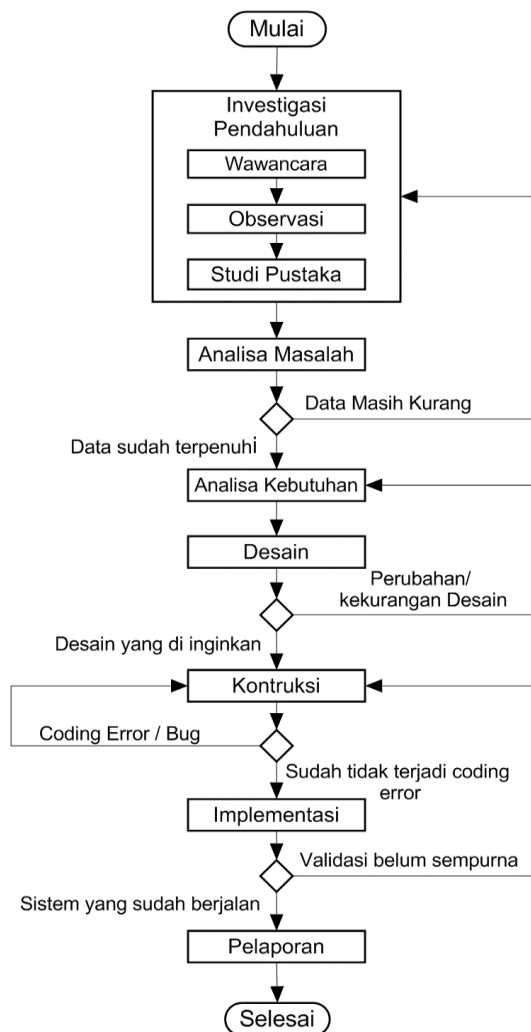
Perbedaan penelitian ini dengan yang telah di lakukan sebelumnya adalah, pada penelitian ini penyusun menggunakan bahasa pemograman PHP dan juga terdapat perbedaan dari jenis data yang di kelola.

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Kerangka Berfikir

Kerangka berfikir pada perancangan aplikasi pengendalian *inventory* sebagai pedoman penulis untuk mencapai tujuan dari penelitian yang telah ditentukan sebelumnya adalah sebagai berikut.



Gambar 3.2 Model MDD (Model Driven Development)

## 3.2 Deskripsi

### 3.2.1 Investigasi Pendahuluan

*Preliminary Investigation* (Pendahuluan Investigasi) adalah langkah pertama dalam melakukan penelitian dengan menginisialisasi masalah, fakta-fakta dan bukti tentang penelitian yang dilakukan. Dimulai dengan mengunjungi Grosir Sembako Pak Asep yang berlokasi di Malaksari RT 03 / RW 10, Desa Patrolsari Kecamatan Arjsari Kabupaten Bandung, lalu menemui bapak Asep Suryadi untuk meminta izin melakukan penelitian dan wawancara perihal Pengelolaan data barang pada grosir yang di maksud.

Adapun pertanyaan yang diajukan oleh penyusun adalah sebagai berikut :

- 1) Apakah pencatatan nota barang belanja masih manual?

Jawaban :

Ya masih manual, jadi ketika akan melakukan penyediaan kembali (*Restock*) *Owenr* Grosir mengecek satu persatu barang yang akan di sediakan kembali.

- 2) Apakah pengumpulan nota belanja sudah tersusun rapih?

Jawaban :

Belum, karena ketika Owner Grosir lupa menyimpan nota belanja, Ower Grosir tidak dapat mengetahui perbandingan harga dari hari sebelumnya (apakah barang yang di beli harganya naik atau turun).

- 3) Bagaimana cara Owner Grosir mencari data harga barang yang akan di jual sebelum di buatkan aplikasi untuk membantu pencarian?

Jawaban :

Owner Grosir melihat dari nota belanja yang di tulis secara manual.

- 4) Apa permasalahan yang saat ini ada pada pengolahan data barang di Grosir Sembako?

Jawaban :

Permasalahan yang ada saat ini adalah : ketika lupa menyimpan nota belanja *Owner* Grosir tidak dapat mengetahui perbandingan harga /

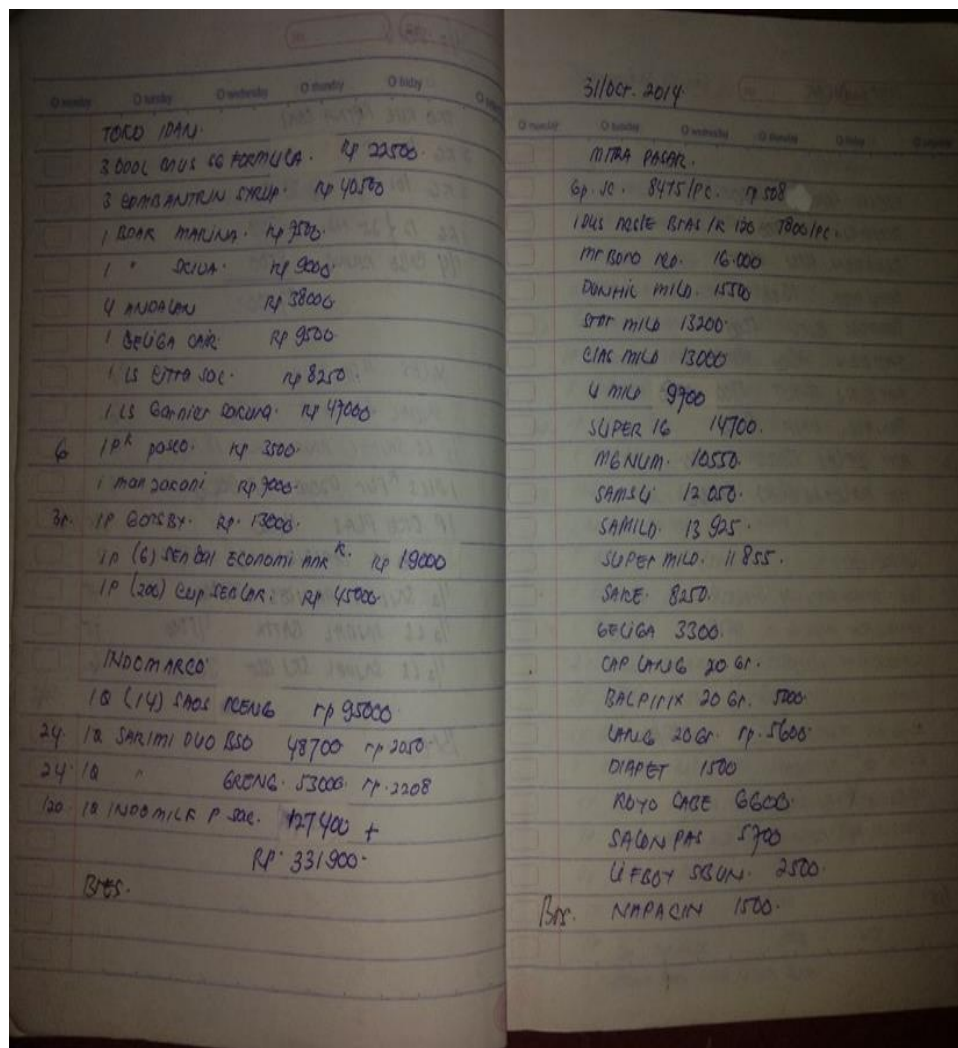
harga jual, ketika lupa mencatat barang barang yang seharusnya di sediakan kembali Grosir sembako tidak mampu memenuhi kebutuhan konumen dan kurangnya efisien-nya penyimpanan nota belanja.

- 5) Bagaimana caranya mengetahui pemasukan / Keuntungan yang di dapatkan pada Grosir Sembako?

Jawaban :

Cara mengetahuinya adalah dengan cara menghitung uang yang ada di laci Grosir saat grosir telah tutup.

Gambar Nota Belanja Grosir Sembako :



Gambar 3.3 Buku Nota Belanja Grosir Sembako

✓ Daun Kacang 12	80.000.
✓ Buryan 12. ✓	105.000.
✓ Susu Keleng pr	95.000.
✓ Bori 12.	55.000.
✓ Bais 5000 pr	98.000.
✗ <del>Bais 1000 pr</del>	85.000.
✓ Kero 12	140.000.
✓ Susu Saret pr	70.000.
✓ Susu Saret Ca	170.000.
✓ Indom AB	82.000.
✓ Sedap AB	82.000.
✓ Sedap Sata	40.000.
✗ <del>Bumbu Apa</del>	20.000.
✓ Adem Sari	20.000.
✓ Badrek EXTRA	785.000.
✓ Badrek Migr	217.000.
✓ Super 5p	324.000.
✓ Samil 1p.	017
✓ Garfit 1p.	2.468.000.

Gambar 3.4 Nota Belanja Grosir Sembako

### 3.2.2 Analisa Masalah

Berikut adalah hasil analisis masalah yang ada pada Grosir Sembako :

- 1) Pencatatan nota barang belanja masih manual.



- 2) Ketika Owner Grosir lupa mencatat barang belanjaan, Grosir Sembako tidak dapat memenuhi kebutuhan konsumen.
- 3) Penyimpanan nota belanja masih manual (Di tumpuk di satu tempat).

Berdasarkan masalah yang ada pada Grosir Sembako, pada tahap ini penulis mencoba membuat suatu aplikasi pengendalian *Inventory* menggunakan metode *Economic Order Quantity* (EOQ) untuk menentukan kuantitas barang yang akan di order di kemudian hari dan *Re-Order Point* (ROP) untuk menentukan kapan harus di lakukan-nya penyediaan ulang.

Jika di dalam tahap ini masih ada kekurangan maka akan kembali ke tahap investigasi awal sampai mendapat hasil yang tepat, kemudian di lanjutkan ketahap berikutnya.

### 3.2.3 Analisis Kebutuhan

Adapun analisis kebutuhan yang diperlukan untuk menjawab pertanyaan dan permasalahan yang ada terdiri dari 4, yaitu:

#### 1. Kebutuhan Prosedur

Prosedur yang dibutuhkan yaitu mengumpulkan data-data yang akan digunakan. Berikut data-data yang dibutuhkan dalam perancangan aplikasi *Inventory* pada Grosir Sembako :

- a. Data jenis barang
- b. Data nama barang
- c. Data satuan barang
- d. Data harga barang

#### 2. Aplikasi Utama

- a. Membuat aplikasi pengendalian *Inventory* pada Grosir Sembako.
- b. Pada aplikasi yang dibuat akan menerapkan metode EOQ dan ROP agar pengendalian *Inventory* dapat berjalan dengan baik.

### 3. Kebutuhan Perangkat Lunak

Adapun perangkat lunak (*Software*) yang dibutuhkan pada penelitian ini adalah sebagai berikut:

- a. MySQL, digunakan untuk menyimpan *database* dari setiap data yang dibutuhkan.
- b. PHP, sebagai bahasa pemrograman yang akan digunakan untuk membuat aplikasi.
- c. XAMPP, digunakan sebagai control panel.
- d. *Sublime Text*, digunakan untuk menjalankan PHP sebagai pengembangan aplikasi.
- e. *Star UML*, digunakan untuk membuat diagram-diagram UML.
- f. *Microsoft Word*, digunakan untuk membuat laporan penelitian.

### 4. Kebutuhan Perangkat Keras

Adapun kebutuhan minimum perangkat keras (*hardware*) dalam pembuatan aplikasi dapat di lihat pada tabel 3.2

*Tabel 3.2 Spesifikasi Minimum Hardware*

Processor	Intel Celeron 1.80 GHz (2CPUs)
RAM	2 GB
Memory	320 GB

Untuk kenyamanan pemakaian dapat menggunakan computer dengan Spesifikasi di atasnya.

#### 3.2.4 Desain

Pada tahap ini penyusun membuat desain aplikasi Penendalian *Inventory* yang akan dibangun dari hasil analisis yang telah dilakukan. Ada beberapa proses tahapan yang akan dilakukan diantaranya sebagai berikut :

## 1. Perancangan Model

Pada perancangan model dilakukan untuk mengetahui alur yang akan dibuat untuk membangun aplikasi pengelolaan *inventory*. Perancangan akan dibuat dengan *Unified Modeling Language* (UML) agar dengan mudah dalam proses pengembangan dan visualisasinya. Diagram UML yang digunakan antara lain:

- a. *Use Case Diagram* digunakan untuk menggambarkan keterhubungan *actor* dan *use case* dalam aplikasi yang akan dibuat. *Actor* dan *use case* yang terlibat pada penelitian ini adalah:
  - Actor : *Administrator, Owner Grosir*.
  - Use case : Pembelian, penjualan dan penyediaan kembali.
- b. *Class Diagram* digunakan untuk menggambarkan atribut, *operation* dan juga *constraint* pada sistem yang akan dibuat. *Class diagram* yang akan dibuat adalah sebagai berikut:
  - 1). Class diagram *owner grosir*
  - 2). Class diagram *administrator*
  - 3). Class diagram transaksi
  - 4). Class diagram data barang
  - 5). Class diagram perencanaan penyediaan
- c. *Activity Diagram* digunakan untuk memodelkan alur kerja dari sistem dan aktivitas dari *actor* dalam aplikasi. Adapun diagram alur yang akan dibuat diantaranya:
  - 1). Proses login
  - 2). Proses input data
  - 3). Proses simpan data
  - 4). Proses hapus data
  - 5). Proses edit data
  - 6). Proses pencarian data barang
  - 7). Proses mencetak data

## 2. *Database*

*Database* adalah salah satu bagian yang paling penting dalam membangun aplikasi pengendalian *inventory*. Pengelolaan data juga menjadi bagian paling utama dalam penelitian ini. Dalam perancangan *database*, data yang dibutuhkan didapatkan dari pengumpulan data sebelumnya. Beberapa data yang digunakan dalam penelitian ini adalah:

- a. Data jenis barang
- b. Data nama barang
- c. Data satuan barang
- d. Data jumlah barang
- e. Data harga barang
- f. Data rencana penyediaan barang

## 3. Perancangan Antar Muka (*User Interface*)

Perancangan antar muka adalah perancangan tahap akhir yang akan dibuat. Adapun perancangan antar muka yang akan dibuat sebagai berikut:

- a. Form login
- b. Form halaman menu utama
- c. Form halaman jenis barang
- d. Form halaman barang terjual
- e. Form halaman rencana penyediaan barang

### 3.2.5 Kontruksi

Setelah tahap *Design* (Perancangan) selesai maka tahap selanjutnya adalah pembuatan aplikasi. Pada tahap pembuatan aplikasi digunakan perangkat lunak dan bahasa pemrograman sebagai berikut:

1. *Sublime Text*, digunakan untuk membuat program interface aplikasi.
2. *Xampp* sebagai server web *apache* untuk pengembangan website.
3. *Database MYSQL*, digunakan untuk menyimpan *database* dari data barang yang ada.

### **3.2.6 Impementasi**

Setelah aplikasi dibuat pada tahap sebelumnya, tahapan selanjutnya yaitu membuat pengujian terhadap aplikasi yang sudah dibuat, yaitu dilakukan pengujian dengan menggunakan metode *black box* yaitu untuk menguji fungsionalitas dari suatu aplikasi.

Jika di dalam tahap ini masih ada kekurangan maka akan kembali ke tahap investigasi awal sampai mendapat hasil yang tepat, kemudian di lanjutkan ketahap berikutnya.

### **3.2.7 Pembuatan Laporan**

Tahap ini dibuat laporan untuk mempertanggung jawabkan hasil dari penelitian yang telah dilakukan di dibuat dituangkan ke dalam bentuk skripsi yang nantinya akan di uji dalam sidang skripsi sebagai syarat memperoleh gelar sarjana Strata 1 (S1) Teknik Informatika.Fakultas Teknologi informasi Universitas Bale Bandung (UNIBBA).

## **BAB IV**

### **ANALISIS DAN PERANCANGA SISTEM**

#### **4.1 Analisis**

##### **4.1.1 Analisis Masalah**

Langkah awal dalam pembuatan sitem adalah meng-identifikasi permasalahan yang ada sebagai dasar untuk mebuat sebuah solusi yang akan disajikan dalam bentuk sebuah aplikasi. Langkah identifikasi di lakukan dengan cara observasi dan wawancara ke grosir sembako Arjasari, seperti yang telah di sebutkan pada bab sebelumnya. Sehinga dapat di lakukan tindakan pengambilan solusi yang tepat utuk menyeesaikan permasalahan tersebut.

Meurut proses analisis yang dilakukan dalam operasional pengendalian persediaan, untuk menentukan jumlah produk yang akan di sediakan ulang, Pemilik Grosir melihat dari penjualan perhari, sedangkan untuk menentukan kapan dilakukan pemesanan (*restock*), Pemilik Grosir melihat rata-rata penjualan produk dari hari sebelumnya, jika jumlah produk mencapai jumlah rata-rata penjualan dan produk barang yang di sediakan hampir habis terjual maka akan dilakukan penyediaan kembali. Dengan proses pengendalian persediaan yang dilakukan seperti saat ini, Pemilik Grosir Sembako mengalami berbagai macam permasalahan. Seperti lupa mecatat barang yang harus di sediakan kembali sehingga menimbulkan kosongnya salah satu produk yang seharusnya di sediakan dan tidak mampu memenuhi kebutuhan konsumen. Oleh sebab itu maka penyusun berinisiatif mengimplementasi kan metode EOQ dan ROP pada aplikasi *inventory* yang aka di buat dengan tujuan meng-efisiensikan pemelik grosir dalam mengeola *invetonry*.

#### 4.1.2 Analisis Sistem

Berbeda dengan analisis data yang lebih pada cara untuk mengelola data menjadi sebuah informasi. Sedangkan analisis sistem ini lebih kepada penjabaran dari suatu sistem informasi yang telah dihasilkan. Analisis sistem adalah penjabaran dari suatu sistem informasi yang utuh ke dalam berbagai bagian komponennya dengan maksud agar bisa mengidentifikasi dan mengevaluasi berbagai macam masalah atau hambatan yang timbul pada sistem sehingga nantinya bisa dilakukan penanggulangan.

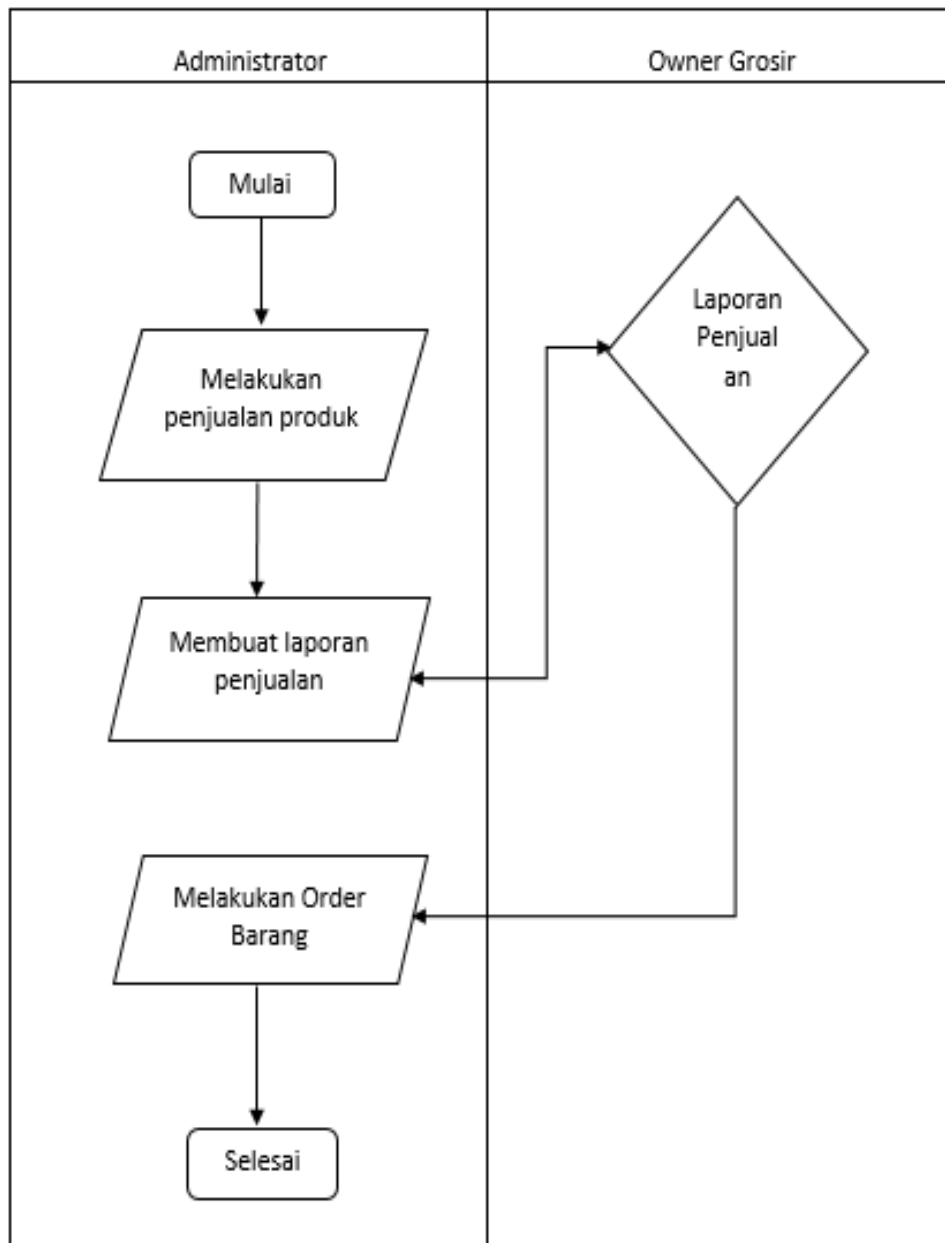
Lalu orang atau kelompok yang melakukan perbaikan atau perancangan suatu sistem di namakan dengan sistem analis. Sistem analis adalah orang atau kelompok yang melaksanakan pengembangan sistem. Sistem analis menekuni permasalahan ataupun kebutuhan dari suatu system dan system analis juga merupakan orang yang bertanggung jawab atas terjadinya proses analisa ataupun perancangan pada system informasi.

Berdasarkan uraian diatas maka dapat di simpulkan bahwa analisis sistem adalah tahap yang bertujuan untuk memahami sistem, mengetahui kekurangan sistem dan menentukan kebutuhan hasil proses pada perangkat lunak dan perangkat keras yang di gunakan. Dengan menganalisis prosedur sistem yang di gunakan dan melakukan pengujian hasil, maka sistem dapat di evaluasi sehingga dapat di jadikan acuan dalam proses pembentukan kesimpulan.

Pada perancangan aplikasi pengelolaan *inventory* ini karena menggunakan perangkat lunak dan perangkat keras untuk membuatnya maka dapat di analisa kebutuhan apa saja dari perangkat lunak dan perangkat keras yang di butuhkan untuk membuat aplikasi ini.

Pada poin ini juga penyusun berperan sebagai sistem analis yang mana penyusun akan melakukan analisis terhadap sistem yang sedang berjalan guna mengetahui hal apa saja yang di butuhkan ketika hendak membuat aplikasi, berikut adalah analisis sistem yang sedang berjalan dan sistem yang di usulkan oleh penyusun :

### 1. Analisis Sistem yang Berjalan

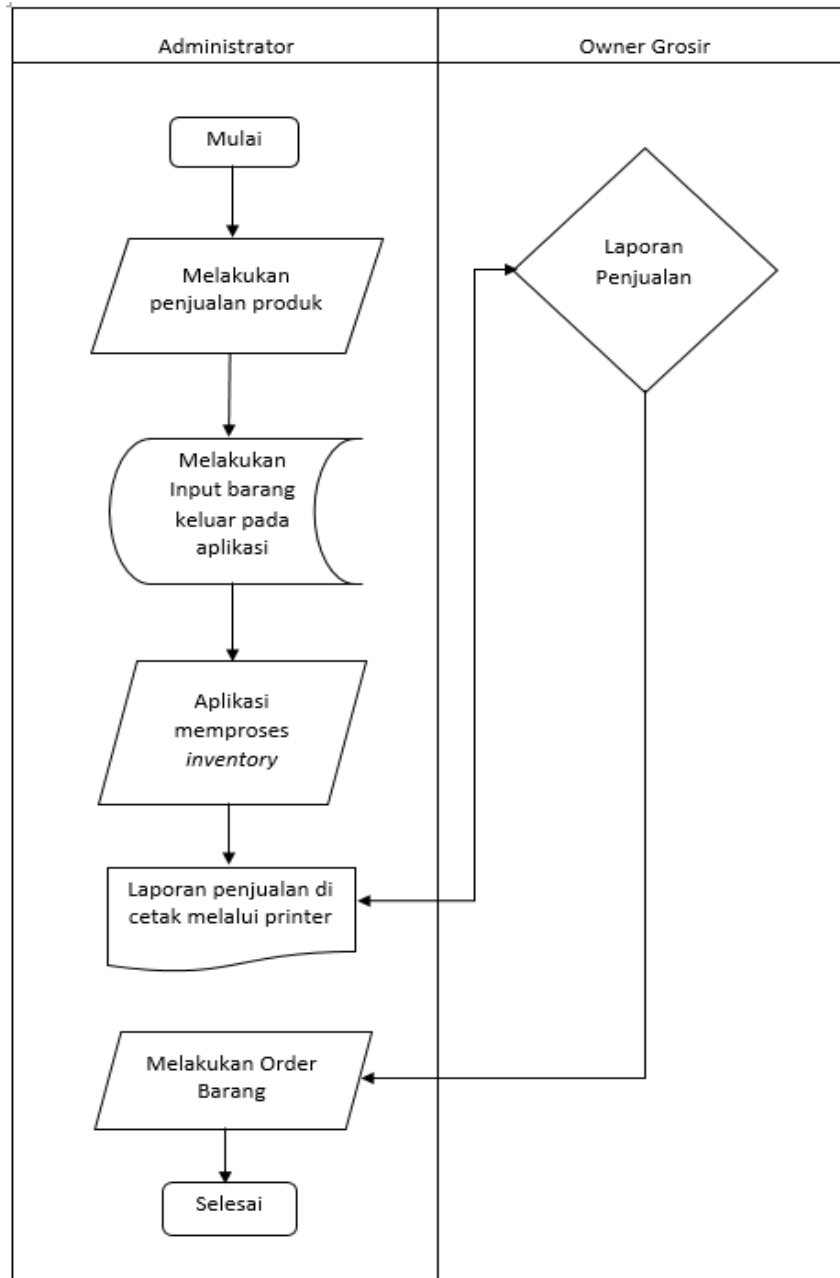


*Gambar 4.5 Flow Map Diagram Sistem yang sedang berjalan*

Pada gambar di atas dapat disimpulkan bahwa segala aktifitas dilakukan secara manual, tidak menggunakan *system* komputasi komputer.



## 2. Analisis Sistem yang di Usulkan



Gambar 4.6 Flow Map yang di Usulkan

Pada gambar di atas dapat disimpulkan bahwa *Admin* Tidak perlu lagi membuat laporan penjualan secara manual karena, laporan penjualan telah dibuat oleh aplikasi yang diusulkan.

### 4.1.3 Analisis Kebutuhan

Analisis Kebutuhan adalah sebuah proses untuk mendapatkan informasi, model dan spesifikasi tentang kebutuhan perangkat lunak dan perangkat keras yang di butuhkan dalam membuat aplikasi pengendalian *inventory* ini.

Analisis kebutuhan juga mencakup pekerjaan-pekerjaan penentuan kebutuhan atau kondisi yang harus di penuhi dalam suatu produk baru atau perubahan produk yang mempertimbangkan berbagai kebutuhan, dari hasil analisis ini harus dapat di laksanakan, diukur, diuji, terkait dengan kebutuhan bisnis yang teridentifikasi, serta di definisikan sampai tingkat detail yang memadai untuk detail sistem. Berikut adalah beberapa software dan hardware yang di butuhkan :

#### 1. Kebutuhan Sistem

*Tabel 4.3 Kebutuhan Sistem*

No	Perangkat Keras	Perangkat Lunak
1	Seperangkat PC / Laptop	<i>Windows 7 32 bit</i>
2	-	<i>Xampp</i>
3	-	<i>Sublime text</i>
4	-	<i>Mozilla Fire fox</i>
5	-	<i>Microsoft word 2013</i>
6	-	<i>Star UML</i>
7	-	<i>Balsamiq mockups 3 V.3.5.17</i>
8	-	<i>Framework CSS</i>

#### 2. Spesifikasi Perangkat Keras

*Tabel 4.4 Spesifikasi perangkat keras*

No	Spesifikasi	Komputer
1	<i>Processor</i>	Intel Celeron 1.80 GHz (2CPUs)
2	<i>Ram (Randaom Acces Memory)</i>	2 GB
3	<i>Hard Disk</i>	320 GB

### 3. Spesifikasi Perangkat Lunak

*Tabel 4.5 Spesifikasi perangkat lunak untuk implementasi*

No	Spesifikasi	Komputer
1	Sistem Operasi	Windows 7 32 bit
2	Bahasa Pemograman	Bahasa Pemograman PHP

#### 4.1.4 Analisis Pengguna

Penganalisaan pengguna adalah yang berkaitan dengan yang akan memakai aplikasi yang akan di buat, pengguna aplikasi ini adalah ;

- owner grosir
- Pegawai

agar aplikasi ini dapat di gunakan dengan baik dan mudah untuk di operasikan penyusun akan menyediakan berbagai menu pada *user interface* yang akan di buat.

#### 4.1.5 User Interface

*User interface* dari aplikasi sangat berpengaruh terhadap minat dari *user* dalam mengoperasikan-nya. Hal ini mencakup perangkat yang di gunakan sebagai piranti masukan dan keluaran dari aplikasi yang akan di buat. Karena aplikasi yang di buat di tujuan untuk mengefisiensi-kan pekerjaan pemilik grosir dalam mengelola *inventory*.

Untuk perangkat masukan yang di gunakan di antaranya : *Mouse*, agar *user* dapat leluasa menggerakkan petunjuk *mouse* untuk memilih menu atau sub menu yang ada, dan *keyboard* yang akan di gunakan sebagai sarana *input* data yang di perlukan dalam pengelolaan *inventory*.

#### 4.1.6 Fitur-Fitur

Fitur-fitur yang di gunakan dalam aplikasi ini di maksudkan agar *user* dapat dengan mudah mengoperasikan-nya, di tunjang dengan metode EOQ dan ROP untuk memberikan biaya paling ekonomis dalam pengelolaan *inventory*, bagian terpenting dari fitur-fitur yang ada adalah sebagai berikut :

- a. *Login user*
- b. *Stok alert*
- c. Penghitungan EOQ otomatis
- d. Penghitungan ROP otomatis
- e. Biaya-biaya dan leadtime bisa di *custom*
- f. *Dashboard* dan grafik sederhana
- g. *Calculate stok* (bukan *trigger*)

Dari hasil yang di dapat dari hal di atas panduan antara *user interface* dengan fitur aplikasi perlu di bentuk untuk menciptakan aplikasi yang dapat memberikan manfaat luas dan nyata bagi pengguna.

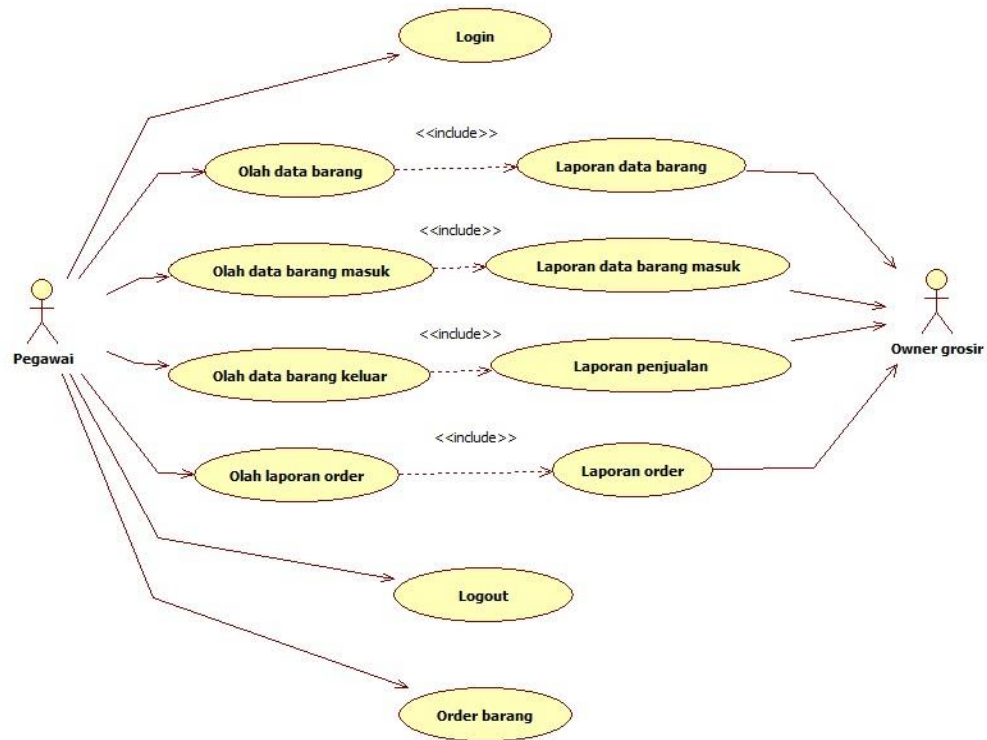
## 4.2 Perancangan Sistem

Perancangan sistem merupakan perancangan yang melibatkan sistem tertentu, dalam arti sistem yang dirancang adalah sistem yang memang ingin di buat dan di kendalikan.

Dengan perancangan, maka sistem dapat di arahkan atau di kendalikan sesuai dengan keinginan. Namun tidak sekedar perancangan yang di pilih melainkan sesuai dengan data yang di peroleh.

Pada tahap ini perangkat lunak dideskripsikan dengan model analisis menggunakan diagram *Use Case*. Analisis digunakan untuk pemetaan awal mengenai perilaku yang diisyaratkan sistem aplikasi kedalam elemen-elemen pemodelan. Untuk membantu perancangan dan melengkapi dokumentasi perancangan.

### 4.2.1 Usecase Diagram



Gambar 4.7 Use case diagram yang di usulkan

Diagram Use Case di atas menunjukkan Actor yang terlibat di dalam Aplikasi pengelolaan Inventory ada 2 Actor, yang masing-masing mempunyai hak sebagai berikut :

#### 1. Definisi Aktor

##### ➤ Pegawai

Pegawai adalah staf Grosir Sembako yang diberikan kewenangan untuk melakukan pengelolaan (*maintenance*) aplikasi. Pegawai dapat melihat dan mengubah semua data sistem, dan dapat membenahi kesalahan yang terjadi dalam sistem.

##### ➤ Owner grosir

Owner Grosir dapat melakukan berbagai hal yang di lakukan pegawai, akan tetapi segala hal yang bersangkutan dengan pengelolaan sistem di

serahkan kepada pegawai, sehingga Owner Grosir dapat menerima berbagai macam laporan dari pegawai, dan apabila Owner Grosir menyetujui laporan order barang yang telah di serahkan pegawai, pegawai dapat melakukan order barang.

## 2. Use case

*Tabel 4.6 Deskripsi Usecase*

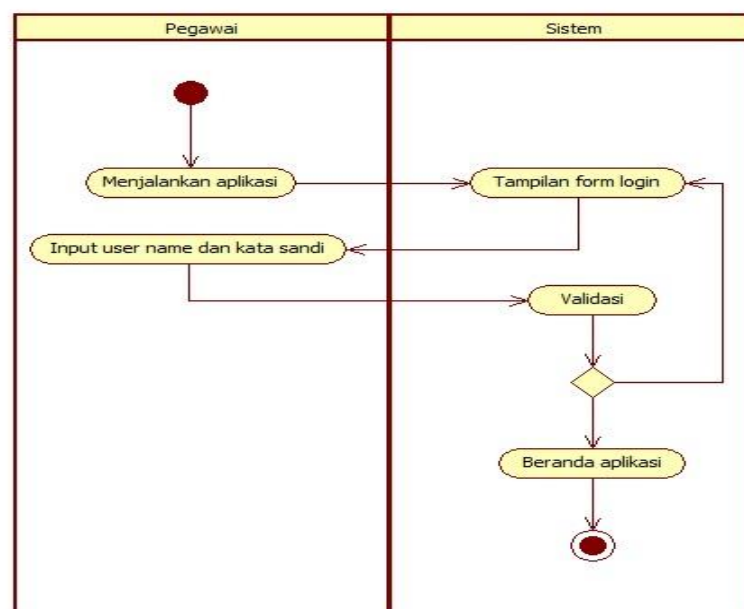
No	Aktor	Use case	Deskripsi
1	Pegawai	<i>Lo gin</i>	Melakukan <i>login</i> pada aplikasi pengelolaan <i>inventory</i>
		Olah data barang	Merupakan proses peng- <i>input</i> -an atau bisa juga penghapusan data
		Olah data barang masuk	Merupakan Proses peng- <i>input</i> -an data barang yang baru
		Olah data barang keluar	Merupakan proses peng- <i>input</i> -an data barang yang terjual
		Olah laporan order	Merupakan proses pengumpulan data barang yang akan di sediakan kembali
		<i>Logout</i>	Melakukan <i>logout</i> dari aplikasi pengendalian <i>inventory</i>
		Order barang	Pegawai malakukan proses penyediaan barang
2	Owner Grosir	Laporan data barang	Merupakan berkas / dokumen yang berisi data barang

		Laporan data barang masuk	Merupakan berkas / dokumen yang berisi data barang yang masuk
		Laporan data barang keluar	Merupakan berkas / dokumen yang berisi data barang yang terjual
		Laporan order	Merupakan berkas / dokumen yang berisi sekumpulan data barang yang akan di sediakan kembali

Lanjutan Tabel 4.7 Deskripsi *Usecase*

## 4.2.2 Activity Diagram

### A. Activity Diagram Log in

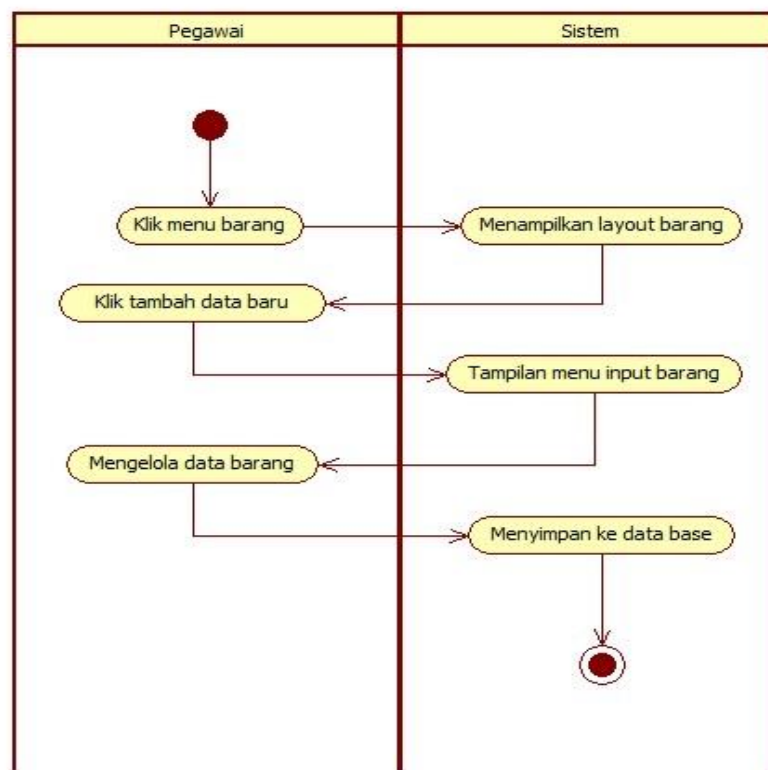


Gambar 4.8 Activity diagram login

Pada Gambar 4.2 diatas pegawai melakukan *login* dengan mengakses menu *login* kemudian masukan *username* dan *password* lalu

sistem akan mem-*verifikasi* jika *username* dan *password* salah maka sistem akan memberikan informasi tidak *valid* jika *username* dan *password* benar maka sistem akan menampilkan tampilan awal aplikasi dan pegawai dapat mengakses menu utama.

### B. Activity Diagram Olah Data Barang

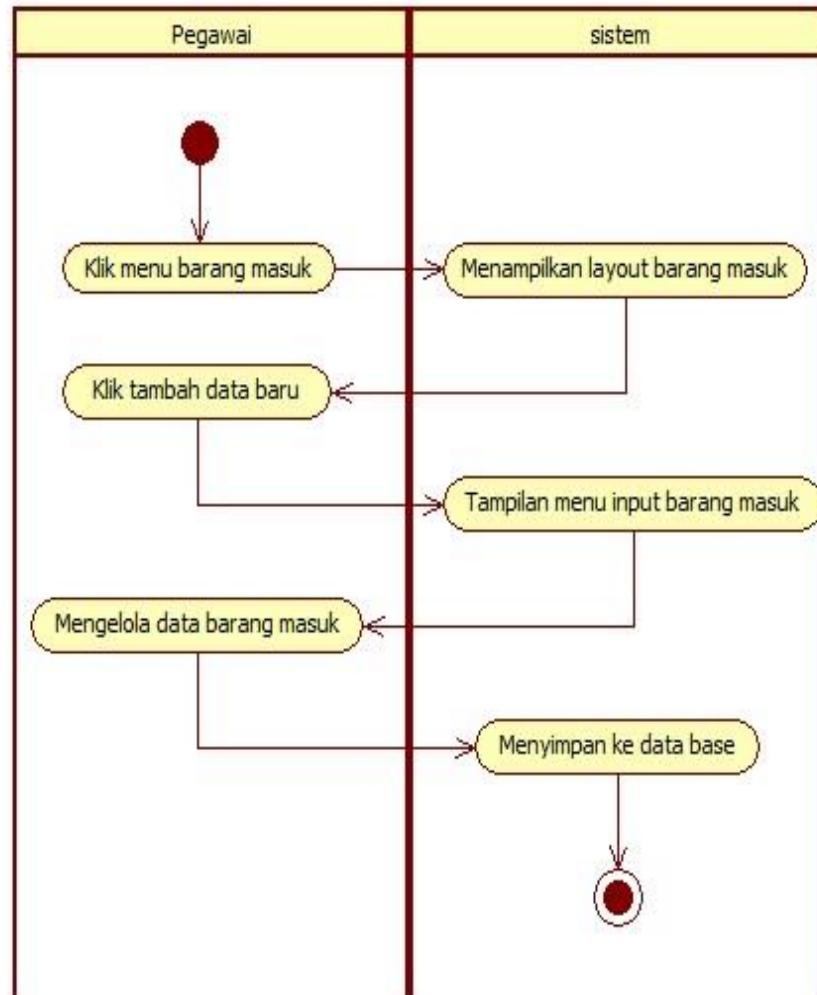


Gambar 4.9 Activity diagram olah data barang

Pada Gambar 4.3 Activity diagram olah data barang menggambarkan proses *input* data barang pada sistem, pegawai meng-klik menu barang lalu akan tampil *layout* barang kemudian pegawai klik tombol tambah data baru dan sistem menampilkan form *input* data barang, lalu pegawai mengelola data barang (bisa *input* atau mengubah data yang sudah ada) lalu menekan tombol *creat* maka data akan otomatis tersimpan ke *database*.



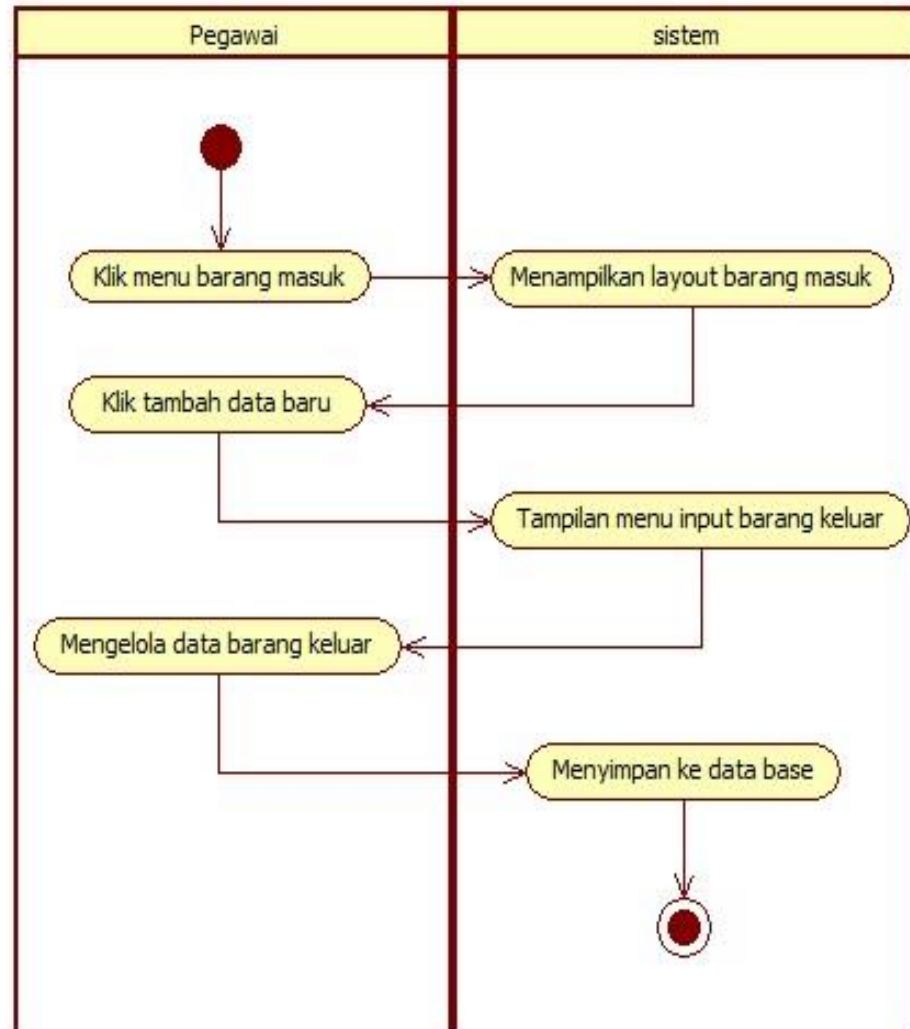
### C. Activity Diagram Olah Data Barang Masuk



Gambar 4.10 Activity diagram olah data barang masuk

Pada Gambar 4.4 Activity diagram olah data barang masuk menggambarkan proses *input* data barang yang baru ke dalam sistem, pertama pegawai meng-klik menu barang masuk lalu akan tampil *layout* barang, kemudian pegawai meng-klik menu tambah data baru dan sistem menampilkan form *input* data barang lalu pegawai mengelola data barang yang baru di order (bisa *input* atau mengubah data yang sudah ada) lalu menekan tombol *save* maka data akan otomatis tersimpan ke *database*.

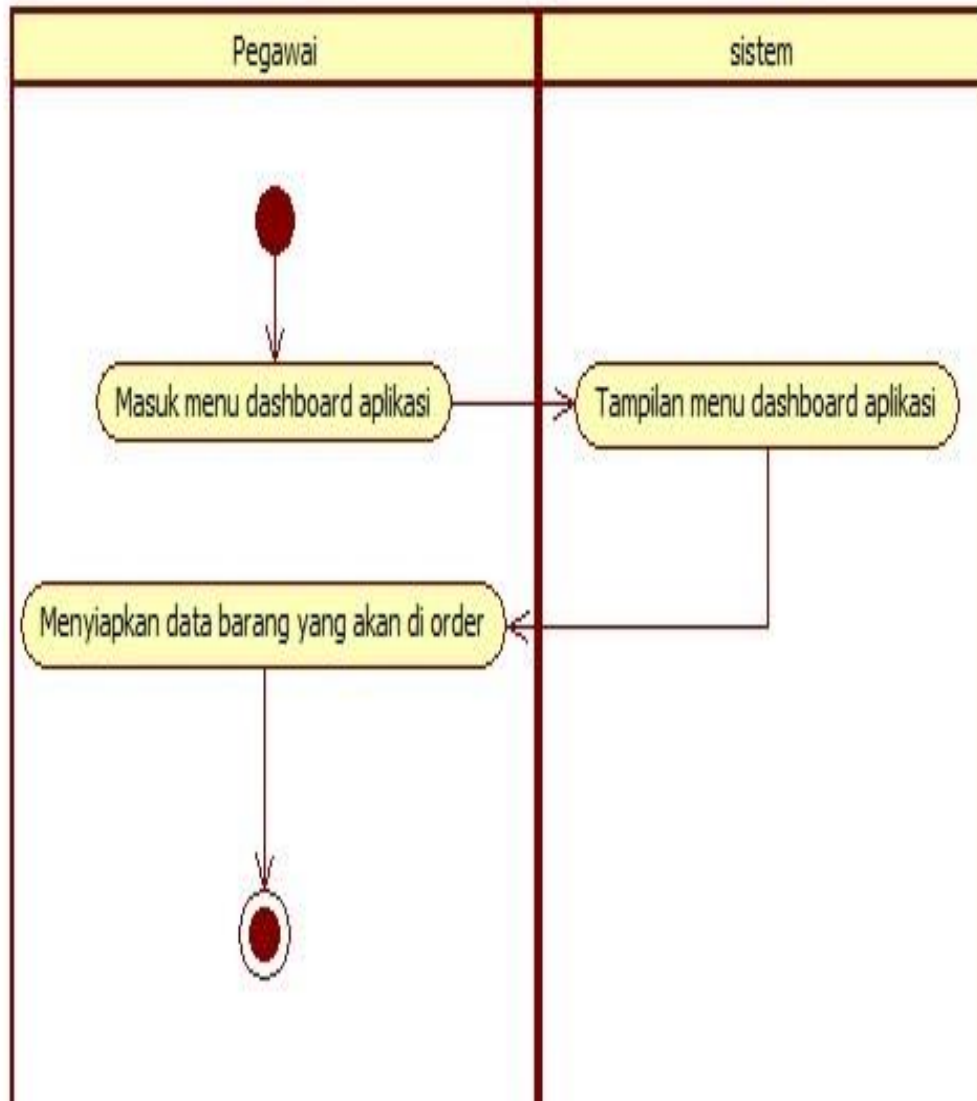
#### D. Activity Diagram Olah data Barang Keluar



Gambar 4.11 Activity diagram olah data barang keluar

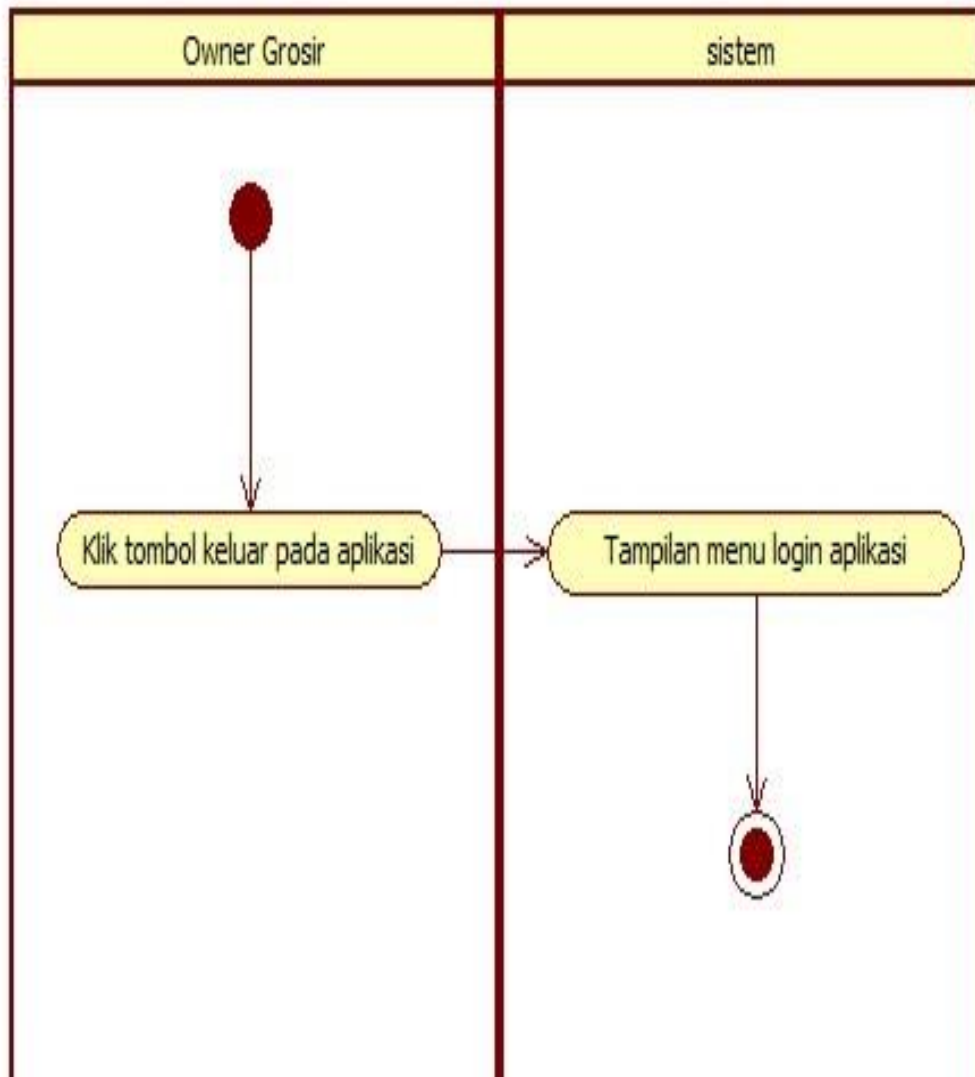
Pada Gambar 4.4 Activity diagram olah data barang keluar menggambarkan proses input data barang yang terjual ke dalam sistem, pegawai meng-klik menu tambah data baru dan sistem menampilkan form *input* data barang lalu pegawai mengelola data barang yang baru di order (bisa *input* atau mengubah data yang sudah ada) lalu menekan tombol *save* maka data akan otomatis tersimpan ke *database*.

### E. Activity Diagram Olah Laporan Order



Gambar 4.12 Activity diagram olah laporan order

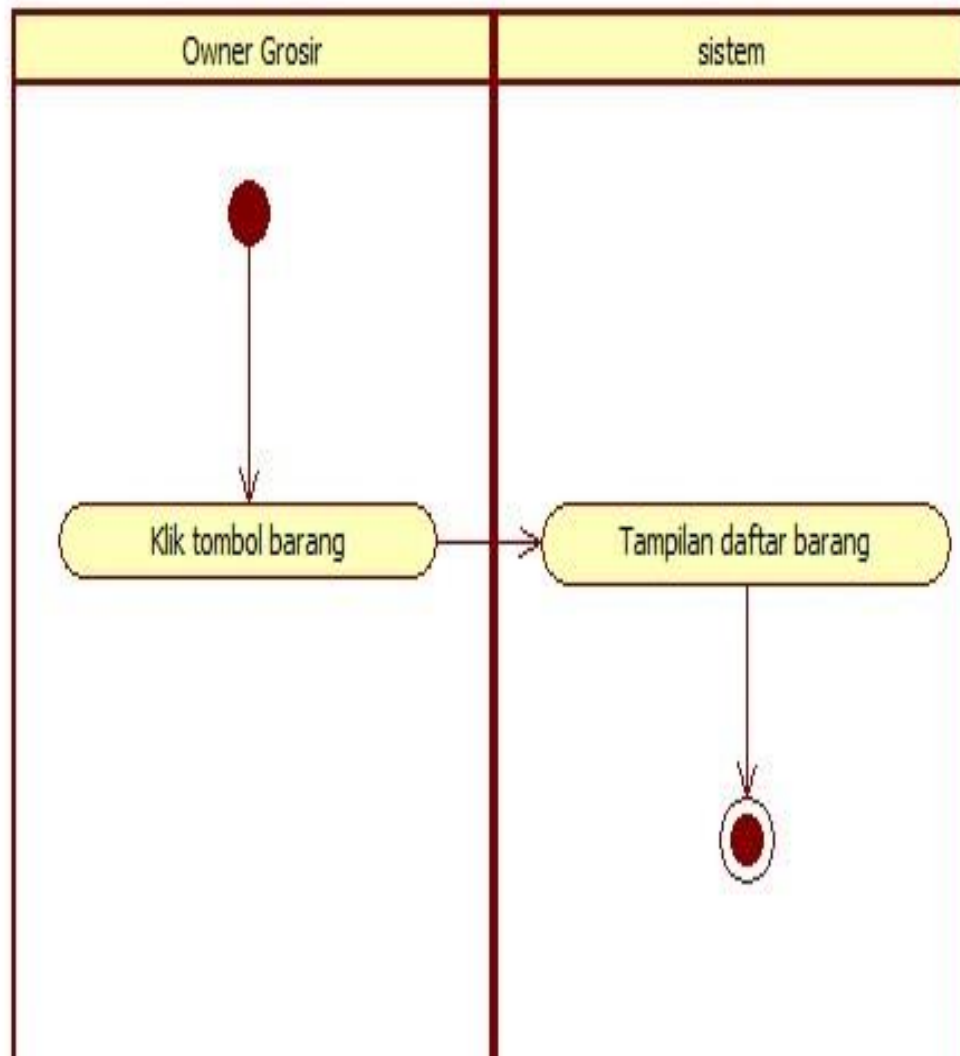
Pada Gambar 4.6 Activity diagram olah laporan order menggambarkan proses pembuatan laporan order pertama pegawai klik menu *dashboard*, lalu pada menu *dashboard* akan tampil barang apa saja yang harus di sediakan kembali.

**F. Activity Diagram Logout**

*Gambar 4.13 Activity diagram logout*

Pada Gambar 4.7 Activity diagram *logout* menggambarkan proses yang dilakukan oleh pegawai ketika akan *logout* dari aplikasi. Pertama pegawai klik tombol keluar yang berada pada pojok kanan atas aplikasi, maka pegawai akan langsung di arahkan ke halaman *login* awal aplikasi.

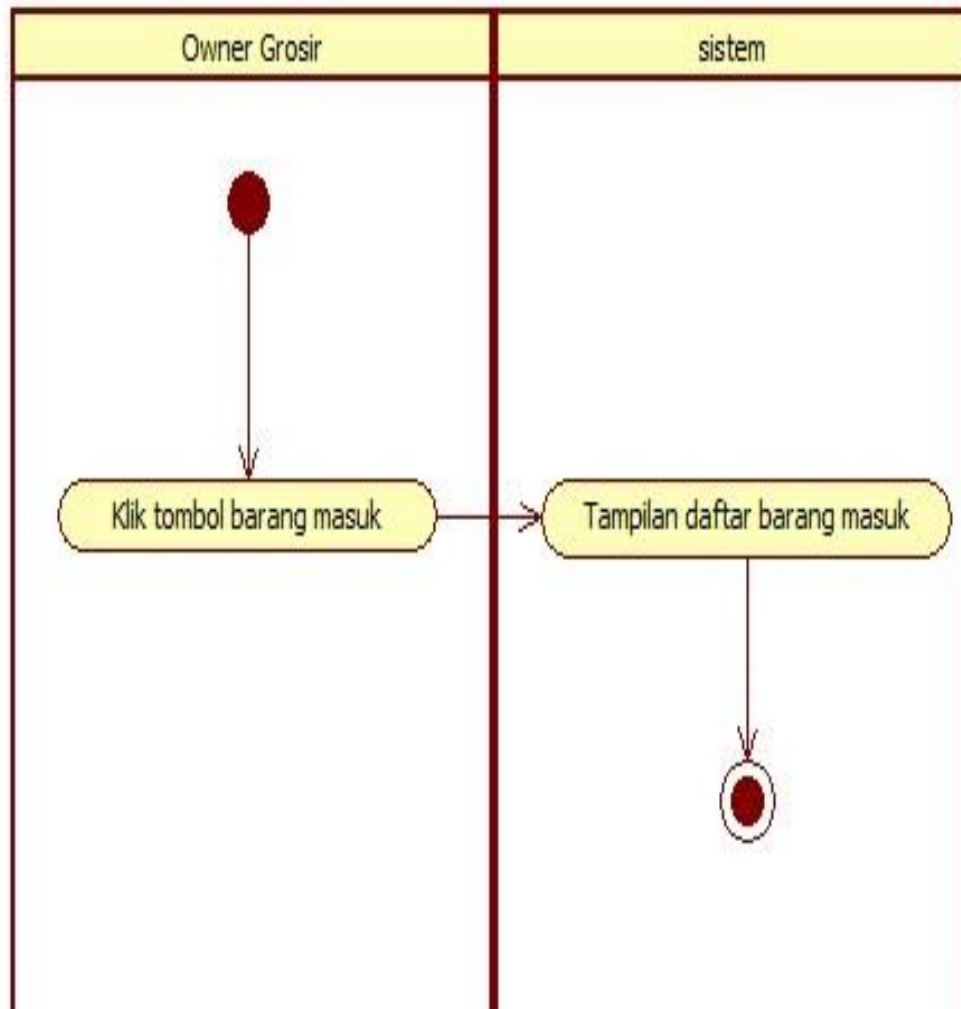
### G. Activity Diagram Laporan Data Barang



*Gambar 4.14 Activity diagram lapoan data barang*

Pada Gambar 4.8 Activity diagram lapoan data barang menggambarkan proses yang di lakukan oleh owner grosir ketika akan melihat laporan data barang yang telah di kelola oleh pegawai, hal pertama yang di lakukan oleh *owner* grosir adalah klik tombol barang, lalu sistem akan menampilkan laporan data barang yang telah di kelola oleh pegawai.

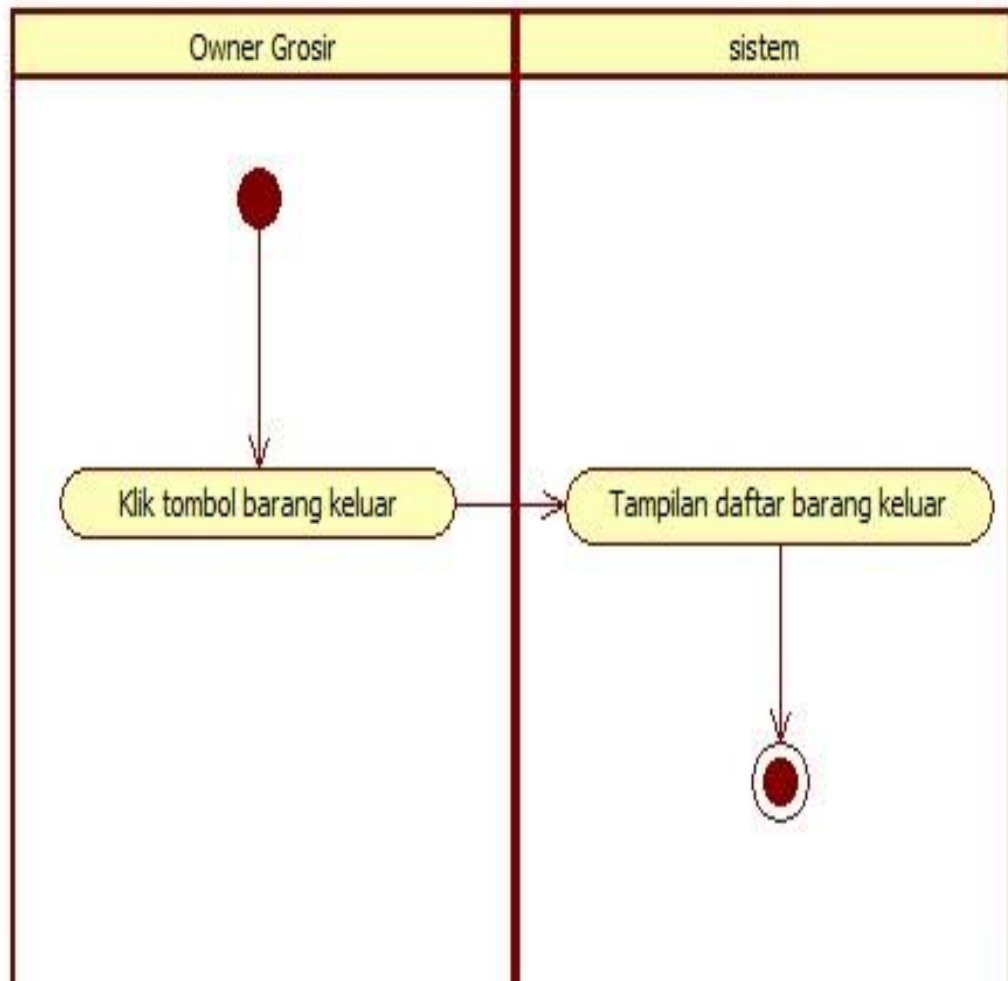
## H. Activity Diagram Data Barang Masuk



Gambar 4.15 Activity diagram data barang masuk

Pada Gambar 4.9 Activity diagram laporan data barang masuk menggambarkan proses yang dilakukan oleh *owner* grosir ketika akan melihat laporan data barang masuk yang telah dikelola oleh pegawai, hal pertama yang dilakukan oleh *owner* grosir adalah klik tombol barang masuk, lalu sistem akan menampilkan laporan data barang masuk yang telah dikelola oleh pegawai.

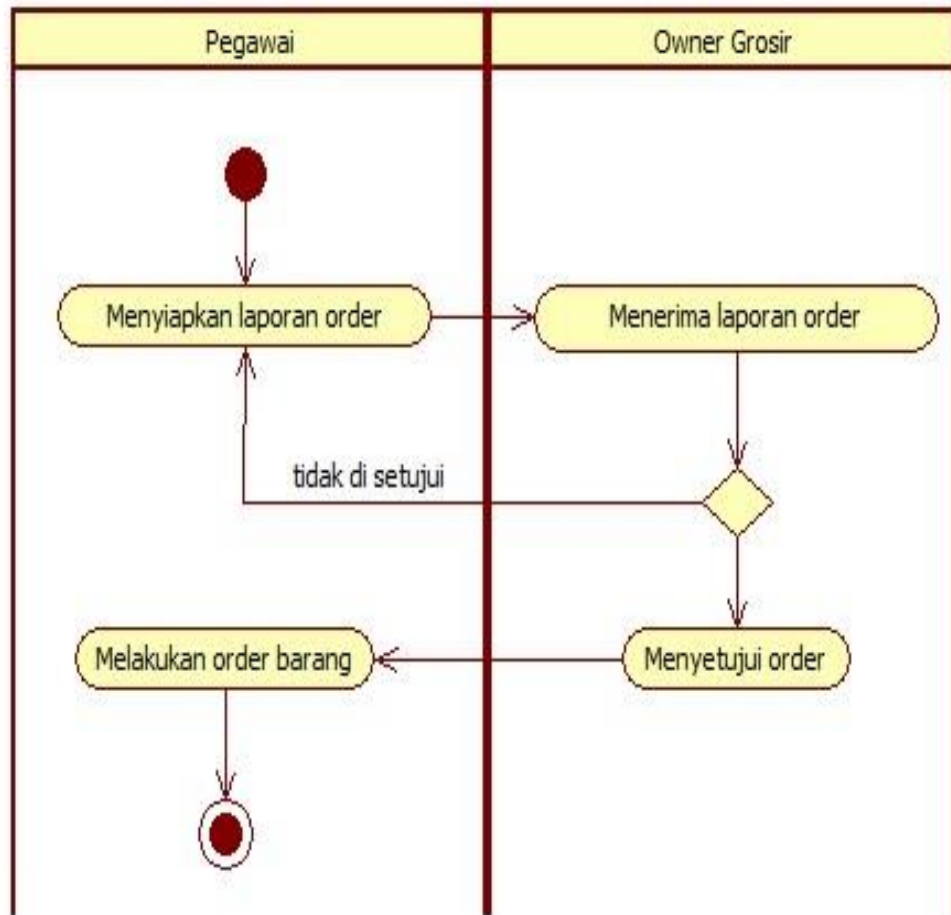
### I. Activity Diagram Data Barang Keluar



Gambar 4.16 Activity diagram data barang keluar

Pada Gambar 4.10 Activity diagram laporan data barang keluar menggambarkan proses yang dilakukan oleh *owner* grosir ketika akan melihat laporan data barang keluar yang telah dikelola oleh pegawai, hal pertama yang dilakukan oleh *owner* grosir adalah klik tombol barang keluar, lalu sistem akan menampilkan laporan data barang keluar yang telah dikelola oleh pegawai.

### J. Activity Diagram Order Barang

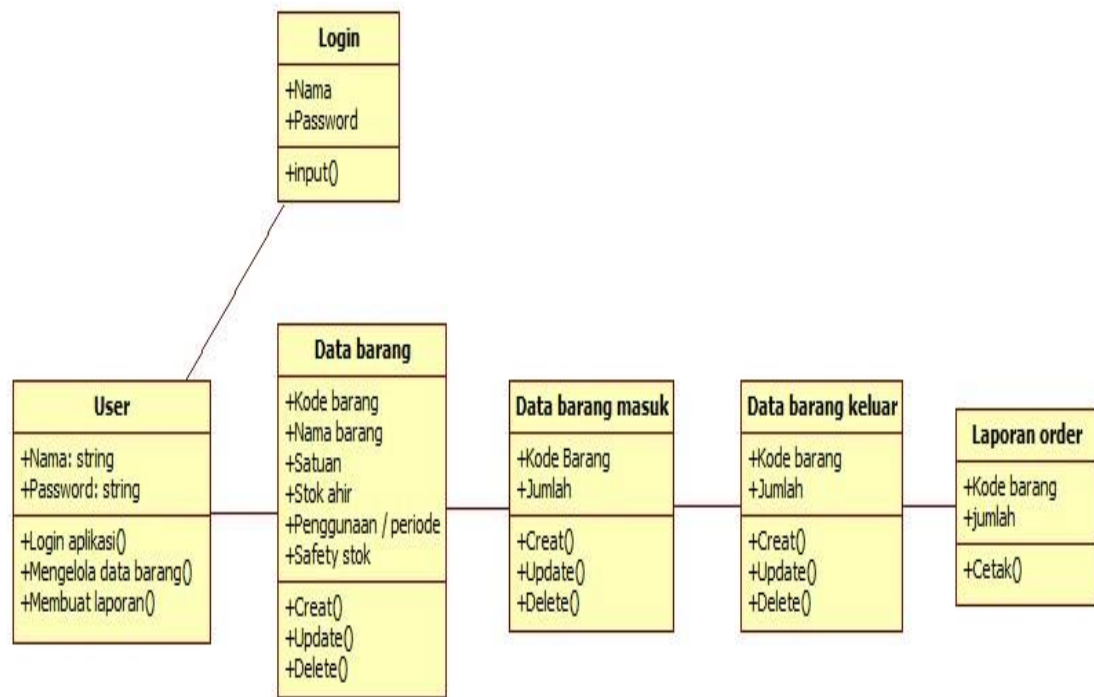


Gambar 4.17 Activity diagram order barang

Pada Gambar 4.10 Activity diagram order barang menggambarkan proses yang dilakukan pegawai ketika akan melakukan order barang, pertama pegawai menyiapkan laporan order menggunakan aplikasi lalu pegawai menyerahkan laporan order kepada owner grosir. Jika laporan order disetujui oleh owner grosir maka pegawai akan langsung melakukan order barang, tetapi jika laporan order tidak disetujui oleh owner grosir maka pegawai akan mempersiapkan kembali laporan order sampai laporan tersebut disetujui oleh owner grosir.



### 4.2.3 Class Diagram



Gambar 4.18 Class diagram aplikasi pengendalian inventory

Pada gambar 4.11 *Class diagram* (diagram kelas) ini menggambarkan struktur sistem aplikasi pengendalian *inventory* yang diusulkan dari segi pendefinisian kelas-kelas yang akan dibuat untuk membuat aplikasi pengendalian *inventory* pada grosir sembako.

### 4.2.4 Perancangan Database

Perancangan database adalah proses untuk menemukan isi dan pengaturan data yang di butuhkan untuk mendukung berbagai rencana apliaksi yang akan di bangun pada aplikasi pengendalian *inventory*. Dapat di lihat pada tabel-tabel berikut ini :

### A. Data barang

Tujuan di buatnya *form input* data barang adalah untuk meng-*input*-kan data ke dalam *database*, dan rancangan *database* nya dapat di lihat pada tabel di bawah ini :

*Tabel 4.8 Perancangan Database Data barang*

No	Field	Type	Panjang	Keterangan
1	Kode_barang	Varchar	5	Menyimpan kode barang
2	Nama_barang	Varchar	20	Menyimpan nama barang
3	Satuan	Varchar	5	Menyimpan satuan barang
4	Stok_ahir	Number	5	Menyimpan stok ahir barang
5	Harga	Number	6	Menyimpan harga barang
6	Penggunaan_periode	Number	3	Menyimpan penggunaan / periode
7	Safety_stok	Number	4	Menyimpan <i>safety stok</i>
8	EOQ	Number	4	Menyimpan EOQ
9	ROP	Number	4	Menyimpan ROP

### B. Data barang masuk

Tujuan di buatnya *form input* data barang masuk adalah untuk meng-*input*-kan data ke dalam *database*, dan rancangan *database* nya dapat di lihat pada tabel berikut ini :

Tabel 4.9 Perancangan Database Data Barang Masuk

No	Field	Type	Panjang	Keterangan
1	Id	Number	4	Menyimpan Id barang
2	Kode_barang	Varchar	5	Menyimpan kode barang
3	Jumlah	Number	4	Menyimpan jumlah barang
4	Tanggal	Varchar	10	Menyimpan tanggal input
5	User_Id	Number	1	Menyimpan Id user

### C. Data barang keluar

Tujuan di buatnya form input data barang keluar adalah agar user dapat mengetahui barang yang keluar sehingga data pada tabel data barang akan berkurang dan untuk meng-input-kan data ke dalam *database*, dan rancangan *database* nya dapat di lihat pada tabel di bawah ini :

Tabel 4.10 Perancangan Database Data Barang Keluar

No	Field	Type	Panjang	Keterangan
1	Id	Number	4	Menyimpan Id barang
2	Kode_barang	Varchar	5	Menyimpan kode barang
3	Jumlah	Number	4	Menyimpan jumlah barang
4	Tanggal	Varchar	10	Menyimpan tanggal input
5	User_Id	Number	1	Menyimpan Id user

#### D. Data rules

Tujuan di buatnya form input *rules* adalah untuk meng-*input*-kan data / aturan ke dalam *database*, dan rancangan *database* nya dapat di lihat pada tabel di bawah ini :

*Tabel 4.11 Perancangan Database Data Rules*

No	Field	Type	Panjang	Keterangan
1	Id	Number	4	Menyimpan Id barang
2	Biaya_pemesanan	Number	4	Menyimpan biaya pemesanan
3	Biaya_penyimpanan	Number	7	Menyimpan biaya penyimpanan
4	Lead_time	Number	1	Menyimpan lead time

#### E. Data user

Tujuan di buatnya form input data *user* adalah untuk meng-*input*-kan data ke dalam *database*, dan rancangan *database* nya dapat di lihat pada tabel di bawah ini :

*Tabel 4.12 Perancangan Database Data User*

No	Field	Type	Panjang	Keterangan
1	User_name	Varchar	20	Menyimpan user name
2	User_password	Varcahr	20	Menyimpan password
3	User_fullname	Varchar	30	Menyimpan user fullname
4	Group	Varchar	7	Menyimpan group

#### 4.2.5 Desain

##### A. Form Log in

The diagram illustrates a web browser window titled "A Web Page". The address bar contains the text "http://". The main content area displays a login form with the following components:

- A "Login" heading.
- A text input field labeled "User name".
- A text input field labeled "Password".
- A checkbox labeled "Remember me".
- A "Login" button.

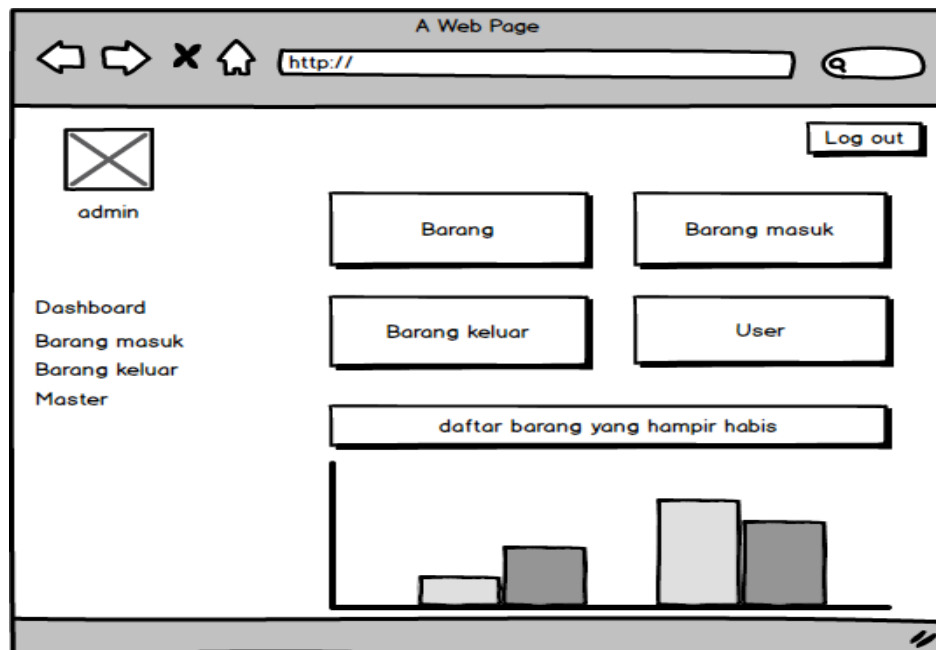
Gambar 4.19 Desain Halaman login aplikasi

Keterangan :

Tabel 4.13 Keterangan tombol pada halaman login user

No	Nama Tombol	Keterangan
1	<i>User Name</i>	Di gunakan untuk mengisi <i>user name</i> user
2	<i>Password</i>	Di gunakan untuk mengisi <i>password</i> user
3	<i>Button Remember me</i>	Di gunakan agar sistem mengingat <i>user name</i> dan <i>password</i> user
4	<i>Button Log in</i>	Di gunakan untuk <i>login</i> aplikasi

## B. Halaman awal aplikasi



Gambar 4.20 Desain Halaman awal aplikasi

Keterangan :

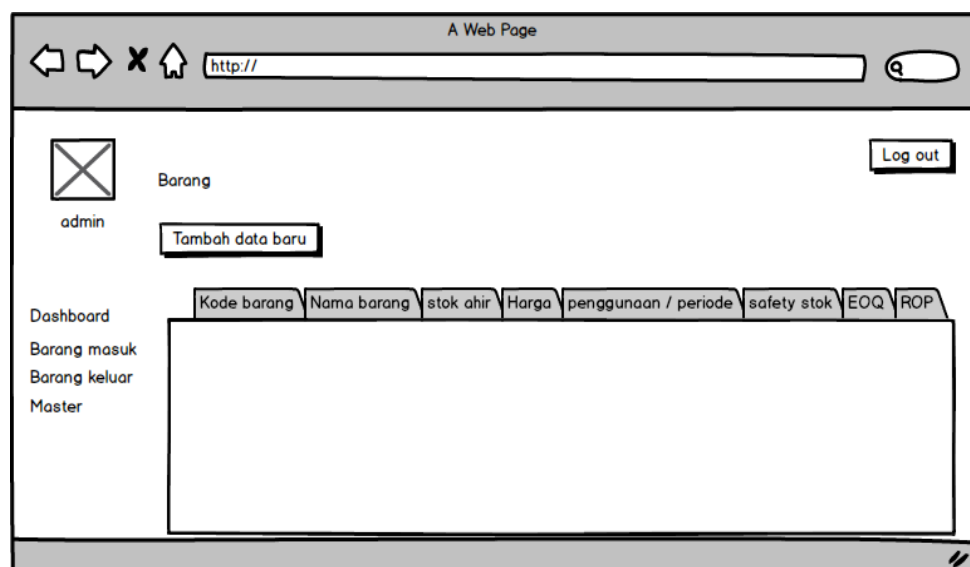
Tabel 4.14 Keterangan tombol pada halaman awal aplikasi

No	Nama Tombol	Keterangan
1	Foto	Di gunakan untuk <i>user avatar profil</i>
2	<i>Dashboard</i>	Di gunakan untuk pergi ke halaman menu utama
3	Barang masuk	Di gunakan untuk pergi ke halaman daftar barang masuk
4	Barang keluar	Di gunakan untuk pergi ke halaman daftar barang keluar
5	Master	Di gunakan untuk pergi ke halaman yang memuat halaman barang, <i>rules</i> , dan <i>user</i>

6	<i>Logout</i>	Di gunakan user untuk keluar dari aplikasi dan kembali ke form login
7	Barang	Di gunakan untuk pergi ke halaman daftar barang
8	Barang masuk	Di gunakan untuk pergi ke halaman daftar barang masuk
9	Barang keluar	Di gunakan untuk pergi ke halaman daftar barang keluar
10	<i>User</i>	Di gunakan untuk pergi ke halaman penampil <i>user</i> aplikasi
11	Daftar barang yang hampir habis	Adalah sebuah <i>pop up</i> yang akan menampilkan barang yang hampir habis
12	Diagram	Adalah sebuah tampilan yang menggambarkan barang yang ada pada aplikasi

*Lanjutan Tabel 4.15 Keterangan tombol pada halaman awal aplikasi*

### C. Halaman Data Barang



*Gambar 4.21 Desain Halaman data barang*

Keterangan :

*Tabel 4.16 Keterangan tombol pada halaman data barang*

No	Nama Tombol	Keterangan
1	Foto	Di gunakan untuk <i>user avatar profil</i>
2	<i>Dashboard</i>	Di gunakan untuk pergi ke halaman menu utama
3	Barang masuk	Di gunakan untuk pergi ke halaman daftar barang masuk
4	Barang keluar	Di gunakan untuk pergi ke halaman daftar barang keluar
5	Master	Di gunakan untuk pergi ke halaman yang memuat halaman barang, <i>rules</i> , dan <i>user</i>
6	<i>Logout</i>	Di gunakan user untuk keluar aplikasi dan kembali ke halaman form login
7	Tambah data baru	Di gunakan untuk pergi ke halaman input data barang
8	Tabel barang	Adalah tampilan tabel yang menunjukan daftar barang yang telah di <i>input</i> -kan user

#### D. Halaman Input Data Barang

The screenshot displays a web application titled 'A Web Page'. The browser address bar shows 'http://'. The user is logged in as 'admin'. The main content area is titled 'Barang keluar' and contains a form for adding new outgoing goods data. The form includes the following fields:

- ± Kode barang
- ± Nama Barang
- ± Satuan
- ± Stok ahir
- ± Harga
- ± Penggunaan / periode
- ± Safety stok

At the bottom of the form are two buttons: 'Create' and 'Reset'. On the left side, there is a navigation menu with the following links:

- Dashboard
- Barang masuk
- Barang keluar
- Master

In the top right corner, there is a 'Log out' button.

*Gambar 4.22 Desain Halaman input data barang*

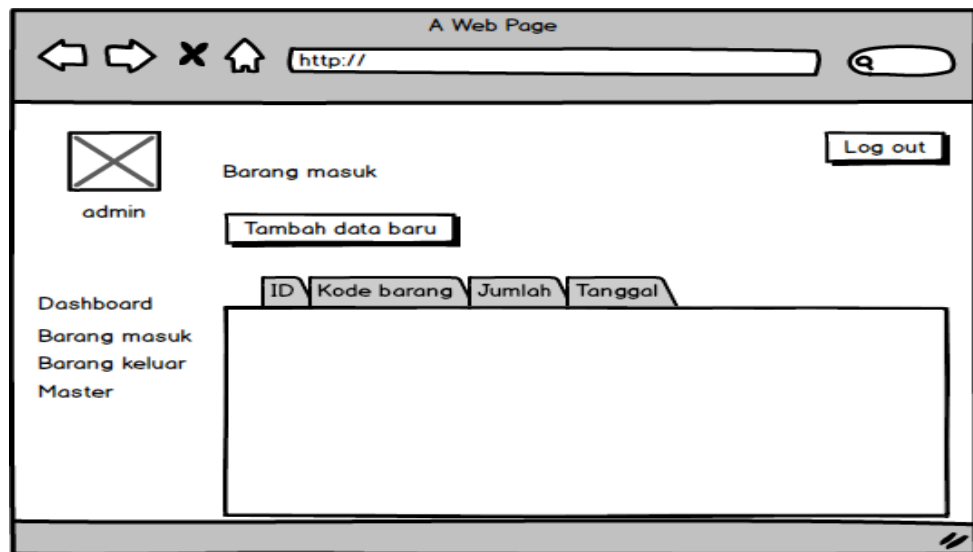


Keterangan :

*Tabel 4.17 keterangan tombol pada halaman input data barang*

No	Nama Tombol	Keterangan
1	Foto	Di gunakan untuk <i>user avatar profil</i>
2	<i>Dashboard</i>	Di gunakan untuk pergi ke halaman menu utama
3	Barang masuk	Di gunakan untuk pergi ke halaman daftar barang masuk
4	Barang keluar	Di gunakan untuk pergi ke halaman daftar barang keluar
5	Master	Di gunakan untuk pergi ke halaman yang memuat halaman barang, <i>rules</i> , dan <i>user</i>
6	<i>Logout</i>	Di gunakan user untuk keluar aplikasi dan kembali ke halaman <i>form login</i>
7	Kode barang	Di gunakan untuk <i>input</i> kode barang
8	Nama barang	Di gunakan untuk <i>input</i> nama barang
9	satuan	Di gunakan untuk <i>input</i> satuan barang
10	Stok ahir	Di gunakan untuk <i>input</i> stok yang tersisa pada grosir
11	Harga	Di gunakan untuk <i>input</i> harga barang
12	Penggunaan / periode	Di gunakan untuk <i>input</i> rata-rata penggunaan barang dalam satu periode
13	<i>Safety stok</i>	Di gunakan untuk <i>input</i> jumlah barang aman
14	<i>Creat</i>	Di gunakan untuk menyimpan data ke <i>database</i>
15	<i>Reset</i>	Di gunakan untuk mengosongkan semua bagian yang telah di isi

### E. Halaman Data Barang Masuk



Gambar 4.23 Desain Halaman data barang masuk

Keterangan :

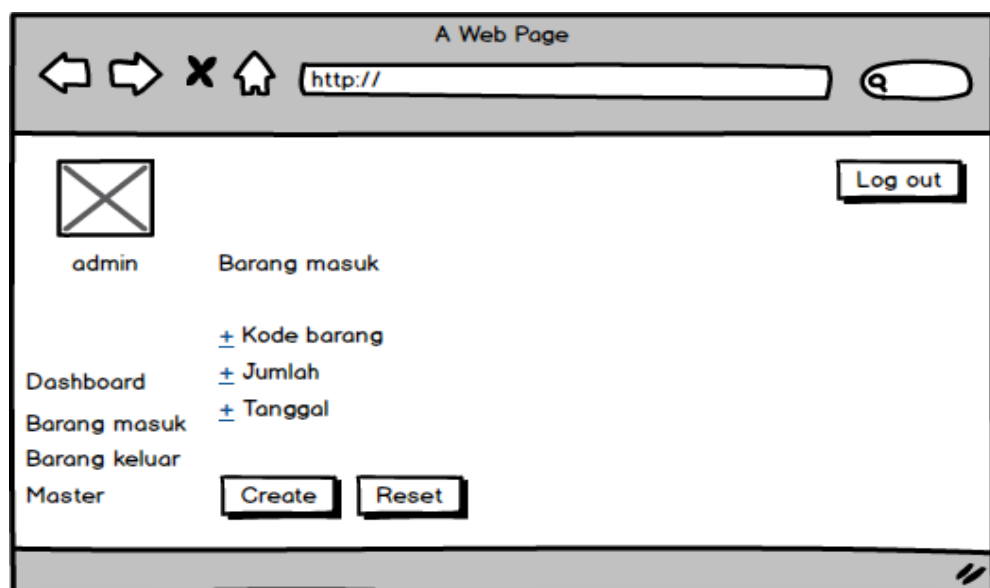
Tabel 4.18 Keterangan tombol pada halaman data barang masuk

No	Nama Tombol	Keterangan
1	Foto	Di gunakan untuk <i>user avatar profil</i>
2	<i>Dashboard</i>	Di gunakan untuk pergi ke halaman menu utama
3	Barang masuk	Di gunakan untuk pergi ke halaman daftar barang masuk
4	Barang keluar	Di gunakan untuk pergi ke halaman daftar barang keluar
5	Master	Di gunakan untuk pergi ke halaman yang memuat halaman barang, <i>rules</i> , dan <i>user</i>
6	<i>Logout</i>	Di gunakan user untuk keluar aplikasi dan kembali ke halaman <i>form login</i>

7	Tambah data baru	Di gunakan untuk input data barang masuk terbaru
8	Tabel	Menampilkan daftar barang baru yang telah di-input-kan

*Lanjutan Tabel 4.19 Keterangan tombol pada halaman data barang masuk*

## F. Halaman Input data Barang Masuk



*Gambar 4.24 Desain Halaman input data barang masuk*

Keterangan :

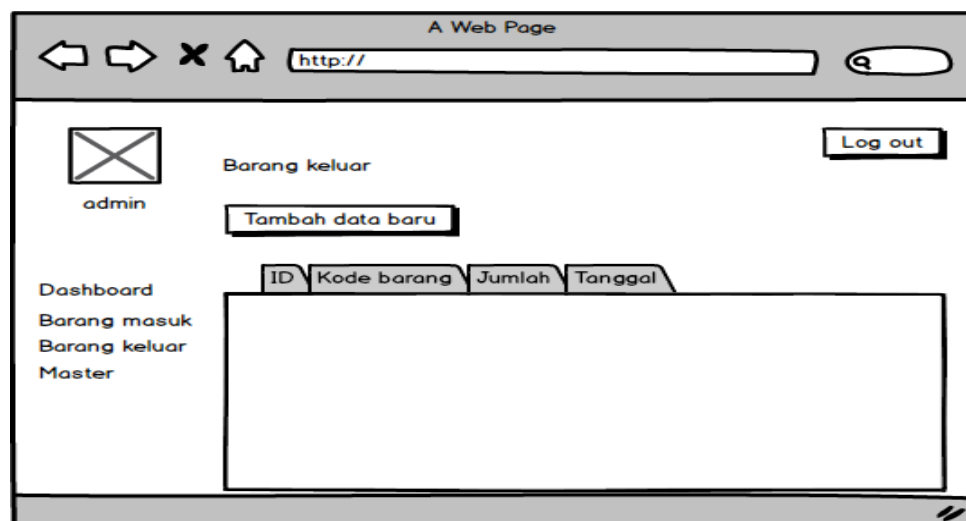
*Tabel 4.20 Keterangan tombol pada halaman input data barang masuk*

No	Nama Tombol	Keterangan
1	Foto	Di gunakan untuk user avatar profil
2	<i>Dashboard</i>	Di gunakan untuk pergi ke halaman menu utama
3	Barang masuk	Di gunakan untuk pergi ke halaman daftar barang masuk

4	Barang keluar	Di gunakan untuk pergi ke halaman daftar barang keluar
5	Master	Di gunakan untuk pergi ke halaman yang memuat halaman barang, <i>rules</i> , dan <i>user</i>
6	<i>Logout</i>	Di gunakan user untuk keluar aplikasi dan kembali ke halaman form login
7	Kode barang	Di gunakan untuk <i>input</i> kode barang
8	Jumlah	Di gunakan untuk <i>input</i> jumlah barang masuk
9	Tanggal	Di gunakan untuk <i>input</i> tanggal barang masuk

*Lanjutan Tabel 4.21 Keterangan tombol pada halaman input data barang masuk*

#### G. Halaman Data Barang Keluar



*Gambar 4.25 Desain Halaman data barang keluar*

Keterangan :

*Tabel 4.22 Keterangan tombol pada halaman data barang keluar*

No	Nama Tombol	Keterangan
1	Foto	Di gunakan untuk <i>user avatar profil</i>

2	<i>Dashboard</i>	Di gunakan untuk pergi ke halaman menu utama
3	Barang masuk	Di gunakan untuk pergi ke halaman daftar barang masuk
4	Barang keluar	Di gunakan untuk pergi ke halaman daftar barang keluar
5	Master	Di gunakan untuk pergi ke halaman yang memuat halaman barang, rules, dan user
6	<i>Logout</i>	Di gunakan user untuk keluar aplikasi dan kembali ke halaman form login
7	Tambah data baru	Di gunakan untuk input data barang keluar terbaru
8	Tabel	Adalah tampilan tabel yang menunjukan daftar barang keluar yang telah di input-kan user

*Lanjutan Tabel 4.23 Keterangan tombol pada halaman data barang keluar*

## H. Halaman Input Data Barang Keluar

The screenshot shows a web browser window titled "A Web Page". The address bar contains "http://". The page layout includes a sidebar on the left with a logo (a square with an 'X') and the text "admin". The sidebar menu items are: "Dashboard", "Barang masuk", "Barang keluar", and "Master". The main content area is titled "Barang keluar" and contains three expandable sections: "+ Kode barang", "+ Jumlah", and "+ Tanggal". At the bottom of the main content area are two buttons: "Create" and "Reset". A "Log out" button is located in the top right corner of the page.

*Gambar 4.26 Desain Halaman input data barang keluar*

Keterangan :

*Tabel 4.24 Keterangan tombol pada halaman input data barang keluar*

No	Nama Tombol	Keterangan
1	Foto	Di gunakan untuk <i>user avatar profil</i>
2	<i>Dashboard</i>	Di gunakan untuk pergi ke halaman menu utama
3	Barang masuk	Di gunakan untuk pergi ke halaman daftar barang masuk
4	Barang keluar	Di gunakan untuk pergi ke halaman daftar barang keluar
5	Master	Di gunakan untuk pergi ke halaman yang memuat halaman barang, <i>rules</i> , dan <i>user</i>
6	<i>Logout</i>	Di gunakan user untuk keluar aplikasi dan kembali ke halaman form login
7	Kode barang	Di gunakan untuk <i>input</i> kode barang
8	Jumlah	Di gunakan untuk <i>input</i> jumlah barang terjual
9	Tanggal	Di gunakan untuk <i>input</i> tanggal barang terjual

## BAB V

### IMPLEMENTASI DAN PENGUJIAN

#### 5.1 Implementasi

Implementasi ini dilakukan sesuai dengan perancangan aplikasi yang dilakukan pada bab sebelumnya. Terdiri dari tampilan struktur menu dan pengujian aplikasi.

##### 5.1.1 Implementasi Perangkat Keras dan Perangkat Lunak

Perangkat keras yang penyusun gunakan dalam tahap implementasi aplikasi inventory berbasis web pada grosir sembako tidak harus yang berspesifikasi tinggi. Adapun spesifikasi perangkat keras dan perangkat lunak yang digunakan adalah sebagai berikut :

- a. Spesifikasi perangkat keras

*Tabel 5.25 Spesifikasi Hardware yang digunakan*

No	Nama Perangkat	Spesifikasi
1	<i>Processor</i>	Intel pentium dual CPU E2160 @ 1,8 Ghz
2	Ram	2 x 2 GB DDR 2
3	<i>Harddisk</i>	232.88 GB

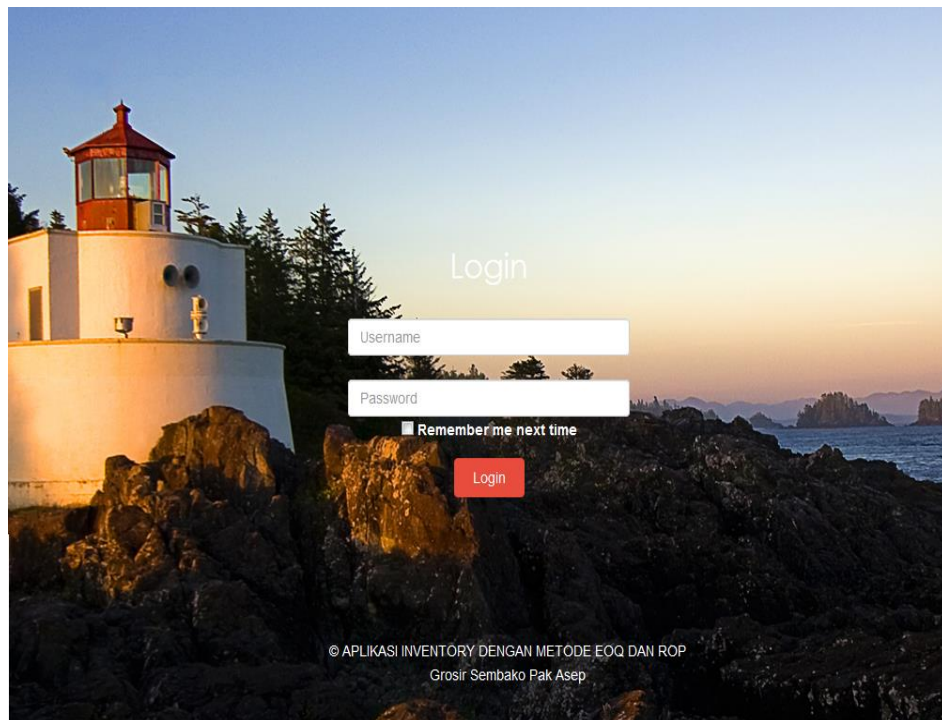
- b. Spesifikasi perangkat lunak

Spesifikasi perangkat lunak yang digunakan untuk menjalankan aplikasi *inventory* ini adalah XAMPP *Cotrol panel* V 3.2.2 dan *Windows* 7 64 bit.

## 5.2 Pengujian Sistem

Berdasarkan perancangan yang dilakukan telah diketahui bahwa terdapat beberapa tampilan *form* pada aplikasi yang bertujuan untuk mempermudah *user* dalam pengoperasian-nya yang terdiri dari *Login*, *Dashboard*, Halaman Data Barang, Halaman Barang Masuk, Halaman Barang Keluar, Halaman *User*, Halaman *Rules*, Form Isi Data Barang, Form Isi Data Barang Masuk, Form Isi Data Barang Keluar, Form Isi Data untuk *Rules*. Berikut ini adalah tahapan pengujian dari berbagai *form*:

### 1. *Login*



Gambar 5.27 Tampilan Interface Menu Login

Halaman *login* di buat untuk mengamankan data user sehingga tidak sembarang orang yang dapat mengaksesnya, terdapat kotak *username* untuk mengisi *username user*, kotak *password* untuk mengisi *password user*, pilihan *remember me* untuk membuat sistem mengingat *username* dan *password user* dan button *login* untuk *login* sistem.

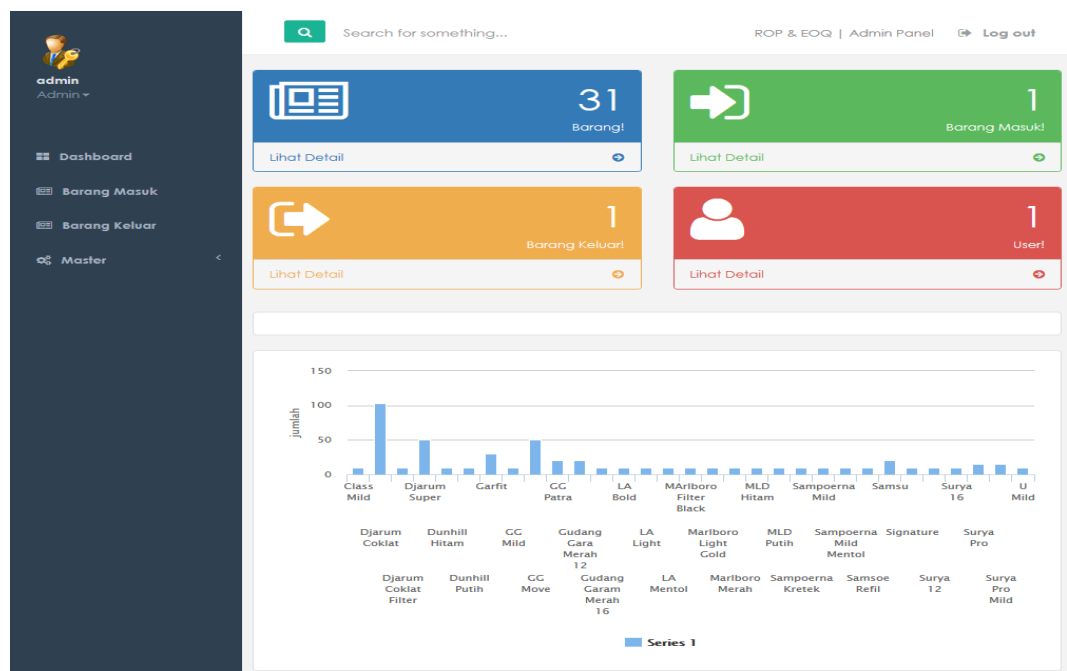


Pengujian :

Tabel 5.26 Pengujian Login

No	Tujuan	Input	Output diharapkan	Output sistem
1	Masuk kedalam menu <i>dashboard</i> aplikasi	<i>Username &amp; password</i>	Login dan aplikasi menampilkan <i>dashboard</i>	*Sukses * <i>Dashboard</i> tampil sesuai dengan <i>output</i>

## 2. Dashboard Aplikasi



Gambar 5.28 Tampilan Interface Dashboard Aplikasi

*Dashboard* aplikasi ini merupakan tampilan awal aplikasi pengendalian *inventory* dan di dalam nya terdapat berbagai macam tombol yang dapat diakses oleh pengguna aplikasi.

Pengujian :

*Tabel 5.27 Pengujian dashboard aplikasi*

<b>No</b>	<b>Tujuan</b>	<b>Input</b>	<b>Output diharapkan</b>	<b>Output sistem</b>
1	Lihat detail barang	Klik lihat detail pada menu barang	Menampilkan detail barang yang ada	*Sukses *Detail barang tampil sesuai dengan <i>output</i>
2	Lihat detail barang masuk	Klik lihat detail pada menu barang masuk	Menampilkan detail barang masuk	*Sukses *Detail barang masuk tampil sesuai dengan <i>output</i>
3	Lihat detail barang keluar	Klik lihat detail pada menu barang keluar	Menampilkan detail barang keluar	*Sukses *Detail barang keluar tampil sesuai dengan <i>output</i>
4	Lihat detail <i>user</i>	Klik lihat detail pada meu user	Menampilkan detail data user	*Sukses *Detail data user tampil sesuai dengan <i>output</i>
5	Menuju menu master	Klik menu master	Menampilkan menu master	*Sukses *Detail menu master tampil sesuai dengan <i>output</i>

### 3. Halaman Data Barang

admin Admin

Dashboard

Barang Masuk

Barang Keluar

Master

Search for something...

ROP & EOQ | Admin Panel Log out

Barang

Home / Master Data / Barang

Tambah Data Baru

Displaying 1-10 of 31 results.

Kode Barang	Nama Barang	Stok Akhir	Harga	Penggunaan/Periode	Safety Stok	Rop	Eoq
0001	Djarum Coklat	103	12500	30	20	20	5
0002	Djarum Super	50	16000	50	10	11	6
0003	MLD Putih	10	22000	10	3	3	3
0004	MLD Hitam	10	18000	10	3	3	3
0005	LA Bold	7	23000	10	3	3	3
0006	LA Light	10	20900	10	3	3	3
0007	LA Mentol	10	20000	10	3	3	3
0008	Djarum Coklat Filter	10	12800	10	3	3	3
0009	Garfil	30	16200	30	5	5	5
0010	Gudang Garam Merah 16	10	16000	10	3	3	3

Go to page: 1 2 3 4 Next

Gambar 5.29 Tampilan Interface Halaman Data Barang

Setelah user menekan tombol lihat detail pada kotak Data Barang maka sistem akan menampilkan daftar barang yang mana pada daftar tersebut memuat kode barang, nama barang, stok akhir, harga barang, penggunaan / periode, *safety stok*, perhitungan EOQ dan ROP.

- *Form input data barang*

admin Admin

Dashboard

Barang Masuk

Barang Keluar

Master

Search for something...

ROP & EOQ | Admin Panel Log out

Barang

Home / Barang / Create

Kode Barang \*

Nama Barang \*

Satuan \*

--Pilih--

Stok Akhir \*

Harga \*

Penggunaan/Periode \*

Safety Stok \*

Create Reset

Gambar 5.30 Tampilan Interface Halaman Input Data Barang

Gambar di berikut merupakan *form input* data barang yang mana *user* meng-*input*-kan kode barang, nama barang, satuan barang, stok ahir barang, harga barang, penggunaan / periode, dan *safety stok*.

Apa bila tidak ada form input data barang maka beberapa data tidak dapat di olah, sehingga menyebabkan aplikasi sulit untuk di pahami oleh user. Tujuan dari form input data barang adalah agar aplikasi dapat beroperasi seara optimal dan data barang memiliki berbagai macam data di antara nya adalah seperti yang telah di sebutkan pada paragraf pertama.

Pengujian :

*Tabel 5.28 Pengujian halaman data barang*

No	Tujuan	Input	Output diharapkan	Output sistem
1	Melihat detail data barang	Klik lihat detail pada menu barang yang ada di dashboard	Menampilkan data barang yang di <i>input</i> -kan user	*Sukses *Data barang tampil pada halaman aplikasi
2	<i>Input</i> data barang baru	Kode barang, nama barang, stok ahir, harga barang, penggunaan / peride, <i>safety stok</i> .	Data dapat di kelola sistem, tersimpan ke data base, dan data tampil pada halaman data barang pada aplikasi.	*Sukses *Data dikelola sistem, tersimpan ke database dan tampil pada halaman data barang pada aplikasi.

#### 4. Halaman Barang Masuk

admin  
Admin ▾

Dashboard

Barang Masuk

Barang Keluar

Master <

Search for something...

ROP & EOQ | Admin Panel Log out

### Barang Masuk

Home / Master Data / Barang Masuk

Tambah Data Baru

Displaying 1-4 of 4 results.

ID	Kode Barang	Jumlah	Tanggal
5	0001	100	2019-08-07
6	0020	10	2019-08-15
7	0027	5	2019-08-15
8	0030	4	2019-08-15

Gambar 5.31 Tampilan Interface Halaman Barang Masuk

Gambar di atas merupakan tampilan daftar data barang masuk yang di *input*-kan oleh user ke dalam aplikasi.

- **Form Isi Data Barang Masuk**

admin  
Admin ▾

Dashboard

Barang Masuk

Barang Keluar

Master <

Search for something...

ROP & EOQ | Admin Panel Log out

### Barang Masuk

Home / Barang / Create

Kode Barang \*

--Pilih--

Jumlah \*

Tanggal \*

2019-08-15

Create Reset

Gambar 5.32 Tampilan Interface Halaman Input Data Barang Masuk

Gambar berikut merupakan *form input* data barang yang mana *user* meng-*input*-kan kode barang, jumlah barang, dan tanggal *input* barang masuk.

Tujuan dari input data barang masuk adalah agar aplikasi memperoleh data yang akan di kelola dan dapat menampilkan data barang terbaru, apabila ada barang masuk tetapi user tidak meng-*input*-kan data barang masuk tersebut maka data tidak akan terolah oleh aplikasi, jadi aplikasi akan menampilkan data jumlah barang yang tidak akurat.

Pengujian :

*Tabel 5.29 Pengujian halaman barang masuk*

No	Tujuan	Input	Output diharapkan	Output sistem
1	Melihat detail barang masuk	Klik lihat detail pada menu barang masuk yang ada di <i>dashboard</i>	Menampilkan data barang masuk yang di inputkan user	*Sukses *Data barang masuk tampil pada halaman aplikasi
2	<i>Input</i> data barang masuk	Kode barang (nama barang), jumlah, taggal.	Data dapat di kelola sistem, data tersimpan ke <i>database</i> , dan tampil pada halaman data barang barang masuk	*Sukses *Data dikelola sistem, tersimpan ke <i>database</i> dan tampil pada halaman data barang masuk.

## 5. Halaman Barang Keluar



admin Admin

Dashboard

Barang Masuk

Barang Keluar

Master

Search for something...

ROP & EOQ | Admin Panel Log out

### Barang Keluar

Home / Master Data / Barang Keluar

Tambah Data Baru

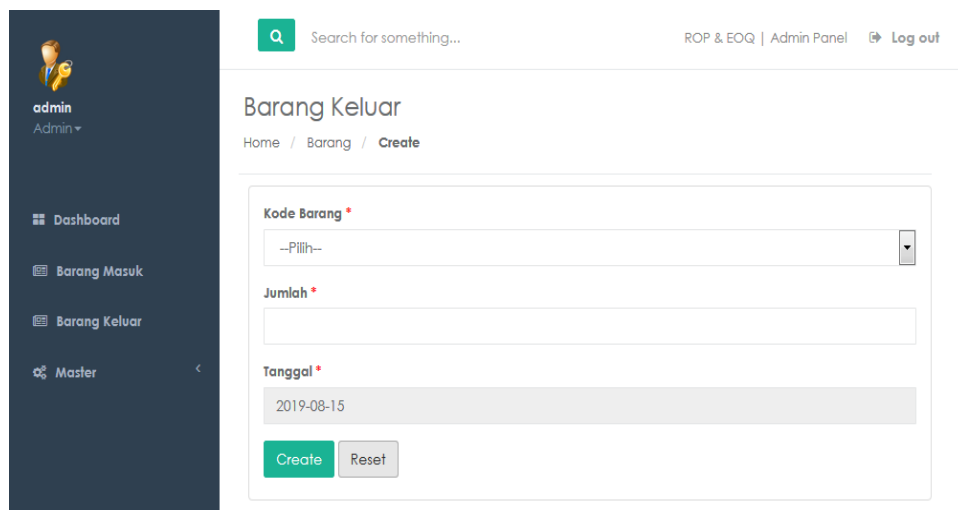
Displaying 1-4 of 4 results.

ID	Kode Barang	Jumlah	Tanggal
4	0001	5	2019-08-07
5	0005	3	2019-08-15
6	0023	2	2019-08-15
7	0019	10	2019-08-15

Gambar 5.33 Tampilan Interface Halaman Barang Keluar

Gambar di atas merupakan tampilan daftar data barang keluar yang di *input*-kan oleh user ke dalam aplikasi.

- *Form* Isi Data Barang Keluar



admin Admin

Dashboard

Barang Masuk

Barang Keluar

Master

Search for something...

ROP & EOQ | Admin Panel Log out

### Barang Keluar

Home / Barang / Create

Kode Barang \*

--Pilih--

Jumlah \*

Tanggal \*

2019-08-15

Create Reset

Gambar 5.34 Tampilan Interface Halaman Barang Keluar

Gambar di atas merupakan *form* input data barang yang mana *user* meng-*input*-kan kode barang, jumlah barang, dan tanggal *input* barang keluar.

Pengujian :

*Tabel 5.30 Pengujian Halaman barang keluar*

No	Tujuan	Input	Output diharapkan	Output sistem
1	Melihat detail barang keluar	Klik lihat detail pada menu barang keluar yang ada di <i>dashboard</i>	Menampilkan data barang keluar yang di inputka user	*Sukses *Data barang keluar tampil pada halaman aplikasi
2	Input data barang keluar	Kode barang (nama barang), jumlah, taggal.	Data dapat di kelola sistem, data tersimpan ke <i>database</i> , dan tampil pada halaman data barang barang masuk	*Sukses *Data dikelola sistem, tersimpan ke <i>database</i> dan tampil pada halaman data barang keluar.

## 6. Halaman User

The screenshot shows a web application interface for user management. On the left is a dark sidebar with a user profile 'admin' and navigation links: Dashboard, Barang Masuk, Barang Keluar, and Master. The top header includes a search bar, 'ROP & EOQ | Admin Panel', and a 'Log out' button. The main content area is titled 'User' and shows a breadcrumb 'Home / Master Data / User'. Below this is a 'Tambah Data Baru' button. A table displays user data with columns 'GROUP\_ID' and 'User Name'. The table shows one user named 'Admin'.

GROUP_ID	User Name
Admin	admin

*Gambar 5.35 Tampilan Halaman Form User*



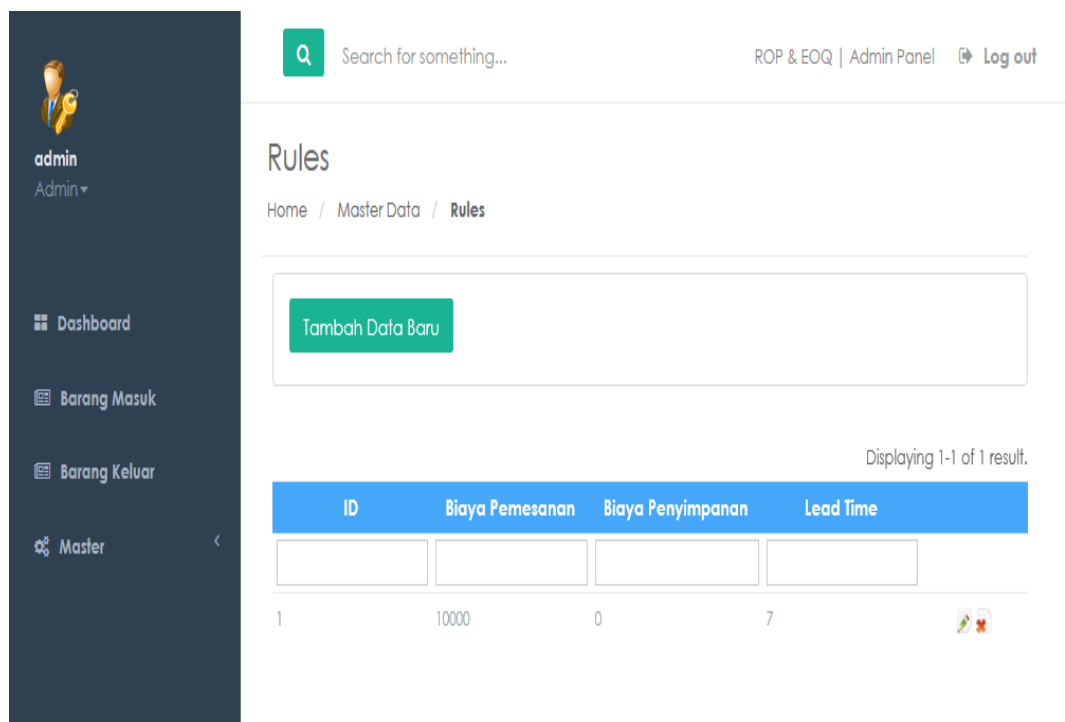
Gambar berikut merupakan *form* yang menampilkan berbagai informasi dari *user* aplikasi.

Pegujian :

*Tabel 5.31 Pengujian form user*

No	Tujuan	Input	Output diharapkan	Output sistem
1	Melihat data pengguna aplikasi	Klik lihat detail pada menu user yang ada di <i>dashboard</i>	Menampilkan data <i>user / pemakai aplikasi</i>	*Sukses *Data <i>user</i> tampil pada halaman aplikasi

## 7. Halama *Rules*



*Gambar 5.36 Tampilan Halaman Rules*

Gambar di atas merupakan tampilan daftar *Rules* yang digunakan untuk penghitungan EOQ dan ROP.

- *Form Isi Data untuk Rules*

Gambar 5.37 Tampilan Interface Form isi data untuk Rules

Gambar tersebut merupakan form *input rules* yang di gunakan untuk menghitung EOQ dan ROP pada aplikasi pengendalian *Inventory*.

Pengujian :

Tabel 5.32 Pengujian halaman rules

No	Tujuan	Input	Output diharapkan	Output sistem
1	Meihat data <i>rules</i> yang telah di inputkan	Klik menu rules yang ada pada <i>dashboard</i> aplikasi	Aplikasi menampilkan data <i>rules</i>	*Sukses *Data <i>rules</i> tampil pada halaman apliaksi
2	Input data <i>Rules</i>	Biaya pemesanan, biaya penyimpanan, dan <i>lead time</i>	Data dikelola oleh aplikasi,data tersimpa ke <i>database</i>	*Sukses *data dapat di kelola oleh aplikasi, dan data tersimpan ke <i>database</i>

### 5.3 Menghitung Economic Order Quantity dan Re-order Point

Total jumlah permintaan produk Djarum Coklat / Tahun di dapat dari data penjualan produk sebesar 18000 pcs. Dengan nilai biaya penyimpanan untuk setiap periode sebesar 25 %, biaya pemesanan sebesar 10%, safety stock = 20 pcs dan lead time = 3 hari. Perhitungannya adalah sebagai berikut :

$$EOQ = \sqrt{(2D \times S) / H}$$

Keterangan :

**D** = Permintaan product / Tahun, **S** = biaya pemesanan, **H** = biaya penyimpanan per periode.

D = 18000 pcs

S = 10%

H = 25%

$EOQ = \sqrt{(2) (18000) (10\%) / (25\%)} = \sqrt{14400} = 120 \text{ pcs}$

$$Reorder\ point = SS + (PT / 365) \times LT$$

Keterangan :

**SS** = *Safety Stock*, **PT** = Penggunaan / Tahun, **LT** = *Lead Time*

SS = 20, PT = 18000 pcs, dan LT = 3

$20 + (18000 / 365) \times 3 = 168$

## **BAB VI**

### **KESIMPULAN DAN SARAN**

#### **1.1 Kesimpulan**

Dengan terselesaikan-nya penelitian ini penyusun dapat mengimplementasi-kan aplikasi *inventory* dengan metode EOQ dan ROP pada Grosir Sembako, yang mana melalui metode EOQ (*Economic Order Quatity*) pemilik grosir dapat menentukan kuantitas pemesanan barang paling ekonomis, di lengkapi dengan metode ROP (*Re-order Point*), metode ini menentukan kapan barang harus di sediakan kembali sehingga mencegah terjadinya ketidak tersediaan produk.

#### **1.2 Saran**

Aplikasi ini masih dalam tahap pengembangan dan masih memiliki banyak kekurangan sehinga tidak menutup kemungkinan apabila terjadi peembangan di kemudia hari. Maka dari itu bila ingin mengembangkan aplikasi *Iventory* ini penyusun menyarankan untuk menambahkan fitur cetak dan menjadikan apliaksi ini *online* sehingga dapat di akses di mana saja.

## DAFTAR PUSTAKA

- Agus Saputra. (2013). *Smarty PHP OOP Engine for PHP Template*. Bandung: PT.Elex Media Computindo.
- Haposan Naingolan. (2010). Perhitungan metode Economic Order Quantity (EOQ), Reorder Point (ROP) dan Safety Stock (SS) dalam mengendalikan persediaan bahan baku pada PT Nutune Batam. Bandung.
- Haris Spuro. (2012). Modul Pembelajaran Praktek Basis Data MySQL.
- Hengki Fitrayco dan Habibi. (2009). Makalah Pengendalian Persediaan (Infentory Control). 3.
- Hengki Fitrayco dan Habibi. (2009). Makalah Pengendalian Persediaan (Infentory Control). 2.
- Ivar Jacobson. (2010). *The Univied Modelig Languge Reverence Manual*.
- Nurhadi. (2017). Pondasi Dasar Pemograman Website.
- R. S. Presman. (2010). *Software Engineering A Practioner's approach 7th ed*. MC Grow Hill.
- Robi Muhamad. (209). Makalah Aplikasi Komputer. .
- Taufik Iqbal, D. A. (2016). Aplikasi mamagement Persediaan Barang Berbasis Economic Order Quantity.
- Tomi Lukmana, D. T. (t.thn.).
- Tomi Lukmana, Diana Trivena. (2015). Penerapan Metode EOQ dan ROP (Studi Kasus: PD. BARU). *Jurnal Teknik Informatika dan Sistem Informasi Volume 1 Nomor 3*.
- Tri Muhamad Hani. (2011). Metode EOQ (Econonomic Order Quantity).
- Turmudi, M. (2017). Aplikasi Optimasi dan Evisiensi Persediaan Menggunakan Metode EOQ, Savety Stock dan ROP di Toko Batik Nusantara.
- Wijaya. (2009). Model Driven Developement .

## LAMPIRAN

### Lampiran 1 Form Wawancara

Narasumber : Asep Suryadi  
Hari / Tanggal : Sabtu / 27 April 2019  
Waktu : 10.00  
Tempat : Grosir Sembako, Malaksari 003/010

No	Pertanyaan	Jawaban	
		Ya	Tidak
1.	Apakah pencatatan nota barang belanja masih manual?	✓	
2.	Apakah pengumpulan nota belanja sudah tersusun rapih?	✓	
3.	Apakah dengan akan di terapkannya sistem komputerisasi dapat membantu pemilik Grosir?	✓	
4.	Apakah sudah ada aplikasi untuk mengelola <i>inventory</i> grosir?	✓	
6.	Apakah pengelolaan yang cepat, efisien dan akurat dapat membantu?	✓	

Bandung, 25 April 2019

Narasumber

Asep Suryadi

## Lampiran 2 Listing Program

### Form Login

```
<style type="text/css">
body
{
    background: url('images/bg.jpg') fixed;
    background-size: cover;
    padding: 0;
    margin: 0;
    font-family: 'Roboto', sans-serif;
}
h1, h2, h3, h4, h5, h6 {
    font-weight: 400;
    color: #fff;
}

label {
    display: inline-block;
    margin-bottom: 5px;
    font-weight: bold;
    color: #fff;
}

p {
    margin: 0px 0px 10px;
    color: #fff;
}
```

```

form-control {
    width: 300px;
}

.post .entry-header, .post .entry-title, .post .entry-
meta, .post .entry-image, .post .entry-content {
    background: none repeat scroll 0% 0% transparent;
    border: medium none;
    color: rgb(255, 255, 255);
}

input, button, select, textarea{
    color: #666;
}

</style>

<script type="text/javascript">
$(document).ready(function () {
    $('.forgot-pass').click(function(event) {
        $(".pr-wrap").toggleClass("show-pass-reset");
    });

    $('.pass-reset-submit').click(function(event) {
        $(".pr-wrap").removeClass("show-pass-reset");
    });
});
</script>

<div class="container" style="padding-bottom: 100px;
padding-top: 200px;">
<div class="entry-meta">

```



```

<center><div class="log">
    <h1><center>Login</center></h1><br>
    <?php $form=$this->beginWidget('CActiveForm', array(
        'id'=>'login-form',
        'enableClientValidation'=>true,
        'clientOptions'=>array(
            'validateOnSubmit'=>true,
        ),
    )); ?>
    <div class="row"><center>
        <?php echo $form-
>textField($model,'username',array('class'=>'form-
control','placeholder'=>'Username')); ?>
        <?php echo $form->error($model,'username'); ?>
    </center></div><br>

    <div class="row"><center>
        <?php echo $form-
>passwordField($model,'password',array('class'=>'form-
control','placeholder'=>'Password')); ?>
        <?php echo $form->error($model,'password'); ?>
    </div>

    <div class="row rememberMe"><center>
        <?php echo $form-
>checkBox($model,'rememberMe'); ?>
        <?php echo $form->label($model,'rememberMe');
?>
        <?php echo $form->error($model,'rememberMe');
-

```

```

div class="row buttons"><center>

    <?php echo
    CHtml::submitButton('Login',array('class'=>'btn btn-
    primary')); ?>

    </center></div>

    <?php $this->endWidget(); ?>
</div></center>
</div> <!-- .entry-meta -->

```

### ***Form Barang***

```

?>
<h1>View Barang #<?php echo $model->kode_barang; ?></h1>

<?php $this->widget('zii.widgets.CDetailView', array(
    'data'=>$model,
    'attributes'=>array(
        'kode_barang',
        'nama_barang',
        'satuan',
        'stok_akhir',
        'harga',
        'penggunaan_tahun',
        'safety_stok',
        'rop',
        'eoq',
    ),
)); ?>

```

## Form Barang Masuk

```

<?php
/* @var $this BarangMasukController */
/* @var $model BarangMasuk */

$this->breadcrumbs=array(
    'Barang Masuks'=>array('index'),
    $model->id,
);
$this->menu=array(
    array('label'=>'List BarangMasuk',
    'url'=>array('index')),
    array('label'=>'Create BarangMasuk',
    'url'=>array('create')),
    array('label'=>'Update BarangMasuk',
    'url'=>array('update', 'id'=>$model->id)),
    array('label'=>'Delete BarangMasuk', 'url'=>'#',
    'linkOptions'=>array('submit'=>array('delete','id'=>$model
->id),'confirm'=>'Are you sure you want to delete this
item?')),
    array('label'=>'Manage BarangMasuk',
    'url'=>array('admin')),
);
?>

<h1>View BarangMasuk #<?php echo $model->id; ?></h1>

<?php $this->widget('zii.widgets.CDetailView', array(
    'data'=>$model,
    'attributes'=>array(

```

```

        'id',
        'kode_barang',
        'jumlah',
        'tanggal',
        'user_id',
    ),
)); ?>

```

### Form Barang Keluar

```

<?php
/* @var $this BarangKeluarController */
/* @var $model BarangKeluar */
$this->breadcrumbs=array(
    'Barang Keluars'=>array('index'),
    $model->id,
);
$this->menu=array(
array('label'=>'List BarangKeluar',
'url'=>array('index')),
array('label'=>'Create BarangKeluar',
'url'=>array('create')),
    array('label'=>'Update BarangKeluar',
'url'=>array('update', 'id'=>$model->id)),
    array('label'=>'Delete BarangKeluar', 'url'=> '#',
'linkOptions'=>array('submit'=>array('delete', 'id'=>$model->id), 'confirm'=>'Are you sure you want to delete this item?')),

```

```

array('label'=>'Manage BarangKeluar',
'url'=>array('admin')),
);
?>

<h1>View BarangKeluar #<?php echo $model->id; ?></h1>
<?php $this->widget('zii.widgets.CDetailView', array(
    'data'=>$model,
    'attributes'=>array(
        'id',
        'kode_barang',
        'jumlah',
        'tanggal',
        'user_id',
    ),
)); ?>

```

## Form Rules

```

<?php
/* @var $this RulesController */
/* @var $model Rules */
$this->breadcrumbs=array(
    'Rules'=>array('index'),
    $model->id,
);
$this->menu=array(
    array('label'=>'List Rules', 'url'=>array('index')),
    array('label'=>'Create Rules',
'url'=>array('create')),
    array('label'=>'Update Rules',
'url'=>array('update', 'id'=>$model->id)),

```

## Form Rules

```

        array('label'=>'Delete Rules', 'url'=>'#',
'linkOptions'=>array('submit'=>array('delete','id'=>$model->id),'confirm'=>'Are you sure you want to delete this item?')),
        array('label'=>'Manage Rules',
'url'=>array('admin')),
    );
?>>
<h1>View Rules #<?php echo $model->id; ?></h1>
<?php $this->widget('zii.widgets.CDetailView', array(
    'data'=>$model,
    'attributes'=>array(
'id',
        'biaya_pemesanan',
        'biaya_penyimpanan',
    ),
));
?>

```

### Lampiran 3 Biodata Penulis



**Pramudita Cahya Firdaus** lahir di Purwakarta, pada 18 November 1996. Anak ke tiga dari tiga bersaudara, yang terlahir dari pasangan Asep Suryadi dan Udi Yuniarti. Mulai mengenyang pendidikan di SDN Pingirsari 1 (2004-2009), kemudian melanjutkan pendidikan ke MTS Al-Basyariyah Ksmpus 3 Arjasari (2009-2012), serta melanjutkan masa SMA di MA Al-basyariyah (2012-2015). Untuk mendapatkan gelar sarjana penulis melanjutkan S1 penulis melanjutkan pendidikan di Universitas Bale Bandung Fakultas Teknologi Informasi Jurusan Teknik Informatika. Penulis juga pernah aktif sebagai anggota di UKM Bale Percussion di FTI UNIBBA pada tahun 2015-2016, selain itu penulis pernah aktif sebagai ketua BEM FTI UIBBA pada tahun 2018.

**“BUKANLAH SEORANG PEMUDA IA YANG BANGGA  
MENGATAKAN INILAH BAPAK SAYA, TETAPI SEORANG PEMUDA  
ADLAH IA YANG DENGAN BANGGANYA MENGATAKAN INI LAH  
SAYA ”**

**“HADAPI DAN CERMATI NISCAYA KAN KAU DAPAT JALAN KELU**