**RANCANG BANGUN APLIKASI SISTEM INFORMASI INVENTORY E-LIQUID MENGGUNAKAN METODE**

**EOQ DI CV. VISYA PRATAMA INDUSTRI**

**SKRIPSI**

Karya Tulis sebagai syarat memperoleh

Gelar Sarjana Teknik dari Fakutas Teknologi Informasi

Universitas Bale Bandung

Disusun oleh :

**EGI NOVIANSYAH**

**NPM C1A150022 / 311150022**



**PROGRAM STRATA 1**

**PROGRAM STUDI TEKNI INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS BALE BANDUNG**

**BANDUNG**

**2021**

LEMBAR PERSETUJUAN PEMBIMBING

RANCANG BANGUN APLIKASI SISTEM INFORMASI

INVENTORY E-LIQUID MENGGUNAKAN METODE

EOQ DI CV. VISYA PRATAMA INDUSTRI

Disusun Oleh :

EGI NOVIANSYAH

NPM C1A150022 / 311150022

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

**SARJANA KOMPUTER**

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS BALE BANDUNG**

Baleendah, Juli 2021

Disetujui Oleh :

|  |  |
| --- | --- |
| Pembimbing utama  Rustiyana, S.T., M.T  NIK. 04104808015 | Pembimbing pendamping  Sutiyono W.P, S.T., M.Kom  NIK. 01043180002 |

# 

LEMBAR PERSETUJUAN PENGUJI

RANCANG BANGUN APLIKASI SISTEM INFORMASI

INVENTORY E-LIQUID MENGGUNAKAN METODE

EOQ DI CV. VISYA PRATAMA INDUSTRI

Disusun Oleh :

EGI NOVIANSYAH

NPM C1A150022 / 311150022

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

**SARJANA KOMPUTER**

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS BALE BANDUNG**

Baleendah, Juli 2021

Disetujui Oleh :

|  |  |
| --- | --- |
| Penguji 1  Rosmalina, S.T, M.Kom  NIK. 04104808122 | Penguji 2  Khilda Nistrina, S.Pd., M.Sc  NIK. 04104820004 |

# 

LEMBAR PENGESAHAN PROGRAM STUDI

RANCANG BANGUN APLIKASI SISTEM INFORMASI

INVENTORY E-LIQUID MENGGUNAKAN METODE

EOQ DI CV. VISYA PRATAMA INDUSTRI

Disusun Oleh :

EGI NOVIANSYAH

NPM C1A150022 / 311150022

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

**SARJANA KOMPUTER**

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS BALE BANDUNG**

Baleendah, Juli 2021

Disetujui Oleh :

|  |  |
| --- | --- |
| Mengetahui,  Dekan  Yudi Herdiana, S.T., M.T.  NIK. 04104808008 | Mengesahkan,  Ketua Prodi Studi  Yusuf Muharam, M.Kom  NIK. 04104820003 |

LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini :

Nama : Egi Noviansyah

NPM : C1A150022 / 311150022

Judul Skripsi : RANCANG BANGUN APLIKASI SISTEM INFORMASI

INVENTORY E-LIQUID MENGGUNAKAN METODE

EOQ DI CV. VISYA PRATAMA INDUSTRI

Menyatakan dengan sebenarnya bahwa penulis laporan skripsi ini berdasarkan hasil penelitian, pemikiran dan pemaparan asli dari penyusun sendiri, baik untuk naskah laporan maupun kegiatan pemograman yang tercantum sebagai bagian dari laporan skripsi ini, jika terdapat karya orang lain maka penyusun akan mencantumkan sumber secara jelas dan apabila ada karya pihak lain yang ternyata memiliki kemiripan dengan karya penyusun yang telah penyusun buat ini, maka hal ini adalah di luar pengetahuan penyusun dan terjadi tanpa kesengajaan.

Dengan demikian pernyataan ini penyusun buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidak benaran dalam pernyataan ini, maka penyusun bersedia menerima sanksi akademik yang sesuai dengan norma yang berlaku di perguruan tinggi ini.

Bandung, Juli 2021

Yang Membuat Pernyataan

Egi Noviansyah

NPM 311150022 / C1A150022

# 

# ABSTRAK

Dalam mengerjakan tugas akhir ini penyusun bertujuan untuk merancang dan membangun aplikasi *inventory* (persediaan barang) pada sebuah perusahaan yang bernama CV.VISYA PRATAMA INDUSTRI. Dimana penelitian ini bertujuan untuk membangun suatu aplikasi yang mana melalui aplikasi tersebut pemilik atau admin perusahaan dapat memantau dan mengendalikan *inventory* (persediaan barang) yang ada pada perusahaan yang dimaksud.

Persediaan (*Inventory*) adalah bahan-bahan atau barang (sumberdaya-sumberdaya organisasi) yang disimpan, yang akan dipergunakan untuk memenuhi tujuan tertentu, misalnya: untuk proses produksi maupun untuk di jual, suatu perusahaan tidak dapat disebut perusahaan apabila tidak memiliki persediaan (*inventory*). *EOQ* merupakan volume atau jumlah pembelian yang paling ekonomis untuk dilaksanakan 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 ditentukan maka dilakukanlah *re-order.* *Reorder* *Point* menjelaskan kapan saat seharusnya diadakan pemesanan lagi (*re*-*stock*), sehingga penerimaan bahan yang di pesan tepat jumlahnya dan juga tepat pada waktunya. Berdsasarkan pernyataan di atas penyusun telah membuat suatu aplikasi yang dapat mengefisiensikan pemilik perusahaan dalam megelola *inventory* sehingga diharapkan melalui aplikasi yang sudah dibuat dapat mencegah terjadinya kehilangan data barang/*stock*, data barang masuk, data barang keluar dan lainnya*.*

Informasi yang telah dihasilkan pada perancangan aplikasi ini adalah dapat mengelola data barang masuk dan keluar, data stok barang, data user, data pelanggan dan data pemesanan barang menggunakan metode EOQ dan ROP.

Kata kunci : *Perusahaan*, *Inventory, EOQ dan ROP*

# 

# 

# *ABSTRACT*

*In doing this final project, the author aims to design and build an inventory application at a company called CV. VISYA PRATAMA INDUSTRI. Where this study aims to build an application through which the owner or company admin can monitor and control the inventory (inventory of goods) in the company in question. Inventory is materials or goods (organizational resources) that are stored, which will be used to fulfill certain purposes, for example: for the production process or for sale, a company cannot be called a company if it does not have inventory. (inventory). EOQ is the most economical volume or number of purchases to be carried out at each purchase. So that by applying the EOQ model in purchasing, ordering costs and storage costs can be suppressed, after the number of orders with the most economical costs is determined, a re-order is carried out. The reorder point explains when it should be re-stocked, so that the material ordered is received in the right quantity and on time. Based on the statement above, the authors have created an application that can streamline company owners in managing inventory so that it is hoped that through the application that has been made it can prevent loss of goods/stock data, incoming goods data, outgoing goods data and others. Because in making this web-based inventory application, the author builds an application that fits what the company needs. In carrying out this research, the authors will use MDD (Model Drive Development) as the basis for the research methodology, EOQ (Economic Order Quantity) and ROP (Re-order Point) methods to control inventory, UML Diagram (Unived Modeling language) to make system designs. The programming language used is Hypertext Preprocessor (PHP) with MySql as the database.*

*Keywords: Company, Inventory, EOQ and ROP MDD, UML, PHP, MySQL.*

# KATA PENGANTAR

Segala puji serta syukur atas kehadirat Allah SWT atas berkat dan limpahan rahmat dan karunia-Nya sehingga laporan skripsi ini dapat diselesaikan dengan baik dan tepat pada waktunya. Laporan skripsi ini merupakan salah satu mata kuliah yang sangat wajib ditempuh selama perkuliahan di Fakultas Teknologi Informasi Universitas Bale Bandung. Laporan penilitian ini yang berjudul “RANCANG BANGUN APLIKASI SISTEM INFORMASI INVENTORY E-LIQUID MENGGUNAKAN METODE EOQ DI CV. VISYA PRATAMA INDUSTRI”. Disusun sebagai salah satu syarat untuk dapat melakukan penelitian dan mendapat gelar sarjana di Fakultas Teknologi Informasi Universitas Bale Bandung.

Dalam penyusunan laporan skripsi ini penulis banyak mendapat saran, dorongan, bimbingan serta keterangan dari berbagai pihak yang merupakan pengalaman yang tidak dapat diukur secara materi. Oleh karena itu, dengan segala hormat dan kerendahan hati perkenankan penulis mengucap terima kasih banyak yang sebesar-besarnya kepada :

1. Bapak Yudi Herdiana, S.T., M.T., selaku Dekan Fakultas Teknologi Informasi Universitas Bale Bandung.
2. Bapak Yusuf Muharam, M.Kom., selaku Ketua Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Bale Bandung.
3. Bapak Rustiyana, S.T., M.T., selaku pembimbing ke-1 proposal skripsi.
4. Bapak Sutiyono, S.T., M.Kom., selaku pembimbing ke-2 proposal skripsi.
5. Semua pihak yang telah membantu dan memberikan dukungan kepada penulis untuk menyelesaikan laporan skripsi ini.

Dalam penyusunan laporan skripsi ini, penyusun menyadari bahwa masih banyak terdapat kekurangan yang di buat baik sengaja maupun tidak sengaja, dikarenakan keterbatasan ilmu pengetahuan dan wawancara serta kurangnya pengalaman yang penyusun miliki. Untuk itu penyusun memohon maaf atas segala kekurangan tersebuat tidak menutup diri atas segala kritik dan saran yang membangun akan penyusun terima dengan baik.

Akhir kata semoga laporan skripsi ini bisa diterima dan memberikan manfaat bagi semua pihak umumnya dan bagi penulis khususnya. Aamiin.

Bandung, April 2021

Egi Noviansyah

NPM C1A150022 / 311150022

# DAFTAR ISI

[ABSTRAK vi](#_Toc82594324)

[*ABSTRACT* vii](#_Toc82594325)

[KATA PENGANTAR viii](#_Toc82594326)

[DAFTAR ISI ix](#_Toc82594327)

[DAFTAR GAMBAR xii](#_Toc82594328)

[DAFTAR TABEL xiv](#_Toc82594329)

[BAB I PENDAHULUAN 1](#_Toc82594330)

[1.1 Latar belakang 1](#_Toc82594331)

[1.2 Rumusan Masalah 2](#_Toc82594332)

[1.3 Batasan Masalah 2](#_Toc82594333)

[1.4 Tujuan Penelitian 3](#_Toc82594334)

[1.5 Metodologi Penelitian 3](#_Toc82594335)

[1.6 Sistematika Penulisan 3](#_Toc82594336)

[BAB II TINJAUAN PUSTAKA 5](#_Toc82594337)

[2.1 Tinjauan Penelitian 5](#_Toc82594338)

[2.1.1 Jurnal 1 5](#_Toc82594339)

[2.1.2 Jurnal 2 6](#_Toc82594340)

[2.1.3 Jurnal 3 6](#_Toc82594341)

[2.2 Dasar Teori 7](#_Toc82594342)

[2.2.1 *Inventory* 7](#_Toc82594343)

[2.2.2 Economic Order Quantity 8](#_Toc82594344)

[2.2.3 Reorder Point 9](#_Toc82594345)

[2.2.4 Rokok Elektrik 10](#_Toc82594346)

[2.2.5 Pengendalian Persediaan 10](#_Toc82594347)

[2.2.6 Sistem Distribusi 11](#_Toc82594348)

[2.2.7 *Model* *Driven* *Developement* (MDD) 12](#_Toc82594349)

[2.2.8 Rancang 16](#_Toc82594350)

[2.2.9 Bangun 16](#_Toc82594351)

[2.2.10 Aplikasi 16](#_Toc82594352)

[2.2.11 Website 17](#_Toc82594353)

[2.2.12 Visual Studio Code 18](#_Toc82594354)

[2.2.13 *Web Server* 20](#_Toc82594355)

[2.2.14 XAMPP 21](#_Toc82594356)

[2.2.15 HyperText Markup Language *(HTML)* 21](#_Toc82594357)

[2.2.16 Cascading Style Sheet *(CSS)* 22](#_Toc82594358)

[2.2.17 My Structured Query Language (MySQL) 23](#_Toc82594359)

[2.2.18 *Use Case* *Diagram* 25](#_Toc82594360)

[2.2.19 Activity Diagram 25](#_Toc82594361)

[2.2.20 *Hypertext Preprocessor* (*PHP*) 25](#_Toc82594362)

[2.2.21 *Unifed Modeling Language* (UML) 26](#_Toc82594363)

[2.2.22 Persamaan Dengan Riset Lain 30](#_Toc82594364)

[2.2.23 Perbedaan Dengan Riset Lain 30](#_Toc82594365)

[BAB III METODOLOGI PENELITIAN 31](#_Toc82594366)

[3.1 Kerangka Berfikir 31](#_Toc82594367)

[3.2 Deskripsi 32](#_Toc82594368)

[3.2.1 Investigasi Pendahuluan 32](#_Toc82594369)

[3.2.2 Analisa Masalah 35](#_Toc82594370)

[3.2.3 Analisis Kebutuhan 36](#_Toc82594371)

[3.2.4 Desain 37](#_Toc82594372)

[3.2.5 Kontruksi 40](#_Toc82594373)

[3.2.6 Implementasi 40](#_Toc82594374)

[3.2.7 Pembuatan Laporan 40](#_Toc82594375)

[BAB IV ANALISIS DAN PERANCANGAN 41](#_Toc82594376)

[4.1 Analisis 41](#_Toc82594377)

[4.1.1 Analisis Masalah 41](#_Toc82594378)

[4.1.2 Analisis Sistem 42](#_Toc82594379)

[4.1.3 Analisis Kebutuhan 44](#_Toc82594380)

[4.1.4 Analisis Pengguna 46](#_Toc82594381)

[4.1.5 User Interface 46](#_Toc82594382)

[4.1.6 Fitur-Fitur 46](#_Toc82594383)

[4.2 Perancangan Sistem 47](#_Toc82594384)

[4.2.1 *Usacase* *Diagram* 48](#_Toc82594385)

[4.2.2 Activity Diagram 50](#_Toc82594386)

[4.2.3 Class Diagram 58](#_Toc82594387)

[4.2.4 Perancangan *Database* 59](#_Toc82594388)

[4.2.5 Desain 63](#_Toc82594389)

BAB V [IMPLEMENTASI DAN PENGUJIAN 85](#_Toc82594391)

[5.1 Implementasi 85](#_Toc82594394)

[5.1.1 Implementasi Perangkat Keras dan Perangkat Lunak 85](#_Toc82594395)

[5.2 Pengujian Sismtem 86](#_Toc82594396)

BAB VI [KESIMPULAN DAN SARAN 102](#_Toc82594398)

[6.1 Kesimpulan 102](#_Toc82594399)

[6.2 Saran 102](#_Toc82594400)

[DAFTAR PUSTAKA 104](#_Toc82594401)

# DAFTAR GAMBAR

[Gambar 2. 1 Model Driven Developement 15](#_Toc78980854)

[Gambar 3. 1 MDD (Model Driven Developement) 32](#_Toc78980855)

[Gambar 3. 2 Buku nota penjualan dan persediaan barang 34](#_Toc78980856)

[Gambar 3. 3 Nota persediaan barang 35](#_Toc78980857)

[Gambar 4. 1 Flow map analisis system yang berjalan 43](#_Toc78980858)

[Gambar 4. 2 Flow map yang di usulkan 44](#_Toc78980859)

[Gambar 4. 3 Usecase Diagram 48](#_Toc78980860)

[Gambar 4. 4 Activity diagram log in 50](#_Toc78980861)

[Gambar 4. 5 Activity diagram olah data barang 51](#_Toc78980862)

[Gambar 4. 6 Activity Diagram Olah Data Barang Masuk 52](#_Toc78980863)

[Gambar 4. 7 Activity Diagram Olah data Barang Keluar 53](#_Toc78980864)

[Gambar 4. 8 Activity Diagram Olah Laporan Order 54](#_Toc78980865)

[Gambar 4. 9 Activity diagram Laporan Data Barang 55](#_Toc78980866)

[Gambar 4. 10 Activity diagram data barang masuk 55](#_Toc78980867)

[Gambar 4. 11 Activity diagram barang keluar 56](#_Toc78980868)

[Gambar 4. 12 Activity diagram order barang 57](#_Toc78980869)

[Gambar 4. 13 Activity diagram datra pelanggan 58](#_Toc78980870)

[Gambar 4. 14 Activity diagram log out 59](#_Toc78980871)

[Gambar 4. 15 Class diagram 59](#_Toc78980872)

[Gambar 4. 16 Mock up log in 64](#_Toc78980873)

[Gambar 4. 17 Mock up halaman login 65](#_Toc78980874)

[Gambar 4. 18 Mock up data pelanggan 66](#_Toc78980875)

[Gambar 4. 19 Mock up input data pelanggan 68](#_Toc78980876)

[Gambar 4. 20 Mock up data barang 70](#_Toc78980877)

[Gambar 4. 21Mock up input data barang 71](#_Toc78980878)

[Gambar 4. 22 Mock up data barang masuk 73](#_Toc78980879)

[Gambar 4. 23 Mock up input data barang masuk 75](#_Toc78980880)

[Gambar 4. 24 Mock up data barang keluar 77](#_Toc78980881)

[Gambar 4. 25 Mock up input data barang keluar 78](#_Toc78980882)

[Gambar 4. 26 Data user 80](#_Toc78980883)

[Gambar 4. 27 Halaman input user 81](#_Toc78980884)

[Gambar 4. 28 Halaman data rules 83](#_Toc78980885)

[Gambar 4. 29 Halaman input data rules 84](#_Toc78980886)

[Gambar 5. 1 Tampilan halaman log in 88](#_Toc78980887)

[Gambar 5. 2 Tampilan halaman dashboard 89](#_Toc78980888)

[Gambar 5. 3 Tampilan halaman data barang 91](#_Toc78980889)

[Gambar 5. 4 Tampilan halaman from input data barang 92](#_Toc78980890)

[Gambar 5. 5 Tampilan halaman data barang masuk 93](#_Toc78980891)

[Gambar 5. 6 Tampilan from isi data barang masuk 94](#_Toc78980892)

[Gambar 5. 7 Tampilan halaman data barang keluar 95](#_Toc78980893)

[Gambar 5. 8 Tampilan halaman from isi data barang keluar 96](#_Toc78980894)

[Gambar 5. 9 Tampilan halaman data user 97](#_Toc78980895)

[Gambar 5. 10 Tampilan form isi data user 98](#_Toc78980896)

[Gambar 5. 11 Tampilan halaman data pelanggan 99](#_Toc78980897)

[Gambar 5. 12 Tampilan halaman from isi data pelanggan 100](#_Toc78980898)

[Gambar 5. 13 Halaman data rules 101](#_Toc78980899)

[Gambar 5. 14 Implementasi metode EOQ dan ROP 102](#_Toc78980900)

[Gambar 5. 15 Penghitungan EOQ dan ROP 102](#_Toc78980901)

[Gambar 5. 16 Tampilan form isi data rules 103](#_Toc78980902)

# DAFTAR TABEL

[Tabel 2. 1 Element-element pada use case diagram 28](#_Toc78980815)

[Tabel 3. 1 Spesifikasi Minimum Hardware 37](#_Toc78980817)

[Tabel 4. 1 Kebutuhan sistem 45](#_Toc78980821)

[Tabel 4. 2 Spesifikasi perangkat keras 45](#_Toc78980822)

[Tabel 4. 3 Spesifikasi perangkat lunak 45](#_Toc78980823)

[Tabel 4. 4 Use case 49](#_Toc78980824)

[Tabel 4. 5 Data barang 60](#_Toc78980825)

[Tabel 4. 6 Data barang masuk 61](#_Toc78980826)

[Tabel 4. 7 Data barang keluar 61](#_Toc78980827)

[Tabel 4. 8 Data pelanggan 62](#_Toc78980828)

[Tabel 4. 9 Data user 63](#_Toc78980829)

[Tabel 4. 10 Data rules 63](#_Toc78980830)

[Tabel 4. 11 Keterangan mock up log in 64](#_Toc78980831)

[Tabel 4. 12 Keterangan mock up halaman login 65](#_Toc78980832)

[Tabel 4. 13 Keterangan mock up data pelanggan 67](#_Toc78980833)

[Tabel 4. 14 Keterangan mock up input data pelanggan 68](#_Toc78980834)

[Tabel 4. 15 Keterangan mock up data barang 70](#_Toc78980835)

[Tabel 4. 16 Keterangan mock up input data barang 72](#_Toc78980836)

[Tabel 4. 17 Keterangan mock up data barang masuk 73](#_Toc78980837)

[Tabel 4. 18 Keterangan mock up input data barang masuk 75](#_Toc78980838)

[Tabel 4. 19 Keterangan mock up data barang keluar 77](#_Toc78980839)

[Tabel 4. 20 Keterangan mock up input data barang keluar 79](#_Toc78980840)

[Tabel 4. 21 Keterangan data user 80](#_Toc78980841)

[Tabel 4. 22 Keterangan halaman input user 82](#_Toc78980842)

[Tabel 4. 23 Keterangan halaman data rules 83](#_Toc78980843)

[Tabel 4. 24 Keterangan halaman input data rules 85](#_Toc78980844)

[Tabel 5. 1 Spesifikasi perangkat keras 87](#_Toc78980845)

[Tabel 5. 2 Keterangan tampilan halaman log in 89](#_Toc78980846)

[Tabel 5. 3 Keterangan tampilan halaman dashboard 89](#_Toc78980847)

[Tabel 5. 4 Keterangan tampilan halaman data barang dan input data barang 92](#_Toc78980848)

[Tabel 5. 5 Keterangan data barang masuk dan from isi data barang masuk 94](#_Toc78980849)

[Tabel 5. 6 Keterangan halaman data barang keluar dan from isi data barang keluar 96](#_Toc78980850)

[Tabel 5. 7 Keterangan tampilan halaman data user dan tampilan form isi data user 98](#_Toc78980851)

[Tabel 5. 8 Keterangan tampilan halaman data pelangggan dan form isi data pelanggan 100](#_Toc78980852)

[Tabel 5. 9 Keterangan tampilan halaman data rules dan form isi data rules 104](#_Toc78980853)

# BAB I PENDAHULUAN

## Latar belakang

Salah satu jenis badan usaha untuk bisnis UMKM, harus memiliki *invetory* (persediaan barang). Tujuan utama dari penyimpanan persediaan adalah untuk memenuhi kebutuhan konsumen, di sisi lain pemilik usaha dituntut untuk mampu mengendalikan *inventory* (persediaan barang) untuk menghindari terjadinya *over stock,* sistem EOQ merupakan serangkaian kebijakan pengendalian untuk menentukan tingkat persediaan yang harus dijaga. Apabila jumlah persediaan terlalu besar (*overstock*) mengakibatkan timbulnya dana menganggur yang besar, juga menimbulkan resiko kerusakan barang yang lebih besar dan biaya penyimpanan yang tinggi. Namun jika persediaan terlalu sedikit mengakibatkan resiko terjadinya kekurangan persediaan (*stockout*) karena seringkali barang tidak dapat didatangkan secara mendadak dan sebesar yang dibutuhkan, yang menyebabkan terhentinya proses produksi, tertundanya penjualan, bahkan hilangnya pelanggan, sehingga pemilik usaha mendapatkan keuntungan yang optimal.

Industri adalah usaha atau kegiatan pengolahan barang mentah atau [setengah jadi](https://kamus.tokopedia.com/b/barang-setengah-jadi/) menjadi barang konsumsi yang memiliki nilai tambah untuk mendapatkan keuntungan bagi produsen. CV.Visya Pratama Industri adalah salah satu perusahaan yang bergerak dalam bidang dagang (*retailer*) yang menjual berbagai macam barang yang dibutuhkan oleh para pengguna vape (rokok elektrik) seperti menjual kapas, *coil* (kawat vape), pod, liquid dengan beragam rasa dan lainnya. Pemilik atau admin perusahaan bertugas mengelola segala aktivitas yang menyangkut pada produksi produk. Saat ini perusahaan menyediakan lebih dari 50 jenis produk dengan rata-rata produksi per-minggu mencapai 500 hingga 1000 pcs untuk sebagian jenis produk.

Dalam operasional pengendalian *inventory* (persediaan barang), untuk menentukan jumlah produk yang akan disediakan ulang, Pemilik perusahaan melihat dari penjualan perhari, sedangkan untuk menentukan kapan dilakukan pemesanan (*restock*), Pemilik perusahaan melihat rata-rata penjualan produk dari hari sebelumnya, jika jumlah produk mencapai jumlah rata-rata penjualan dan produk barang yang disediakan hampir habis terjual maka akan dilakukan penyediaan kembali. Dengan proses pengendalian persediaan yang dilakukan seperti saat ini, pemilik perusahaan mengalami berbagai macam permasalahan. Seperti lupa mencatat barang yang harus disediakan kembali sehingga dapat menimbulkan kosongnya salah satu produk yang seharusnya disediakan dan tidak mampu memenuhi kebutuhan konsumen.

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

## Rumusan Masalah

Adapun rumusan masalah yakni sebagai berikut:

1. Bagaimana menangani dan mendata proses pemesanan barang ?
2. Bagaiman membangun *inventory dengan* menggunakan sistem EOQ dan ROP yang dapat memberikan informasi yang cepat, tepat dan akurat ?

## Batasan Masalah

Dalam perancangan dan pembuatan sistem ini terdapat batasan masalah, antara lain:

1. Sistem ini di bangun dan di rancang meliputi proses pemesanan barang, persediaan barang, data pelanggan, serta transaksi pembayaran.
2. Sistem aplikasi dibuat dengan menggunkan metode EOQ dan ROP
3. Sistem aplikasi akan berjalan dalam jaringan internet dan hanya dipergunakan oleh perusahaan CV.Visya Pratama Industri.

## Tujuan Penelitian

Adapun tujuan lain dari rancang bangun aplikasi ini adalah:

1. Untuk mempermudah menangani dan mendata proses pemesanan barang.
2. Membangun sistem inventori dengan menggunakan metode EOQ dan ROP untuk mendapatkan kan informasi yang cepat, tepat dan akurat.

## Metodologi Penelitian

Metode yang digunakan untuk mengambangkan aplikasi ini menggunakan *Model Driven Development.* Yang terdiri dari tahap-tahap berikut ini:

1. Investigasi Pendahuluan
2. Analisis Masalah
3. Analisis Kebutuhan
4. Perancangan
5. Kontruksi
6. Implementasi

## 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 ladasan 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 dibutuhkan 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 inimembahas tentang kesimpulan yang diperoleh dari pembuatan sistem ini serta saran yang bertujuan untuk pengembangan sistem dimasayang akan datang.

# BAB II TINJAUAN PUSTAKA

## Tinjauan Penelitian

Tinjauan penelitian berisi ringkasan dari beberapa jurnal terkait judul dan objek penelitian yang diambil, berikut ini adalah beberapa referensi judul jurnal yang digunakan dalam proses penelitian, yaitu:

### Jurnal 1

**Judul :** **Sistem Informasi InventoryStok Barang Pada CV.Artha Palembang** (Rahmawati, April 2017)

**Abstrak :**  CV. Artha merupakan perusahaan yang bergerak di bidang distributor baja ringan, atap, rangka baja, gypsum dan aksesoris bagunan lainnya. Adapun masalah yang terjadi pada sistem yang sedang berjalan saat ini yaitu untuk pengolahan data barang masuk dan keluar, data pengiriman barang, data pelanggan masih dicatat kedalam buku besar sesuai dengan kwitansi atau surat jalan dari pelanggan atau *supplier* yang berisi berapa banyak barang yang masuk dan keluar untuk dikirim ke pelanggan dan pendataan persediaan stok barang hanya dicatat di selembar kertas kemudian baru disalin kembali ke komputer oleh bagian kantor, sehingga terkadang mengalami kesulitan dalam perhitungan barang dan untuk mendapatkan informasi stok barang mengalami kesulitan terkadang informasi yang diberikan tidak sesuai dengan ketersediaan barang yang ada. Tujuan penelitian ini adalah membuat Sistem Informasi *Inventory* Stok Barang, yang dapat mengelola barang masuk dan keluar, stok barang, data pelanggan, pengiriman, transaksi pemesanan dengan menggunakan metode *Prototype* dan dengan menggunakan *Data Flow Diagram* (DFD) untuk pemodelan terstruktur. Sistem Informasi yang telah dihasilkan pada perancangan ini dapat mengelola data barang masuk dan keluar, data stok barang, data pelanggan, data pengiriman, data transaksi pemesanan, serta adanya grafik visual presentase penjualan dan barang terlaris serta pemesanan barang yang bisa dilakukan secara *online.*

### Jurnal 2

**Judul : Perancangan Aplikasi Inventory Barang Materials dan Product (**Gita Ayu Syafarina, 2016)

**Abstrak :** Sistem informasi merupakan hal yang penting dalam suatu organisasi atau perusahaan. Dengan adanya sistem informasi, organisasi atau perusahaan dapat tenjamin kualitas informasi yang disajikan dan dapat mengambil keputusan berdasarkan informasi tersebut.

Amanta Mitra Mandiri (AMM) adalah salah satu perusahaan layanan telekomunikasi terbesar didunia. Amanta Mitra Mandiri (AMM) dibentuk dan untuk merealisasikan pengoperasian layanan 2G dan 3G di Indonesia di bawah bendera 3. mempunyai dua produk, yaitu produk yang dijual dan produk yang tidak dijual. Produk yang dijual seperti kartu perdana dan voucher, sedangkan produk yang tidak dijual, yakni bahan-bahan promosi atau biasa disebut Material () mempunyai bentuk-bentuk yang beragam.

Dengan semakin bertambahnya jumlah barang dagangan muncul permasalahan yaitu kesulitan untuk mendapatkan informasi Inventory barang yang cepat, tepat dan akurat. Melihat dari permasalan yang dihadapi, Penulis mencoba membuat suatu sistem Inventory barang yang pengolahannya sudah komputerisasi. Dengan adanya sistem informasi yang terkomputerisasi ini diharapkan perusahaan tersebut bisa bersaing dengan perusahaan yang lain dalam meningkatkan kualitas pelayanannya.

### Jurnal 3

**Judul : Rancang dan Bangun Sistem Informasi Inventory Barang Berbasis Web (Studi Kasus PT. Nusantara Sejahtera Raya)** (Maulana Hasanudin, 2018)

**Abstrak :**  Dengan semakin berkembangnya jaman, PT. Nusantara Sejahtera Raya menggunakan alat-alat komputerisasi di semua jaringan bioskop untuk menunjang sistem yang sedang berjalan. kekurangan yang ada pada sistem order barang yang sedang berjalan sekarang yaitu dalam pencatatan stok barang masih dihitung manual membuat celah lupa mencatat sehingga tidak seimbang antara jumlah stok barang di catatan dengan fisik yang ada, bioskop tidak mengetahui ketersediaan barang yang ada di gudang, untuk luar kota order barang masih menggunakan sistem fax ke kantor pusat terkadang fax tersebut hilang dan order tersebut tidak di proses, tidak adanya pencatatan data order barang. Tujuan penelitian ini adalah membuat sistem informasi inventori barang, yang dapat mengelola barang masuk dan keluar, stok barang, transaksi pemesanan barang dengan menggunakan *Data Flow Diagram* (DFD) untuk pemodelan terstruktur.

Sistem Informasi yang telah dihasilkan pada perancangan ini dapat mengelola data barang masuk dan keluar, data stok barang, pemesanan barang dalam satu aplikasi berbasis *web*.

## Dasar Teori

Bab ini berisi uraian tentang landasan teori yang diambil dan dasar teori yang dipakai dalam pada penelitian ini. Landasan teori mengambil dari berbagai jurnal penelitian yang bertujuan untuk melakukan perbandingan dengan apa yang akan dilakukan dalam penelitian nantinya, sedangkan untuk dasar teori lebih memahami aspek apa saja yang akan dibahas dalam membantu penelitian.

### *Inventory*

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

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

* 1. Sebagai antisipasi kemungkinan terjadinya keterlambatan kedatangan barang ataupun barang-barang yang dibutuhkan perusahaan dalam aktifitas usahanya.
  2. Sebagai antisipasi kemungkinan terjadinya cacat barang pada yang dipesan sehingga harus di retur kembali ke perusahaan asal.
  3. Sebagai antisipasi terjadinya kelangkaan barang-barang tertentu yang tidak dapat diproduksi sepanjang musim.
  4. Untuk mempertahankan dan menjaga aktifitas oprasional perusahaan sekaligus menjamin keberlangsungan aktifitas produksi dalam perusahaan.
  5. Untuk mengantisipasi terjadiya kehabisan barang yang seharusnya disediakan.

### 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 di beli maupun yang dibuat sendiri, dan banyak digunakan sampai saat ini karena penggunaannya relatif mudah. Model ini mampu untuk menjawab pertanyaan tentang kapan pemesanan atau pembelian harus dilakukan dan berapa banyak jumlah yang harus di pesan agar biaya total (penjumlahan antara biaya pemesanan dengan biaya penyimpanan) menjadi minimum.

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

**EOQ = {(2.D x S) / H}**

Keterangan :

D : Kebutuhan barang per periode

Adalah jumlah barang yang dibutuhkan dalam satu periode (hari, minggu, bulan, tahun).

S : Biaya order pesanan

Adalah biaya tranfortasi, telepon dan lain-lain yang dikeluarkan ketika hendak melakukan penyediaan kembali (*Re*-*Stock*).

H : Biaya penyimpanan

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

### Reorder Point

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

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

1. Safety Stock adalah persediaan minimal yang ada dalam perusahaan.
2. Kebutuhan barang perperiode.

Untuk menghitung *Reorder* *Point*, dirumuskan sebagai berikut :

**Rumus ROP = Safety stock + (D/ 365 )**

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).

### Rokok Elektrik

Rokok elektrik adalah salah satu jenis dari penghantar nikotin elektronik. Rokok jenis ini dirancang untuk membantu pecandu rokok tembakau mulai berhenti merokok. Dengan beralih dari rokok tembakau ke rokok elektrik, secara perlahan mereka belajar untuk berhenti merokok.

Rokok jenis ini terdapat dalam berbagai bentuk dan ukuran, tetapi terdapat tiga komponen utama dalam rokok elektrik, yaitu baterai, elemen pemanas, dan tabung yang berisi cairan (*cartridge*). Cairan dalam tabung ini mengandung nikotin, *propilen glikol* atau *gliserin*, serta penambah rasa, seperti rasa buah-buahan dan cokelat. Beberapa rokok elektrik memiliki baterai dan *cartridge* yang dapat di isi ulang.

Rokok elektrik bekerja dengan cara memanaskan cairan yang ada dalam tabung dan kemudian menghasilkan uap seperti asap yang umumnya mengandung berbagai zat kimia. Pengguna mengisap zat kimia ini langsung dari corongnya.

### Pengendalian Persediaan

Pengendalian persediaan adalah suatu usaha yang dilakukan suatu perusahan untuk menjaga stabilitas produksi, sehinga dapat memenuhi kebutuhan kosumen. Dengan dilakukannya pengendalian persediaan jumlah produk yang disediakan akan stabil sehingga dapat meghemat biaya produksi dan memaksimalkan keuntungan perusahan.

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

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

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

### Sistem Distribusi

Sistem distribusi (Distributed System) adalah suatu system jaringan komputer yang saling berhubungan demikian rupa sehingga seolah-olah menjadi sebuah komputer yang digunakan oleh suatu pengguna.

Dengan terintegrasinya berbagai komputer sehingga seolah-olah merupakan satu komputer, dengan sendirinya akan menciptakan optimalisasi sumberdaya pengolahan data.

System distribusi menekankan prinsip transparansi jaringan, yaitu bahwa pengguna tidak menyadari bahwa sistem tersebut sebenarnya tersebar di berbagai lokasi.

System distribuasi liquid dilakukan dengan cara yang pertama melakukan penyetokan persediaan barang yang meliputi bahan baku dan hasil jadi serta mendistribusikannya ke berbagai wilayah di seluruh Indonesia.

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

Teknik pengembangan berbasis model (MDD) menekankan gambar model untuk membantu memvisualisasikan dan menganalisis masalah, mendefinisikan kebutuhan bisnis, dan merancang sistem informasi. Analisis dan desain sistem terstruktur - berpusat pada proses Teknik informasi (IE) - berpusat pada data Analisis dan desain berorientasi obyek (OOAD) - terpusat pada objek (integrasi data dan masalah proses) *Rute model driven development*. Wijaya. (2009).”*Model Driven Development”.*

Tahapan yang dipakai pada penelitian adalah sebagai berikut:

1. *Preliminary investigation* (investigasi awal)
2. *Problem analysis* (Analsis masalah)
3. *Requirements analysis* (Analisis Kebutuhan)
4. *Design* (Desain)
5. *Construction* (Kontruksi)
6. *Implementation* (implementasi)
7. *Preliminary Investigation Phase*

Tahap ini merupakan tahap awal dari pengembangan sistem. Fase ini berisikan investigasi awal ketika ingin merancang sebuah sistem, seperti wawancara, tinjauan langsung dan mempelajari dokumen perusahaan.

Tujuan dari tahap ini ialah menjawab pertanyaan mengenai apakah proyek ini cukup berharga untuk diperhatikan. Untuk menjawab pertanyaan ini perlu didefinisikan terlebih dahulu masalah, kesempatan, dan resiko-resiko dalam melanjutkan proyek. Kerangka kerja PIECES dapat digunakan untuk menjawab pertanyaan ini namun hasilnya bukanlah solusi permasalahan melainkan kategori-kategori masalah (dengan asumsi bahwa proyek ini berharga untuk diperhatikan) menetapkan rincian proyek yang akan menetapkan lingkup, kebutuhan dan hambatan proyek, anggota proyek, biaya, dan jadwal.

1. *Problem Analysis Phase*

Problem Analysis ialah menganalisa masalah-masalah yang terdapat di lapangan. Tahap ini merupakan pengembangan dari tahap pertama. Pada tahap ini dilakukan analisis terhadap sistem yang telah ada saat itu. Tahap ini memberikan pemahaman yang lebih dalam bagi tim proyek mengenai permasalahan yang dihadapi. Analisis ini dilakukan untuk menjawab pertanyaan apakah keuntungan yang diperoleh setelah pemecahan masalah lebih besar daripada biaya yang dikeluarkan.

Input utama dari tahap ini adalah project charter dari tahap sebelumnya. Informasi yang digunakan dalam memperlajari permasalahan yang dihadapi adalah fakta-fakta yang terdapat dalam sistem, masalah, akibat, penyebab dari permasalahan, dan spesialis IT yang merancang sistem yang telah ada.

*Output* yang dihasilkan adalah system *improvement objectives* yang menyatakan kriteria bisnis yang akan digunakana untuk mengevaluasi sistem. Kadang-kadang dilakukan repesentasi pada tahap ini.

1. *Requirement Analysis Phase*

Requirement Analysis ialah melakukan analisa terhadap kebutuhan perusahaan. Pekerjaan pada tahap ini adalah mendefinisikan apa saja yang perlu di lakukan oleh sistem, apa yang di butuhkan dan diinginkan oleh pengguna dari sistem baru.

Tahap ini memerlukan perhatian yang besar karena jika terjadi kesalahan dalam menerjemahkan kebutuhan dan keinginan pengguna sistem maka dapat mengakibatkan adanya rasa tidak puas pada sistem final dan perlu diadakan modifikasi yang tentunya akan kembali mengeluarkan biaya.

1. Input dari tahap ini adalah sistem *improvement* *objectives* yang dihasilkan pada tahap sebelumnya. Pada tahap ini, tim akan mengumpulkan dan mendiskusikan kebutuhan dan prioritas berdasarkan informasi yang di peroleh dari kuesioner, wawancara, dan rapat-rapat. Tantangannya adalah untuk memvalidasi semua kebutuhan informasi.
2. Output yang di hasilkan dari tahap ini adalah business requirement statement. Tahap ini pun merupakan tahap yang penting karena dapat menimbulkan ketidakpuasan dari pengguna sistem yang merasa kebutuhannya tidak terpenuhi. Tim proyek harus dapat membedakan antara apa yang dibutuhkan oleh pengguna dan bagaiman sebaiknya sistem yang baru bekerja.
3. *Desain Phase*

Setelah diperoleh proposal sistem yang disetujui, maka dapat mulai dilakukan proses desain dari sistem target. Tujuan dari tahap ini adalah untuk mentransformasikan *business requirement statement* menjadi spesifikasi desain untuk proses *construksi*. Dengan kata lain, tahap desain menyatakan bagaimana teknologi akan digunakan dalam sistem yang baru. Tahap ini memerlukan ide dan opini dari pengguna, vendor, dan spesialis IT.

Pada akhir tahap ini masih terdapat beberapa alternatif keputusan mengenai proyek walaupun pembatalan proyek jarang dilakukan pada tahap ini (kecuali benar-benar over budget atau sangat terlambat dari jadwal). Perubahan lingkup menjadi lebih kecil masih dapat terjadi. Selain itu, mungkin juga terjadi perubahaan ulang jadwal untuk menghasilkan solusi yang lebih lengkap.

1. *Construction Phase*

*Construction* *Phase* ialah tahapan melaksanakan pengujian pada komponen sistem secara individu dan sistem secara keseluruhan.

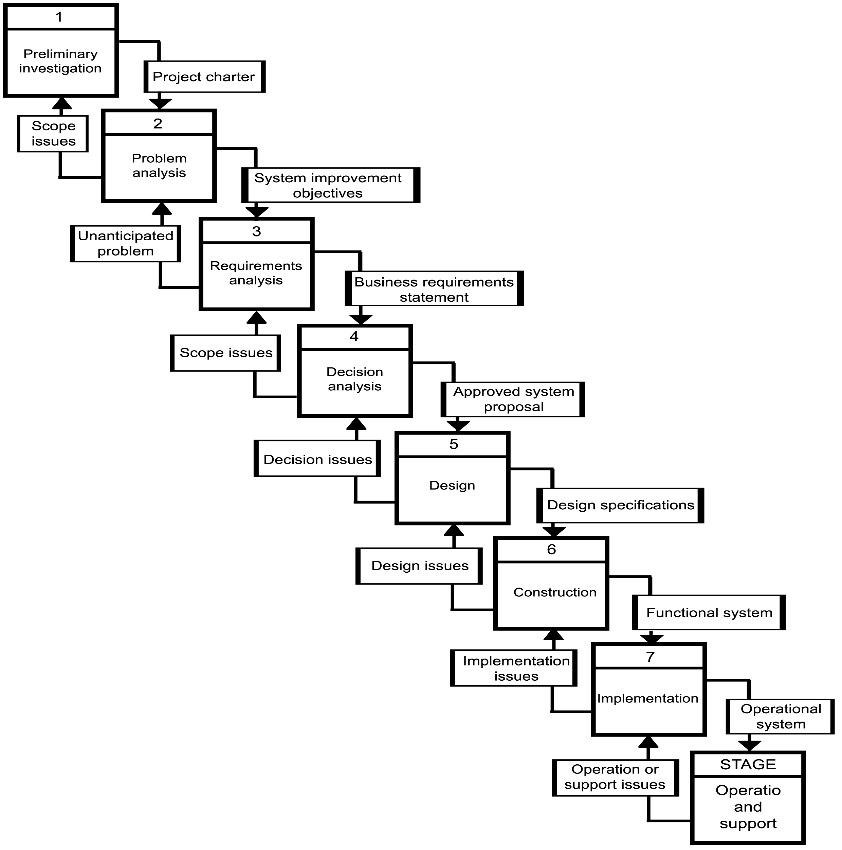
Tujuan dari tahap ini adalah :

1. Membangun dan menguji sistem yang memenuhi *business requirement* dan spesifikasi desain.
2. Mengimplementasikan penghubung antara sitem baru dan sistem lama, termasuk instalasi dari software yang di beli atau di sewa
3. Pada tahap ini dilakukan konstruksi basis data, program aplikasi, dan penghubung antara sistem dan pengguna. Beberapa dari komponen ini telah ada sebelumnya.

Setelah dilakukan pengujian, maka sistem dapat mulai diimplementasikan.

1. *Implementtion Phase*

*Implementtion Phase* adalah tahap pengimlementasian berbagai hal yang telah didapat dari tahap-tahap sebelumnya apabila pada tahapan ini diketemukan masih terdapat kekurangan maka dapat kembali ke tahapan sebelumnya, hal tersebut berlaku sampai proyek dapat terselesaikan. Berikut adalah gambar implemetasi *model* *driven* *development*.



Gambar 2. Model Driven Developement

### 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 dipahmi sehingga mudah digunakan.

Perancangan adalah sebuah proses untuk mendefinisikan sesuatu yang akan dikerjakan dengan menggunakan teknik yang bervariasi serta di dalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan dialami dalam proses pengerjan-nya.

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

### Bangun

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

Jadi dapat disimpulkan 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 demikian pengertian rancang bagun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan *system* tersebut atau memperbaiki *system* yang sudah ada sebelumnya.

### Aplikasi

Aplikasi adalah program siap pakai yang dapat digunakan 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 ditulis dalam suatu bahasa pemograman dan dipergunakan untuk menyelesaikan masalah tertentu (Robi Muhamad, 209).

### Website

Secara *basic* *website* digunakan untuk publikasi informasi. Adapun informasi yang akan disediakan adalah beraneka ragam dari *profil* pribadi hingga *copany profil*. *Website* sering juga disebut *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 diakses 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 halamanya tidak berubah, biasanya untuk melakukan perubahan dilakukan secara manual dengan mengubah kode. *Website* statis informasinya merupakan informasi satu arah, yakni hanya berasal dari pemilik *softwarenya* saja, hanya bisa di *update* oleh pemiliknya saja, contoh dari pengertian *website* statis ini adalah *profil* perusahaan.

* *Website* Dinamis

Merupkan *web* yang halamanya selalu *update,* biasanya terdapat halman *backend* (halaman administrator) yang digunakan untuk menambah atau merubah konten. *Web* diamis mebutuhkan *database* untuk menyimpan data. Web dinamis mepunyai arus informasi dua arah, yakni berasal dari pengguna dan pemilik, sehinga peng-*update*-an dapat dilakukan oleh penguna dan pemilik *website*. Contoh dari pegertian *website* dinamis ini yaitu *friendster, Multiply, Facebook* dan lain-lain.

### Visual Studio Code

Visual studio Code merupakan aplikasi *cross platform* yang dapat digunakan berbagai sitem operasi seperti windows, Linux, dan Mac OS. VS Code termasuk software yang ringan namun kuat editor sumbernya dengan deskop. Menggunakan berbagai macam bahasa pemprograman seperti Java, JavaSkrip, Go, C++, dan masih banyak yang lainnya. Komponen dari Visual Studio juga sama seperti yang digunakan di Azura DevOps. Visual Studio memiliki lintas platform kode editor yang ringan, dapat digunakan oleh siapa saja untuk membuat atau membangun aplikasi web.

1. Komponen Pada Visual Studio Code
2. *Customize* : Digunakan untuk menambahkan ekstensi bahasa pemprograman. konfigurasi  dan kustomisasi template dengan menambahkan ekstensi bahasa pemprograman maka kita tidak perlu selalu mengingat fungsi bahasa
3. Command Palette: *command palette* menyediakan banyak akses perintah, kita bisa memberikan perintah editor membuka file, mencari file dan sebagainya dengan cepat dan mudah.untuk membuka *commad palette* bisa dengan tekan Ctrl+Shift+p.
4. *Integreted* *Termina*l: Integ*rital Terminal* digunakan untuk mengeksekusi skrip di editor. Kita bisa mengeksekusi skrip editor secara langsung di terminal tanpa harus membuka terminal tambahan , ini adalah salah satu kelebihan dari visual studio code.
5. *Extention* : *extention* adalah fungsi tambahan dalam ynag berfungsi untuk memperluas kemampuan dari editor yang dapat membantu *developer* dalam melakukan *programing*.
6. *Search* : Fitur *search Visual Studio Code* juga sangat cepatnya kemudahan yang diberikan selain kecepatan *query* pencarian data juga dia akan mencari sampai ke level  kontennya
7. *Grid Editor Layout* : Kita juga mudah dalam menejemen *layout visual studio code*, kita dapat dengan mudah mengatur grup editor dalam tata letak apapun baik secara *vertikal* maupun *horizontal*.
8. *Color Themes* : *Color Themes* digunakan untuk memodifikasi warna dalam antarmuka Visual studio Code agar sesuai dengan selera  yang diinginkan , caranya dengan pilih File >Prefences>Color Theme, lalu geser cursor keatas dan kebawah untuk memilih tema yang diinginkan.
9. *Cloud Enviroment* : Kita juga bisa melakukan sesuatu di lingkungan *cloud* melalui *Visual Studio Code* seperti membuat *databtse*, melakukan perintah , *insert, update, delete*, dan sebagainya di *cloud*.
10. Macam-macam *Extention* pada *Visual studio Code*
11. *Live server*: memungkinkan kita untuk memuat ulang halaman web secara  otomatis ketika kamu mengubah di *Visual Studio Code*.
12. *GitLens* : membantu kita untuk memvisualisasikan kepengarangan kode sekilas memalui anotasitanpa hambatan.
13. Prettier (*Code Formatter*) : Digunakan untuk menata format  *code prettier* mendorong gaya penulisan code yang konsisten dengan memperasing kode anda dan memencetnya kembali dengan aturannya sendiri.
14. *Auto Rename Tag* : *Auto rename tag* membantu kita ketika ingin mengganti sebuah tag dan juga untuk mengganti tag pembuka dan tag penutub bersamaan pada HTML dan XML.
15. *Beautify* : *Beautify* membantu anda dalam formatting codingan kita, kita tidak perlu merapikan codingan kita secara manual cukup dengan *extention* ini akan otomatis terformat.
16. *Open In Browser* : Digunakan untuk membuat langsung *project*  anda pada browser.
17. CSS *Peak* : berguna untuk melihant sebuat element HTML , yang terkena CSS , CSS *Peak* juga sangat mempermudah bagi *web designer*.
18. *Indent Rainow* : Digunakan untuk untuk membantu kita memberikan warna dan juga mengelompokkan warnai sesuai tag awal dari struktur dari pemprograman yang kita buat .
19. SQL *server* : adalah *extention* dimana kita adalah seorang *database engier SQL server* , *extention* ini juga mempermudah  anda bekerja dalam *Visual studio Code*.
20. Kelebihan *Visual Studio Code*
21. Mudah untuk mengelola *extention*
22. Memiliki *extention* yang banyak
23. Kontribusi tampilan
24. Dukungan bahasa
25. Text editor gratis
26. Dapat membuat *Snippet* sendiri

### *Web Server*

Web Server adalah suatu suatu program komputer yang mempunyai tanggung jawab atau tugas menerima permintaan HTTP dari komputer Klien, yang dikenal dengan nama web browser dan melayani mereka dengan menyediakan respon HTTP berupa konten data, biasanya berupa halaman web yang terdiri dari dokumen HTML dan objek terkait seperti gambar dan lain-lain.

Untuk dapat menjalankan PHP yang disertai dengan MySQL dapat digunakan dua jenis web server yaitu Online mode dan Offline mode. Pada Online mode, selain komputer harus mempersiapkan domain dan hosting serta koneksi internet yang memadai untuk mengelolanya sehingga harus keluar biaya ekstra terlebih dahulu. Sedangkan pada Offline mode yang anda persiapkan cukup komputer dan beberapa software untuk membuat web server lokal. Pada kedua cara kedua ini komputer PC akan dibuat menjadi web server lokasi atau localhost sebelum benar-benar meng-upload-nya ke web server Internet.

Ada beberapa jenis software untuk membangun web server lokal atau localhost yang support sistem Windows diantaranya adalah WampServer, Appserv, XAMPP, PHP Triad atau Vertrig.

### XAMPP

XAMPP adalah sebuah paket perangkat lunak (software) komputer yang sistem penamaannya diambil dari akronim kata Apache, MySQL (dulu) / MariaDB (sekarang), PHP, dan Perl. Sementara imbuhan huruf “X” yang terdapat pada awal kata berasal dari istilah cross platform sebagai simbol bahwa aplikasi ini bisa dijalankan di empat sistem operasi berbeda, seperti OS Linux, OS Windows, Mac OS, dan juga Solaris.

Sejarah mencatat, software XAMPP pertama kali dikembangkan oleh tim proyek bernama Apache Friends dan sampai saat ini sudah masuk dalam rilis versi **7.3.9** yang bisa didapatkan secara gratis dengan label GNU (General Public License).

### HyperText Markup Language *(HTML)*

Hypertext Markup Language (HTML) adalah bahasa standard yang digunakan untuk menampilkan halaman web. Yang bisa dilakukan dengan HTML yaitu :

1. Mengatur tampilan dari halaman web dan isinya.
2. Membuat tabel dalam halaman web.
3. Mempublikasikan halaman web secara online.
4. Membuat form yang bisa digunakan untuk menangani registrasi dan transaksi via web.
5. Menambahkan objek-objek seperti citra, audio, vidio, animasi, java applet, dalam halaman web.
6. Menampilkan area gambar (canvas) di browser. (Priyanto Hidayatuloh, Jauhari Khairul Kawistaran. 2017:15)

HTML atau Hyper Text Markup Language merupaka salah satu bahasa pemrograman yang digunakan di dalam pembuatan aplikasi dan layanan berbasis web. Bahasa di awal-awal website ada, HTML merupakan salah satu bahasa populer yang digunakan (selain Perl). HTML memiliki peran didalam kesuksesan Search Eginer Optimization (SEO) pada website anda. Melalui HTML, seorang Web Master atau sekaligus pemilik website bersangkutan, dapat menambahkan Meta Teg bahkan konten di dalam halaman website untuk artikel, tautan ke URL suatu file, penambahan gambar, video, dan audio, pada halaman website.

Meningat bahwa hingga saat ini HTML (Hyper Text Markup Language) masih menjadi salah satu bahasa dasar yang umum di gunakan di dalam pembuatan dan perbaikan aplikasi dan layanan berbasis web, termasuk juga di dalam penyediaan dukungan SEO (Search Engine Optimization), maka keberadaan HTML (Hyper Text Markup Language) masih dipandang penting dan perlu di dalam mendukung ke suksesan SEO. Saat ini HTML telah menyediakan HTML 5, yang dijadikan patokan di dalam penyediaan layanan interaktif berbasis web, yang berangsur-angsur mulai menggantikan teknologi animasi Flash.(Putu Agus Eka Pratama.2015:71-72)

### Cascading Style Sheet *(CSS)*

CSS (Cascading Style Sheet) adalah salah satu bahasa desain web (style sheet language) yang mengontrol format tampilan sebuah halaman web yang ditulis dengan menggunakan penanda markup laguage. Biasanya CSS digunakan untuk mendesain sebuah halaman HTML dan XHTML, tetapi sekarang CSS bisa diaplikasikan untuk segala dokumenXML, termasuk SVG dan XUL bahkan ANDROID.

CSS dibuat untuk memisahkan konten utama dengan tampilan dokumen yang meliputi layout, warna dan jenis huruf (font). Pemisahan ini dapat meningkatkann daya akses konten pada web, menyediakan lebih banyak fleksibilitas dan kontrol dalam spesifikasi dari sebuah karakteristik dari sebuah tampilan, memungkinkan untuk membagi halaman untuk sebuah formating dan mengurangi kerumitan dalam penulisan kode dan struktur dari konten, contohnya teknik tableless pada desain web.

CSS juga memungkinkan sebuah halaman untuk ditampilkan dalam berbagai style dengan menggunakan metode pembawaan yang berbeda pula, seperti on-screen, in-print, by voice, dan lain-lain. Sementara itu, pemilik konten web bisa menentukan link yang menghubungkan konten dengan file CSS.

### My Structured Query Language (MySQL)

MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya sangat cepat, *multi user* serta menggunakan peintah dasar SQL (*Structured Query Language*). MySQL merupakan dua bentuk lisensi, yaitu *FreeSoftware* dan *Shareware*. MySQL yang biasa kita gunakan adalah MySQL *FreeSoftware* yang berada di bawah 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 di rintis 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 di gunakan sebagai *client* mupun *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* ).

1. **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 bertupuk 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 di akses 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 uniq(*Unique*).

• MySQL memliki kecepatan dalam pembuatan table maupun pengupdatean table.

### *Use Case* *Diagram*

Use case atau diagram use case merupakan pemodelan untuk melakkan (behavior) sistem informasi yang akan dibuat. Use Case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan di buat. Secara kasar, use case digunakan untuk mengtahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.(Rosa A.S, M.Shalahuddin, 2013:155)

### Activity Diagram

Diagram Activity atau activity diagram merupakan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan di sini bahwa diagram aktivity menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.(Rosa A.S, M.Shalahuddin, 2013:16)

### *Hypertext Preprocessor* (*PHP*)

Bahasa pemograman PHP merupakan bahasa pemograman yang cukup populer dan banyak digunakan oleh para *programmer* di dunia. PHP atau yang memiliki kepanjangan *Hypertext Preprocessor*, merupakan suatu bahasa pemograman yang difungsikan untuk membangun suatu website dinamis, PHP menyatu degan kode HTML. HTML digunakan sebagai pembangun atau pondasi dari kerangka *layout web,* sedangkan PHP difungsikan sebagai prosesnya, sehingga dengan adanya PHP tersebut sebuah web akan mudah di *maintenence*, (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* sehinga dapat dipakai secara gratis, dan mampu lintas *platform*, yaitu dapat berjalan pada sistem operasi *widow* maupun *linux.*

### *Unifed 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:

* 1. *Structural things*

*Structural things* adalah kata benda dari model UML

* 1. *Behavioral things*

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

* 1. *Grouping things*

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

* 1. *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:

1. *Dependency*

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

1. *Association*

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

1. *Generalization*

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

1. *Realization*

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

1. *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 digunakan:

1. *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 di panggil dari luar class yang bersangkutan
      * *Protected*, hanya dapat di panggil oleh class yang bersangkutan dan anak-anak yang mewarisinya
      * *Public*, dapat di panggil oleh siapa saja

1. *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. Element-element pada use case diagram

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

1. *Activity diagram*

*Activity Diagram* adalah diagram yang menggambarkan *worlflow* (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*: Menunjukan awal aktivitas dimulai.
* *Final Activity* atau *final state*: Menunjukan 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 di pecah 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 tangung jawab obyek-obyek yang melakukan aktivitas.
* *Transition*: Menunjukan aktivitas selanjutnya setelah aktivitas sebelumnya.

### Persamaan Dengan Riset Lain

Adapun persamaan penelitia ini dengan penelitian nomor 3 adalah sama sama mengunakan metode MDD untuk metode pengendalian persediaan, dan persamaan dengan penelitian nomor 1 dan 2 adalah sama-sama rancang bangun aplikasi sistem informasi inventory

### Perbedaan Dengan Riset Lain

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

# 

# BAB III METODOLOGI PENELITIAN

## Kerangka Berfikir

Kerangka fikir adalah suatu metode yang digunakan oleh penyusun untuk dapat mengetahui masalah atau kekurangan yang berada di CV.Visya Pratama Industri untuk membangun suatu aplikasi system informasi inventory E-Liquid, kerangka pikir pada implementasi system ini menggunakan metode Model Driven Development (MDD). Berikut gambaran kerangka pemikirannya :

#### 

Gambar 3. Kerangka berfikir

## 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 CV.Visya Pratama Industri dengan studi kasus di Kp. Butul RT 05/RW 01. Desa Cipejeuh. Kecamatan Pacet. Kab. Bandung kode pos 40385**,** lalu menemui pemilik perusahaan untuk meminta izin melakukan penelitian dan wawancara perihal pengelolaan data barang pada perusahaan yang dimaksud dan wawancara dimulai dan diwakili oleh sekertarisnya.

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 (*Re*-*stock*) perusahaan atau karyawan mengecek satu persatu barang yang akan disediakan kembali.

1. Apakah pengumpulan nota belanja sudah tersusun rapih?

Jawaban :

Belum, karena ketika prusahaan atau karyawan lupa menyimpan nota belanja, perusahaan tidak dapat mengetahui perbandingan harga dari hari sebelumnya (apakah barang yang dibeli harganya naik atau turun).

1. Bagaimana cara perusahaan atau karyawan mencari data harga barang yang akan dijual sebelum dibuatkan aplikasi untuk membantu pencarian?

Jawaban :

Pihak perusahaan atau karyawan melihat melihat kembali dari nota belanja yang ditulis secara manual.

1. Apa permasalahan yang saat ini ada pada pengolahan data barang di perusahaan?

Jawaban :

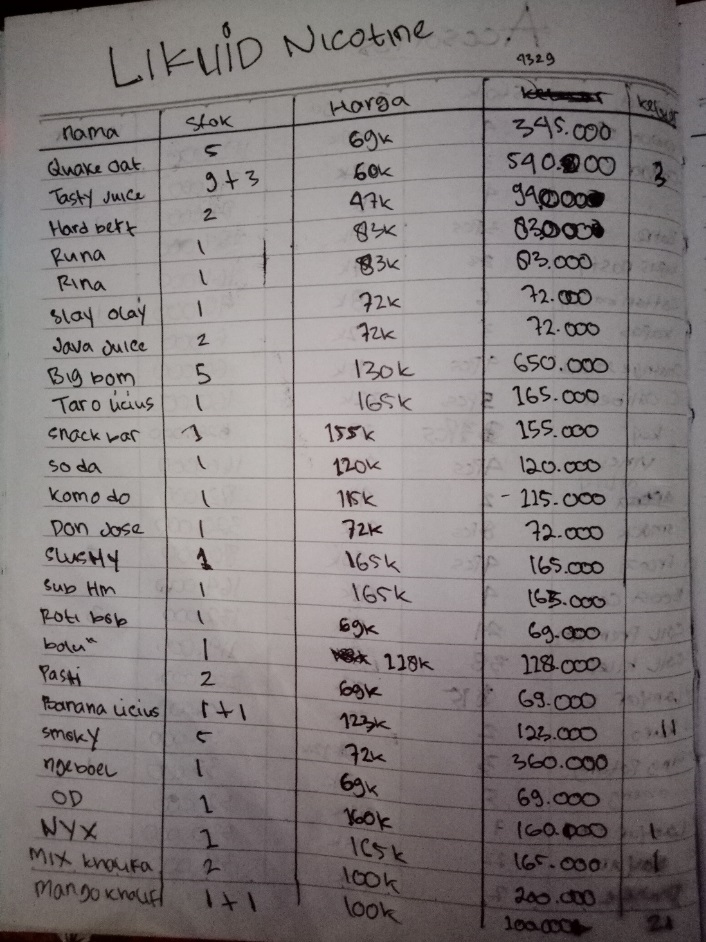
Permasalahan yang ada saat ini adalah : ketika nota belanja rusak sobek atau terkena air dan lupa menyimpan nota belanja persediaan gudang perusahaan tidak dapat mengetahui perbandingan harga / harga jual, ketika lupa mencatat barang barang yang seharusnya disediakan kembali perusahaan tidak mampu memenuhi kebutuhan konumen dan kurangnya efisien-nya penyimpanan nota belanja.

1. Bagaimana caranya mengetahui pemasukan / Keuntungan yang didapatkan pada perusahaan?

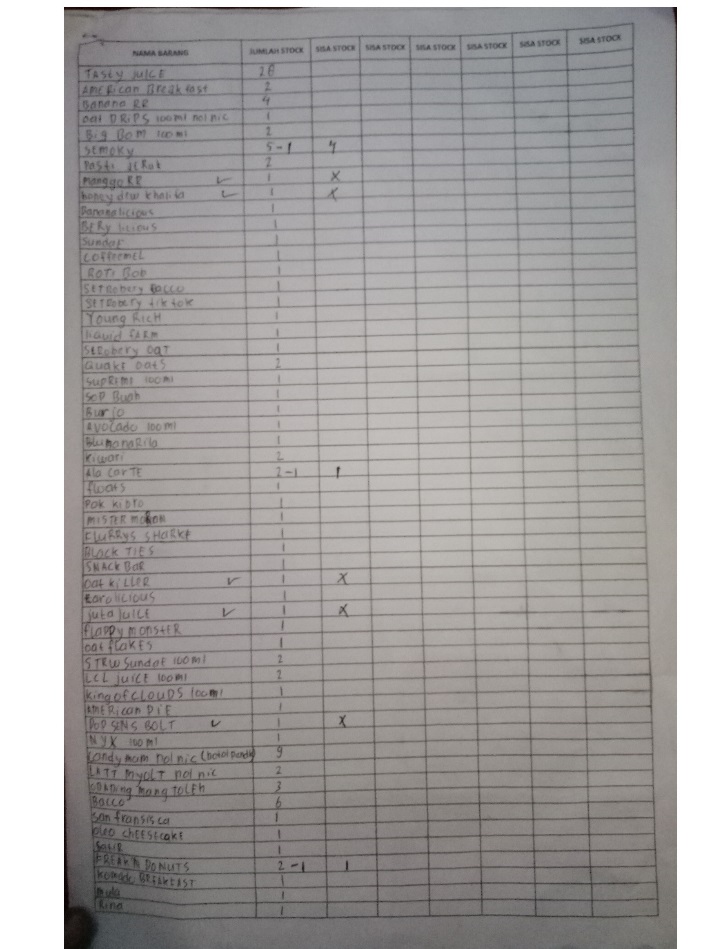
Jawaban :

Cara mengetahuinya adalah dengan cara menghitung uang yang ada di brankas setelah semua pendistribusian atau penjualan barang sudah selsai dan perusahaan tutup.

Gambar Buku Nota Penjualan dan Persediaan Barang :



Gambar 3. Buku nota penjualan dan persediaan barang



Gambar 3. Nota persediaan barang

### 3.2.2 Analisa Masalah

Berikut adalah hasil analisis masalah yang ada pada Perusahaan :

1. Pencatatan nota persediaan barang masih manual.
2. Ketika perusahaan lupa mencatatat barang yang dibutuhkan oleh *reseler* atau distributor, maka perusahaan tidak dapat memenuhi kebutuhan konsumen.
3. Penyimpanan nota belaja atau persediaan barang masih manual (Ditumpuk di satu tempat).

Berdasarkan masalah yang ada pada perusahaan, 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 dilakukannya penyediaan ulang, serta metode penelitian menggunakan metode Model *Driven Development* (MDD) dalam menganalisis, desain, pengkodean, pengujian serta pemeliharaan.

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

### 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 perusahaan :

1. Data jenis barang
2. Data nama barang
3. Data satuan barang
4. Data harga barang
5. Data pelanggan
6. Data Rules
7. **Aplikasi Utama**
8. Membuat aplikasi pengendalian *Inventory* pada perusahaan.
9. Pada aplikasi yang dibuat akan menerapkan metode EOQ dan ROP agar pengendalian inventory dapat berjalan dengan baik.
10. **Kebutuhan Perangkat Lunak**

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

1. MySQL, digunakan untuk menyimpan *database* dari setiap data yang dibutuhkan.
2. PHP, sebagai bahasa pemograman yang akan digunakan untuk membuat aplikasi.
3. XAMPP, digunakan sebagai control panel.
4. *Visual Studio Code*, digunakan untuk menjalankan PHP sebagai pengembangan aplikasi.
5. *Star* UML, digunakan untuk membuat diagram-diagram UML.
6. *Microsoft* *Word*, digunakan untuk membuat laporan penelitian.
7. *Framework CSS*
8. *Balsamiq Mockups* 3 V 3.5.17
9. *Web server*
10. HTML
11. Sistem Oprasi *Windows 8*
12. **Kebutuhan Perangkat Keras**

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

Tabel 3. 1 Spesifikasi Minimum Hardware

|  |  |
| --- | --- |
| Proccessor | Intel® Celeron® CPU 1005M @ 1.90GHz Display 14” HD 1366 x 768 resolution, Anti-Glare |
| RAM | 2 GB |
| Memory | 320 GB |

Untuk kenyamanan pemakaian dapat meggunakan komputer dengan spesifikasi diatasnya.

### 3.2.4 Desain

Pada tahap ini penyusun membuat desain aplikasi Penendalian *Inventory* yang akan dibangun dari hasil analisis yang telah dilakukan sebelumnya. 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 pengendalian inventory. Perancangan akan dibuat dengan *Unified Modeling Language* (UML*)* agar dengan mudah dalam proses pengembangan dan visualisasinya. Diagram UML yang digunakan antara lain:

* 1. *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 : Pegawai dan pemilik perusahaan.
* *Use case* : Pendataan, pembelian, penjualan dan penyedian kembali.
  1. *Class Diagram* digunakan untuk menggambarkan atribut, *operation* dan juga *constraint* pada sistem yang akan dibuat. *Class diagram* yang akan di buat adalah sebagai berikut:

1. Class diagram *User*
2. Class diagram *Login/logout*
3. Class diagram data pelanggan
4. Class diagram data barang
5. Class diagram data barang masuk
6. Class diagram data barang keluar
7. Class diagram lapoan order
8. Class diagram rules
   1. *Activity Diagram* digunakan untuk memodelkan alur kerja dari sistem dan aktivitas dari *actor* dalam aplikasi. Adapun diagram alur yang akan dibuat diantaranya:
9. Proses login/logout
10. Proses olah data barang
11. Proses olah data barang masuk
12. Proses olah data barang keluar
13. Proses olah data laporan order
14. Proses olah data pelanggan
15. Proses olah data rules
16. Proses akses laporan data barang
17. Proses akses data barang masuk
18. Proses akses data barang keluar
19. Proses order barang
20. **Perancangan** ***Database***

*Database* adalah salah satu bagian yang paling penting dalam membangun aplikasi pengendalian *invetory*. 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:

1. Data barang
2. Data barang keluar
3. Data barang masuk
4. Data pelanggan
5. Data user
6. Data rules
7. **Perancangan Antar Muka (*User Interface*)**

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

1. Form login/logout
2. Form halaman awal aplikasi
3. From halaman data pelanggan
4. From halaman input data pelanggan
5. Form halaman data barang
6. Form halaman input data barang
7. Form halaman data barang masuk
8. Form halaman input data barang masuk
9. Form halaman data barang keluar
10. Form halaman input data barang keluar
11. Form halaman data rules
12. From halaman input data rules

### 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. *Visual Studo Code,* digunakan untuk membuat program interface aplikasi*.*
2. *Xampp* sebagi server web *apache* utuk pengembangan website.
3. *Database MYSQL*, digunakan untuk menyimpan *database* dari data barang yang ada.
4. *Framework* CSS digunakan untuk memperindah tampilan aplikasi.

### 3.2.6 Implementasi

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 dilanjutkan ketahap berikutnya.

### 3.2.7 Pembuatan Laporan

Tahap ini dibuat laporan untuk mempertanggung jawabkan hasil dari penelitian yang telah dilakukan, dibuat, dan 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 PERANCANGAN

## 4.1 Analisis

### 4.1.1 Analisis Masalah

Langkah awal dalam pembuatan sistem adalah meng-identifikasi permasalahan yang ada sebagai dasar untuk mebuat sebuah solusi yang akan disajikan dalam bentuk sebuah aplikasi. Langkah identifikasi dilakukan dengan cara observasi dan wawancara ke CV.Visya Pratama Industri, seperti yang telah disebutkan pada bab sebelumnya. Sehinga dapat dilakukan tindakan pengambilan solusi yang tepat utuk menyelesaikan permasalahan tersebut.

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

### 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.

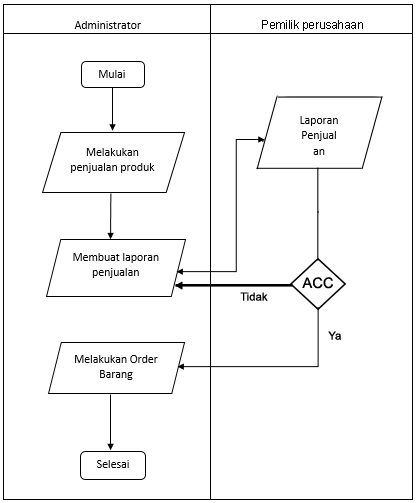
Lalu orang atau kelompok yang melakukan perbaikan atau perancangan suatu sistem dinamakan dengan sistem analis. Sistem analis adalah orang atau kelompok yang melaksanakan pengembangan sistem. Sistem analis menekuni permasalahan ataupun kebutuhan dari suatu sistem dan sistem analis juga merupakan orang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

Berdasarkan uraian di atas, maka dapat disimpulkan 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 digunakan. Dengan menganalisis prosedur sistem yang digunakan dan melakukan pengujian hasil, maka sistem dapat dievaluasi sehingga dapat dijadikan 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 perangkan keras yang dibutuhkan 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 dibutuhkan ketika hendak membuat aplikasi, berikut adalah analisis sistem yang sedang berjalan dan sistem yang diusulkan oleh penyusun :

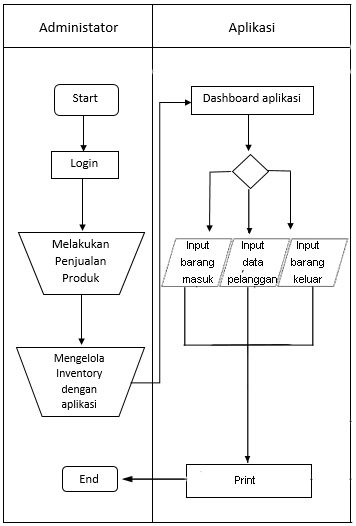
* 1. **Analisis Sistem yang Berjalan**



Gambar 4. Flow map analisis system yang berjalan

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

* 1. **Analisis Sistem yang di Usulkan**

**­**

Gambar 4. Flow map yang diusulkan

Pada gambar di atas dapat disimpulkan bahwa pegawaitidak perlu lagi membuat laporan penjualan secara manual karena, laporan penjualan telah di buat 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 dibutuhkan dalam membuat aplikasi pengendalian *inventory* ini.

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

* + 1. Kebutuhan Sistem

Tabel 4. Kebutuhan sistem

|  |  |  |
| --- | --- | --- |
| **No** | **Perangkat Keras** | **Perangkat Lunak** |
| 1 | Seperangkat PC / Laptop | *Windows* 8 32 bit |
| 2 | - | *Xampp* |
| 3 | - | *Visual Studio Code* |
| 4 | - | *Mozila* *Fire* *fox* |
| 5 | - | *Microsoft* *word* 2013 |
| 6 | - | *Star* UML |
| 7 | - | *Balsamiq mockups* 3V.3.5.17 |
| 8 | - | *Framework CSS* |

* + 1. Spesifikasi Perangkat Keras

Tabel 4. Spesifikasi perangkat keras

|  |  |  |
| --- | --- | --- |
| **No** | **Spesifikasi** | **Komputer** |
| 1 | *Processor* | Intel Celeron 1.80 GHz (2CPUs) |
| 2 | Ram (*Randaom Acces Memory*) | 2 GB |
| 3 | *Hard* *Disk* | 320 GB |

* + 1. Spesifikasi Perangkat Lunak

Tabel 4. Spesifikasi perangkat lunak

|  |  |  |
| --- | --- | --- |
| **No** | **Spesifikasi** | **Komputer** |
| 1 | Sistem Operasi | *Windows* 8 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 dibuat, pengguna aplikasi ini adalah ;

1. Pemilik perusahaan
2. Pegawai

agar aplikasi ini dapat digunakan dengan baik dan mudah untuk dioperasikan penyusun akan menyediakan berbagai menu pada *user* *interface* yang akan dibuat.

### 4.1.5 User Interface

*User* *interface* dari aplikasi sangat berpengaruh terhadap minat dari *user* dalam mengoperasikannya. Hal ini mencakup perangkat yang digunakan sebagai piranti masukan dan keluaran dari aplikasi yang akan dibuat. Karena aplikasi yang dibuat ditujukan untuk mengefisiensikan pekerjaan pemilik perusahaan dalam mengelola *inventory.*

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

### 4.1.6 Fitur-Fitur

Fitrur-fitur yang digunakan dalam aplikasi ini dimaksudkan agar *user* dapat dengan mudah mengoperasikannya, ditunjang dengan metode EOQ dan ROP untuk memberikan biaya paling ekonomis dalam pengelolaan *inventory,* bagianterpenting dari fitur-fitur yang ada adalah sebagai berikut :

1. *Login* / *logout user*
2. Data pelanggan
3. Data barang, barang masuk dan keluar
4. Penghitungan EOQ
5. Penghitungan ROP
6. Biaya-biaya
7. *Dashboard* dan grafik sederhana

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

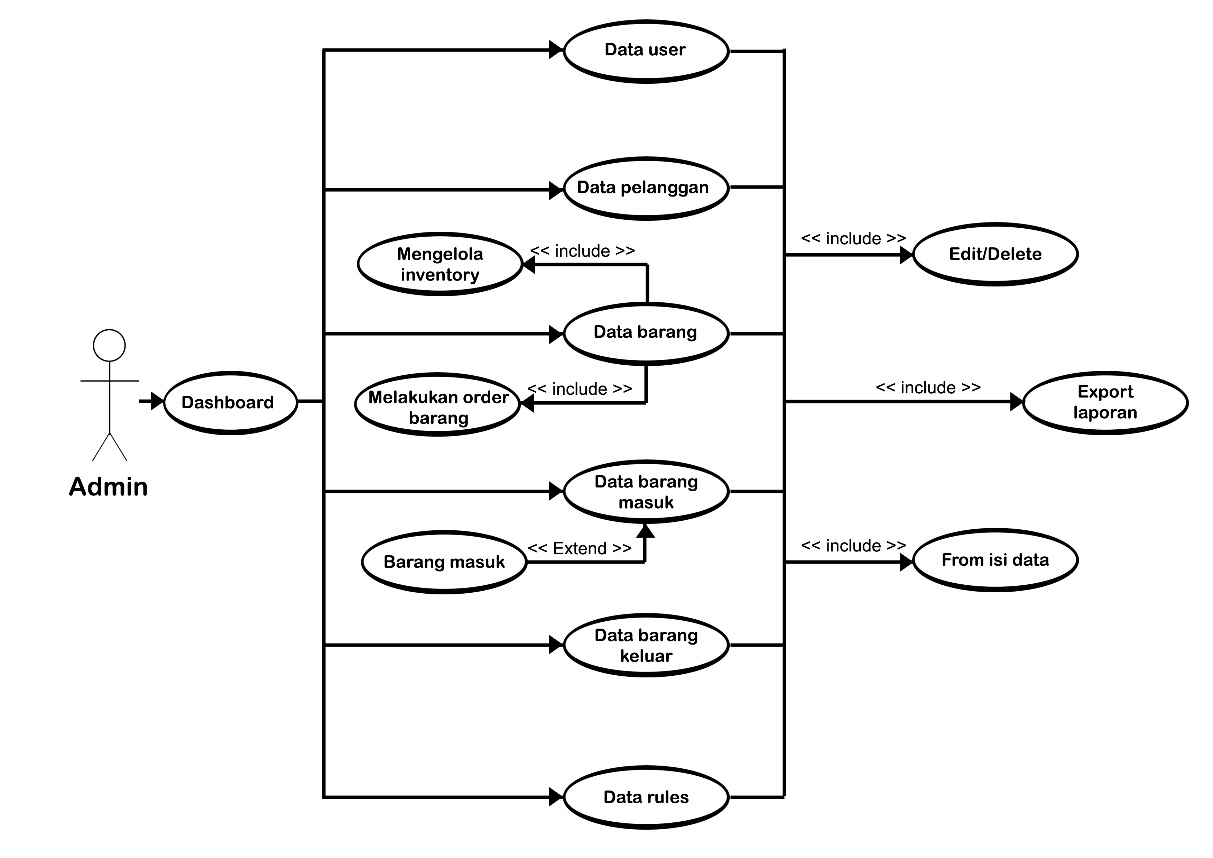
## 4.2 Perancangan Sistem

Perancangan sistem merupakan perancangan yang melibatkan sistem tertentu, dalam arti sistem yang dirancang adalah sistem yang memang ingin dibuat dan dikendalikan.

Dengan perancangan, maka sistem dapat diarahkan atau dikendalikan sesuai dengan keinginan. Namun tidak sekedar perancangan yang dipilih melainkan sesuai dengan data yang diperoleh.

Pada tahap ini perangkat lunak dideskripsikan dengan model analisis penggunakan 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 *Usacase* *Diagram*



Gambar 4. 3 Usecase Diagram

Diagram Use Case di atas menunjukkan Actor yang terlibat di dalam Aplikasi pengelolaan Inventory ada 1 aktor, yang mempunyai hak sebagai berikut :

1. Definisi Aktor

* Pegawai

Pegawai adalah admin dari perusahaan yang diberikan kewenangan untuk melakukan pengelolaan (*maintenance*) aplikasi. Dalam sistem yang akan dikembangkan peran administrator dipegang oleh pegawai. Pegawai dapat melihat dan mengubah semua data sistem, dan dapat membenahi kesalahan yang terjadi dalam sistem.

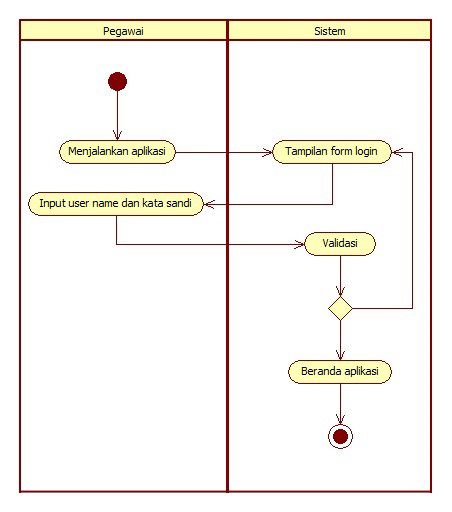
2. *Use case*

Tabel 4. 4 Use case

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Aktor** | ***Use case*** | **Deskripsi** |
| 1 | Pegawai | *Log in* | Melakukan *log in* atau masuk ke dalam aplikasi pengelolaan *inventory.* |
|  |  | Input data | Merupakan proses penginputan data, penghapusan data, pengolahan data barang masuk, barang keluar, data pelanggan, laporan, rules dan lainnya. |
|  |  | Edit data | Merupakan Proses pengeditan data barang yang tidak sesuai. |
|  |  | Hapus data | Merupakan proses penghapusan data yang salah. |
|  |  | Log out | Merupakan proses *log out* atau keluar dari aplikasi pengelolaan *inventory.* |

### 4.2.2 Activity Diagram

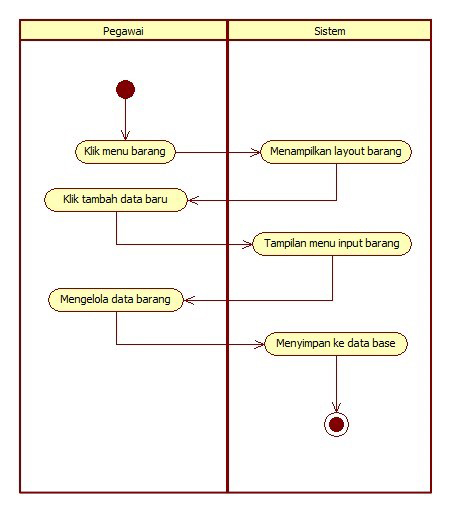
1. ***Activity Diagram* *Log in***



Gambar 4. Activity diagram log in

Pada Gambar *Activity diagram* 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.

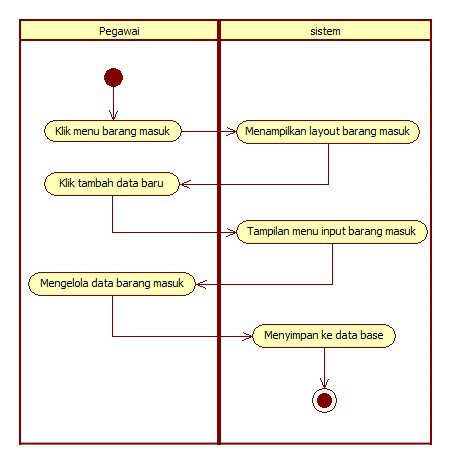
1. ***Activity Diagram* Olah Data Barang**



Gambar 4. Activity diagram olah data barang

Pada Gambar *Activity* diagram olah data barang menggambarkan proses *input* data barang pada sistem, pegawai meng-klik menu barang lalu akan tampil *layout* barang kemudian pegawai atau pemilik perusahaan 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 *submit*  maka data akan otomatis tersimpan ke dalam *database*.

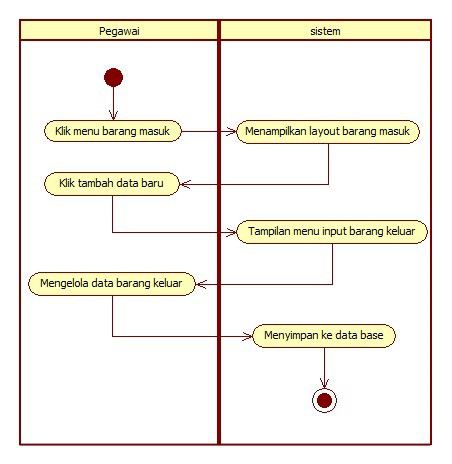
1. ***Activity Diagram* Olah Data Barang Masuk**



Gambar 4. Activity Diagram Olah Data Barang Masuk

Pada Gambar *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 memilih 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*.

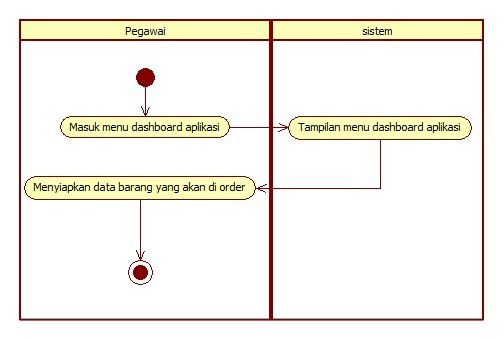
1. ***Activity Diagram* Olah data Barang Keluar**



Gambar 4. Activity Diagram Olah data Barang Keluar

Pada Gambar *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*.

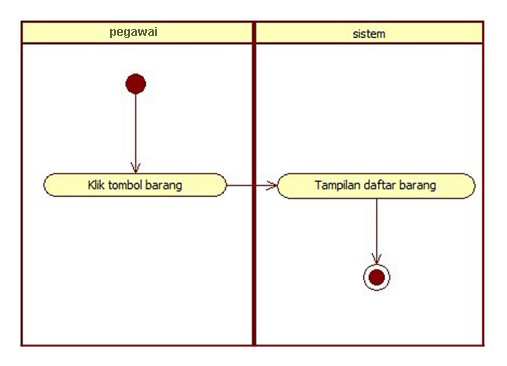
1. ***Activity Diagram* Olah Laporan Order**



Gambar 4. Activity Diagram Olah Laporan Order

Pada Gambar 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 disedaikan kembali.

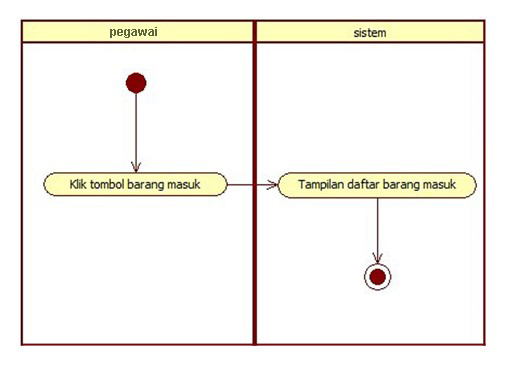
1. ***Activity Diagram* Laporan Data Barang**



Gambar 4. Activity diagram Laporan Data Barang

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

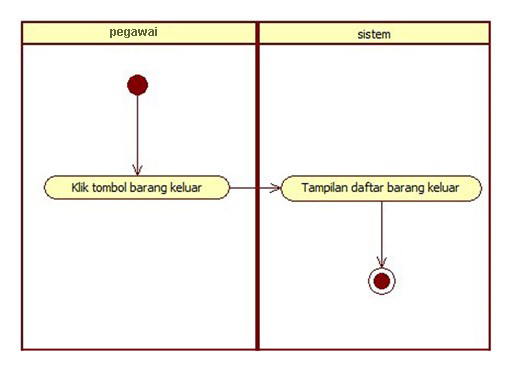
1. ***Activity Diagram* Data Barang Masuk**



Gambar 4. Activity diagram data barang masuk

Pada Gambar Activity diagram laporan data barang masuk menggambarkan proses yang dilakukan oleh pegawai ketika akan melihat laporan data barang masuk yang telah di kelola, hal pertama yang dilakukan oleh pegawai atau pemilik perusahaan adalah klik tombol barang masuk, lalu sistem akan menampilkan laporan data barang masuk yang telah di kelola.

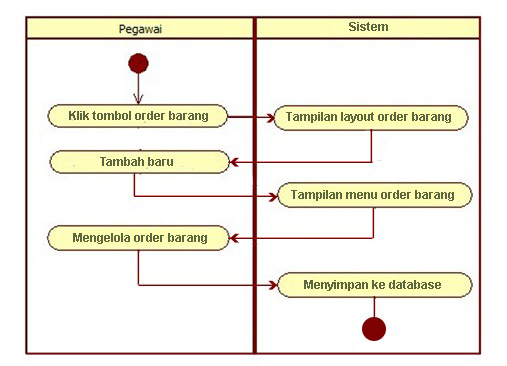
1. ***Activity Diagram* Data Barang Keluar**



Gambar 4. Activity diagram barang keluar

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

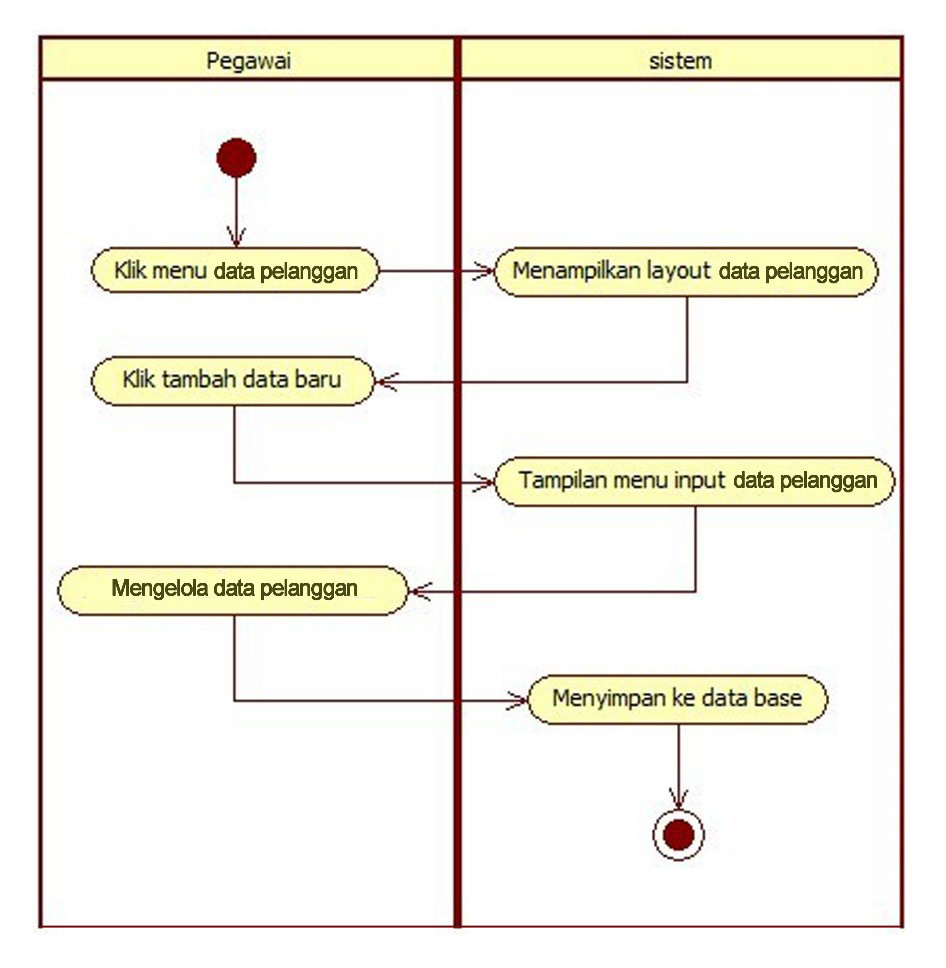
1. ***Activity Diagram* Order Barang**



Gambar 4. Activity diagram order barang

Pada Gambar *Activity* diagram order barang menggambarkan proses yang dilakukan pegawai ketika akan melakukan order barang, pertama kali yang pegawai lakukan adalah klik tombol order barang, lalu klik tombol tambah baru, lalu pegawai meng-*input* order barang apa saja yang akan di beli lalu klik *save* dan data akan tersimpan otomatis kedalam *database.*

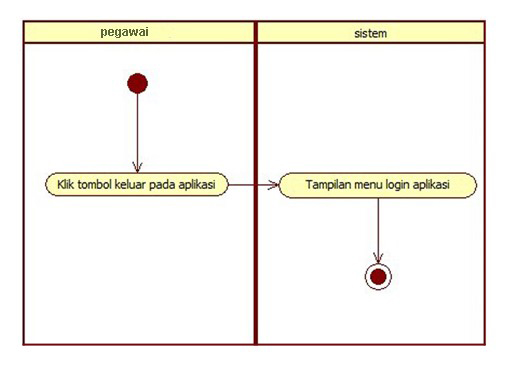
1. ***Activity Diagram* Data pelanggan**



Gambar 4. Activity diagram datra pelanggan

Pada Gambar *Activity* diagram olah data pelanggan menggambarkan proses *input* data pelanggan baru ke dalam sistem, pertama pegawai meng-klik menu barang masuk lalu akan tampil *layout* data pelanggan, kemudian pegawai memilih menu input data pelanggan dan sistem menampilkan *form* *input* data pelanggan lalu pegawai meng-*input* identitas pelanggan dan data barang apa saja yang baru dibeli oleh pelanggan (bisa *input* atau mengubah data yang sudah ada) lalu menekan tombol *save* maka data akan tersimpan otomatis ke dalam *database*.

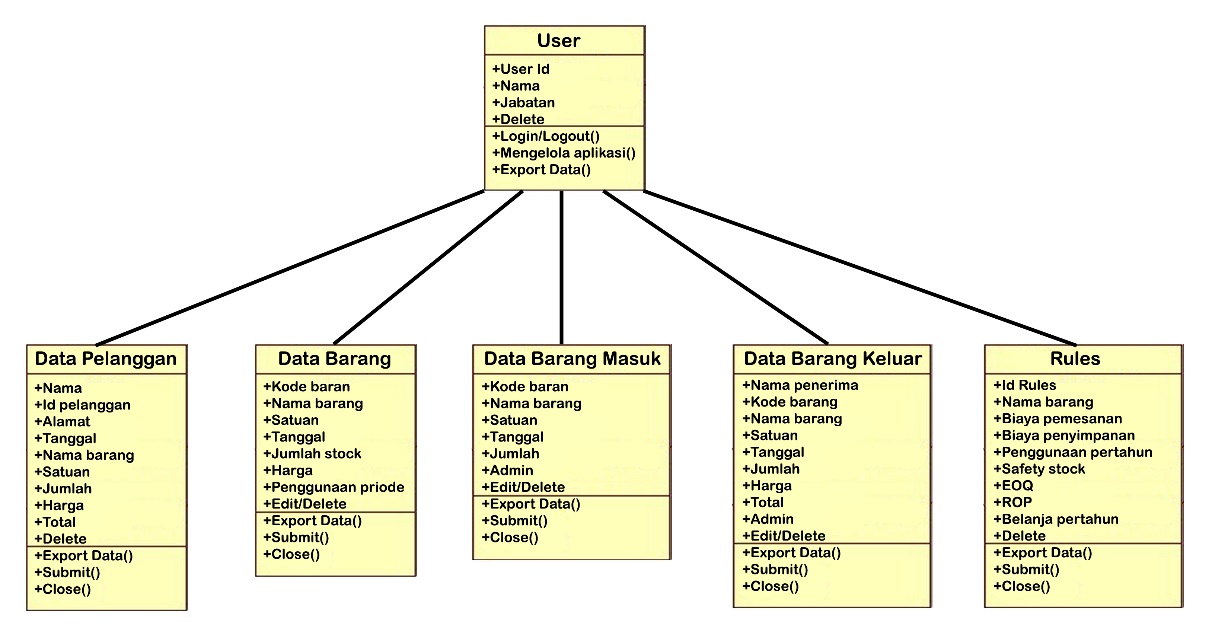
1. ***Activity Diagram* *Logout***



Gambar 4. Activity diagram log out

Pada gambar 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 diarahkan ke halaman *login* awal aplikasi.

### 4.2.3 Class Diagram



Gambar 4. Class diagram

Pada gambar *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 perusahaan.

### 4.2.4 Perancangan *Database*

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

1. **Data barang**

Tujuan dibuatnya *form* *input* data barang adalah untuk menginputkan data ke dalam *database*, dan rancangan *database* nya dapat dilihat pada tabel di bawah ini :

Tabel 4. 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 | Tanggal | *Number* | 10 | Menyimpan tanggal |
| 5 | Jumlah stock | *Number* | 5 | Menyimpan jumlah stock barang |
| 6 | Harga | *Number* | 6 | Menyimpan harga barang |
| 7 | Penggunaan\_periode | *Number* | 3 | Menyimpan penggunaan / periode |

1. **Data barang masuk**

Tujuan dibuatnya *form* *input* data barang masuk adalah untuk menginputkan data ke dalam *database*, dan rancangan *database* nya dapat dilihat pada tabel berikut ini :

Tabel 4. Data barang masuk

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | ***Field*** | ***Type*** | **Panjang** | **Keterangan** |
| 1 | Kode\_barang | *Varchar* | 5 | Menyimpan kode barang |
| 2 | Nama Barang | *Varchar* | 30 | Menyimpan nama barang |
| 3 | Satuan | *Varchar* | 20 | Menyimpan jenis satuan |
| 4 | Tanggal | *Varchar* | 10 | Menyimpan tanggal input |
| 5 | Jumlah | *Number* | 4 | Menyimpan jumlah barang |
| 6 | Admin | *Varchar* | 10 | Menyimpan nama admin |

1. **Data barang keluar**

Tujuan dibuatnya form input data barang keluar adalah agar user dapat mengetahui barang yang keluar sehinnga data pada tabel data barang akan berkurang dan untuk menginputkankan data ke dalam *database*, dan rancangan *database* nya dapat dilihat pada tabel di bawah ini :

Tabel 4. Data barang keluar

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | ***Field*** | ***Type*** | **Panjang** | **Keterangan** |
| 1 | Nama penerima | *Varchar* | 25 | Menyimpan nama pembeli/penerima barang |
| 2 | Kode barang | *Varchar* | 5 | Menyimpan kode barang |
| 3 | Nama Barang | *Varchar* | 25 | Menyimpan nama barang |
| 4 | Satuan | *Varchar* | 15 | Menyimpan jenis satuan |
| 5 | Tanggal | *Varchar* | 10 | Menyimpan tanggal barang keluar |
| 6 | Jumlah | *Number* | 4 | Menyimpan jumlah barang |
| 7 | Harga | *Number* | 10 | Menyimpan harga barang |
| 8 | Total | *Number* | 15 | Menyimpan total belanja |
| 9 | Admin | *Varchar* | 25 | Menyimpan nama admin |

1. **Data Pelanggan**

Tujuan dibuatnya form input data *user* adalah untuk menginputkan data ke dalam *database*, dan rancangan *database*nya dapat dilihat pada tabel di bawah ini :

Tabel 4. Data pelanggan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | ***Field*** | ***Type*** | **Panjang** | **Keterangan** |
| 1 | Nama | *Varchar* | 25 | Menyimpan nama |
| 2 | Id Pelanggan | *Number* | 5 | Menyimpan id pelanggan |
| 3 | Alamat | *Varcahar* | 30 | Menyimpan alamat |
| 3 | Tanggal | *Number* | 10 | Menyimpan tanggal |
| 4 | Nama barang | *Varchar* | 20 | Menyimpan jenis barang |
| 5 | Satuan | *Varchar* | 10 | Menyimpan jenis satuan |
| 5 | Jumlah | *Number* | 10 | Menyimpan jenis jumlah |
| 6 | Harga | *Number* | 30 | Menyimpan jenis harga |
| 7 | Total | *Number* | 30 | Menyimpan jenis total |

1. **Data user**

Tujuan dibuatnya form input data *user* adalah untuk menginputkan data ke dalam *database*, dan rancangan *database* nya dapat dilihat pada tabel di bawah ini :

Tabel 4. Data user

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | ***Field*** | ***Type*** | **Panjang** | **Keterangan** |
| 1 | User Id | *Number* | 5 | Menyimpan *user id* |
| 2 | Nama | *Varchar* | 20 | Menyimpan nama |
| 3 | Jabatan | *Varchar* | 20 | Menyimpan jabatan |

1. **Rules**

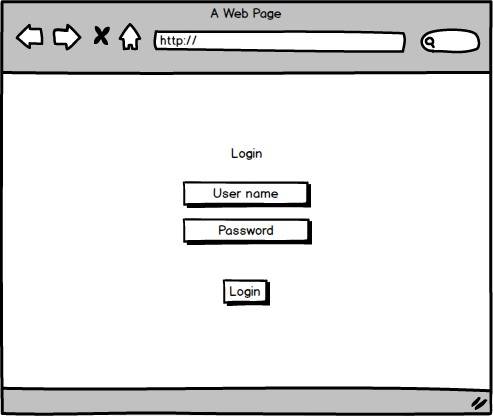
Tujuan dibuatkannya form input data *rules* adalah untukmeng-*input*-kan data ke dalam *database*, dan rancangan *database* nya dapat dilihat pada tabel di bawah ini :

Tabel 4. Data rules

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | ***Field*** | ***Type*** | **Panjang** | **Keterangan** |
| 1 | Id Rules | *Number* | 5 | Menyimpan *id rules* |
| 2 | Nama Barang | *Varchar* | 20 | Menyimpan nama barang |
| 3 | Biaya Pemesanan | *Number* | 10 | Menyimpan biaya pemesanan |
| 4 | Biaya Penyimpanan | *Number* | 10 | Menyimpan biaya penympanan |
| 5 | Penggunaan Pertahun | *Number* | 10 | Menyimpan penggunaan pertahun |
| 6 | Safety Stock | *Number* | 10 | Menyimpan safety stock |
| 7 | EOQ | *Number* | 10 | Menyimpan EOQ |
| 8 | ROP | *Number* | 10 | Menyimpan ROP |
| 9 | Belanja Pertahun | *Number* | 10 | Menyimpan belanja pertahun |

### 4.2.5 Desain

1. ***Form Log in***



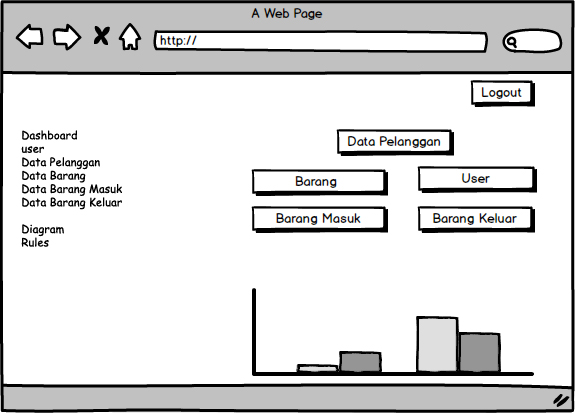
Gambar 4. Mock up log in

Keterangan :

Tabel 4. Keterangan mock up log in

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Tombol** | **Keterangan** |
| 1 | *User Name* | Digunakan untuk mengisi *user name* *user* |
| 2 | *Password* | Digunakan untuk mengisi *password* *user* |
| 3 | *Button Log in* | Digunakan untuk *login* aplikasi |

1. **Halaman awal aplikasi**



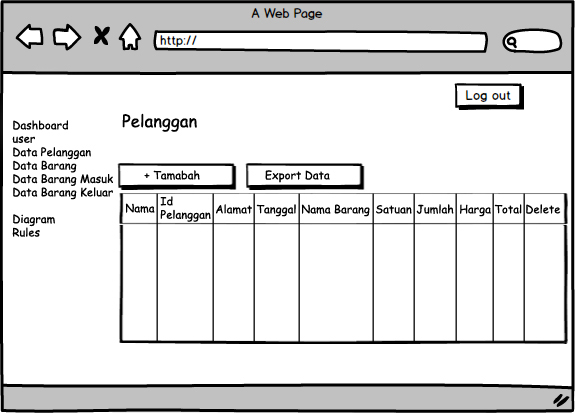
Gambar 4. Mock up halaman login

Keterangan :

Tabel 4. Keterangan mock up halaman login

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Tombol** | **Keterangan** |
| 1 | *Dashboard* | Digunakan untuk pergi ke halaman menu utama |
| 2 | Data pelanggan | Digunakan untuk pergi ke halaman data pelanggan |
| 3 | Barang masuk | Digunakan untuk pergi ke halaman daftar barang masuk |
| 4 | Barang keluar | Digunakan untuk pergi ke halaman daftar barang keluar |
| 5 | *Logout* | Digunakan user untuk keluar dari aplikasi dan kembali ke form login |
| 6 | Barang | Digunakan untuk pergi ke halaman daftar barang |
| 7 | *User* | Digunakan untuk pergi ke halaman penampil *user* aplikasi |
| 8 | Diagram | Adalah sebuah tampilan yang menggambarkan barang yang ada pada aplikasi |
| 9 | Rules | Digunakan untuk pergi ke halaman daftar rules |

1. **Halaman Data Pelanggan**



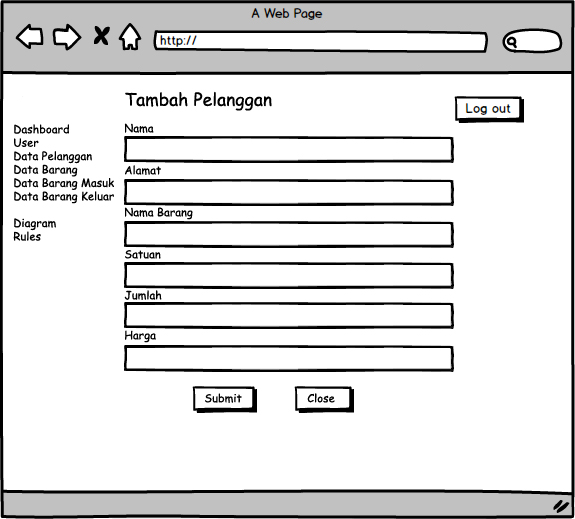
Gambar 4. Mock up data pelanggan

Keterangan :

Tabel 4. Keterangan mock up data pelanggan

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Tombol** | **Keterangan** |
| 1 | *Dashboard* | Digunakan untuk pergi ke halaman menu utama |
| 2 | *User* | Digunakan untuk pergi ke halaman user |
| 3 | Data pelanggan | Digunakan untuk pergi ke halaman data pelanggan |
| 4 | Data barang | Digunakan untuk pergi ke halaman daftar barang |
| 5 | Data barang masuk | Digunakan untuk pergi ke halaman daftar barang masuk |
| 6 | Data barang keluar | Digunakan untuk pergi ke halaman daftar barang keluar |
| 7 | Diagram | Adalah sebuah tampilan yang menggambar- kan barang yang ada pada aplikasi |
| 8 | *Rules* | Digunakan untuk pergi ke halaman daftar *rules* |
| 9 | *Log out* | Digunakan user untuk keluar dari aplikasi dan kembali ke form login |
| 10 | Tambah | Digunakan untuk pergi ke halaman input data pelanggan |
| 11 | *Export* | Digunakan untuk pergi ke halaman export ke file (PDF, *excel, word, print*) |
| 12 | Tabel pelanggan | Adalah tampilan tabel yang menunjukan daftar pelanggan yang telah di *input*-kan user |

1. **Halaman Input Data Pelanggan**



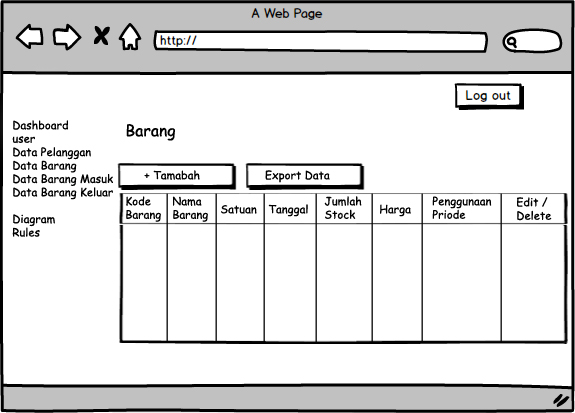
Gambar 4. Mock up input data pelanggan

Keterangan :

Tabel 4. Keterangan mock up input data pelanggan

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Tombol** | **Keterangan** |
| 1 | *Dashboard* | Digunakan untuk pergi ke halaman menu utama |
| 2 | *User* | Digunakan untuk pergi ke halaman user |
| 3 | Data pelanggan | Digunakan untuk pergi ke halaman data pelanggan |
| 4 | Data barang | Digunakan untuk pergi ke halaman daftar barang |
| 5 | Data barang masuk | Digunakan untuk pergi ke halaman daftar barang masuk |
| 6 | Data barang keluar | Digunakan untuk pergi ke halaman daftar barang keluar |
| 7 | Diagram | Adalah sebuah tampilan yang menggambar- kan barang yang ada pada aplikasi |
| 8 | *Rules* | Digunakan untuk pergi ke halaman daftar *rules* |
| 9 | *Log out* | Digunakan user untuk keluar dari aplikasi dan kembali ke form login |
| 10 | Tabel pelanggan | Adalah tampilan tabel yang menunjukan daftar pelanggan yang telah di *input*-kan user |
| 11 | Nama | Digunakan untuk *input* nama |
| 12 | Alamat | Digunakan untuk *input* alamat |
| 13 | Nama Barang | Digunakan untuk *input* nama barang |
| 14 | Satuan | Digunakan untuk *input* satuan |
| 15 | Jumlah | Digunakan untuk *input* jumlah |
| 16 | Harga | Digunakan untuk *input* harga |
| 17 | Submit | Digunakan untuk menyimpan data ke *database* |
| 18 | Close | Digunakan untuk mengosongkan semua bagian yang telah di isi |

1. **Halaman Data Barang**



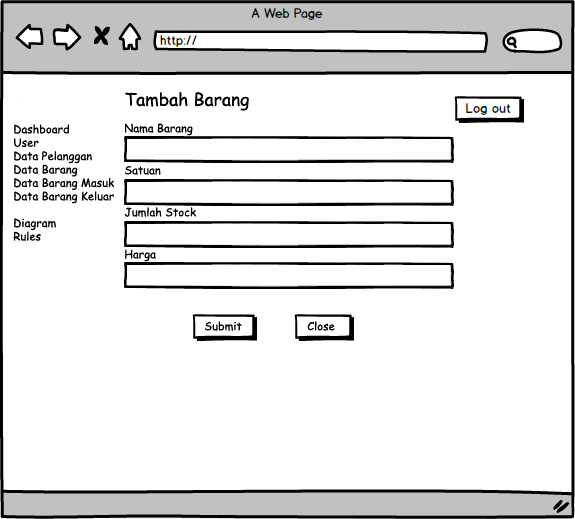
Gambar 4. Mock up data barang

Keterangan :

Tabel 4. Keterangan mock up data barang

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Tombol** | **Keterangan** |
| 1 | *Dashboard* | Digunakan untuk pergi ke halaman menu utama |
| 2 | *User* | Digunakan untuk pergi ke halaman user |
| 3 | Data pelanggan | Digunakan untuk pergi ke halaman data pelanggan |
| 4 | Data barang | Digunakan untuk pergi ke halaman daftar barang |
| 5 | Data barang masuk | Digunakan untuk pergi ke halaman daftar barang masuk |
| 6 | Data barang keluar | Digunakan untuk pergi ke halaman daftar barang keluar |
| 7 | Diagram | Adalah sebuah tampilan yang menggambar- kan barang yang ada pada aplikasi |
| 8 | *Rules* | Digunakan untuk pergi ke halaman daftar *rules* |
| 9 | *Log out* | Digunakan user untuk keluar dari aplikasi dan kembali ke form login |
| 10 | Tambah | Digunakan untuk pergi ke halaman input data barang |
| 11 | *Export* | Digunakan untuk pergi ke halaman export ke file (PDF, *excel, word, print*) |
| 12 | Tabel barang | Adalah tampilan tabel yang menunjukan daftar barang yang telah di *input*-kan user |

1. **Halaman Input Data Barang**



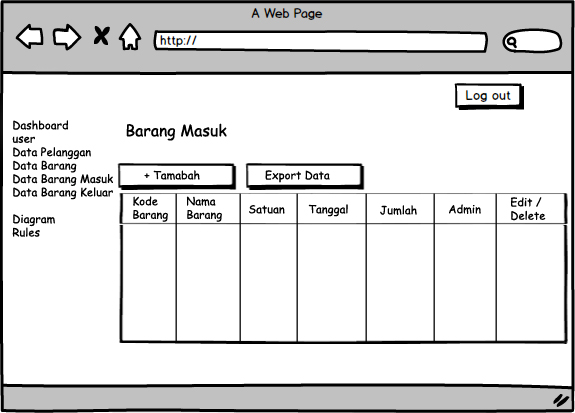
Gambar 4. Mock up input data barang

Keterangan :

Tabel 4. Keterangan mock up input data barang

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Tombol** | **Keterangan** |
| 1 | *Dashboard* | Digunakan untuk pergi ke halaman menu utama |
| 2 | *User* | Digunakan untuk pergi ke halaman user |
| 3 | Data pelanggan | Digunakan untuk pergi ke halaman data pelanggan |
| 4 | Data barang | Digunakan untuk pergi ke halaman daftar barang |
| 5 | Data barang masuk | Digunakan untuk pergi ke halaman daftar barang masuk |
| 6 | Data barang keluar | Digunakan untuk pergi ke halaman daftar barang keluar |
| 7 | Diagram | Adalah sebuah tampilan yang menggambar- kan barang yang ada pada aplikasi |
| 8 | *Rules* | Digunakan untuk pergi ke halaman daftar *rules* |
| 9 | *Log out* | Digunakan user untuk keluar dari aplikasi dan kembali ke form login |
| 10 | Tabel barang | Adalah tampilan tabel yang menunjukan daftar barang yang telah di *input*-kan user |
| 11 | Nama barang | Digunakan untuk *input* nama barang |
| 12 | Satuan | Digunakan untuk *input* satuan |
| 13 | Jumlah stock | Digunakan untuk *input* jumlah *stock* |
| 14 | Harga | Digunakan untuk *input* harga |
| 15 | Submit | Digunakan untuk menyimpan data ke *database* |
| 16 | Close | Digunakan untuk mengosongkan semua bagian yang telah di isi |

1. **Halaman Data Barang Masuk**



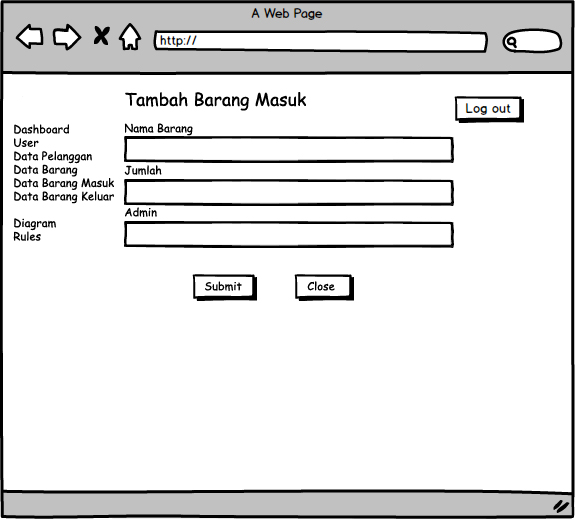
Gambar 4. Mock up data barang masuk

Keterangan :

Tabel 4. Keterangan mock up data barang masuk

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Tombol** | **Keterangan** |
| 1 | *Dashboard* | Digunakan untuk pergi ke halaman menu utama |
| 2 | *User* | Digunakan untuk pergi ke halaman user |
| 3 | Data pelanggan | Digunakan untuk pergi ke halaman data pelanggan |
| 4 | Data barang | Digunakan untuk pergi ke halaman daftar barang |
| 5 | Data barang masuk | Digunakan untuk pergi ke halaman daftar barang masuk |
| 6 | Data barang keluar | Digunakan untuk pergi ke halaman daftar barang keluar |
| 7 | Diagram | Adalah sebuah tampilan yang menggambar- kan barang yang ada pada aplikasi |
| 8 | *Rules* | Digunakan untuk pergi ke halaman daftar *rules* |
| 9 | *Log out* | Digunakan user untuk keluar dari aplikasi dan kembali ke form login |
| 10 | Tambah | Digunakan untuk pergi ke halaman input data barang masuk |
| 11 | *Export* | Digunakan untuk pergi ke halaman export ke file (PDF, *excel, word, print*) |
| 12 | Tabel barang masuk | Adalah tampilan tabel yang menunjukan daftar barang masuk yang telah di *input*-kan user |

1. **Halaman Input data Barang Masuk**



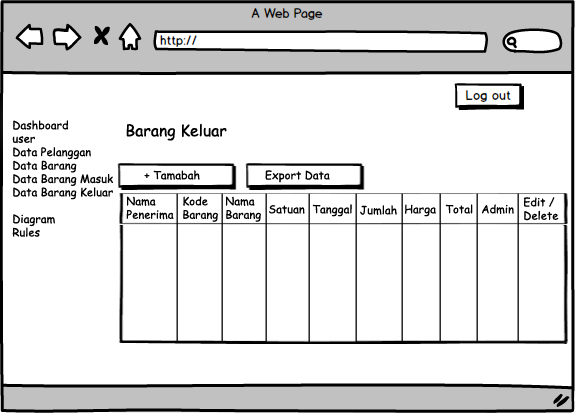
Gambar 4. Mock up input data barang masuk

Keterangan :

Tabel 4. Keterangan mock up input data barang masuk

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Tombol** | **Keterangan** |
| 1 | *Dashboard* | Digunakan untuk pergi ke halaman menu utama |
| 2 | *User* | Digunakan untuk pergi ke halaman user |
| 3 | Data pelanggan | Digunakan untuk pergi ke halaman data pelanggan |
| 4 | Data barang | Digunakan untuk pergi ke halaman daftar barang |
| 5 | Data barang masuk | Digunakan untuk pergi ke halaman daftar barang masuk |
| 6 | Data barang keluar | Digunakan untuk pergi ke halaman daftar barang keluar |
| 7 | Diagram | Adalah sebuah tampilan yang menggambar- kan barang yang ada pada aplikasi |
| 8 | *Rules* | Digunakan untuk pergi ke halaman daftar *rules* |
| 9 | *Log out* | Digunakan user untuk keluar dari aplikasi dan kembali ke form login |
| 10 | Tabel barang masuk | Adalah tampilan tabel yang menunjukan daftar barang masuk yang telah di *input*-kan user |
| 11 | Nama barang | Digunakan untuk *input* nama barang |
| 12 | Jumlah | Digunakan untuk *input* jumlah |
| 13 | Admin | Digunakan untuk *input* admin |
| 14 | Submit | Digunakan untuk menyimpan data ke *database* |
| 15 | Close | Digunakan untuk mengosongkan semua bagian yang telah di isi |

1. **Halaman Data Barang Keluar**



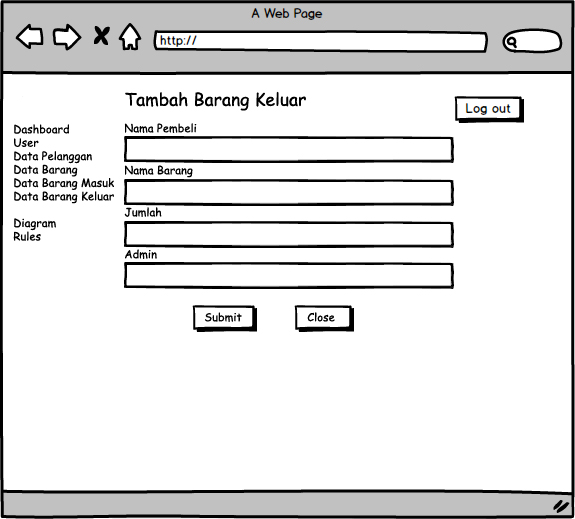
Gambar 4. Mock up data barang keluar

Keterangan :

Tabel 4. Keterangan mock up data barang keluar

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Tombol** | **Keterangan** |
| 1 | *Dashboard* | Digunakan untuk pergi ke halaman menu utama |
| 2 | *User* | Digunakan untuk pergi ke halaman user |
| 3 | Data pelanggan | Digunakan untuk pergi ke halaman data pelanggan |
| 4 | Data barang | Digunakan untuk pergi ke halaman daftar barang |
| 5 | Data barang masuk | Digunakan untuk pergi ke halaman daftar barang masuk |
| 6 | Data barang keluar | Digunakan untuk pergi ke halaman daftar barang keluar |
| 7 | Diagram | Adalah sebuah tampilan yang menggambar- kan barang yang ada pada aplikasi |
| 8 | *Rules* | Digunakan untuk pergi ke halaman daftar *rules* |
| 9 | *Log out* | Digunakan user untuk keluar dari aplikasi dan kembali ke form login |
| 10 | Tambah | Digunakan untuk pergi ke halaman input data barang keluar |
| 11 | *Export* | Digunakan untuk pergi ke halaman export ke file (PDF, *excel, word, print*) |
| 12 | Tabel barang keluar | Adalah tampilan tabel yang menunjukan daftar barang keluar yang telah di *input*-kan user |

1. **Halaman Input Data Barang Keluar**



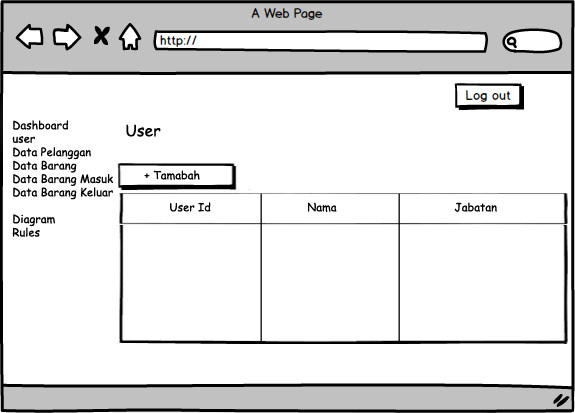
Gambar 4. Mock up input data barang keluar

Keterangan :

Tabel 4. Keterangan mock up input data barang keluar

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Tombol** | **Keterangan** |
| 1 | *Dashboard* | Digunakan untuk pergi ke halaman menu utama |
| 2 | *User* | D gunakan untuk pergi ke halaman user |
| 3 | Data pelanggan | Digunakan untuk pergi ke halaman data pelanggan |
| 4 | Data barang | Digunakan untuk pergi ke halaman daftar barang |
| 5 | Data barang masuk | Digunakan untuk pergi ke halaman daftar barang masuk |
| 6 | Data barang keluar | Digunakan untuk pergi ke halaman daftar barang keluar |
| 7 | Diagram | Adalah sebuah tampilan yang menggambar- kan barang yang ada pada aplikasi |
| 8 | *Rules* | Digunakan untuk pergi ke halaman daftar *rules* |
| 9 | *Log out* | Digunakan user untuk keluar dari aplikasi dan kembali ke form login |
| 10 | Tabel barang keluar | Adalah tampilan tabel yang menunjukan daftar barang keluar yang telah di *input*-kan user |
| 11 | Nama penerima | Digunakan untuk *input* nama penerima |
| 12 | Nama barang | Digunakan untuk *input* nama barang |
| 13 | Jumlah | Digunakan untuk *input* jumlah |
| 14 | Admin | Digunakan untuk *input* admin |
| 15 | Submit | Digunakan untuk menyimpan data ke *database* |
| 16 | Close | Digunakan untuk mengosongkan semua bagian yang telah di isi |

1. **Halaman Data User**

****

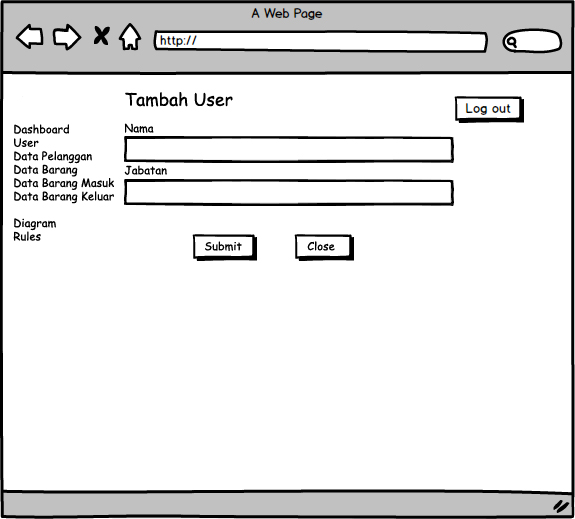
Gambar 4. Data user

Keterangan :

Tabel 4. Keterangan data user

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Tombol** | **Keterangan** |
| 1 | *Dashboard* | Digunakan untuk pergi ke halaman menu utama |
| 2 | *User* | Digunakan untuk pergi ke halaman user |
| 3 | Data pelanggan | Digunakan untuk pergi ke halaman data pelanggan |
| 4 | Data barang | Digunakan untuk pergi ke halaman daftar barang |
| 5 | Data barang masuk | Digunakan untuk pergi ke halaman daftar barang masuk |
| 6 | Data barang keluar | Digunakan untuk pergi ke halaman daftar barang keluar |
| 7 | Diagram | Adalah sebuah tampilan yang menggambar- kan barang yang ada pada aplikasi |
| 8 | *Rules* | Digunakan untuk pergi ke halaman daftar *rules* |
| 9 | *Log out* | Digunakan user untuk keluar dari aplikasi dan kembali ke form login |
| 10 | Tambah | Digunakan untuk pergi ke halaman input data user |
| 11 | *Export* | Digunakan untuk pergi ke halaman export ke file (PDF, *excel, word, print*) |
| 12 | Tabel user | Adalah tampilan tabel yang menunjukan daftar identitas user yang telah di *input*-kan user |

1. **Halaman Input User**

****

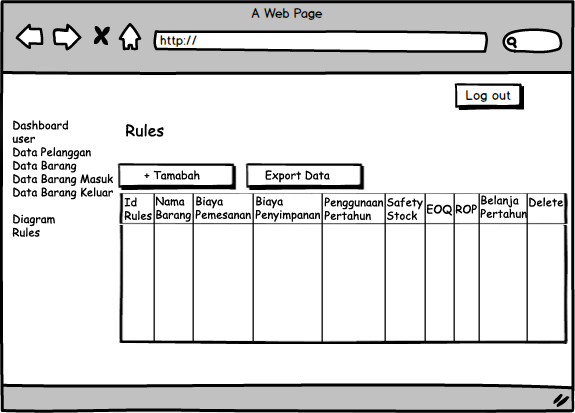
Gambar 4. Halaman input user

Keterangan :

Tabel 4. Keterangan halaman input user

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Tombol** | **Keterangan** |
| 1 | *Dashboard* | Digunakan untuk pergi ke halaman menu utama |
| 2 | *User* | Digunakan untuk pergi ke halaman user |
| 3 | Data pelanggan | Digunakan untuk pergi ke halaman data pelanggan |
| 4 | Data barang | Digunakan untuk pergi ke halaman daftar barang |
| 5 | Data barang masuk | Digunakan untuk pergi ke halaman daftar barang masuk |
| 6 | Data barang keluar | Digunakan untuk pergi ke halaman daftar barang keluar |
| 7 | Diagram | Adalah sebuah tampilan yang menggambar- kan barang yang ada pada aplikasi |
| 8 | *Rules* | Digunakan untuk pergi ke halaman daftar *rules* |
| 9 | *Log out* | Digunakan user untuk keluar dari aplikasi dan kembali ke form login |
| 10 | Tabel data user | Adalah tampilan tabel yang menunjukan daftar data identitas user yang telah di *input*-kan user |
| 11 | User Id | Digunakan untuk *input* *user id* |
| 12 | Nama | Digunakan untuk *input* nama |
| 13 | Jabatan | Digunakan untuk *input* jabatan |
| 14 | Submit | Digunakan untuk menyimpan data ke *database* |
| 15 | Close | Digunakan untuk mengosongkan semua bagian yang telah di isi |

1. **Halaman Data Rules**

****

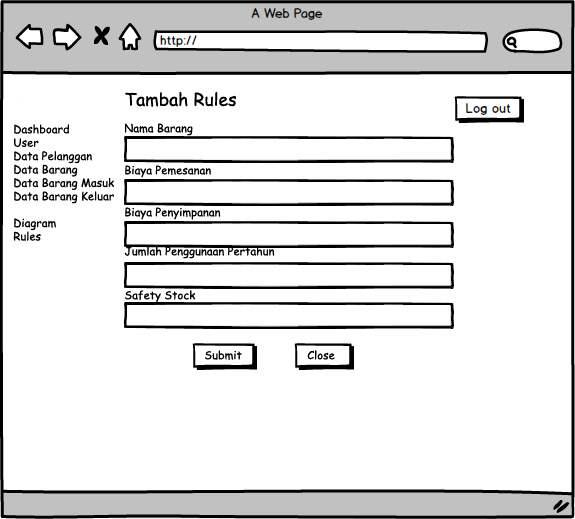
Gambar 4. Halaman data rules

Keterangan :

Tabel 4. Keterangan halaman data rules

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Tombol** | **Keterangan** |
| 1 | *Dashboard* | Digunakan untuk pergi ke halaman menu utama |
| 2 | *User* | Digunakan untuk pergi ke halaman user |
| 3 | Data pelanggan | Digunakan untuk pergi ke halaman data pelanggan |
| 4 | Data barang | Digunakan untuk pergi ke halaman daftar barang |
| 5 | Data barang masuk | Digunakan untuk pergi ke halaman daftar barang masuk |
| 6 | Data barang keluar | Digunakan untuk pergi ke halaman daftar barang keluar |
| 7 | Diagram | Adalah sebuah tampilan yang menggambar- kan barang yang ada pada aplikasi |
| 8 | *Rules* | Digunakan untuk pergi ke halaman daftar *rules* |
| 9 | *Log out* | Digunakan user untuk keluar dari aplikasi dan kembali ke form login |
| 10 | Tambah | Digunakan untuk pergi ke halaman input data *rules* |
| 11 | *Export* | Digunakan untuk pergi ke halaman export ke file (PDF, *excel, word, print*) |
| 12 | Tabel rules | Adalah tampilan tabel yang menunjukan daftar rules yang telah di *input*-kan user |

1. **Halaman Input Rules**



Gambar 4. Halaman input data rules

Keterangan :

Tabel 4. Keterangan halaman input data rules

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Tombol** | **Keterangan** |
| 1 | *Dashboard* | Digunakan untuk pergi ke halaman menu utama |
| 2 | *User* | Digunakan untuk pergi ke halaman user |
| 3 | Data pelanggan | Digunakan untuk pergi ke halaman data pelanggan |
| 4 | Data barang | Digunakan untuk pergi ke halaman daftar barang |
| 5 | Data barang masuk | Digunakan untuk pergi ke halaman daftar barang masuk |
| 6 | Data barang keluar | Digunakan untuk pergi ke halaman daftar barang keluar |
| 7 | Diagram | Adalah sebuah tampilan yang menggambar- kan barang yang ada pada aplikasi |
| 8 | *Rules* | Digunakan untuk pergi ke halaman daftar *rules* |
| 9 | *Log out* | Digunakan user untuk keluar dari aplikasi dan kembali ke form login |
| 10 | Tabel data rules | Adalah tampilan tabel yang menunjukan daftar data rules yang telah di *input*-kan user |
| 11 | Id rules | Digunakan untuk *input* *id rules* |
| 12 | Nama barang | Digunakan untuk *input* nama barang |
| 13 | Biaya pemesanan | Digunakan untuk *input* biaya pemesanan |
| 14 | Biaya penyimpanan | Digunakan untuk *input* biaya penyimpanan |
| 15 | Jumlah penggunaan pertahun | Digunakan untuk *input* jumlah penggunaan pertahun |
| 16 | Safety stock | Digunakan untuk *input* *safety stock* |
| 17 | Submit | Digunakan untuk menyimpan data ke *database* |
| 18 | Close | Digunakan untuk mengosongkan semua bagian yang telah di isi |

# BAB V

# IMPLEMENTASI DAN PENGUJIAN



## 5.1 Implementasi

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

### Implementasi Perangkat Keras dan Perangkat Lunak

Perangkat keras yang penyusun gunakan dalam tahap implementasi aplikasi inventory berbasis *web* pada perusahaan CV.Visya Pratama Industri tidak harus yang berspesifikasi tinggi. Adapun spesifikasi perangkat keras dan perangkat lunak yang digunakan adalah sebagai berikut :

1. Spesifikasi perangkat keras

Tabel 5. Spesifikasi perangkat keras

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Perangkat** | **Spesifikasi** |
| 1 | *Processor* | Intel pentium dual CPU E2160 @ 1,8 Ghz |
| 2 | Ram | 2 x 2 GB DDR 2 |
| 3 | *Harddisk* | 232.88 GB |

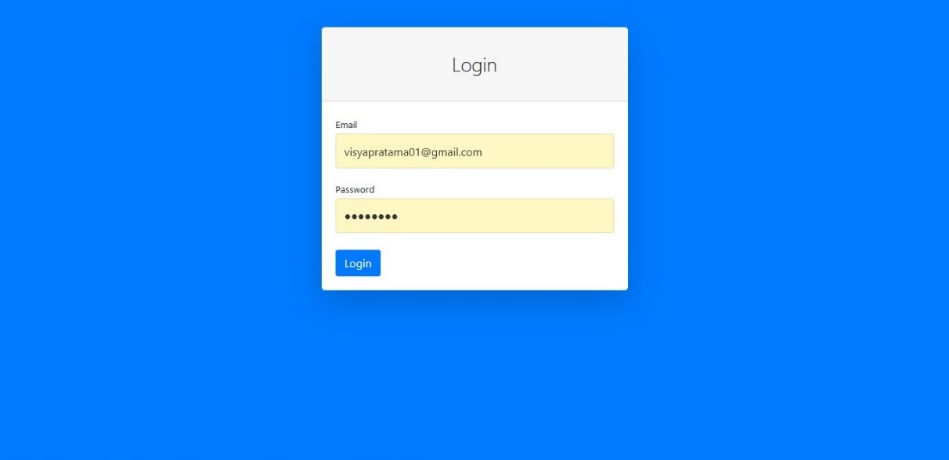
1. Spesifikasi perangkat lunak

Spesifikasi perangkat lunak yang digunakan untuk menjalankan apliksi *inventory* ini adalah XAMPP *Cotrol* *panel* V 3.2.2 dan *Windows* 7 32 bit

## 5.2 Pengujian Sismtem

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

#### Login



Gambar 5. Tampilan halaman log in

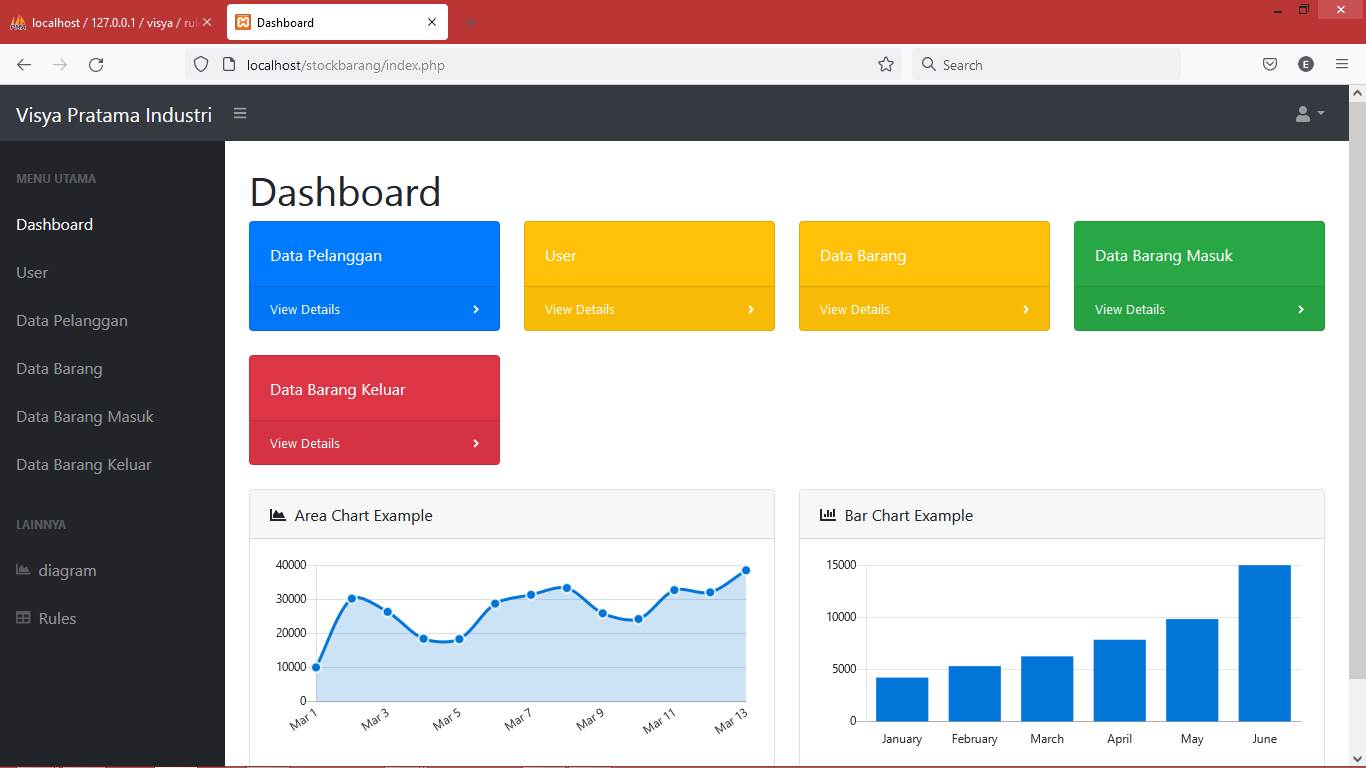
Halaman *login* dibuat untuk mengamankan data user sehingga tidak sembarang orang yang dapat mengaksesnya, terdapat kotak *username* untuk mengisi *username* *user*, kotak *password* untuk mengisi *password* *user*, pilihan *remember* *me* untuk membuat sistem mengingat *username* dan *password* *user*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

Pengujian :

Tabel 5. Keterangan tampilan halaman log in

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Tujuan** | ***Input*** | ***Output* diharapkan** | ***Output* sistem** |
| 1 | Masuk kedalam menu *dashboard* aplikasi | *Username* & *password* | *Login* dan aplikasi menampilkan *dashboard* | \*Sukses  \**Dashboard* tampil sesuai dengan *output* |

1. *Dashboard* Aplikasi



Gambar 5. Tampilan halaman dashboard

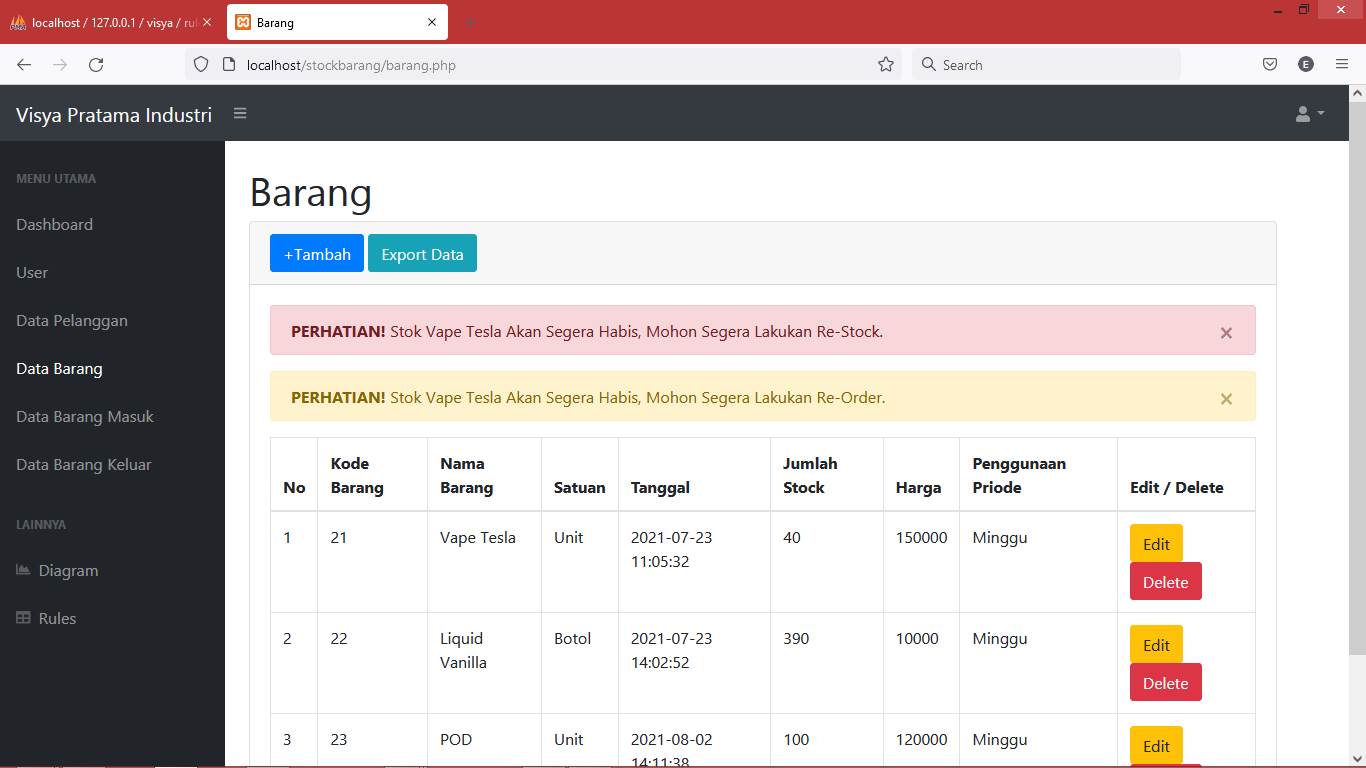
*Dashboard* aplikasi ini merupakan tampilan awal aplikasi pengendalian *invetory* dan di dalam nya terdapat berbagai macam tombol yang daptat diakses oleh penguna aplikasi.

Pengujian :

Tabel 5. Keterangan tampilan halaman dashboard

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Tujuan** | ***Input*** | ***Output* diharapkan** | ***Output* sistem** |
| 1 | Lihat detail pelanggan | Klik data pelanggan pada menu data pelanggan | Menampilkasn detail pelanggan yang ada | \*Sukses  \*Detail pelanggan sesuai dengan *output* |
| 2 | Lihat detail barang | Klik data barang pada menu data barang | Menampilkan detail barang yang ada | \*Sukses  \*Detail barang tampil sesuai dengan *output* |
| 3 | Lihat detail barang masuk | Klik data barang masuk pada menu data barang masuk | Menampilkan detail barang masuk | \*Sukses  \*Detail barang masuk tampil sesuai dengan *output* |
| 4 | Lihat detail barang keluar | Klik data barang keluar pada menu data data barang keluar | Menampilkan detail barang keluar | \*Sukses  \*Detail barang keluar tampil sesuai dengan *output* |
| 5 | Lihat detail *user* | Klik user pada menu user | Menampilkan detail data user | \*Sukses  \*Detail data user tampil sesuai dengan *output* |
| 7 | Lihat detai diagram | Klik diagram pada menu diagram | Menampilkan detail diagram | \*Sukses  \*Detail diagram tampil sesuai dengan *output* |
| 8 | Lihat detail *rules* | Klik rules pada menu rules | Menampilkan detail data rules | \*Sukses  \*Detail data rules tampil sesuai dengan *output* |
| 6 | Menuju menu awal | Klik menu *dashboard* | Menampilkan menu *dashboard* | \*Sukses  \*Detail menu *dashboard*  tampil sesuai dengan *output* |

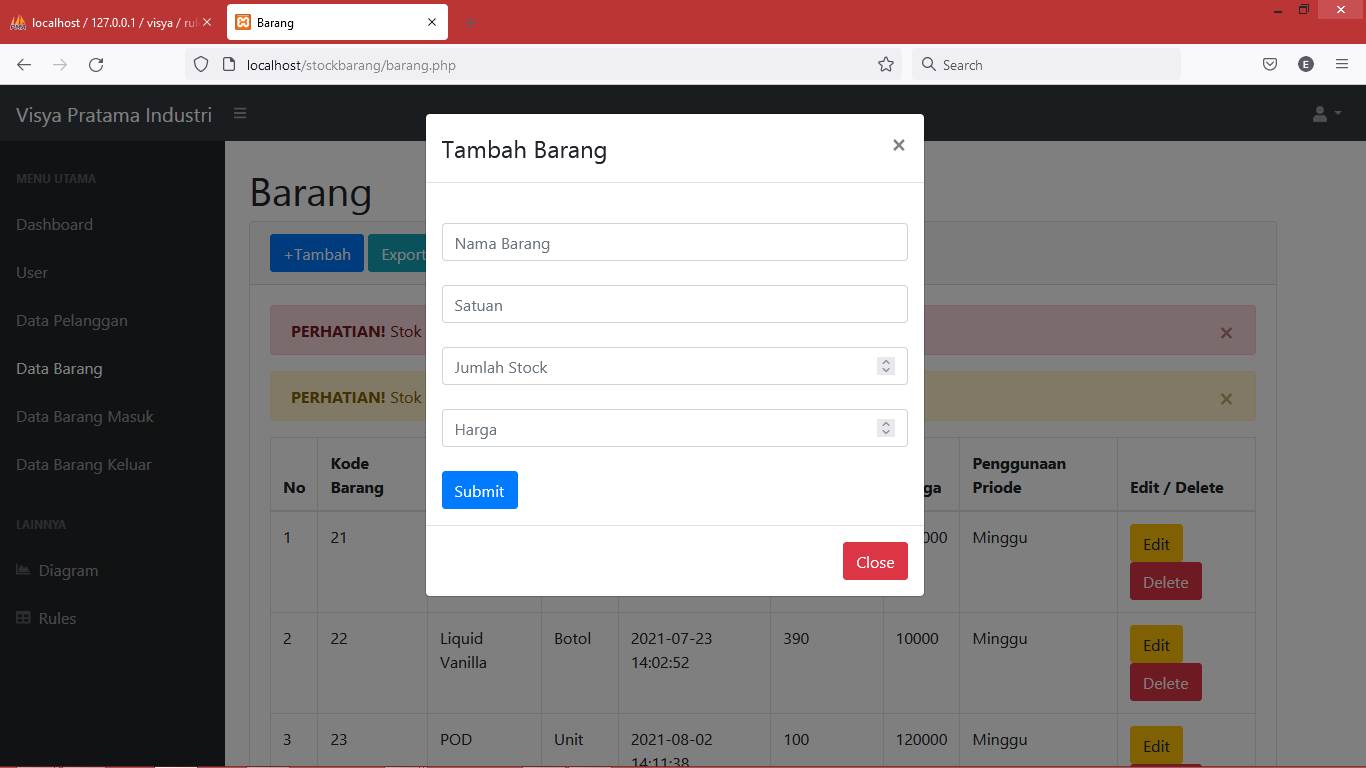
1. Halaman Data Barang



Gambar 5. Tampilan halaman data barang

Setelah user menekan tombol lihat detail pada kotak Data Barang maka sitem akan menapilkan daftar barang yang mana pada daftar tersebut memuat kode barang, nama barang, satuan, tanggal, jumlah stock, harga, penggunaan priode dan edit/delete.

* *Form* *input* data barang



Gambar 5. Tampilan halaman from input data barang

Gambar di atas merupakan *form* *input* data barang yang mana *user* menginputkan nama barang, satuan, jumlah stok dan harga.

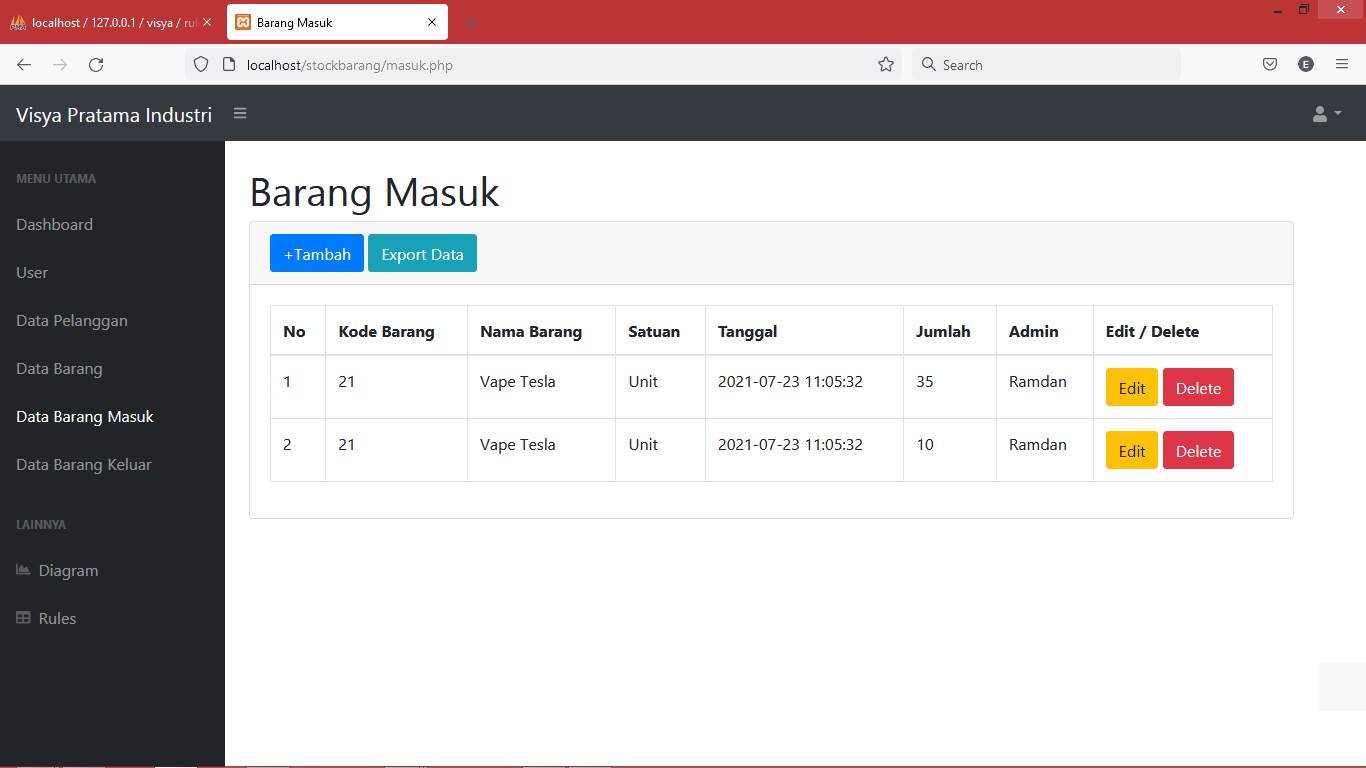
Apa bila tidak ada form input data barang maka beberapa data tidak dapat diolah, sehingga menyebabkan aplikasi sulit untuk dipahami oleh user. Tujuan dari form input data barang adalah agar aplikasi dapat beroperasi seara optimal dan data barang memiliki berbagai macam data diantaranya adalah seperti yang telah disebutkan pada paragraf pertama.

Pengujian :

Tabel 5. Keterangan tampilan halaman data barang dan input data barang

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Tujuan** | ***Input*** | ***Output* diharapkan** | ***Output* sistem** |
| 1 | Melihat detail data barang | Klik lihat detail pada menu data barang yang ada di *dashboard* | Menampilkan data barang yang di *input*-kan user | \*Sukses  \*Data barang tampil pada halaman aplikasi |
| 2 | *Input* data barang baru | Nama barang, satuan, jumlah stock dan harga. | 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. |

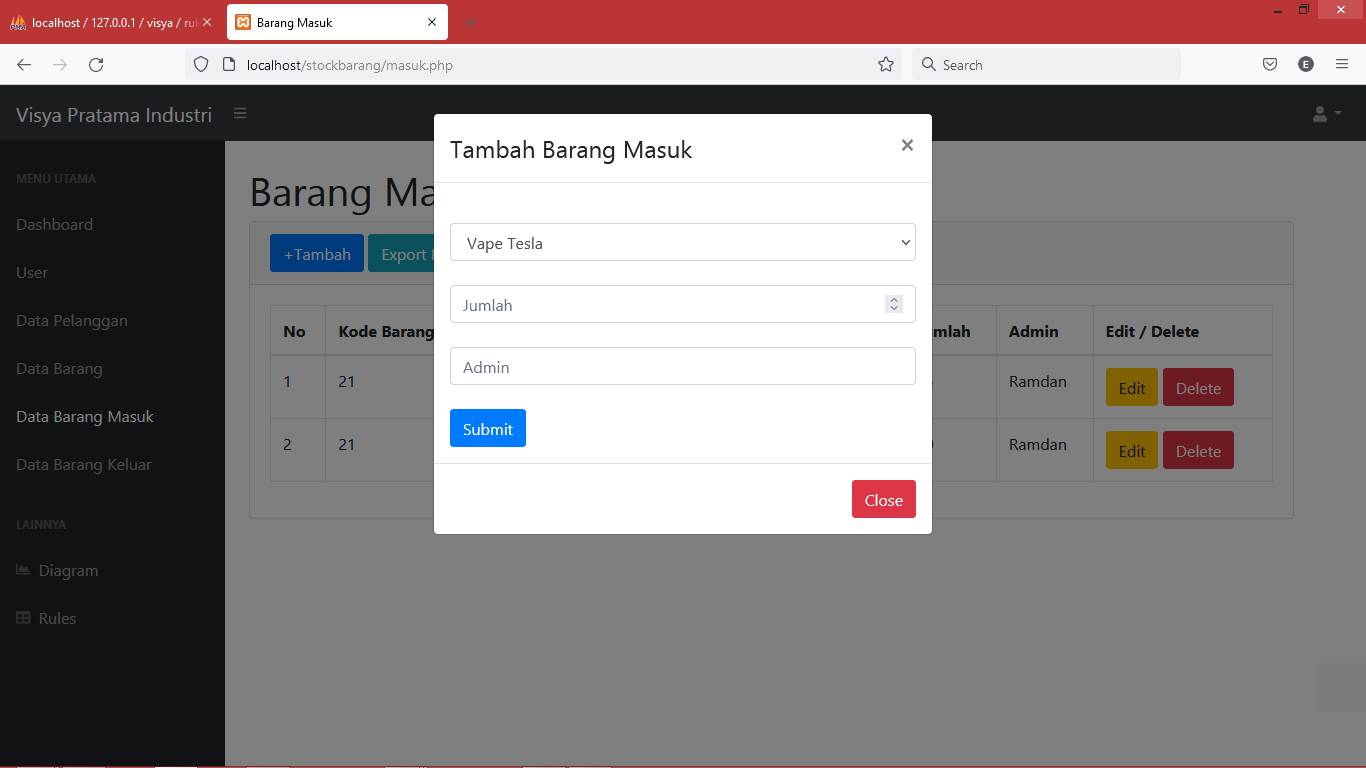
1. Halaman Data Barang Masuk



Gambar 5. Tampilan halaman data barang masuk

Gambar di atas merupakan tampilan daftar data barang masuk yang diinputkan oleh user ke dalam aplikasi.

* *Form* Isi Data Barang Masuk



Gambar 5. Tampilan from isi data barang masuk

Gambar di atas merupakan *form* *input* data barang masuk, yang mana *user* menginputkan nama barang, jumlah dan admin.

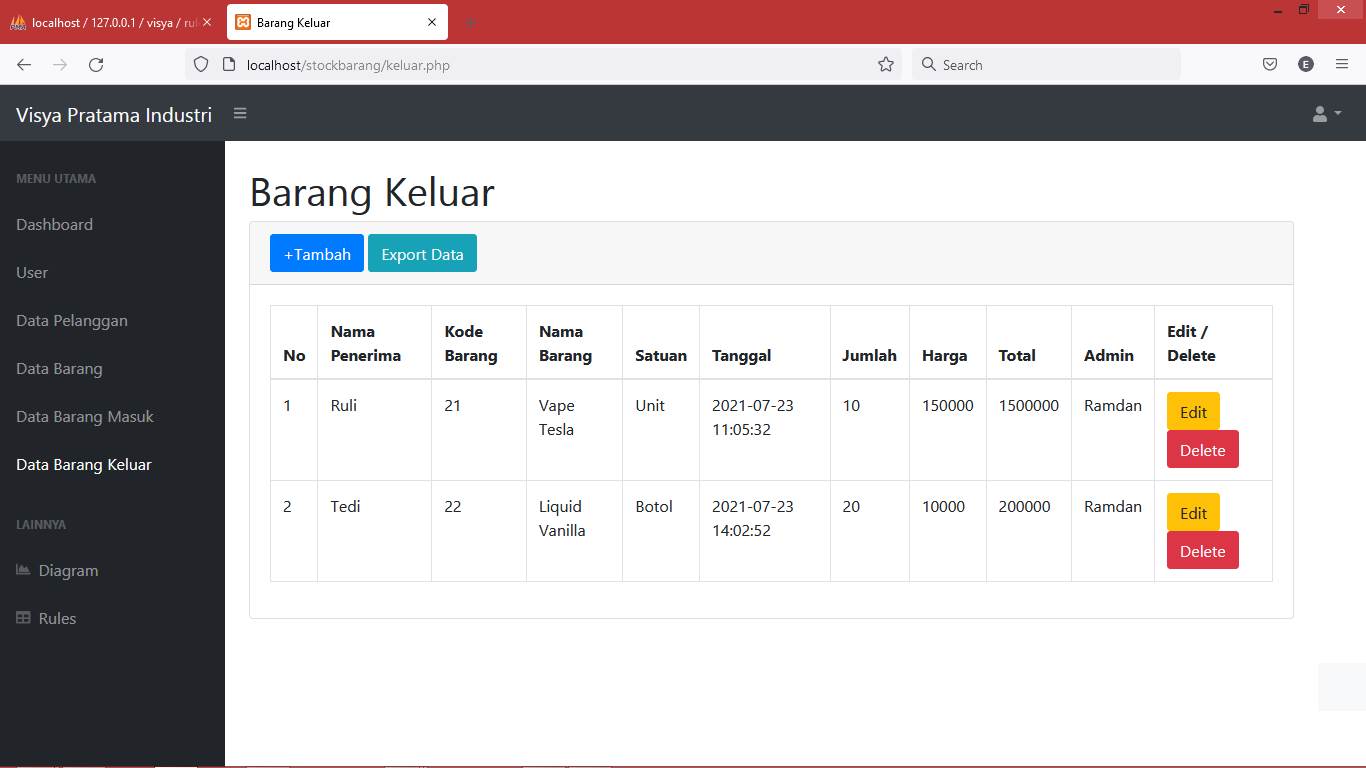
Tujuan dari input data barang masuk adalah agar aplikasi memperoleh data yang akan dikelola dan dapat menampilkan data barang terbaru, apabila ada barang masuk tetapi user tidak meng-input-kan data barang masuk tersebut maka data tidak akan terolah oleh aplikasi, jadi aplikasi akan menampilkan data jumlah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

Pengujian :

Tabel 5. Keterangan data barang masuk dan from isi data barang masuk

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Tujuan** | **Input** | **Output diharapkan** | **Output sistem** |
| 1 | Melihat detail barang masuk | Klik data barang masuk pada menu data barang masuk yang ada di *dashboard,* kemudian klik tombol tambah | Menampilkan data barang masuk yang di inputkan user | \*Sukses  \*Data barang masuk tampil pada halaman aplikasi |
| 2 | *Input* data barang masuk | Nama barang, jumlah dan admin | Data dapat di kelola sistem, data tersimpan ke *database*, dan tampil pada halaman data barang masuk | \*Sukses  \*Data dikelola sistem, tersimpan ke *database* dan tampil pada halaman data barang masuk. |

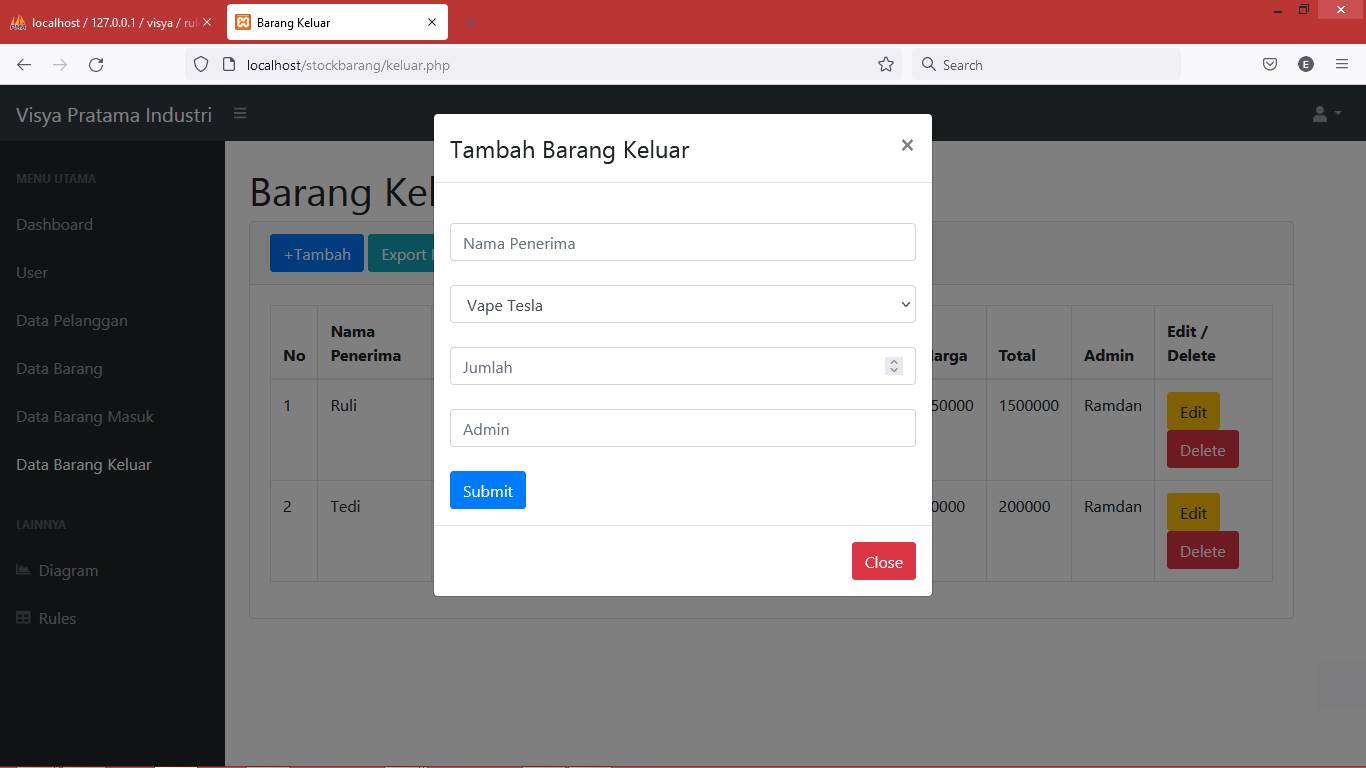
1. Halaman Data Barang Keluar



Gambar 5. Tampilan halaman data barang keluar

Gambar di atas merupakan tampilan daftar data barang keluar yang diinputkankan oleh user ke dalam aplikasi.

* *Form* Isi Data Barang Keluar



Gambar 5. Tampilan halaman from isi data barang keluar

Gambar di atas merupakan *form* input data barang keluar yang mana *user* menginputkan nama penerima, nama barang, jumlah dan admin.

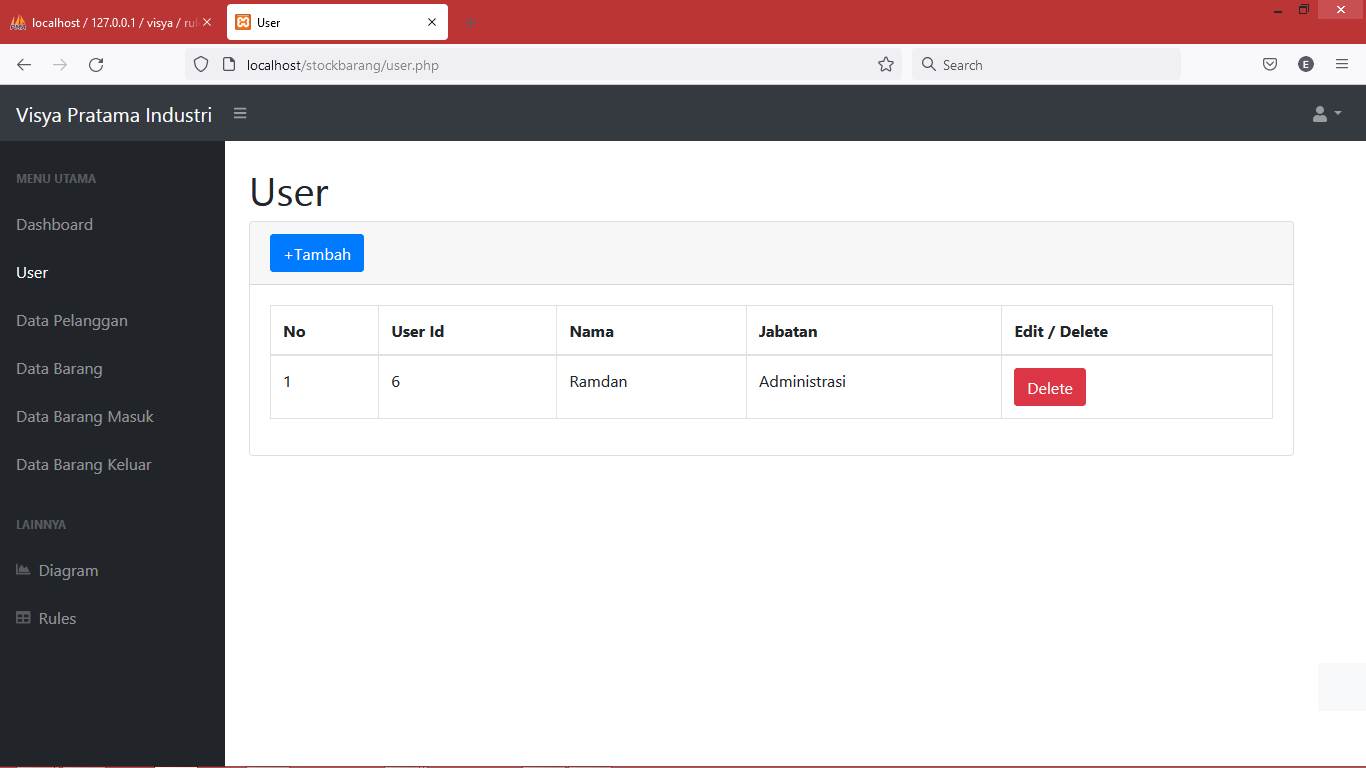
Tujuan dari input data barang keluar adalah agar aplikasi memperoleh data yang akan dikelola dan dapat menampilkan data terbaru, apabila ada barang keluar tetapi user tidak meng-input-kan data barang keluar tersebut maka data tidak akan terolah oleh aplikasi, jadi aplikasi akan menampilkan data barang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

Pengujian :

Tabel 5. Keterangan halaman data barang keluar dan from isi data barang keluar

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Tujuan** | ***Input*** | ***Output* diharapkan** | ***Output* sistem** |
| 1 | Melihat detail barang keluar | Klik data barang keluar pada menu data barang keluar yang ada di *dashboard,* kemudian klik tombol tambah | Menampilkan data barang keluar yang di inputka user | \*Sukses  \*Data barang keluar tampil pada halaman aplikasi |
| 2 | *Input* data barang keluar | Nama penerima, nama barang, jumlah dan admin | Data dapat di kelola sistem, data tersimpan ke *database*, dan tampil pada halaman data barang keluar | \*Sukses  \*Data dikelola sistem, tersimpan ke *database* dan tampil pada halaman data barang keluar. |

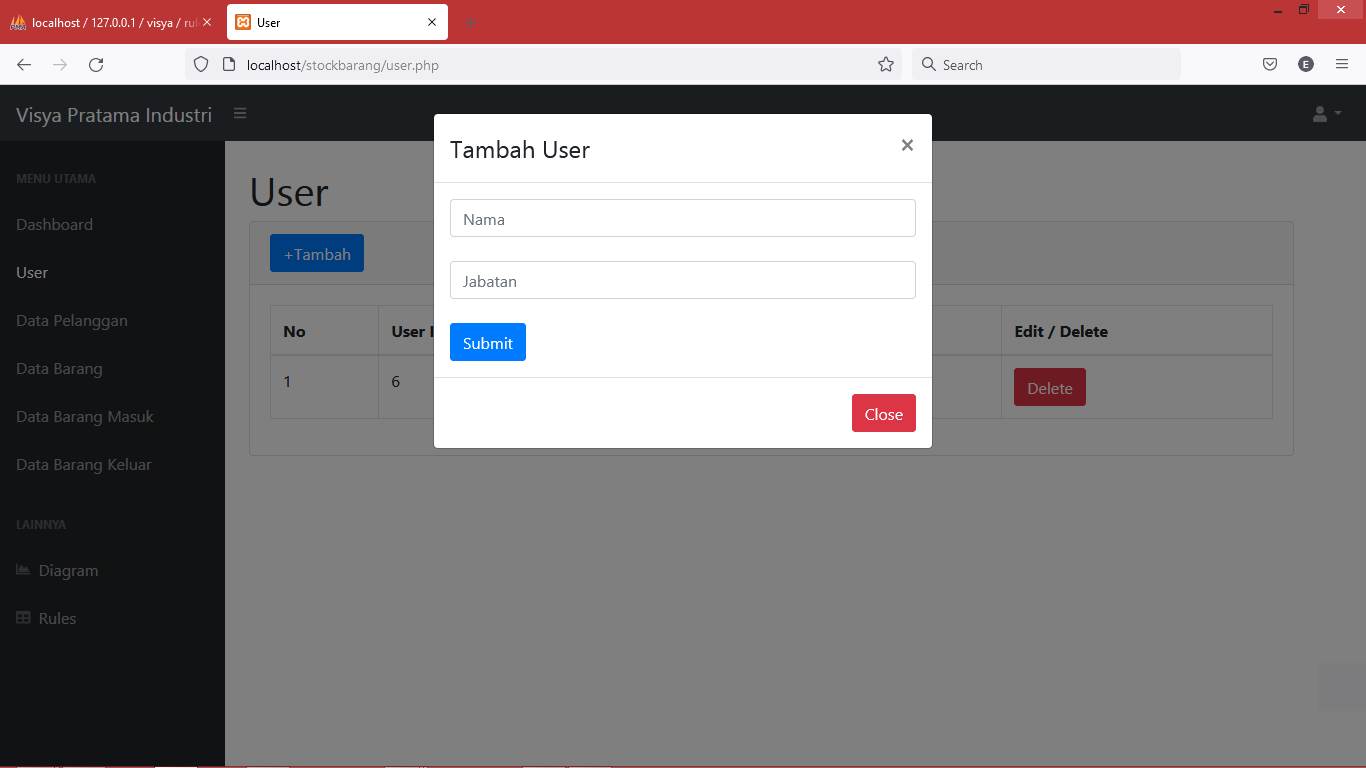
1. Halaman *User*



Gambar 5. Tampilan halaman data user

Gambar di atas merupakan tampilan daftar data *user* yang diinputkan oleh user ke dalam aplikasi.

* *Form* Isi Data User



Gambar 5. Tampilan form isi data user

Gambar di atas merupakan *form* input data *user* yang mana *user* menginputkan nama dan jabatan.

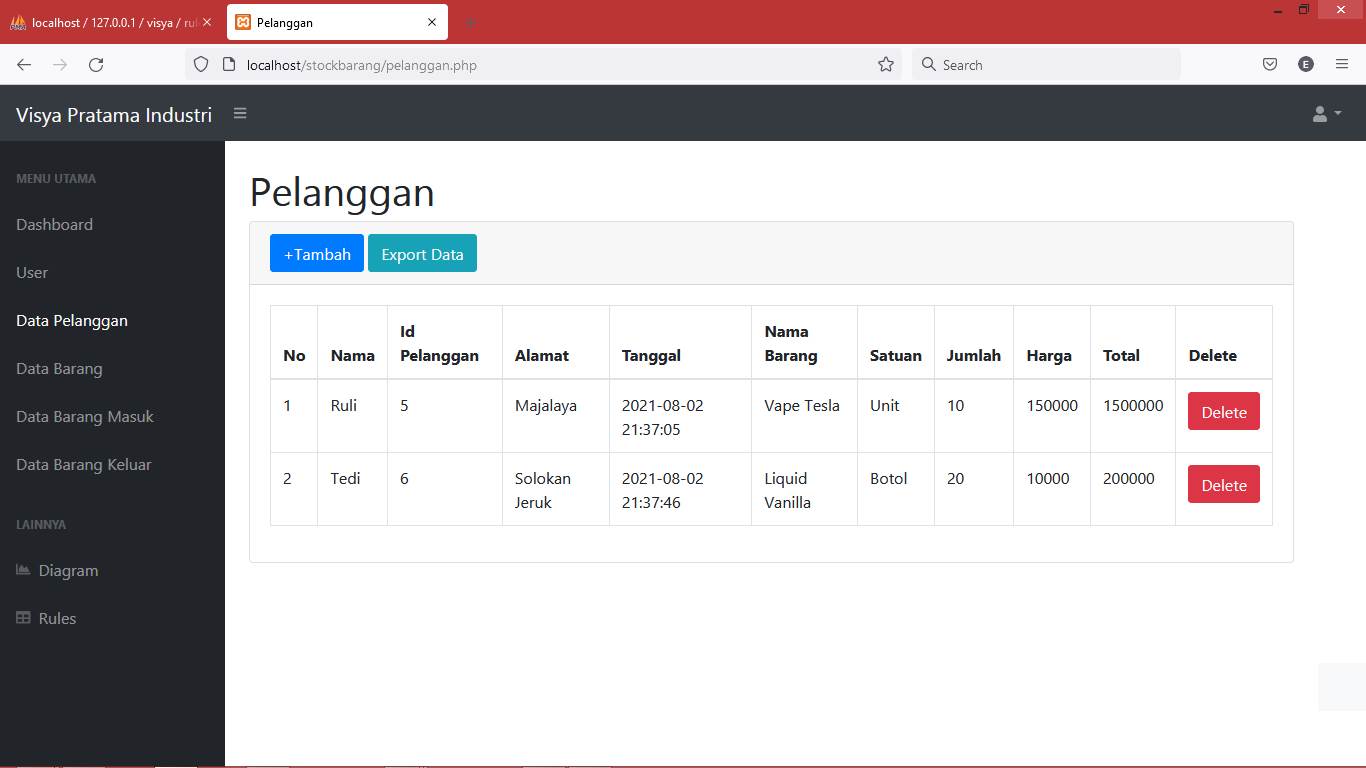
Tujuan dari input data user adalah agar aplikasi memperoleh data user terdaftar.

Pegujian :

Tabel 5. Keterangan tampilan halaman data user dan tampilan form isi data user

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Tujuan** | ***Input*** | ***Output* diharapkan** | ***Output* sistem** |
| 1 | Melihat detail data user | Klik data user pada menu data user yang ada di *dashboard,* kemudian klik tombol tambah | Menampilkan data user yang di inputka user | \*Sukses  \*Data user tampil pada halaman aplikasi |
| 2 | *Input* data *user* | Nama dan jabatan | Data dapat di kelola sistem, data tersimpan ke *database*, dan tampil pada halaman data user | \*Sukses  \*Data dikelola sistem, tersimpan ke *database* dan tampil pada halaman data user. |

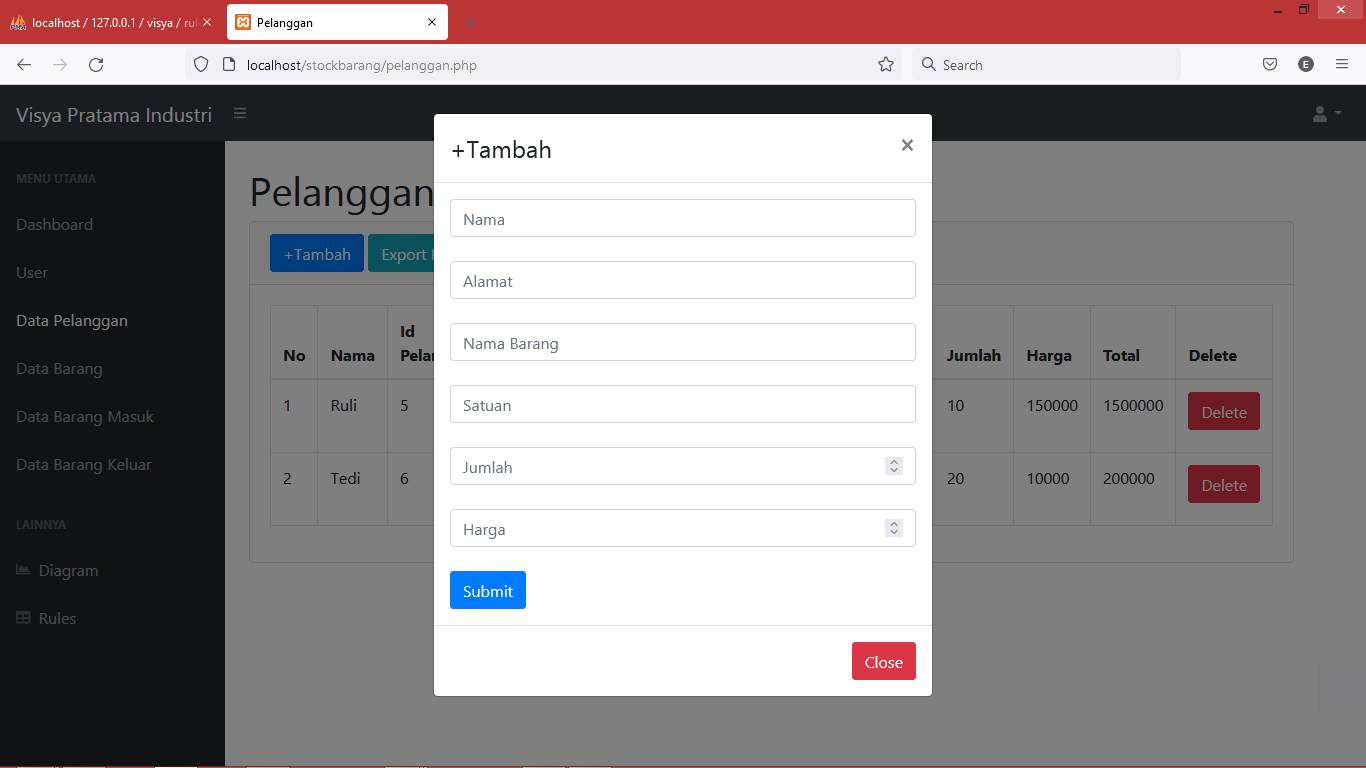
1. Halaman Data Pelanggan



Gambar 5. Tampilan halaman data pelanggan

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

* *From* Isi Data Pelanggan



Gambar 5. Tampilan halaman from isi data pelanggan

Gambar di atas merupakan *form* input data pelanggan yang mana *user* menginputkan nama, alamat, nama barang, satuan, jumlah dan harga.

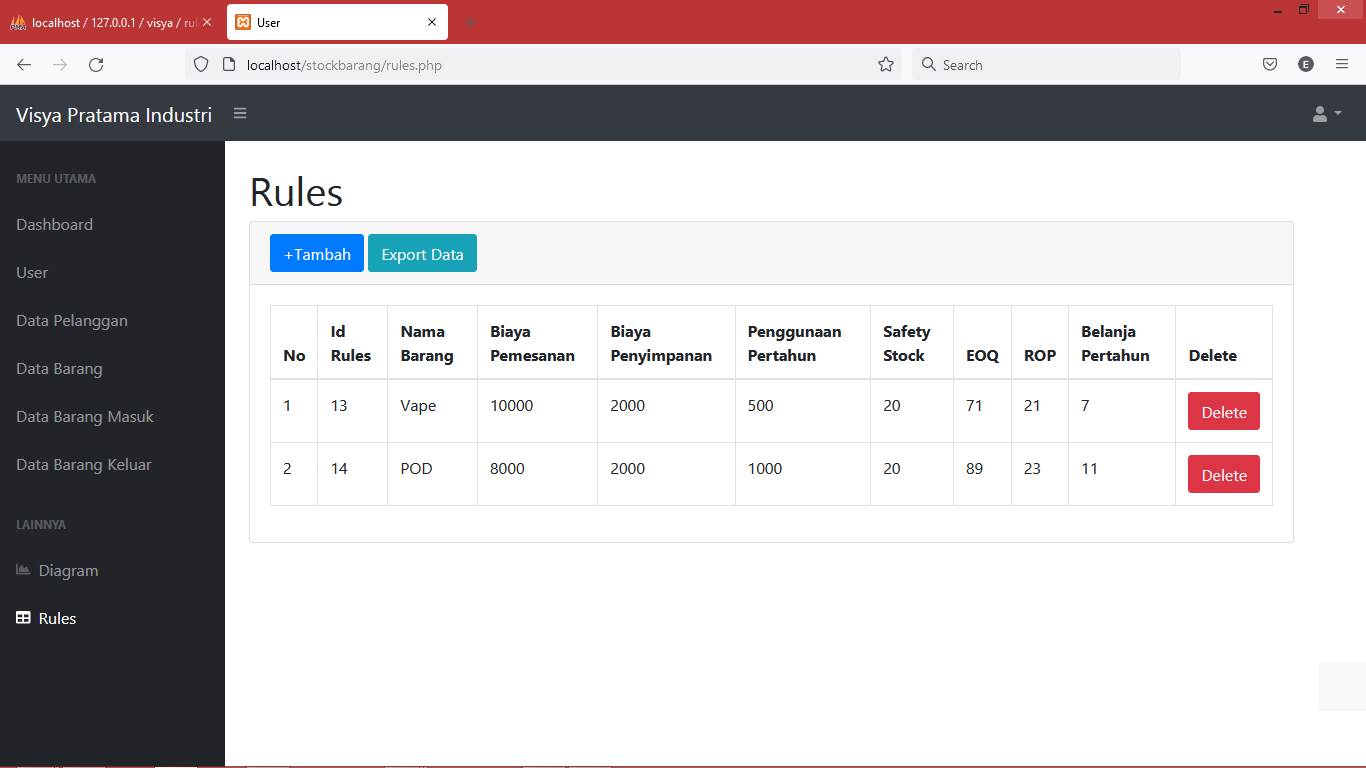
Tujuan dari input data pelanggan adalah agar aplikasi memperoleh data pelanggan apakah semakin naik atau turun jumlah pelanggan terdaftar.

Pengujian :

Tabel 5. Keterangan tampilan halaman data pelangggan dan form isi data pelanggan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Tujuan** | ***Input*** | ***Output* diharapkan** | ***Output* sistem** |
| 1 | Melihat detail data pelanggan | Klik data pelanggan pada menu data pelanggan yang ada di *dashboard* | Menampilkan data pelanggan yang di inputka user | \*Sukses  \*Data pelanggan tampil pada halaman aplikasi |
| 2 | *Input* data data pelanggan | Nama, alamat, nama barang, satuan, jumlah dan harga | Data dapat di kelola sistem, data tersimpan ke *database*, dan tampil pada halaman data pelanggan | \*Sukses  \*Data dikelola sistem, tersimpan ke *database* dan tampil pada halaman data pelanggan. |

1. Halaman Data Rules

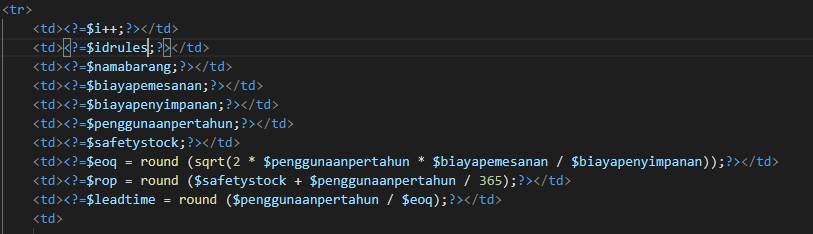


Gambar 5. Halaman data rules

Gambar di atas merupakan tampilan daftar data *rules* yang diinputkan oleh user ke dalam aplikasi.

* Implementasi Metode EOQ dan ROP

Sebagaimana yang tertera pada halaman judul tugas akhir ini, penyusun menggunakan metode *Economic order Quantity* (EOQ) dan *Re-order Point* (ROP) untuk mengelola inventory pada perusahaan. Maka dari itu penyusun menyisipkan rumus EOQ dan ROP pada aplikasi yang di buat.



Gambar 5. Implementasi metode EOQ dan ROP

Pada gambar di atas dapat dilihat penyusun menggunakan rumus EOQ dan ROP, terdapat rumus ROP : **SS + (D/365)**, dan terdapat rumus EOQ : **{(2D x S) / H}**. setelah memastikan rumus pada aplikasi selanjutnya adalah pembuktian penghitungan EOQ dan ROP pada aplikasi.



Gambar 5. Penghitungan EOQ dan ROP

Pada gambar di atas dapat dilihat aplikasi menampilkan ROP : 21 dan EOQ : 71 untuk vape. Berikut adalah penghitungan secara manual :

1. Menghitung EOQ

Rumus : **{(2D x S) / H}**.

**D** = Penggunaan *product* pertahun, **S** = biaya pemesanan, **H** = biaya penyimpanan perperiode.

D = 500, S = 10.000, H = 2.000

EOQ = √(2) x (500) x (10.000) : (2.000)= = 70.710678118655 di bulatkan total menjadi “**71**” untuk EOQ

1. Menghitung ROP

Rumus : **SS + (D/365)**

**SS** = *Safety Stock,* **D** = Penggunaan / Tahun

SS = 20, D = 500 pcs

20 + (500 / 365) = 21.369863013699 di bulatkan total menjadi “**21**” untuk ROP

1. Menghitung belanja pertahun

Rumus : **D / EOQ**

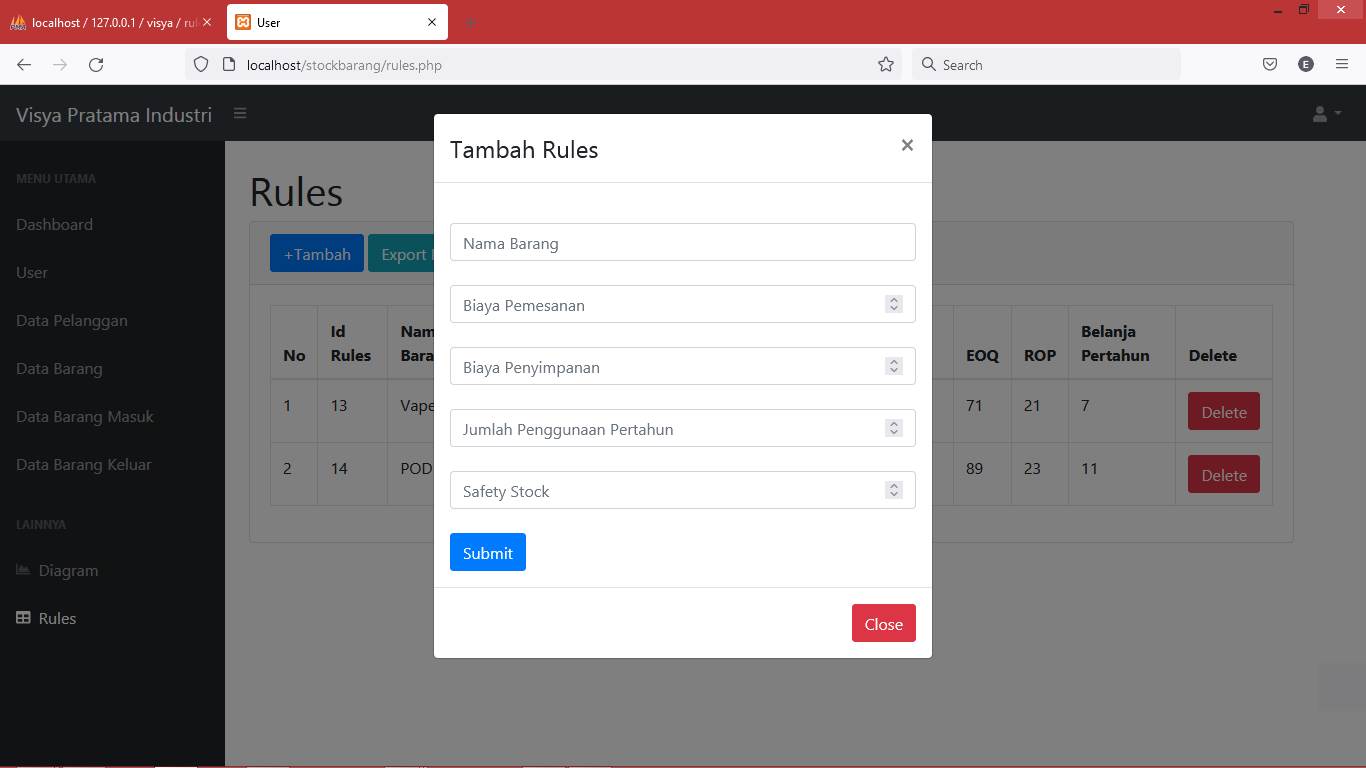
D = Penggunaan pertahun, EOQ

D = 500, EOQ = 71

500 : 71 = 7.0422535211268 di bulatkan total menjadi “**7**” untuk pelanja pertahun

Keterangan : jadi perusahan dapat belanja sebanyak 7 kali dalam satu tahun dengan jumlah belanja 71 unit dalam setiap melakukan belanja untuk memenuhi kebutuhan perusahaan.

* *Form* Isi Data Rules



Gambar 5. Tampilan form isi data rules

Gambar di atas merupakan *form* input data rules yang mana *user* menginputkan nama barang, biaya pemesanan, biaya penyimpanan, jumlah peggunaan pertahun dan *safety stock*.

Tujuan dari input data rules adalah agar aplikasi memperoleh data rules.

Pengujian :

Tabel 5. Keterangan tampilan halaman data rules dan form isi data rules

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Tujuan** | ***Input*** | ***Output* diharapkan** | ***Output* sistem** |
| 1 | Melihat detail data rules | Klik data rules pada menu data rules yang ada di *dashboard* | Menampilkan data rules yang di inputka user | \*Sukses  \*Data rules tampil pada halaman aplikasi |
| 2 | *Input* data data rules | Nama barang, biaya pemesanan, biaya penyimpanan, jumlah penggunaan pertahun dan *safety stock* | Data dapat di kelola sistem, data tersimpan ke *database*, dan tampil pada halaman data rules | \*Sukses  \*Data dikelola sistem, tersimpan ke *database* dan tampil pada halaman data rules. |

# 

# BAB VI

# KESIMPULAN DAN SARAN

## Kesimpulan

Berdasarkan penelitian yang dilakukan penulis melalui beberapa tahapan yang dilakukan pada bab-bab sebelumya maka penulis dapat menyimpulkan :

1. Dengan terselesaikannya penelitian ini peulis dapat membuat suatu aplikasi yang mana tujuan dari aplikasi tersebut adalah memudahkan perusahaan dalam mengeola *inventory*.
2. Dari uji coba yang dilakukan pada perusahaan, penulis dapat menyipulkan bahwa aplikasi ini dapat mengefisiensikan pegawai perusahaan dalam mengelola *inventory* dengan menggunakan metode EOQ dan ROP.
3. Dengan adanya aplikasi ini penulis dapat memahami bagimana cara megelola *inventory* pada perusahaan, yang mana melalui *inventory* berbasis *web* perusahaan dapat mengetahui jumlah barang yang tersisa, jumlah barang masuk dan keluar melalui aplikasi.

## Saran

Dalam proses pembuatan Aplikasi *Inventory* Berbasis *Web* Pada CV.Visya Pratama Industri penulis masih banyak sekali memiliki kekurangan dan jauh dari sempurna, sehingga penulis berharap untuk peneliti selanjutnya dapat mengembangkan lagi aplikasi ini, diantaranya :

1. Aplikasi ini masih memiliki kekurangan dalam mengolah data user maka alangkah lebih baik jika dapat dikembangkan lagi sehinga dapat lebih memudahkan dalam menampilkan identitas user yang mengelola aplikasi ini.
2. Aplikasi ini dapat ditambahkan fitur ataupun performansinya, selain itu dapat dikembangkan menjadi berbasis mobile dengan bahasa pemrograman lain.
3. Aplikasi ini membutuhkan *hardware* pedukung seperti *keyboard, mouse*, dan lain sebagainya.

# DAFTAR PUSTAKA

Agus Saputra. (2013). *Smarty PHP OOP Engine for PHP Template.* Bandung: PT.Elex Media Computindo.

Gita Ayu Syafarina. (2016) Perancangan Aplikasi Inventory Barang Materials dan Product.

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.*

Maulana Hasanudin. (2018) Rancang dan Bangun Sistem Informasi Inventory Barang Berbasis Web (Studi Kasus PT.Nusantara Sejahtera Raya).

Nurhadi. (2017). Pondasi Dasar Pemograman *Website*.

R. S. Presman. (2010). *Software Engineering A Practioner's approach 7th ed.* MC Grow Hill.

Rahmawati. (2017) Sistem Informasi InventoryStok Barang Pada CV.Artha Palembang.

Tri Muhamad Hani. (2011). Metode EOQ (Econonomic Order Quantity).

Robi Muhamad. (209). Makalah Aplikasi Komputer. .

Wijaya. (2009). *Model* *Driven* *Developement* .

**LAMPIRAN**