

**RANCANG BANGUN SISTEM INFORMASI PREDIKSI PENGAMBILAN STOK BARANG OLEH DISTRIBUTOR PADA PT. IRAWAN DJAJA AGUNG JEMBER MENGGUNAKAN METODE *JUST IN TIME***

**SKRIPSI**

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan Program Studi Sistem Informasi (S1) dan mendapatkan gelar Sarjana Komputer

Oleh

**Rizky Ila Khoiriyah**

**NIM 112410101093**

**PROGRAM STUDI SISTEM INFORMASI**

**UNIVERSITAS JEMBER**

**2016**

# PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

# MOTTO

# PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : Rizky Ila Khoiriyah

NIM : 112410101093

menyatakan dengan sesungguhnya bahwa karya tulis ilmiah dengan judul “Rancang Bangung Sistem Informasi Prediksi Pengambilan Stok Barang Oleh Distributor Pada PT. Irawan Djaja Agung Jember Menggunakan Metode *Just In Time*”adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada instansi manapun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak mana pun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

|  |  |
| --- | --- |
|  | Jember,  Yang menyatakan,  Rizky Ila Khoiriyah  NIM.112410101093 |

# SKRIPSI

**RANCANG BANGUN SISTEM INFORMASI PREDIKSI PENGAMBILAN STOK BARANG OLEH DISTRIBUTOR PADA PT. IRAWAN DJAJA AGUNG JEMBER MENGGUNAKAN METODE *JUST IN TIME***

Oleh

Rizky Ila Khoiriyah

NIM 112410101093

Pembimbing

|  |  |  |
| --- | --- | --- |
|  |  | |
| Dosen Pembimbing Utama | : | Prof. Drs. Slamin, M.Comp. Sc., Ph.D |
|  |  | |
| Dosen Pembimbing Anggota | : | Nelly Oktavia A, S. Si, MT. |

# PENGESAHAN PEMBIMBING

Skripsi berjudul “Rancang Bangung Sistem Informasi Prediksi Pengambilan Stok Barang Oleh Distributor Pada PT. Irawan Djaja Agung Jember Menggunakan Metode *Just In Time*”, telah diuji dan disahkan pada:

hari, tanggal :

tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh :

|  |  |  |
| --- | --- | --- |
| Pembimbing I, |  | Pembimbing II, |
|  |  |  |
| Prof. Drs. Slamin, M.Comp.Sc., Ph.D |  | Nelly Oktavia A, S. Si, MT. |
| NIP. 196704201992011001 |  | NIP. |

# PENGESAHAN

Skripsi berjudul “Rancang Bangung Sistem Informasi Prediksi Pengambilan Stok Barang Oleh Distributor Pada PT. Irawan Djaja Agung Jember Menggunakan Metode *Just In Time*”, telah diuji dan disahkan pada:

hari, tanggal :

tempat : Program Studi Sistem Informasi Universitas Jember

|  |  |  |
| --- | --- | --- |
| Penguji 1, |  | Penguji 2, |
|  |  |  |
|  |  |  |
| NIP. |  | NIP. |

Mengesahkan

Ketua Program Studi,

Prof. Drs. Slamin, M.Comp.Sc., Ph.D

NIP. 196704201992011001

# RINGKASAN

# PRAKATA

# DAFTAR ISI

# DAFTAR TABEL

# DAFTAR GAMBAR

# BAB 1. PENDAHULUAN

Bab ini merupakan langkah awal dari penulisan skripsi ini. Bab ini berisi penjelasan mengenai latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah, dan sistematika penulisan.

## Latar Belakang

PT. Irawan Djaja Agung adalah sebuah perusahaan yang bergerak pada bidang pendistributor beberapa produk farmasi yang meliputi kosmetik, obat-obatan, makanan, dan minuman di seluruh wilayah Indonesia. Sebagai salah satu perusahaan yang bergerak pada bidang distributor, sangatlah pasti selalu berhubungan dengan produsen, konsumen, jaringan distribusi logistik, dan pusat distribusi atau gudang.

Perusahaan yang mengantarkan produknya ke pelanggan atau customer selalu menggunakan jaringan distribusi logistik. Dimana sebuah jaringan distribusi yang terdiri dari aliran produk dari produsen ke konsumen melalui titik-titik pemindahan, pusat distribusi (gudang), dan pengecer. Peranan jaringan distribusi dan manajemennya merupakan hal yang sangat penting bagi perusahaan untuk meningkatkan penjualan dan keuntungan.

Dinamika lingkungan dimana rantai pasokan berkembang dan tuntutan untuk memperpendek masa siklus distribusi produk mewajibkan perusahaan merancang ulang jaringan distribusi logistik (Morales, 2001). Beberapa masalah yang harus diperhatikan perusahaan adalah waktu produksi, lokasi persediaan, dan penempatan gudang untuk pelanggan. Keputusan yang diambil harus memperhatikan semua faktor dan dikoordinasikan dengan semua rantai pasokan demi terciptanya efisiensi. Koordinasi ini terutama diperlukan dalam lingkungan yang dinamis dimana pengaturan jaringan distribusi logistik kadang berubah secara signifikan dari perencanaan awal.

Jaringan distribusi logistik diibaratkan terdiri dari set fasilitas, yang masing-masing terdiri dari satu pabrik produksi dengan sebuah gudang yang terhubung, dan satu set pelanggan. Masing-masing pabrik dengan kapasitas yang sudah diketahui dan terbatas. Dan setiap pelanggan ditempatkan atau dihubungkan ke fasilitas dengan perencanaan tertentu karena permintaan pelanggan biasanya membentuk pola musiman. Karena setiap gudang dihubungkan dengan pabrik tertentu, diasumsikan bahwa biaya transportasi antara pabrik dan gudang termasuk dalam biaya produksi (Morales, 2001).

Tidak hanya biaya transpotasi yang menjadi permasalahan pada PT. Irawan Djaja Agung ini, melainkan banyaknya biaya tak terduga meliputi biaya perawatan produk dalam gudang, pajak, asuransi, dan lain sebagainya sehingga mengakibatkan perolehan laba yang kurang maksimal. Selain itu alokasi persediaan yang kurang tepat mengakibatkan biaya-biaya tersebut semakin bertambah. Oleh sebab itu kami menggunakan metode *Just In Time* untuk menekan biaya tak terduga tersebut.

*Just In Time* adalah suatu konsep dimana bahan baku yang akan digunakan untuk aktifitas produksi didatangkan dari pemasok atau suplier tepat pada waktu bahan itu dibutuhkan oleh proses produksi, sehingga akan sangat menghemat bahkan mentiadakan biaya persediaan barang atau penyimpanan barang, laba yang diperoleh juga diharapkan akan lebih meningkat.

Berdasarkan pembahasan di atas, penulis mencoba membuat sistem informasi peramalan pengambilan stok oleh distributor pada perusahaan dagang mengunakan metode *Just In Time* yang akan membantu perusahaan meramalkan berapa banyak stok minimal yang harus dialokasikan oleh distributor menggunakan metode *Just In Time* dan berapa banyak barang yang dapat diambil oleh distributor. Dengan adanya sistem ini, penulis berharap dapat membantu perusahaan dalam meramalkan stok pada perusahaan dan membantu pihak manajemen dalam mengambil keputusan.

## Rumusan Masalah

Berdasarkan uraian pada latar belakang yang telah dipaparkan di atas, dapat dirumuskan permasalahan dalam penelitian ini yaitu:

1. Bagaimana pengaruh pemilihan metode pengendalian persediaan menggunakan metode *Just In Time* terhadap tingkat efisiensi pengambilan stok barang pada PT. Irawan Djaja Agung Jember?
2. Bagaimana merancang dan membangun sistem informasi prediksi pengambilan stok minimal barang metode *Just In Time*?

## Tujuan dan Manfaat

Tujuan dan manfaat merupakan jawaban dari rumusan masalah yang telah diuraikan diatas serta tujuan yang ingin dicapai pada penelitian ini.

### Tujuan

Berdasarkan rumusan masalah yang telah disampaikan sebelumnya maka dapat ditetapkan tujuan yang ingin dicapai dari penelitian ini, yaitu merancang dan membangun sebuah aplikasi untuk memprediksi pengambilan stok minimal barang oleh distributor menggunakan metode *Just In Time* pada PT. Irawan Djaja Agung, untuk mencapai tujuan tersebut penelitian menggunakan model *waterfall* sebagai metode pengembangan aplikasi dalam melakukan penelitian.

### Manfaat

Manfaat penelitian ini adalah :

1. Manfaat Akademis

Hasil penelitian ini diharapkan dapat memberikan informasi dan studi literatur bagi dunia pendidikan, khususnya di bidang sistem informasi.

1. Manfaat bagi perusahaan

Sistem ini bermanfaat bagi perusahaan dalam memprediksi stok barang yang akan diambil selanjutnya, dengan menerapkan metode *Just In Time* diharapkan perhitungan yang dilakukan akan lebih efisien.

1. Manfaat bagi penulis

Penulis dapat mengetahui bagaimana proses penerapan metode *Just In Time* dalam peramalan stok minimal barang.

## Batasan Masalah

1. Aplikasi hanya mencatat data barang masuk, data barang keluar, data barang retur,dan prediksi pengambilan stok minimal barang selanjutnya.
2. Prediksi pengambilan stok barang dalam aplikasi ini menggunakan metode *Just In Time*.
3. Pengujian pada aplikasi hanya dilakukan pada bagian metode/pada fitur prediksi pengambilan stok minimal barang.

## Sistematika Penulisan

Sistematika penulisan dalam penyusunan tugas akhir ini adalah sebagai berikut :

1. Pendahuluan

Bab ini terdiri atas latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah serta sistematika penulisan.

1. Tinjauan Pustaka

Bab ini berisi tentang kajian materi dan informasi apa saja yang digunakan dalam penelitian.

1. Metodologi Penelitian

Bab ini menguraikan tentang metode-metode yang digunakan untuk menyelesaikan tujuan penelitian.

1. Desain dan Implementasi

Bab ini menguraikan tentang desain dan implementasi desain kedalam bahasa pemrograman untuk membangun aplikasi.

1. Hasil dan Pembahasan

Bab ini membahas tentang hasil dari penelitian yang telah dilakukan, dengan memaparkan hasil penelitian dan hasil percobaan pengimplementasian aplikasi.

1. Penutup

Bab ini berisi kesimpulan dari penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya.

# BAB 2. TINJAUAN PUSTAKA

Bab ini memaparkan teori-teori serta sumber pustaka yang dipakai pada saat penelitian dilakukan. Teori-teori diambil dari buku literatur, jurnal maupun website yang sesuai.

## Aplikasi

Aplikasi merupakan penggunaan dalam suatu komputer, instruksi *(instruction)* atau pernyataan *(statement)* yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi output (Ipulhe, 2014). Definisi lain menurut Hengky W.Pramana, aplikasi adalah suatu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti sistem perniagaan, game palayanan masyarakat, periklanan, atau semua proses yang hampir dilakukan manusia (Setyawan, 2012)

Penjelasan diatas dapat disimpulkan bahwa aplikasi merupakan suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna.

## Prediksi

Pengertian prediksi adalah sama dengan ramalan atau perkiraan. Menurut Kamus Besar Bahasa Indonesia, prediksi adalah hasil dari kegiatan memprediksi atau meramalkan atau memperkirakan (Antonio, 2013)

Dalam dunia usaha sangat diperlukan hal-hal yang dimasa depan sebagai dasar untuk pengambilan keputusan. Maka sangat penting dilakukannya prediksi atau permalan. Prediksi atau peramalan adalah suatu usaha untuk memprediksi keadaan di masa mendatang melalui pengujian keadaan di masa lalu (Handoko, 1999). Dalam dunia perdagangan aktivitas peramalan merupakan suatu fungsi bisnis yang berusaha memperkirakan penjualan dan penggunaan produk sehingga produk tersebut dapat dibuat dalam kuantitas yang tepat (Gaspersz, 2005)

## Persediaan

Persediaan adalah aset yang tersedia untuk dijual dalam kegiatan usaha biasa, entah dalam proses produksi untuk penjualan atau dalam bentuk bahan atau perlengkapan untuk digunakan dalam proses produksi atau pemberian jasa.(Ikatan Akuntan Indonesia, 2001)

Persedian juga meliputi segala macam barang yang menjadi objek pokok aktivitas perusahaan yang tersedia untuk diolah dalam proses produksi atau dijual. (Syakur, 2009)

Sedangkan menurut Keiso, Weygant dan Warfield(2007) pengertian persediaan adalah pos-pos aktiva yang dimiliki oleh perusahaan untuk di jual dalam operasi bisnis normal, atau barang yang akan di gunakan atau di konsumsi dalam membuat barang yang akan dijual.

Menurut Ristono (2009) persediaan dapat diartikan sebagai barang-barang yang disimpan untuk digunakan atau dijual pada masa atau periode yang akan datang. Persediaan terdiri dari persediaan bahan baku, persediaan bahan setengah jadi dan persediaan barang jadi. Persediaan bahan baku dan bahan setengah jadi disimpan sebelum digunakan atau dimasukkan ke dalam proses produksi, sedangkan persediaan barang jadi atau barang dagangan disimpan sebelum dijual atau dipasarkan. Dengan demikian setiap perusahaan yang melakukan kegiatan usaha umumnya memiliki persediaan.

Perusahaan yang melakukan kegiatan produksi (industry manufaktur) akan memiliki tiga jenis persediaan, yaitu :

1. Persediaan bahan baku dan penolong
2. Persediaan bahan setengah jadi
3. Persediaan barang jadi

Sedangkan perusahaan perdagangan minimal memiliki satu jenis persediaan, yaitu persediaan barang dangang. Adanya berbagai macam persediaan ini menuntut pengusaha untuk melakukan tindakan yang berbeda untuk masing-masing persediaan, dan ini akan sangat terkait dengan permasalahan lain seperti masalah prediksi kebutuhan bahan baku serta prediksi penjualan atau permintaan konsumen. Apabila terjadi kesalahan dalam menetapkan besarnya persediaan maka akan berdampak ke masalah yang lain, misalnya tidak terpenuhinya permintaan konsumen atau bahkan berlebihan persediaan sehingga tidak semuanya terjual, timbulnya biaya ekstra penyimpanan atau pesanan bahan dan sebagainya.

## Metode *Just In Time*

Filosofi *Just In Time* didasarkan pada konsep pengiriman bahan mentah (material) serta produksi jadi dilakukan hanya pada saat dibutuhkan saja. Berdasarkan filosofi tersebut, fokus *Just In Time* diarahkan pada upaya minimasi persediaan bahan mentah, barang dalam proses, serta barang jadi dengan tujuan untuk menghemat biaya inventory seta mengurangi berbagai pemborosan (inefisiensi) yang terjadi dalam perusahaan. Dengan demikian, penerapan *Just In Time* pada suatu perusahaan didasarkan pada dua prinsip, yaitu : pengurangan pemborosan serta pemanfaatan secara optimal sumberdaya manusia, perlengkapan, material, serta komponen-komponenya dalam suatu perusahaan. (Vuppalapti, 1995)

Penerapan *Just In Time* difokuskan pada usaha pelaksanaan proses produksi tanpa ada pemborosan (Brown, 1991) melalui penyederhanaan proses produksi. penerapan konsep ini dapat kita lihat pada usaha peniadaan atau minimalisasi jumlah persediaan sehingga biaya persediaan yang ditanggung perusahaan akan optimal. Fokus pada minimalisasi persediaan bahan baku, dan barang dalam proses akan berakibat pada pengangguran biaya persediaan dan masalah-masalah yang berkaitan dengan efisiensi. (Hadioetomo, 2009)

Dalam penghitungan efisiensi biaya produksi menggunakan pendekatan *Just In Time* dapat menggunakan formula model regresi linier berganda seperti sebagai berikut : (Rahayu, 2003)

Keterangan :

Y = efektifitas dan efisiensi biaya produksi

= pembelian

= produksi

= pengiriman bahan baku

= pengiriman bahan jadi

= lingkungan JIT

= intersep atau konstanta

= koefisien regresi

= faktor pengganggu

Adapun penghitungan untuk prediksi stok barang adalah sebagai berikut : (Rahayu, 2003)

Dimana :

Ft = nilai peramalan

= indeks waktu

= perkiraan tahunan

Untuk mencari perkiraan indeks waktu dan perkiraan tahunan dapat menggunakan rumus sebagai berikut :

Dimana n adalah prosentase terhadap total data

atau

Keterangan :

= variabel yang mempengaruhi

y = kecenderungan nilai peramalan

a = konstanta

b = tingkat kecenderungan

K1 = kelompok 1

K2 = kelompok 2

tK1 = tahun dasar kelompok 1

tK2 = tahun dasar kelompok 2

## Model *Waterfall*

*Waterfall* merupakan model yang dikembangkan untuk pengembangan dan perancangan perangkat lunak. Model berkembang secara sistematis dari satu tahap ke tahap lain dalam mode seperti air terjun, *output* dari setiap tahapmerupakan *input* bagi tahap berikutnya. Berikut merupakan tahap-tahap perancangan model *waterfall* :

Analisa Kebutuhan

Implementasi pembuatan sistem

Penerapan Program dan Pemeliharaan

Desain Sistem

Pengujian sistem

Gambar 2.1Model Waterfall (Darmawan, 2015)

1. Analisis Kebutuhan

Tahap ini menganalisa kebutuhan sistem yang akan dibuat dalam bentuk yang dapat dimengerti oleh klien dan staf pengembang. Dalam tahap ini klien atau pengguna menjelaskan segala kendala dan tujuan serta mendefinisikan apa yang diinginkan dari sistem.

1. Desain Sistem

Tahap ini pengembang akan menghasilkan sebuah arsitektur sistem secara keseluruhan dan menentukan alur perangkat lunak hingga pada tahap algoritma yang detail

1. Implementasi Pembuatan Sistem

Tahap ini mengimplementasikan desain yang telah dibuat ke dalam kode program*.*

1. Pengujian Sistem

Pengujian sistem dilakukan untuk menemukan kesalahan-kesalahan yang mungkin terjadi, serta melakukan perbaikan untuk menyempurnakan sistem yang dibuat.

1. Pemeliharaan

Pemeliharaan suatu software sangat diperlukan, termasuk di dalamnya adalah pengembangan, karena *software* yang dibuat tidak selamanya hanya seperti itu, ketika dijalankan mungkin saja masih ada error kecil yang tidak ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada software tersebut. Pengembangan diperlukan ketika adanya perubahan dari eksternal perusahaan seperti ketika ada pergantian sistem operasi, atau perangkat lainnya.

### 2.5.1 Kelebihan Model Waterfall

Kelebihan model ***waterfall*** adalah pencerminan kepraktisan rekayasa, yang membuat kualitas *software* tetap terjaga karena pengembangannya yang terstruktur dan terawasi. Model ini merupakan jenis model yang bersifat dokumen lengkap, sehingga proses pemeliharaan dapat dilakukan dengan mudah. Dokumentasi kode program yang lengkap secara tak langsung menghapus ketergantungan pengembang terhadap pemprogram yang keluar dari tim pengembang, hal ini menguntungkan bagi pihak pengembang karena proses pengembangan perangkat lunak tetap dapat dilanjutkan tanpa bergantung pada pemprogram tertentu.

### 2.5.2 Kekurangan Model Waterfall

Kekurangan **model *waterfall*** yang utama adalah lambatnya proses pengembangan perangkat lunak, karena prosesnya yang tidak bisa diloncat-loncat menjadikan model klasik ini sangat memakan waktu dalam pengembangannya. Di sisi lain, pihak klien tidak dapat mencoba sistem sebelum sistem benar-benar selesai pembuatannya. Kekurangan yang lain adalah kinerja personil yang tidak optimal dan efisien karena terdapat proses menunggu suatu tahapan selesai terlebih dahulu.

# BAB 3. METODOLOGI PENELITIAN

Bagian ini menguraikan tentang metode-metode yang digunakan untuk menyelesaikan tujuan penelitian, dengan adanya metode, alur serta langkah yang di lakukan akan semakin jelas dan terarah. Metode yang digunakan dalam penelitian ini adalah metode pengembangan sistem yaitu model *waterfall* yang merujuk pada gambar 2.1 yang memiliki beberapa tahapan yang akan dijadikan sub bab dalam bab ini.

## Analisis Kebutuhan

Analisis kebutuhan yang diperlukan untuk menyelesaikan tujuan penelitian ini didapat dari tahapan pengumpulan data yang akan dijelaskan pada sub bab berikutnya, setelah pengumpulan data tersebut akan didapat gambaran sistem yang dapat digunakan sebagai acuan dalam pembuatan aplikasi.

### Pengumpulan Data

Pengumpulan data dilakukan untuk memenuhi kebutuhan data yang diperlukan untuk menyelesaikan tujuan penelitian, pengumpulan data tersebut dapat diperoleh dengan beberapa cara sebagai berikut

1. Studi literatur

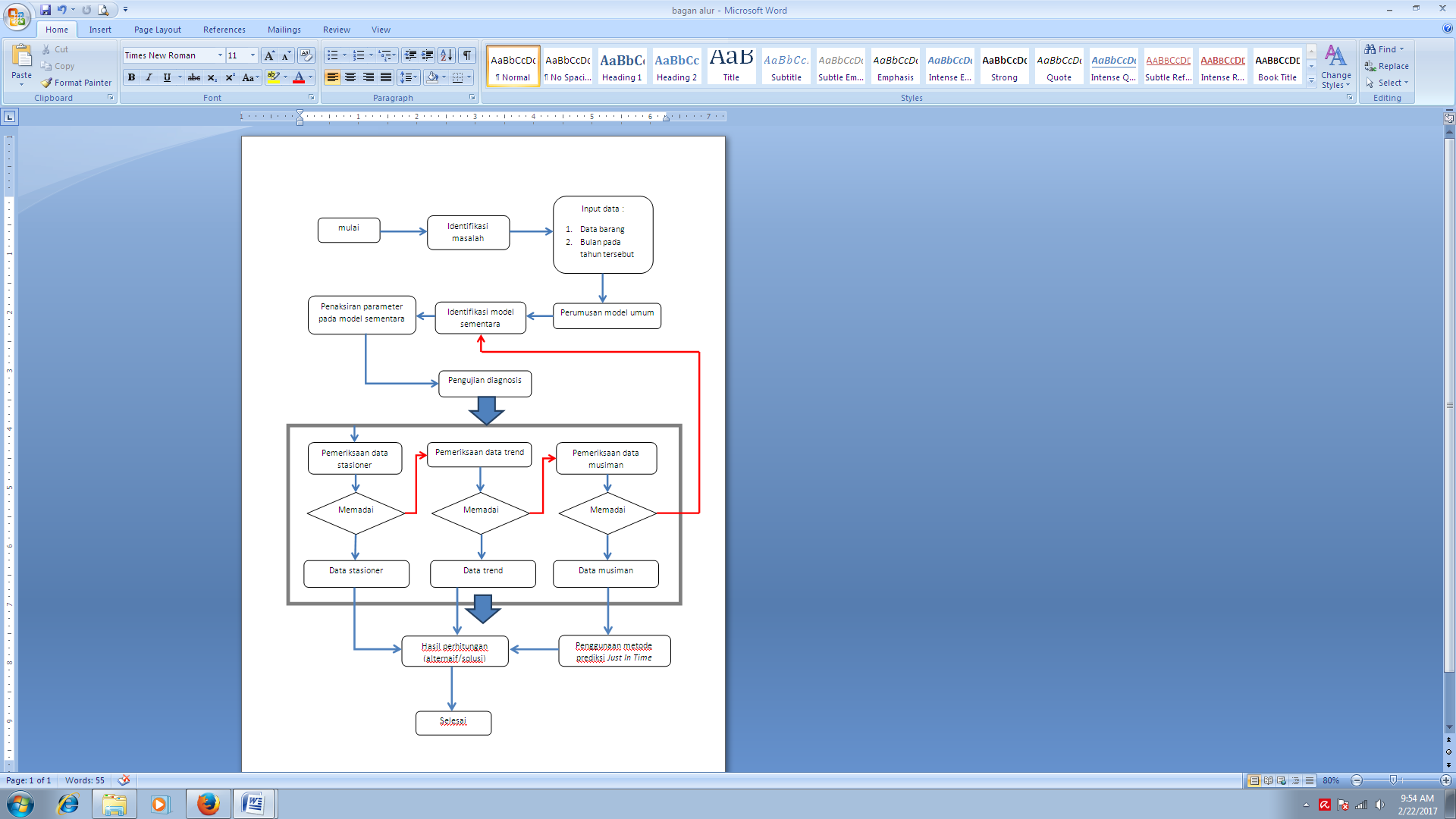
Studiliteratur digunakan untuk mendukung teori-teori yang berkaitan dengan masalah yang akan dibahas. Studi literatur dalam penelitian ini diperoleh melalui data yang berasal dari buku, karya ilmiah, jurnal online, website dan penelitian yang dilakukan sebelumnya, dengan studi literatur tersebut peneliti mendapat referensi untuk menyelesaikan tujuan penelitian dalam pengumpulan data yang dibutuhkan untuk membangun aplikasi.

1. Wawancara

Wawancara yang dilakukan yaitu melalui komunikasi dua arah antara peneliti dan narasumber dan mendapatkan data-data yang diperlukan dalam penelitian. Data-data yang dibutuhkan dalam wawancara berupa data persediaan barang.

Dalam penelitian ini penulis melakukan wawancara langsung dengan Ibu Agustini selaku Admin distributor PT. Irawan Djaja Agung Jember. Pertanyaan yang diajukan meliputi, produk apa saja yang masuk ke PT tersebut, bagaimana prosedur manual pengelolaan persediaan barang, dan bagaimana pencatatan persediaan barang.

Data yang diperoleh dari cara tersebut digunakan sebagai acuan dalam memilih metode yang tepat untuk memprediksi jumlah barang yang akan diambil oleh distributor. Dalam penelitian ini metode *Just In Time* merupakan metode yang akan diterapkan dalam aplikasi untuk melakukan peramalan jumlah stok barang. Berikut ini merupakan proses atau alur dari proses peramalan menggunakan metode *Just In Time*



Gambar 3.1 Diagram Alir Sistem Informasi Prediksi Pengambilan Stok Barang

Gambar 3.1 menunjukkan bahwa adanya proses untuk memprediksi pengambilan stok barang selanjutnya, yang pertama adalah mulai dengan pengidenfikasian masalah dengan memasukkan data barang dan bulan yang diinginkan. Setelah itu data-data yang telah dimasukkan akan diidentifikasi dengan model sementara. Data-data tersebut akan dijadikan parameter pada model sementara dan kemudian akan diuji, apakah parameter tersebut masuk kedalam data stasioner, data trend, atau data musiman. Jika parameter tersebut termasuk data musiman maka parameter tersebut akan menjadi data musiman kemudian akan diuji menggunakan metode *Just In Time* dan akan keluar sebagai hasil dari perhitungan data tersebut. Namun apabila parameter tersebut merupakan data stasioner dan data trend maka akan langsung keluar hasil perhitungan tanpa melalui metode perhitung *Just In Time* tersebut.

### Gambaran Umum Aplikasi

Sistem prediksi pengambilan stok barang adalah sebuah aplikasi yang dapat memprediksi penentuan stok minimal yang akan dialokasikan untuk stok periode selanjutnya oleh admin distributor kepada produsen. Pada tahap awal sistem melakukan pengindentifikasian masalah dengan inputan jumlah stok barang dan jumlah barang retur. Setelah itu sistem akan mengidentifikasi model dan penaksiran parameter pada model sementara. Selanjutnya model sementara akan masuk pada pengujian diagnosis. Pada pengujian diagnosis, data yang sudah masuk akan diperiksa apakah data tersebut masuk ke dalam data stasioner, data trend, atau data musiman. Karena data untuk stok barang ini adalah data stasioner maka data langsung diolah menggunakan metode *Just In Time* dan setelah itu akan muncul hasil dari perhitungan data menggunakan *Just In Time*.

## Desain Aplikasi

Desain dibangun menggunakan *United Modeling language* (UML), penggunaan UML ini sudah menerapkan konsep *Object Oriented Design* yang tentunya sangat memudahkan developer untuk membangun sebuah sistem. Berikut beberapa diagram yang dibuat dalam UML :

### *Business Process*

Digunakan untuk menggambarkan masukan data, keluaran dari aplikasi dan tujuan dari pembuatan aplikasi, dirancang sesuai dengan analisis kebutuhan aplikasi.

### *Usecase Diagram*

Digunakan untuk mendeskripsikan hak akses antara aktor dengan aplikasi.Dirancang sesuai dengan hasil gambaran *bussiness process* yang telah dibuat.

### *Scenario Diagram*

Digunakan untuk menjabarkan alur kerja tiap step-step *usecase* yang sudah di tentukan.

### *Activity Diagram*

Digunakan untuk menggambarkan *scenario* atau aktifitas dari aplikasi untuk dapat mengetahui alur yang dilakukan aktor serta respon aplikasi sesuai yang ditentukan.

### *Sequence Diagram*

Digunakan untuk menggambarkan interaksi-interaksi antar objek di dalam aplikasi. Dirancang sesuai dengan *activity diagram* aplikasi optimalisasi yang telah dibuat, agar dapat mengetahui *method* yang berjalan ketika terjadi aksi.

### *Class Diagram*

Digunakan untuk menggambarkan struktur dari segi pendefinisian kelas-kelas yang ada pada aplikasi sesuai dengan *sequence* yang telah dibuat

### *Entity Relationship Diagram*

Digunakan untuk menggambarkan *database* aplikasi yang dibangun, agar dapat mengetahui tabel *database* yang diperlukan aplikasi sesuai yang telah ditentukan.

## Implementasi Pembuatan Aplikasi

Mengimplementasikan desain yang telah dibuat ke dalam kode program. Penulisan kode program sistem ini menggunakan bahasa pemprogaman *Hypertext Prepocessor* (PHP) dengan bantuan *Framework Code Igniter* (CI). Manajemen basisdata yang digunakan adalah *DBMSXAMPP.*

## Pengujian Aplikasi

Tahap ini merupakan tahap ujicoba terhadap aplikasi yang telah dibangun untuk menemukan kesalahan yang mungkin terjadi agar dapat melakukan perbaikan dan menyempurnakan aplikasi. Pengujian aplikasi menggunakan metode *whitebox* dan *blackbox*, dimana metode *whitebox* dilakukan untuk menguji aplikasi dari segi desain dan kode program. Pengujian ini dilakukan dengan memeriksa logika serta kesalahan yang mungkin terjadi. Metode *blackbox* dilakukan untuk menguji aplikasi dari segi spesifikasi fungsional yaitu untuk mengetahui apakah fungsi masukan, keluaran dari aplikasi sesuai dengan kebutuhan.

# BAB 4. DESAIN DAN IMPLEMENTASI

Bab ini akan menguraikan tentang tentang proses pendesainan dan implementasi desain ke dalam bahasa pemprograman untuk membangun aplikasi prediksi pengambilan stok barang pada PT. Irawan Djaja Agung menggunakan metode *Just In Time*. Pengerjaan desain aplikasi ini menggunakan metode *waterfall* dengan menentukan analisis kebutuhan fungsional dan non fungsional, desain aplikasi (*bussiness process, usecase diagram, squence diagram, activity diagram, classdiagram*, dan *entity relationship diagram*), pengkodean dan *testing*.

## 4.1 Analisis Kebutuhan

Analisis kebutuhan dalam penelitian ini dilakukan dengan cara mengidentifikasi masalah yang ada, dengan mengidentifikasi permasalahan maka analisis kebutuhan dapat ditentukan dalam bentuk kebutuhan fungsional dan non fungsional

Kebutuhan fungsional aplikasi pada penelitian ini adalah sebagai berikut :

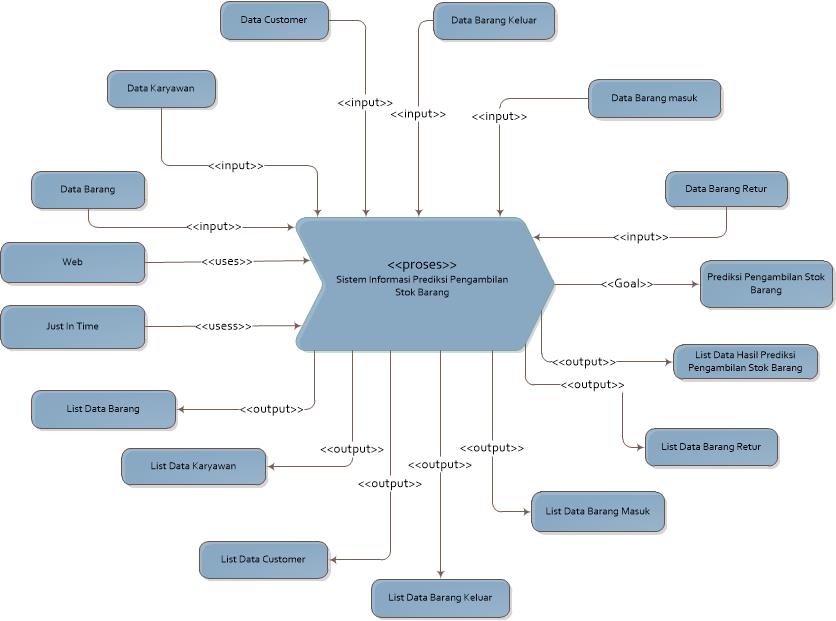
1. Aplikasi dapat menyimpan dan melakukan tambah, dan edit data barang
2. Aplikasi dapat melakukan tambah, edit dan menyimpan data customer
3. Aplikasi dapat melakukan tambah, edit dan menyimpan data karyawan
4. Aplikasi dapat melakukan tambah dan menyimpan data barang masuk
5. Aplikasi dapat melakukan tambah dan menyimpan data barang keluar
6. Aplikasi dapat melakukan tambah dan menyimpan data barang retur
7. Aplikasi dapat menampilkan dan mengupdate kondisi stok barang
8. Aplikasi dapat menampilkan dan memprediksi pengambilan stok barang berikutnya

Kebutuhan non-fungsional aplikasi pada penelitian ini adalah sebagai berikut:

1. Sistem dapat diakses selama 24 jam
2. Sistem menggunakan tampilan yang *user friendly*, sehingga pengguna tidak kesulitan untuk mengoperasikannya.

## 4.2 *Bussiness Process*

*Bussiness process* berguna dalam peyelesaian rangkaian aktivitas dengan tujuan dan hasil tertentu, dengan adanya sebuah proses disertai dengan *input* dan *output* yang dihasilkan maka tujuan/*goal* dari sebuah aplikasiakan tercapai. Berikut merupakan *bussiness process* dari aplikasi prediksi pengambilan stok barang menggunakan metode *Just In Time*

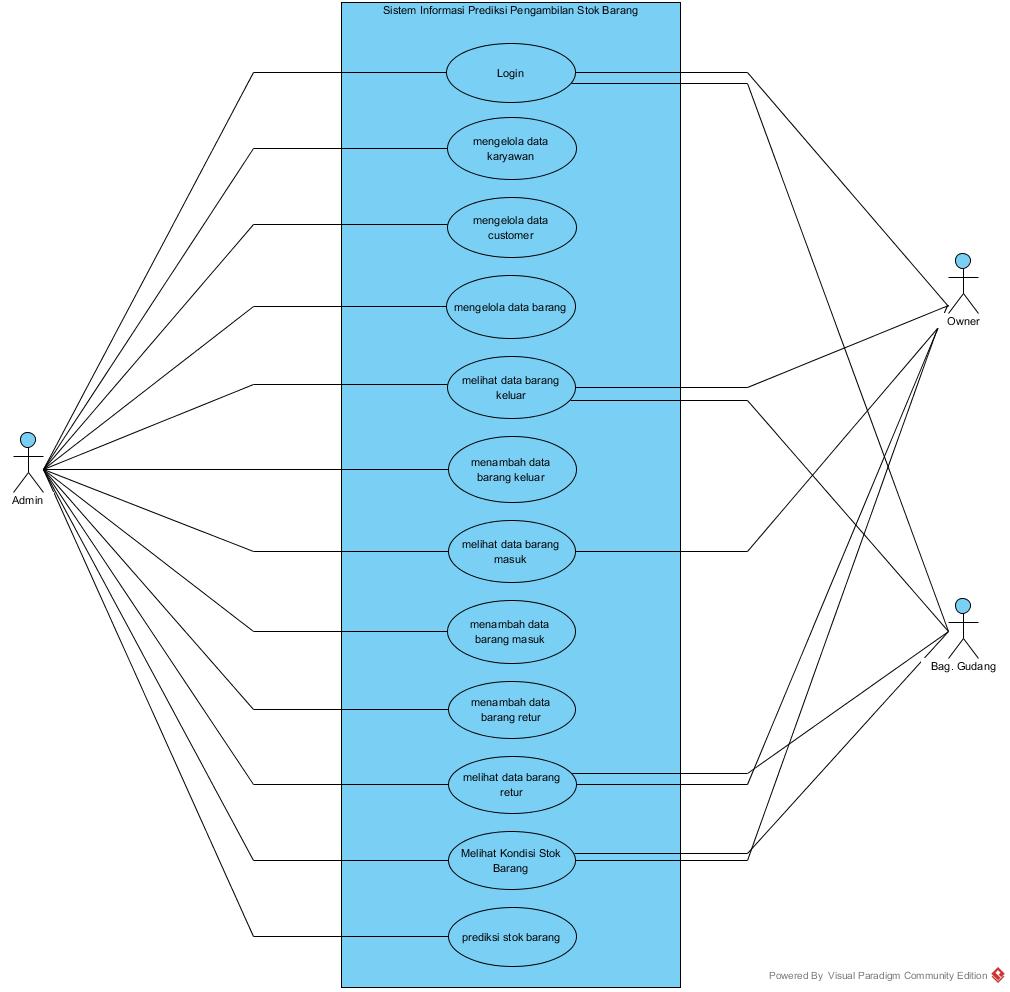


Gambar 4.1 *Bussiness Process* Sistem Informasi Prediksi Pengambilan Stok Barang

Gambar 4.1 menjelaskan adanya proses untuk aplikasi prediksi pengambilan stok barang dengan inputan data karyawan, data customer, data barang, data barang keluar, data barang masuk, data barang retur, serta data untuk melakukan perhitungan prediksi pengambilan stok barang. Adanya inputan tersebut mengharuskan sistem memiliki *output* yang menampilkan list data barang, list data karyawan, list data customer, list data barang keluar, list data barang masuk, list data barang retur, serta hasil perhitungan prediksi pengambilan stok barang. Dengan *input* dan *output* tersebut sistem menghasilkan *goal* yaitu prediksi pengambilan stok barang.

## 4.3 *Usecase* Diagram

*Usecase* berguna untuk menggambarkan dialog antara aktor denganaplikasiserta menentukan fitur apa saja yang dapat di akses oleh masing-masing aktor. Berikut merupakan *usecase* aplikasi prediksi pengambian stok barang PT. Irawan Djaja Agung :



Gambar 4.2 *Usecase Diagram*

Pada *usecase diagram* tersebut terdapat tiga aktor yang dapat mengakses aplikasi prediksi stok barang. Tiga aktor tersebut adalah super admin (admin), pemilik (owner), dan bagian gudang. Deskripsi aktor serta *usecase* akan dijelaskan pada 1 dan 4.2 berikut :

Tabel 4.1 Definisi Aktor *Usecase*

|  |  |  |
| --- | --- | --- |
| **No.** | **Aktor** | **Definisi Tugas** |
| **1.** | Super Admin (Admin) | Admin dapat melakukan :   1. Login 2. Mengelola data barang 3. Mengelola data karyawan 4. Mengelola data customer 5. Melihat dan mengelola data barang keluar 6. Melihat dan mengelola data barang masuk 7. Melihat dan mengelola data barang retur 8. Melihat kondisi stok barang 9. Melakukan prediksi pengambilan stok barang |
| **2.** | Pemilik (Owner) | User dapat melakukan :   1. Login 2. Melihat data barang masuk 3. Melihat data barang keluar 4. Melihat data barang retur 5. Melihat kondisi stok barang |
| **3.** | Bagian Gudang | User dapat melakukan :   1. Login 2. Melihat data barang keluar 3. Melihat data barang retur 4. Melihat kondisi stok barang |

Tabel 4.2 Definisi *Usecase*

|  |  |  |
| --- | --- | --- |
| **No.** | **Usecase** | **Deskripsi** |
| **1.** | Login | Login merupakan langkah awal untuk aktor dapat masuk ke halaman aplikasi |
| **2.** | Mengelola Data Karyawan | Data Karyawan merupakan fitur yang berisi data-data karyawan yang terdapat pada perusahaan tersebut.  Aktor dapat melihat, menambah, serta mengubah data karyawan |
| **3.** | Mengelola Data Customer | Data Customer merupakan fitur yang berisi data-data customer.  Aktor dapat melihat, menambah, serta mengubah data karyawan. |
| **4.** | Mengelola Data Barang | InputBarang merupakan fitur yang berisi data barang yang berada dalam gudang.  Aktor dapat melihat, menambah, serta mengubah data barang |
| **5.** | Melihat Data Barang Keluar | Data Barang Keluar merupakan fitur yang berisi data transaksi barang keluar dari gudang.  Aktor dapat melakukan aksi melihat list tabel data barang keluar |
| **6.** | Menambah Data Barang Keluar | Input Barang Keluar merupakan fitur yang berisi data transaksi barang keluar dari gudang.  Aktor dapat menambah data barang keluar |
| **7.** | Melihat Data Barang Masuk | Data Barang Masuk merupakan fitur yang berisi data transakasi barang masuk ke gudang.  Aktor dapat melakukan aksi melihat list tabel data barang masuk |
| **8.** | Menambah Data Barang Masuk | Input Barang Masuk merupakan fitur yang berisi data transaksi barang masuk ke gudang.  Aktor dapat menambah data barang masuk. |
| **9.** | Melihat Data Barang Retur | Data Barang Retur merupakan fitur yang berisi data transaksi barang retur.  Aktor dapat melakukan aksi melihat list tabel data barang retur. |
| **10.** | Menambah Data Barang Retur | Input Barang Retur merupakan fitur yang berisi data transaksi barang retur.  Aktor dapat menambah data barang retur |
| **11.** | Melihat Kondisi Stok Barang | Kondisi Barang merupakan fitur yang berisi kondisi stok barang yang terdapat pada gudang.  Aktor dapat melakukan aksi melihat list tabel data kondisi barang. |
| **12.** | Prediksi Stok Barang | Merupakan fitur yang digunakan untuk memprediksi pengambilan stok barang terbaik untuk periode selanjutnya.  Aktor dapat melakukan prediksi pengambilan stok barang. |

## *Scenario*

*Scenario* berguna untuk menjabarkan alur kerja tiap step-step *usecase* yang sudah di tentukan. *Scenario usecase* ini terdiri dari nama *usecase*, aktor, pra-kondisi, pasca-kondisi, skenario utama dan skenario alternatif.

### *Scenario* Login

*Scenario login* merupakan interaksi antara aktor dengan aplikasi untuk dapat menggunakan fitur *login*, senghingga aktor dapat masuk ke halaman utama/*dashboard* aplikasi. Setiap aktor disini harus memiliki *username* dan *password* yang sudah terdaftar pada aplikasi.

Pada *scenario login* terdapat beberapa *scenario* utama dan *scenario* alternatif. *Scenario* utama menjelaskan alur dari pertama kali aktor membuka *aplikasi*sampai aktor berhasil masuk ke halaman utama/*dashboard*. Sedangkan *scenario* alternatif merupakan bagian yang menangani alur pengecualian apabila terdapat kesalahan pada saat aktor ingin masuk ke dalam aplikasi. *Scenario login* ke aplikasi oleh aktor admin dapat dilihat pada tabel 4.5 sedangkan *scenario login* ke aplikasi oleh aktor owner dapat dilihat pada tabel 4.6 dan *scenario login* ke aplikasi oleh aktor bagian gudang dapat dilihat pada tabel 4.7

Tabel 4.3 *Scenario Login* Admin

|  |  |  |
| --- | --- | --- |
| ***Scenario Login* Admin** | | |
|  | |  |
| **Nama *Usecase*** | | *Login* |
| **Aktor** | | Admin |
| **Pra-Kondisi** | | Admin membuka halaman aplikasi |
| **Pasca-Kondisi** | | Admin berhasil *login* |
| **Aktor** | **Aplikasi** | |
| **Skenario Utama**  **“*Login*”** | | |
| 1. Membuka aplikasi prediksi pemngambilan stok barang |  | |
|  | 1. Menampilkan halaman *login* untuk masuk ke aplikasi | |
| 1. Memasukkan *username* dan *password* 2. Klik tombol “*sign in*” |  | |
|  | 1. Mengecek data *user* ke *database* 2. Menampilkan halaman utama admin | |
| **Skenario Alternatif**  **“*Username* dan *Password* Salah”** | | |
| 4a. Klik tombol “*sign in*” |  | |
|  | 5a. Mengecek data *user* ke *database*  6a. Menampilkan pesan *error* “*username* dan *password* salah” | |

Tabel 4.4 *Scenario Login* Owner

|  |  |  |
| --- | --- | --- |
| ***Scenario Login* Owner** | | |
|  | |  |
| **Nama *Usecase*** | | *Login* |
| **Aktor** | | Owner |
| **Pra-Kondisi** | | Owner membuka halaman aplikasi |
| **Pasca-Kondisi** | | Owner berhasil *login* |
| **Aktor** | **Aplikasi** | |
| **Skenario Utama**  **“*Login*”** | | |
| 1. Membuka aplikasi prediksi pemngambilan stok barang |  | |
|  | 1. Menampilkan halaman *login* untuk masuk ke aplikasi | |
| 1. Memasukkan *username* dan *password* 2. Klik tombol “*sign in*” |  | |
|  | 1. Mengecek data *user* ke *database* 2. Menampilkan halaman utama admin | |
| **Skenario Alternatif**  **“*Username* dan *Password* Salah”** | | |
| 4a. Klik tombol “*sign in*” |  | |
|  | 5a. Mengecek data *user* ke *database*  6a. Menampilkan pesan *error* “*username* dan *password*salah” | |

Tabel 4.5 *Scenario Login* Bagian Gudang

|  |  |  |
| --- | --- | --- |
| ***Scenario Login* Bagian Gudang** | | |
|  | |  |
| **Nama *Usecase*** | | *Login* |
| **Aktor** | | Bagian Gudang |
| **Pra-Kondisi** | | Bagian Gudang membuka halaman aplikasi |
| **Pasca-Kondisi** | | Bagian Gudang berhasil *login* |
| **Aktor** | **Aplikasi** | |
| **Skenario Utama**  **“*Login*”** | | |
| 1. Membuka aplikasi prediksi pemngambilan stok barang |  | |
|  | 1. Menampilkan halaman *login* untuk masuk ke aplikasi | |
| 1. Memasukkan *username* dan *password* 2. Klik tombol “*sign in*” |  | |
|  | 1. Mengecek data *user* ke *database* 2. Menampilkan halaman utama admin | |
| **Skenario Alternatif**  **“*Username* dan *Password* Salah”** | | |
| 4a. Klik tombol “*sign in*” |  | |
|  | 5a. Mengecek data *user* ke *database*  6a. Menampilkan pesan *error* “*username* dan *password* salah” | |

### *Scenario* Mengelola Data Karyawan

*Scenario* mengelola data karyawan merupakan interaksi antara aktor dengan aplikasi pada fitur karyawan, aktor yang menjalankan fitur ini adalah admin dan owner. Pada *scenario* ini terdapat beberapa sub proses, yaitu :

1. *Scenario* Tambah Data Karyawan

*Scenario* tambah data karyawan menjelaskan alur proses untuk menambah data karyawan, dalam *scenario* ini terdapat *scenario* normal dan *scenario*alternatif. *Scenario* normal menjelaskan alur keberhasilan aktor dalam melakukan tambah data karyawan, sedangkan *scenario* alternatif menjelaskan alur pengecualian yang menangani *exception* apabila ada data kosong, dan penanganan “klik tombol back” untuk proses kembali ke halaman utama data karyawan tanpa menyimpan data yang sudah diinputkan. *Scenario* tambah data karyawan dapat dilihat pada lampiran A1.

1. *Scenario* Ubah Data Karyawan

*Scenario* ubah data karyawan menjelaskan alur proses untuk mengubah data karyawan, dalam *scenario* ini terdapat *scenario* normal dan *scenario* alternatif. *Scenario* normal menjelaskan alur keberhasilan aktor dalam melakukan ubah data karyawan, sedangakan *scenario* alternatif menjelaskan alur pengecualian yang menangani *exception* apabila ada *input* tidak *valid* atau data kosong, dan penanganan “klik tombol back” untuk proses kembali ke halaman utama data karyawan tanpa menyimpan data yang sudah diinputkan. *Scenario* ubah data karyawan dapat dilihat pada lampiran A1.

### *Scenario* Mengelola Data Customer

*Scenario* mengelola data customer merupakan interaksi antara aktor dengan aplikasi pada fitur customer, aktor yang menjalankan fitur ini adalah admin dan owner. Pada *scenario* ini terdapat beberapa sub proses, yaitu :

1. *Scenario* Tambah Data Customer

*Scenario* tambah data customer menjelaskan alur proses untuk menambah data customer, dalam *scenario* ini terdapat *scenario* normal dan *scenario* alternatif. *Scenario* normal menjelaskan alur keberhasilan aktor dalam melakukan tambah data customer, sedangkan *scenario* alternatif menjelaskan alur pengecualian yang menangani *exception* apabila ada data kosong, dan penanganan “klik tombol back” untuk proses kembali ke halaman utama data customer tanpa menyimpan data yang sudah diinputkan. *Scenario* tambah data customerdapat dilihat pada lampiran A1.

1. *Scenario* Ubah Data Customer

*Scenario* ubah data customer menjelaskan alur proses untuk mengubah data customer, dalam *scenario* ini terdapat *scenario* normal dan *scenario* alternatif. *Scenario* normal menjelaskan alur keberhasilan aktor dalam melakukan ubah data customer, sedangkan *scenario* alternatif menjelaskan alur pengecualian yang menangani *exception* apabila ada *input* tidak *valid* atau data kosong, dan penanganan “klik tombol back” untuk proses kembali ke halaman utama data customer tanpa menyimpan data yang sudah diinputkan. *Scenario* ubah data customer dapat dilihat pada lampiran A1.

### *Scenario* Mengelola Data Barang

*Scenario* mengelola data barang merupakan interaksi antara aktor dengan aplikasi pada fitur data barang, aktor yang menjalankan fitur ini adalah admin. Pada *scenario* ini terdapat beberapa sub proses, yaitu :

1. *Scenario* Tambah Data Barang

*Scenario* tambah data barang menjelaskan alur proses untuk menambah data barang, dalam *scenario* ini terdapat *scenario* normal dan *scenario* alternatif. *Scenario* normal menjelaskan alur keberhasilan aktor dalam melakukan tambah data barang, sedangkan *scenario* alternatif menjelaskan alur pengecualian yang menangani *exception* apabila ada data kosong, dan penanganan “klik tombol back” untuk proses kembali ke halaman utama data barang tanpa menyimpan data yang sudah diinputkan. *Scenario* tambah data barang dapat dilihat pada lampiran A1.

1. *Scenario* Ubah Data Barang

*Scenario* ubah data barang menjelaskan alur proses untuk mengubah data barang, dalam *scenario* ini terdapat *scenario* normal dan *scenario* alternatif. *Scenario* normal menjelaskan alur keberhasilan aktor dalam melakukan ubah data barang, sedangakan *scenario* alternatif menjelaskan alur pengecualian yang menangani *exception* apabila ada *input* tidak *valid* atau data kosong, dan penanganan “klik tombol back” untuk proses kembali ke halaman utama data barang tanpa menyimpan data yang sudah diinputkan. *Scenario* ubah data barang dapat dilihat pada lampiran A1.

### *Scenario* Menambah Data Barang Masuk

*Scenario* menambah data barang masuk merupakan interaksi antara aktor dengan aplikasi pada fitur barang masuk, aktor yang menjalankan fitur ini adalah admin. Pada *scenario* tambah data barang masuk menjelaskan alur proses untuk menambah data barang masuk, dalam *scenario* ini terdapat *scenario* normal dan *scenario* alternatif. *Scenario* normal menjelaskan alur keberhasilan aktor dalam melakukan tambah data barang masuk, sedangkan *scenario* alternatif menjelaskan alur pengecualian yang menangani *exception* apabila ada data kosong dan penghapusan data barang yang kurang sesuai pada saat transaksi berlangsung. *Scenario* tambah data barang masuk dapat dilihat pada lampiran A1.

### *Scenario* Melihat Data Barang Masuk

*Scenario* melihat barang masuk merupakan interaksi antara aktor dengan aplikasi pada fitur barang masuk dan sub fitur data barang masuk, aktor yang menjalankan fitur ini adalah owner. Pada *scenario* melihat data barang masuk menjelaskan alur proses untuk melihat seluruh transaksi pada fitur data barang masuk. *Scenario* melihat data barang masuk dapat dilihat pada lampiran A1.

### *Scenario* Menambah Data Barang Keluar

*Scenario* menambah data barang keluar merupakan interaksi antara aktor dengan aplikasi pada fitur barang keluar, aktor yang menjalankan fitur ini adalah admin. Pada *scenario* tambah data barang keluar menjelaskan alur proses untuk menambah data barang keluar, dalam *scenario* ini terdapat *scenario* normal dan *scenario* alternatif. *Scenario* normal menjelaskan alur keberhasilan aktor dalam melakukan tambah data barang keluar, sedangkan *scenario* alternatif menjelaskan alur pengecualian yang menangani *exception* apabila ada data kosong, dan penghapusan data barang apabila ada data barang yang kurang sesuai saat transaksi berlangsung. *Scenario* tambah data barang keluar dapat dilihat pada lampiran A1.

### *Scenario* Melihat Data Barang Keluar

*Scenario* melihat data barang keluar merupakan interaksi antara aktor dengan aplikasi pada fitur barang keluar dan sub fitur data barang keluar, aktor yang menjalankan fitur ini adalah owner. Pada*scenario* melihat data barang keluar menjelaskan alur proses untuk melihat seluruh transaksi pada fitur data barang keluar. *Scenario* melihat data barang keluar dapat dilihat pada lampiran A1.

### *Scenario* Menambah Data Barang Retur

*Scenario* menambah data barang retur merupakan interaksi antara aktor dengan aplikasi pada fitur barang retur, aktor yang menjalankan fitur ini adalah admin. Pada *scenario* tambah data barang retur menjelaskan alur proses untuk menambah data barang retur, dalam *scenario* ini terdapat *scenario* normal dan *scenario* alternatif. *Scenario* normal menjelaskan alur keberhasilan aktor dalam melakukan tambah data barang retur, sedangkan *scenario* alternatif menjelaskan alur pengecualian yang menangani *exception* apabila ada data kosong. *Scenario* tambah data barang retur dapat dilihat pada lampiran A1.

### *Scenario* Melihat Data Barang Retur

*Scenario* melihat data barang keluar merupakan interaksi antara aktor dengan aplikasi pada fitur barang retur, aktor yang menjalankan fitur ini adalah admin, owner, dan bagian gudang. Pada*scenario* melihat data barang retur menjelaskan alur proses untuk melihat seluruh transaksi pada fitur data barang retur. *Scenario* melihat data barang retur dapat dilihat pada lampiran A1.

### *Scenario* Melihat Kondisi Stok Barang

*Scenario* melihat kondisi stok barang merupakan interaksi antara actor dengan aplikasi pada fitur barang retur, actor yang menjalankan fitur ini adalah admin, owner, dan bagian gudang. Pada *scenario* melihat kondisi stok barang menjelaskan alur proses untuk melihat kondisi stok barang terakhir pada fitur kondisi stok. *Scenario* melihat kondisi stok barang dapat dilihat pada lampiran A1.

### *Scenario* Prediksi Pengambilan Stok Barang

*Scenario* prediksi pengambilan stok barang merupakan interaksi antara aktor dengan aplikasi untuk melakukan prediksi pengambilan stok barang pada fitur prediksi. Aktor yang menjalankan fitur ini adalah admin, admin dapat melakukan prediksi pengambilan stok barang ketika menu data barang masuk, data barang keluar, dan data barang retur telah diinputkan, apabila admin melakukan prediksi dengan memilih periode yang datanta belum diinputkan, maka akan terdapat beberapa kolom yang bernilai 0 dan jumlah barang yang akan diambil tidak diketahui hasilnya. Dalam *scenario* prediksi pengambilan stok barang hanya terdapat *scenario* normal yang menjelaskan keberhasilan admin dalam melakukan prediksi pengambilan stok barang. *Scenario* prediksi pengambilan stok barang dapat dilihat pada tabel 4.6

Tabel 4.6 *Scenario* Prediksi Perngambilan Stok Barang

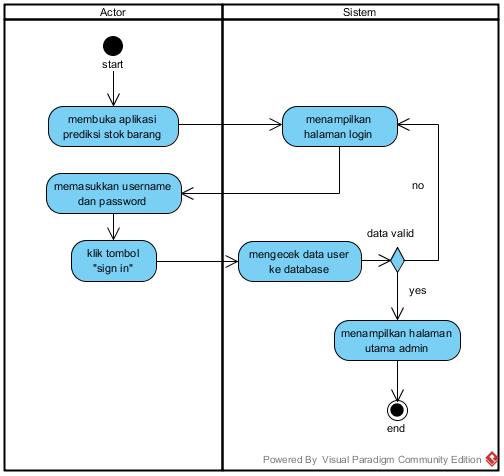
|  |  |  |
| --- | --- | --- |
| ***Scenario* Prediksi Pengambilan Stok Barang** | | |
|  | |  |
| **Nama *Usecase*** | | Prediksi Stok Barang |
| **Aktor** | | Admin |
| **Pra-Kondisi** | | Admin telah berhasil login |
| **Pasca-Kondisi** | | Admin berhasil memprediksi pengambilan stok barang |
| **Aktor** | **Sistem** | |
| **Skenario Utama**  **“Prediksi Pengambilan Stok Barang”** | | |
| 1. Klik menu “Prediksi” |  | |
|  | 1. Menampilkan form prediksi pengambilan stok barang | |
| 1. Memilih barang apa dan bulan ke berapa yang ingin diprediksi 2. Klik tombol “add” |  | |
|  | 1. Menampilkan jumlah biaya barang masuk, barang keluar, dan retur pada barang tersebut | |
| 1. Klik tombol “Start” |  | |
|  | 1. Melakukan perhitungan 2. Menampilkan hasil perhitungan | |

## *ActivityDiagram*

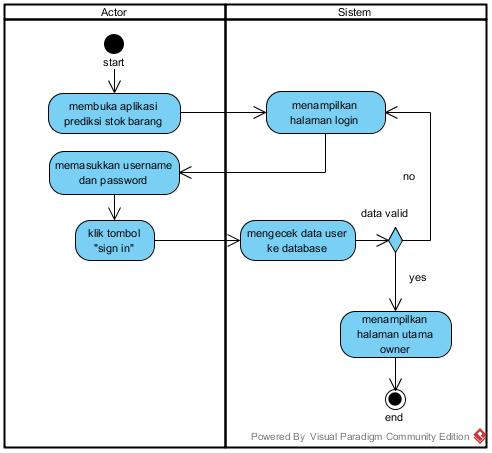
*Activity diagram* merupakan gambaran tentang alur aktifitas dalam aplikasi yang akan dibangun, bagaimana masing-masing alur berasal, *decision* yang mungkin terjadi, serta bagaimana alur berakhir. Pembuatan *activity diagram* ini mengacu pada *usecase* dan *scenario* yang telah dibuat sebelumnya. Berikut adalah *activity diagram* dari aplikasi yang akan dibangun.

### *Activity Diagram Login*

*Activity diagram login* menggambarkan alur aktifitas ketika aktor akan mengakses aplikasi, *login* dilakukan saat aktor memiliki *username* dan *password* yang sesuai dan sudah terdaftar. Jika berhasil *login*, maka aktor akan memasuki tampilan utama menu utama. *Activity diagram* dapat dilihat pada gambar

**

Gambar 4.3 *Activity Diagram Login* Admin



Gambar 4.4 *Activity Diagram Login* Owner



Gambar 4.5 *Activity Diagram Login* Bagian Gudang

### *Activity Diagram* Mengelola Data Karyawan

*Activity diagram* mengelola data karyawan menggambarkan alur aktifitas aktor untuk melakukan tambah, ubah, dan menyimpan data karyawan. Aktor yang dapat mengakses fitur ini adalah admin.

1. *Activity Diagram* Tambah Data Karyawan

*Activity diagram* tambah data karyawan menggambarkan aktifitas antar aktor dan aplikasi untuk melakukan aksi tambah data karyawan. Aktifitas aktor dan aplikasi saling berhubungan dalam suatu aktifitas atau *event*, aplikasi memberikan respon pada aktifitas yang dilakukan aktor saat melakukan tambah data karyawan. *Activity diagram* tambah data karyawan dapat dilihat pada lampiran A2.

1. *Activity Diagram* Ubah Data Karyawan

*Activity diagram* ubah data karyawan menggambarkan aktifitas antara aktor dan aplikasi untuk melakukan aksi ubah data karyawan. Aktifitas aktor dan aplikasi saling berhubungan dalam suatu aktifitas atau *event*, aplikasi memberikan respon pada aktifitas yang dilakukan aktor saat melakukan ubah data karyawan. *Activity diagram* ubah data karyawan dapat dilihat pada lampiran A2.

### *Activity Diagram* Mengelola Data Customer

*Activity diagram* mengelola data customer menggambarkan alur aktifitas aktor untuk melakukan tambah, ubah, dan menyimpan data customer. Aktor yang dapat mengakses fitur ini adalah admin.

1. *Activity Diagram* Tambah Data Customer

*Activity diagram* tambah data customer menggambarkan aktifitas antara aktor dan aplikasi untuk melakukan aksi tambah data customer. Aktifitas aktor dan aplikasi saling berhubungan dalam suatu aktifitas atau *event*, aplikasi memberikan respon pada aktifitas yang dilakukan aktor saat melakukan tambah data customer. *Activity diagram* tambah data customer dapat dilihat pada lampiran A2.

1. *Activity Diagram* Ubah Data Customer

*Activity diagram* ubah data customer menggambarkan aktifitas antara aktor dan aplikasi untuk melakukan aksi ubah data customer. Aktifitas aktor dan aplikasi saling berhubungan dalam suatu aktifitas atau *event*, aplikasi memberikan respon pada aktifitas yang dilakukan aktor saat melakukan ubah data customer. *Activity diagram* ubah data customer dapat dilihat pada lampiran A2.

### *Activity Diaram* Mengelola Data Barang

*Activity diagram* mengelola data barang menggambarkan alur aktifitas aktor untuk melakukan tambah, ubah, dan menyimpan data barang. Aktor yang dapat mengakses fitur ini adalah admin.

1. *Activity Diagram* Tambah Data Barang

*Activity diagram* tambah data barang menggambarkan aktifitas antara aktor dan aplikasi untuk melakukan aksi tambah data barang. Aktifitas aktor dan aplikasi saling berhubungan dalam suatu aktifitas atau *event*, aplikasi memberikan respon pada aktifitas yang dilakukan aktor saat melakukan tambah data barang. *Activity diagram* tambah data barang dapat dilihat pada lampiran A2.

1. *Activity Diagram* Ubah Data Barang

*Activity diagram* ubah data barang menggambarkan aktifitas antara aktor dan aplikasi untuk melakukan aksi ubah data barang. Aktifitas aktor dan aplikasi saling berhubungan dalam suatu aktifitas atau *event*, aplikasi memberikan respon pada aktifitas yang dilakukan aktor saat melakukan ubah data barang. *Activity diagram* ubah data barang dapat dilihat pada lampiran A2.

### *Activity Diagram* Menambah Data Barang Masuk

*Activity diagram* menambah data barang masuk menggambarkan alur aktifitas aktor untuk melakukan tambahdata barang masuk. Aktor yang dapat mengakses fitur ini adalah admin. *Acrivity diagram* tambah data barang masuk menggambarkan aktifitas antara aktor dan aplikasi untuk melakukan aksi tambah data barang masuk. Aktifitas aktor dan aplikasi saling berhubungan dalam suatu aktifitas atau *event*, aplikasi memberikan respon pada aktifitas yang dilakukan aktor saat melakukan tambah data barang masuk. *Activity diagram* tambah data barang masuk dapat dilihat pada lampiran A2.

### *Activity Diagram* Melihat Data Barang Masuk

*Activity diagram*melihat data barang masuk menggambarkan alur aktifitas aktor untuk melihatdata barang masuk. Aktor yang dapat mengakses fitur ini adalah admin dan owner. Aktifitas aktor dan aplikasi saling berhubungan dalam suatu aktifitas atau *event*, aplikasi memberikan respon pada aktifitas yang dilakukan aktor saat melakukan aksi lihat data barang masuk. *Activity diagram* melihat data barang masuk dapat dilihat pada lampiran A2.

### *Activity Diagram* Mengelola Data Barang Keluar

*Activity diagram* mengelola data barang keluar menggambarkan alur aktifitas aktor untuk melakukan tambah data barang keluar. Aktor yang dapat mengakses fitur ini adalah admin. *Activity diagram* tambah data barang keluar menggambarkan aktifitas antara aktor dan aplikasi untuk melakukan aksi tambah data barang keluar. Aktifitas aktor dan aplikasi saling berhubungan dalam suatu aktifitas atau *event*, aplikasi memberikan respon pada aktifitas yang dilakukan aktor saat melakukan tambah data barang keluar. *Activity diagram* tambah data barang keluar dapat dilihat pada lampiran A2.

### *Activity Diagram* Melihat Data Barang Keluar

*Activity diagram* melihat data barang keluar menggambarkan alur aktifitas aktor untuk melihatdata barang keluar. Aktor yang dapat mengakses fitur ini adalah admin, owner, dan bagian gudang. Aktifitas aktor dan aplikasi saling berhubungan dalam suatu aktifitas atau *event*, aplikasi memberikan respon pada aktifitas yang dilakukan aktor saat melakukan aksi lihat data barang keluar. *Activity diagram*melihat data barang keluar dapat dilihat pada lampiran A2.

### *Activity Diagram* Menambah Data Barang Retur

*Activity diagram* menambah data barang retur menggambarkan alur aktifitas aktor untuk melakukan tambah data barang retur. Aktor yang dapat mengakses fitur ini adalah admin. *Activity diagram* tambah data barang retur menggambarkan aktifitas antara aktor dan aplikasi untuk melakukan aksi tambah data barang retur. Aktifitas aktor dan aplikasi saling berhubungan dalam suatu aktifitas atau *event*, aplikasi memberikan respon pada aktifitas yang dilakukan aktor saat melakukan tambah databarang retur. *Activity diagram* tambah data barang retur dapat dilihat pada lampiran A2.

### *Activity Diagram* Melihat Data Barang Retur

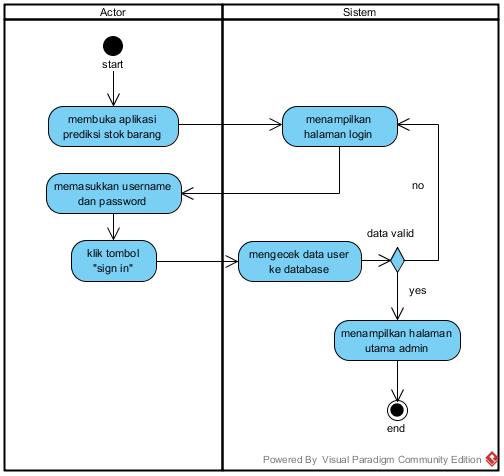
*Activity diagram* melihat data barang retur menggambarkan alur aktifitas aktor untuk melihatdata barang retur. Aktor yang dapat mengakses fitur ini adalah admin, owner, dan bagian gudang. Aktifitas aktor dan aplikasi saling berhubungan dalam suatu aktifitas atau *event*, aplikasi memberikan respon pada aktifitas yang dilakukan aktor saat melakukan aksi lihat data barang retur. *Activity diagram*melihat data barang retur dapat dilihat pada lampiran A2.

### *Activity Diagram* Melihat Kondisi Stok Barang

*Activity diagram* melihat kondisi stok barang menggambarkan alur aktifitas aktor untuk melihatkondisi stok barang. Aktor yang dapat mengakses fitur ini adalah admin, owner, dan bagian gudang. Aktifitas aktor dan aplikasi saling berhubungan dalam suatu aktifitas atau *event*, aplikasi memberikan respon pada aktifitas yang dilakukan aktor saat melakukan aksi lihat kondisi stok barang. *Activity diagram*melihat kondisi stok barang dapat dilihat pada lampiran A2.

### *Activity Diagram* Prediksi Pengambilan Stok Barang

*Activity diagram* prediksi pengambilan stok barang menggambarkan alur aktifitas aktor untuk memprediksi pengambilan stok barang selanjutnya. Aktor yang dapat mengakses fitur ini adalah admin. Aktor mengklik menu “Prediksi” dan aplikasi akan menampilkan halaman utama prediksi pengambilan stok barang. Kemudian aktor memasukkan dat yang diperlukan, setelah ditekan tombol start maka aplikasi akan memulai perhitungan dan selanjutnya aplikasi menampilkan hasil perhitungan beserta penjelasannya. *Activity diagram* prediksi pengambilan stok barang dapat dilihat pada gambar 4.6



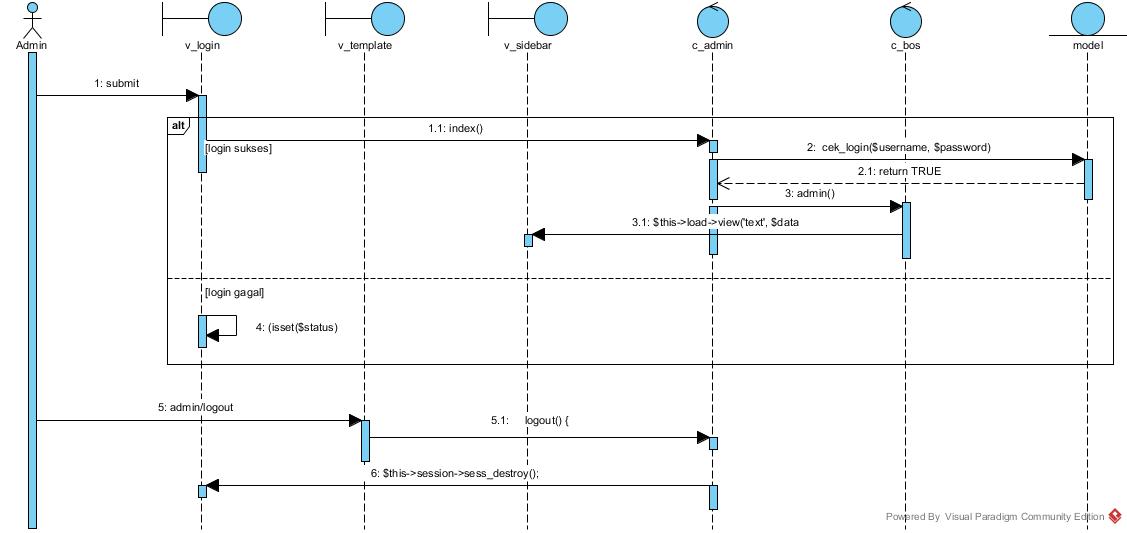
Gambar 4. 6 *Activity Diagram* Prediksi Pengambilan Stok Barang

## *Squence Diagram*

*Sequence diagram* merupakan suatu diagram yang menampilkan interaksi-interaksi antar objek didalam aplikasi yang dibangun. Dengan kata lain, *sequence diagram* memodelkan aliran logika dalam sebuah sistem dengan cara yang visual. Berikut adalah *squence diagram* aplikasi prediksi pengambilan stok barang menggunakan metode *Just In Time*.

### *Squence Diagram Login*

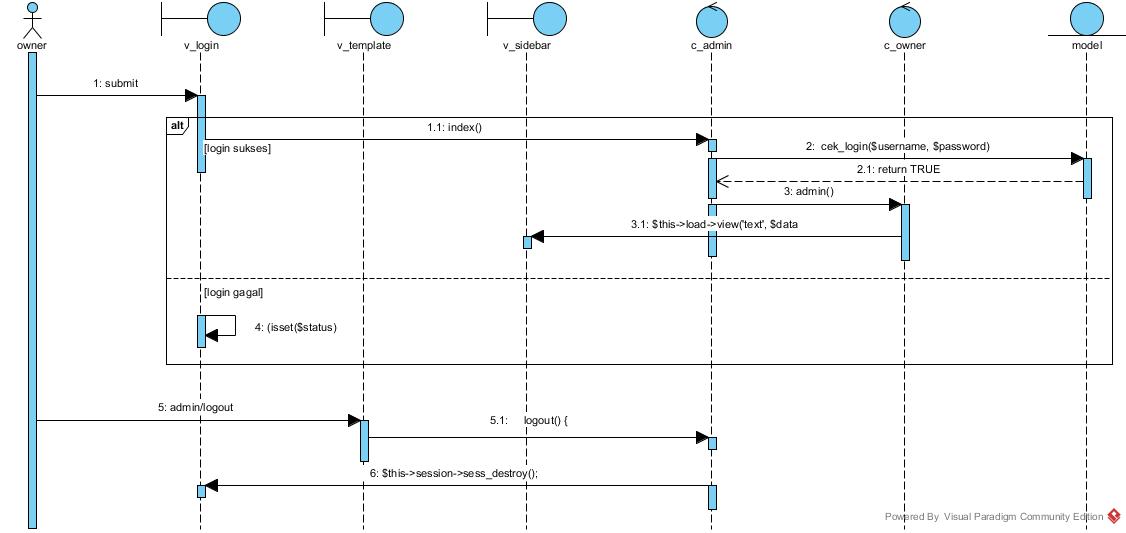
1. *Squence Diagram Login* Admin



Gambar 4.7 *Squence Diagram Login* Admin

Gambar 4.7 merupakan *squence diagram login* admin yang mengambarkan alur logika pada fitur *login* dengan aktor admin. *Squence diagram login* admin terdapat tiga *view* dengan v\_login sebagai *view default* ketika *website* pertama kali diakses. Admin melakukan input *username* dan *password* pada *form login* yang ditampilakn oleh *view* v\_login.*Controller* yang menangani proses *login* adalah c\_admin yang berinteraksi dengan *model* model. Pada proses pengecekan berhasil dilakukan, dan data benar maka sistem otomatis melakukan pengecekan terhadap level *user* melalui *controller* c\_admin yang diteruskan ke c\_bos sesuai dengan level *user* yang digunakan pada saat *login.* Level *user* yang digunakan pada *squencen diagram login*admin adalah *user* admin yang berarti c\_admin akan berinteraksi dengan *controller* c\_bos dan c\_bos bertugas menampilkan halaman awal admin. Pengecekan yang menghasilkan data salah maka akan dikembalikan ke tampilan awal *login* yaitu v\_login oleh c\_admin.

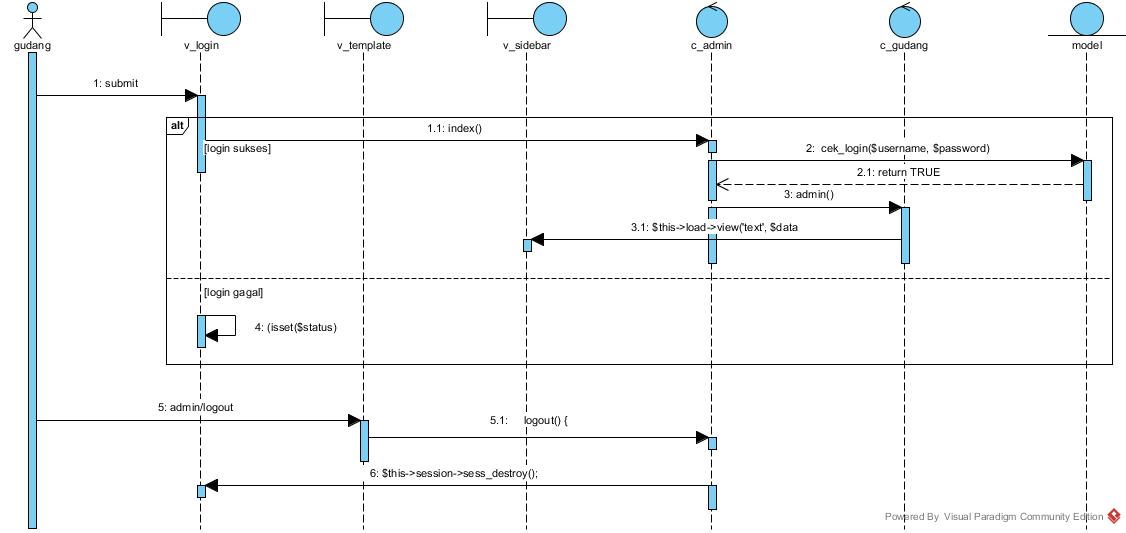
1. *Squence Diagram Login* Owner



Gambar 4.8 *Squence Diagram Login* Owner

Gambar 4.8 merupakan *squence diagram login* owner yang mengambarkan alur logika pada fitur *login* dengan aktor owner. *Squence diagram login* admin terdapat tiga *view* dengan v\_login sebagai *view default* ketika *website* pertama kali diakses. Owner melakukan input *username* dan *password* pada *form login* yang ditampilakn oleh *view* v\_login. *Controller* yang menangani proses *login* adalah c\_admin yang berinteraksi dengan *model* model. Pada proses pengecekan berhasil dilakukan, dan data benar maka sistem otomatis melakukan pengecekan terhadap level *user* melalui *controller* c\_admin yang diteruskan ke c\_owner sesuai dengan level *user* yang digunakan pada saat *login.* Level *user* yang digunakan pada *squencen diagram login* owner adalah *user* owner yang berarti c\_admin akan berinteraksi dengan *controller* c\_owner dan c\_owner bertugas menampilkan halaman awal owner. Pengecekan yang menghasilkan data salah maka akan dikembalikan ke tampilan awal *login* yaitu v\_login oleh c\_admin.

1. *Squence Diagram Login* Bagian Gudang



Gambar 4.9 *Squence Diagram Login* Bagian Gudang

Gambar 4.9 merupakan *squence diagram login* bagian gudang yang mengambarkan alur logika pada fitur *login* dengan aktor bagian gudang. *Squence diagram login* bagian gudang terdapat tiga *view* dengan v\_login sebagai *view default* ketika *website* pertama kali diakses. Admin melakukan input *username* dan *password* pada *form login* yang ditampilakn oleh *view* v\_login. *Controller* yang menangani proses *login* adalah c\_admin yang berinteraksi dengan *model* model. Pada proses pengecekan berhasil dilakukan, dan data benar maka sistem otomatis melakukan pengecekan terhadap level *user* melalui *controller* c\_admin yang diteruskan ke c\_gudang sesuai dengan level *user* yang digunakan pada saat *login.* Level *user* yang digunakan pada *squencen diagram login* bagian gudang adalah *user* bagian gudang yang berarti c\_admin akan berinteraksi dengan *controller* c\_gudang dan c\_gudang bertugas menampilkan halaman awal bagian gudang. Pengecekan yang menghasilkan data salah maka akan dikembalikan ke tampilan awal *login* yaitu v\_login oleh c\_admin.

### *Squence Diagram* Mengelola Data Karyawan

*Squence diagram* mengelola data karyawan menggambar alur logika pada fitur karyawan, actor yang mengakses fitur ini adalah admin. Actor dapat melakukan aksi tambah dan ubah data karyawan

1. *Squence Diagram* Tambah Data Karyawan

*Squence diagram* tambah data karyawan menggambarkan alur logika tambah data karyawan pada fitur karyawan. Pada *squence* ini terdapat v\_sidebar, v\_karyawan, dan v\_addkaryawan yang berfungsi sebagai tampilan utama saat *controller* memanggil *method* untuk menampilkan data dan form data karyawan, *controller* pada *squence* ini adalah c\_bos dan c\_simpan, kemudian terdapat *model* model yang berhubungan langsung dengan *database*. *Method* didalam *model* akan berjalan ketika ada perintah dari *controller*, jadi ketika *controller* akan menampilkan atau menyimpan data, *controller* ini akan menggukan *method*nya untuk berinteraksi dengan *model*. Alur *method* pada *squence* ini ketika aktor menambah data adalah dengan mengklik addkaryawan pada v\_karyawan, *method* yang berjalan dari v\_karyawan menuju c\_bos adalah addkaryawan($id). *Method* tersebut berfungsi untuk memanggil form sehingga nantinya aktor bisa menambah data karyawan. *Squence diagram* tambah data karyawan dapat dilihat pada lampiran A3.

1. *Squence Diagram* Ubah Data Karyawan

*Squence diagram* ubah data karyawan menggambarkan alur logika ubah data karyawan pada fitur karyawan. Pada *squence* ini terdapat v\_sidebar, v\_karyawan, dan v\_addkaryawan yang berfungsi sebagai tampilan utama saat *controller* memanggil *method* untuk menampilkan data dan form data karyawan, *controller* pada *squence* ini adalah c\_bos dan c\_simpan, kemudian terdapat *model* model yang berhubungan langsung dengan *database*. *Method* didalam *model* akan berjalan ketika ada perintah dari *controller*, jadi ketika *controller* akan menampilkan atau menyimpan data, *controller* ini akan menggukan *method*nya untuk berinteraksi dengan *model*. Alur *method* pada *squence* ini ketika aktor mengubah data adalah dengan mengklik addkaryawan pada v\_karyawan, *method* yang berjalan dari v\_karyawan menuju c\_bos adalah addkaryawan($id). *Method* tersebut berfungsi untuk memanggil form sehingga nantinya aktor bisa mengubah data karyawan. *Squence diagram* tambah data karyawan dapat dilihat pada lampiran A3.

### *Squence Diagram* Mengelola Data Customer

*Squence diagram* mengelola data customer menggambar alur logika pada fitur customer, actor yang mengakses fitur ini adalah admin. Aktor dapat melakukan aksi tambah dan ubah data customer

1. *Squence Diagram* Tambah Data Customer

*Squence diagram* tambah data customer menggambarkan alur logika tambah data customer pada fitur customer. Pada *squence* ini terdapat v\_sidebar, v\_customer, dan v\_addcustomer yang berfungsi sebagai tampilan utama saat *controller* memanggil *method* untuk menampilkan data dan form data customer, *controller* pada *squence* ini adalah c\_bos dan c\_simpan, kemudian terdapat *model* model yang berhubungan langsung dengan *database*. *Method* didalam *model* akan berjalan ketika ada perintah dari *controller*, jadi ketika *controller* akan menampilkan atau menyimpan data, *controller* ini akan menggukan *method*nya untuk berinteraksi dengan *model*. Alur *method* pada *squence* ini ketika aktor menambah data adalah dengan mengklik addcustomer pada v\_customer, *method* yang berjalan dari v\_customer menuju c\_bos adalah addcustomer($id). *Method* tersebut berfungsi untuk memanggil form sehingga nantinya aktor bisa menambah data customer. *Squence diagram* tambah data customer dapat dilihat pada lampiran A3.

1. *Squence Diagram* Ubah Data Customer

*Squence diagram* ubah data customermenggambarkan alur logika ubah data customer pada fitur customer. Pada *squence* ini terdapat v\_sidebar, v\_customer, dan v\_addcustomer yang berfungsi sebagai tampilan utama saat *controller* memanggil *method* untuk menampilkan data dan form data customer, *controller* pada *squence* ini adalah c\_bos dan c\_simpan, kemudian terdapat *model* model yang berhubungan langsung dengan *database*. *Method* didalam *model* akan berjalan ketika ada perintah dari *controller*, jadi ketika *controller* akan menampilkan atau menyimpan data, *controller* ini akan menggukan *method*nya untuk berinteraksi dengan *model*. Alur *method* pada *squence* ini ketika aktor mengubah data adalah dengan mengklik addcustomer pada v\_customer, *method* yang berjalan dari v\_customer menuju c\_bos adalah addcustomer($id). *Method* tersebut berfungsi untuk memanggil form sehingga nantinya aktor bisa mengubah data customer. *Squence diagram* tambah data customer dapat dilihat pada lampiran A3.

### *Squence Diagram* Mengelola Data Barang

*Squence diagram* mengelola data barang menggambar alur logika pada fitur data barang, aktor yang mengakses fitur ini adalah admin. Aktor dapat melakukan aksi tambah dan ubah data barang

1. *Squence Diagram* Tambah Data Barang

*Squence diagram* tambah data barang menggambarkan alur logika tambah data barang pada fitur data barang. Pada *squence* ini terdapat v\_sidebar, v\_barang, dan v\_addbarang yang berfungsi sebagai tampilan utama saat *controller* memanggil *method* untuk menampilkan data dan form data barang, *controller* pada *squence* ini adalah c\_bos dan c\_simpan, kemudian terdapat *model* model yang berhubungan langsung dengan *database*. *Method* didalam *model* akan berjalan ketika ada perintah dari *controller*, jadi ketika *controller*akan menampilkan atau menyimpan data, *controller* ini akan menggukan *method*nya untuk berinteraksi dengan *model*. Alur *method* pada *squence* ini ketika aktor menambah data adalah dengan mengklik addbarang pada v\_barang, *method* yang berjalan dari v\_barang menuju c\_bos adalah addbarang($id). *Method* tersebut berfungsi untuk memanggil form sehingga nantinya aktor bisa menambah data barang. *Squence diagram* tambah data barang dapat dilihat pada lampiran A3.

1. *Squence Diagram* Ubah Data Barang

*Squence diagram* ubah data barangmenggambarkan alur logika ubah data barang pada fitur data barang. Pada *squence* ini terdapat v\_sidebar, v\_barang, dan v\_addcustomer yang berfungsi sebagai tampilan utama saat *controller* memanggil *method* untuk menampilkan data dan form data barang, *controller* pada *squence* ini adalah c\_bos dan c\_simpan, kemudian terdapat *model* model yang berhubungan langsung dengan *database*. *Method* didalam *model* akan berjalan ketika ada perintah dari *controller*, jadi ketika *controller* akan menampilkan atau menyimpan data, *controller* ini akan menggukan *method*nya untuk berinteraksi dengan *model*. Alur *method* pada *squence* ini ketika aktor mengubah data adalah dengan mengklik addbarang pada v\_barang, *method* yang berjalan dari v\_barangmenuju c\_bos adalah addbarang($id). *Method* tersebut berfungsi untuk memanggil form sehingga nantinya aktor bisa mengubah data barang. *Squence diagram* tambah data barang dapat dilihat pada lampiran A3.

### *Squence Diagram* Menambah Data Barang Masuk

*Squence diagram* menambah data barang masuk menggambar alur logika pada fitur “Barang Masuk”, aktor yang mengakses fitur ini adalah admin. Aktor dapat melakukan aksi tambah data barang masuk. *Squence diagram* tambah data barang masuk menggambarkan alur logika tambah data barang masuk pada fitur “Input Barang Masuk”. Pada *squence* ini terdapat v\_sidebar, v\_brg\_masuk, dan v\_dbrg\_masuk yang berfungsi sebagai tampilan utama saat *controller* memanggil *method* untuk menampilkan data dan form data barang masuk, *controller* pada *squence* ini adalah c\_bos dan c\_simpan, kemudian terdapat *model* model yang berhubungan langsung dengan *database*. *Method* didalam *model* akan berjalan ketika ada perintah dari *controller*, jadi ketika *controller* akan menampilkan atau menyimpan data, *controller* ini akan menggukan *method*nya untuk berinteraksi dengan *model*. Alur *method* pada *squence* ini ketika aktor menambah data adalah dengan mengklik <?=base\_url().$level ?>/brg\_masuk pada v\_sidebar, jika data benar maka*method* yang berjalan dari v\_sidebar menuju c\_bos adalah brg\_masuk(). *Method* tersebut berfungsi untuk memanggil form sehingga nantinya aktor bisa menambah data barang masuk. *Squence diagram* tambah data barang masuk dapat dilihat pada lampiran A3.

### *Squence Diagram* Melihat Data Barang Masuk

*Squence diagram* melihat data barang masuk menggambarkan alur logika untuk melihat data barang masuk dan detail barang masuk pada fitur “Data Barang Masuk” pleh aktor, aktor yang mengakses fitur ini adalah admin dan owner.

1. *Squence Diagram* Melihat Data Barang Masuk – Admin

*Squence diagaram* melihat data barang masuk oleh aktor admin menggambarakan alur logika untuk melihat data barang masuk dan detail barang masuk pada fitur “Data Barang Masuk” oleh admin. *Controller* c\_bos berinteraksi dengan *model* model untuk mengambil data pada *database* dan menampilkannya pada halaman utama data barang masuk yaitu pada *view* v\_dbrg\_masuk sehingga admin dapaat melihat data-data yang ada pada fitur “Data Barang Masuk” dalam bentuk tabel. Sedangkan untuk melihat detail data barang masuk *controller* c\_bos berinteraksi dengan *model* model untuk mengambil data pada *database* dan menampilkan halaman utama detail data barang masuk yaitu pada *view* v\_rbrg\_masuk sehingga admin dapat melihat data-data yang ada pada fitur “Data Barang Masuk” dalam bentuk tabel. *Squence diagram* melihat data barang masuk oleh admin dapat dilihat pada lampiran A3.

1. *Squence Diagram* Melihat Data Barang Masuk – Owner

*Squence diagaram* melihat data barang masuk oleh aktor owner menggambarakan alur logika untuk melihat data barang masuk dan detail barang masuk pada fitur “Data Barang Masuk” oleh admin. *Controller* c\_owner berinteraksi dengan *model* model untuk mengambil data pada *database* dan menampilkannya pada halaman utama data barang masuk yaitu pada *view* v\_dbrg\_masuk sehingga owner dapat melihat data-data yang ada pada fitur “Data Barang Masuk” dalam bentuk tabel. Sedangkan untuk melihat detail data barang masuk *controller* c\_owner berinteraksi dengan *model* model untuk mengambil data pada *database* dan menampilkan halaman utama detail data barang masuk yaitu pada *view* v\_rbrg\_masuk sehingga owner dapat melihat data-data yang ada pada fitur “Data Barang Masuk” dalam bentuk tabel. *Squence diagram* melihat data barang masuk oleh owner dapat dilihat pada lampiran A3.

### *Squence Diagram* Menambah Data Barang Keluar

*Squence diagram* menambah data barang keluar menggambar alur logika *pada* fitur “Barang Keluar”, aktor yang mengakses fitur ini adalah admin. Aktor dapat melakukan aksi tambah data barang keluar. *Squence diagram* tambah data barang keluar menggambarkan alur logika tambah data barang keluar pada fitur “Input Barang Keluar”. Pada *squence* ini terdapat v\_sidebar, v\_brg\_keluar, dan v\_dbrg\_keluar yang berfungsi sebagai tampilan utama saat *controller* memanggil *method* untuk menampilkan data dan form data barang keluar, *controller* pada *squence* ini adalah c\_bos dan c\_simpan, kemudian terdapat *model* model yang berhubungan langsung dengan *database*. *Method* didalam *model* akan berjalan ketika ada perintah dari *controller*, jadi ketika *controller* akan menampilkan atau menyimpan data, *controller* ini akan menggukan *method*nya untuk berinteraksi dengan *model*. Alur *method* pada *squence* ini ketika aktor menambah data adalah dengan mengklik <?=base\_url().$level ?>/brg\_keluar pada v\_sidebar, jika data benar maka *method* yang berjalan dari v\_sidebar menuju c\_bos adalah brg\_keluar(). *Method* tersebut berfungsi untuk memanggil form sehingga nantinya aktor bisa menambah data barang keluar. *Squence diagram* tambah data barang keluar dapat dilihat pada lampiran A3.

### *Squence Diagram* Melihat Data Barang Keluar

*Squence diagram* melihat data barang keluar menggambarkan alur logika untuk melihat data barang keluar dan detail barang keluar pada fitur “Data Barang Keluar” pleh aktor, aktor yang mengakses fitur ini adalah admin, owner, dan bagian gudang.

1. *Squence Diagram* Melihat Data Barang Keluar – Admin

*Squence diagaram* melihat data barang keluar oleh aktor admin menggambarakan alur logika untuk melihat data barang keluar dan detail barang keluar pada fitur “Data Barang Keluar” oleh admin. *Controller* c\_bos berinteraksi dengan *model* model untuk mengambil data pada *database* dan menampilkannya pada halaman utama data barang keluar yaitu pada *view* v\_dbrg\_keluar sehingga admin dapaat melihat data-data yang ada pada fitur “Data Barang Keluar” dalam bentuk tabel. Sedangkan untuk melihat detail data barang keluar *controller* c\_bos berinteraksi dengan *model* model untuk mengambil data pada *database* dan menampilkan halaman utama detail data barang keluar yaitu pada *view* v\_rbrg\_keluar sehingga admin dapat melihat data-data yang ada pada fitur “Data Barang Keluar” dalam bentuk tabel. *Squence diagram* melihat data barang keluar oleh admin dapat dilihat pada lampiran A3.

1. *Squence Diagram* Melihat Data Barang Keluar – Owner

*Squence diagaram* melihat data barang keluar oleh aktor owner menggambarakan alur logika untuk melihat data barang keluar dan detail barang keluar pada fitur “Data Barang Keluar” oleh owner. *Controller* c\_owner berinteraksi dengan *model* model untuk mengambil data pada *database* dan menampilkannya pada halaman utama data barang keluar yaitu pada *view* v\_dbrg\_keluar sehingga owner dapat melihat data-data yang ada pada fitur “Data Barang Keluar” dalam bentuk tabel. Sedangkan untuk melihat detail data barang keluar *controller* c\_owner berinteraksi dengan *model* model untuk mengambil data pada *database* dan menampilkan halaman utama detail data barang kelur yaitu pada *view* v\_rbrg\_keluar sehingga admin dapat melihat data-data yang ada pada fitur “Data Barang Keluar” dalam bentuk tabel. *Squence diagram* melihat data barang keluar oleh owner dapat dilihat pada lampiran A3.

1. *Squence Diagram* Melihat Data Barang Keluar – Bagian Gudang

*Squence diagaram* melihat data barang keluar oleh aktor bagian gudang menggambarakan alur logika untuk melihat data barang keluar dan detail barang keluar pada fitur “Data Barang Keluar” oleh bagian gudang. *Controller* c\_gudang berinteraksi dengan *model* model untuk mengambil data pada *database* dan menampilkannya pada halaman utama data barang keluar yaitu pada *view* v\_dbrg\_keluar sehingga bagian gudang dapat melihat data-data yang ada pada fitur “Data Barang Keluar” dalam bentuk tabel. Sedangkan untuk melihat detail data barang keluar *controller* c\_gudang berinteraksi dengan *model* model untuk mengambil data pada *database* dan menampilkan halaman utama detail data barang kelur yaitu pada *view* v\_rbrg\_keluar sehingga admin dapat melihat data-data yang ada pada fitur “Data Barang Keluar” dalam bentuk tabel. *Squence diagram* melihat data barang keluar oleh bagian gudang dapat dilihat pada lampiran A3.

### *Squence Diagram* Menambah Data Barang Retur

*Squence diagram* menambah data barang retur menggambar alur logika pada fitur “Barang Retur”, aktor yang mengakses fitur ini adalah admin. Aktor dapat melakukan aksi tambah data barang retur. *Squence diagram* tambah data barang retur menggambarkan alur logika tambah data barang retur pada fitur “Input Barang Retur”. Pada *squence* ini terdapat v\_sidebar, v\_brg\_retur, dan v\_dbrg\_retur yang berfungsi sebagai tampilan utama saat *controller* memanggil *method* untuk menampilkan data dan form data barang retur, *controller* pada *squence* ini adalah c\_bos dan c\_simpan, kemudian terdapat *model* model yang berhubungan langsung dengan *database*. *Method* didalam *model* akan berjalan ketika ada perintah dari *controller*, jadi ketika *controller* akan menampilkan atau menyimpan data, *controller* ini akan menggukan *method*nya untuk berinteraksi dengan *model*. Alur *method* pada *squence* ini ketika aktor menambah data adalah dengan mengklik <?=base\_url().$level ?>/trans\_brgretur pada v\_sidebar, jika data benar maka *method* yang berjalan dari v\_sidebar menuju c\_bos adalah trans\_brgretur(). *Method* tersebut berfungsi untuk memanggil form sehingga nantinya aktor bisa menambah data barang retur. *Squence diagram* tambah data barang retur dapat dilihat pada lampiran A3.

### *Squence Diagram* Melihat Data Barang Retur

*Squence diagram* melihat data barang retur menggambarkan alur logika untuk melihat data barang retur dan detail barang retur pada fitur “Data Barang Retur” oleh aktor, aktor yang mengakses fitur ini adalah admin, owner, dan bagian gudang.

1. *Squence Diagram* Melihat Data Barang Retur – Admin

*Squence diagaram* melihat data barang retur oleh aktor admin menggambarakan alur logika untuk melihat data barang retur dan detail barang retur pada fitur “Data Barang Retur” oleh admin. *Controller* c\_bos berinteraksi dengan *model* model untuk mengambil data pada *database* dan menampilkannya pada halaman utama data barang retur yaitu pada *view* v\_dbrg\_retur sehingga admin dapat melihat data-data yang ada pada fitur “Data Barang Retur” dalam bentuk tabel. Sedangkan untuk melihat detail data barang retur*controller* c\_bos berinteraksi dengan *model* model untuk mengambil data pada *database* dan menampilkan halaman utama detail data barang retur yaitu pada *view* v\_rbrg\_retur sehingga admin dapat melihat data-data yang ada pada fitur “Data Barang Retur” dalam bentuk tabel. *Squence diagram* melihat data barang retur oleh admin dapat dilihat pada lampiran A3.

1. *Squence Diagram* Melihat Data Barang Retur – Owner

*Squence diagaram* melihat data barang retur oleh aktor owner menggambarakan alur logika untuk melihat data barang retur dan detail barang retur pada fitur “Data Barang Retur” oleh owner. *Controller* c\_owner berinteraksi dengan *model* model untuk mengambil data pada *database* dan menampilkannya pada halaman utama data barang retur yaitu pada *view* v\_dbrg\_retur sehingga admin dapaat melihat data-data yang ada pada fitur “Data Barang Retur” dalam bentuk tabel. Sedangkan untuk melihat detail data barang retur *controller* c\_owner berinteraksi dengan *model* model untuk mengambil data pada *database* dan menampilkan halaman utama detail data barang retur yaitu pada *view* v\_rbrg\_retur sehingga admin dapat melihat data-data yang ada pada fitur “Data Barang Retur” dalam bentuk tabel. *Squence diagram* melihat data barang retur oleh owner dapat dilihat pada lampiran A3.

1. *Squence Diagram* Melihat Data Barang Retur – Bagian Gudang

*Squence diagaram* melihat data barang retur oleh aktor bagian gudang menggambarakan alur logika untuk melihat data barang retur dan detail barang retur pada fitur “Data Barang Retur” oleh bagian gudang. *Controller* c\_gudang berinteraksi dengan *model* model untuk mengambil data pada *database* dan menampilkannya pada halaman utama data barang retur yaitu pada *view* v\_dbrg\_retur sehingga bagian gudang dapat melihat data-data yang ada pada fitur “Data Barang Retur” dalam bentuk tabel. Sedangkan untuk melihat detail data barang retur*controller* c\_gudang berinteraksi dengan *model* model untuk mengambil data pada *database* dan menampilkan halaman utama detail data barang retur yaitu pada *view* v\_rbrg\_retur sehingga admin dapat melihat data-data yang ada pada fitur “Data Barang Retur” dalam bentuk tabel. *Squence diagram* melihat data barang retur oleh bagian gudang dapat dilihat pada lampiran A3.

### *Squence Diagram* Melihat Kondisi Stok Barang

*Squence diagram* melihat kondisi stok barang menggambarkan alur logika untuk melihat kondisi stok barang pada fitur “Kondisi Stok Barang” oleh aktor, aktor yang mengakses fitur ini adalah admin, owner, dan bagian gudang.

1. *Squence Diagram* Melihat Kondisi Stok Barang – Admin

*Squence diagram* melihat kondisi stok barang oleh aktor admin menggambarkan alur logika untuk melihat kondisi stok barang pada fitur “Kondisi Barang” oleh admin. *Controller* c\_bos berinteraksi dengan *model* model untuk mengambil data pada *database* dan menampilkannya pada halaman utama data kondisi stok barang yaitu pada *view* v\_kondisi\_stokbrgsehingga admin dapat melihat data-data yang ada pada fitur “Kondisi Barang” dalam bentuk tabeldalam bentuk tabel. *Squence diagram* melihat kondisi stok barang oleh admin dapat dilihat pada lampiran A3.

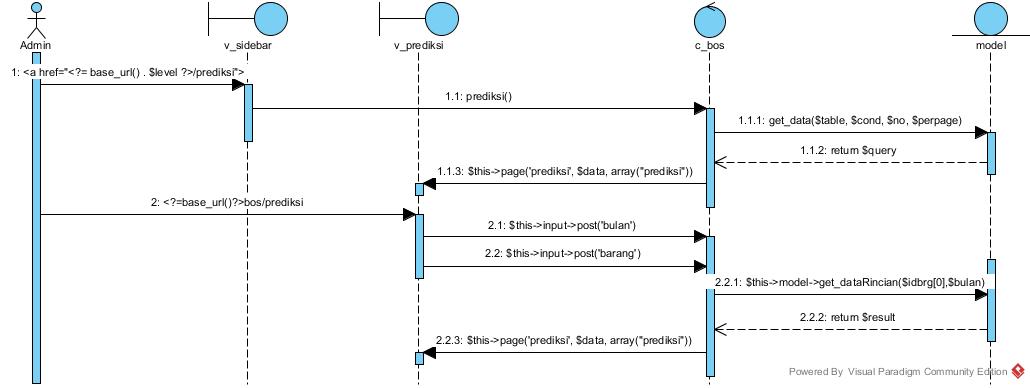
1. *Squence Diagram* Melihat Data Kondisi Stok Barang – Owner

*Squence diagram* melihat kondisi stok barang oleh aktor owner menggambarkan alur logika untuk melihat kondisi stok barang pada fitur “Kondisi Barang” oleh owner. *Controller* c\_owner berinteraksi dengan *model* model untuk mengambil data pada *database* dan menampilkannya pada halaman utama data kondisi stok barang yaitu pada *view* v\_kondisi\_stokbrg sehingga owner dapat melihat data-data yang ada pada fitur “Kondisi Barang” dalam bentuk tabel dalam bentuk tabel. *Squence diagram* melihat kondisi stok barang oleh owner dapat dilihat pada lampiran A3.

1. *Squence Diagram* Melihat Kondisi Stok Barang – Bagian Gudang

*Squence diagram* melihat kondisi stok barang oleh aktor bagian gudang menggambarkan alur logika untuk melihat kondisi stok barang pada fitur “Kondisi Barang” oleh bagian gudang. *Controller* c\_gudang berinteraksi dengan *model* model untuk mengambil data pada *database* dan menampilkannya pada halaman utama data kondisi stok barang yaitu pada *view* v\_kondisi\_stokbrg sehingga bagian gudang dapat melihat data-data yang ada pada fitur “Kondisi Barang” dalam bentuk tabel dalam bentuk tabel. *Squence diagram* melihat kondisi stok barang oleh bagian gudang dapat dilihat pada lampiran A3.

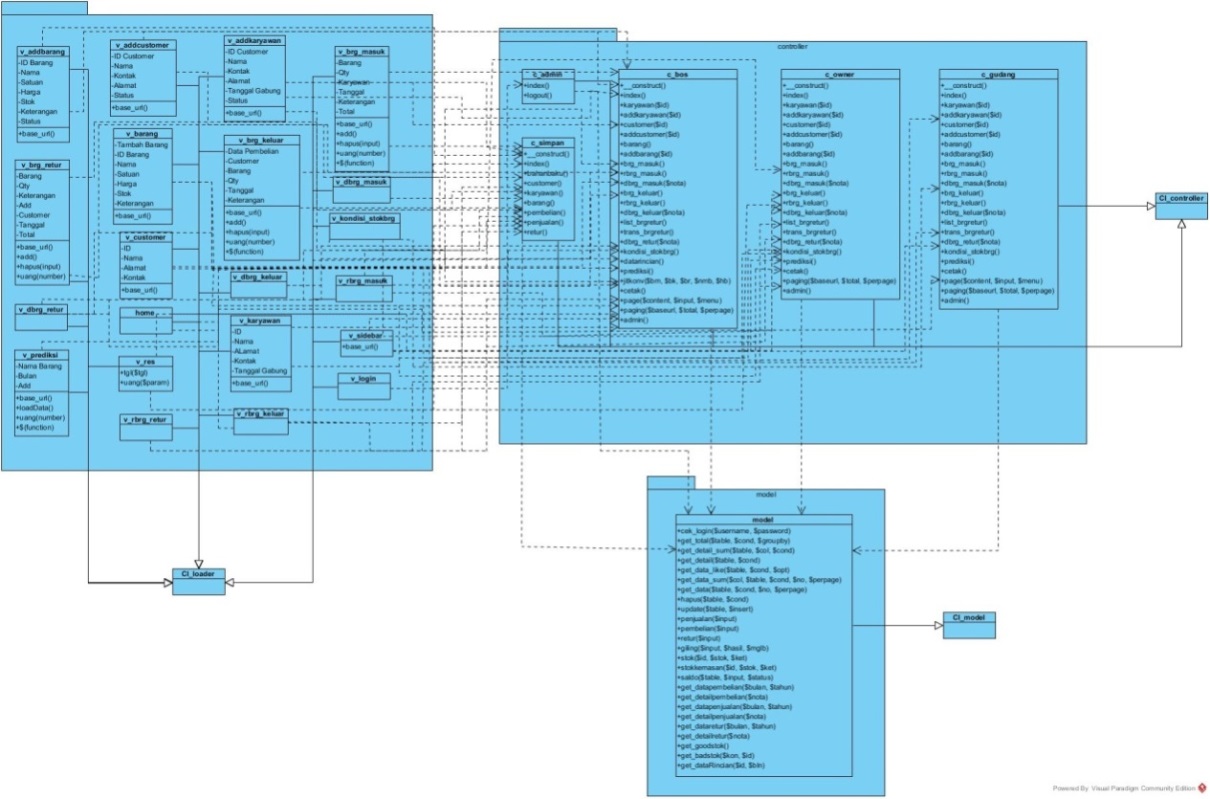
### *Squence Diagram* Prediksi Pengambilan Stok Barang

Gambar 4.10 *Squence Diagram* Prediksi Pengambilan Stok Barang

Gambar 4.10 merupakan *squence diagram* prediksi stok barang yang menggambarkan alur logika untuk memprediksi pengambilan barang selanjutnya pada fitur “Prediksi” oleh admin. Pada *squence* ini *controller* c\_bos berinteraksi dengan *model* model untuk mengambil data pada *database* dan menampilkannya pada halaman utama data perhitungan prediksi yaitu pada *view* v\_prediksi sehingga admin dapat melihat hasil perhitungan data yang ada pada fitur “Prediksi” dalam bentuk penjelasan singkat.

## *Class Diagram*

Setelah melalui tahap pembuatan desain dengan *squence diagram,* tahap selanjutnya adalah membuat desain perancangan *class diagram. Class diagram* terdiri dari *model, view,* dan *controller* yang masing-masing berisi *method,* dan data yang berbeda namun saling memiliki hubungan dengan yang lainnya. Berikut adalah *class diagram* aplikasi prediksi pengambilan stok barang menggunakan metode *Just In Time*.

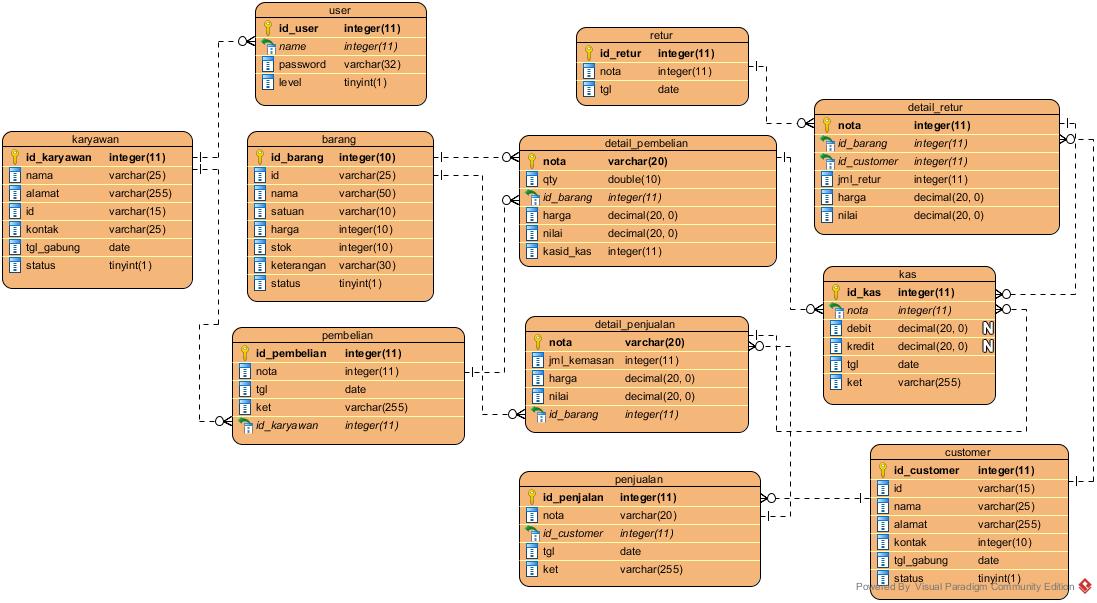


Gambar 4.11 *Class Diagram* Sistem Informasi Prediksi Pengambilan Stok Barang

Gambar 4.11 merupakan *class diagram* aplikasi prediksi pengambilan stok barang, dalam *class* tersebut terdapat *view*, *model*, dan *controller*. *View* dalam *class* tersebut terhubung ke *controller* sesuai dengan aktor yangmengakses. Untuk *view* yang terhubung pada *controller* c\_bos maka tabel yang ditampilkan hanya untuk aktor admin, pada *controller* c\_owner maka tabel yang ditampilkan hanya untuk aktor owner, dan pada *controller* c\_gudang maka tabel yang ditampilkan hanya untuk aktor bagian gudang. *View* dalam *class* tersebut terhubung ke *controller* menggunakan garis *dependency*, begitu juga *controller* yang terhubung ke *model* menggunakan garis *dependency*.

## *EntityRelationship Diagram* (ERD)

*Entity relationship diagram* merupakan model struktur data dan hubungan antar data. Objek dalam ERD digambarkan dengan sebuah entitas yang memiliki atribut-atribut yang berelasi dengan entitas lainnya. Proses pembangunan aplikasi pada penelitian ini menggunakan 11 entitas yang telah dinormalisasi untuk menampung data pada *database* yang direlasikan dengan relasi *one to many*. *Entity relationship diagram* aplikasi prediksi pengambilan stok barang menggunakan metode *Just In Time* pada PT. Irawan Djaja Agung Jember dapat dilihat pada gambar 4.12 Berikut



Gambar 4.12 ERD Prediksi Pengambilan Stok Barang

## *Implementasi Perancangan*

Setelah tahap desain perancangan selesai, tahap selanjutnya dalam penelitian ini adalah tahap pengimplementasian desain perancangan ke dalam bahasa pemprograman. Bahasa pemprograman yang dipakai adalah bahasa pemrograman PHP (*Hypertext Preprocessor*) dengan memanfaatkan *framework CodeIgniter* yang sudah menerapkan konsep *Object Oriented Programming* (OOP), dan *database* yang digunakan adalah MySql (PhpMyadmin).

## *Pengujian Aplikasi*

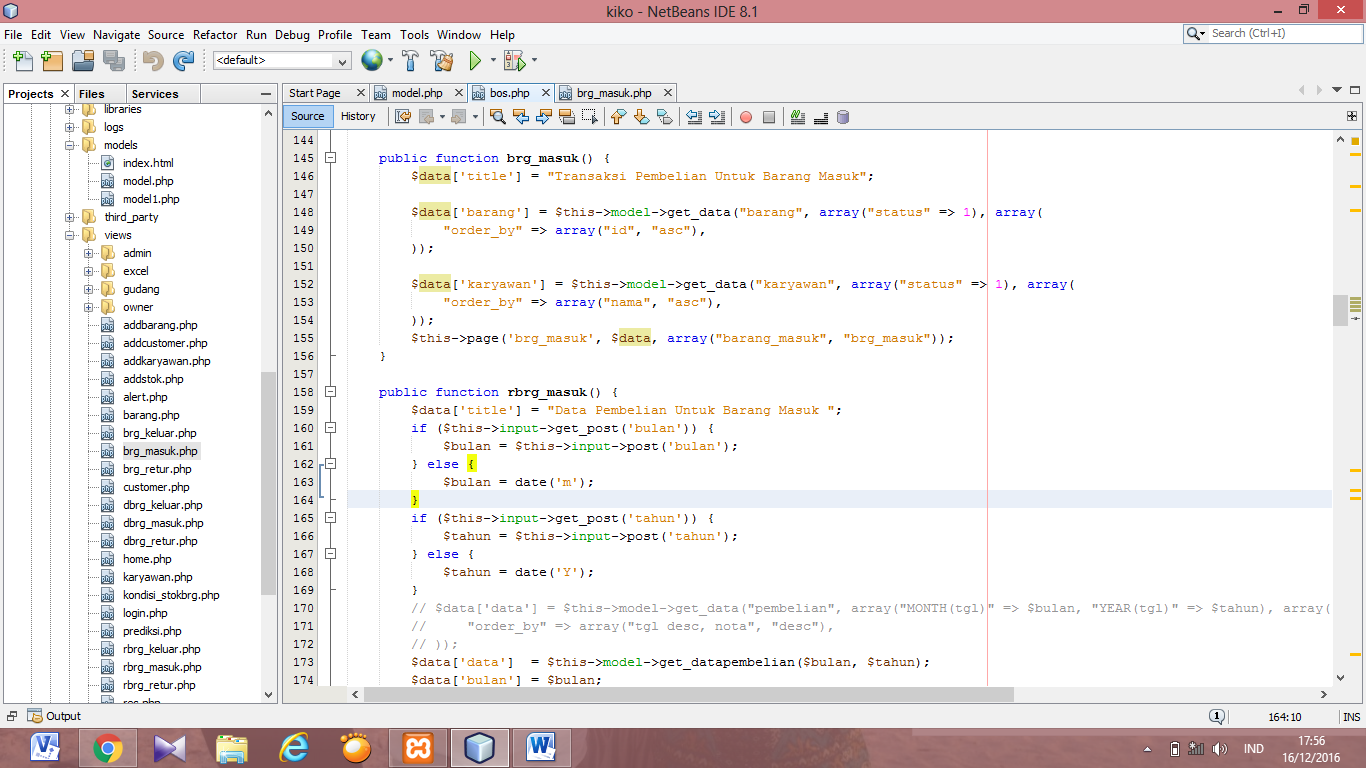
Pengujian aplikasi merupakan tahapan akhir dalam pembangunan aplikasiperhitungan perkiraan pengambilan stok barang. Pada tahapan ini penulis menggunakan dua metode pengujian aplikasi yaitu *White Box Testing* dan *Black Box Testing.*

### White Box Testing

Pengujian *white box* dilakukan untuk mengevaluasi aplikasi yang telah dibuat. Pengujian *white box* yang dilakukan pada penelitian ini meliputi *listing program*, grafik alir, *cyclomatic complexity* (CC), basis set, dan *test case.* Berikut adalah pengujian *white box* pada fitur prediksi pengambilan stok barang yang menggunakan metode *Just In Time*.

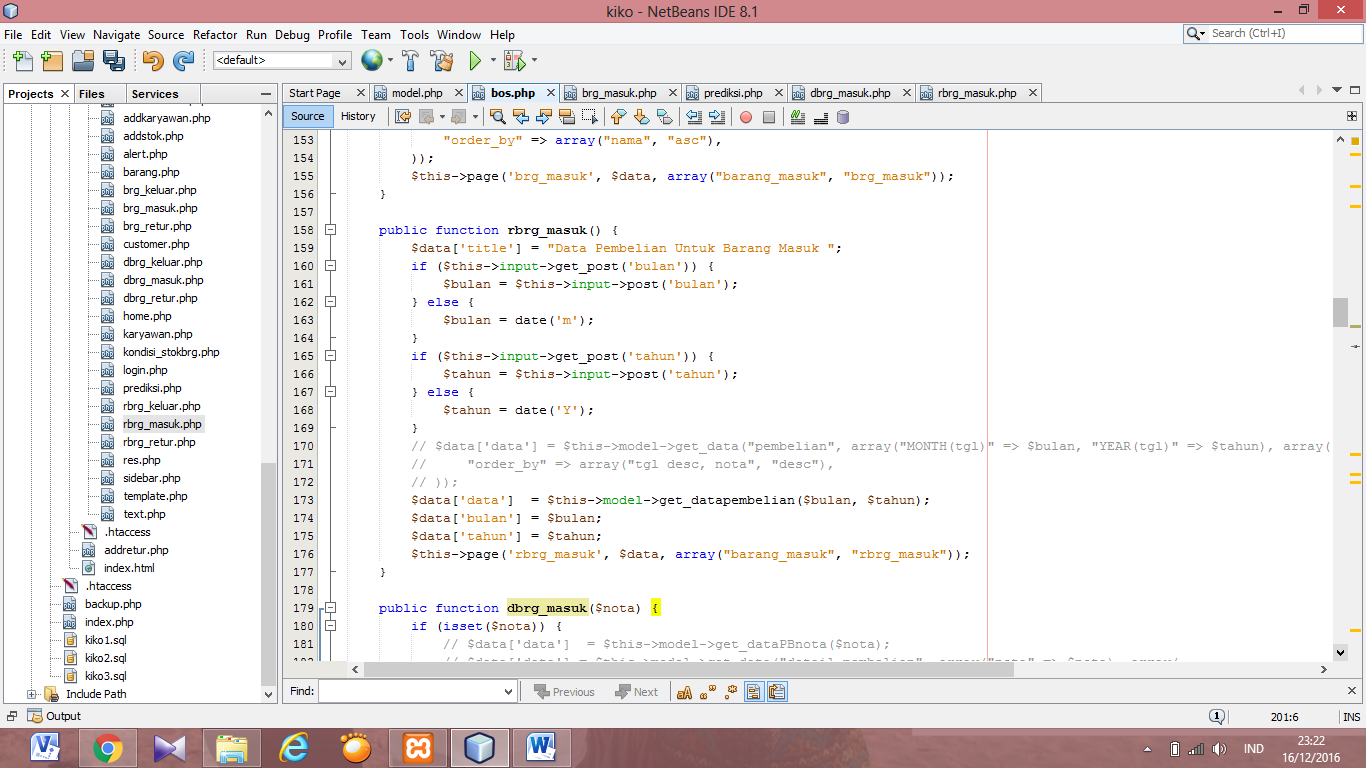
1. *Listing Program*

*Listing program* merupakan susunan dari beberapa kode yang kemudian menjadi sebuah perintah programing yang digunakan dalam menyusun sebuah perintah software. Berikut adalah *listing program method* yang ada di sistem informasi prediksi



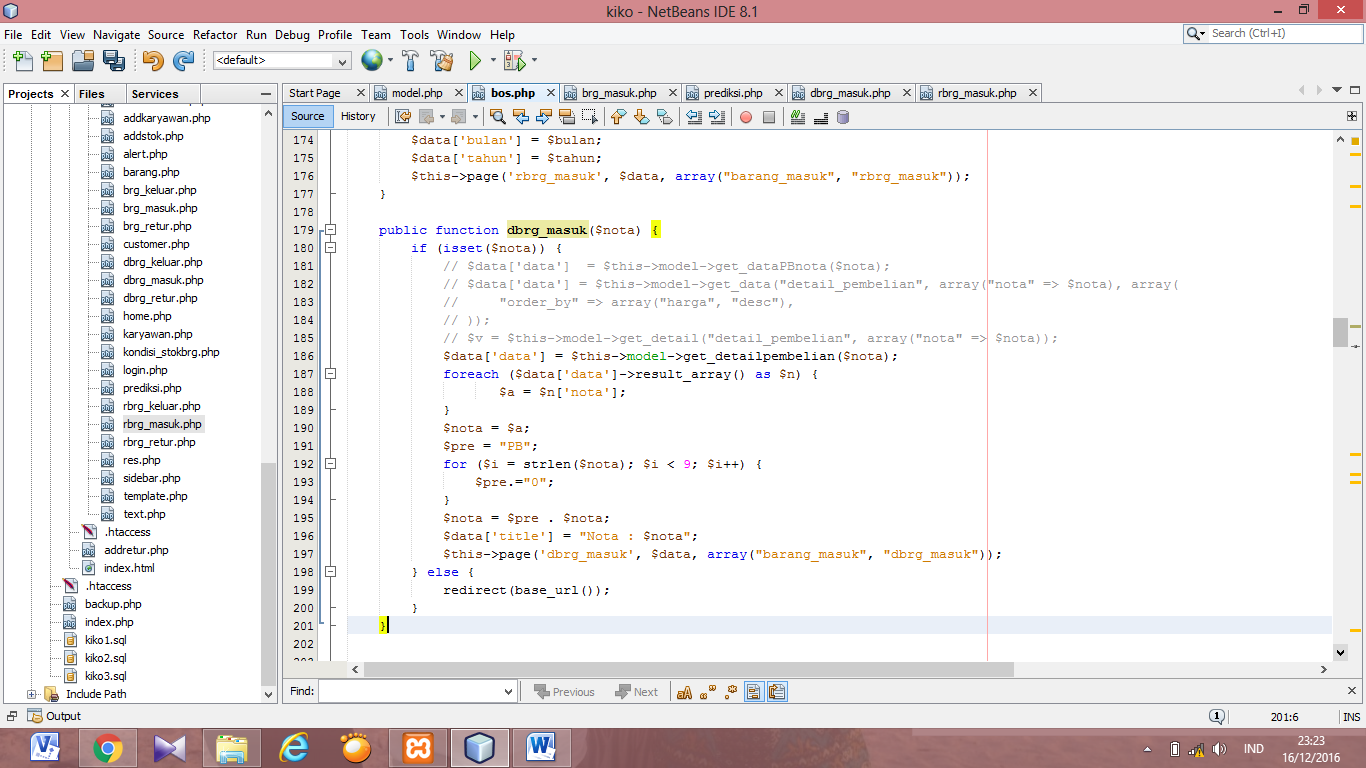
Gambar 4.13 *Method* brg\_masuk()

Gambar 4.13 merupakan kode *method* brg\_masuk() dimana *method* tersebut berfungsi untuk menampilkan halaman utama barang masuk



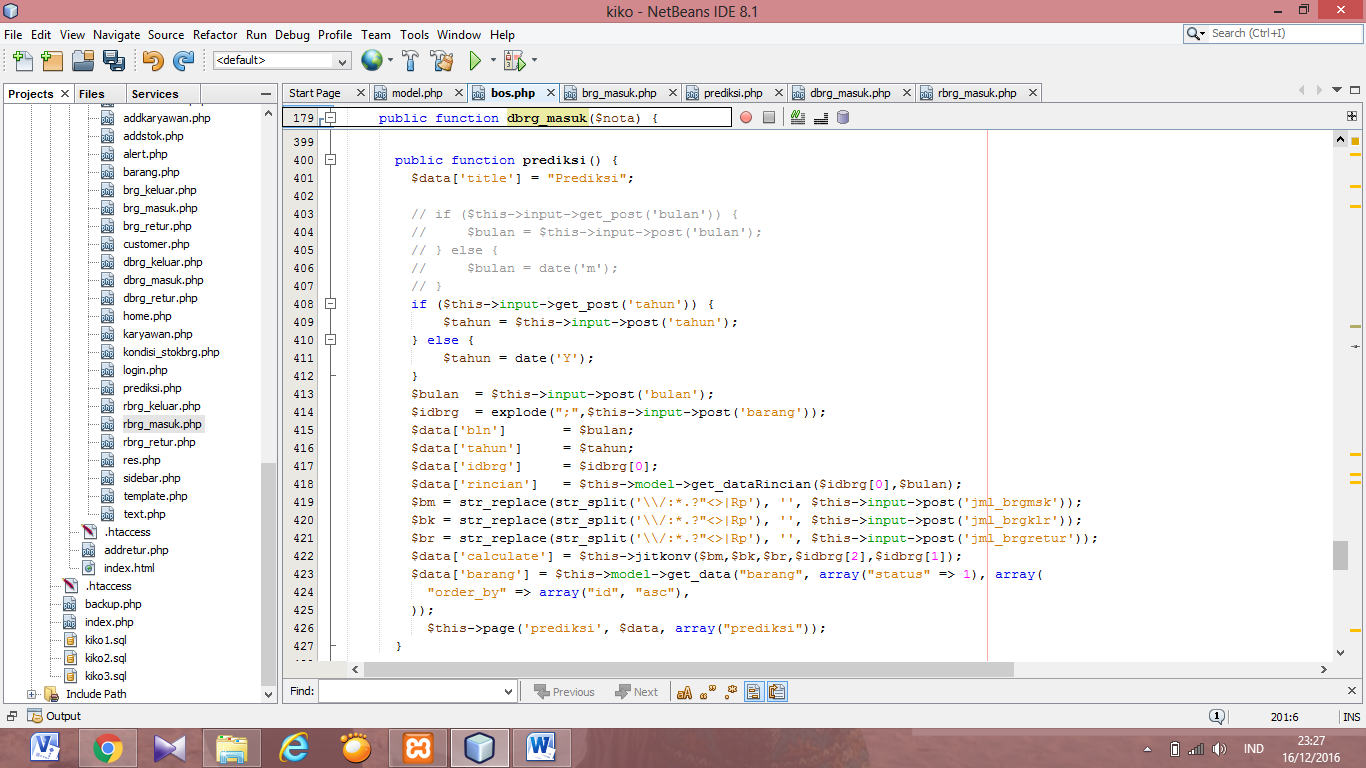
Gambar 4.14 *Method* rbrg\_masuk()

Gambar 4.14 merupakan kode *method r*brg\_masuk() dimana *method* tersebut berfungsi untuk menampilkan nota pembelian barang masuk dan detail pembelian barang masuk



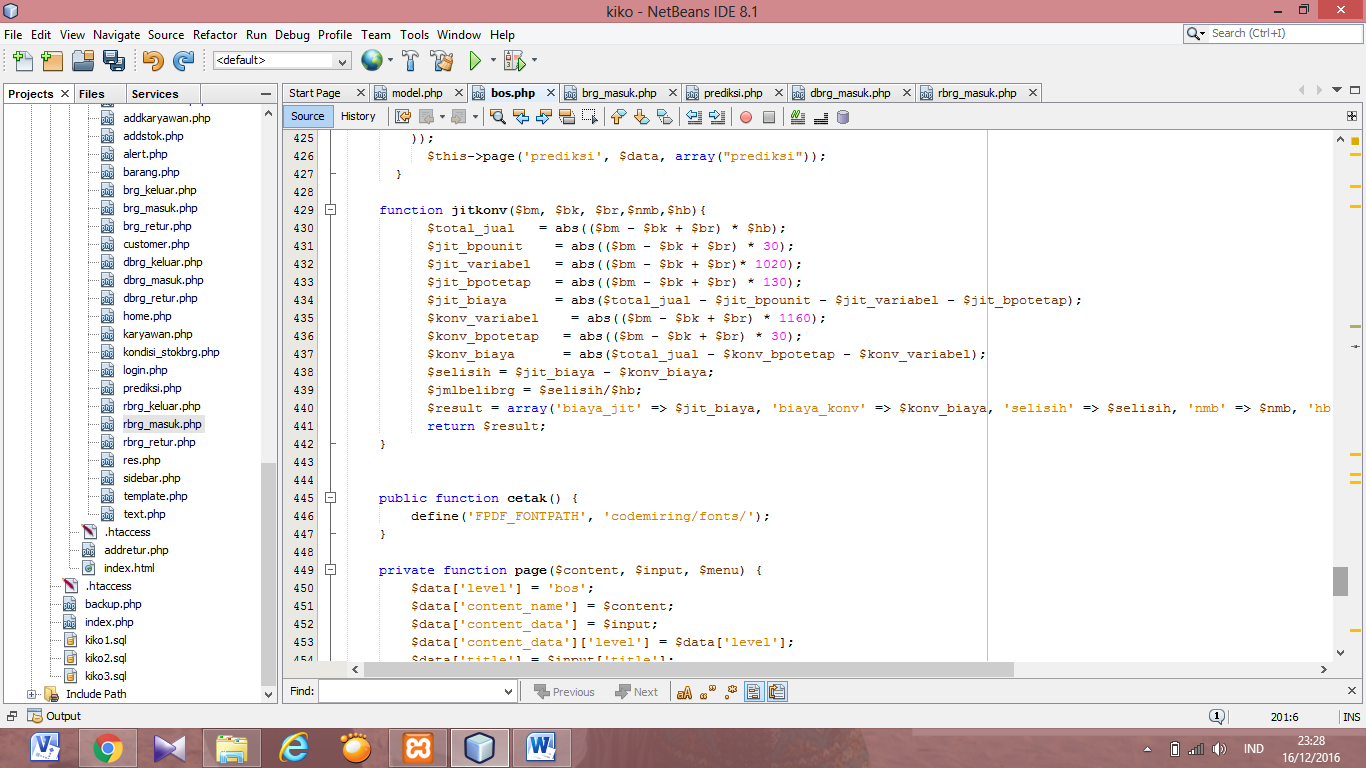
Gambar 4.15 *Method* dbrg­\_masuk($nota)

Gambar 4.15 merupakan kode *method* dbrg\_masuk($nota) dimana *method* tersebut digunakan untuk menambah dan menyimpan nota ke database secara otomatis



Gambar 4.16 *Method* prediksi()

Gambar 4.16 merupakan kode *method* prediksi() dimana *method* ini adalah implementasi metode *Just In Time* yang digunakan untuk memprediksi jumlah pengambilan stok ke dalam bahasa pemprograman. Pada code tersebut tertulis alur perhitungan data hingga penampilan hasil perhitungan.

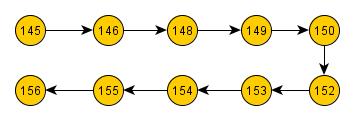


Gambar 4.17 *Method* jitkonv($bm, $bk, $br,$nmb,$hb)

Gambar 4.17 merupakan barisan code perhitungan perbandingan laba yang menggunakan metode secara konvensioal dengan yang menggunakan metode *Just In Time*.

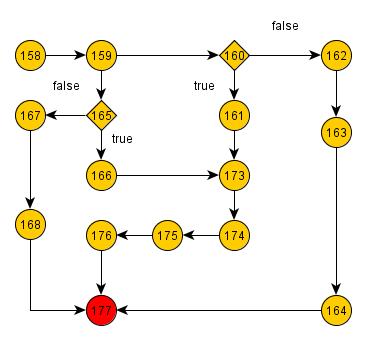
1. Diagram Alir

Diagram alir merupakan penulisan algoritma programan dengan menggunakan simbol-simbol yang dapat memudahkan dalam memahami alur program. Berikut merupakan alur *method* yang ada di dalam sistem informasi pengambilan stok barang menggunakan metode *Just In Time*.



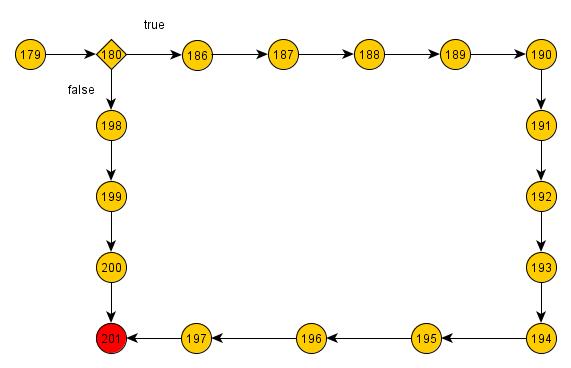
Gambar 4.18 Diagram Alir *Method* brg\_masuk()

Gambar 4.18 merupakan diagram alir *method* brg\_masuk(). Kode program tersebut dimulai dari baris ke 145 kemudian diteruskan sampai baris ke 156. Karena pada *method* ini tidak ada *looping* atau perulangan maka gambar atau alur kodenya lurus.



Gambar 4.19 Diagram Alir *Method* rbrg\_masuk()

Gambar 4.19 merupakan diagram alir *method* rbrg\_masuk(). Kode program tersebut dimulai pada baris ke 158 kemudian ke baris 159. Pada baris 159 terjadi percabangan, pada kode di baris 159 jika bulan yang dimasukkan benar maka kode akan diteruskan ke baris 160, kemudian ke baris 161 sampai 177 (end). Jika bulan yg dimasukkan salah maka kode akan diteruskan ke baris 160, kemudian ke baris 162 sampai 177 (end). Pada percabangan baris 159 jika tahun yang dimasukkan benar maka kode akan diteruskan pada baris 165, kemudian ke baris 166 sampai 177 (end). Apabila tahun yang dimasukkan salah maka kode akan diteruskan dari 165 menuju 167 sampai 177 (end).



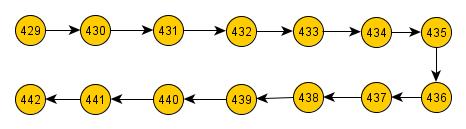
Gambar 4.20 Diagram Alir *Method* drbg\_masuk($nota)

Gambar 4.20 merupakan diagram alir *method* drbg\_masuk($nota). Kode program tersebut dimulai dari baris 179. Pada baris 180 terdapat percabangan, apabila data yang dimasukkan benar maka kode akan diteruskan ke baris 186 sampai dengan 201(end). Apabila data yang dimasukkan salah, maka kode akan diteruskan ke baris 198 sampai dengan 201(end).



Gambar 4.21 Diagram Alir *Method* prediksi()

Gambar 4.21 merupakan diagram alir *method* prediksi(). Kode pada program tersebut dimulai pada baris 400. Pada baris 408 terdapat percabangan, apabila data yangg dimasukkan benar, maka kode akan diteruskan pada baris 409 sampai dengan 427(end). Apabila data yang dimasukkan salah maka kode akan diteruskan pada baris 410, kemudian ke baris 411, selanjutnya ke baris 427(end).



Gambar 4. 22 Diagram Alir *Method* jitkonv($bm, $bk, $br,$nmb,$hb)

Gambar 4.22 merupakan diagram alir *method* jitkonv($bm, $bk, $nmb, $b). Kode *method* tersebut dimulai pada baris 429 kemudian diteruskan ke baris 430 sampai dengan baris 442(end). Karena pada kode tersebut tidak ada perulangan atau percabangan, maka diagram alir *method* ini lurus.

1. Perhitungan *Cyclomatic Complexity*

*Cyclomatic complexity* merupakan suatu sistem pengukuran yang menyediakan ukuran kuantitatif dari kompleksitas logika suatu program. *Cyclomatic complexity* digunakan untuk mencari jumlah path dalam suatu flowgraph. Perhitungan *cyclomatic complexity* V(G) adalah sebagai berikut :

*Method* brg\_masuk() : V(G) = Edge – Node + 2

= 9 – 10 + 2 = 1

*Method* rbrg\_masuk() : V(G) = Edge – Node + 2

= 18 – 16 + 2 = 4

*Method* dbrg\_masuk($nota) : V(G) = Edge – Node + 2

= 18 – 18 + 2 = 2

*Method* prediksi() : V(G) = Edge – Node + 2

= 20 – 20 + 2 = 2

*Method* jitkonv($bm, $bk, $br,$nmb,$hb): V(G) = Edge – Node + 2

= 13 – 14 + 2 = 1

Dari perhitungan diatas dapat di jelaskan bahwa jika V(G) = 1 itu berarti jumlah path dalam flowgraph tersebut adalah 1 (satu) baris saja dan itu berarti *method* tersebut digolongkan *method* yang sederhana. Sedangkan jika hasil V(G) < 1, maka jumlah path dalam flowgraph tersebut juga lebih dari satu, sehingga dapat disimpulkan bahwa *method* tersebut termasuk dalam golongan kompleks. Semakin banyak jumlah path dalam flowgraph maka semakin komplek pula sebuah program tersebut.

Dari hasil perhitungan diatas dapat diketahui bahwa yang mempunyai path hanya satu baris ada dua *method*, yaitu *method* brg\_masuk() dan *method* jitkonv($bm, $bk, $br,$nmb,$hb). Dan untuk *method* rbrg\_masuk() mempunyai jumlah 4 path, *method* dbrg\_masuk($nota) memiliki 2 path, dan *method* prediksi() memiliki 2 path.

1. Basis Set

Basis set merupakan gambaran alur path dalam suatu flowgraph. Berikut adalah basis set pada *method* yang di uji :

|  |  |  |
| --- | --- | --- |
| *Method* brg\_masuk() | : | 145 – 146 – 148 – 149 – 150 – 152 – 153 – 154 – 155 – 156 |
| *Method*rbrg\_masuk() | : | 158 – 159 – 165 – 166 – 173 – 174 – 175 – 176 – 177 (Path 1) |
|  |  | 158 – 159 – 165 – 167 – 168 – 177  (Path 2) |
|  |  | 158 – 159 – 160 – 161 – 173 – 174 – 175 – 176 – 177 (Path 3) |
|  |  | 158 – 159 – 160 – 162 – 163 – 164 – 177 (Path 4) |
| *Method* dbrg\_masuk() | : | 179 – 180 – 186 – 187 – 188 – 189 – 190 – 191 – 192 – 193 – 194 – 195 – 196 – 197 – 201 (Path 1) |
|  |  | 179 – 180 – 198 – 199 – 200 – 201  (Path 2) |
| *Method*prediksi() | : | 400 – 401 – 408 – 409 – 413 – 414 – 415 – 416 – 417 – 418 – 419 – 420 – 421 – 423 – 424 – 425 – 426 – 427 (Path 1) |
|  |  | 400 – 401 – 408 – 410 – 411 – 427  (Path 2) |
| *Method* jitkonv($bm, $bk, $br,$nmb,$hb) | : | 429 – 430 – 431 – 432 – 433 – 434 – 435 -436 – 437 – 438 – 439 – 440 – 442 |

1. *Test Case*

*Test case* merupakan suatu tes yang menggunakan struktur kendali dari desain prosedural. Pada tes ini menunjukkan bahwa apakah dalam proses menampilkan target pada fitur yang diharapkan sesuai dengan alur pada *method* atau tidak. Berikut adalah *tase case* pada fitur Data Barang Masuk dan fitur Prediksi.

Tabel 4.7 *Test Case* Fitur Data Barang Masuk

|  |  |
| --- | --- |
| *Method* brg\_masuk() | |
| *Test Case* | Menampilkan form data barang masuk |
| Target yang diharapkan | Proses menampilkan form data barang masuk |
| Hasil Pengujian | Benar |
| Path | 145 – 146 – 148 – 149 – 150 – 152 – 153 – 154 – 155 – 156 |
| *Method* rbrg\_masuk() | |
| *Test Case* | Jika berhasil menyimpan data |
| Target yang diharapkan | Menampilkan data barang yang telah diinputkan |
| Hasil Pengujian | Benar |
| Path | 158 – 159 – 165 – 166 – 173 – 174 – 175 – 176 – 177 (Path 1) |
| *Method* rbrg\_masuk() | |
| *Test Case* | Jika data tidak tersimpan |
| Target yang diharapkan | Menampilkan form data barang masuk |
| Hasil pengujian | Benar |
| Path | 158 – 159 – 165 – 167 – 168 – 177  (Path 2) |
| *Method* rbrg\_masuk() | |
| *Test Case* | Jika berhasil menyimpan data |
| Target yang diharapkan | Menampilkan data barang yang telah diinputkan |
| Hasil pengujian | Benar |
| Path | 158 – 159 – 160 – 161 – 173 – 174 – 175 – 176 – 177 (Path 3) |
| *Method* rbrg\_masuk() | |
| *Test Case* | Jika data tidak tersimpan |
| Target yang diharapkan | Menampilkan form data barang masuk |
| Hasil pengujian | Benar |
| path | 158 – 159 – 160 – 162 – 163 – 164 – 177 (Path 4) |
| *Method* dbrg\_masuk() | |
| *Test Case* | Jika terdapat data barang masuk |
| Target yang diharapkan | Menampilkan detail data barang masuk |
| Hasil Pengujian | Benar |
| Path | 179 – 180 – 186 – 187 – 188 – 189 – 190 – 191 – 192 – 193 – 194 – 195 – 196 – 197 – 201 (Path 1) |
| *Method* dbrg\_masuk() | |
| *Test Case* | Jika data barang masuk tidak ada |
| Target yang diharapkan | Kembali ke tampilan awal |
| Hasil Pengujian | Benar |
| Path | 179 – 180 – 198 – 199 – 200 – 201 (Path 2) |

Tabel 4.8 *Test Case* Fitur Prediksi

|  |  |
| --- | --- |
| *Method* prediksi() | |
| *Test Case* | Jika data sesuai |
| Target yang diharapkan | Menampilkan rincian data |
| Hasil Pengujian | Benar |
| Path | 400 – 401 – 408 – 409 – 413 – 414 – 415 – 416 – 417 – 418 – 419 – 420 – 421 – 423 – 424 – 425 – 426 – 427 (Path 1) |
| *Method* prediksi() | |
| *Test Case* | Jika data tidak sesuai |
| Target yang diharapkan | Data tidak ditampilkan |
| Hasil Pengujian | Benar |
| Path | 400 – 401 – 408 – 410 – 411 – 427 (Path 2) |
| *Method* jitkonv($bm, $bk, $br,$nmb,$hb) | |
| *Test Case* | Menghitung data yang ditampilkan |
| Target yang diharapkan | Proses menampilkan data hasil perhitungan data |
| Hasil Pengujian | Benar |
| Path | 429 – 430 – 431 – 432 – 433 – 434 – 435 -436 – 437 – 438 – 439 – 440 – 442 |

### Black Box Testing

*Black Box Testing* merupakan pengujian yang menekankan pada pengujian fungsionalitas sistem agar keluaran sesuai dengan apa yang diharapkan pengguna. Dokumentasi hasil pengujian dengan metode *black box testing* dapat dilihat pada tabel 4.9 berikut

Tabel 4.9 Black Box Testing Data Barang Masuk

|  |  |  |
| --- | --- | --- |
| **Kasus** | **Hasil** | **Ket** |
| *Field* pada *form* tambah data barang masuk terisi dengan benar, kemudian admin klik tombol “add” | Menampilkan list data pada tabel data barang masuk | [√ ] Berhasil  [ ] Gagal |
| *Field* pad *form* tambah data barang masuk terisi dengan benar, kemudian admin klik tombol “submit” | Menyipan data ke database dan menampilkan list data barang masuk dalam bentuk tabel | [√ ] Berhasil  [ ] Gagal |
| *Field* pada *form* tambah data barang masuk tidak dengan benar | Menampilkan pesan *error* pada *field* | [√ ] Berhasil  [ ] Gagal |
| *Field* pada *form* tambah data tidak terisi semua/terdapat form yang kosong, kemudian user klik tombol “submit” | Menampilkan pesan *error* pada *field* | [√ ] Berhasil  [ ] Gagal |

Tabel 4.10 Black Box Testing Perhitungan Prediksi Pengambilan Stok Barang

|  |  |  |
| --- | --- | --- |
| **Kasus** | **Hasil** | **Ket** |
| *Field* pada *form* tambah data terisi dengan benar, kemudian admin klik tombol “add” | Menampilkan perhitungan data barang masuk, data barang keluar, dan data barang retur | [√ ] Berhasil  [ ] Gagal |
| *Field* pad *form* tambah data barang masuk terisi dengan benar, kemudian admin klik tombol “start” | Menampilkan hasil perhitungan pengambilan stok barang beserta penjelasannya. | [√ ] Berhasil  [ ] Gagal |

# BAB 5. HASIL DAN PEMBAHASAN

Bab ini menjelaskan tentang hasil penelitian dan hasil pembuatan aplikasi serta pembahasannya. Penilitian ini menghasilkan sebuah aplikasi yang mampu meprediksi pengambilan stok barang menggunakan metode *Just In Time*. Pembahasan bertujuan untuk menjelaskan bagiamana penelitian ini menjawab rumusan masalah dan tujuan dari aplikasi.

## Hasil Penelitian

Hasil penelitian yang diperoleh dari penelitian ini adalah aplikasi prediski pengambilan stok barang yang berguna untuk memprediksi jumlah barang yang akan diambil bulan berikutnya pada PT. Irawan Djaja Agung. Aplikasi ini dibangung dengan menggunakan metode *Just In Time* yang didalamnya memasukkan semua unsur BOP, dengan melakukan penelitian inilah penulis dapat mengetahui BOP yang sudah ditetapkan oleh perusahaan untuk menghitung seluruh laba yang diperoleh oleh perusahaan tersebut. Selain BOP, jumlah barang masuk, barang keluar, serta barang retur juga tercantum dalam aplikasi ini, sehingga beberapa hasil penelitian yang penulis dapatkan dari wawancara adalah BOP, data barang, data barang masuk, data barang keluar, serta data barang retur. Data-data tersebut tercantum di pembahasan dalam bab ini.

## Hasil Pembuatan Aplikasi

Hasil penelitian ini menjelaskan tentang *output* dari pembuatan aplikasi prediksi pengambilan stok barang menggunakan metode *Just In Time* beserta fitur-fiturnya. Aplikasi ini dapat di akses oleh dua pihak kepentingan, yaitu admin, owner, dan bagian gudang. Fitur-fitur yang terdapat dalam aplikasi prediksi pengambilan stok barang adalah sebagai berikut.

### Tampilan Fitur *Login*

Fitur *login* merupakan tampilan halaman utama yang otomatis terbuka saat aplikasi diakses. Halaman ini digunakan oleh *user* admin, *user* owner, maupun *user* bagian gudang sebelum bisa mengakses halaman beranda aplikasi. *User* diwajibkan memiliki *username* dan *password* untuk bisa mengakses fitur-fitur yang ada di dalam aplikasi. Pengecekan hak akses *user* juga dilakukan pada saat proses *login* dikerjakan. *User* dengan level admin diarahkan ke halaman utama admin, *user* dengan level owner diarahkan ke halaman utama owner, dan *user* dengan level bagian gudang diarahkan ke halaman utama bagian gudang. Tampilan fitur *login* dapat dilihat pada gambar 5.1 berikut



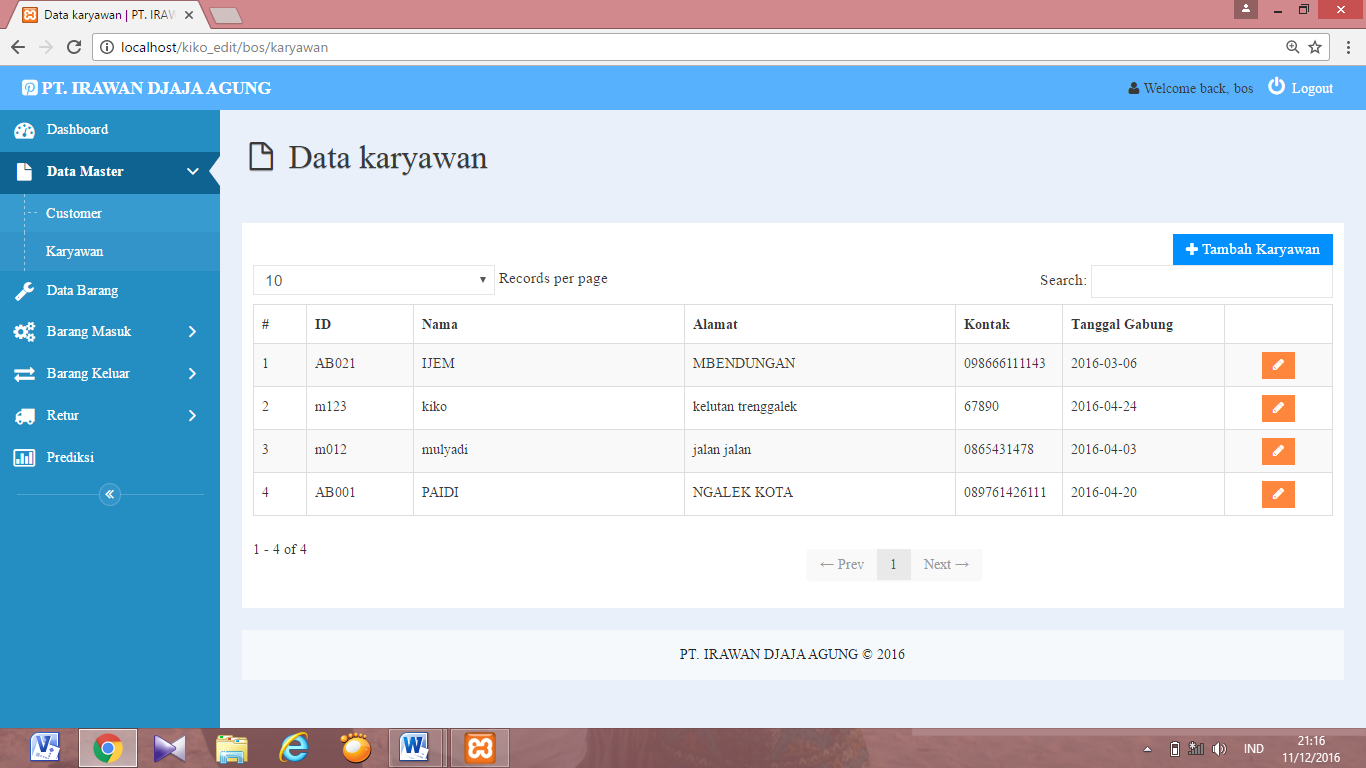
Gambar 5.1 Tampilan Fitur *Login*

### Tampilan *Home*

Setelah memasukkan username dan password dengan benar, user akan masuk di halaman utama aplikasi/*home* dengan tampilan grafik rekap penjualan dan pembelian di PT. Irawan Djaja Agung Jember.

### Tampilan Fitur Karyawan

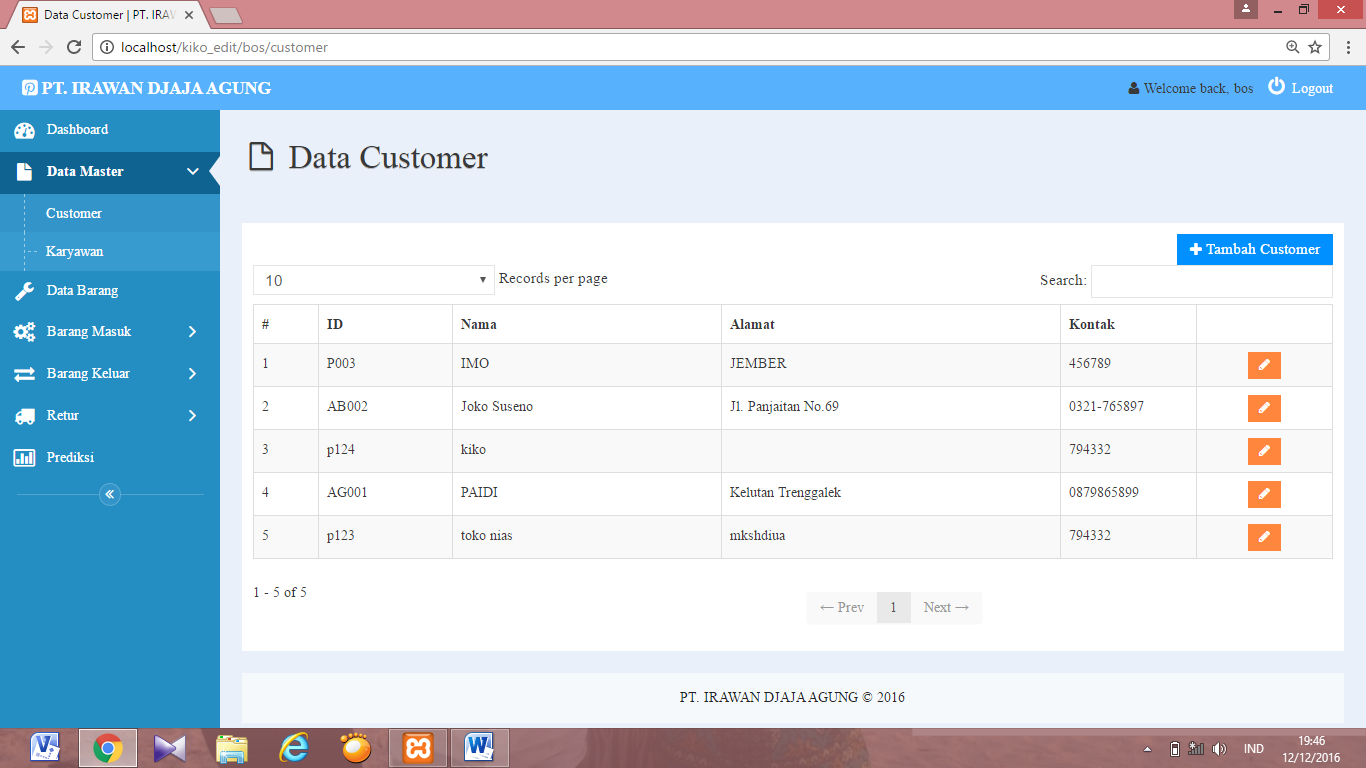
Fitur karyawan merupakan fitur yang digunakan untuk mencatat data-data karyawan di PT. Irawan Djaja Agung, dalam fitur ini *user* dapat menambah dan mengedit data karyawan. Gambar 5.2 berikut merupakan tampilan utama data karyawan, sedangkan untuk tampilan tambah dan *edit* data dapat dilihat pada lampiran B1.



Gambar 5.2 Tampilan Fitur Karyawan

### Tampilan Fitur Customer

Fitur customer merupakan fitur yang digunakan untuk mencatat data-data customer yang berlangganan dengan PT. Irawan Djaja Agung, dalam fitur ini *user* dapat menambah dan mengedit data karyawan. Gambar 5.3 bertikut merupakan tampilan utama data customer, sedangkan untuk tampilan tambah dan *edit* data dapat dilihat pada lampiran B2.



Gambar 5.3 Tampilan Fitur Customer

### Tampilan Fitur Data Barang

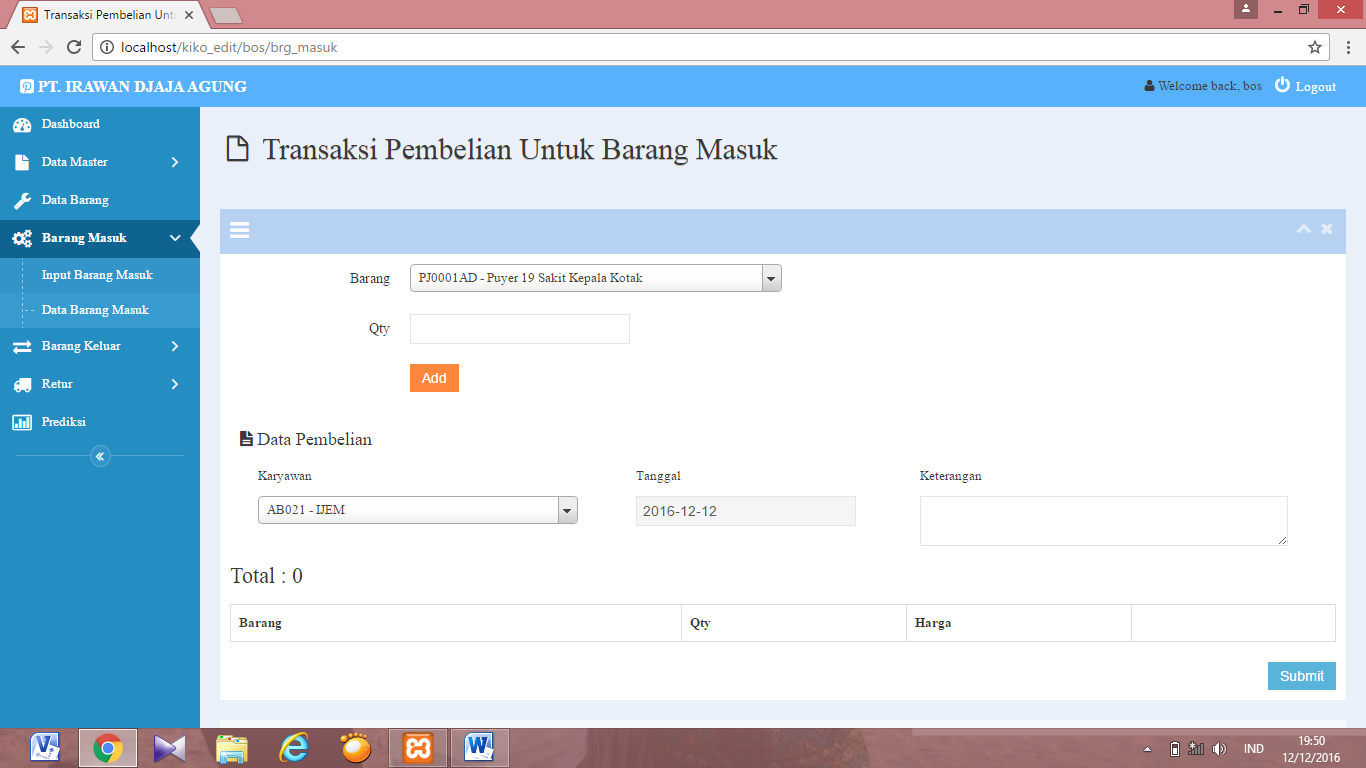
Fitur data barang merupakan fitur yang digunakan untuk mencatat data barang yang terdapat pada PT. Irawan Djaja Agung, dalam fitur ini *user* dapat menambah dan mengedit data barang. Gambar 5.4 berikut merupakan tampilan utama data barang, sedangkan untuk tampilan tambah dan *edit* data barang dapat dilihat pada lampiran B3.



Gambar 5.4 Tampilan Fitur Data Barang

### Tampilan Fitur Input Data Barang Masuk

Fitur input data barang massuk merupakan fituryang digunakan untuk mencatat transaksi pembelian barang dari pemasok ke PT. Irawan Djaja Agung, dalam fitur ini *user* dapat menambah dan menghapus data barang masuk. Gambar 5.5 berikut merupakan tampilan utama data barang masuk.



Gambar 5.5 Tampilan fitur Input Barang Masuk

### Tampilan Fitur Data Barang Masuk

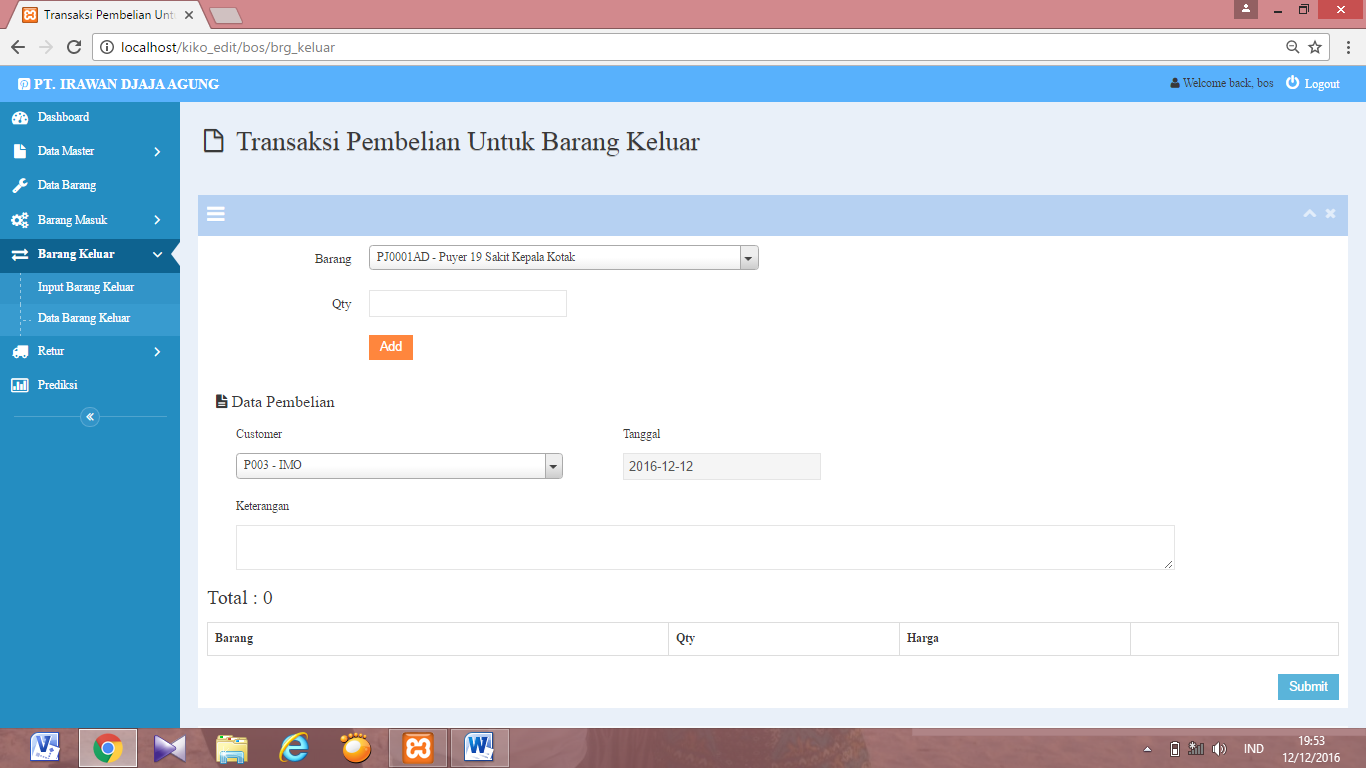
Fitur data barang masuk merupakan fitur yang digunakan melihat rekapitulasi transaksi data barang yang masuk ke PT. Irawan Djaja Agung, dalam fitur ini *user* hanya dapat melihat data barang masuk dan detail barang masuk saja. Gambar 5.6 berikut merupakan tampilan utama data barang masuk, sedangkan untuk tampilan detail data barang masuk dapat dilihat pada lampiran B4.



Gambar 5.6 Tampilan Fitur Data Barang Masuk

### Tampilan Fitur Input Data Barang Keluar

Fitur input data barang keluar merupakan fitur yang digunakan untuk mencatat transaksi penjualan barang dari PT. Irawan Djaja Agung kepada customer, dalam fitur ini *user* dapat menambah dan menghapus data barang keluar. Gambar 5.7 berikut merupakan tampilan utama data barang keluar



Gambar 5.7 Tampilan Fitur Barang Keluar

### Tampilan Fitur Data Barang Keluar

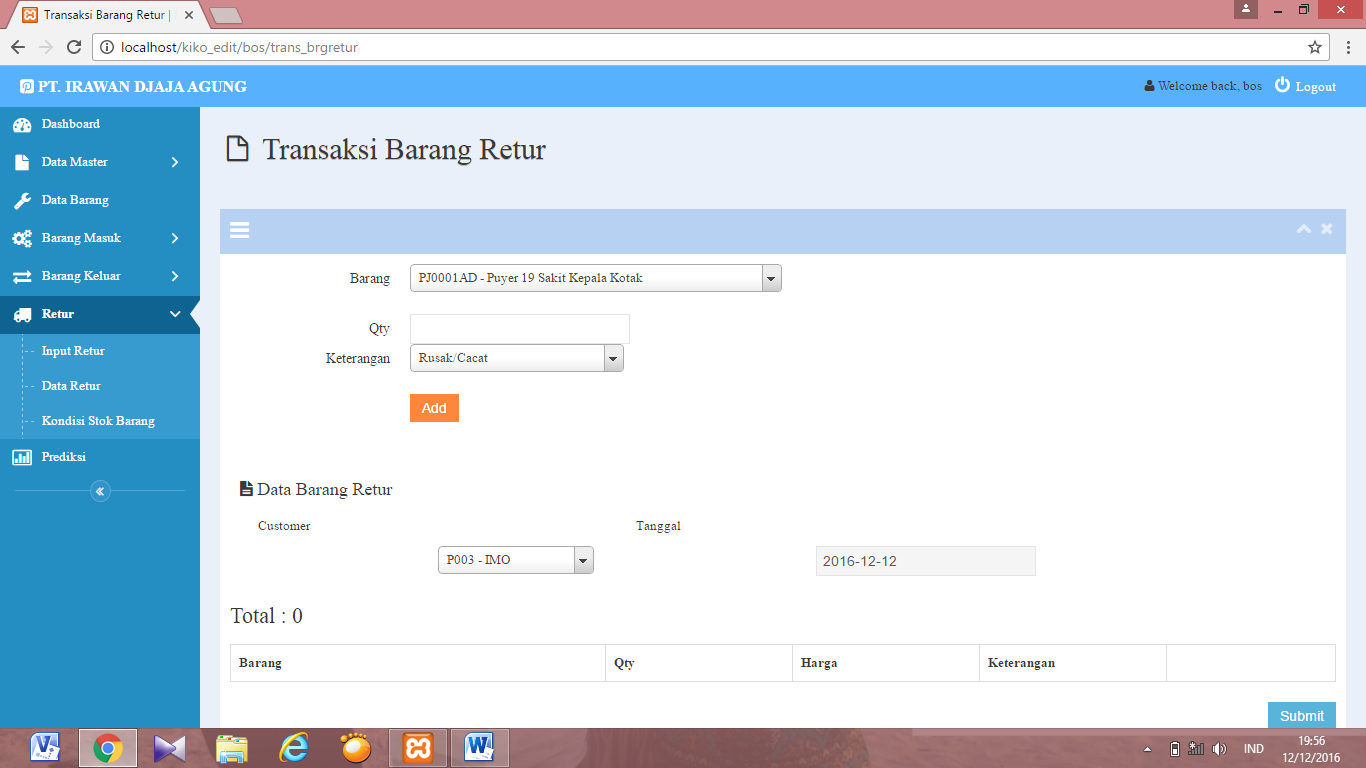
Fitur data barang keluar merupakan fitur yang digunakan untuk melihat rekapitulasi tranksaksi data barang yang keluar dari PT. Irawan Djaja Agung, dalam fitur ini *user* hanya dapat melihat data barang keluar dan detail barang keluar saja. Gambar 5.8 berikut merupakan tampilan utama data barang keluar, sedangkan untuk tampilan detail data barang keluar dapat dilihat pada lampiran B7.



Gambar 5.8 Tampilan Fitur Data Barang Keluar

### Tampilan Fitur Input Data Barang Retur

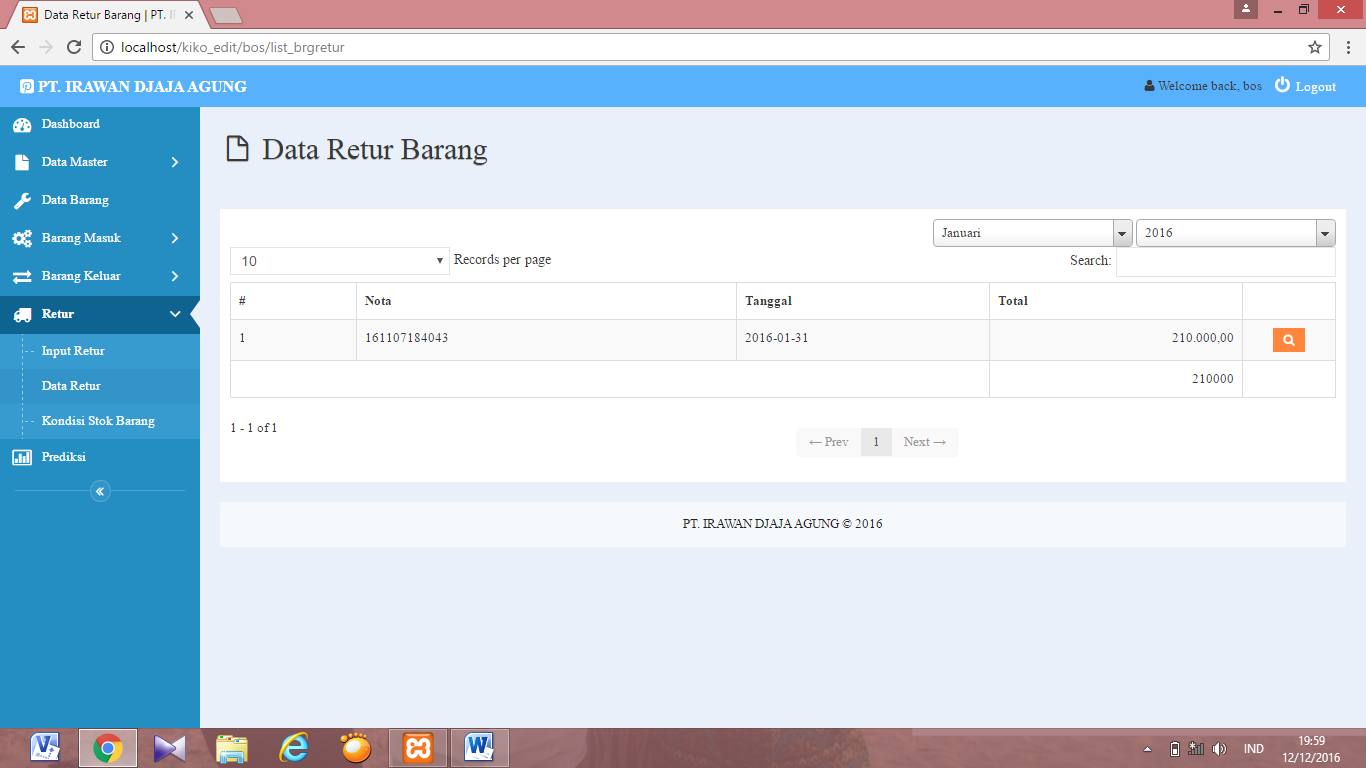
Fitur input data barang retur merupakan fitur yang digunakan untuk mencatat transaksi pengembalian barang dari customer ke PT. Irawan Djaja Agung karena cacat atau kadaluarsa, dalam fitur ini *user* dapat menambah dan menghapus data barang retur. Gambar 5.9 berikut merupakan tampilan utama data barang retur.



Gambar 5.9 Tampilan Fitur Input Barang Retur

### Tampilan Fitur Data Barang Retur

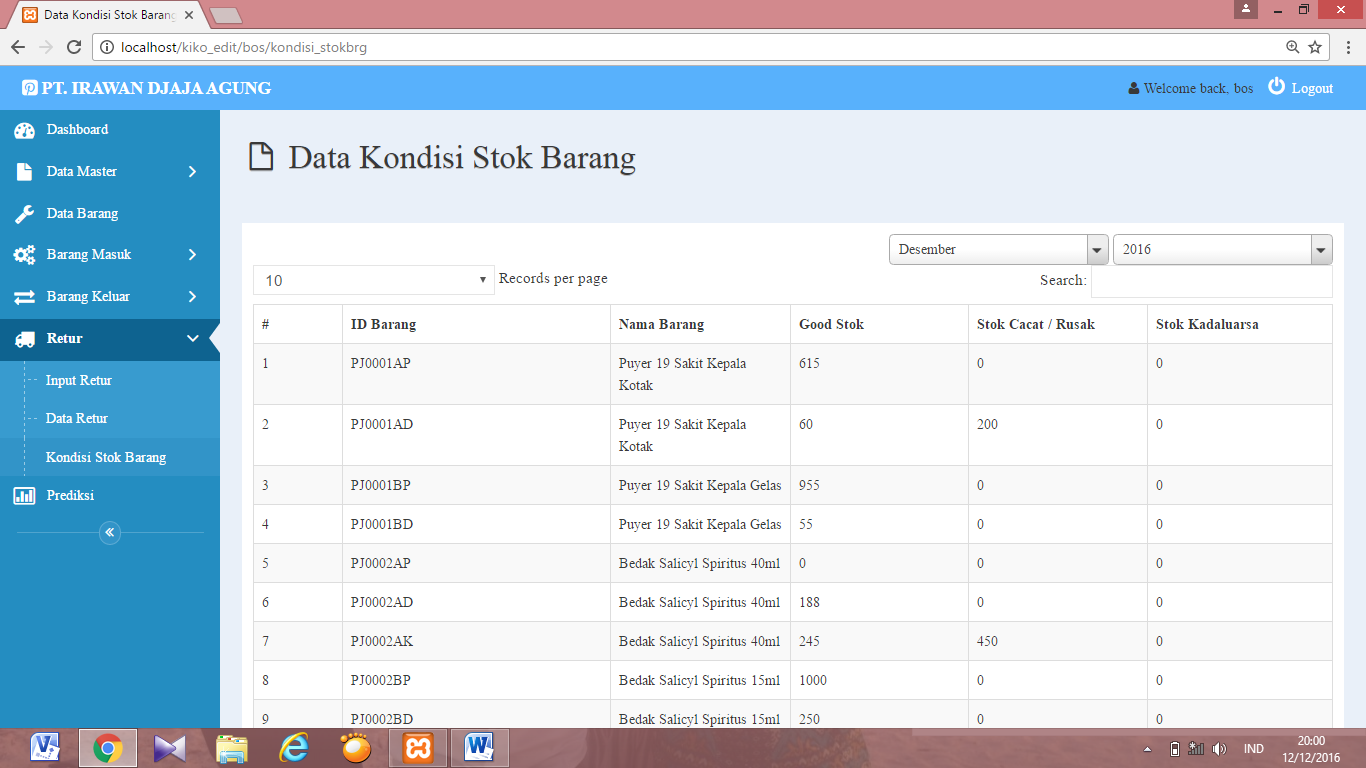
Fitur data barang retur merupakan fitur yang digunakan untuk melihat rekapitulasi tranksaksi data barang retur, dalam fitur ini *user* hanya dapat melihat data barang retur dan detail barang retur saja. Gambar 5.10 berikut merupakan tampilan utama data barang retur, sedangkan untuk tampilan detail data barang retur dapat dilihat pada lampiran B6.



Gambar 5.10 Tampilan Fitur Data Barang Retur

### Tampilan Fitur Kondisi Barang

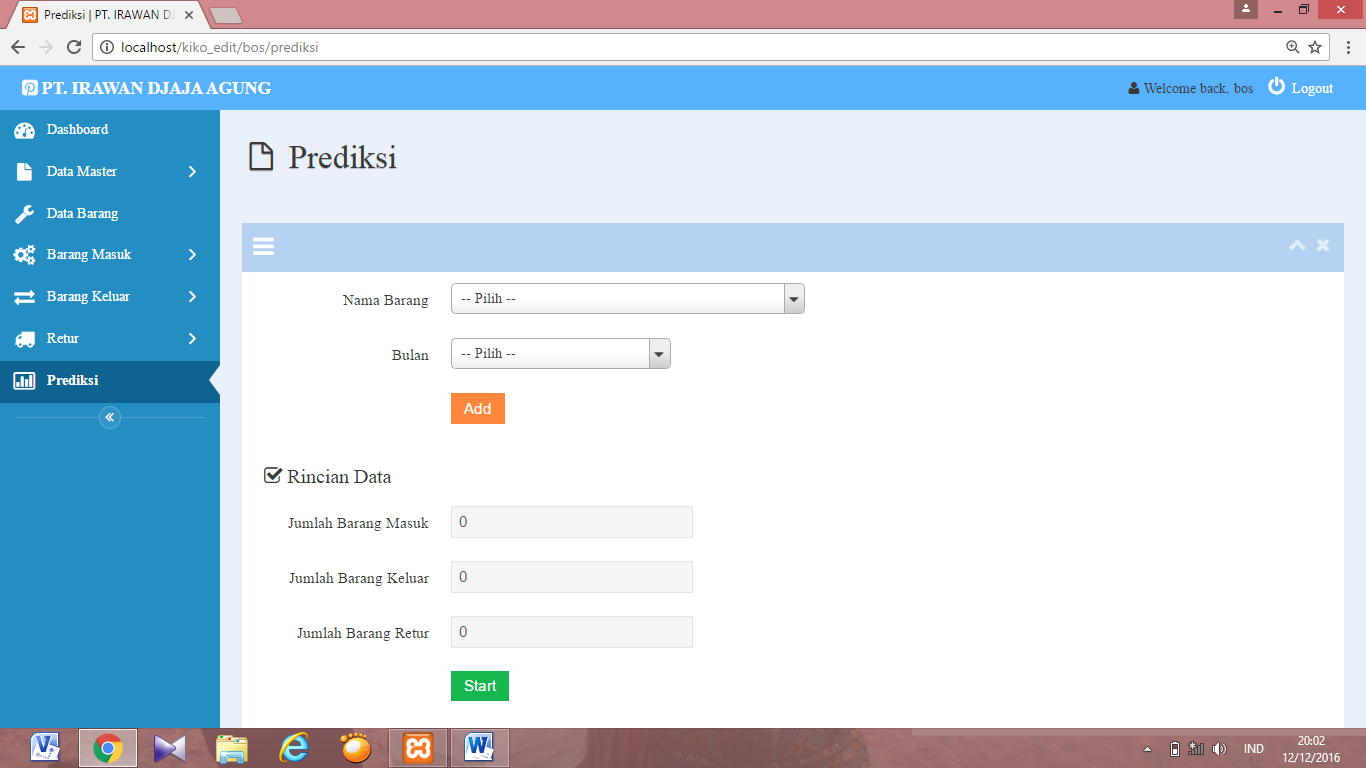
Fitur kondisi barang merupakan fitur yang digunakan untuk melihat kondisi dan stok barang di PT. Irawan Djaja Agung, dalam fitur ini *user* hanya dapat melihat kondisi dan stok barang dalam bentuk tabel. Gambar 5.11 berikut merupakan tampilan utama fitur kondisi barang.



Gambar 5.11 Tampilan Fitur Kondisi Stok Barang

### Tampilan Fitur Prediksi Pengambilan Stok Barang

Fitur prediksi pengambilan stok barang merupakan fitur yang digunakan untuk memprediksi pengambilan barang untuk stok selanjutnya. Dalam fitur ini *user* dapat mengetahui jumlah barang yang masuk, barang yang keluar, dan barang yang dikembalikan dalam satu id barang pada kurun waktu satu bulan. Selain itu *user* juga dapat mengetahui jumlah laba yang diperoleh dari hasil penghitungan prediksi menggunakan metode *Just In Time* tersebut. Dari situlah *user* juga mengetahui berapa banyak barang yang dapat diambil untuk selanjutnya. Gambar 5.12 berikut merupakan tampilan utama fitur prediksi pengambilan stok barang.



Gambar 5.12 Tampilan Fitur Prediksi

## Pembahasan

Pada sub bab ini akan dibahas tingkat akurasi system prediksi mengunakan metode *Just In Time*. Kecocokan data dengan metode sangat diperhitungkan agar data yang diprediksi memiliki akurasi yang tinggi dan mendekati data riil.

Untuk dapat melakukan prediksi data selanjutnya setidaknya data yang digunakan paling sedikit adalah tiga periode. Data yang diperlukan harus terisi seperti data penjualan dan data tahun yang akan diprediksi. Hasil uji coba prediksi dilakukan pada salah satu data penjualan produk Balsem Hijau Cap Betet ukuran kecil 7,5 gram. Dan tahun yang dipilih adalah tahun 2016.

Sebelum melakukan prediksi atau implementasi prediksi, data yang digunakan untuk prediksi adalah data penjualan Balsem Cap Betet ukuran kecil 7,5 gram selama tiga tahun yaitu tahun 2013, 2014, dan 2015. Berikut adalah data penjualan Balsem Hijau Cap Betet ukuran kecil 7,5 gram selama tiga tahun berturut-turut di PT Irawan Djaja Agung.

Tabel 5.1 Data Penjualan Balsem Hijau Cap Betet Ukuran Kecil 7,5 gram Tahun 2013

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TGL** | **BULAN** | | | | | | | | | | | |
| **Jan** | **Feb** | **Mart** | **Apr** | **Mei** | **Jun** | **Jul** | **Agst** | **Sept** | **Okt** | **Nov** | **Des** |
| 1 | 196.700 | 397.900 | 320.800 | 672.700 | 238.600 | 423.700 | 448.500 | - | 444.500 | 469.500 | 195.800 | 779.800 |
| 2 | 320.700 | 732.500 | 557.400 | 376.700 | 316.300 | 466.500 | 230.700 | - | 372.700 | 245.600 | 257.300 | 698.700 |
| 3 | 477.200 | 639.300 | 344.600 | 1.088.800 | 214.700 | 203.500 | 468.400 | 413.500 | 619.700 | 491.000 | 508.600 | 549.900 |
| 4 | 99.600 | 472.400 | 543.100 | 424.800 | 342.600 | 378.500 | 350.900 | 556.100 | 699.300 | 256.500 | 251.800 | 392.200 |
| 5 | 578.300 | 292.200 | 545.800 | 552.500 | 265.100 | 428.800 | 360.700 | 507.900 | 420.500 | - | 552.800 | 219.300 |
| 6 | 393.200 | 279.700 | 279.500 | 332.100 | 503.600 | 155.800 | 278.200 | 842.000 | 446.300 | 759.600 | 421.500 | 187.300 |
| 7 | 450.100 | 323.900 | 452.400 | 377.400 | 421.600 | 669.600 | 251.400 | 964.400 | 651.400 | 1.018.600 | 376.900 | 743.200 |
| 8 | 415.000 | 295.900 | 78.000 | 293.700 | 268.200 | 471.400 | 311.900 | 1.196.600 | 541.600 | 416.500 | 559.600 | 725.500 |
| 9 | 499.600 | 343.100 | 389.800 | 276.100 | 319.300 | 326.800 | 581.200 | 534.300 | 376.700 | 401.100 | 498.400 | 479.400 |
| 10 | 268.000 | 498.500 | 304.900 | 464.900 | 354.200 | 413.800 | 338.900 | 236.600 | 992.100 | 656.600 | 586.200 | 356.900 |
| 11 | 409.000 | 276.600 | 586.700 | 538.800 | 658.300 | 308.800 | 375.100 | 663.700 | 425.100 | 623.300 | 664.300 | 566.200 |
| 12 | 520.200 | 515.800 | 645.500 | 339.100 | 576.800 | 743.000 | 679.700 | 1.242.000 | 921.000 | 433.500 | 515.700 | 211.800 |
| 13 | 215.000 | 545.000 | 238.800 | 520.600 | 737.800 | 311.100 | 751.700 | 541.000 | 270.000 | 471.100 | 677.900 | - |
| 14 | 427.000 | 500.700 | 282.400 | 434.800 | 321.900 | 259.900 | 870.200 | 421.300 | 384.000 | 405.200 | 289.100 | - |
| 15 | 184.800 | 280.500 | 211.900 | 366.300 | 298.100 | 235.300 | 510.400 | 483.800 | 657.800 | 1.223.500 | 175.600 | 292.100 |
| 16 | 497.100 | 95.300 | 427.700 | 649.200 | 363.700 | 262.100 | 483.600 | 295.100 | 919.400 | 353.400 | - | 407.200 |
| 17 | 405.800 | 295.000 | 490.000 | 510.400 | 229.400 | 288.700 | 523.000 | 412.700 | 545.600 | 498.300 | 372.000 | 266.500 |
| 18 | 429.100 | 229.700 | 575.300 | 346.500 | 139.000 | 301.400 | 299.100 | 886.900 | 272.700 | 215.700 | 374.100 | 241.100 |
| 19 | 467.400 | 292.500 | 436.500 | 198.800 | 482.300 | 339.600 | 33.800 | 195.600 | 408.600 | 512.400 | 468.400 | 545.500 |
| 20 | 1.081.200 | 416.300 | 369.400 | 372.000 | 383.400 | 259.000 | - | 449.400 | 323.300 | 1.333.800 | 482.500 | 662.400 |
| 21 | 697.800 | 195.600 | 701.100 | 236.900 | 739.800 | 535.000 | 329.700 | 462.500 | 287.400 | 546.800 | 537.900 | 781.200 |
| 22 | 481.700 | 396.150 | 501.600 | 356.500 | 564.200 | 147.000 | 216.100 | 269.000 | 248.500 | 540.800 | 561.700 | 406.300 |
| 23 | 476.600 | 177.100 | 340.700 | 380.700 | 380.100 | 451.500 | 350.600 | 414.600 | 560.300 | 510.200 | 503.000 | 273.700 |
| 24 | 462.500 | 316.000 | 669.200 | 474.900 | 521.300 | 168.600 | 494.200 | 403.000 | 339.800 | 341.700 | 868.800 | 420.600 |
| 25 | 295.200 | 220.300 | 589.400 | 400.800 | 547.500 | 320.100 | 502.000 |  | 308.600 | 228.300 | 525.700 | 452.200 |
| 26 | 255.900 | 366.300 | 563.200 | 660.400 | 607.700 | 236.000 | 259.900 | 385.100 | 580.700 | 194.000 | 794.800 | 415.600 |
| 27 | 331.800 | 255.300 | 450.400 | 216.900 | 502.800 | 237.900 | 160.900 | 521.100 | 608.800 | 221.100 | 328.800 | 271.700 |
| 28 | 204.500 | 557.200 | 585.000 | 227.100 | 585.000 | 382.500 | - | 578.600 | 481.300 | 367.800 | 679.700 | 991.000 |
| 29 | - | - | 43.800 | 537.500 | 43.800 | 270.400 | - | 314.500 | 986.900 | 472.200 | 295.600 | 172.300 |
| 30 | 327.600 | - | 454.500 | 292.100 | 462.400 | 180.000 | - | - | 247.300 | 315.200 | 415.800 | 134.200 |
| 31 | 430.000 | - | 447.500 | - | 486.100 | - | - | 399.800 | - | 334.000 | - | 225.500 |
| **total** | **12.298.600** | **10.206.750** | **13.426.900** | **12.920.000** | **12.875.600** | **10.176.300** | **10.460.800** | **14.591.100** | **15.341.900** | **14.857.300** | **13.740.300** | **12.869.300** |

Tabel 5.2 Data Penjualan Balsem Hijau Cap Betet Ukuran Kecil 7,5 gram Tahun 2014

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tgl** | **BULAN** | | | | | | | | | | | |
| **Jan** | **Feb** | **Mart** | **Apr** | **Mei** | **Jun** | **Jul** | **Agst** | **Sept** | **Okt** | **Nov** | **Des** |
| 1 | 117.500 | 388.500 | 472.900 | 396.000 | 465.200 | 406.900 | 366.200 | 1.565.300 | 1.066.200 | 657.800 | - | 623.000 |
| 2 | 531.800 | 674.700 | 304.400 | 518.900 | 427.700 | 631.000 | 190.900 | 1.046.800 | 685.700 | 476.200 | 830.800 | 772.000 |
| 3 | 367.500 | 520.100 | 365.100 | - | 461.200 | 446.400 | 475.700 | 801.300 | 258.800 | 365.400 | 401.500 | 543.000 |
| 4 | 906.800 | 967.900 | 319.500 | - | 811.700 | 199.700 | 141.200 | 1.137.500 | 592.900 | 625.800 | 572.000 | 538.700 |
| 5 | 893.700 | 361.700 | 706.000 | - | 263.800 | 402.200 | 1.570.000 | 1.113.400 | 545.400 | 647.400 | 417.000 | 122.000 |
| 6 | 645.700 | 671.300 | 451.400 | 421.400 | 339.100 | 776.900 | 332.600 | 661.300 | 812.600 | 591.000 | 917.200 | 189.500 |
| 7 | 800.800 | 354.500 | 363.100 | 244.400 | 280.200 | - | 375.100 | 1.011.300 | 538.000 | 540.500 | 409.400 | 615.100 |
| 8 | 622.500 | 419.300 | 589.800 | 601.800 | 520.100 | 1.066.800 | 241.500 | 1.880.000 | 382.700 | 355.500 | 1.063.900 | 724.500 |
| 9 | 412.900 | 326.700 | 460.500 | 320.500 | 267.000 | 248.600 | 337.600 | 734.000 | 976.500 | 663.300 | 460.000 | 345.400 |
| 10 | - | 339.000 | 508.200 | 555.700 | 352.500 | 504.000 | 2.295.500 | 739.500 | 552.400 | 245.000 | 500.500 | 266.100 |
| 11 | 373.900 | 344.000 | 247.000 | 251.600 | 491.900 | 615.900 | 189.500 | 635.500 | 341.500 | 370.000 | 559.700 | 628.000 |
| 12 | 533.100 | 281.800 | 348.000 | 355.400 | 466.000 | 289.100 | 404.500 | 682.800 | 890.300 | 18.139.000 | 3.134.800 | 481.500 |
| 13 | 242.500 | 687.200 | 838.800 | 671.300 | 339.000 | 252.900 | 375.200 | 464.500 | 544.500 | 396.400 | 467.400 | 212.800 |
| 14 | 903.100 | 548.400 | 203.600 | 645.600 | 544.700 | 459.500 | 271.500 | 676.700 | 630.500 | 786.500 | 772.700 | 960.500 |
| 15 | 337.200 | 358.600 | 596.900 | 501.000 | 256.700 | 249.100 | 265.500 | 741.800 | 613.800 | 310.900 | 636.200 | 215.800 |
| 16 | 689.000 | 1.138.500 | 234.000 | 524.300 | 420.500 | 133.900 | 234.100 | 833.500 | 560.200 | 314.000 | 527.500 | 618.000 |
| 17 | 651.000 | 581.500 | 381.500 | 390.800 | 480.000 | 1.170.700 | - | 568.500 | 616.000 | 450.800 | 400.800 | 515.200 |
| 18 | 417.800 | 440.500 | 479.800 | 508.700 | 440.100 | 160.100 | - | 409.300 | 485.500 | 462.400 | 378.300 | 428.200 |
| 19 | 424.700 | 116.800 | 262.900 | 844.400 | 550.800 | 212.900 | - | 883.700 | 616.500 | 1.142.800 | 577.700 | 237.300 |
| 20 | 567.900 | 547.700 | 337.500 | 290.900 | 493.700 | 415.000 | - | 772.000 | 681.000 | 598.000 | 269.800 | 277.000 |
| 21 | 1.322.400 | 679.300 | 260.200 | 318.400 | 343.100 | 193.800 | 897.500 | 379.700 | 215.700 | 422.300 | 568.500 | 474.500 |
| 22 | 294.700 | 333.400 | 617.100 | 649.000 | 398.600 | 128.400 | 1.158.500 | 606.000 | 390.000 | 304.600 | 590.200 | 526.200 |
| 23 | 504.100 | 441.700 | 253.500 | 281.900 | 280.800 | 163.300 | 1.114.300 | 104.000 | 428.000 | 568.900 | 558.500 | 286.000 |
| 24 | 256.100 | 389.900 | 388.500 | 469.500 | 787.000 | 131.000 | 634.000 | 556.700 | 397.400 | 378.400 | 460.700 | - |
| 25 | 394.000 | 355.000 | 357.600 | 623.800 | 482.500 | 231.500 | 1.621.500 | 810.000 | 435.200 | 706.100 | 64.000 | - |
| 26 | 701.900 | 418.100 | 474.400 | 603.200 | 574.000 | 90.600 | 3.011.000 | 498.700 | 391.800 | 1.549.000 | 422.700 | - |
| 27 | 452.800 | 415.300 | 487.300 | 660.900 | 399.700 | 236.400 | 2.248.800 | 405.300 | 607.300 | 642.000 | 630.500 | 440.500 |
| 28 | 515.600 | 309.500 | 309.700 | 549.000 | 332.700 | 389.500 | 1.290.500 | 392.000 | 409.000 | 581.100 | 172.000 | 649.700 |
| 29 | 651.500 | - | 302.500 | 285.900 | 245.900 | 246.600 | 2.653.400 | 185.500 | 462.000 | 350.000 | 708.900 | 189.000 |
| 30 | 215.000 | - | 360.700 | 293.600 | 428.700 | - | 920.500 | - | 638.300 | 553.500 | 757.000 | 442.000 |
| 31 | 274.500 | - | 547.600 | - | 781.000 | - | 714.000 | 887.000 | - | 203.000 | - | 184.200 |
| **total** | **16.022.000** | **13.410.900** | **12.830.000** | **12.777.900** | **13.725.900** | **10.452.700** | **24.330.600** | **22.183.600** | **16.765.700** | **34.397.600** | **18.230.200** | **12.505.700** |

Tabel 5.3 Data Penjualan Balsem Hijau Cap Betet Ukuran Kecil 7,5 gram Tahun 2015

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tgl** | **BULAN** | | | | | | | | | | | |
| **Jan** | **Feb** | **Mart** | **Apr** | **Mei** | **Jun** | **Jul** | **Agst** | **Sept** | **Okt** | **Nov** | **Des** |
| 1 | - | 480.000 | 427.500 | 378.200 | - | 209.000 | 174.000 | 974.500 | 1.066.500 | 393.000 | 4.341.000 | 486.200 |
| 2 | 363.500 | 539.500 | 666.200 | 302.100 | - | 418.500 | 337.000 | 528.000 | 1.450.300 | 333.000 | 280.000 | 638.000 |
| 3 | 600.600 | 436.500 | 170.500 | 723.500 | 461.700 | 483.100 | 508.000 | 820.500 | 380.600 | 2.094.500 | 435.100 | 378.000 |
| 4 | 819.000 | 469.700 | 539.800 | 814.500 | 383.100 | 215.000 | 304.700 | 802.000 | 651.600 | 250.000 | 536.800 | 520.900 |
| 5 | 736.500 | 303.200 | 111.900 | 258.000 | 408.500 | - | 516.000 | 627.500 | 772.400 | 401.000 | 344.900 | 212.000 |
| 6 | 413.500 | 521.000 | 209.500 | 401.700 | 399.000 | 860.400 | - | 229.500 | 460.300 | 421.100 | 551.700 | 226.000 |
| 7 | 581.000 | 732.500 | 1.753.800 | 357.800 | 568.000 | 247.500 | - | 746.000 | 551.900 | 553.500 | 158.000 | 375.000 |
| 8 | 544.000 | 256.000 | 1.234.900 | 269.500 | 461.000 | 266.100 | - | 793.500 | 303.800 | 2.274.500 | 562.000 | 339.400 |
| 9 | 181.000 | 746.300 | 457.500 | 463.900 | 284.000 | 325.100 | - | 832.800 | 560.200 | 566.900 | 218.000 | 597.700 |
| 10 | 766.400 | 455.900 | 311.500 | 676.200 | 219.900 | 260.500 | - | 887.600 | 336.300 | 488.700 | 657.000 | 386.900 |
| 11 | 375.900 | 528.400 | 265.900 | 227.000 | 452.500 | 195.500 | 1.003.000 | 536.400 | 455.400 | 560.500 | 288.600 | 189.400 |
| 12 | 1.481.000 | 346.800 | 285.400 | 566.000 | 386.500 | 298.900 | 1.076.000 | 343.000 | 455.400 | 323.000 | 414.600 | 350.000 |
| 13 | 460.000 | 472.500 | 676.800 | 267.600 | 767.500 | 289.800 | 1.245.500 | 159.500 | 392.700 | 389.800 | 325.800 | 441.000 |
| 14 | 454.700 | 605.500 | 635.500 | 82.600 | - | 376.200 | 1.657.000 | 417.000 | 479.000 | 383.000 | 370.400 | 295.000 |
| 15 | 493.600 | 1.016.500 | 1.137.000 | 278.000 | 871.500 | 211.600 | 2.608.800 | 582.300 | 215.700 | 307.000 | 11.730.500 | 333.300 |
| 16 | 342.000 | 756.000 | 1.412.200 | 221.000 | 319.500 | 272.700 | 2.099.000 | 721.500 | 421.500 | 777.800 | 639.100 | 3.794.200 |
| 17 | 889.000 | 510.500 | 1.366.500 | 544.000 | 455.000 | 271.000 | 7.695.000 | 741.100 | - | 559.000 | 526.200 | 385.600 |
| 18 | 559.200 | 770.700 | 611.500 | 476.000 | 359.000 | 237.200 | 1.819.000 | 350.300 | - | 418.500 | 1.003.700 | - |
| 19 | 694.000 | 659.000 | - | 455.500 | 301.700 | 261.000 | 1.103.000 | 914.500 | 962.500 | 538.000 | 541.000 | - |
| 20 | 483.100 | 312.400 | - | 358.500 | 415.000 | 409.300 | 1.900.000 | 202.000 | 742.000 | 414.500 | 1.098.500 | - |
| 21 | 382.400 | 432.000 | 251.200 | 220.700 | 343.700 | 195.000 | 450.000 | 518.600 | 1.483.000 | 406.600 | 527.700 | - |
| 22 | 980.000 | 1.753.400 | 421.800 | 387.000 | 403.900 | 262.200 | 771.500 | 558.100 | 243.500 | 282.500 | 931.400 | 503.400 |
| 23 | 645.200 | 519.500 | 633.200 | 399.800 | 451.100 | 220.000 | 1.330.000 | 446.000 | 865.400 | 715.400 | 466.000 | 430.000 |
| 24 | 552.000 | 276.000 | 258.500 | 520.000 | 401.000 | 227.000 | 1.420.400 | 659.000 | 311.600 | 499.600 | 1.431.300 | 341.500 |
| 25 | 414.000 | 280.000 | 359.500 | 1.049.500 | 510.500 | 130.000 | - | 172.800 | 687.000 | 621.000 | 264.600 | 467.000 |
| 26 | 310.000 | 209.500 | 324.200 | 794.900 | 424.300 | - | 1.277.500 | 609.100 | 999.400 | 280.000 | 259.200 | 645.000 |
| 27 | 150.500 | 457.000 | 313.500 | 530.000 | 370.300 | 320.500 | 1.239.500 | 468.100 | 3.432.500 | 728.500 | 966.000 | 487.000 |
| 28 | 1.089.000 | 270.000 | 366.300 | 267.000 | 282.500 | 337.500 | 898.500 | 650.900 | 269.000 | 473.000 | 770.000 | - |
| 29 | 695.500 | - | 1.579.500 | 421.700 | 458.200 | 269.000 | 962.600 | 240.500 | 453.200 | 361.800 | 218.500 | 482.800 |
| 30 | 668.000 | - | 454.000 | 415.400 | 453.600 | 134.700 | 352.000 | 333.700 | 407.800 | 788.000 | 367.700 | 178.500 |
| 31 | 327.600 | - | 355.500 | - | 455.500 | - | 850.500 | 405.600 | - | 362.500 | - | 234.000 |
| **total** | **17.452.200** | **15.116.300** | **17.591.100** | **13.127.600** | **12.068.000** | **8.204.300** | **33.598.500** | **17.271.900** | **19.810.500** | **17.966.200** | **31.225.300** | **13.717.800** |

Dari rekapan data penjualan selama tiga tahun berturut-turut tersebut, maka dapat diketahui indeks waktunya. Indeks waktu digunakan untuk menghitung nilai perkiraan. Indek waktu setiap bulannya akan berbeda nilainya tergantung dari jumlah penjualan dan rata-rata penjualan setiap bulannya. Untuk mencari indeks waktu pertama harus mengetahui terlebih dahulu prosentase terhadap total dengan menggunakan rumus . Misalkan rata-rata penjualan Balsem Hijau Cap Betet ukuran kecil 7,5gram di bulan Januari selama tiga tahun terakhir adalah Rp. 15.257.600 dan total rata-rata penjualan Balsem Hijau Cap Betet ukuran kecil 7,5gram selama tiga tahun terakhir adalah Rp. 192.848.983, maka

Prosentase terhadap total =

= 7,9%

Dari perhitungan diatas, maka diketahui bahwa prosentase terhadap total di bulan Januari selama tiga tahun terakhir adalah 7,9%. Setelah mengetahui prosentase terhadap total maka selanjutnya dihitung indeks waktu menggunakan rumus , dimana n adalah prosentase terhadap total. Berikut perhitungan indeks waktu di bulan Januari :

Dari perhitungan diatas diketahui bahwa indeks waktu untuk bulan Januari adalah 94,9. Indeks waktu juga akan berubah setiap bulannya tergantung dari prosentase terhadap total. Berikut adalah tabel perhitungan prosentase terhadap total dan indeks waktu.

Tabel 5.4 Perhitungan Indeks Waktu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| BULAN | JUMLAH PENJUALAN | RATA-RATA PENJUALAN | PROSENTASE TERHADAP TOTAL (%) | INDEKS WAKTU |
| Januari | 45.772.800 | 15.257.600 | 7,9 % | 94,9 |
| Februari | 38.733.950 | 12.911.317 | 6,7 % | 80,3 |
| Maret | 43.847.900 | 14.615.967 | 7,6 % | 90,9 |
| April | 38.825.500 | 12.941.833 | 6,7 % | 80,5 |
| Mei | 38.669.200 | 12.889.733 | 6,7 % | 80,2 |
| Juni | 28.833.300 | 9.611.100 | 5 % | 59,8 |
| Juli | 68.389.900 | 22.796.633 | 11,8 % | 141,9 |
| Agustus | 54.046.600 | 18.015.533 | 9,3 % | 112,1 |
| September | 51.918.100 | 17.306.033 | 9 % | 107,7 |
| Oktober | 67.221.100 | 22.407.033 | 11,6 % | 139,4 |
| November | 63.195.800 | 21.065.267 | 10,9 % | 131,1 |
| Desember | 39.092.800 | 13.030.933 | 6,8 % | 81,1 |
| **Total** | **578.546.950** | **192.848.983** | **100 %** | **1.200** |

Dari hasil perhitungan prosentase terhadap total dan indeks waktu, perkiraan nilai stok minimal pada tahun 2016 dapat diketahui dengan perhitungan rumus yang sudah tertera di sub bab 2.4 tentang metode *Just In Time*. Pada sub bab 2.4 di jelaskan bahwa untuk mencari nilai perkiraan dapat menggunakan rumus . Dimana Ft adalah nilai perkiraan dan adalah perkiraan tahunan.

Sebelum mencari nilai dari Ft, kita harus mencari nilai dari terlebih dahulu, dengan rumus . Untuk mengetahui nilai kita harus mengetahui terlebih dahulu nilai a, b, dan terlebih dahulu. Berikut adalah tabel perhitungannya

Tabel 5.5 Tabel Untuk Mengetahui Nilai

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tahun | Total Penjualan | Rata-Rata Tengah (a) | Nilai Tahun 2013 | Nilai Tahun 2015 |
| 2013(K1) | 153.764.750 |  | -1 | -2 |
| 2014 | 207.632.800 | 180.698.775 | 0 | -1 |
| 2015(K2) | 217.149.400 | 212.391.100 | 1 | 0 |

Untuk mencari nilai b dapat menggunakan rumus b =

Tabel 5.6 Tabel Perhitungan Nilai Tahun 2013

|  |  |
| --- | --- |
| Nilai | Tahun 2013 |
| -1 |  |
| 0 |  |
| 1 |  |

Tabel 5.7 Tabel Perhitungan Nilai Tahun 2014

|  |  |
| --- | --- |
| Nilai | Tahun 2014 |
| -2 |  |
| -1 |  |
| 0 |  |

Tabel 5.8 Tabel Hasil dan Rata-Rata Nilai Tahun 2014 dan 2015

|  |  |  |
| --- | --- | --- |
| No | Nilai Tahun 2014 | Nilai Tahun 2015 |
| 1. | 149.006.450 | 149.006.450 |
| 2. | 180.698.775 | 180.698.775 |
| 3. | 212.391.100 | 212.391.100 |
| Rata-rata | 180.698.775 | 180.698.775 |

Setelah mengetahui nilai , maka perkiraan nilai stok minimal untuk Balsem Hijau Cap Betet ukuran kecil 7,5 gram di tahun 2016 dapat dihitung. Berikut adalah tabel perhitungannya :

Tabel 5.9 Hasil Perhitungan Perkiraan Nilai Stok Minimal Balsem Hijau Cap Betet Ukuran Kecil 7,5 gram Untuk Tahun 2015

|  |  |
| --- | --- |
| Bulan | Perkiraan Nilai Stok Minimal |
| Januari |  |
| Febuari |  |
| Maret |  |
| April |  |
| Mei |  |
| Juni |  |
| Juli |  |
| Agustus |  |
| September |  |
| Oktober |  |
| November |  |
| Desember |  |

Dari tabel diketahui jumlah nilai stok minimal Balsem Hijau Cap Betet ukuran kecil 7,5 gram maka akan diketahui banyaknya barang yang akan diambil untuk stok minimal. Berikut tabel jumlah barang yang dapat diambil untuk menghitung stok minimal di tahun 2016 :

Tabel 5.10 Jumlah Stok Minimal Balsem Hijau Cap Betet Ukuran Kecil 7,5 gr di Tahun 2016

|  |  |  |
| --- | --- | --- |
| BULAN | HASIL PERHITUNGAN PERKIRAAN NILAI STOK MINIMAL (Rp.) | JUMLAH STOK MINIMAL |
| Januari | 14.296.314 | 6.808 pcs |
| Februari | 12.097.855 | 5.761 pcs |
| Maret | 13.695.106 | 6.521 pcs |
| April | 12.126.449 | 5.774 pcs |
| Mei | 12.077.632 | 5.751 pcs |
| Juni | 9.005.565 | 4.288 pcs |
| Juli | 21.360.360 | 10.172 pcs |
| Agustus | 16.880.487 | 8.038 pcs |
| September | 16.215.688 | 7.722 pcs |
| Oktober | 20.995.306 | 9.998 pcs |
| November | 19.738.076 | 9.399 pcs |
| Desember | 12.209.936 | 5.814 pcs |
| Rata-rata | 15.058.231 | 7.171 pcs |

Dari hasil perhitungan perkiraan penjualan diatas dapat diketahui bahwa rata-rata stok minimal di tahun 2016 sebanyak 7.171 pcs untuk setiap bulannya. Dengan jumlah barang sejumlah 7.171 pcs untuk stok minimal diharap pemakaian gudang akan lebih diminimalisir, karena distributor hanya akan mengambil barang lagi jika stok yang ada digudang telah mencapai stok minimal tersebut. Dan biaya-biaya yang tidak perlu juga akan berkurang dengan otomatis dan customer akan terlayani dengan baik karena perusahaan tidak kehabisan stok barang. Dengan begitu diharapkan laba yang diperoleh akan meningkat dibandingkan dengan sebelum menggunakan metode ini.

Fungsi control dilakukan oleh admin, owner, dan bagian gudang PT Irawan Djaja Agung. Fungsi control oleh admin dapat mengakses semua fitur yang disediakan oleh aplikasi, sedangkan untuk owner dan bagian gudang hanya bisa mengakses beberapa fitur saja. Kemampuan aplikasi dalam memprediksi jumlah pengambilan stok barang masih terdapat banyak bagian yang harus diperbaiki. Aplikasi ini termasuk dalam aplikasi yang tingkat komplesitasnya sedang, namun masih butuh untuk lebih dikembangkan lagi.

# BAB 6. PENUTUP

Bab ini berisi mengenai kesimpulan dan saran dari peneliti tentang penelitian yang telah dilakukan. Kesimpulan dan saran tersebut diharapkan dapat digunakan sebagai acuan pada penelitian selanjutnya.

## 6.1 Kesimpulan

Kesimpulan dari hasil penelitian yang telah dilakukan oleh peneliti adalah sebagai berikut:

1. Dari hasil perhitungan prediksi, penerapan metode *Just In Time* dapat menentukan jumlah barang yang akan diambil diperiode selanjutnya sehingga para distributor dapat memperkirakan jumlah target yang akan dicapai di periode selanjutnya.
2. Penggunaan gudang dan biaya yang tidak diperlukan lebih bisa ditekan dengan menggunakan metode *Just In Time* karena distributor hanya akan mengambil barang lagi jika stok yang ada di gudang telah mencapai stok minimal.
3. Dengan menggunakan metode *Just In Time* diharapkan laba yang diperoleh perusahaan akan meningkat dari sebelumnya
4. Aplikasi prediksi pengambilan stok barang dibangun dan dirancang menggunakan bahasa pemrograman PHP (*Page Hypertext Pre-Processor*) dengan *Framework Code Igniter* yang terintegrasi dengan *database* menggunakan MySQL dan *tool* XAMPP. Aplikasi ini memiliki fitur utama yaitu prediksi pengambilan stok yang dapat menghitung laba dan memprediksi jumlah barang yang akan diambil sesuai data yang telah dimasukkan.
5. Metode pengembangan aplikasi dengan oleh *waterfall* memberikan manfaat bagi penulis dalam proses pembuatan aplikasi, karena dengan tahap yang terstruktur menghasilkan dokumen lengkap yang memudahkan penulis dalam memperbaiki aplikasi ketika terjadi kesalahan.

## 6.2 Saran

Pengembangan system dapat dilakukan oleh peneliti selanjutnya dengan membuat system prediksi menggunakan platform android, iOS, atau windows phone dan disarankan untuk menggunakan metode lain untuk menciptakan perbandingan antar metode.

# DAFTAR PUSTAKA

Antonio, M. (2013). *Info dan Pengertian*. Retrieved July 2016, from Pengertian Peramalan (forecasting): globallavebookx.blogspot.co.id/2013/12/penegrtian-peramalan-forecasting.html

Brown, T. M. (1991). Compatison of Just In Time and Batch Manufacturing. *The Role of Perfomance Obstancles. Academy of Management Journal*, 34, 906-917.

Darmawan, G. (2015). *Model Waterfall*. Dipetik April 20, 2015, dari http://www.academia.edu/7585995/BAB\_I\_Model\_Waterfall: www.academia.edu

Efrianti, D. (2014). Pengaruh Pengendlian Persediaan Just In Time Terhadap Efisiensi Pengadaan Persediaan Bahan Baku. *Jurnal Program Studi Akuntansi, Sekolah Tinggi Ilmu Ekonomi Kesatuan*, 103.

Gaspersz. (2005).

Hadioetomo. (2009). Pengaruh Penerapan JIT Terhadap Kinerja dan Keunggulan Kompetitif Perusahaan Manufaktur. *Kajian Akuntansi, Volume 4, Nomor 2*, 105-106.

Handoko. (1999).

Ipulhe. (2014, 6 7). *Pengertian Aplikasi*. Dipetik 1 15, 2016, dari http://www.ipulhe.com/pengertian-aplikasi/: http://www.ipulhe.com

Morales, R. d. (2001).

Rahayu, S. M. (2003). Pengaruh Aplikasi Strategi Just In Time Terhadap Efektivitas dan Efisiensi Biaya Produksi Pada PT. Santosa Jaya Abadi Sidoarjo. *Jurnal Sekolah Tinggi Ilmu Ekonomi Indonesia (STIESIA) Surabaya*, 453.

Setyawan, D. (2012, agustus). *Definisi Aplikasi*. Dipetik Januari 15, 2016, dari http://definisimu.blogspot.co.id/2012/08/definisi-aplikasi.html: http://definisimu.blogspot.co.id

Vuppalapti, K. (1995). JIT and TQM : A Case for Joint Implementation. *International Journal of Operation and Management*, 15, 553-564.

# LAMPIRAN

## LAMPIRAN A. PERANCANGAN APLIKASI

### A1. Perancangan *Scenario*

1. *Scenario* Mengelola Data Karyawan
2. Tambah Data Karyawan

|  |  |  |
| --- | --- | --- |
| ***Scenario* Tambah Data Karyawan** | | |
|  | |  |
| **Nama *Usecase*** | | Mengelola Data Karyawan |
| **Aktor** | | Admin |
| **Pra-Kondisi** | | Admin telah berhasil masuk ke halaman dashboard |
| **Pasca-Kondisi** | | Admin berhasil menambahkan data karyawan |
| **Aktor** | **Aplikasi** | |
| **Skenario Utama**  **“Tambah Data Karyawan”** | | |
| 1. Klik menu “Data Master” 2. Pilih dan klik submenu “Karyawan” |  | |
|  | 1. Menampilkanhalaman utama data karyawan | |
| 1. Klik tombol “Tambah Karyawan” |  | |
|  | 1. Menampilkan form tambah data karyawan | |
| 1. Mengisi form tambah data karyawan 2. Klik “Submit” |  | |
|  | 1. Menyimpan data karyawan ke*database* 2. Menampilkan halaman utama list data karyawan | |
| **Skenario Alternatif**  **“*Form* Kosong”** | | |
| 7a. Klik tombol “Submit” |  | |
|  | 8a. Menampilkan pesar *error* pada *field* | |
| **Skenario Alternatif**  **“Klik Tombol Kembali”** | | |
| 7b. Klik “Back” |  | |
|  | 8b. Menampilkan halaman utama list data karyawan | |

1. Ubah Data Karyawan

|  |  |  |  |
| --- | --- | --- | --- |
| ***Scenario* Ubah Data Karyawan** | | | |
|  | | |  |
| **Nama *Usecase*** | | | Mengelola Data Karyawan |
| **Aktor** | | | Admin |
| **Pra-Kondisi** | | | Admin telah berhasil masuk ke halaman data karyawan |
| **Pasca-Kondisi** | | | Admin berhasil mengubah data karyawan |
| **Aktor** | | **Aplikasi** | |
| **Skenario Utama**  **“Ubah Data Karyawan”** | | | |
| 1. Klik icon “ubah” pada kolom yang datanya ingin dirubah | |  | |
|  | | 1. Menampilkan form ubah dengan data sebelumnya | |
| 1. Mengisi data yang ingin diubah 2. Klik “Submit” | |  | |
|  | | 1. Menyimpan data karyawan ke*database* 2. Menampilkan halaman utama list data karyawan | |
| **Skenario Alternatif**  **“Pilih dan Klik Drop Down Aktif/Non aktif”** | | | |
| 3a. Pilih dan klik non aktif  4a. Klik tombol “Submit” |  | | |
|  | 5a. Memperbarui data karyawan ke *database*.  6a. Menampilkan halaman utama list data karyawan | | |
| **Skenario Alternatif**  **“*Form* Kosong”** | | | |
| 4b. Klik tombol “Submit” | |  | |
|  | | 5b. Menampilkan pesar *error* pada*field* | |
| **Skenario Alternatif**  **“Klik Tombol Kembali”** | | | |
| 4c. Klik “Back” | |  | |
|  | | 5c. Menampilkan halaman utama list data karyawan | |

1. Mengelola Data Customer
2. Tambah Data Customer

|  |  |  |
| --- | --- | --- |
| ***Scenario* Tambah Data Customer** | | |
|  | |  |
| **Nama *Usecase*** | | Mengelola Data Customer |
| **Aktor** | | Admin |
| **Pra-Kondisi** | | Admin telah berhasil masuk ke halaman dashboard |
| **Pasca-Kondisi** | | Admin berhasil menambahkan data customer |
| **Aktor** | **Aplikasi** | |
| **Skenario Utama**  **“Tambah Data Customer”** | | |
| 1. Klik menu “Data Master” 2. Pilih dan klik submenu “Customer” |  | |
|  | 1. Menampilkanhalaman utama data customer | |
| 1. Klik tombol “Tambah Customer” |  | |
|  | 1. Menampilkan form tambah data customer | |
| 1. Mengisi form tambah data customer 2. Klik “Submit” |  | |
|  | 1. Menyimpan data customer ke*database* 2. Menampilkan halaman utama list data customer | |
| **Skenario Alternatif**  **“*Form* Kosong”** | | |
| 7a. Klik tombol “Submit” |  | |
|  | 8a. Menampilkan pesar *error* pada *field* | |
| **Skenario Alternatif**  **“Klik Tombol Kembali”** | | |
| 7b. Klik “Back” |  | |
|  | 8b. Menampilkan halaman utama list data customer | |

1. Ubah Data Customer

|  |  |  |  |
| --- | --- | --- | --- |
| ***Scenario* Ubah Data Customer** | | | |
|  | | |  |
| **Nama *Usecase*** | | | Mengelola Data Customer |
| **Aktor** | | | Admin |
| **Pra-Kondisi** | | | Admin telah berhasil masuk ke halaman data customer |
| **Pasca-Kondisi** | | | Admin berhasil mengubah data customer |
| **Aktor** | | **Aplikasi** | |
| **Skenario Utama**  **“Ubah Data Customer”** | | | |
| 1. Klik icon “ubah” pada kolom yang datanya ingin dirubah | |  | |
|  | | 1. Menampilkan form ubah dengan data sebelumnya | |
| 1. Mengisi data yang ingin diubah 2. Klik “Submit” | |  | |
|  | | 1. Menyimpan data customer ke*database* 2. Menampilkan halaman utama list data customer | |
| **Skenario Alternatif**  **“Pilih dan Klik Drop Down Aktif/Non aktif”** | | | |
| 3a. Pilih dan klik non aktif  4a. Klik tombol “Submit” |  | | |
|  | 5a. Memperbarui data customer ke *database*.  6a. Menampilkan halaman utama list data customer | | |
| **Skenario Alternatif**  **“*Form* Kosong”** | | | |
| 4b. Klik tombol “Submit” | |  | |
|  | | 5b. Menampilkan pesar *error* pada *field* | |
| **Skenario Alternatif**  **“Klik Tombol Kembali”** | | | |
| 4c. Klik “Back” | |  | |
|  | | 5c. Menampilkan halaman utama list data customer | |

1. Mengelola Data Barang
2. Tambah Data Barang

|  |  |  |
| --- | --- | --- |
| ***Scenario* Tambah Data Barang** | | |
|  | |  |
| **Nama *Usecase*** | | Mengelola Data Barang |
| **Aktor** | | Admin |
| **Pra-Kondisi** | | Admin telah berhasil masuk ke halaman dashboard |
| **Pasca-Kondisi** | | Admin berhasil menambahkan data barang |
| **Aktor** | **Aplikasi** | |
| **Skenario Utama**  **“Tambah Data Barang”** | | |
| 1. Klik menu “Data Barang” |  | |
|  | 1. Menampilkan halaman utama data barang | |
| 1. Klik tombol “Tambah Barang” |  | |
|  | 1. Menampilkan form tambah data barang | |
| 1. Mengisi form data tambah barang 2. Klik “Submit” |  | |
|  | 1. Menyimpan data barang ke*database* 2. Menampilkan halaman utama list data barang | |
| **Skenario Alternatif**  **“*Form* Kosong”** | | |
| 7a. Klik tombol “Submit” |  | |
|  | 8a. Menampilkan pesar *error* pada *field* | |
| **Skenario Alternatif**  **“Klik Tombol Kembali”** | | |
| 7b. Klik “Back” |  | |
|  | 8b. Menampilkan halaman utama list data barang | |

1. Ubah Data Barang

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Scenario* Ubah Data Barang** | | | | |
|  | | |  | |
| **Nama *Usecase*** | | | Mengelola Data Barang | |
| **Aktor** | | | Admin | |
| **Pra-Kondisi** | | | Admin telah berhasil masuk ke halaman “Data Barang” | |
| **Pasca-Kondisi** | | | Admin berhasil mengubah data barang | |
| **Aktor** | | **Aplikasi** | | |
| **Skenario Utama**  **“Ubah Data Barang”** | | | | |
| 1. Pilih dan klik icon “ubah” pada kolom yang datanya ingin di ubah | |  | | |
|  | | 1. Menampilkan form ubah dengan data sebelumnya | | |
| 1. Mengisi form data yang ingin diubah 2. Klik “Submit” | |  | | |
|  | | 1. Menyimpan data barang ke*database* 2. Menampilkan halaman utama list data barang | | |
| **Skenario Alternatif**  **“Pilih dan Klik Drop Down Aktif/Non aktif”** | | | |
| 3a. Pilih dan klik non aktif  4a. Klik tombol “Submit” |  | | |
|  | 5a. Memperbarui data barang ke *database*.  6a. Menampilkan halaman utama list data barang | | |
| **Skenario Alternatif**  **“*Form* Kosong”** | | | | |
| 4a. Klik tombol “Submit” | |  | | |
|  | | 5a. Menampilkan pesar *error* pada *field* | | |
| **Skenario Alternatif**  **“Klik Tombol Kembali”** | | | | |
| 4b. Klik “Back” | |  | | |
|  | | 5b. Menampilkan halaman utama list data barang | | |

1. Menambah Data Barang Keluar

|  |  |  |
| --- | --- | --- |
| ***Scenario* Tambah Data Barang Keluar** | | |
|  | |  |
| **Nama *Usecase*** | | Menambah Data Barang Keluar |
| **Aktor** | | Admin |
| **Pra-Kondisi** | | Admin telah berhasil masuk ke halaman dashboard |
| **Pasca-Kondisi** | | Admin berhasil menambah data barang keluar |
| **Aktor** | **Aplikasi** | |
| **Skenario Utama**  **“Tambah Data Barang Keluar”** | | |
| 1. Klik menu “Barang Keluar” 2. Pilih dan klik submenu “Input Barang Keluar” |  | |
|  | 1. Menampilkan form data barang keluar | |
| 1. Mengisi form data barang keluar 2. Klik tombol “Add” |  | |
|  | 1. Menampilkan data barang pada tabel data pembelian | |
| 1. Mengisi data customer dan keterangan 2. Klik tombol “Submit” |  | |
|  | 1. Menampilkan pop up pada field | |
| 1. Klik tombol “Ok” |  | |
|  | 1. Menyimpan data barang keluar ke *database* 2. Menampilkan halaman utama “Data Barang Keluar” dan list nota barang keluar | |
| **Skenario Alternatif**  **“*Form* Kosong”** | | |
| 5a. Klik tombol “Add” |  | |
|  | 6a. Menampilkan pesan *error* pada *field* | |
| 7a. Klik tombol “Submit” |  | |
|  | 8a. Menampilkan pesan *error* pada *field* | |
| **Skenario Alternatif**  **“Klik Icon Hapus”** | | |
| 7b. Klik icon “hapus” pada data barang yang ingin dihapus | 8b. Menghapus data barang keluar | |
| **Skenario Alternatif**  **“Klik Tombol Batal”** | | |
| 10c. Klik tombol “Cancel” |  | |
|  | 11c. Menampilkan data barang pada tabel data pembelian | |

1. Melihat Data Barang Keluar
2. Melihat Data Barang Keluar – Admin

|  |  |  |
| --- | --- | --- |
| ***Scenario* Melihat Data Barang Keluar** | | |
|  | |  |
| **Nama *Usecase*** | | Melihat Data Barang Keluar |
| **Aktor** | | Admin |
| **Pra-Kondisi** | | Admin telah berhasil masuk ke halaman dashboard |
| **Pasca-Kondisi** | | Admin berhasil melihat data barang keluar |
| **Aktor** | **Aplikasi** | |
| **Skenario Utama**  **“Melihat Data Barang Keluar”** | | |
| 1. Klik menu “Barang Keluar” 2. Pilih dan klik submenu “Data Barang Keluar” |  | |
|  | 1. Menampilkan halaman utama dan list nota data barang keluar | |
| 1. Klik icon “lihat” pada kolom nota yang ingin dilihat |  | |
|  | 1. Menampilkan rincian data barang keluar | |

1. Melihat Data Barang Keluar – Owner

|  |  |  |
| --- | --- | --- |
| ***Scenario* Melihat Data Barang Keluar** | | |
|  | |  |
| **Nama *Usecase*** | | Melihat Data Barang Keluar |
| **Aktor** | | Owner |
| **Pra-Kondisi** | | Owner telah berhasil masuk ke halaman dashboard |
| **Pasca-Kondisi** | | Owner berhasil melihat data barang keluar |
| **Aktor** | **Aplikasi** | |
| **Skenario Utama**  **“Melihat Data Barang Keluar”** | | |
| 1. Klik menu “Barang Keluar” 2. Pilih dan klik submenu “Data Barang Keluar” |  | |
|  | 1. Menampilkan halaman utama data barang keluar | |
| 1. Klik icon “lihat” pada kolom nota yang ingin dilihat |  | |
|  | 1. Menampilkan rincian data barang keluar | |

1. Melihat Data Barang Keluar - Bagian Gudang

|  |  |  |
| --- | --- | --- |
| ***Scenario* Melihat Data Barang Keluar** | | |
|  | |  |
| **Nama *Usecase*** | | Melihat Data Barang Keluar |
| **Aktor** | | Bagian Gudang |
| **Pra-Kondisi** | | Bagian Gudang telah berhasil masuk ke halaman dashboard |
| **Pasca-Kondisi** | | Bagian Gudang berhasil melihat data barang keluar |
| **Aktor** | **Aplikasi** | |
| **Skenario Utama**  **“Melihat Data Barang Keluar”** | | |
| 1. Klik menu “Barang Keluar” 2. Pilih dan klik submenu “Data Barang Keluar” |  | |
|  | 1. Menampilkan halaman utama data barang keluar | |
| 1. Klik icon “lihat” pada kolom nota yang ingin dilihat |  | |
|  | 1. Menampilkan rincian data barang keluar | |

1. Menambah Data Barang Masuk

|  |  |  |
| --- | --- | --- |
| ***Scenario* Tambah Data Barang Masuk** | | |
|  | |  |
| **Nama *Usecase*** | | Menambah Data Barang Masuk |
| **Aktor** | | Admin |
| **Pra-Kondisi** | | Admin telah berhasil masuk ke halaman dashboard |
| **Pasca-Kondisi** | | Admin berhasil menambah data barang masuk |
| **Aktor** | **Aplikasi** | |
| **Skenario Utama**  **“Tambah Data Barang Masuk”** | | |
| 1. Klik menu “Barang Masuk” 2. Pilih dan klik submenu “Input Barang Masuk” |  | |
|  | 1. Menampilkan form data barang masuk | |
| 1. Mengisi form data barang keluar 2. Klik tombol “Add” |  | |
|  | 1. Menampilkan data barang pada field data pembelian | |
| 1. Mengisi data karyawan dan keterangan 2. Klik tombol “Submit” |  | |
|  | 1. Menampilkan pop up pada field | |
| 1. Klik tombol “Ok” |  | |
|  | 1. Menyimpan data barang keluar ke *database* 2. Menampilkan halaman utama “Data Barang Masuk” dan list nota barang masuk | |
| **Skenario Alternatif**  **“*Form* Kosong”** | | |
| 5a. Klik tombol “Add” |  | |
|  | 6a. Menampilkan pesan *error* pada *field* | |
| 7a. Klik tombol “Submit” |  | |
|  | 8a. Menampilkan pesan *error* pada *field* | |
| **Skenario Alternatif**  **“Klik Icon Hapus”** | | |
| 7b. Klik icon “hapus” pada data barang yang ingin dihapus | 8b. Menghapus data barang masuk | |
| **Skenario Alternatif**  **“Klik Tombol Batal”** | | |
| 10b. Klik tombol “Cancel” |  | |
|  | 11b. Menampilkan data barang pada field data pembelian | |

1. Melihat Data Barang Masuk
2. Melihat Data Barang Masuk – Admin

|  |  |  |
| --- | --- | --- |
| ***Scenario* Melihat Data Barang Masuk** | | |
|  | |  |
| **Nama *Usecase*** | | Melihat Data Barang Masuk |
| **Aktor** | | Admin |
| **Pra-Kondisi** | | Admin telah berhasil masuk ke halaman dashboard |
| **Pasca-Kondisi** | | Admin berhasil melihat data barang masuk |
| **Aktor** | **Aplikasi** | |
| **Skenario Utama**  **“Melihat Data Barang Masuk”** | | |
| 1. Klik menu “Barang Masuk” 2. Pilih dan klik submenu “Data Barang Masuk” |  | |
|  | 1. Menampilkan halaman utama data barang masuk | |
| 1. Klik icon “lihat” pada kolom data yang ingin dilihat |  | |
|  | 1. Menampilkan rincian data barang masuk | |

1. Melihat Data Barang Masuk – owner

|  |  |  |
| --- | --- | --- |
| ***Scenario* Melihat Data Barang Masuk** | | |
|  | |  |
| **Nama *Usecase*** | | Melihat Data Barang Masuk |
| **Aktor** | | Owner |
| **Pra-Kondisi** | | Owner telah berhasil masuk ke halaman dashboard |
| **Pasca-Kondisi** | | Owner berhasil melihat data barang masuk |
| **Aktor** | **Aplikasi** | |
| **Skenario Utama**  **“Melihat Data Barang Masuk”** | | |
| 1. Klik menu “Barang Masuk” 2. Pilih dan klik submenu “Data Barang Masuk” |  | |
|  | 1. Menampilkan halaman utama data barang masuk | |
| 1. Klik icon “lihat” pada kolom nota yang ingin dilihat |  | |
|  | 1. Menampilkan rincian data barang masuk | |

1. Menambah Data Barang Retur

|  |  |  |
| --- | --- | --- |
| ***Scenario* Tambah Data Barang Retur** | | |
|  | |  |
| **Nama *Usecase*** | | Menambah Data Barang Retur |
| **Aktor** | | Admin |
| **Pra-Kondisi** | | Admin telah berhasil masuk ke halaman dashnoard |
| **Pasca-Kondisi** | | Admin berhasil menambah data barang retur |
| **Aktor** | **Aplikasi** | |
| **Skenario Utama**  **“Tambah Data Barang Retur”** | | |
| 1. Klik menu “Retur Barang” 2. Pilih dan klik submenu “Input Retur” |  | |
|  | 1. Menampilkan form data barang retur | |
| 1. Mengisi form data barang retur 2. Klik tombol “Add” |  | |
|  | 1. Menampilkan data barang retur pada tabel list retur | |
| 1. Mengisi data cutomer dan keterangan 2. Klik tombol “Submit” |  | |
|  | 1. Menampilkan pop up pada field | |
| 1. Klik tombol “Ok” |  | |
|  | 1. Menyimpan data barang retur ke *database* 2. Menampilkan halaman utama data barang retur dan list nota barang retur | |
| **Skenario Alternatif**  **“*Form* Kosong”** | | |
| 5a. Klik tombol “Add” |  | |
|  | 6a. Menampilkan pesan *error* pada *field* | |
| 7a. Klik tombol “Submit” |  | |
|  | 8a. Menampilkan pesan *error* pada *field* | |
| **Skenario Alternatif**  **“Klik Icon Hapus”** | | |
| 7b. Klik icon “hapus” pada data barang yang ingin dihapus | 8b. Menghapus data barang retur | |
| **Skenario Alternatif**  **“Klik Tombol Batal”** | | |
| 10b. Klik tombol “Cancel” |  | |
|  | 11b. Menampilkan data barang retur pada tabel list retur | |

1. Melihat Data Barang Retur
2. Melihat Data Barang Retur – Admin

|  |  |  |
| --- | --- | --- |
| ***Scenario* Melihat Data Barang Retur** | | |
|  | |  |
| **Nama *Usecase*** | | Melihat Data Barang Retur |
| **Aktor** | | Admin |
| **Pra-Kondisi** | | Admin telah berhasil masuk ke halaman dashnoard |
| **Pasca-Kondisi** | | Admin berhasil melihat data barang retur |
| **Aktor** | **Aplikasi** | |
| **Skenario Utama**  **“Melihat Data Barang Retur”** | | |
| 1. Klik menu “Retur Barang” 2. Pilih dan klik submenu “Data Barang Retur” |  | |
|  | 1. Menampilkan halaman utama dan list nota data barang retur | |
| 1. Klik icon “lihat” pada kolom data yang diinginkan |  | |
|  | 1. Menampilkan detail data barang retur | |

1. Melihat Data Barang Retur – Owner

|  |  |  |
| --- | --- | --- |
| ***Scenario* Melihat Data Barang Retur** | | |
|  | |  |
| **Nama *Usecase*** | | Melihat Data Barang Retur |
| **Aktor** | | Owner |
| **Pra-Kondisi** | | Owner telah berhasil masuk ke halaman dashnoard |
| **Pasca-Kondisi** | | Owner berhasil melihat data barang retur |
| **Aktor** | **Aplikasi** | |
| **Skenario Utama**  **“Melihat Data Barang Retur”** | | |
| 1. Klik menu “Retur Barang” 2. Pilih dan klik submenu “Data Barang Retur” |  | |
|  | 1. Menampilkan halaman utama data barang retur | |

1. Melihat Data Barang Retur – Bagian Gudang

|  |  |  |
| --- | --- | --- |
| ***Scenario* Melihat Data Barang Retur** | | |
|  | |  |
| **Nama *Usecase*** | | Melihat Data Barang Retur |
| **Aktor** | | Bagian Gudang |
| **Pra-Kondisi** | | Bagian Gudang telah berhasil masuk ke halaman dashnoard |
| **Pasca-Kondisi** | | Bagian Gudang berhasil melihat data barang retur |
| **Aktor** | **Aplikasi** | |
| **Skenario Utama**  **“Melihat Data Barang Retur”** | | |
| 1. Klik menu “Retur Barang” 2. Pilih dan klik submenu “Data Barang Retur” |  | |
|  | 1. Menampilkan halaman utama data barang retur | |

1. Melihat Kondisi Stok Barang
2. Melihat Kondisi Stok Barang – Admin

|  |  |  |
| --- | --- | --- |
| ***Scenario* Melihat Stok Barang Terakhir** | | |
|  | |  |
| **Nama *Usecase*** | | Melihat Data Barang Retur |
| **Aktor** | | Admin |
| **Pra-Kondisi** | | Admin Berhasil masuk ke dashboard |
| **Pasca-Kondisi** | | Admin berhasil melihat data barang terakhir |
| **Aktor** | **Aplikasi** | |
| **Skenario Utama**  **“Melihat Data Barang Terakhir”** | | |
| 1. Klik menu “Retur Barang” 2. Pilih dan klik submenu “Kondisi Stok Barang” |  | |
|  | 1. Menampilkan halaman utama kondisi data barang | |

1. Melihat Kondisi Stok Barang – Owner

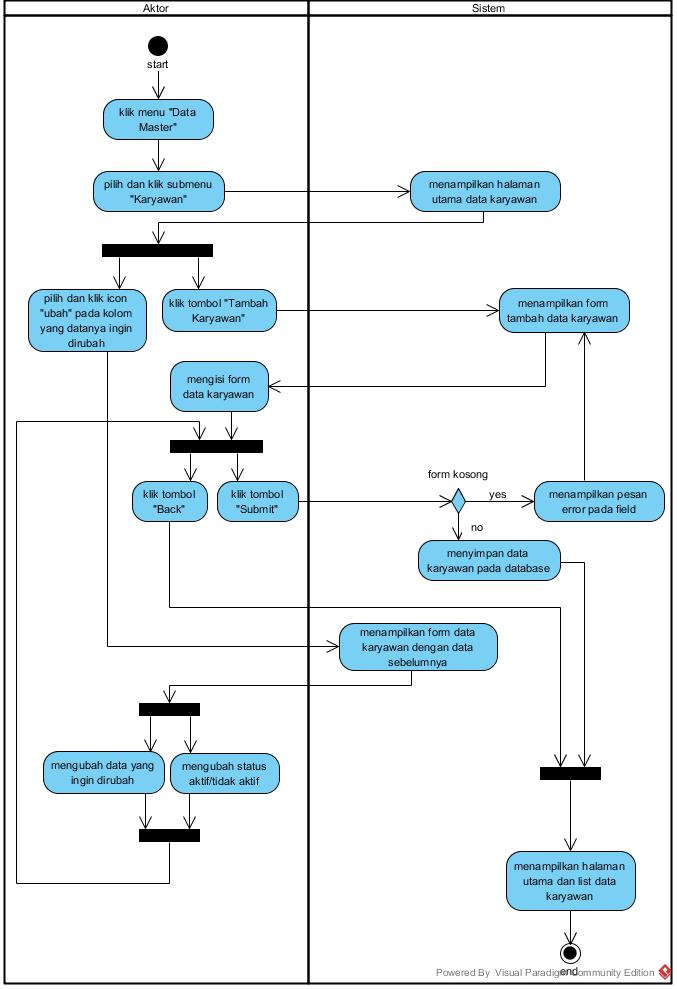
|  |  |  |
| --- | --- | --- |
| ***Scenario* Melihat Stok Barang Terakhir** | | |
|  | |  |
| **Nama *Usecase*** | | Melihat Data Barang Retur |
| **Aktor** | | Owner |
| **Pra-Kondisi** | | Owner Berhasil masuk ke dashboard |
| **Pasca-Kondisi** | | Owner berhasil melihat data barang terakhir |
| **Aktor** | **Aplikasi** | |
| **Skenario Utama**  **“Melihat Data Barang Terakhir”** | | |
| 1. Klik menu “Retur Barang” 2. Pilih dan klik submenu “Kondisi Stok Barang” |  | |
|  | 1. Menampilkan halaman utama kondisi data barang | |

1. Melihat Kondisi Stok Barang – Bagian Gudang

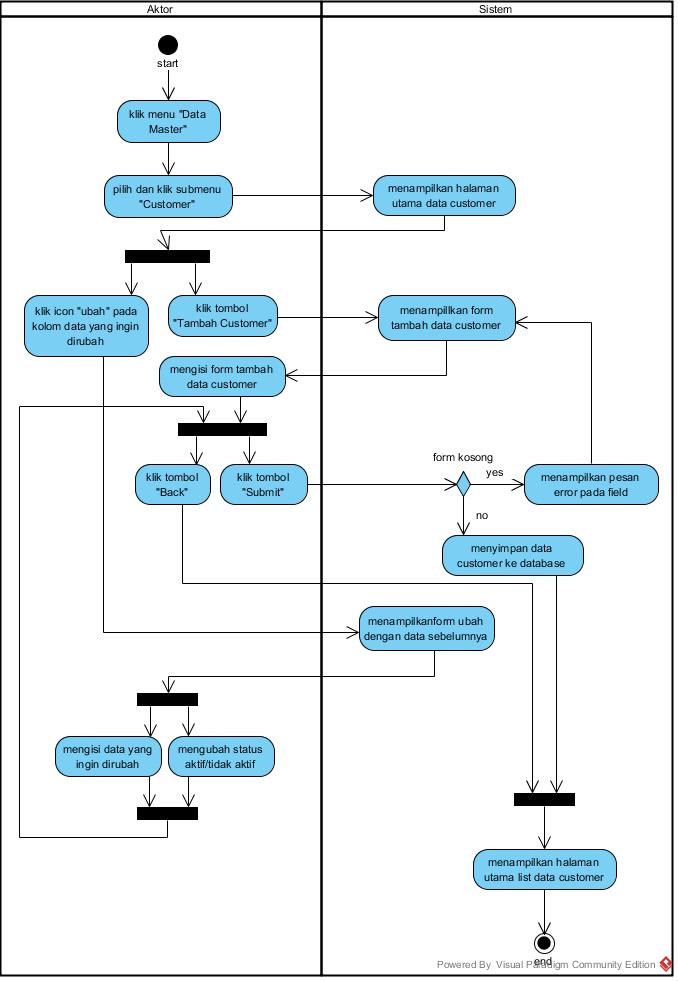
|  |  |  |
| --- | --- | --- |
| ***Scenario* Melihat Stok Barang Terakhir** | | |
|  | |  |
| **Nama *Usecase*** | | Melihat Data Barang Retur |
| **Aktor** | | Bagian Gudang |
| **Pra-Kondisi** | | Bagian Gudang Berhasil masuk ke dashboard |
| **Pasca-Kondisi** | | Bagian Gudang berhasil melihat data barang terakhir |
| **Aktor** | **Aplikasi** | |
| **Skenario Utama**  **“Melihat Data Barang Terakhir”** | | |
| 1. Klik menu “Retur Barang” 2. Pilih dan klik submenu “Kondisi Stok Barang” |  | |
|  | 1. Menampilkan halaman utama kondisi data barang | |

### A2. Perancangan *Activity Diagram*

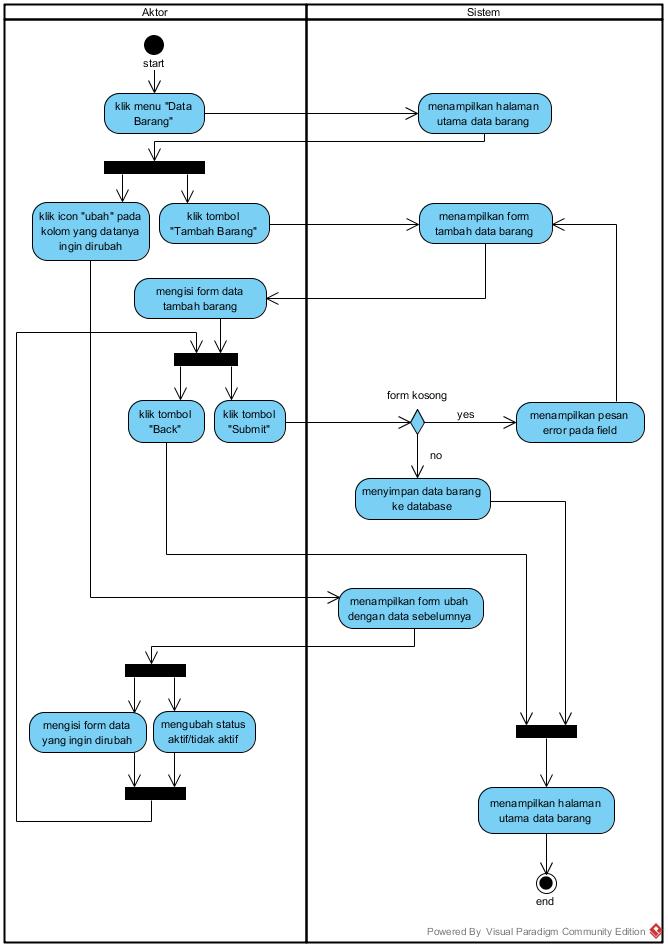
1. *Activity Diagram* Mengelola Data Karyawan



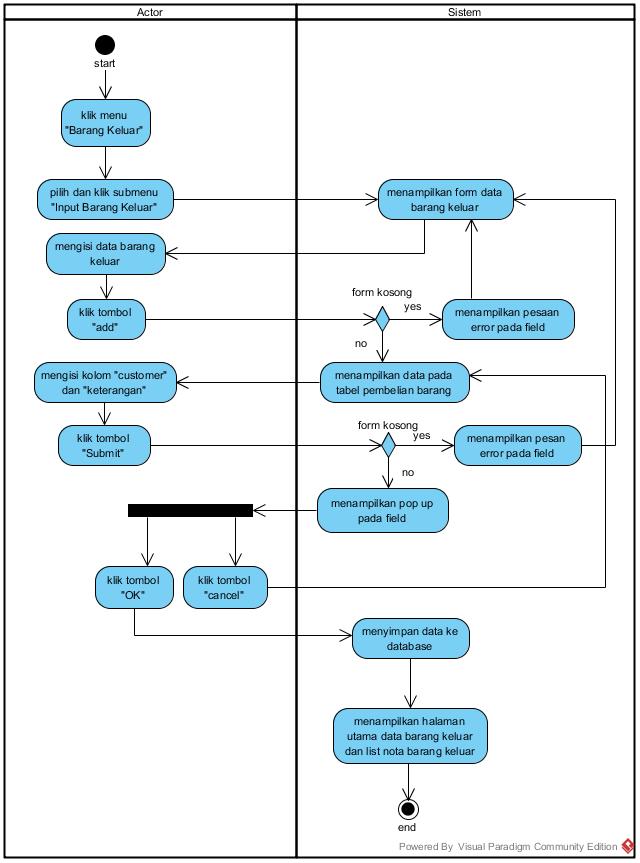
1. *Activity Diagram* Mengelola Data Customer



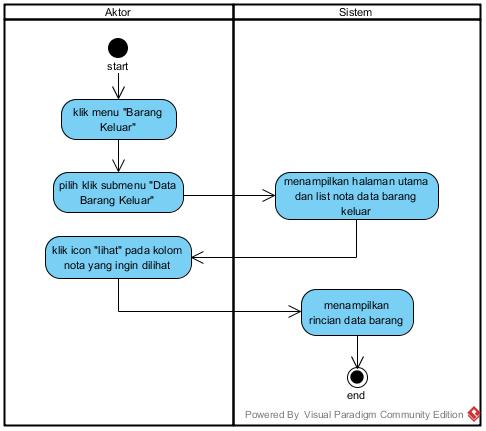
1. *Activity Diagram* Mengelola Data Barang



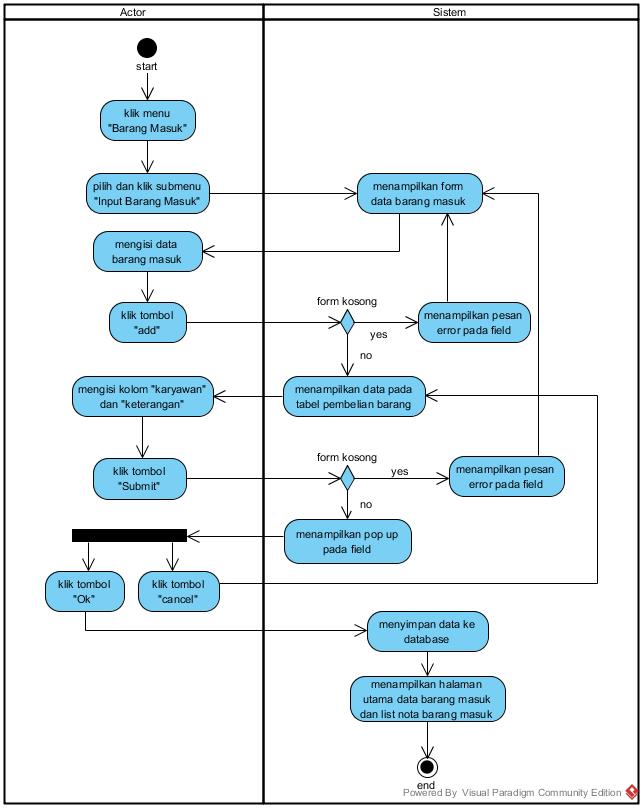
1. *Activity Diagram* Menambah Data Barang Keluar



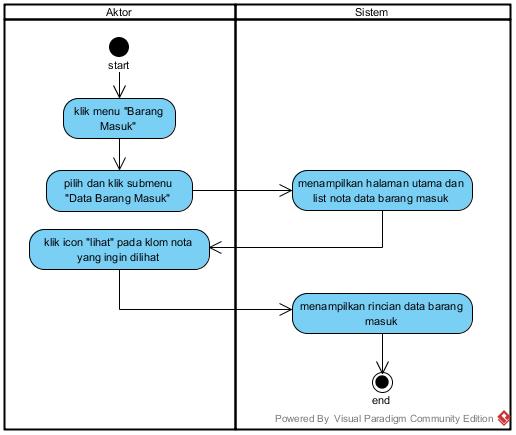
1. *Activity Diagram* Melihat Data Barang Keluar



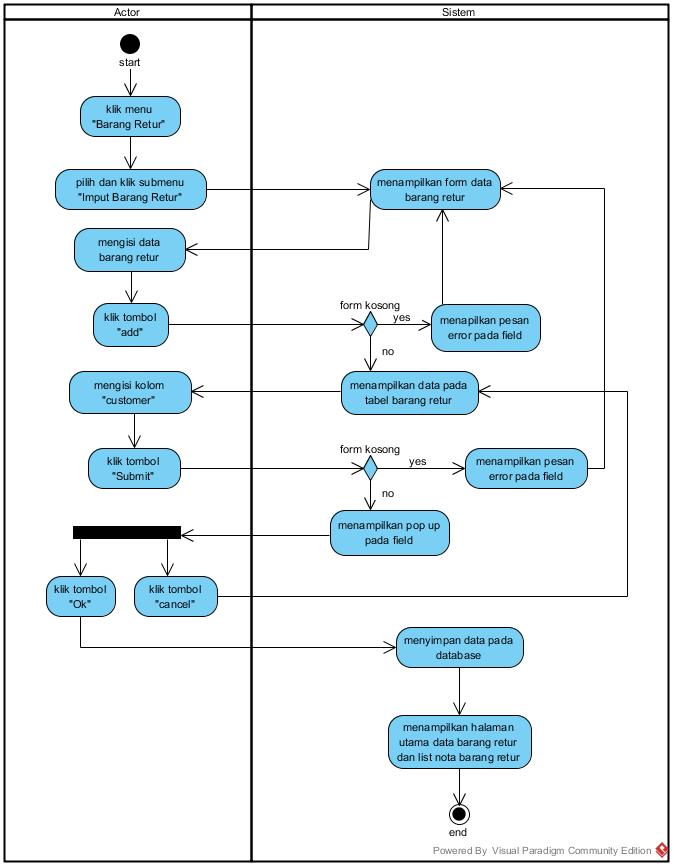
1. *Activity Diagram* Menambah Data Barang Masuk



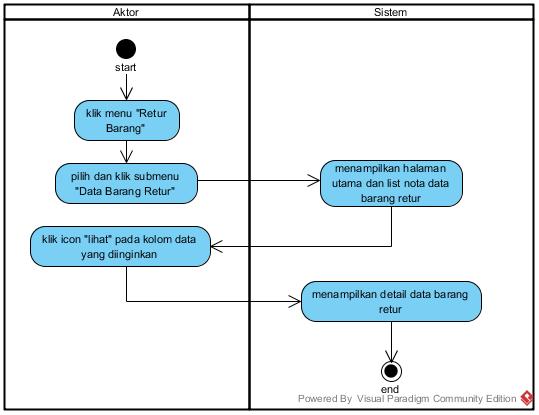
1. *Activity Diagram* Melihat Data Barang Masuk



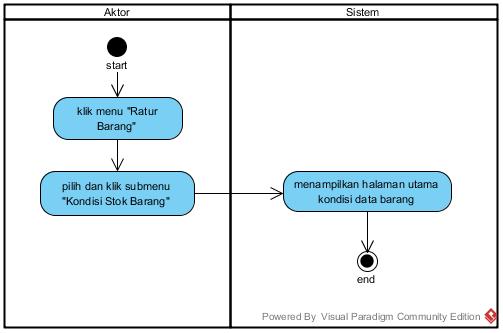
1. *Activity Diagram* Menambah Data Barang Retur



1. *Activity Diagram* Melihat Data Barang Masuk

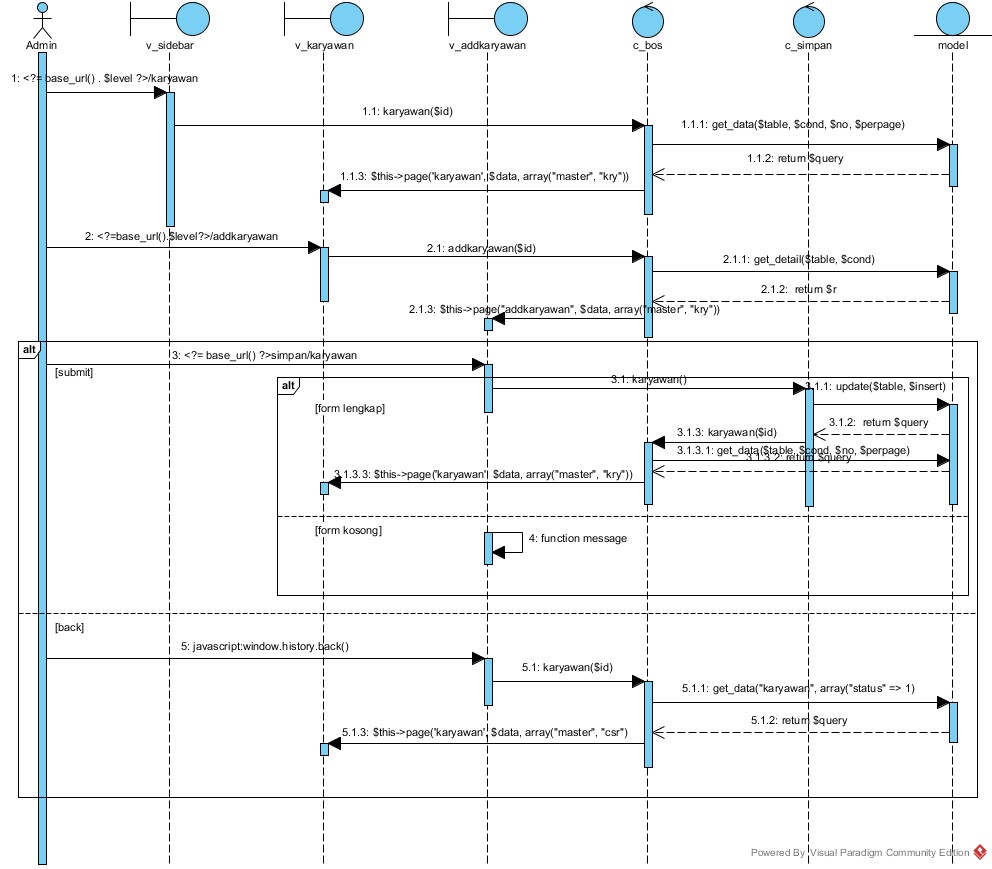


1. *Activity Diagram* Melihat Kondisi Stok

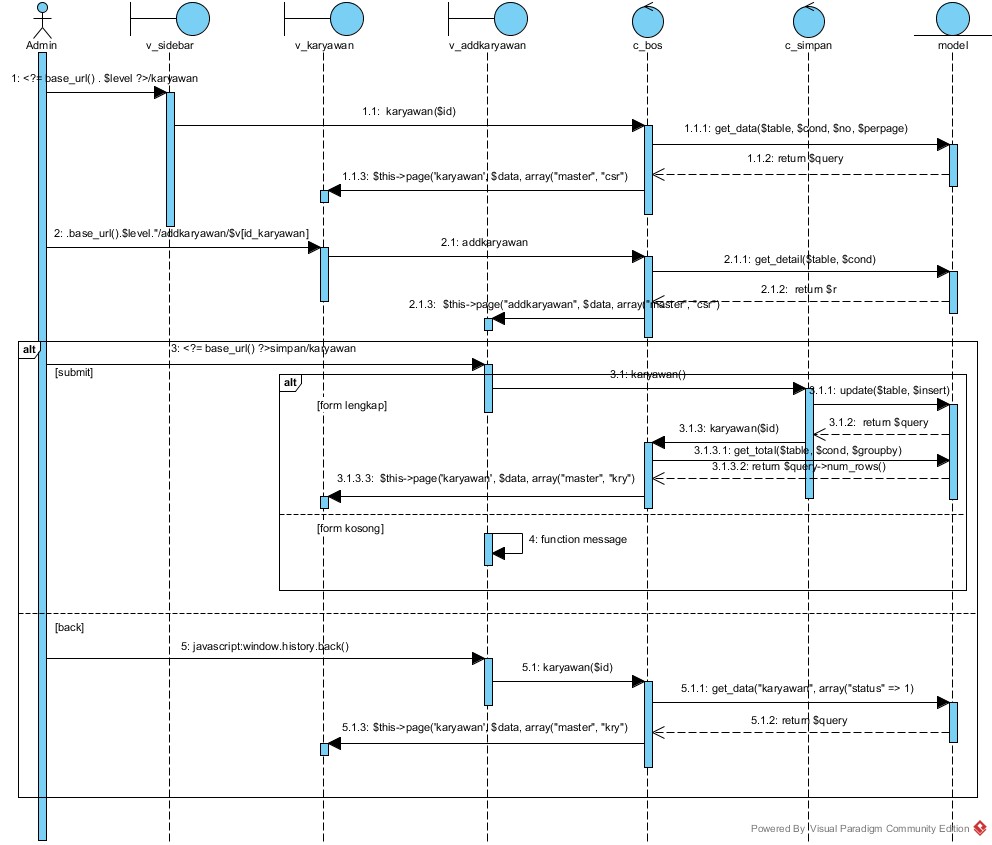


### A3. Perancangan *Squence Diagram*

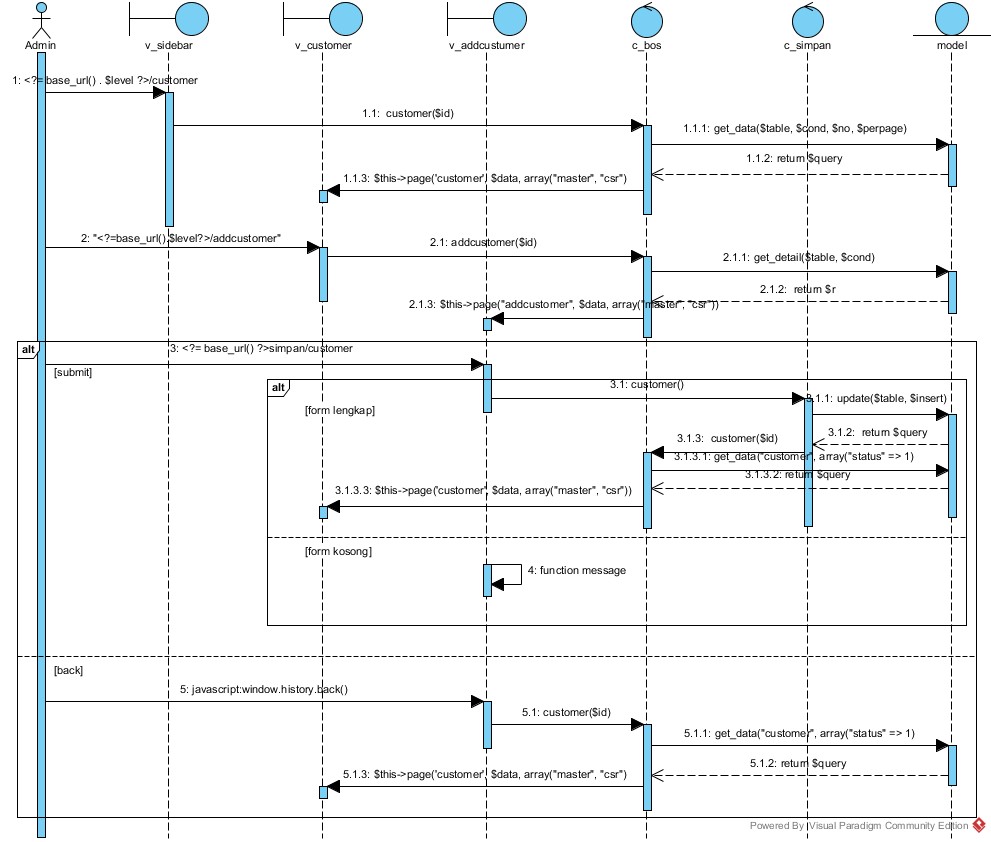
1. *Squence Diagram* Mengelola Data Karyawan
2. *Squence Diagram* Tambah Data Karyawan



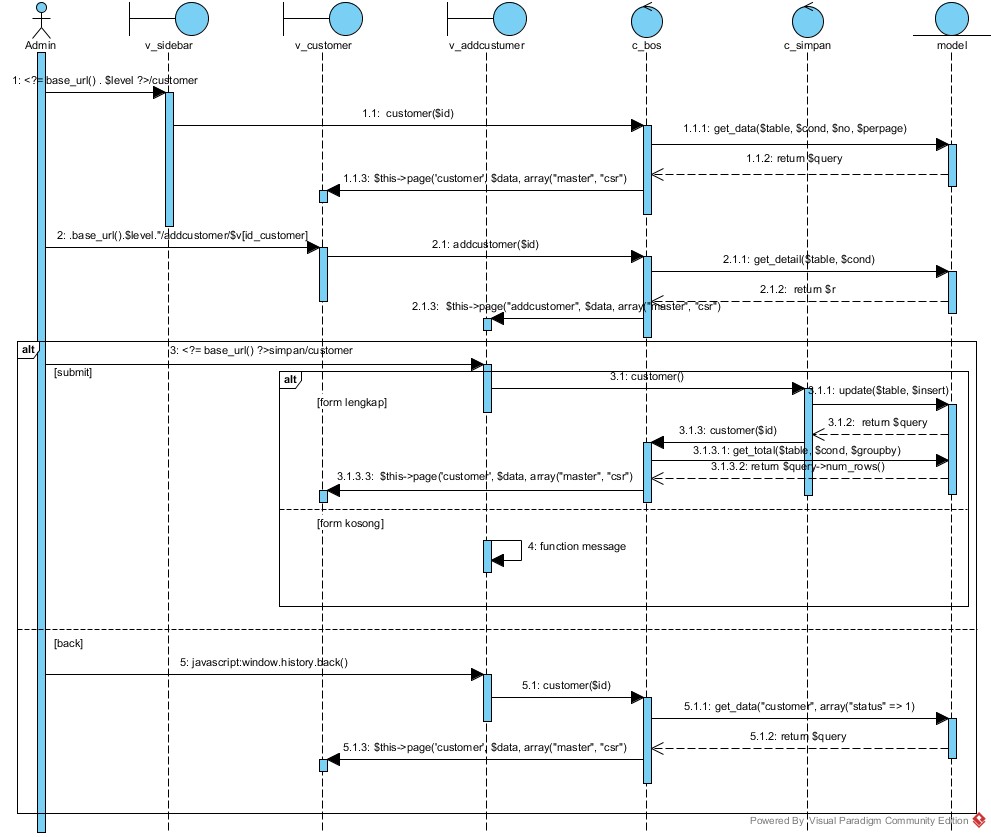
1. *Squence Diagram* Ubah Data Karyawan



1. *Squence Diagram* Mengelola Data Customer
2. *Squence Diagram* Tambah Data Customer



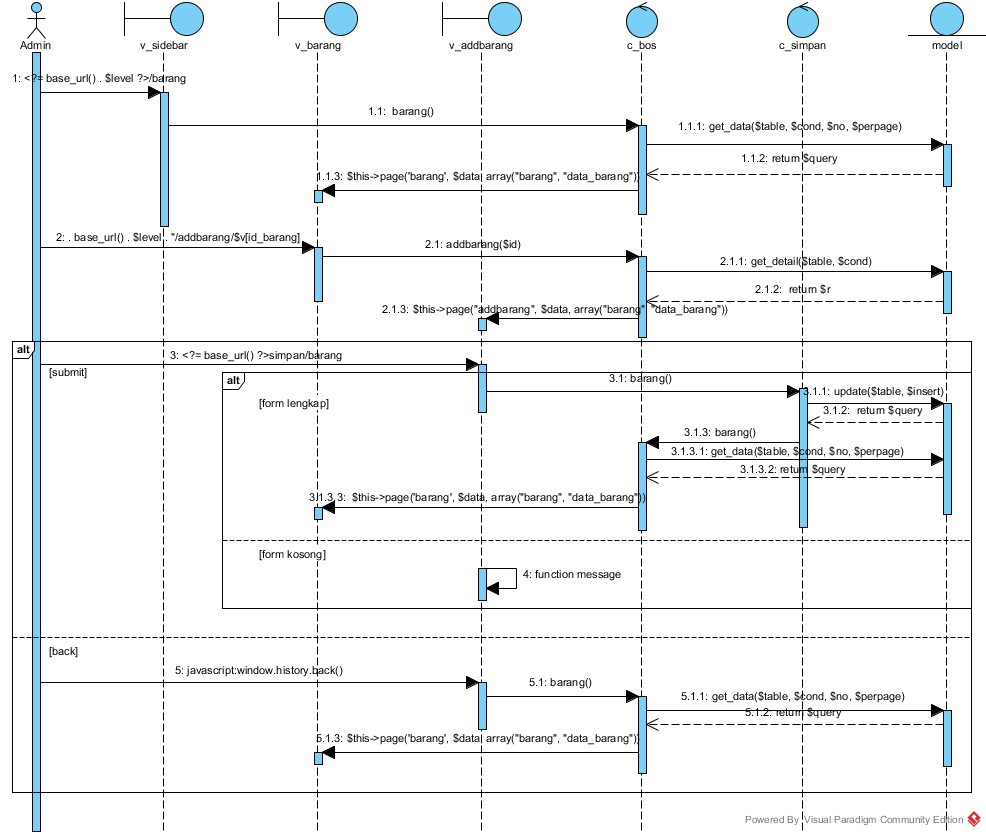
1. *Squence Diagram* Ubah Data Customer



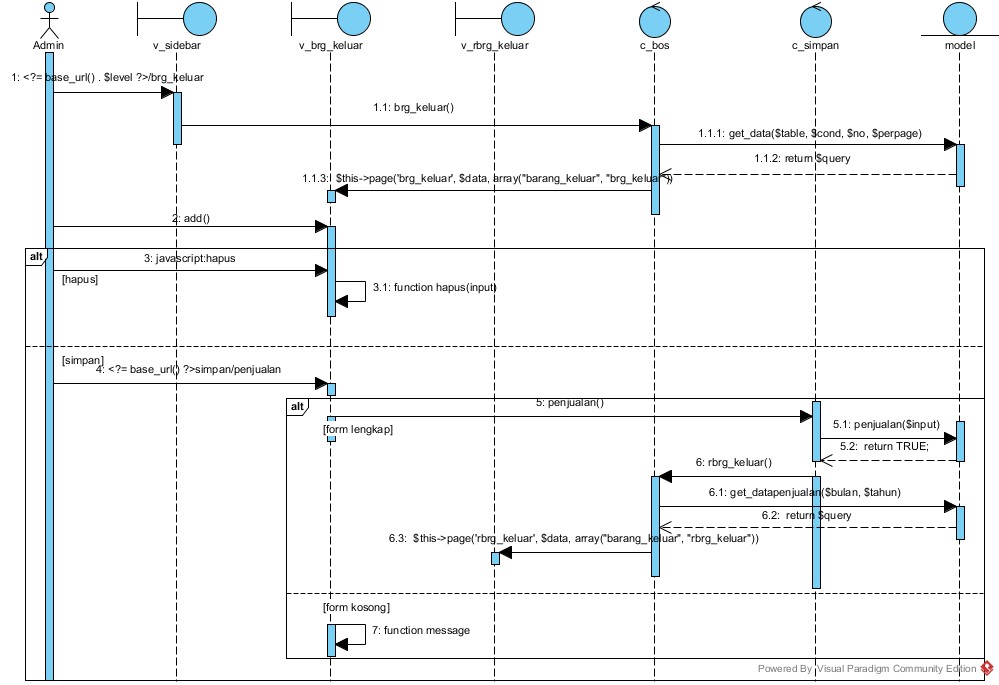
1. *Squence Diagram* Mengelola Data Barang
2. *Squence Diagram* Tambah Data Barang



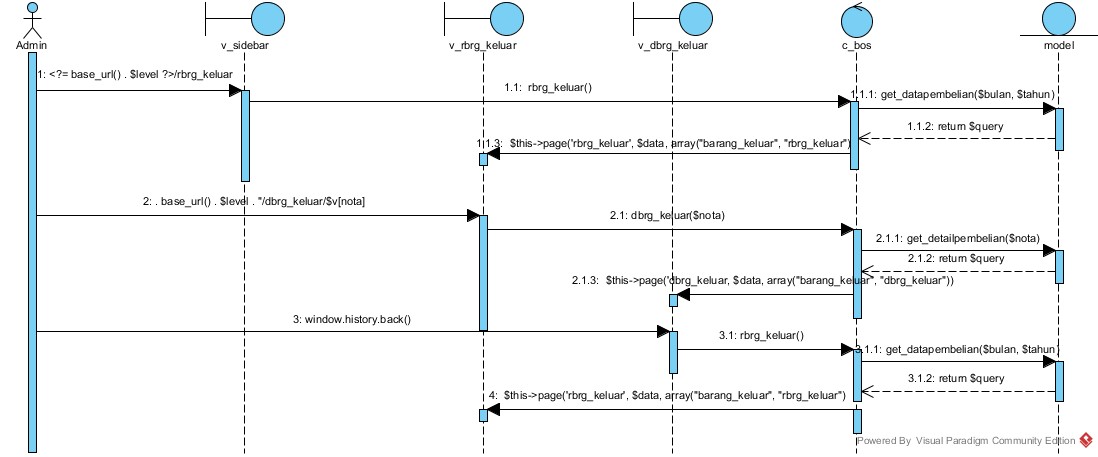
1. *Squence Diagram* Ubah Data Barang



1. *Squence Diagram* Menambah Data Barang Keluar



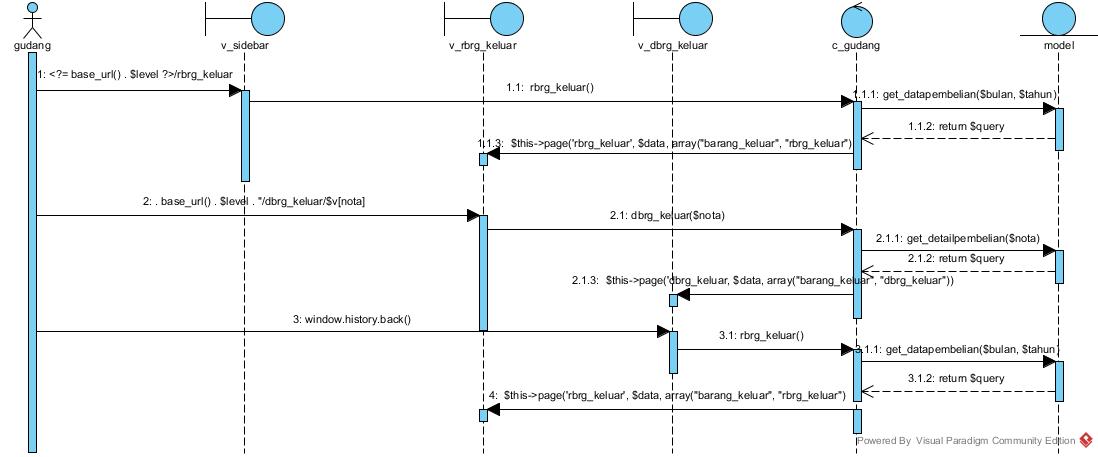
1. *Squence Diagram* Melihat Data Barang Keluar
2. *Squence Diagram* Melihat Data Barang Keluar – Admin



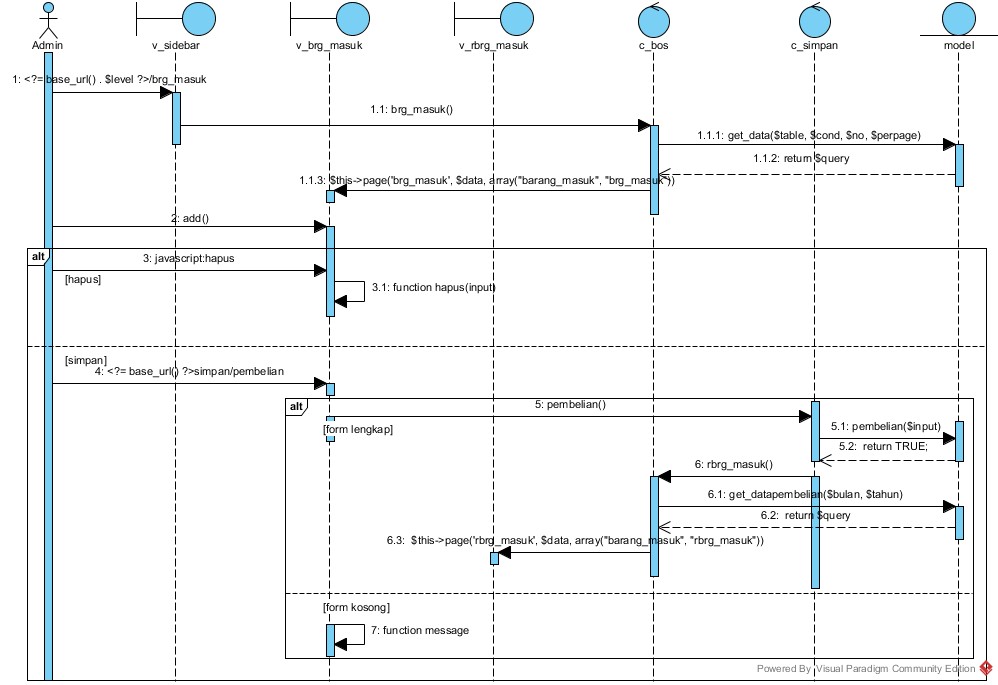
1. *Squence Diagram* Melihat Data Barang Keluar – Owner



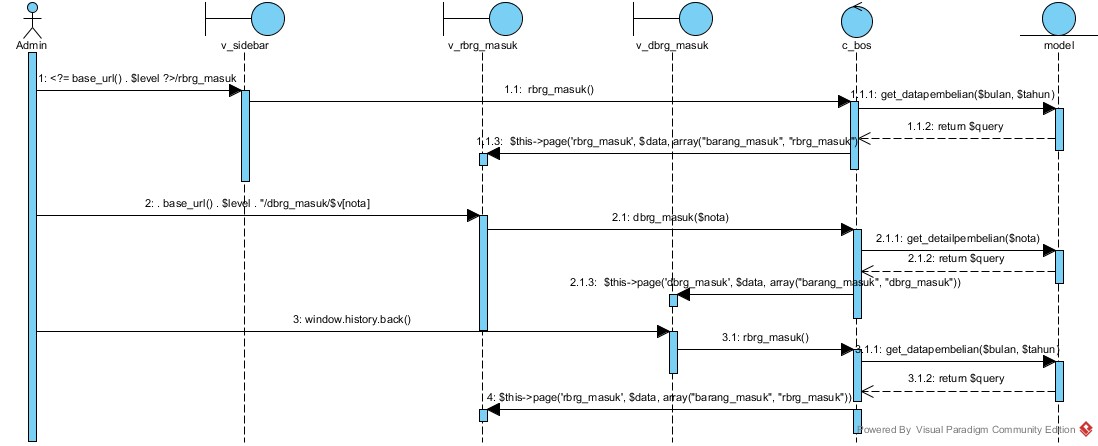
1. *Squence Diagram* Melihat Data Barang Keluar – Bagian Gudang



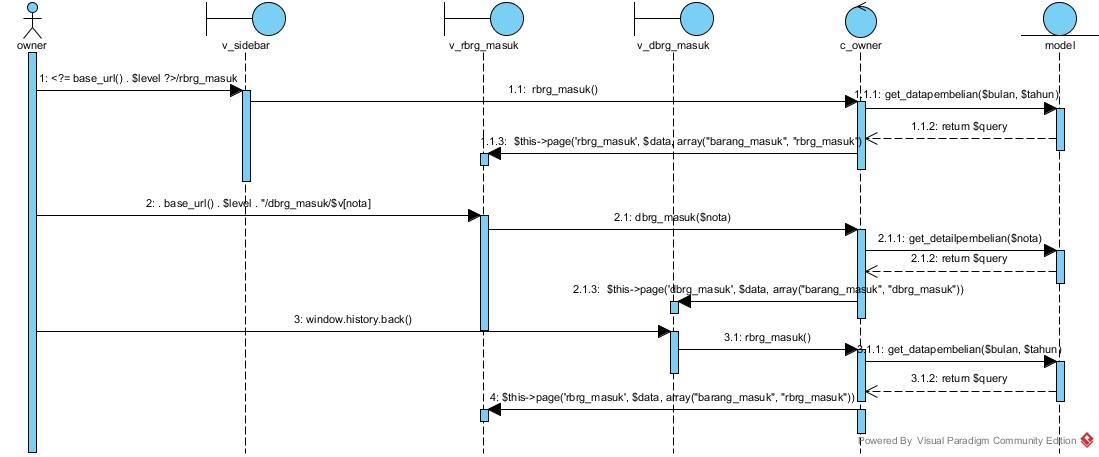
1. *Squence Diagram* Menambah Data Barang Masuk



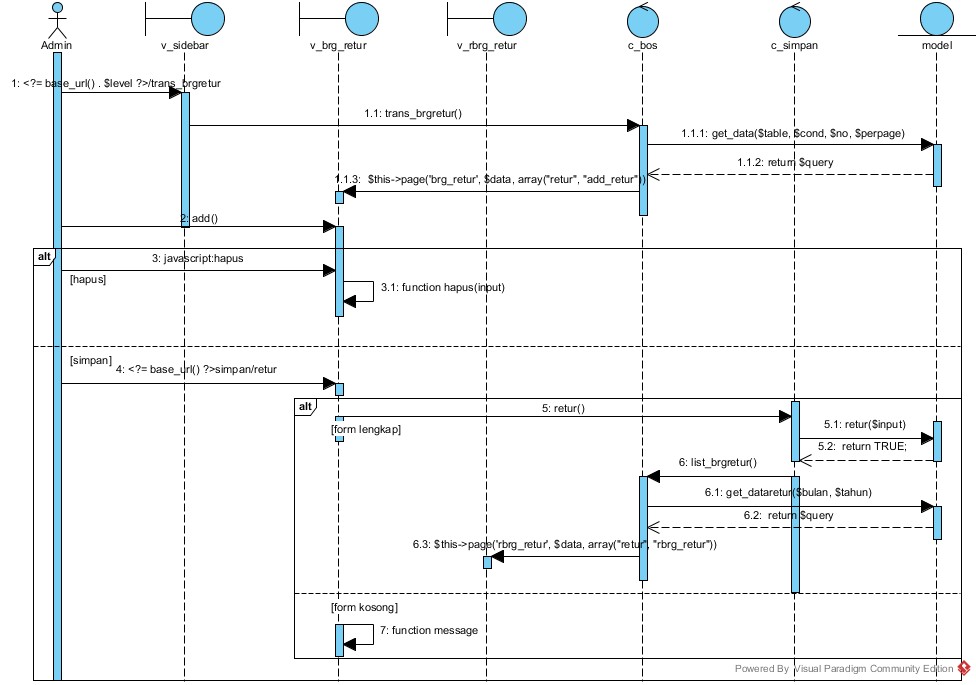
1. *Squence Diagram* Melihat Data Barang Masuk
2. *Squence Diagram* Melihat Data Barang Masuk – Admin



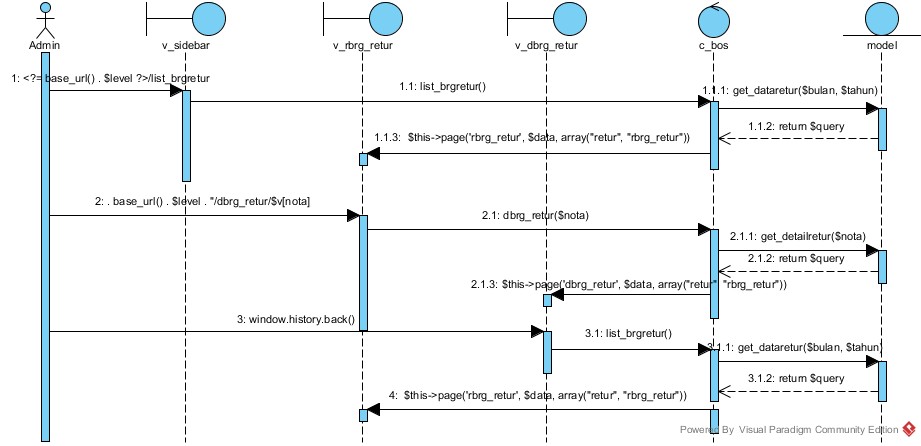
1. *Squence Diagram* Melihat Data Barang Masuk – Owner



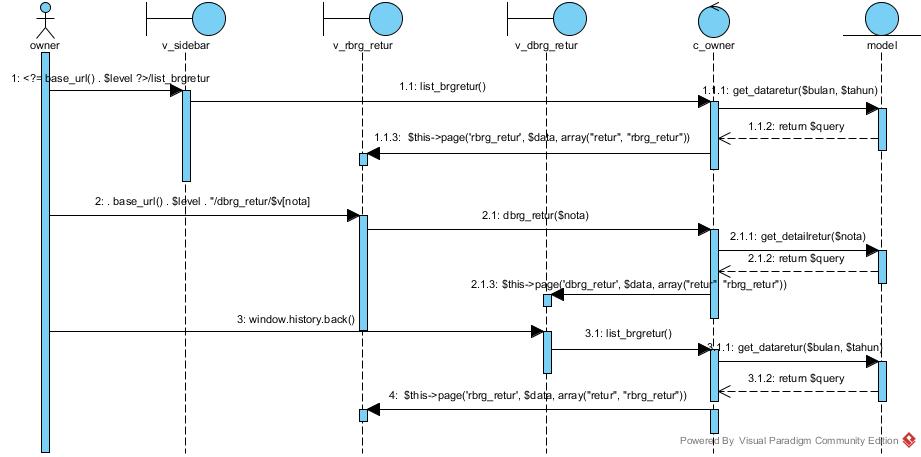
1. *Squence Diagram* Menambah Data Barang Retur



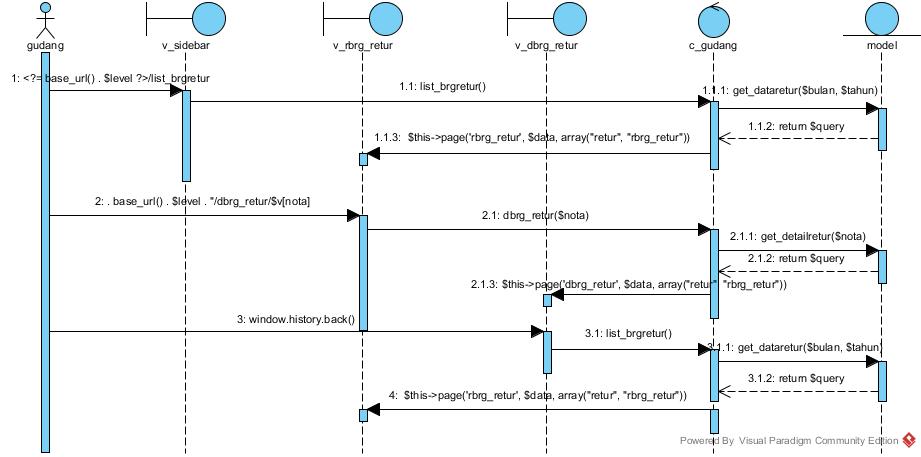
1. *Squence Diagram* Melihat Data Barang Retur
2. *Squence Diagram* Melihat Data Barang Retur – Admin



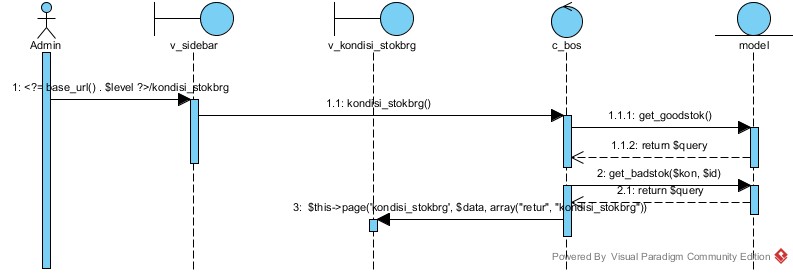
1. *Squence Diagram* Melihat Data Barang Retur – Owner



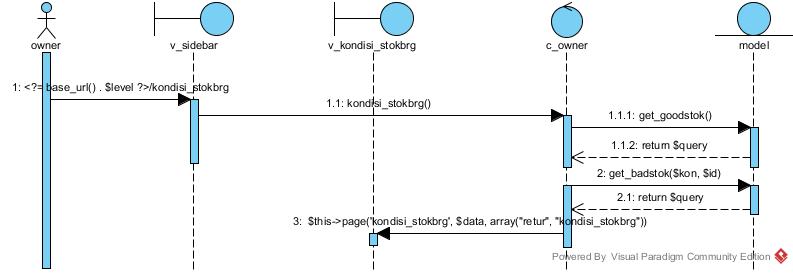
1. *Squence Diagram* Melihat Data Barang Retur – Bagian Gudang



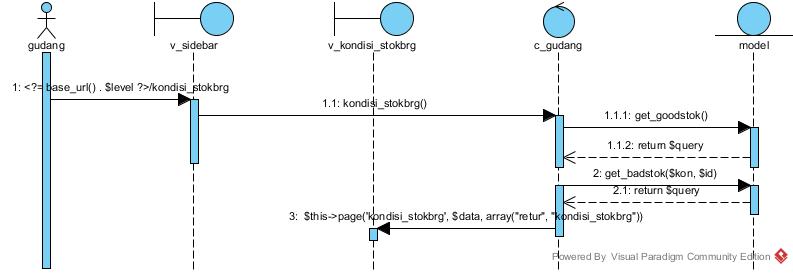
1. *Squence Diagram* Melihat Kondisi Stok
2. *Squence Diagram* Melihat Kondisi Stok – Admin



1. *Squence Diagram* Melihat Kondisi Stok – Owner



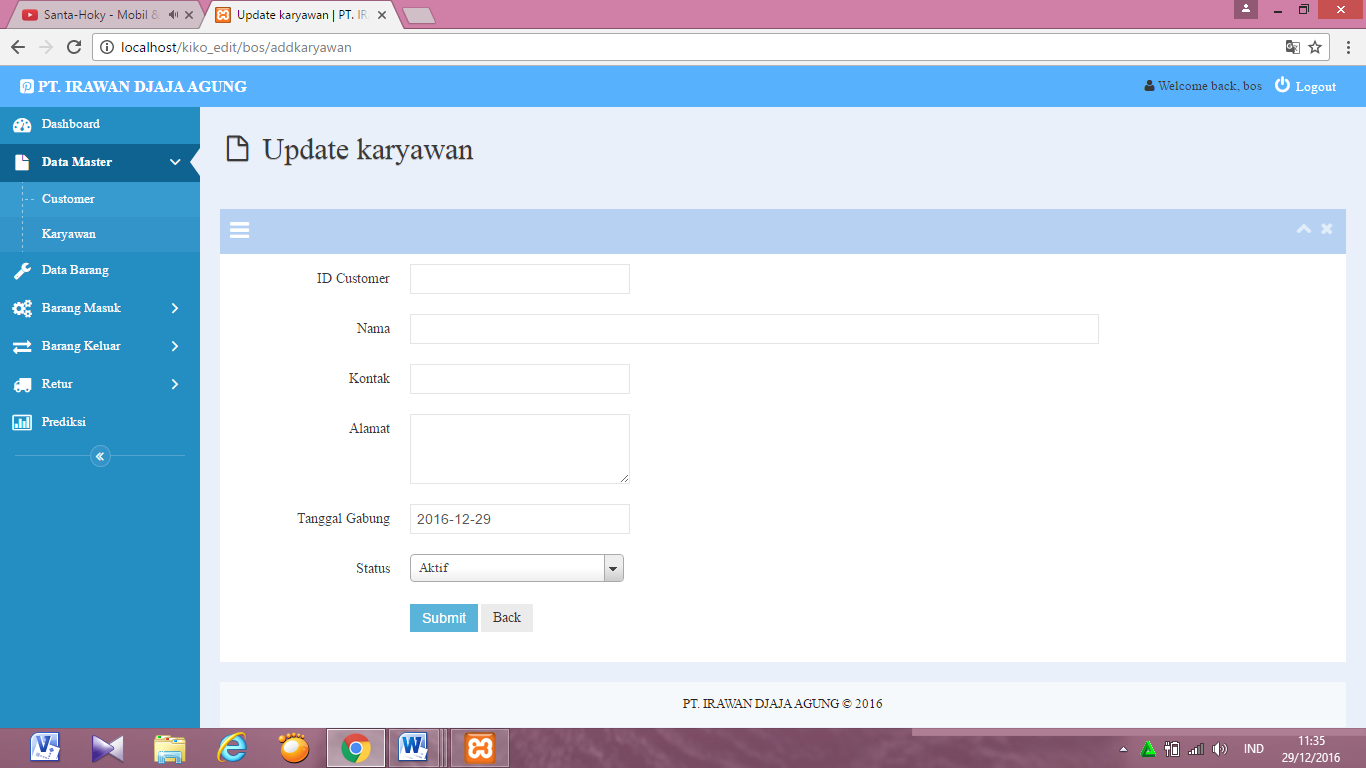
1. *Squence Diagram* Melihat Kodisi Stok – Bagian Gudang



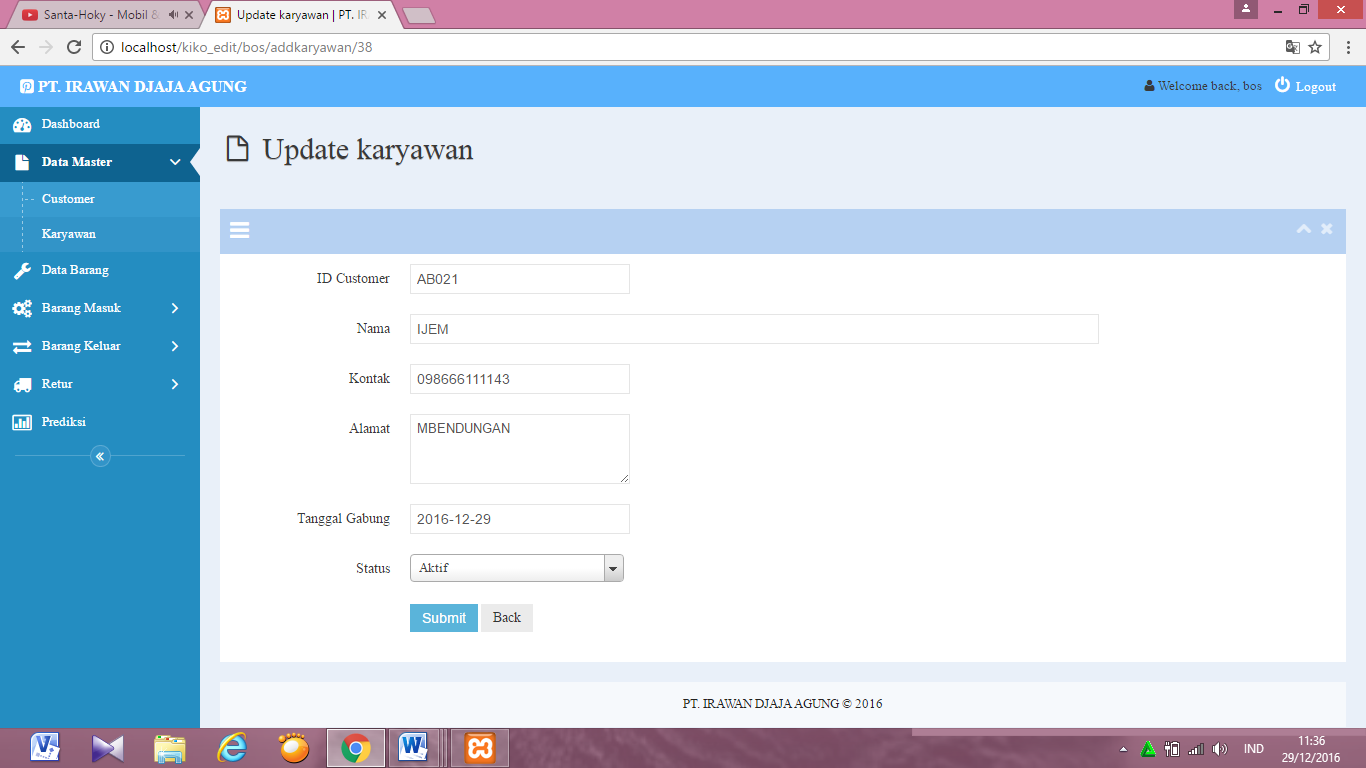
## LAMPIRAN B. FITUR APLIKASI

### B1. Fitur Karyawan

1. Tambah Data Karyawan

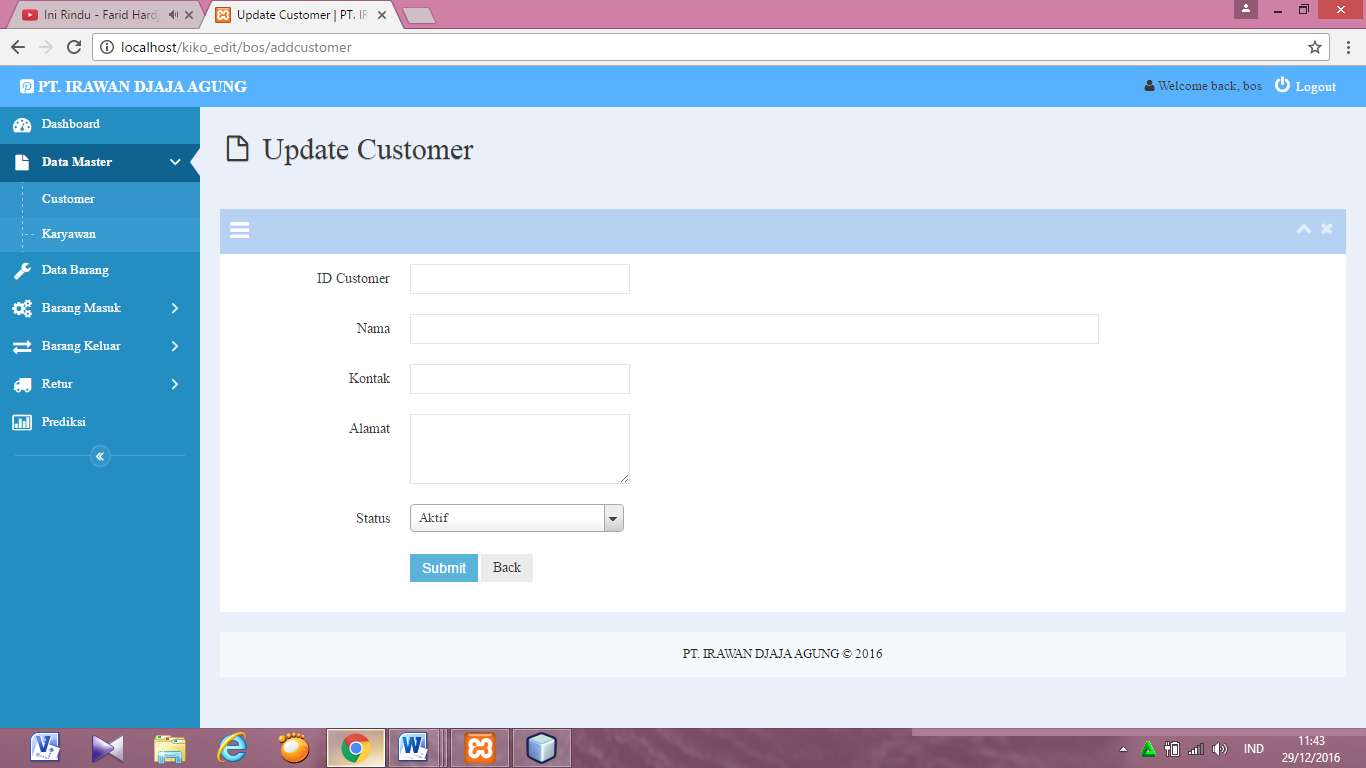


1. Ubah Data Karyawan

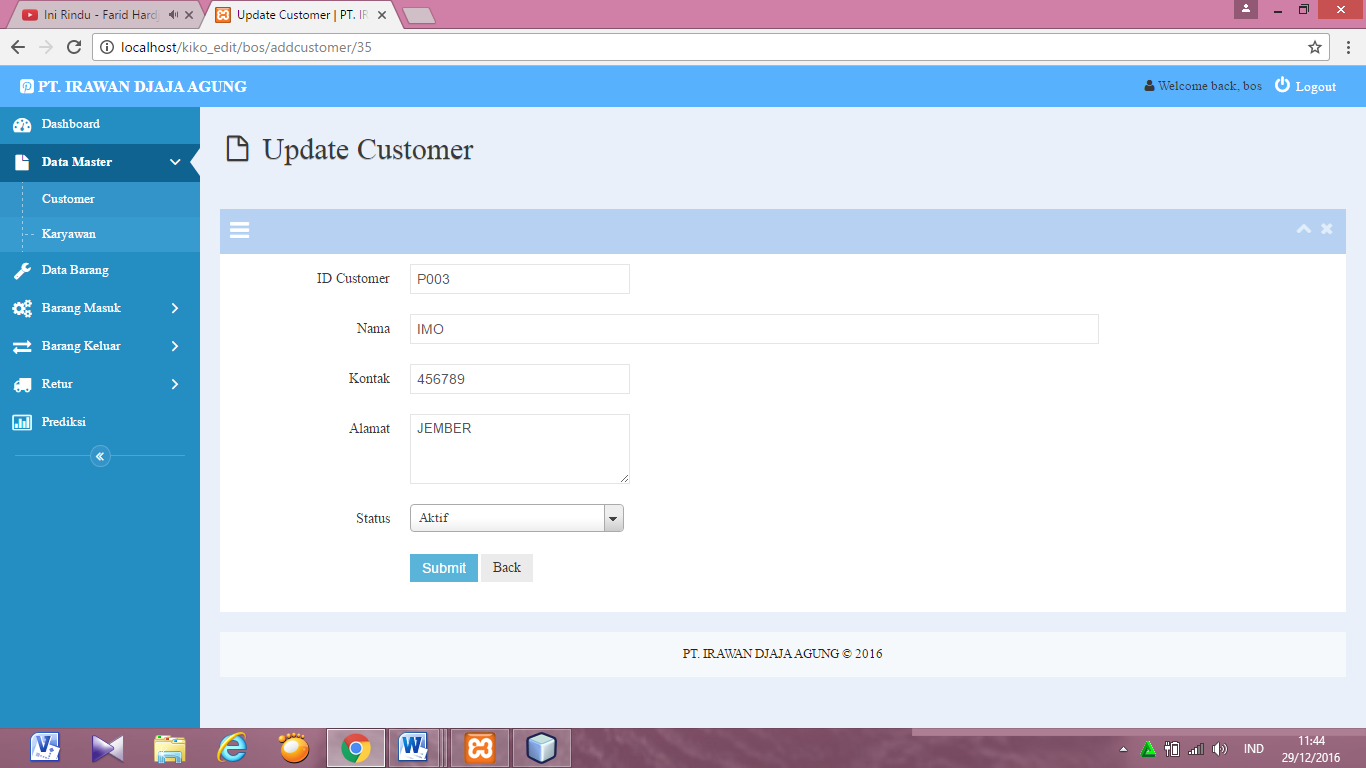


### B2. Fitur Customer

1. Tambah Data Customer

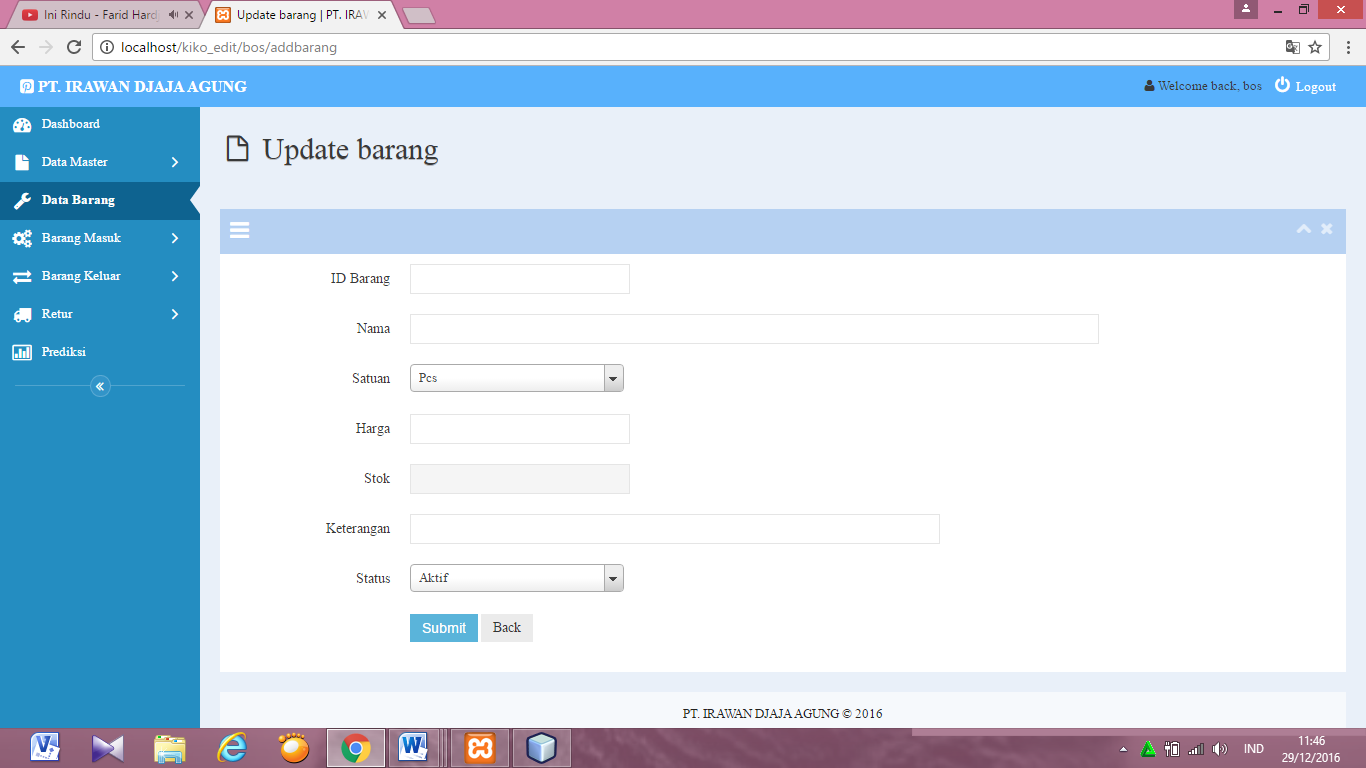


1. Ubah Data Customer

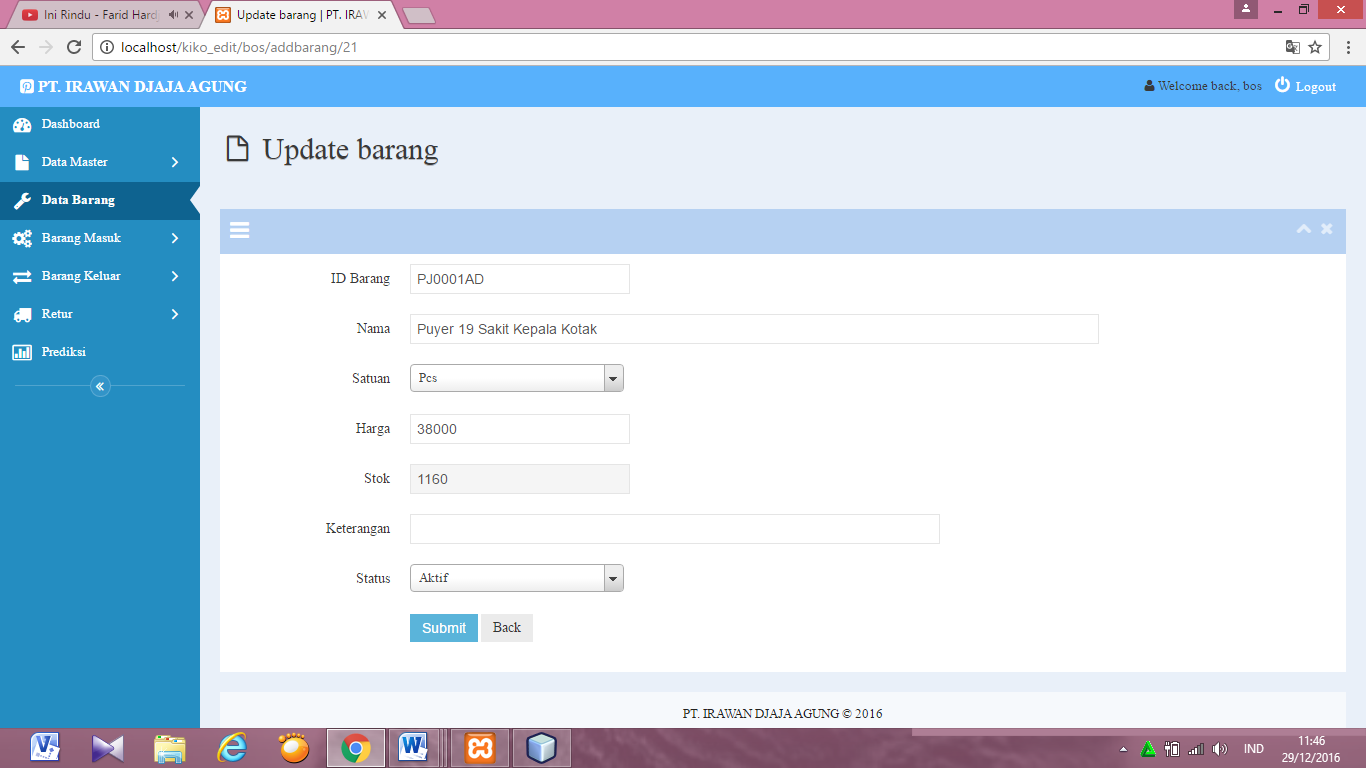


### B3. Fitur Data Barang

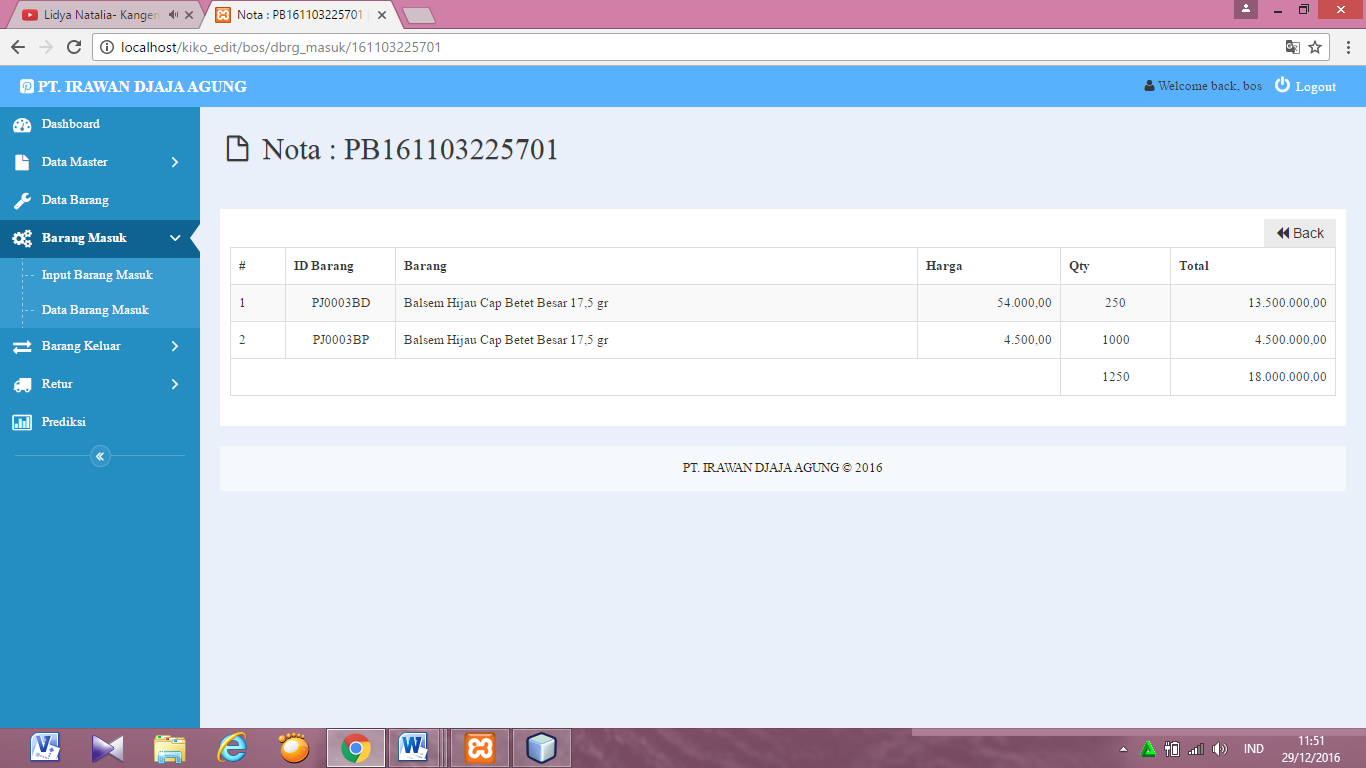
1. Tambah Data Barang



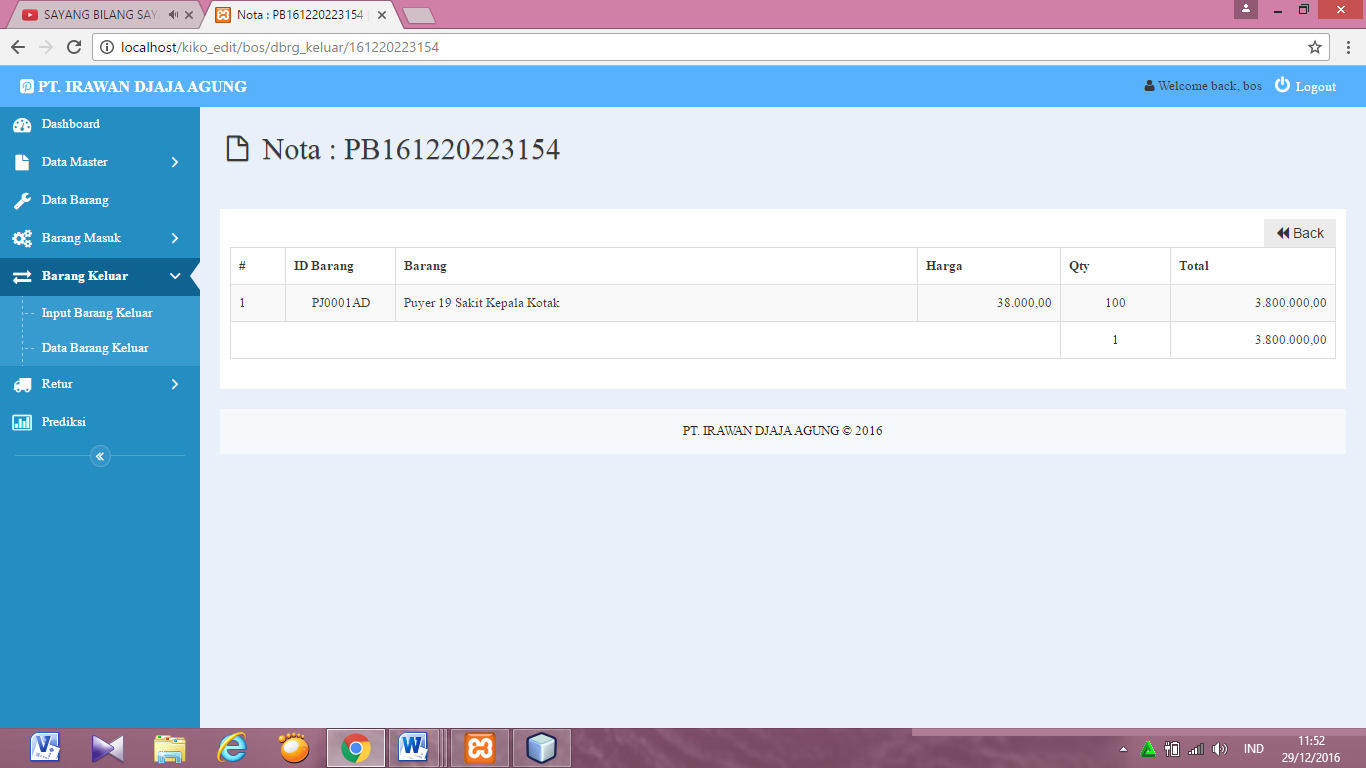
1. Ubah Data Barang



### B4. Fitur Detail Data Barang Masuk



### B5. Fitur Detail Data Barang Keluar



### B6. Fitur Detail Data Barang Retur

