



**IMPLEMENTASI METODE *JUST IN TIME* PADA SISTEM INFORMASI  
PREDIKSI STOK MINIMAL PADA DISTRIBUTOR FARMASI  
(STUDI KASUS PT. IRAWAN DJAJA AGUNG JEMBER)**

**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**

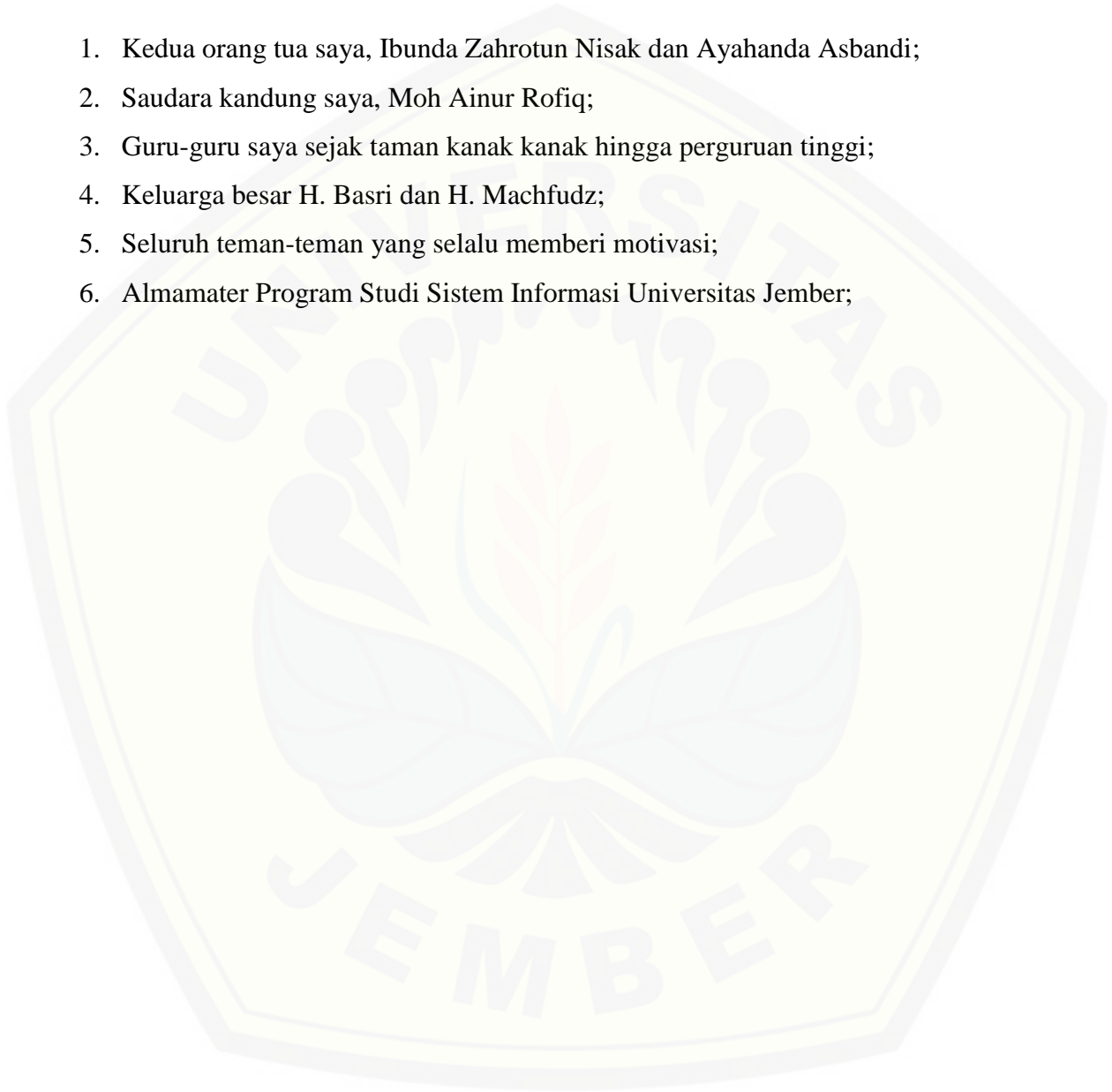
**UNIVERSITAS JEMBER**

**2016**

**PERSEMBAHAN**

Skripsi ini saya persembahkan untuk:

1. Kedua orang tua saya, Ibunda Zahrotun Nisak dan Ayahanda Asbandi;
2. Saudara kandung saya, Moh Ainur Rofiq;
3. Guru-guru saya sejak taman kanak kanak hingga perguruan tinggi;
4. Keluarga besar H. Basri dan H. Machfudz;
5. Seluruh teman-teman yang selalu memberi motivasi;
6. Almamater Program Studi Sistem Informasi Universitas Jember;



**PENGESAHAN**

Skripsi berjudul Implementasi Metode *Just In Time* Pada Sistem Informasi Prediksi Stok Minimal Pada Distributor Farmasi (Studi Kasus Pt. Irawan Djaja Agung Jember)”, telah diuji dan disahkan pada:

hari, tanggal :

tempat : Program Studi Sistem Informasi Universitas Jember

Penguji 1,

Penguji 2,

Anang Andrianto, S.T., M.T.

NIP. 196906151997021002

Fahrobby Adnan, S.Kom., M.M.Si

NIP.198706192014041001

Mengesahkan

Ketua Program Studi,

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

NIP. 196704201992011001

**MOTTO**

*“Dia yang tahu, tidak bicara. Dia yang bicara, tidak tahu”*

- Lao Tse-



**PERNYATAAN**

Saya yang bertanda tangan dibawah ini:

Nama : Rizky Ila Khoiriyah

NIM : 112410101093

menyatakan dengan sesungguhnya bahwa karya tulis ilmiah dengan judul “Implementasi Metode *Just In Time* Pada Sistem Informasi Prediksi Stok Minimal Pada Distributor Farmasi(Studi Kasus Pt. Irawan Djaja Agung Jember)” 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, 09 Juni 2017

Yang menyatakan,

Rizky Ila Khoiriyah

NIM.112410101093

**SKRIPSI**

**IMPLEMENTASI METODE *JUST IN TIME* PADA SISTEM INFORMASI  
PREDIKSI STOK MINIMAL PADA DISTRIBUTOR FARMASI  
(STUDI KASUS PT. IRAWAN DJAJA AGUNG JEMBER)**

Oleh

Rizky Ila Khoiriyah

NIM 112410101093

Pembimbing

Dosen Pembimbing Utama : Prof. Drs. Slamir, M.Comp. Sc., Ph.D

Dosen Pembimbing Anggota : Nelly Oktavia A, S. Si, MT.

**PENGESAHAN PEMBIMBING**

Skripsi Berjudul “Implementasi Metode *Just In Time* Pada Sistem Informasi Prediksi Stok Minimal Pada Distributor Farmasi(Studi Kasus Pt. Irawan Djaja Agung Jember)”, telah diuji dan disahkan pada:

hari, tanggal :

tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh :

Pembimbing I,

Pembimbing II,

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

Nelly Oktavia A, S. Si, MT.

NIP. 196704201992011001

NIP. 198410242009122008

## RINGKASAN

**Implementasi Metode *Just In Time* Pada Sistem Informasi Prediksi Stok Minimal Pada Distributor Farmasi (Studi Kasus Pt. Irawan Djaja Agung Jember);** Rizky Ila Khoitiyah, 112410101093; 2017; Program Studi Sistem Informasi Universitas Jember.

Distribusi merupakan kegiatan ekonomi yang menjembatani kegiatan produksi dan konsumsi. Masalah pendistribusian tidak terlepas dari bagaimana para pihak terkait dapat mengatur strategi dalam menyalurkan barang maupun jasa agar proses penyaluran dapat berjalan dengan baik. Strategi pendistribusian yang terencana dengan baik dimaksudkan untuk mengefektifkan penelusuran jalur yang optimal untuk pendistribusian barang.

PT. Irawan Djaja Agung adalah sebuah perusahaan yang bergerak dibidang distributor produk farmasi, meliputi kosmetik, obat-obatan, makanan, dan minuman. PT. Irawan Djaja Agung memiliki beberapa cabang di Povinsi Jawa Timur, salah satunya bertempat di Kecamatan Ajung, Jember. Untuk menjaga ketersediaan barang, maka PT. Irawan Djaja Agung Cabang Jember ini melakukan pengecekan dan penambahan stok pada gudang setiap seminggu sekali. Admin yang bertugas mengecek ketersediaan barang melakukan konfirmasi kepada gudang terkait adanya kekurangan atau kelebihan barang dan juga kerusakan barang yang terdapat pada gudang. Dikarenakan jenis barang yang beragam dan gudang dapat melakukan penerimaan barang dari pemasok dan pengiriman barang ke pelanggan dalam jumlah yang besar dengan jenis yang beragam pula, maka berakibat banyaknya ketidakcocokan antara data yang ada pada admin dan gudang. Akibatnya terjadi kekurangan dan kelebihan stok persediaan dalam gudang. Kekurangan stok dalam gudang dapat mengakibatkan bagian gudang tidak dapat memenuhi pesanan pelanggan dan menyebabkan perusahaan rugi secara materi. Kelebihan stok dalam gudang dapat



menimbulkan perusahaan menginvestasi dana berlebihan untuk biaya penyimpanan gudang.

Pada permasalahan tersebut penulis bermaksud untuk membuat sistem informasi prediksi stok minimal pada distributor menggunakan metode *Just In Time*. Penelitian ini bertujuan untuk membantu perusahaan memprediksi jumlah stok minimal yang dapat diambil oleh perusahaan pada tahun selanjutnya. Penggunaan sistem informasi prediksi stok minimal ini memiliki banyak manfaat. Hal itu dikarenakan sistem ini digunakan untuk menghitung kemungkinan stok minimal yang dapat diambil di tahun selanjutnya berdasarkan tahun-tahun sebelumnya sehingga perusahaan dapat meminimalisir penggunaan gudang agar tidak terlalu berlebih. Selain itu diharapkan laba yang diperoleh oleh perusahaan juga akan meningkat. Metode yang digunakan untuk membuat sistem Prediksi pengambilan stok barang adalah *Just In Time*. *Just In Time* adalah suatu konsep dimana bahan baku yang akan digunakan untuk aktifitas produksi didatangkan dari pemasok atau supplier 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.

## PRAKATA

Puji syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul Implementasi Metode *Just In Time* Pada Sistem Informasi Prediksi Stok Minimal Pada Distributor Farmasi (Studi Kasus Pt. Irawan Djaja Agung Jember)”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Prof. Drs. Slamun, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Prof. Drs. Slamun, M.Comp.Sc., Ph.D., selaku Dosen Pembimbing Utama, dan Nelly Oktavia A, S. Si, MT selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi ini;
3. Seluruh Bapak dan Ibu dosen beserta staf karyawan Program Studi Sistem Informasi;
4. Bapak, Ibu, dan adik tercinta yang telah memberikan do’a dan semangat kepada penulis sehingga skripsi ini dapat diselesaikan;
5. Teman-teman mahasiswa Program Studi Sistem Informasi seluruh angkatan;
6. Semua pihak yang tidak bisa disebutkan satu persatu;

Jember, 09 Juni 2017

Penulis

DAFTAR ISI

PERSEMBAHAN .....	ii
PENGESAHAN .....	iii
MOTTO .....	iv
SKRIPSI .....	vi
PENGESAHAN PEMBIMBING .....	vii
RINGKASAN.....	viii
PRAKATA .....	x
DAFTAR ISI .....	xi
DAFTAR TABEL .....	xvi
DAFTAR GAMBAR.....	xvii
BAB 1. PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan dan Manfaat .....	3
1.3.1 Tujuan.....	3
1.3.2 Manfaat.....	3
1.4 Batasan Masalah .....	4
1.5 Sistematika Penulisan .....	4
BAB 2. TINJAUAN PUSTAKA.....	5
2.1 Aplikasi .....	6
2.2 Prediksi .....	6
2.3 Persediaan .....	7
2.4 Metode <i>Just In Time</i> .....	8
2.5 Model <i>Waterfall</i> .....	10
2.5.1 Kelebihan Model <i>Waterfall</i> .....	11
2.5.2 Kekurangan Model <i>Waterfall</i> .....	12

BAB 3. METODOLOGI PENELITIAN .....	13
3.1 Analisis Kebutuhan .....	13
3.1.1 Pengumpulan Data.....	13
3.1.2 Gambaran Umum Aplikasi.....	16
3.2 Desain Aplikasi.....	16
3.2.1 <i>Business Process</i> .....	16
3.2.2 <i>Usecase Diagram</i> .....	17
3.2.3 <i>Scenario Diagram</i> .....	17
3.2.4 <i>Activity Diagram</i> .....	17
3.2.5 <i>Sequence Diagram</i> .....	17
3.2.6 <i>Class Diagram</i> .....	17
3.2.7 <i>Entity Relationship Diagram</i> .....	17
3.3 Implementasi Pembuatan Aplikasi .....	17
3.4 Pengujian Aplikasi .....	18
BAB 4. DESAIN DAN IMPLEMENTASI.....	19
4.1 Analisis Kebutuhan .....	19
4.2 <i>Bussiness Process</i> .....	20
4.3 <i>Usecase Diagram</i> .....	21
4.4 <i>Scenario</i> .....	25
4.4.1 <i>Scenario Login</i> .....	25
4.4.2 <i>Scenario Mengelola Data Karyawan</i> .....	29
4.4.3 <i>Scenario Mengelola Data Customer</i> .....	29
4.4.4 <i>Scenario Mengelola Data User</i> .....	30
4.4.5 <i>Scenario Mengelola Data Barang</i> .....	31
4.4.6 <i>Scenario Menambah Data Barang Masuk</i> .....	32
4.4.7 <i>Scenario Melihat Data Barang Masuk</i> .....	32
4.4.8 <i>Scenario Menambah Data Barang Keluar</i> .....	32
4.4.9 <i>Scenario Melihat Data Barang Keluar</i> .....	33

4.4.10	<i>Scenario Menambah Data Barang Retur</i> .....	33
4.4.11	<i>Scenario Melihat Data Barang Retur</i> .....	33
4.4.12	<i>Scenario Melihat Kondisi Stok Barang</i> .....	34
4.4.13	<i>Scenario Prediksi Pengambilan Stok Barang</i> .....	34
4.5	<i>ActivityDiagram</i> .....	35
4.5.1	<i>Activity Diagram Login</i> .....	35
4.5.2	<i>Activity Diagram Mengelola Data Karyawan</i> .....	39
4.5.3	<i>Activity Diagram Mengelola Data Customer</i> .....	40
4.5.4	<i>Activity Diagram Mengelola Data User</i> .....	41
4.5.5	<i>Activity Diaram Mengelola Data Barang</i> .....	41
4.5.6	<i>Activity Diagram Menambah Data Barang Masuk</i> .....	42
4.5.7	<i>Activity Diagram Melihat Data Barang Masuk</i> .....	42
4.5.8	<i>Activity Diagram Mengelola Data Barang Keluar</i> .....	42
4.5.9	<i>Activity Diagram Melihat Data Barang Keluar</i> .....	43
4.5.10	<i>Activity Diagram Menambah Data Barang Retur</i> .....	43
4.5.11	<i>Activity Diagram Melihat Data Barang Retur</i> .....	43
4.5.12	<i>Activity Diagram Melihat Kondisi Stok Barang</i> .....	44
4.5.13	<i>Activity Diagram Prediksi Pengambilan Stok Barang</i> .....	44
4.6	<i>Sequence Diagram</i> .....	45
4.6.1	<i>Sequence Diagram Login</i> .....	46
4.6.2	<i>Sequence Diagram Mengelola Data Karyawan</i> .....	49
4.6.3	<i>Sequence Diagram Mengelola Data Customer</i> .....	50
4.6.4	<i>Sequence Diagram Mengelola Data User</i> .....	51
4.6.5	<i>Sequence Diagram Mengelola Data Barang</i> .....	52
4.6.6	<i>Sequence Diagram Menambah Data Barang Masuk</i> .....	53
4.6.7	<i>Sequence Diagram Melihat Data Barang Masuk</i> .....	54
4.6.8	<i>Sequence Diagram Menambah Data Barang Keluar</i> .....	55
4.6.9	<i>Sequence Diagram Melihat Data Barang Keluar</i> .....	56

4.6.10	<i>Sequence Diagram</i> Menambah Data Barang Retur.....	58
4.6.11	<i>Sequence Diagram</i> Melihat Data Barang Retur .....	58
4.6.12	<i>Sequence Diagram</i> Melihat Kondisi Stok Barang.....	60
4.6.13	<i>Sequence Diagram</i> Prediksi Pengambilan Stok Barang.....	61
4.7	<i>Class Diagram</i> .....	62
4.8	<i>EntityRelationship Diagram</i> (ERD).....	63
4.9	<i>Implementasi Perancangan</i> .....	63
4.10	<i>Pengujian Aplikasi</i> .....	64
4.10.1	White Box Testing.....	64
4.10.2	Black Box Testing .....	69
BAB 5. HASIL DAN PEMBAHASAN.....		70
5.1	Hasil Penelitian .....	70
5.2	Hasil Pembuatan Aplikasi.....	70
5.2.1	Tampilan Fitur <i>Login</i> .....	70
5.2.2	Tampilan <i>Home</i> .....	71
5.2.3	Tampilan Fitur Karyawan.....	71
5.2.4	Tampilan Fitur Customer .....	72
5.2.5	Tampilan Fitur User .....	73
5.2.6	Tampilan Fitur Data Barang.....	73
5.2.7	Tampilan Fitur Input Data Barang Masuk.....	73
5.2.8	Tampilan Fitur Data Barang Masuk.....	74
5.2.9	Tampilan Fitur Input Data Barang Keluar.....	75
5.2.10	Tampilan Fitur Data Barang Keluar .....	76
5.2.11	Tampilan Fitur Input Data Barang Retur.....	76
5.2.12	Tampilan Fitur Data Barang Retur .....	77
5.2.13	Tampilan Fitur Kondisi Barang.....	78
5.2.14	Tampilan Fitur Prediksi Pengambilan Stok Barang .....	78
5.3	Pembahasan .....	79

BAB 6. PENUTUP.....	94
6.1 Kesimpulan.....	94
6.2 Saran.....	95
DAFTAR PUSTAKA.....	96
LAMPIRAN A. PERANCANGAN APLIKASI.....	97
A1. Perancangan <i>Scenario</i> .....	97
A2. Perancangan <i>Activity Diagram</i> .....	113
A3. Perancangan <i>Squence Diagram</i> .....	119
LAMPIRAN B. FITUR APLIKASI.....	133
B1. Fitur Karyawan.....	133
B2. Fitur Customer.....	134
B3. Fitur Data User.....	135
B4. Fitur Data Barang.....	136
B5. Fitur Detail Data Barang Masuk.....	137
B5. Fitur Detail Data Barang Keluar.....	137
B8. Fitur Detail Data Barang Retur.....	138

DAFTAR TABEL

Tabel 4.1 Definisi Aktor <i>Usecase</i> .....	22
Tabel 4.2 Definisi <i>Usecase</i> .....	23
Tabel 4.3 <i>Scenario Login Admin</i> .....	26
Tabel 4.4 <i>Scenario Login Owner</i> .....	26
Tabel 4.5 <i>Scenario Login Bagian Gudang</i> .....	27
Tabel 4.6 <i>Scenario Login Sales</i> .....	28
Tabel 4.7 <i>Scenario Prediksi Pengambilan Stok Barang</i> .....	34
Tabel 4.8 <i>Test Case</i> Fitur Prediksi .....	68
Tabel 4.9 Black Box Testing Perhitungan Prediksi Pengambilan Stok Barang .....	69
Tabel 5.1 Data Penjualan Balsem Hijau Cap Betet Ukuran Kecil 7,5 gram Tahun 2013 .....	80
Tabel 5.2 Data Penjualan Balsem Hijau Cap Betet Ukuran Kecil 7,5 gram Tahun 2014 .....	82
Tabel 5.3 Data Penjualan Balsem Hijau Cap Betet Ukuran Kecil 7,5 gram Tahun 2015 .....	84
Tabel 5.4 Perhitungan Indeks Waktu .....	86
Tabel 5.5 Tabel Untuk Mengetahui Nilai $x$ .....	87
Tabel 5.6 Tabel Perhitungan Nilai $\bar{y}$ Tahun 2013 .....	88
Tabel 5.7 Tabel Perhitungan Nilai $\bar{y}$ Tahun 2015 .....	88
Tabel 5.8 Tabel Hasil dan Rata-Rata Nilai $\bar{y}$ Tahun 2014 dan 2015 .....	89
Tabel 5.9 Hasil Perhitungan Perkiraan Nilai Stok Minimal Balsem Hijau Cap Betet Ukuran Kecil 7,5 gram Untuk Tahun 2016 .....	89
Tabel 5.10 Jumlah Stok Minimal Balsem Hijau Cap Betet Ukuran Kecil 7,5 gr di Tahun 2016.....	91



## DAFTAR GAMBAR

Gambar 2.1 Model Waterfall (Darmawan, 2015).....	10
Gambar 3.1 Diagram Alir Sistem Informasi Prediksi Pengambilan Stok Barang.....	15
Gambar 4.1 <i>Bussiness Process</i> Sistem Informasi Prediksi Pengambilan Stok Barang	20
Gambar 4.2 <i>Usecase Diagram</i> .....	21
Gambar 4.3 <i>Activity Diagram Login Admin</i> .....	36
Gambar 4.4 <i>Activity Diagram Login Owner</i> .....	37
Gambar 4.5 <i>Activity Diagram Login Bagian Gudang</i> .....	38
Gambar 4.6 <i>Activity Diagram Login Sales</i> .....	39
Gambar 4.7 <i>Activity Diagram</i> Prediksi Pengambilan Stok Barang.....	45
Gambar 4.8 <i>Sequence Diagram Login Admin</i> .....	46
Gambar 4.9 <i>Sequence Diagram Login Owner</i> .....	47
Gambar 4.10 <i>Sequence Diagram Login Bagian Gudang</i> .....	48
Gambar 4.11 <i>Sequence Diagram</i> Prediksi Pengambilan Stok Barang.....	61
Gambar 4.12 <i>Class Diagram</i> Sistem Informasi Prediksi Pengambilan Stok Barang...	62
Gambar 4.13 ERD Prediksi Pengambilan Stok Barang .....	63
Gambar 4.14 <i>Method</i> prediksi_jit() .....	64
Gambar 4.15 <i>Method</i> proses_prediksi_jit() .....	65
Gambar 4.16 Diagram Alir <i>Method</i> prediksi_jit().....	66
Gambar 4.17 Diagram Alir <i>Method</i> proses_prediksi_jit() .....	66
Gambar 5.1 Tampilan Fitur <i>Login</i> .....	71
Gambar 5.2 Tampilan Fitur Karyawan.....	72
Gambar 5.3 Tampilan Fitur Customer.....	72
Gambar 5.4 Tampilan Fitur User.....	73

Gambar 5.5 Tampilan Fitur Data Barang .....	74
Gambar 5.6 Tampilan fitur Input Barang Masuk .....	74
Gambar 5.7 Tampilan Fitur Data Barang Masuk .....	75
Gambar 5.8 Tampilan Fitur Barang Keluar .....	75
Gambar 5.9 Tampilan Fitur Data Barang Keluar .....	76
Gambar 5.10 Tampilan Fitur Input Barang Retur .....	77
Gambar 5.11 Tampilan Fitur Data Barang Retur .....	77
Gambar 5.12 Tampilan Fitur Kondisi Stok Barang .....	78
Gambar 5.13 Tampilan Fitur Prediksi .....	79
Gambar 5.14 Detail Prediksi Pengambilan Stok Barang Minimal Pada Tahun 2016.	93

## 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.

### 1.1 Latar Belakang

Pendistribusian identik dengan kata penyaluran atau saluran, dimana dapat didefinisikan dalam beberapa arti. Umumnya definisi yang ada memberi gambaran tentang pendistribusian ini sebagai suatu jalur (saluran). Penyaluran merupakan suatu jalur yang dilalui oleh arus barang yang berupa fisik maupun nonfisik dari beberapa pihak terkait yaitu antara penghasil dan penerima. Distribusi merupakan kegiatan ekonomi yang menjembatani kegiatan produksi dan konsumsi. Masalah pendistribusian tidak terlepas dari bagaimana para pihak terkait dapat mengatur strategi dalam menyalurkan barang maupun jasa agar proses penyaluran dapat berjalan dengan baik. Strategi pendistribusian yang terencana dengan baik dimaksudkan untuk mengefektifkan penelusuran jalur yang optimal untuk pendistribusian barang.

Strategi distribusi yang efektif dan dapat bekerja secara optimal memiliki tiga faktor penting. Faktor yang pertama adalah faktor wilayah, yang berarti bahwa diperlukannya pengetahuan tentang wilayah distribusi agar distributor dapat menggarap wilayah distribusi secara optimal. Faktor kedua adalah faktor persediaan, yang berarti diperlukan adanya pertimbangan dalam pengambilan keputusan mengenai berapa jumlah persediaan untuk pengiriman berikutnya. Faktor ketiga adalah transportasi, yang berarti diperlukannya proses yang mengatur perencanaan jadwal pengiriman (Eroy, 2010).

PT. Irawan Djaja Agung adalah sebuah perusahaan yang bergerak dibidang distributor produk farmasi, meliputi kosmetik, obat-obatan, makanan, dan minuman. PT. Irawan Djaja Agung memiliki beberapa cabang di Povinsi Jawa Timur, salah satunya bertempat di Kecamatan Ajung, Jember. Perusahaan ini hanya mendistribusikan produk berlisensi IDA, Cap 19, Cap Betet, dan Cap Burung & 2 Bidadari. Pada PT. Irawan Djaja Agung Cabang Jember ini bertugas untuk mendistribusikan persediaan ke setiap gerai penjual produk IDA, Cap 19, Cap Betet, dan Cap Burung & 2 Bidadari yang ada di area Kabupaten Jember dan sekitarnya.

Untuk menjaga ketersediaan barang, maka PT. Irawan Djaja Agung Cabang Jember ini melakukan pengecekan dan penambahan stok pada gudang setiap seminggu sekali. Admin yang bertugas mengecek ketersediaan barang melakukan konfirmasi kepada gudang terkait adanya kekurangan atau kelebihan barang dan juga kerusakan barang yang terdapat pada gudang. Dikarenakan jenis barang yang beragam dan gudang dapat melakukan penerimaan barang dari pemasok dan pengiriman barang ke pelanggan dalam jumlah yang besar dengan jenis yang beragam pula, maka berakibat banyaknya ketidak cocokan antara data yang ada pada admin dan gudang. Akibatnya terjadi kekurangan dan kelebihan stok persediaan dalam gudang. Kekurangan stok dalam gudang dapat mengakibatkan bagian gudang tidak dapat memenuhi pesanan pelanggan dan menyebabkan perusahaan rugi secara materi. Kelebihan stok dalam gudang dapat menimbulkan perusahaan menginvestasi dana berlebihan untuk biaya penyimpanan gudang.

Pada permasalahan tersebut penulis bermaksud untuk membuat sistem informasi prediksi stok minimal pada distributor menggunakan metode *Just In Time*. Penelitian ini bertujuan untuk membantu perusahaan memprediksi jumlah stok minimal yang dapat diambil oleh perusahaan pada tahun selanjutnya. Penggunaan sistem informasi prediksi stok minimal ini memiliki banyak manfaat. Hal itu dikarenakan sistem ini digunakan untuk menghitung kemungkinan stok minimal yang dapat diambil di tahun selanjutnya berdasarkan tahun-tahun sebelumnya sehingga perusahaan dapat meminimalisir penggunaan gudang agar tidak terlalu berlebih.

Selain itu diharapkan laba yang diperoleh oleh perusahaan juga akan meningkat. Metode yang digunakan untuk membuat sistem Prediksi pengambilan stok barang adalah *Just In Time*. *Just In Time* adalah adalah suatu konsep dimana bahan baku yang akan digunakan untuk aktifitas produksi didatangkan dari pemasok atau supplier 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.

## 1.2 Rumusan Masalah

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

1. Bagaimana menerapkan metode *Just In Time* pada pembuatan sistem prediksi stok minimal pada distributor farmasi?
2. Bagaimana merancang dan membangun implementasi metode *Just In Time* pada sistem informasi prediksi stok minimal pada distributor farmasi?

## 1.3 Tujuan dan Manfaat

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

### 1.3.1 Tujuan

Berdasarkan rumusan masalah yang telah disampaikan sebelumnya maka dapat ditetapkan tujuan yang ingin dicapai dari penelitian ini, yaitu merancang dan membangun implementasi metode *Just In Time* pada sistem informasi prediksi stok minimal pada distributor farmasi, untuk mencapai tujuan tersebut penelitian menggunakan model *waterfall* sebagai metode pengembangan aplikasi dalam melakukan penelitian.

### 1.3.2 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.

2. 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.

3. Manfaat bagi penulis

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

#### 1.4 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.

#### 1.5 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.

2. Tinjauan Pustaka

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

3. Metodologi Penelitian

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

4. Desain dan Implementasi

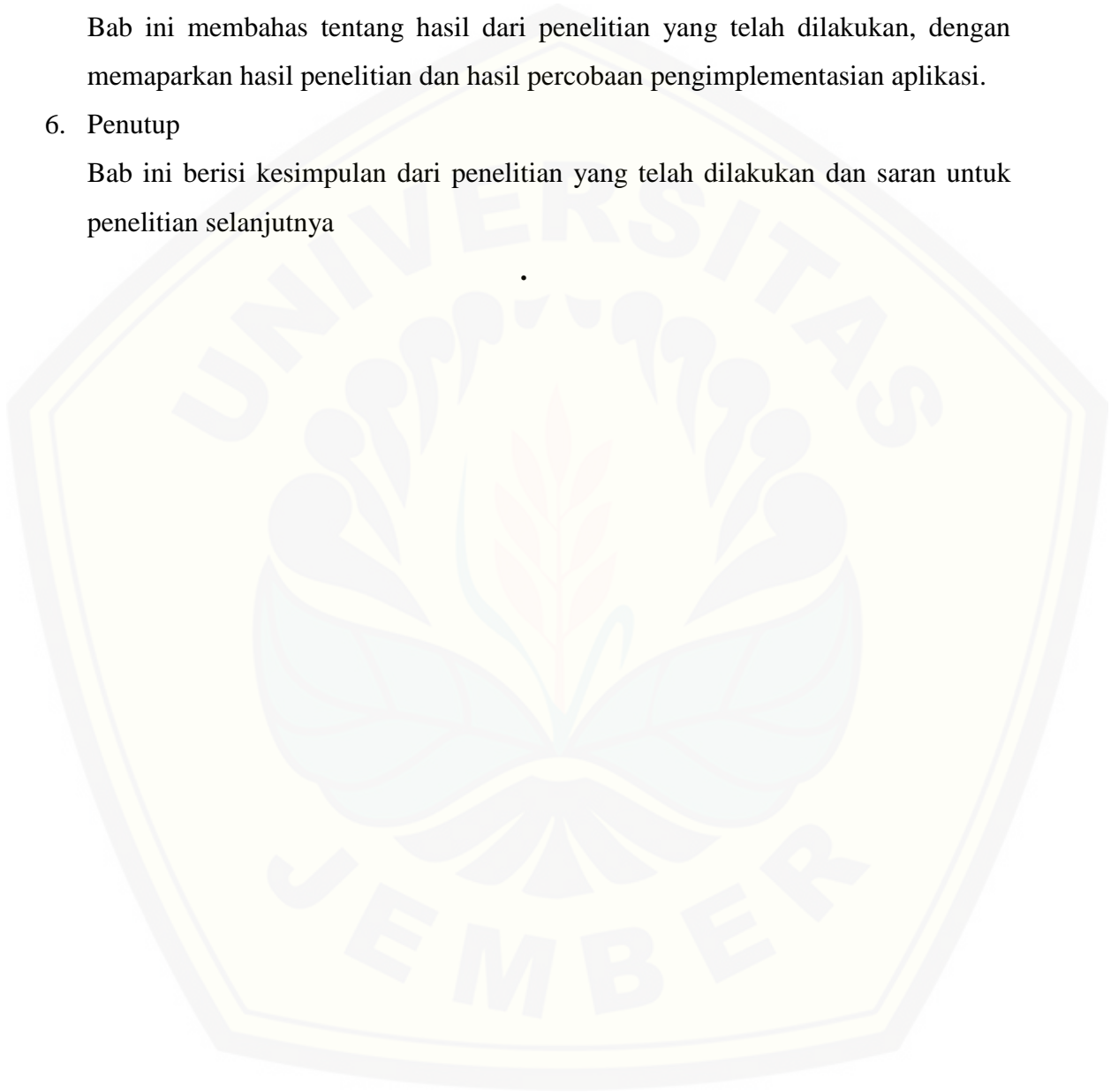
Bab ini menguraikan tentang desain dan implementasi desain kedalam bahasa pemrograman untuk membangun aplikasi.

5. Hasil dan Pembahasan

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

6. 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.

### 2.1 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 pelayanan 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.

### 2.2 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 peramalan. 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)



### 2.3 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)

Persediaan 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.

#### 2.4 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 serta 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-komponennya 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)

$$Y = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + b_4X_4 + b_5X_5 + e_i$$

Keterangan :

Y = efektifitas dan efisiensi biaya produksi

- $X_1$  = pembelian  
 $X_2$  = produksi  
 $X_3$  = pengiriman bahan baku  
 $X_4$  = pengiriman bahan jadi  
 $X_5$  = lingkungan JIT  
 $b_0$  = intersep atau konstanta  
 $b_1 - b_5$  = koefisien regresi  
 $e_i$  = faktor pengganggu

Adapun penghitungan untuk prediksi stok barang adalah sebagai berikut:

(Rahayu, 2003)

$$Ft = \frac{\bar{t}}{\sum \bar{t}} \times \bar{y}$$

Dimana :

- $Ft$  = nilai peramalan  
 $\bar{t}$  = indeks waktu  
 $\bar{y}$  = perkiraan tahunan

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

$$\bar{t} = n \times 12$$

Dimana n adalah prosentase terhadap total data

$$\bar{y} = a + b(x)$$

$$b = \frac{\sum xy - \bar{x}\bar{y}}{\sum x^2 - \bar{x}\sum x} \text{ atau } b = \frac{K2 - K1}{(tK1 - tK2)}$$

Keterangan :

- $x$  = variabel yang mempengaruhi  
 $y$  = kecenderungan nilai peramalan  
 $a$  = konstanta  
 $b$  = tingkat kecenderungan  
 $K1$  = kelompok 1

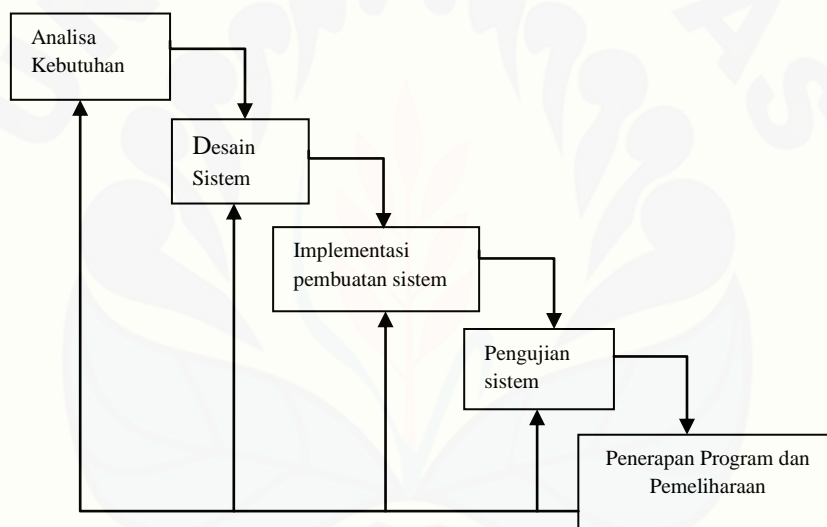
K2 = kelompok 2

tK1 = tahun dasar kelompok 1

tK2 = tahun dasar kelompok 2

## 2.5 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 tahap merupakan *input* bagi tahap berikutnya. Berikut merupakan tahap-tahap perancangan model *waterfall* :



Gambar 2.1 Model 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.

## 2. Desain Sistem

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

## 3. Implementasi Pembuatan Sistem

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

## 4. Pengujian Sistem

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

## 5. Pemeliharaan

Pemeliharaan suatu software sangat diperlukan, termasuk di dalamnya adalah pengembangan, karena *software* yang dibuat tidak selamanya 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 pemrogram yang keluar dari tim pengembang, hal ini menguntungkan bagi pihak pengembang karena proses pengembangan perangkat lunak tetap dapat dilanjutkan tanpa bergantung pada pemrogram 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. Disilain, 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 dilakukan 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.

#### 3.1 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.

##### 3.1.1 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

###### a. Studi literatur

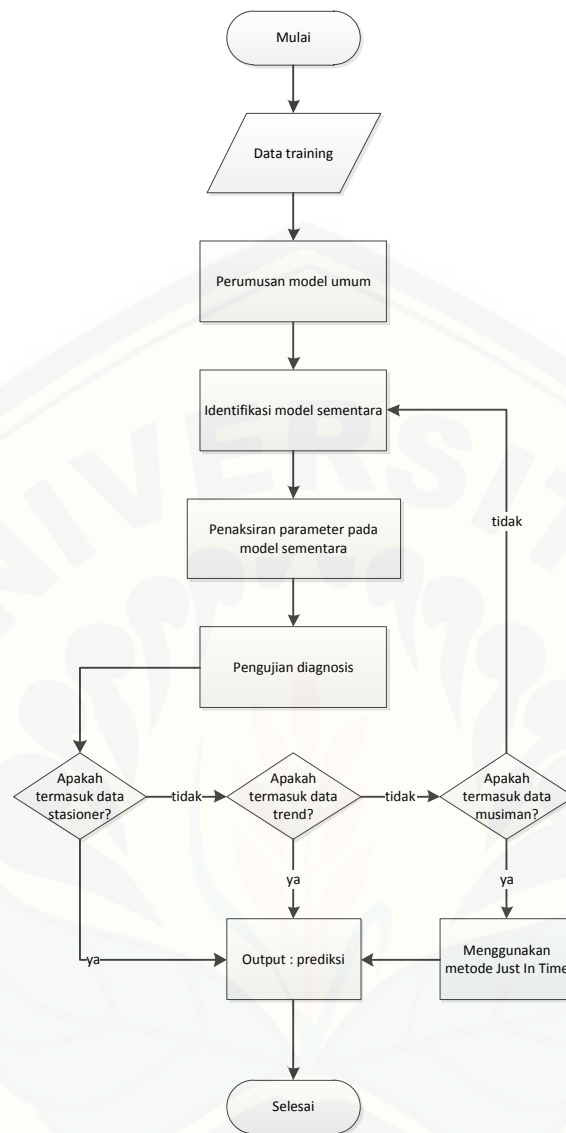
Studi literatur digunakan untuk mendukung teori-teori yang berkaitan dengan masalah yang akan dibahas. Studi literatur dalam penelitian ini diperoleh melalui data yang berasal dari buku, karya ilmiah, jurnal online, website dan penelitian yang dilakukan sebelumnya, dengan studi literatur tersebut peneliti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

b. 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 pengidentifikasian masalah menggunakan data training yang sudah dimasukkan sebelumnya. 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.

### 3.1.2 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 konsumen. Pada tahap awal sistem melakukan pengidentifikasian masalah menggunakan data training yang sudah diinputkan sebelumnya. 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 musiman maka data langsung diolah menggunakan metode *Just In Time* dan setelah itu akan muncul hasil dari perhitungan data menggunakan *Just In Time*.

## 3.2 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 :

### 3.2.1 *Business Process*

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

### 3.2.2 *Usecase Diagram*

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

### 3.2.3 *Scenario Diagram*

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

### 3.2.4 *Activity Diagram*

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

### 3.2.5 *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.

### 3.2.6 *Class Diagram*

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

### 3.2.7 *Entity Relationship Diagram*

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

## 3.3 Implementasi Pembuatan Aplikasi

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

### 3.4 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 pemrograman 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 :

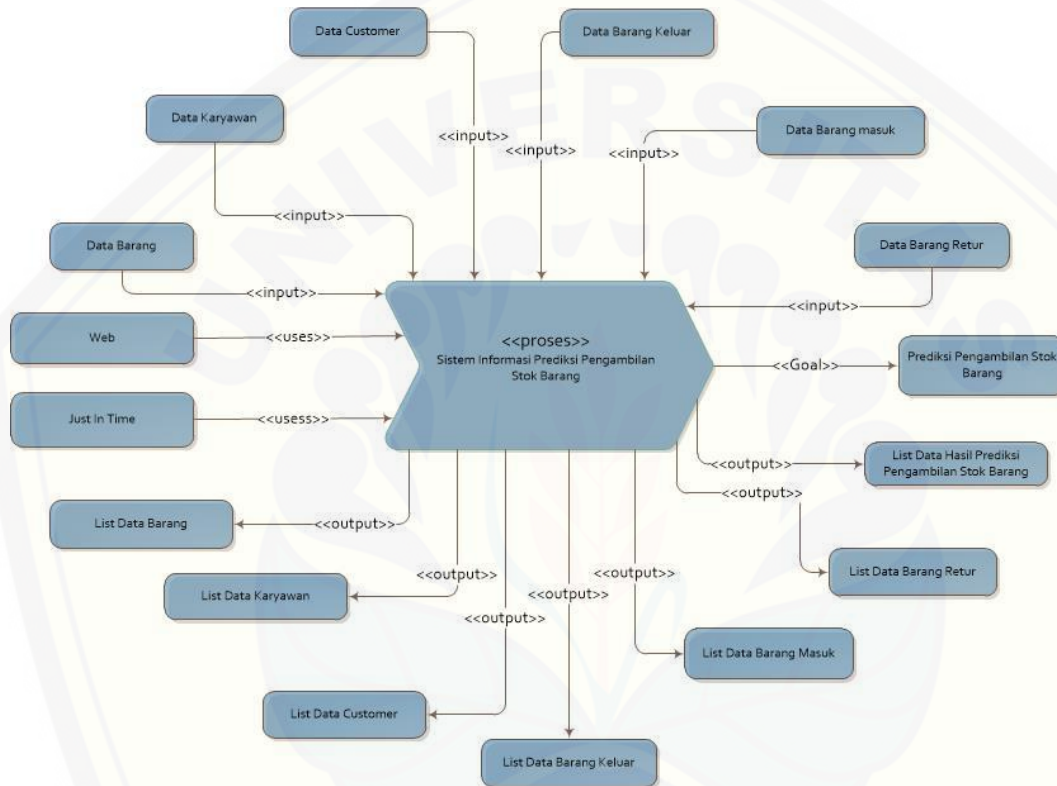
1. Aplikasi dapat menyimpan dan melakukan tambah, dan edit data barang
2. Aplikasi dapat melakukan tambah, edit dan menyimpan data customer
3. Aplikasi dapat melakukan tambah, edit dan menyimpan data karyawan
4. Aplikasi dapat melakukan tambah dan menyimpan data barang masuk
5. Aplikasi dapat melakukan 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 Business Process

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



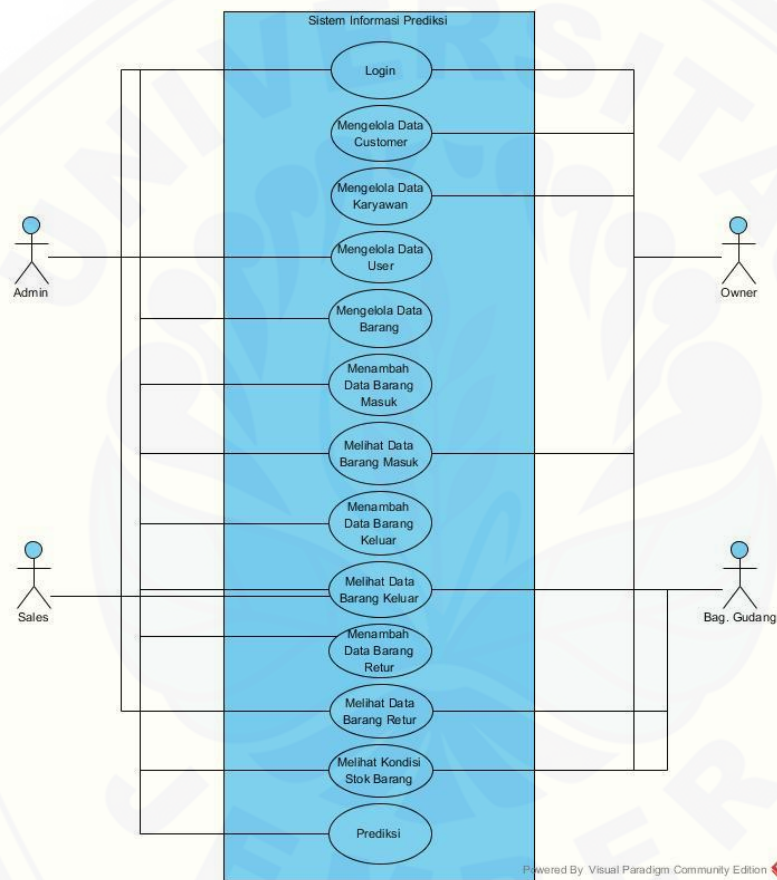
Gambar 4.1 *Business Process* Sistem Informasi Prediksi Pengambilan Stok Barang

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 pengambilan stok barang PT. Irawan Djaja Agung :



Gambar 4.2 Usecase Diagram

Pada *usecase diagram* tersebut terdapat empat aktor yang dapat mengakses aplikasi prediksi stok barang. Empat aktor tersebut adalah super admin (admin),

pemilik (owner), bagian gudang, dan sales. Deskripsi aktor serta *usecase* akan dijelaskan pada tabel 4.1 berikut :

Tabel 4.1 Definisi Aktor *Usecase*

No.	Aktor	Definisi Tugas
1.	Super Admin (Admin)	<p>Admin dapat melakukan :</p> <ol style="list-style-type: none"> <li>a. Login</li> <li>b. Mengelola data user</li> <li>c. Mengelola data barang</li> <li>d. Menambah data barang masuk</li> <li>e. Melihat data barang masuk</li> <li>f. Menambah data barang keluar</li> <li>g. Melihat data barang keluar</li> <li>h. Menambah data barang retur</li> <li>i. Melihat data barang retur</li> <li>j. Melihat kondisi stok barang</li> <li>k. Melakukan prediksi pengambilan stok barang</li> </ol>
2.	Pemilik (Owner)	<p>User dapat melakukan :</p> <ol style="list-style-type: none"> <li>a. Login</li> <li>b. Mengelola data karyawan</li> <li>c. Mengelola data customer</li> <li>d. Melihat data barang masuk</li> <li>e. Melihat data barang keluar</li> <li>f. Melihat data barang retur</li> <li>g. Melihat kondisi stok barang</li> </ol>
3.	Bagian Gudang	<p>User dapat melakukan :</p> <ol style="list-style-type: none"> <li>a. Login</li> <li>b. Melihat data barang keluar</li> </ol>



- 
- c. Melihat data barang retur
  - d. Melihat kondisi stok barang
- 

User dapat melakukan :

- a. Login
  - b. Melihat data barang keluar
  - c. Melihat data barang retur
  - d. Melihat kondisi stok barang
- 
4. Sales
- 

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 customer.
4.	Mengelola Data User	Data User merupakan fitur yang berisi data-data user yang memiliki hak akses pada aplikasi ini. Aktor dapat melihat, menambah, serta mengubah data user
5.	Mengelola Data Barang	InputBarang merupakan fitur yang berisi data barang yang berada dalam gudang. Aktor dapat melihat, menambah, serta

		mengubah data barang
6.	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
7.	Menambah Data Barang Keluar	Input Barang Keluar merupakan fitur yang berisi data transaksi barang keluar dari gudang. Aktor dapat menambah data barang keluar
8.	Melihat Data Barang Masuk	Data Barang Masuk merupakan fitur yang berisi data transaksi barang masuk ke gudang. Aktor dapat melakukan aksi melihat list tabel data barang masuk
9.	Menambah Data Barang Masuk	Input Barang Masuk merupakan fitur yang berisi data transaksi barang masuk ke gudang. Aktor dapat menambah data barang masuk.
10.	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.
11.	Menambah Data Barang Retur	Input Barang Retur merupakan fitur yang berisi data transaksi barang retur. Aktor dapat menambah data barang retur
12.	Melihat Kondisi Stok	Kondisi Barang merupakan fitur yang berisi kondisi stok barang yang terdapat pada

	Barang	gudang. Aktor dapat melakukan aksi melihat list tabel data kondisi barang.
13.	Prediksi Stok Barang	Merupakan fitur yang digunakan untuk memprediksi pengambilan stok barang terbaik untuk periode selanjutnya. Aktor dapat melakukan prediksi pengambilan stok barang.

#### 4.4 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.

##### 4.4.1 Scenario Login

*Scenario login* merupakan interaksi antara aktor dengan aplikasi untuk dapat menggunakan fitur *login*, sehingga 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 4.3 sedangkan *scenario login* ke aplikasi oleh aktor owner dapat dilihat pada tabel 4.4, sedangkan *scenario login* ke aplikasi oleh aktor bagian gudang dapat dilihat pada tabel 4.5 dan *scenario login* ke aplikasi oleh aktor sales dapat dilihat pada tabel 4.6

Tabel 4.3 Scenario Login Admin

**Scenario Login Admin**

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

Tabel 4.4 Scenario Login Owner

**Scenario Login Owner**

<b>Nama Usecase</b>	<i>Login</i>
<b>Aktor</b>	Owner
<b>Pra-Kondisi</b>	Owner membuka halaman aplikasi
<b>Pasca-Kondisi</b>	Owner berhasil <i>login</i>
<b>Aktor</b>	<b>Aplikasi</b>

<b>Skenario Utama</b> <b>“Login”</b>	
1. Membuka aplikasi prediksi pengambilan stok barang	
	2. Menampilkan halaman <i>login</i> untuk masuk ke aplikasi
3. Memasukkan <i>username</i> dan <i>password</i>	
4. Klik tombol “ <i>sign in</i> ”	
	5. Mengecek data <i>user</i> ke <i>database</i>
	6. Menampilkan halaman utama owner
<b>Skenario Alternatif</b> <b>“Username dan Password Salah”</b>	
4a. Klik tombol “ <i>sign in</i> ”	
	5a. Mengecek data <i>user</i> ke <i>database</i>
	6a. Menampilkan pesan <i>error</i> “ <i>username dan passwordsalah</i> ”

Tabel 4.5 *Scenario Login* Bagian Gudang  
**Scenario Login Bagian Gudang**

<b>Nama Usecase</b>	<i>Login</i>
<b>Aktor</b>	Bagian Gudang
<b>Pra-Kondisi</b>	Bagian Gudang membuka halaman aplikasi
<b>Pasca-Kondisi</b>	Bagian Gudang berhasil <i>login</i>
<b>Aktor</b>	<b>Aplikasi</b>
<b>Skenario Utama</b> <b>“Login”</b>	
1. Membuka aplikasi prediksi pengambilan stok barang	
	2. Menampilkan halaman <i>login</i> untuk masuk ke aplikasi
3. Memasukkan <i>username</i> dan <i>password</i>	
4. Klik tombol “ <i>sign in</i> ”	

	5. Mengecek data <i>user</i> ke <i>database</i>
	6. Menampilkan halaman utama bagian gudang
<b>Skenario Alternatif</b>	
<b>“Username dan Password Salah”</b>	
4a. Klik tombol “ <i>sign in</i> ”	
	5a. Mengecek data <i>user</i> ke <i>database</i>
	6a. Menampilkan pesan <i>error</i> “ <i>username</i> dan <i>password</i> salah”

Tabel 4.6 Scenario LoginSales  
Scenario Login Sales

<b>Nama Usecase</b>	<i>Login</i>
<b>Aktor</b>	Sales
<b>Pra-Kondisi</b>	Sales membuka halaman aplikasi
<b>Pasca-Kondisi</b>	Sales berhasil <i>login</i>
<b>Aktor</b>	<b>Aplikasi</b>
<b>Skenario Utama</b>	
<b>“Login”</b>	
1. Membuka aplikasi prediksi pengambilan stok barang	
	2. Menampilkan halaman <i>login</i> untuk masuk ke aplikasi
3. Memasukkan <i>username</i> dan <i>password</i>	
4. Klik tombol “ <i>sign in</i> ”	
	5. Mengecek data <i>user</i> ke <i>database</i>
	6. Menampilkan halaman utama sales
<b>Skenario Alternatif</b>	
<b>“Username dan Password Salah”</b>	
4a. Klik tombol “ <i>sign in</i> ”	
	5a. Mengecek data <i>user</i> ke <i>database</i>
	6a. Menampilkan pesan

---

*error*“username dan password salah”

---

#### 4.4.2 *Scenario* Mengelola Data Karyawan

*Scenario* mengelola data karyawan merupakan interaksi antara aktor dengan aplikasi pada fitur karyawan, aktor yang menjalankan fitur ini adalah 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.

##### 2. *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, 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 karyawan tanpa menyimpan data yang sudah diinputkan. *Scenario* ubah data karyawan dapat dilihat pada lampiran A1.

#### 4.4.3 *Scenario* Mengelola Data Customer

*Scenario* mengelola data customer merupakan interaksi antara aktor dengan aplikasi pada fitur customer, aktor yang menjalankan fitur ini adalah 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 customer dapat dilihat pada lampiran A1.

2. *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.

#### 4.4.4 *Scenario* Mengelola Data User

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

1. *Scenario* Tambah Data User

*Scenario* tambah data customer menjelaskan alur proses untuk menambah data user, dalam *scenario* ini terdapat *scenario* normal dan *scenario* alternatif. *Scenario* normal menjelaskan alur keberhasilan aktor dalam melakukan tambah data user, 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 user tanpa menyimpan data yang sudah diinputkan. *Scenario* tambah data user dapat dilihat pada lampiran A1.

## 2. *Scenario* Ubah Data User

*Scenario* ubah data user menjelaskan alur proses untuk mengubah data user, dalam *scenario* ini terdapat *scenario* normal dan *scenario* alternatif. *Scenario* normal menjelaskan alur keberhasilan aktor dalam melakukan ubah data user, 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 user tanpa menyimpan data yang sudah diinputkan. *Scenario* ubah data user dapat dilihat pada lampiran A1.

### 4.4.5 *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.

#### 2. *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, 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 barang tanpa menyimpan data yang sudah diinputkan. *Scenario* ubah data barang dapat dilihat pada lampiran A1.

#### 4.4.6 *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.

#### 4.4.7 *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 admin dan 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.

#### 4.4.8 *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

sales. 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.

#### 4.4.9 *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 admin dan 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.

#### 4.4.10 *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 sales. 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.

#### 4.4.11 *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.

#### 4.4.12 *Scenario* Melihat Kondisi Stok Barang

*Scenario* melihat kondisi stok barang merupakan interaksi antara aktor dengan aplikasi pada fitur barang retur, aktor 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.

#### 4.4.13 *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 apabila admin melakukan prediksi dengan memilih periode yang data 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 4.7

Tabel 4.7 *Scenario* Prediksi Pengambilan Stok Barang

#### ***Scenario* Prediksi Pengambilan Stok Barang**

<b>Nama Usecase</b>	Prediksi Stok Barang
<b>Aktor</b>	Admin
<b>Pra-Kondisi</b>	Admin telah berhasil login
<b>Pasca-Kondisi</b>	Admin berhasil memprediksi pengambilan stok barang
<b>Aktor</b>	<b>Sistem</b>
<b>Skenario Utama</b>	

<b>“Prediksi Pengambilan Stok Barang”</b>	
1. Klik menu “Prediksi”	
	2. Menampilkan form prediksi pengambilan stok barang
3. Memilih barang apa dan tahun ke berapa yang ingin diprediksi	
4. Klik tombol “prediksi”	
	5. Menampilkan halaman detail prediksi
6. Klik submenu “Indeks Waktu”	
	7. Menampilkan tabel hasil perhitungan indeks waktu perkiraan
8. Klik submenu “Nilai Perkiraan”	
	9. Menampilkan tabel hasil perhitungan nilai perkiraan.
10. Klik submenu “Hasil Perhitungan Perkiraan Stok Minimal”	
	11. Menampilkan tabel hasil perhitungan prediksi stok minimal tahun berikutnya

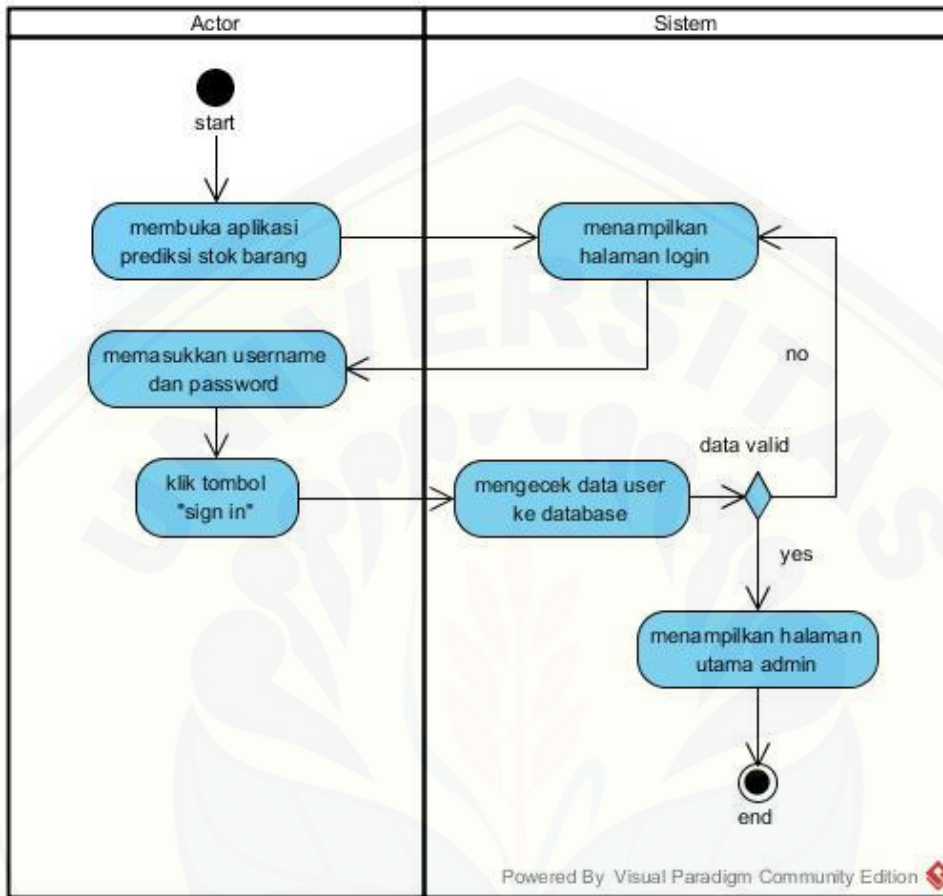
#### 4.5 ActivityDiagram

*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.

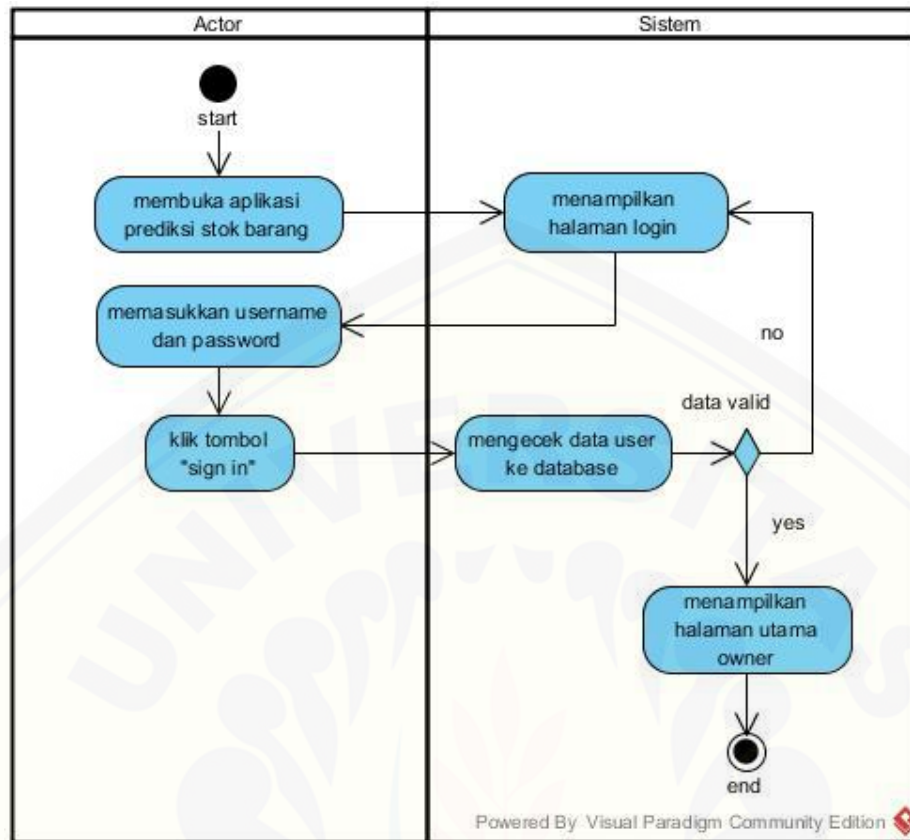
##### 4.5.1 Activity Diagram Login

*Activity diagram login* menggambarkan alur aktifitas ketika aktor akan mengakses aplikasi, *login* dilakukan saat aktor memiliki *username* dan *password*

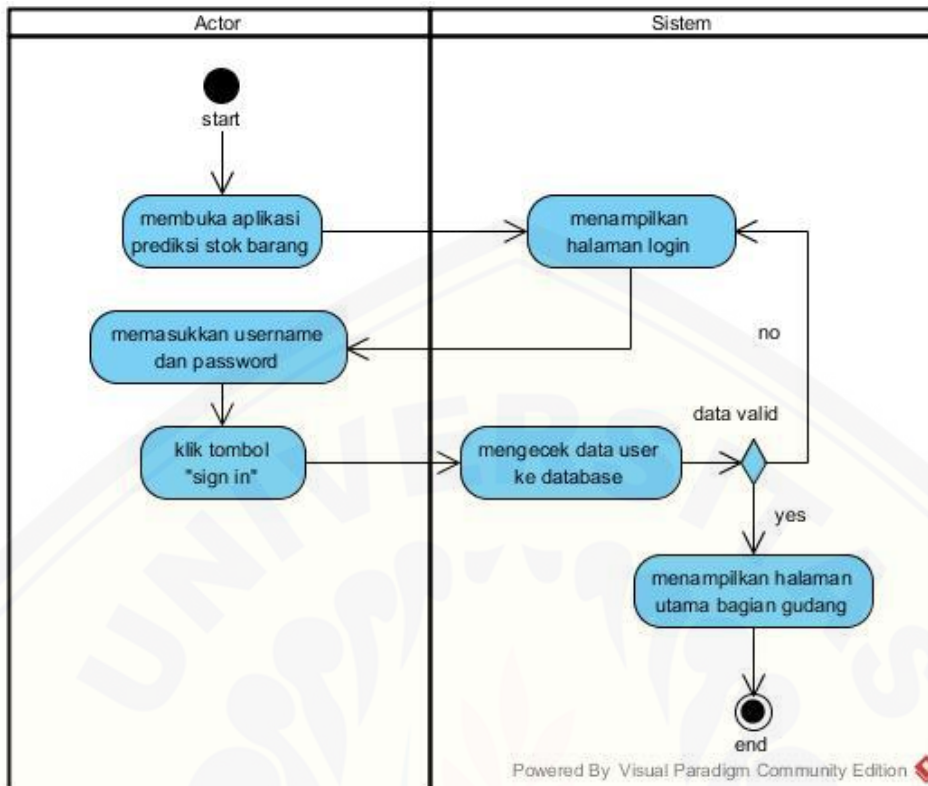
yang sesuai dan sudah terdaftar. Jika berhasil *login*, maka aktor akan memasuki tampilan utama menu utama. *Activity diagram* dapat dilihat pada gambar 4.3



Gambar 4.3 Activity Diagram Login Admin

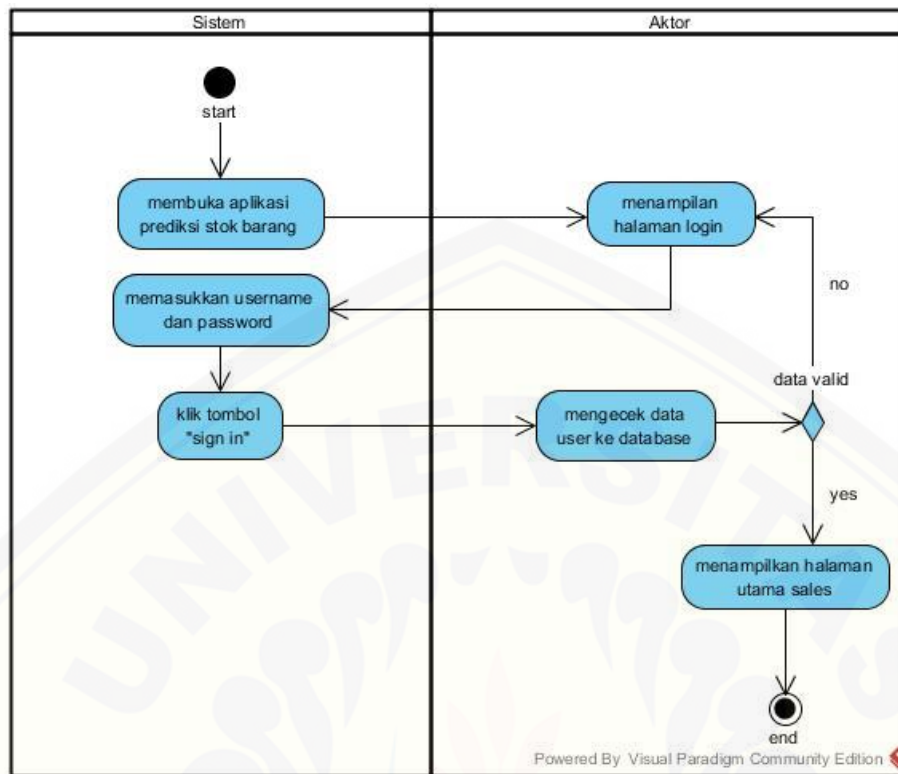


Gambar 4.4 Activity Diagram Login Owner



Gambar 4.5 Activity Diagram Login Bagian Gudang





Gambar 4.6 Activity Diagram Login Sales

#### 4.5.2 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 owner.

##### 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.

## 2. *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.

### 4.5.3 *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 owner.

#### 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.

#### 2. *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.

#### 4.5.4 *Activity Diagram* Mengelola Data User

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

1. *Activity Diagram* Tambah Data User

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

*Activity diagram* tambah data user dapat dilihat pada lampiran A2.

2. *Activity Diagram* Ubah Data User

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

*Activity diagram* ubah data user dapat dilihat pada lampiran A2.

#### 4.5.5 *Activity Diagram* 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.

## 2. *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.

### 4.5.6 *Activity Diagram* Menambah Data Barang Masuk

*Activity diagram* menambah data barang masuk menggambarkan alur aktifitas aktor untuk melakukan tambah data barang masuk. Aktor yang dapat mengakses fitur ini adalah admin. *Activity 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.

### 4.5.7 *Activity Diagram* Melihat Data Barang Masuk

*Activity diagram* melihat data barang masuk menggambarkan alur aktifitas aktor untuk melihat data 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.

### 4.5.8 *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.

#### 4.5.9 *Activity Diagram* Melihat Data Barang Keluar

*Activity diagram* melihat data barang keluar menggambarkan alur aktifitas aktor untuk melihat data barang keluar. Aktor yang dapat mengakses fitur ini adalah admin, owner, bagian gudang, dan sales. 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.

#### 4.5.10 *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 data barang retur. *Activity diagram* tambah data barang retur dapat dilihat pada lampiran A2.

#### 4.5.11 *Activity Diagram* Melihat Data Barang Retur

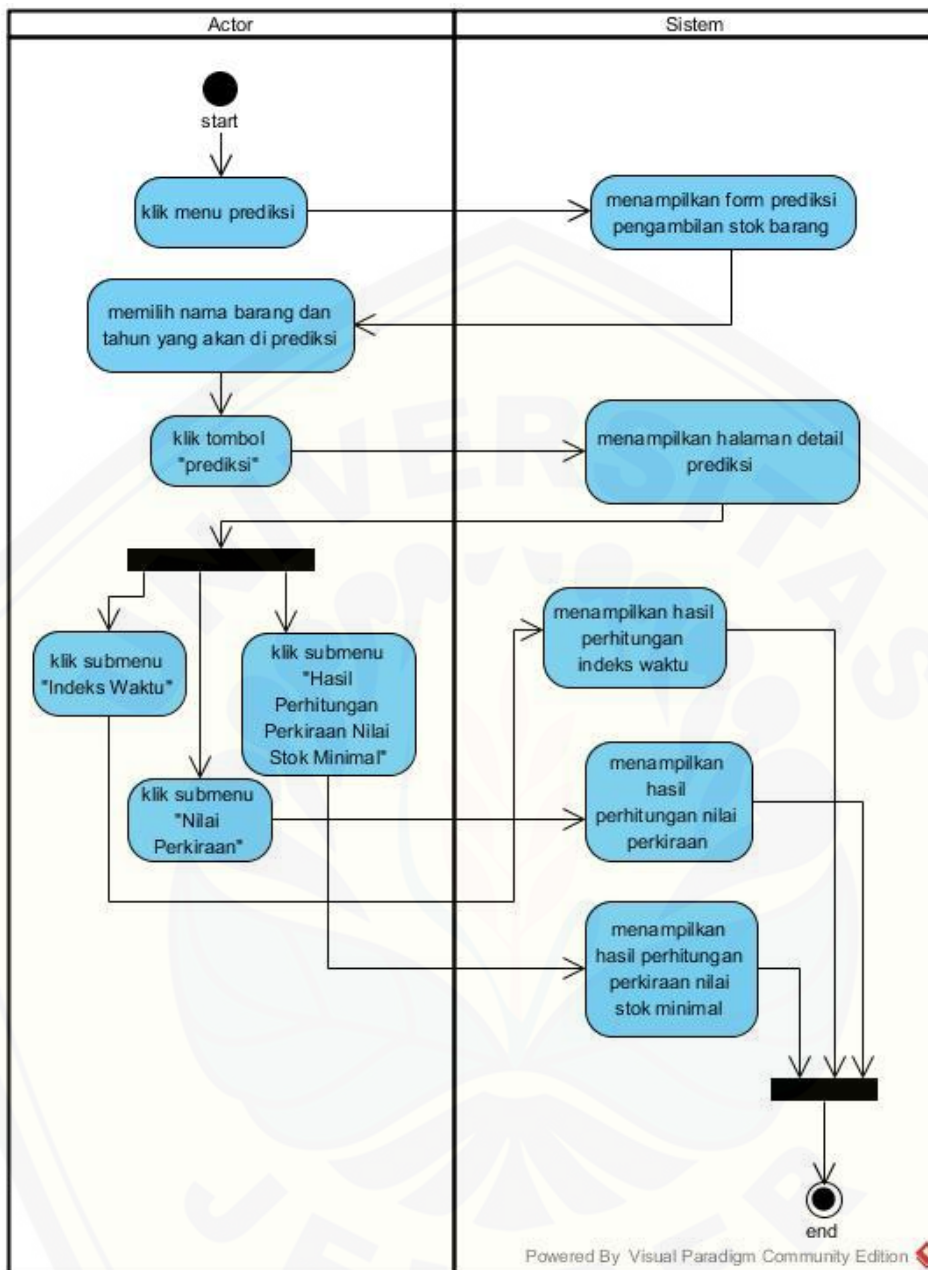
*Activity diagram* melihat data barang retur menggambarkan alur aktifitas aktor untuk melihat data barang retur. Aktor yang dapat mengakses fitur ini adalah admin, owner, bagian gudang, sales. Aktifitas aktor dan aplikasi saling berhubungan dalam suatu aktifitas atau *event*, aplikasi memberikan respon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

#### 4.5.12 *Activity Diagram* Melihat Kondisi Stok Barang

*Activity diagram* melihat kondisi stok barang menggambarkan alur aktifitas aktor untuk melihat kondisi stok barang. Aktor yang dapat mengakses fitur ini adalah admin, owner, bagian gudang, dan sales. 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.

#### 4.5.13 *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 data yang diperlukan, setelah ditekan tombol prediksi maka aplikasi akan memulai perhitungan dan selanjutnya aplikasi menampilkan hasil perhitungan beserta penjelasannya. *Activity diagram* prediksi pengambilan stok barang dapat dilihat pada gambar 4.7



Gambar 4.7 Activity Diagram Prediksi Pengambilan Stok Barang

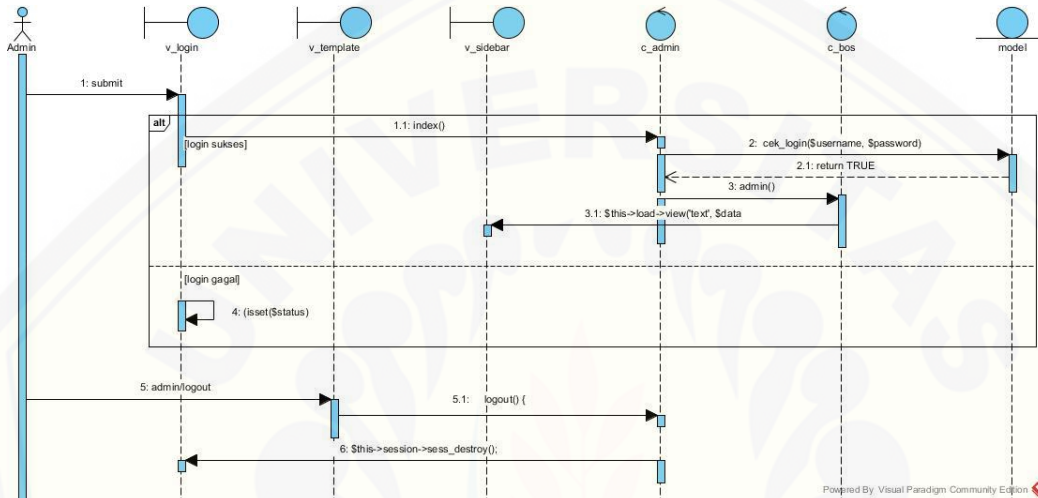
#### 4.6 Sequence Diagram

*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 *sequence diagram* aplikasi prediksi pengambilan stok barang menggunakan metode *Just In Time*.

#### 4.6.1 Sequence Diagram Login

##### 1. Sequence Diagram Login Admin



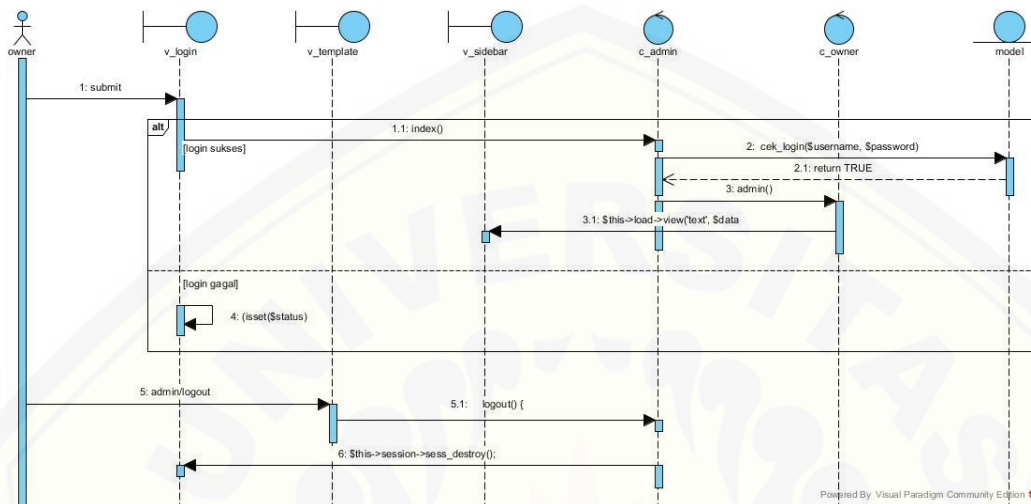
Gambar 4.8 Sequence Diagram Login Admin

Gambar 4.8 merupakan *sequence diagram login* admin yang menggambarkan alur logika pada fitur *login* dengan aktor admin. *Sequence 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 ditampilkan 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 *sequencen diagram loginadmin* 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.

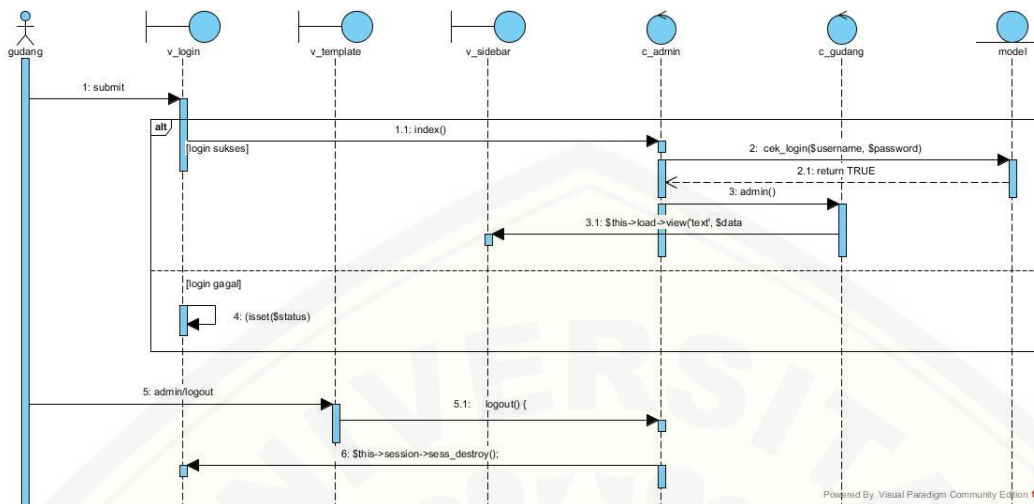
## 2. Sequence Diagram Login Owner



Gambar 4.9 Sequence Diagram Login Owner

Gambar 4.9 merupakan *sequence diagram login owner* yang menggambarkan alur logika pada fitur *login* dengan aktor *owner*. *Sequence 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 ditampilkan 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 *sequencen 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*.

### 3. Sequence Diagram Login Bagian Gudang



Gambar 4.10 Sequence Diagram Login Bagian Gudang

Gambar 4.10 merupakan *sequence diagram login* bagian gudang yang menggambarkan alur logika pada fitur *login* dengan aktor bagian gudang. *Sequence 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 ditampilkan 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 *sequencen 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*.

#### 4.6.2 *Sequence Diagram* Mengelola Data Karyawan

*Sequence 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. *Sequence Diagram* Tambah Data Karyawan

*Sequence diagram* tambah data karyawan menggambarkan alur logika tambah data karyawan pada fitur karyawan. Pada *sequence* 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 *sequence* 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 menggunakan *methodnya* untuk berinteraksi dengan *model*. Alur *method* pada *sequence* 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. *Sequence diagram* tambah data karyawan dapat dilihat pada lampiran A3.

##### 2. *Sequence Diagram* Ubah Data Karyawan

*Sequence diagram* ubah data karyawan menggambarkan alur logika ubah data karyawan pada fitur karyawan. Pada *sequence* 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 *sequence* 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 menggunakan *methodnya* untuk berinteraksi dengan *model*. Alur *method* pada

*sequence* 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. *Sequence diagram* tambah data karyawan dapat dilihat pada lampiran A3.

#### 4.6.3 *Sequence Diagram* Mengelola Data Customer

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

##### 1. *Sequence Diagram* Tambah Data Customer

*Sequence diagram* tambah data customer menggambarkan alur logika tambah data customer pada fitur customer. Pada *sequence* 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 *sequence* 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 menggunakan *methodnya* untuk berinteraksi dengan *model*. Alur *method* pada *sequence* 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. *Sequence diagram* tambah data customer dapat dilihat pada lampiran A3.

##### 2. *Sequence Diagram* Ubah Data Customer

*Sequence diagram* ubah data customer menggambarkan alur logika ubah data customer pada fitur customer. Pada *sequence* 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 *sequence* 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 menggunakan *methodnya* untuk berinteraksi dengan *model*. Alur *method* pada *sequence* 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. *Sequence diagram* tambah data customer dapat dilihat pada lampiran A3.

#### 4.6.4 *Sequence Diagram* Mengelola Data User

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

##### 3. *Sequence Diagram* Tambah Data User

*Sequence diagram* tambah data user menggambarkan alur logika tambah data user pada fitur user. Pada *sequence* ini terdapat *v\_sidebar*, *v\_user*, dan *v\_user\_add* yang berfungsi sebagai tampilan utama saat *controller* memanggil *method* untuk menampilkan data dan form data user, *controller* pada *sequence* 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 menggunakan *methodnya* untuk berinteraksi dengan *model*. Alur *method* pada *sequence* ini ketika aktor menambah data adalah dengan mengklik tambah user pada *v\_user*, *method* yang berjalan dari *v\_user* menuju *c\_bos* adalah *add\_user(\$id)*. *Method* tersebut berfungsi untuk memanggil form sehingga nantinya aktor bisa

menambah data user. *Sequence diagram* tambah data user dapat dilihat pada lampiran A3.

#### 4. *Sequence Diagram* Ubah Data User

*Sequence diagram* ubah data user menggambarkan alur logika ubah data customer pada fitur user. Pada *sequence* ini terdapat *v\_sidebar*, *v\_user*, dan *v\_add\_user* yang berfungsi sebagai tampilan utama saat *controller* memanggil *method* untuk menampilkan data dan form data user, *controller* pada *sequence* 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 menggunakan *methodnya* untuk berinteraksi dengan *model*. Alur *method* pada *sequence* ini ketika aktor mengubah data adalah dengan mengklik tambah user pada *v\_user*, *method* yang berjalan dari *v\_customer* menuju *c\_bos* adalah *add\_user(\$id)*. *Method* tersebut berfungsi untuk memanggil form sehingga nantinya aktor bisa mengubah data user. *Sequence diagram* tambah data user dapat dilihat pada lampiran A3.

#### 4.6.5 *Sequence Diagram* Mengelola Data Barang

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

##### 1. *Sequence Diagram* Tambah Data Barang

*Sequence diagram* tambah data barang menggambarkan alur logika tambah data barang pada fitur data barang. Pada *sequence* 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 *sequence* 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 menggunakan *methodnya* untuk berinteraksi dengan *model*. Alur *method* pada *sequence* 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. *Sequence diagram* tambah data barang dapat dilihat pada lampiran A3.

## 2. *Sequence Diagram* Ubah Data Barang

*Sequence diagram* ubah data barang menggambarkan alur logika ubah data barang pada fitur data barang. Pada *sequence* 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 *sequence* ini adalah *c\_bos* dan *c\_simpan*, kemudian terdapat *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 menggunakan *methodnya* untuk berinteraksi dengan *model*. Alur *method* pada *sequence* ini ketika aktor mengubah 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 mengubah data barang. *Sequence diagram* tambah data barang dapat dilihat pada lampiran A3.

### 4.6.6 *Sequence Diagram* Menambah Data Barang Masuk

*Sequence diagram* menambah data barang masuk menggambarkan alur logika pada fitur “Barang Masuk”, aktor yang mengakses fitur ini adalah admin. Aktor dapat melakukan aksi tambah data barang masuk. *Sequence diagram* tambah data barang masuk menggambarkan alur logika tambah data barang masuk pada fitur “Input

Barang Masuk”. Pada *sequence* 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 *sequence* 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 menggunakan *methodnya* untuk berinteraksi dengan *model*. Alur *method* pada *sequence* 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. *Sequence diagram* tambah data barang masuk dapat dilihat pada lampiran A3.

#### 4.6.7 *Sequence Diagram* Melihat Data Barang Masuk

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

##### 1. *Sequence Diagram* Melihat Data Barang Masuk – Admin

*Sequence diagram* melihat data barang masuk oleh aktor admin menggambarkan 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 dapat 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. *Sequence diagram* melihat data barang masuk oleh admin dapat dilihat pada lampiran A3.

## 2. *Sequence Diagram* Melihat Data Barang Masuk – Owner

*Sequence diagram* melihat data barang masuk oleh aktor owner menggambarkan 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. *Sequence diagram* melihat data barang masuk oleh owner dapat dilihat pada lampiran A3.

### 4.6.8 *Sequence Diagram* Menambah Data Barang Keluar

*Sequence diagram* menambah data barang keluar menggambarkan alur logika pada fitur “Barang Keluar”, aktor yang mengakses fitur ini adalah admin. Aktor dapat melakukan aksi tambah data barang keluar. *Sequence diagram* tambah data barang keluar menggambarkan alur logika tambah data barang keluar pada fitur “Input Barang Keluar”. Pada *sequence* 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 *sequence* 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 menggunakan *methodnya* untuk berinteraksi dengan *model*. Alur *method* pada *sequence* 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. *Sequence diagram* tambah data barang keluar dapat dilihat pada lampiran A3.

#### 4.6.9 *Sequence Diagram* Melihat Data Barang Keluar

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

##### 1. *Sequence Diagram* Melihat Data Barang Keluar – Admin

*Sequence diagram* melihat data barang keluar oleh aktor admin menggambarkan 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 dapat 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. *Sequence diagram* melihat data barang keluar oleh admin dapat dilihat pada lampiran A3.

## 2. *Sequence Diagram* Melihat Data Barang Keluar – Owner

*Sequence diagram* melihat data barang keluar oleh aktor owner menggambarkan 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 keluar yaitu pada *view* *v\_rbrg\_keluar* sehingga admin dapat melihat data-data yang ada pada fitur “Data Barang Keluar” dalam bentuk tabel. *Sequence diagram* melihat data barang keluar oleh owner dapat dilihat pada lampiran A3.

## 3. *Sequence Diagram* Melihat Data Barang Keluar – Bagian Gudang

*Sequence diagram* melihat data barang keluar oleh aktor bagian gudang menggambarkan 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 keluar yaitu pada *view* *v\_rbrg\_keluar* sehingga admin dapat melihat data-data yang ada pada fitur “Data Barang Keluar” dalam bentuk tabel. *Sequence diagram* melihat data barang keluar oleh bagian gudang dapat dilihat pada lampiran A3.

#### 4.6.10 *Sequence Diagram* Menambah Data Barang Retur

*Sequence diagram* menambah data barang retur menggambarkan alur logika pada fitur “Barang Retur”, aktor yang mengakses fitur ini adalah admin. Aktor dapat melakukan aksi tambah data barang retur. *Sequence diagram* tambah data barang retur menggambarkan alur logika tambah data barang retur pada fitur “Input Barang Retur”. Pada *sequence* 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 *sequence* 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 menggunakan *methodnya* untuk berinteraksi dengan *model*. Alur *method* pada *sequence* 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. *Sequence diagram* tambah data barang retur dapat dilihat pada lampiran A3.

#### 4.6.11 *Sequence Diagram* Melihat Data Barang Retur

*Sequence 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. *Sequence Diagram* Melihat Data Barang Retur – Admin

*Sequence diagram* melihat data barang retur oleh aktor admin menggambarkan 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. *Sequence diagram* melihat data barang retur oleh admin dapat dilihat pada lampiran A3.

## 2. *Sequence Diagram* Melihat Data Barang Retur – Owner

*Sequence diagram* melihat data barang retur oleh aktor owner menggambarkan 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 dapat 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. *Sequence diagram* melihat data barang retur oleh owner dapat dilihat pada lampiran A3.

## 3. *Sequence Diagram* Melihat Data Barang Retur – Bagian Gudang

*Sequence diagram* melihat data barang retur oleh aktor bagian gudang menggambarkan 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. *Sequence diagram* melihat data barang retur oleh bagian gudang dapat dilihat pada lampiran A3.

#### 4.6.12 *Sequence Diagram* Melihat Kondisi Stok Barang

*Sequence 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. *Sequence Diagram* Melihat Kondisi Stok Barang – Admin

*Sequence 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\_stokbrg* sehingga admin dapat melihat data-data yang ada pada fitur “Kondisi Barang” dalam bentuk tabel dalam bentuk tabel. *Sequence diagram* melihat kondisi stok barang oleh admin dapat dilihat pada lampiran A3.

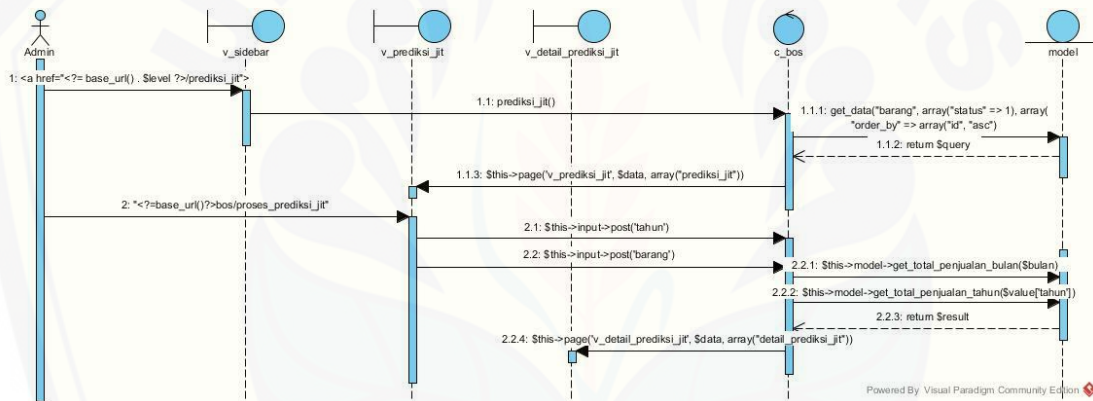
##### 2. *Sequence Diagram* Melihat Data Kondisi Stok Barang – Owner

*Sequence 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. *Sequence diagram* melihat kondisi stok barang oleh owner dapat dilihat pada lampiran A3.

### 3. Sequence Diagram Melihat Kondisi Stok Barang – Bagian Gudang

*Sequence 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. *Sequence diagram* melihat kondisi stok barang oleh bagian gudang dapat dilihat pada lampiran A3.

#### 4.6.13 Sequence Diagram Prediksi Pengambilan Stok Barang

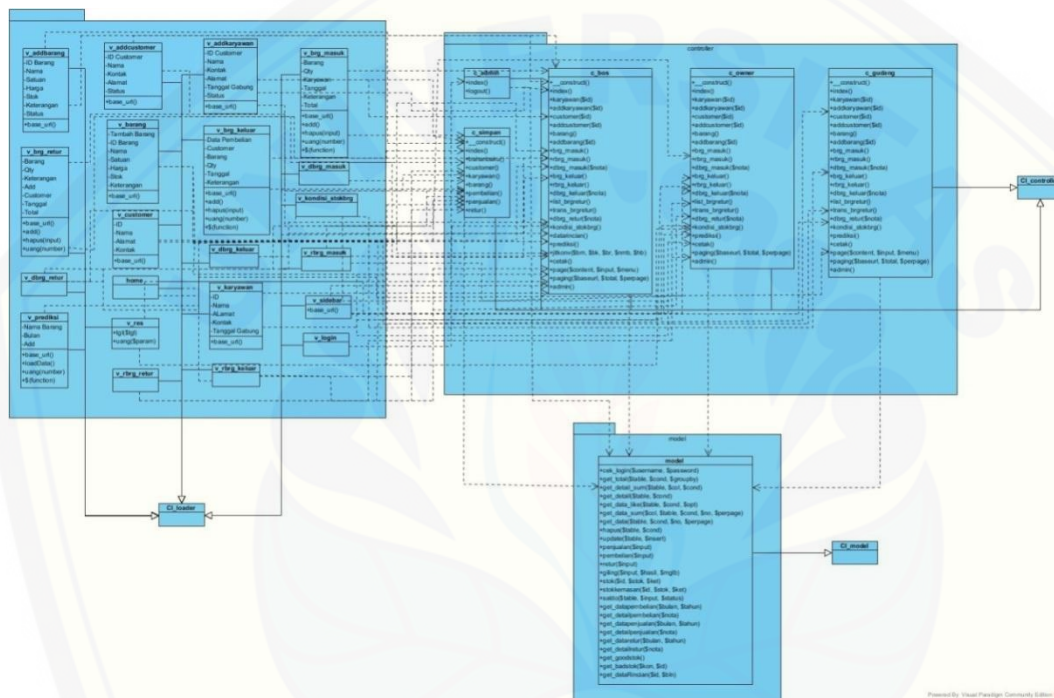


Gambar 4.11 *Sequence Diagram* Prediksi Pengambilan Stok Barang

Gambar 4.11 merupakan *sequence diagram* prediksi stok barang yang menggambarkan alur logika untuk memprediksi pengambilan barang selanjutnya pada fitur “Prediksi” oleh admin. Pada *sequence* 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\_detail\_prediksi\_jit* sehingga admin dapat melihat hasil perhitungan data yang ada pada fitur “Prediksi” dalam bentuk penjelasan singkat.

#### 4.7 Class Diagram

Setelah melalui tahap pembuatan desain dengan *sequence 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 4.12 Class Diagram Sistem Informasi Prediksi Pengambilan Stok Barang

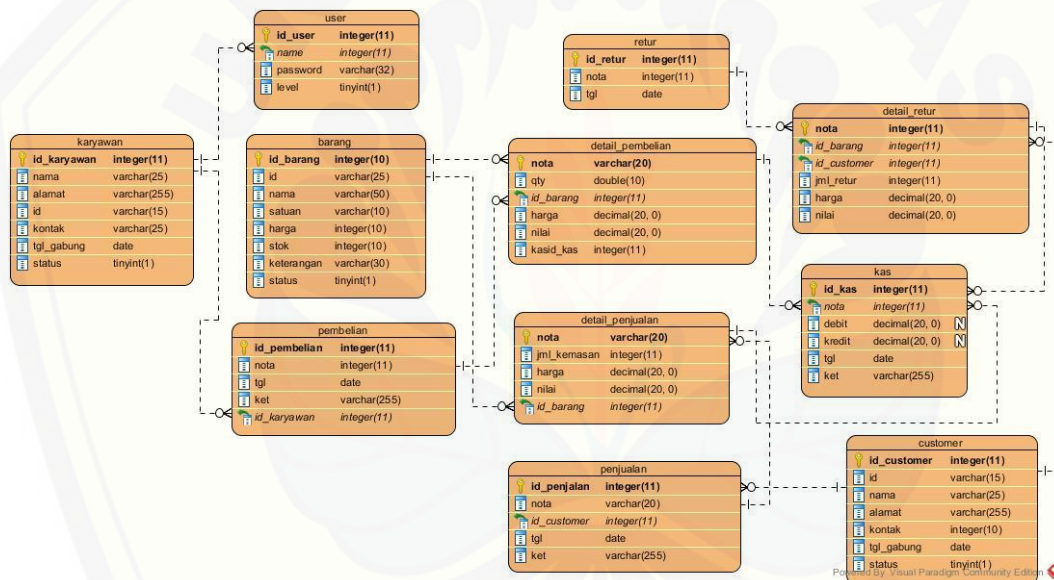
Gambar 4.12 merupakan *class diagram* aplikasi prediksi pengambilan stok barang, dalam *class* tersebut terdapat *view*, *model*, dan *controller*. *View* dalam *class* tersebut terhubung ke *controller* sesuai dengan aktor yang mengakses. Untuk *view* yang terhubung pada *controller* `c_bos` maka tabel 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*.

#### 4.8 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 4.13 Berikut



Gambar 4.13 ERD Prediksi Pengambilan Stok Barang

#### 4.9 Implementasi Perancangan

Setelah tahap desain perancangan selesai, tahap selanjutnya dalam penelitian ini adalah tahap pengimplementasian desain perancangan ke dalam bahasa pemrograman. Bahasa pemrograman 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).

#### 4.10 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*.

##### 4.10.1 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

```

435
436 public function prediksi_jit() {
437     $data['title'] = "Prediksi";
438
439     $get_tahun = $this->db->query("SELECT MAX( DISTINCT YEAR(`tgl` ) as tahun FROM `penjualan`");
440     $tahun = $get_tahun->row()->tahun + 1;
441
442     $data['tahun']     = $tahun;
443     $data['idbrg']    = $idbrg[0];
444     $data['barang'] = $this->model->get_data("barang", array("status" => 1), array(
445         "order_by" => array("id", "asc"),
446     ));
447     $this->page('v_prediksi_jit', $data, array("prediksi_jit"));
448 }
449

```

Gambar 4.14 Method prediksi\_jit()

Gambar 4.14 merupakan kode *method* prediksi() dimana *method* ini untuk menampilkan nama barang dan tahun yang akan diprediksi.

```

449 public function proses_prediksi_jit() {
451     $tahun_prediksi = $this->input->post('tahun');
452     $idbrg = explode(";", $this->input->post('barang'));
453     $id_barang = $idbrg[0];
454     $nama_barang = $this->model->get_barang_by_id($id_barang)->nama;
455     $harga_barang = $this->model->get_barang_by_id($id_barang)->harga;
456     $data['idbrg'] = $id_barang;
457
458     $total_penjualan = $this->model->get_total_penjualan()->total;
459     for ($i=1; $i < 13; $i++) {
460         $bulan = "0".$i;
461         $total_penjualan_bulan[] = $this->model->get_total_penjualan_bulan($bulan)->total;
462     }
463
464     $bulan = array('Januari', 'Februari', 'Maret', 'April', 'Mei', 'Juni', 'Juli', 'Agustus', 'September', 'Oktober', 'Nopember', 'Des');
465     $tahun_penjualan = $this->model->get_tahun_penjualan();
466
467     foreach ($tahun_penjualan as $key => $value) {
468         $total_penjualan_tahun[$value['tahun']] = $this->model->get_total_penjualan_tahun($value['tahun'])->total;
469     }
470     $count_tahun = count($tahun_penjualan);
471     for ($i=0; $i < $count_tahun; $i++) {
472         for ($j=0; $j < $count_tahun; $j++) {
473             if ($i==$j){
474                 $nilaix[$i][$j] = 0;
475             } else {
476                 $nilaix[$i][$j] = $i-$j;
477             }
478         }
479     }
480
481     $data = array(
482         'tahun' => $tahun_prediksi ,
483         'nilaix' => $nilaix ,
484         'bulan' => $bulan,
485         'total_penjualan_bulan' => $total_penjualan_bulan,
486         'total_penjualan_tahun' => $total_penjualan_tahun,
487         'total_penjualan' => $total_penjualan ,
488         'tahun_penjualan' => $tahun_penjualan ,
489         'harga_barang' => $harga_barang,
490     );
491
492     $data['title'] = "Detail Prediksi <br>".$nama_barang." Tahun ".$tahun_prediksi;
493     $this->page('v_detail_prediksi_jit', $data, array("detail_prediksi_jit"));
494 }
495

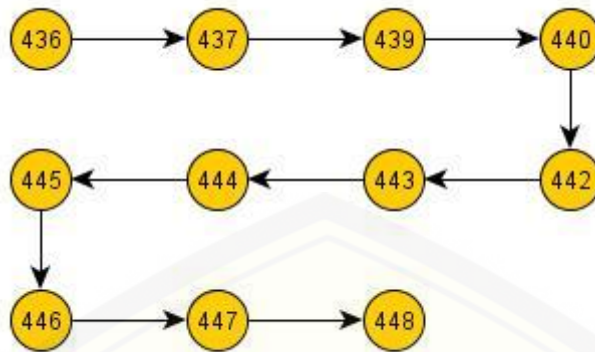
```

Gambar 4.15 Method proses\_prediksi\_jit()

Gambar 4.15 merupakan kode *method* prediksi() dimana *method* ini adalah implementasi metode *Just In Time* yang digunakan untuk memprediksi jumlah pengambilan stok ke dalam bahasa pemrograman. Pada code tersebut tertulis alur perhitungan data hingga penampilan hasil perhitungan.

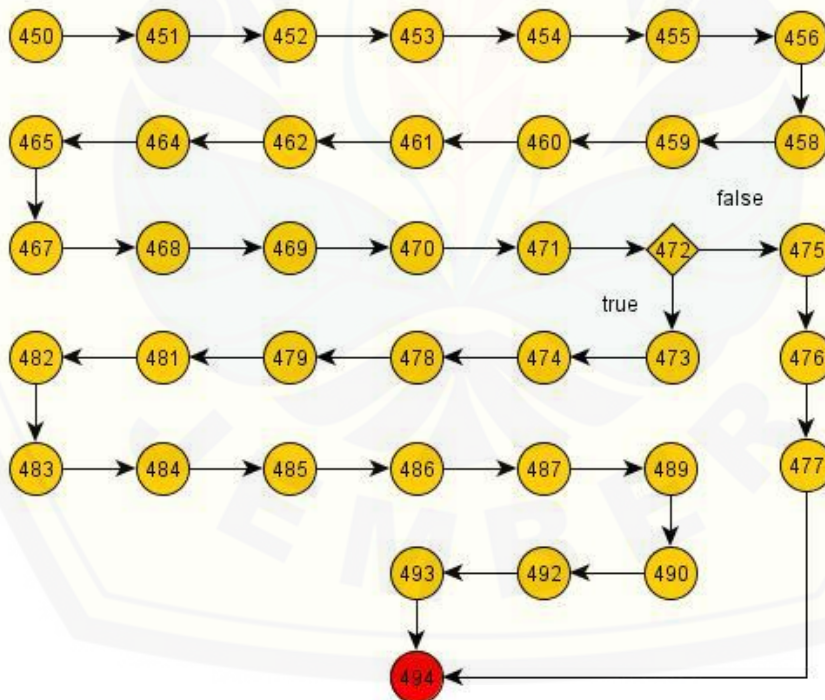
## 2. Diagram Alir

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 4.16 Diagram Alir *Method* prediksi\_jit()

Gambar 4.16 merupakan diagram alir *method* prediksi\_jit(). Kode program tersebut dimulai dari baris ke 436 kemudian diteruskan sampai baris ke 448. Karena pada *method* ini tidak ada *looping* atau perulangan maka gambar atau alur kodenya lurus.



Gambar 4.17 Diagram Alir *Method* proses\_prediksi\_jit()

Gambar 4.17 merupakan diagram alir *method* proses\_prediksi\_jit(). Kode program tersebut dimulai pada baris ke 450 sampai baris ke 472. Pada baris 472 terjadi percabangan, pada kode di baris 472 jika tahun penjualan yang dimasukkan benar maka kode akan diteruskan ke baris 494, kemudian ke baris 474 sampai 479 (end). Jika tahun penjualan yg dimasukkan salah maka kode akan diteruskan ke baris 475, kemudian ke baris 476 sampai 494 (end).

### 3. 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 :

$$\begin{aligned} \text{Method prediksi\_jit()} & : V(G) = \text{Edge} - \text{Node} + 2 \\ & = 10 - 11 + 2 = 1 \\ \text{Method proses\_prediksi\_jit()} & : V(G) = \text{Edge} - \text{Node} + 2 \\ & = 39 - 39 + 2 = 2 \end{aligned}$$

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 kompleks pula sebuah program tersebut.

Dari hasil perhitungan diatas dapat diketahui bahwa yang mempunyai path hanya satu baris adalah *method* prediksi\_jit() dan path yang mempunyai dua baris adalah *method* proses\_prediksi\_jit().

### 4. Basis Set

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

$$\begin{aligned} \text{Method prediksi\_jit()} & : 436 - 437 - 439 - 440 - 442 - 443 - 444 - \\ & 445 - 446 - 447 - 449 \end{aligned}$$

*Method* proses\_prediksi\_jit()      450 – 451 – 452 – 453 – 454 – 455 – 456 –  
 458 – 459 – 460 – 461 – 462 – 464 – 465 –  
 467 – 468 – 469 – 470 – 471 – 472 – 473 –  
 :  
 474 – 478 – 479 – 481 – 482 – 483 – 484 –  
 485 – 486 – 487 – 489 – 490 – 492 – 493 –  
 494 (Path 1)  
 450 – 451 – 452 – 453 – 454 – 455 – 456 –  
 458 – 459 – 460 – 461 – 462 – 464 – 465 –  
 467 – 468 – 469 – 470 – 471 – 472 – 475 –  
 476 – 477 – 494 (Path 2)

### 5. 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 4.8 *Test Case* Fitur Prediksi

<b>Method prediksi_jit()</b>	
<b>Test Case</b>	Menampilkan form prediksi
<b>Target yang diharapkan</b>	Proses menampilkan form prediksi
<b>Hasil Pengujian</b>	Benar
<b>Path</b>	436 – 437 – 439 – 440 – 442 – 443 – 444 – 445 – 446 – 447 – 449
<b>Method proses_prediksi_jit()</b>	
<b>Test Case</b>	Jika data sesuai
<b>Target yang diharapkan</b>	Menampilkan hasil perhitungan prediksi
<b>Hasil Pengujian</b>	Benar
<b>Path</b>	450 – 451 – 452 – 453 – 454 – 455 – 456 – 458 – 459 – 460 – 461 – 462 – 464 – 465 – 467 – 468 – 469 – 470 – 471 – 472 – 473 – 474 – 478 – 479 – 481 – 482 – 483 – 484 – 485 – 486 – 487 – 489 – 490 – 492 – 493 – 494 (Path 1)
<b>Method proses_prediksi_jit()</b>	
<b>Test Case</b>	Jika data tidak sesuai
<b>Target yang diharapkan</b>	Data tidak ditampilkan
<b>Hasil Pengujian</b>	Benar
<b>Path</b>	450 – 451 – 452 – 453 – 454 – 455 – 456 – 458 – 459 – 460 – 461 – 462 – 464 – 465 – 467 – 468 – 469 – 470 –

471 – 472 – 475 – 476 – 477 – 494 (Path 2)
--

#### 4.10.2 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 4.8 berikut

Tabel 4.9 Black Box Testing Perhitungan Prediksi Pengambilan Stok Barang

Kasus	Hasil	Ket
<i>Field</i> pada <i>form</i> tambah data terisi dengan benar, kemudian admin klik tombol “prediksi”	Menampilkan perhitungan data indeks waktu, nilai perkiraan, dan jumlah barang minimal yang akan diambil	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
<i>Field</i> pada <i>form</i> tambah data tidak terisi dengan benar, kemudian admin klik tombol “prediksi”	Menampilkan pesan error pada field	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

## 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. Tahap penyelesaian metode *Just In Time* pada penelitian ini menggunakan beberapa tahap yaitu perhitungan rata-rata penjualan, perhitungan indeks waktu, perhitungan nilai perkiraan, sehingga didapat hasil prediksi stok minimal pada tahun berikutnya. Perhitungan indeks waktu didapatkan dari data penjualan dari tahun 2013, 2014, dan 2015. Data penjualan didapatkan dari survei penelitian di PT Irawan Djaja Agung Jember, dengan jumlah data yang didapat sebanyak 1091 data. Hasil penyelesaian menggunakan metode *Just In Time* menghasilkan prediksi banyaknya jumlah stok minimal yang akan dibutuhkan pada tahun berikutnya. Hasil perhitungan yang dilakukan pada penelitian ini dapat diketahui bahwa stok minimal pada tahun 2016 sebanyak 7.171 pcs untuk setiap bulannya.
2. Sistem prediksi stok minimal pada distributor farmasi menggunakan metode *Just In Time* memiliki empat hak akses yaitu Admin, Owner, Bagian Gudang dan Sales. Admin memiliki fitur mengelola data user, mengelola data barang, menambah data barang masuk, melihat data barang masuk, menambah data barang keluar, melihat data barang keluar, menambah data barang retur, melihat kondisi stok barang, dan prediksi. Owner memiliki fitur mengelola data customer, mengelola data karyawan, melihat data barang masuk, melihat data barang keluar, melihat data barang retur, melihat kondisi stok barang. Bagian gudang memiliki fitur melihat data barang retur, melihat data barang keluar, dan melihat kondisi stok barang. Sales memiliki fitur melihat data barang keluar, melihat data



barang retur, dan melihat kondisi stok barang. Tahap pembangunan dan pengembangan sistem ini menggunakan model *waterfall* yang dapat digunakan dalam pembangunan sistem skala kecil, sehingga dokumentasi pengembangan pada penelitian ini dapat terorganisir dengan baik. Pada tahap pengujian white box nilai kompleksitas yang didapatkan adalah 2 sehingga bari kode metode pada sistem ini terbilang memiliki tingkat kompleksitas yang tidak terlalu tinggi dan dapat dilakukan proses *maintenance* dengan mudah jika terjadi kesalahan.

## 6.2 Saran

Dikernakan perkembangan teknologi yang semakin berkembang dan banyaknya aplikasi yang dapat diakses oleh smartphone maka penulis menyarankan untuk pengembangan system dapat dilakukan oleh peneliti selanjutnya dengan membuat system prediksi menggunakan platform android, iOS, atau windows phone. Selain itu penulis juga menyarankan untuk menggunakan metode lain untuk menciptakan perbandingan antar metode.

## DAFTAR PUSTAKA

- Brown, T. M. (1991). Comparison of Just In Time and Batch Manufacturing. *The Role of Performance Obstacles. 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](http://www.academia.edu/7585995/BAB_I_Model_Waterfall):  
www.academia.edu
- Efrianti, D. (2014). Pengaruh Pengendalian Persediaan Just In Time Terhadap Efisiensi Pengadaan Persediaan Bahan Baku. *Jurnal Program Studi Akuntansi, Sekolah Tinggi Ilmu Ekonomi Kesatuan*, 103.
- Hadioetomo. (2009). Pengaruh Penerapan JIT Terhadap Kinerja dan Keunggulan Kompetitif Perusahaan Manufaktur. *Kajian Akuntansi, Volume 4, Nomor 2*, 105-106.
- Ipulhe. (2014, 6 7). *Pengertian Aplikasi*. Dipetik 1 15, 2016, dari <http://www.ipulhe.com/pengertian-aplikasi/>: <http://www.ipulhe.com>
- Oktavin, W. (2012, 08 24). *Pengertian Time Series*. Retrieved 11 2016, from Wenthy's Blog: [http://wenthyoktavin.blogspot.co.id/2011/12/time-series-deret-berkala\\_3237.html](http://wenthyoktavin.blogspot.co.id/2011/12/time-series-deret-berkala_3237.html)
- 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>
- Stice, S. (2009). *Akuntansi Intermediate*. Jakarta: Salemba Empat.
- Vuppalapati, K. (1995). JIT and TQM : A Case for Joint Implementation. *International Journal of Operation and Management*, 15, 553-564.
- Yulianto, M. (2012). Analisa Time Series. *Pengenalan analisa serial waktu (time series)*.

LAMPIRAN

LAMPIRAN A. PERANCANGAN APLIKASI

A1. Perancangan *Scenario*

1. *Scenario* Mengelola Data Karyawan
  - a. Tambah Data Karyawan

*Scenario* Tambah Data Karyawan

<b>Nama Usecase</b>	Mengelola Data Karyawan
<b>Aktor</b>	Owner
<b>Pra-Kondisi</b>	Owner telah berhasil masuk ke halaman dashboard
<b>Pasca-Kondisi</b>	Owner berhasil menambahkan data karyawan
<b>Aktor</b>	<b>Aplikasi</b>
<b>Skenario Utama "Tambah Data Karyawan"</b>	
1. Klik menu "Data Master"	
2. Pilih dan klik submenu "Karyawan"	
	3. Menampilkan halaman utama data karyawan
4. Klik tombol "Tambah Karyawan"	
	5. Menampilkan form tambah data karyawan
6. Mengisi form tambah data karyawan	
7. Klik "Submit"	
	8. Menyimpan data karyawan ke database
	9. Menampilkan halaman utama list data karyawan
<b>Skenario Alternatif "Form Kosong"</b>	
7a. Klik tombol "Submit"	
	8a. Menampilkan pesan <i>error</i> pada <i>field</i>
<b>Skenario Alternatif "Klik Tombol Kembali"</b>	
7b. Klik "Back"	
	8b. Menampilkan halaman utama

list data karyawan

b. Ubah Data Karyawan

**Scenario Ubah Data Karyawan**

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

- 2. Mengelola Data Customer
  - a. Tambah Data Customer

**Scenario Tambah Data Customer**

<b>Nama Usecase</b>	Mengelola Data Customer
<b>Aktor</b>	Owner
<b>Pra-Kondisi</b>	Owner telah berhasil masuk ke halaman dashboard
<b>Pasca-Kondisi</b>	Owner berhasil menambahkan data customer
<b>Aktor</b>	<b>Aplikasi</b>
<b>Skenario Utama "Tambah Data Customer"</b>	
1. Klik menu "Data Master"	
2. Pilih dan klik submenu "Customer"	
	3. Menampilkan halaman utama data customer
4. Klik tombol "Tambah Customer"	
	5. Menampilkan form tambah data customer
6. Mengisi form tambah data customer	
7. Klik "Submit"	
	8. Menyimpan data customer ke <i>database</i>
	9. Menampilkan halaman utama list data customer
<b>Skenario Alternatif "Form Kosong"</b>	
7a. Klik tombol "Submit"	
	8a. Menampilkan pesan <i>error</i> pada <i>field</i>
<b>Skenario Alternatif "Klik Tombol Kembali"</b>	
7b. Klik "Back"	
	8b. Menampilkan halaman utama list data customer

- b. Ubah Data Customer

**Scenario Ubah Data Customer**

<b>Nama Usecase</b>	Mengelola Data Customer
<b>Aktor</b>	Owner
<b>Pra-Kondisi</b>	Owner telah berhasil masuk ke halaman data customer
<b>Pasca-Kondisi</b>	Owner berhasil mengubah data customer

Aktor	Aplikasi
-------	----------

<b>Skenario Utama "Ubah Data Customer"</b>
--

1. Klik icon "ubah" pada kolom yang datanya ingin dirubah	
	2. Menampilkan form ubah dengan data sebelumnya
3. Mengisi data yang ingin diubah	
4. Klik "Submit"	
	5. Menyimpan data customer ke <i>database</i>
	6. Menampilkan halaman utama list data customer

<b>Skenario Alternatif "Pilih dan Klik Drop Down Aktif/Non aktif"</b>
---

3a. Pilih dan klik non aktif	
4a. Klik tombol "Submit"	
	5a. Memperbarui data customer ke <i>database</i> .
	6a. Menampilkan halaman utama list data customer

<b>Skenario Alternatif "Form Kosong"</b>
--

4b. Klik tombol "Submit"	
	5b. Menampilkan pesan <i>error</i> pada <i>field</i>

<b>Skenario Alternatif "Klik Tombol Kembali"</b>
--

4c. Klik "Back"	
	5c. Menampilkan halaman utama list data customer

3. Mengelola Data Barang
  - a. Tambah Data Barang

**Scenario Tambah Data Barang**

<b>Nama Usecase</b>	Mengelola Data Barang
<b>Aktor</b>	Admin
<b>Pra-Kondisi</b>	Admin telah berhasil masuk ke halaman dashboard
<b>Pasca-Kondisi</b>	Admin berhasil menambahkan data barang

Aktor	Aplikasi
-------	----------

<b>Skenario Utama "Tambah Data Barang"</b>
--

1. Klik menu "Data Barang"	
	2. Menampilkan halaman utama data barang
3. Klik tombol "Tambah Barang"	
	4. Menampilkan form tambah data barang
5. Mengisi form data tambah barang	
6. Klik "Submit"	
	7. Menyimpan data barang ke <i>database</i>
	8. Menampilkan halaman utama list data barang

<b>Skenario Alternatif "Form Kosong"</b>
--

7a. Klik tombol "Submit"	
	8a. Menampilkan pesan <i>error</i> pada <i>field</i>

<b>Skenario Alternatif "Klik Tombol Kembali"</b>
--

7b. Klik "Back"	
	8b. Menampilkan halaman utama list data barang

b. Ubah Data Barang

**Scenario Ubah Data Barang**

---

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

4. Menambah Data Barang Keluar

**Scenario Tambah Data Barang Keluar**



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

11c. Menampilkan data barang pada tabel data pembelian

- 5. Melihat Data Barang Keluar
  - a. Melihat Data Barang Keluar – Admin

**ScenarioMelihat Data Barang Keluar**

<b>Nama Usecase</b>	Melihat Data Barang Keluar
<b>Aktor</b>	Sales
<b>Pra-Kondisi</b>	Sales telah berhasil masuk ke halaman dashboard
<b>Pasca-Kondisi</b>	Sales berhasil melihat data barang keluar
<b>Aktor</b>	<b>Aplikasi</b>
<b>Skenario Utama</b>	
<b>“Melihat Data Barang Keluar”</b>	
1. Klik menu “Barang Keluar”	
2. Pilih dan klik submenu “Data Barang Keluar”	
	3. Menampilkan halaman utama dan list nota data barang keluar
4. Klik icon “lihat” pada kolom nota yang ingin dilihat	
	5. Menampilkan rincian data barang keluar

- b. Melihat Data Barang Keluar – Owner

**ScenarioMelihat Data Barang Keluar**

<b>Nama Usecase</b>	Melihat Data Barang Keluar
<b>Aktor</b>	Owner
<b>Pra-Kondisi</b>	Owner telah berhasil masuk ke halaman dashboard
<b>Pasca-Kondisi</b>	Owner berhasil melihat data barang keluar
<b>Aktor</b>	<b>Aplikasi</b>
<b>Skenario Utama</b>	
<b>“Melihat Data Barang Keluar”</b>	
1. Klik menu “Barang Keluar”	
2. Pilih dan klik submenu “Data Barang Keluar”	

	3. Menampilkan halaman utama data barang keluar
4. Klik icon “lihat” pada kolom nota yang ingin dilihat	
	5. Menampilkan rincian data barang keluar

c. Melihat Data Barang Keluar - Bagian Gudang

**Scenario Melihat Data Barang Keluar**

<b>Nama Usecase</b>	Melihat Data Barang Keluar
<b>Aktor</b>	Bagian Gudang
<b>Pra-Kondisi</b>	Bagian Gudang telah berhasil masuk ke halaman dashboard
<b>Pasca-Kondisi</b>	Bagian Gudang berhasil melihat data barang keluar

Aktor	Aplikasi
<b>Skenario Utama “Melihat Data Barang Keluar”</b>	
1. Klik menu “Barang Keluar”	
2. Pilih dan klik submenu “Data Barang Keluar”	
	3. Menampilkan halaman utama data barang keluar
4. Klik icon “lihat” pada kolom nota yang ingin dilihat	
	5. Menampilkan rincian data barang keluar

6. Menambah Data Barang Masuk

**Scenario Tambah Data Barang Masuk**

<b>Nama Usecase</b>	Menambah Data Barang Masuk
<b>Aktor</b>	Admin
<b>Pra-Kondisi</b>	Admin telah berhasil masuk ke halaman dashboard
<b>Pasca-Kondisi</b>	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” |  |
|  | 3. Menampilkan form data barang masuk  |
| 4. Mengisi form data barang keluar             |  |
| 5. Klik tombol “Add”                           |  |
|  | 6. Menampilkan data barang pada field data pembelian                         |
| 7. Mengisi data karyawan dan keterangan        |  |
| 8. Klik tombol “Submit”                        |  |
|  | 9. Menampilkan pop up pada field   |
| 10. Klik tombol “Ok”                           |  |
|  | 11. Menyimpan data barang keluar ke <i>database</i>                          |
|  | 12. Menampilkan halaman utama “Data Barang Masuk” dan list nota barang masuk |

**Skenario Alternatif**  
**“Form Kosong”**

- |                          |  |
|--------------------------|--|
| 5a. Klik tombol “Add”    |  |
|                          | 6a. Menampilkan pesan <i>error</i> pada <i>field</i> |
| 7a. Klik tombol “Submit” |  |
|                          | 8a. Menampilkan pesan <i>error</i> pada <i>field</i> |

**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 |

7. Melihat Data Barang Masuk

a. Melihat Data Barang Masuk – Admin

**ScenarioMelihat Data Barang Masuk**

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

b. Melihat Data Barang Masuk – owner

**ScenarioMelihat Data Barang Masuk**

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

barang masuk

8. Menambah Data Barang Retur

**Scenario Tambah Data Barang Retur**

<b>Nama Usecase</b>	Menambah Data Barang Retur
<b>Aktor</b>	Sales
<b>Pra-Kondisi</b>	Sales telah berhasil masuk ke halaman dashboard
<b>Pasca-Kondisi</b>	Sales berhasil menambah data barang retur

Aktor	Aplikasi
-------	----------

<b>Skenario Utama "Tambah Data Barang Retur"</b>
--

1. Klik menu "Retur Barang"	
2. Pilih dan klik submenu "Input Retur"	
	3. Menampilkan form data barang retur
4. Mengisi form data barang retur	
5. Klik tombol "Add"	
	6. Menampilkan data barang retur pada tabel list retur
7. Mengisi data customer dan keterangan	
8. Klik tombol "Submit"	
	9. Menampilkan pop up pada field
10. Klik tombol "Ok"	
	11. Menyimpan data barang retur ke <i>database</i>
	12. Menampilkan halaman utama data barang retur dan list nota barang retur

<b>Skenario Alternatif "Form Kosong"</b>
--

5a. Klik tombol "Add"	
	6a. Menampilkan pesan <i>error</i> pada <i>field</i>
7a. Klik tombol "Submit"	

8a. Menampilkan pesan <i>error</i> pada <i>field</i>	
<b>Skenario Alternatif "Klik Icon Hapus"</b>	
7b. Klik icon "hapus" pada data barang yang ingin dihapus	8b. Menghapus data barang retur
<b>Skenario Alternatif "Klik Tombol Batal"</b>	
10b. Klik tombol "Cancel"	
11b. Menampilkan data barang retur pada tabel list retur	

9. Melihat Data Barang Retur  
 a. Melihat Data Barang Retur – Admin

**ScenarioMelihat Data Barang Retur**

<b>Nama Usecase</b>	Melihat Data Barang Retur
<b>Aktor</b>	Admin
<b>Pra-Kondisi</b>	Admin telah berhasil masuk ke halaman dashnoard
<b>Pasca-Kondisi</b>	Admin berhasil melihat data barang retur
<b>Aktor</b>	<b>Aplikasi</b>
<b>Skenario Utama "Melihat Data Barang Retur"</b>	
1. Klik menu "Retur Barang"	
2. Pilih dan klik submenu "Data Barang Retur"	
3. Menampilkan halaman utama dan list nota data barang retur	
4. Klik icon "lihat" pada kolom data yang diinginkan	
5. Menampilkan detail data barang retur	

- b. Melihat Data Barang Retur – Owner

**ScenarioMelihat Data Barang Retur**

<b>Nama Usecase</b>	Melihat Data Barang Retur
---------------------	---------------------------

<b>Aktor</b>	Owner
<b>Pra-Kondisi</b>	Owner telah berhasil masuk ke halaman dashboard
<b>Pasca-Kondisi</b>	Owner berhasil melihat data barang retur

Aktor	Aplikasi
<b>Skenario Utama</b> <b>“Melihat Data Barang Retur”</b>	
<ol style="list-style-type: none"> <li>1. Klik menu “Retur Barang”</li> <li>2. Pilih dan klik submenu “Data Barang Retur”</li> </ol>	
<ol style="list-style-type: none"> <li>3. Menampilkan halaman utama data barang retur</li> </ol>	

c. Melihat Data Barang Retur – Bagian Gudang

**ScenarioMelihat Data Barang Retur**

<b>Nama Usecase</b>	Melihat Data Barang Retur
<b>Aktor</b>	Bagian Gudang
<b>Pra-Kondisi</b>	Bagian Gudang telah berhasil masuk ke halaman dashboard
<b>Pasca-Kondisi</b>	Bagian Gudang berhasil melihat data barang retur

Aktor	Aplikasi
<b>Skenario Utama</b> <b>“Melihat Data Barang Retur”</b>	
<ol style="list-style-type: none"> <li>1. Klik menu “Retur Barang”</li> <li>2. Pilih dan klik submenu “Data Barang Retur”</li> </ol>	
<ol style="list-style-type: none"> <li>3. Menampilkan halaman utama data barang retur</li> </ol>	



10. Melihat Kondisi Stok Barang

a. Melihat Kondisi Stok Barang – Admin

**ScenarioMelihat Stok Barang Terakhir**

<b>Nama Usecase</b>	Melihat Data Barang Retur
<b>Aktor</b>	Admin
<b>Pra-Kondisi</b>	Admin Berhasil masuk ke dashboard
<b>Pasca-Kondisi</b>	Admin berhasil melihat data barang terakhir
<b>Aktor</b>	<b>Aplikasi</b>
<b>Skenario Utama “Melihat Data Barang Terakhir”</b>	
<ol style="list-style-type: none"> <li>1. Klik menu “Retur Barang”</li> <li>2. Pilih dan klik submenu “Kondisi Stok Barang”</li> </ol>	
	<ol style="list-style-type: none"> <li>3. Menampilkan halaman utama kondisi data barang</li> </ol>

b. Melihat Kondisi Stok Barang – Owner

**ScenarioMelihat Stok Barang Terakhir**

<b>Nama Usecase</b>	Melihat Data Barang Retur
<b>Aktor</b>	Owner
<b>Pra-Kondisi</b>	Owner Berhasil masuk ke dashboard
<b>Pasca-Kondisi</b>	Owner berhasil melihat data barang terakhir
<b>Aktor</b>	<b>Aplikasi</b>
<b>Skenario Utama “Melihat Data Barang Terakhir”</b>	
<ol style="list-style-type: none"> <li>1. Klik menu “Retur Barang”</li> <li>2. Pilih dan klik submenu “Kondisi Stok Barang”</li> </ol>	
	<ol style="list-style-type: none"> <li>3. Menampilkan halaman utama kondisi data barang</li> </ol>

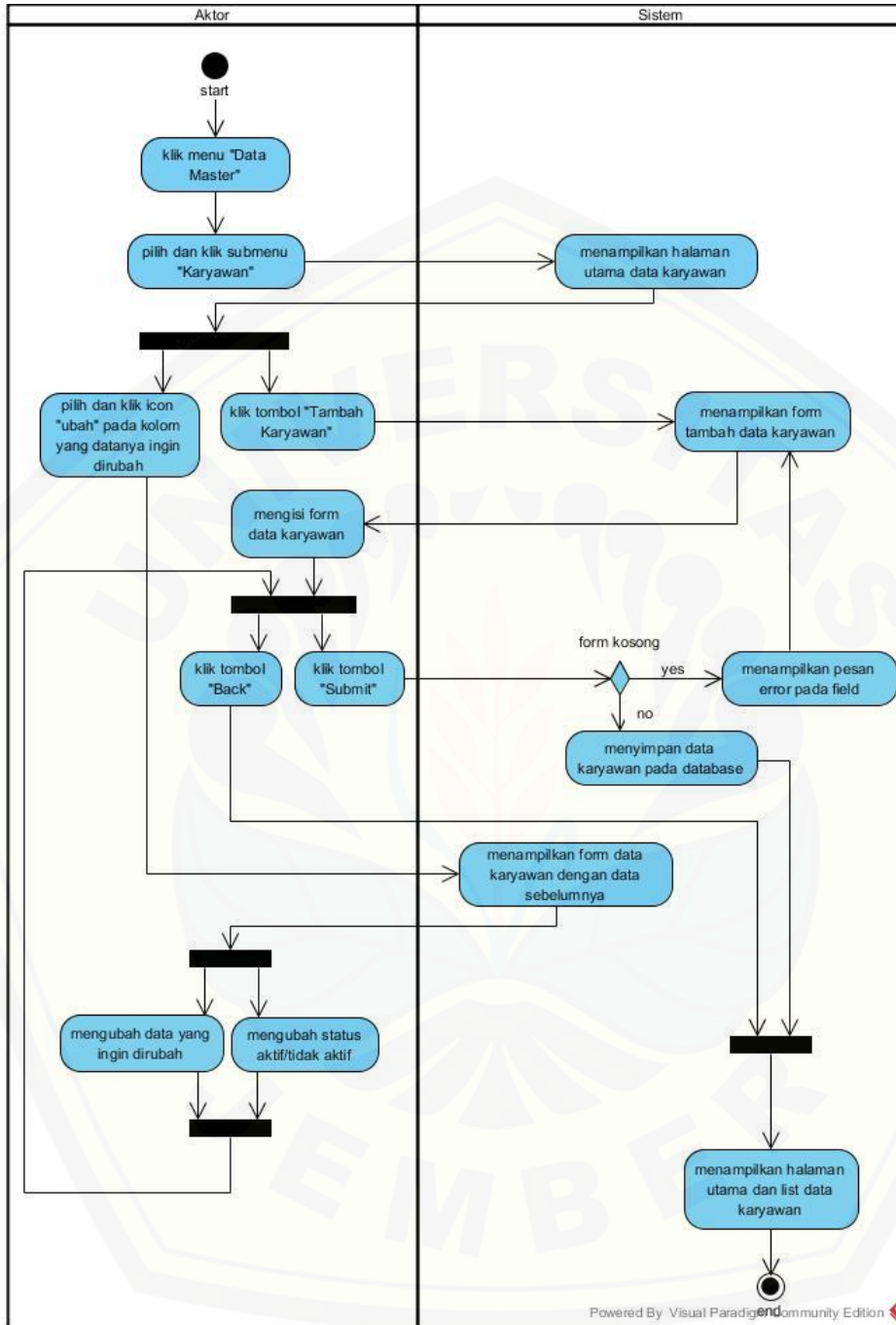
c. Melihat Kondisi Stok Barang – Bagian Gudang

**ScenarioMelihat Stok Barang Terakhir**

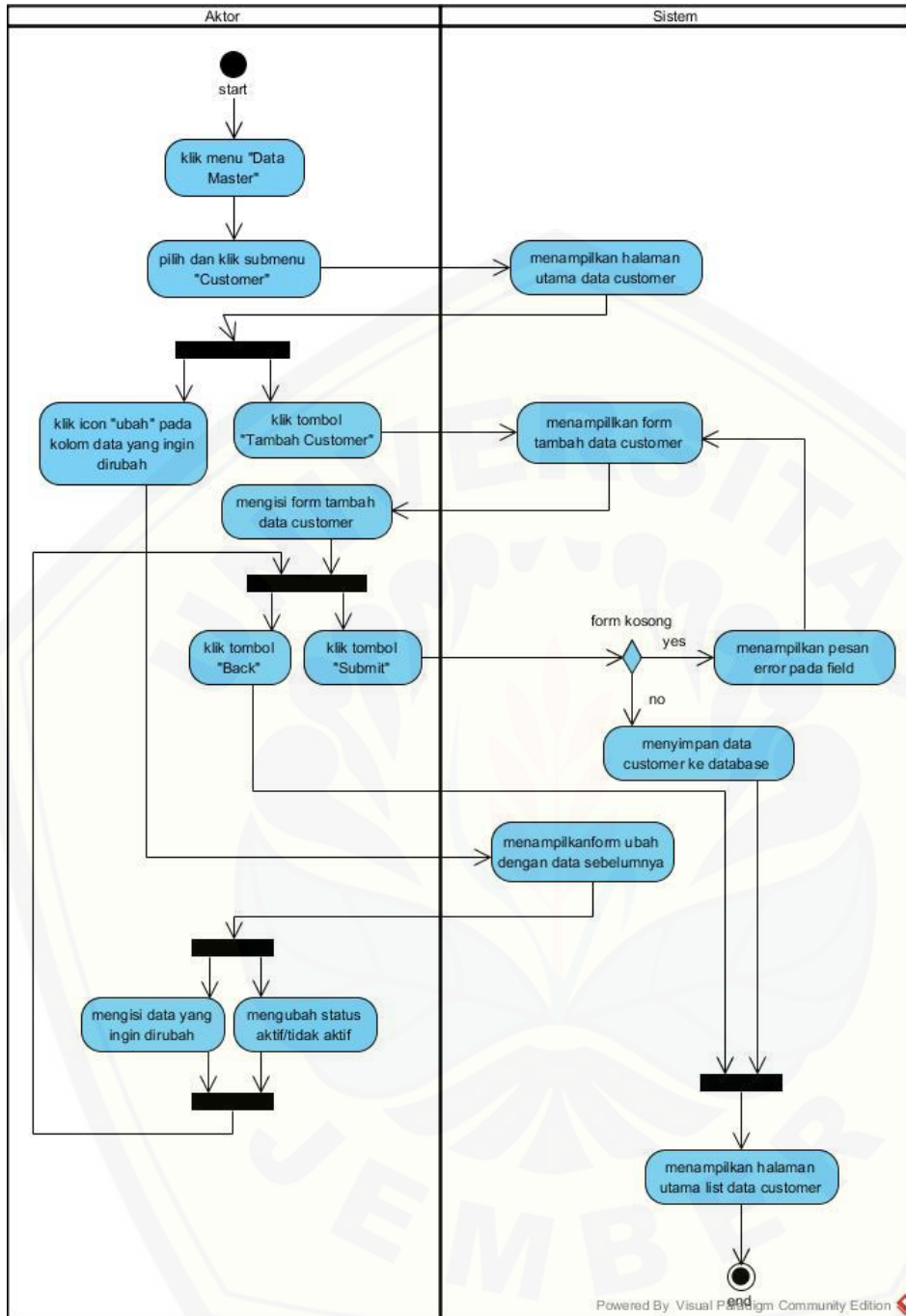
<b>Nama Usecase</b>	Melihat Data Barang Retur
<b>Aktor</b>	Bagian Gudang
<b>Pra-Kondisi</b>	Bagian Gudang Berhasil masuk ke dashboard
<b>Pasca-Kondisi</b>	Bagian Gudang berhasil melihat data barang terakhir
<b>Aktor</b>	<b>Aplikasi</b>
<b>Skenario Utama</b>	
<b>“Melihat Data Barang Terakhir”</b>	
1.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

**A2. Perancangan Activity Diagram**

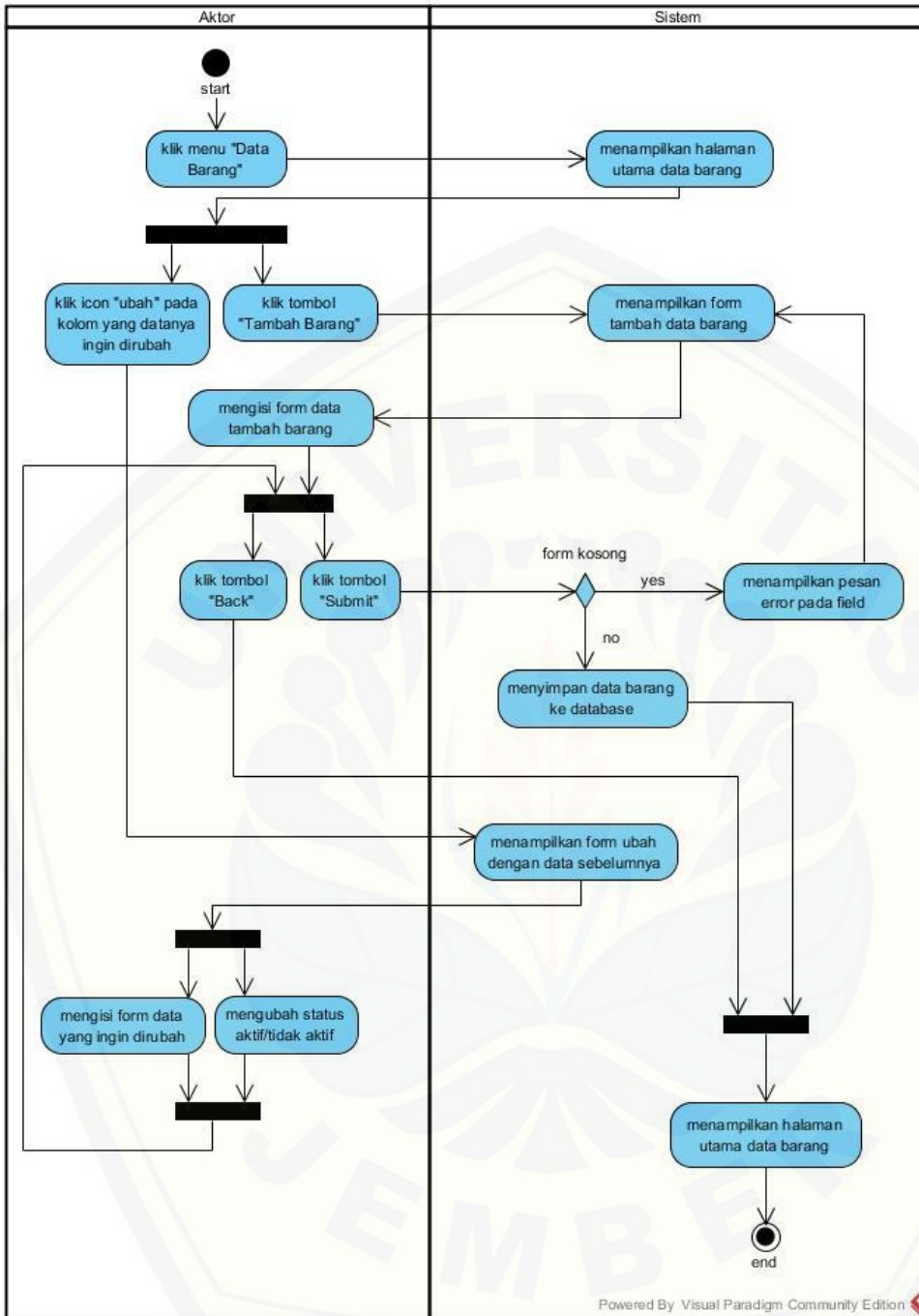
1. Activity Diagram Mengelola Data Karyawan



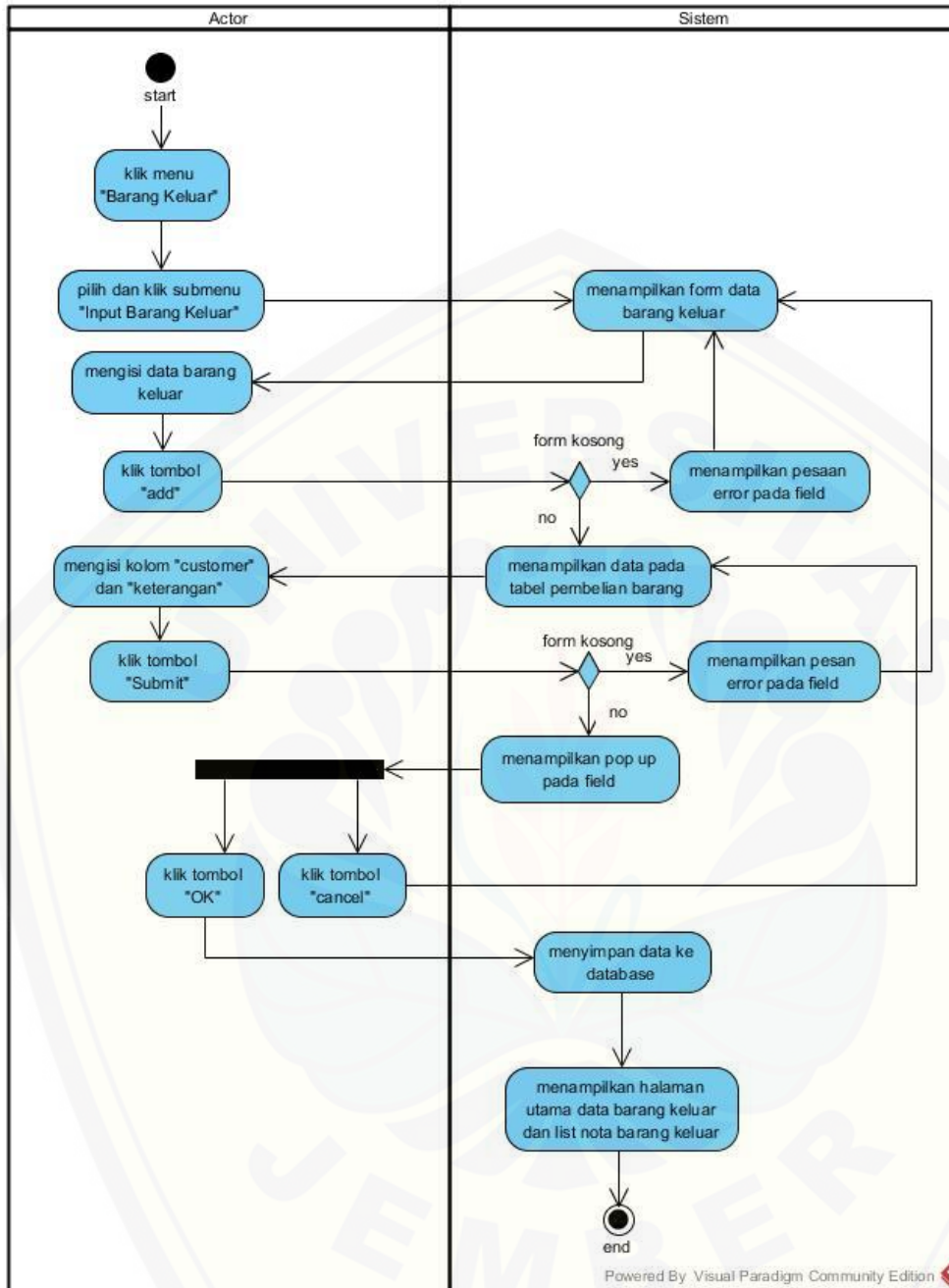
2. Activity Diagram Mengelola Data Customer



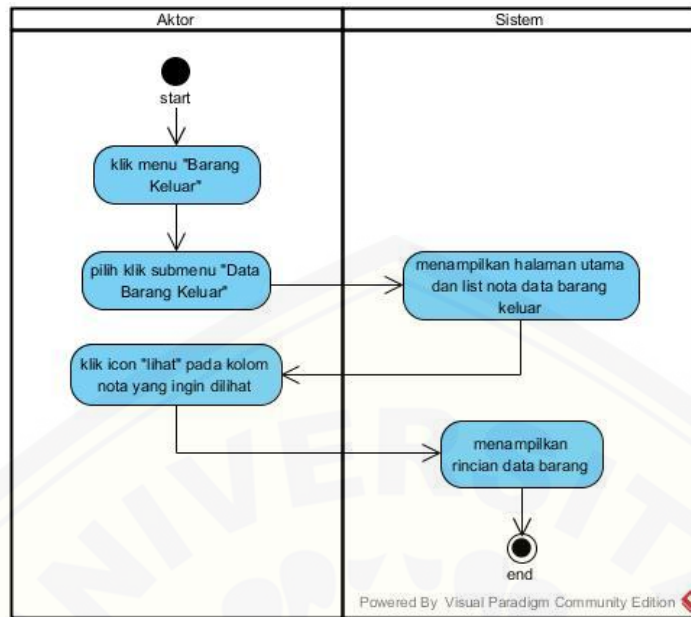
3. Activity Diagram Mengelola Data Barang



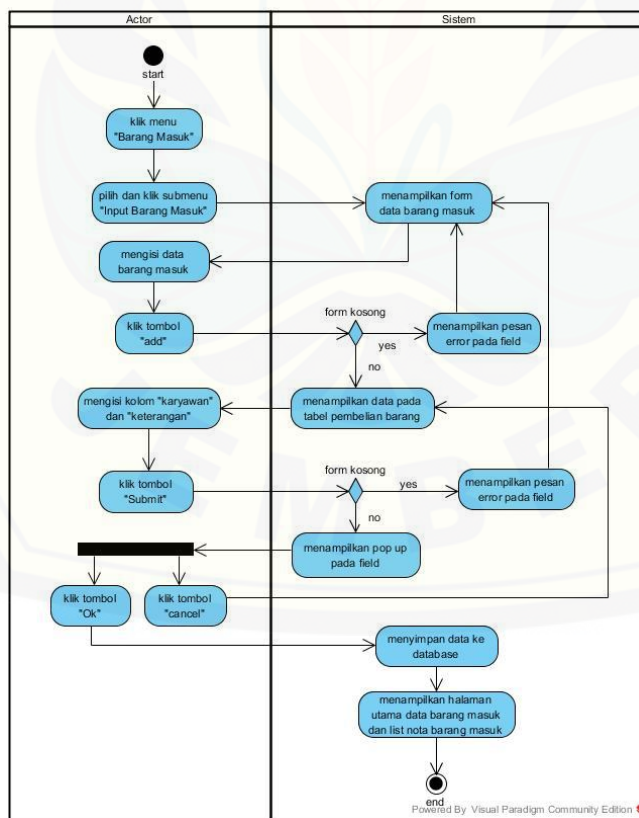
4. Activity Diagram Menambah Data Barang Keluar



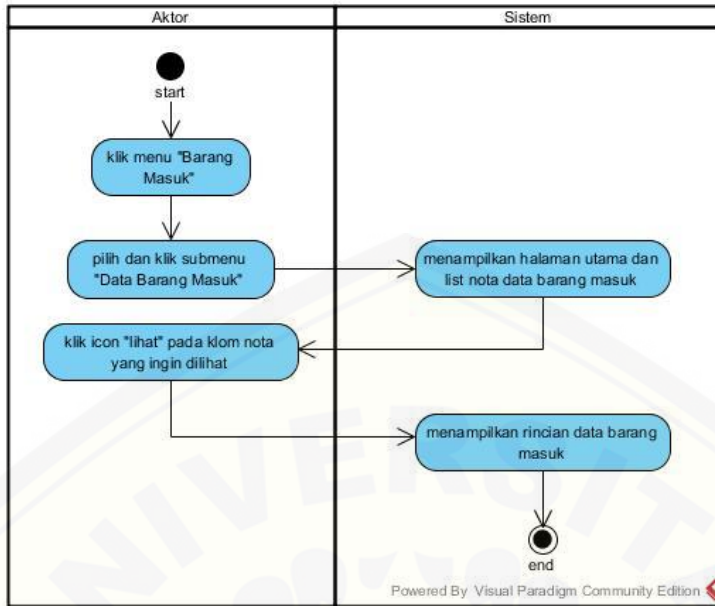
5. Activity Diagram Melihat Data Barang Keluar



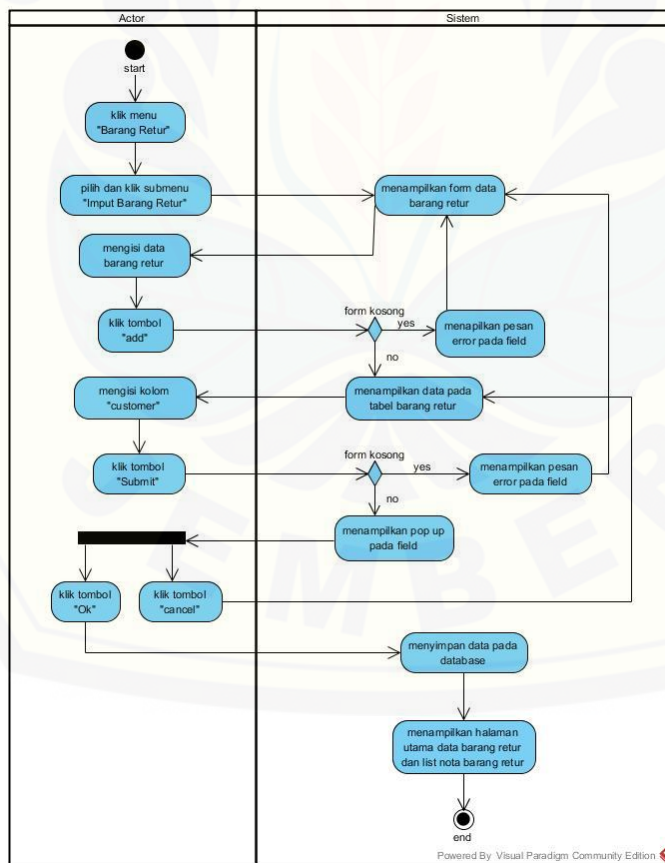
6. Activity Diagram Menambah Data Barang Masuk



7. Activity Diagram Melihat Data Barang Masuk

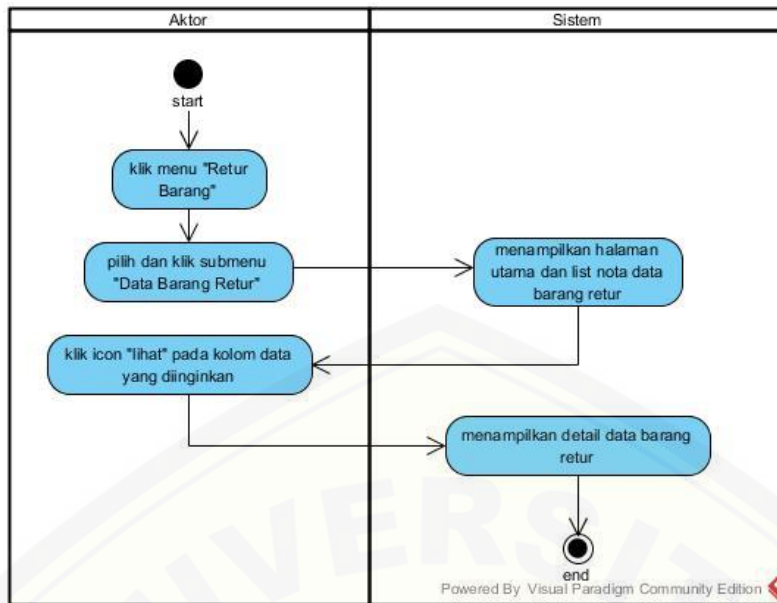


8. Activity Diagram Menambah Data Barang Retur

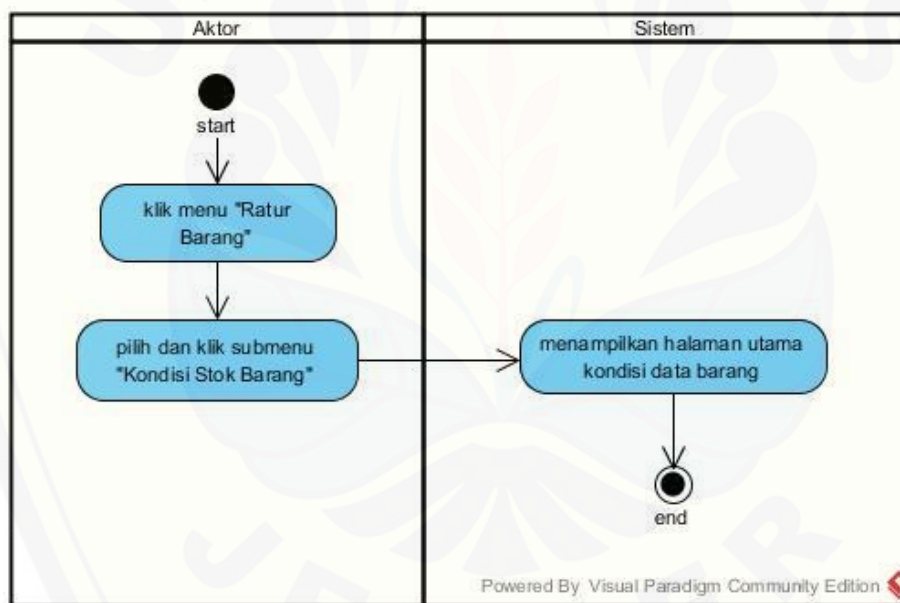


9. Activity Diagram Melihat Data Barang Masuk



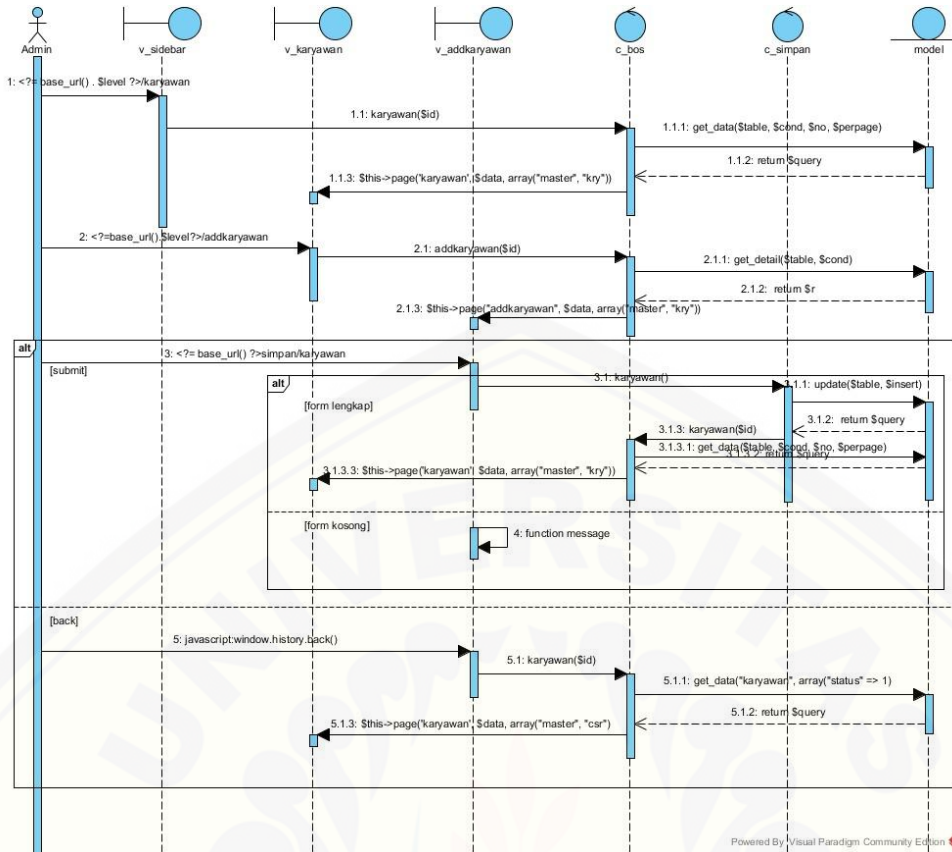


10. Activity Diagram Melihat Kondisi Stok

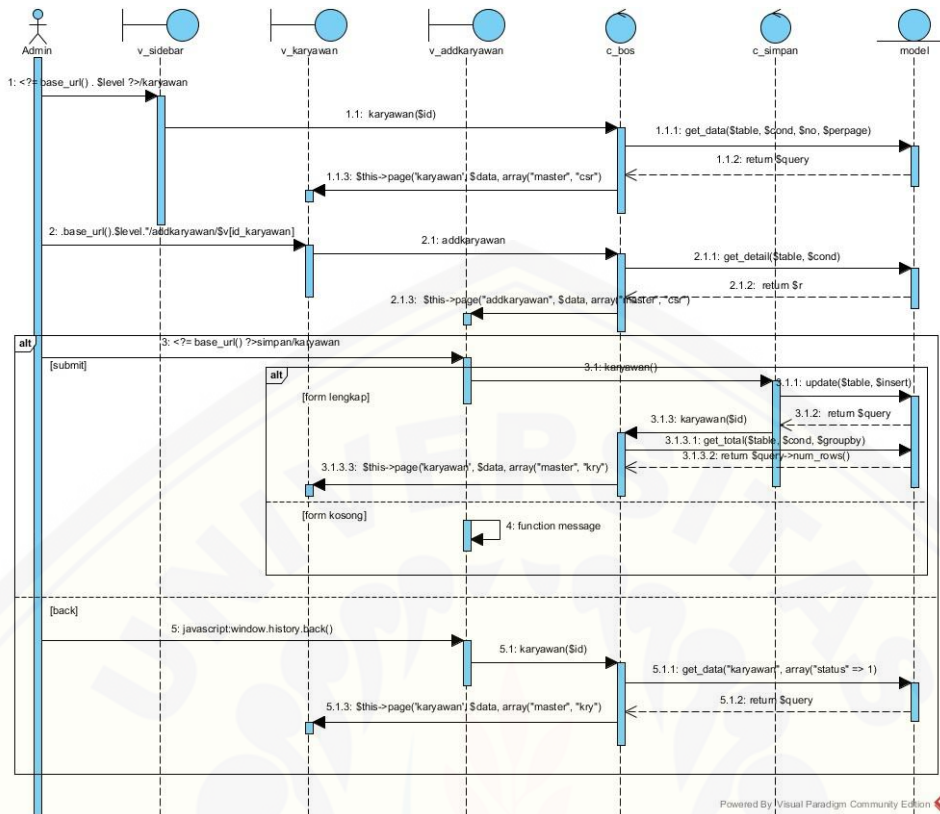


**A3. Perancangan Squence Diagram**

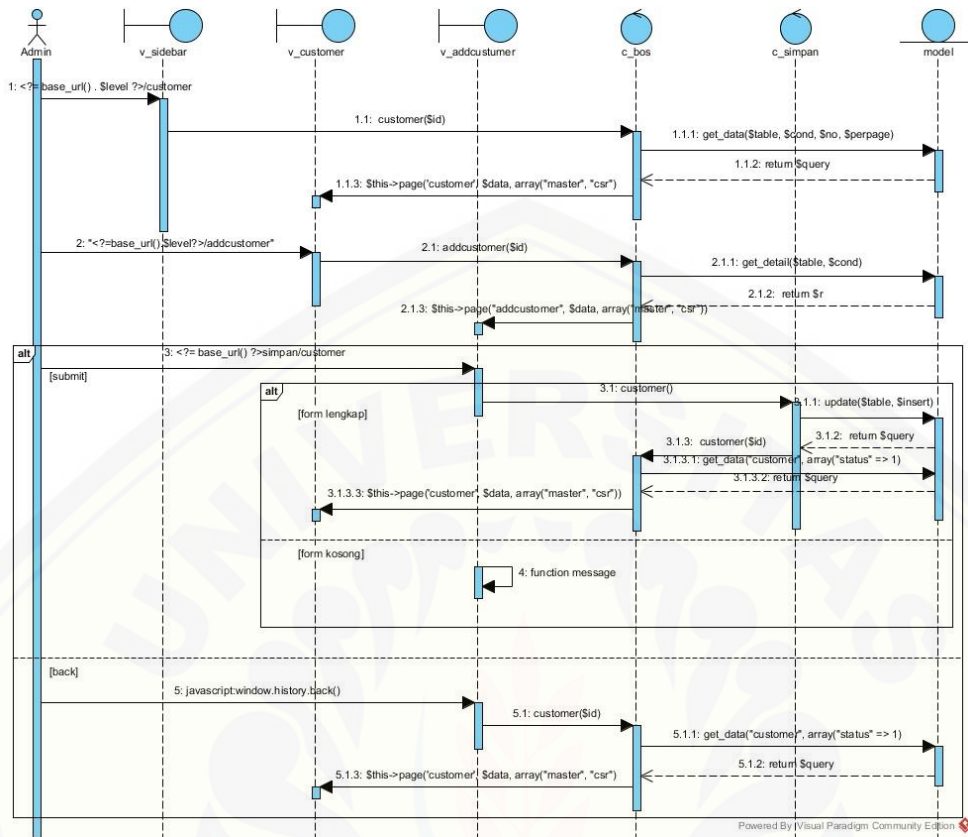
1. *Squence Diagram* Mengelola Data Karyawan
  - a. *Squence Diagram* Tambah Data Karyawan



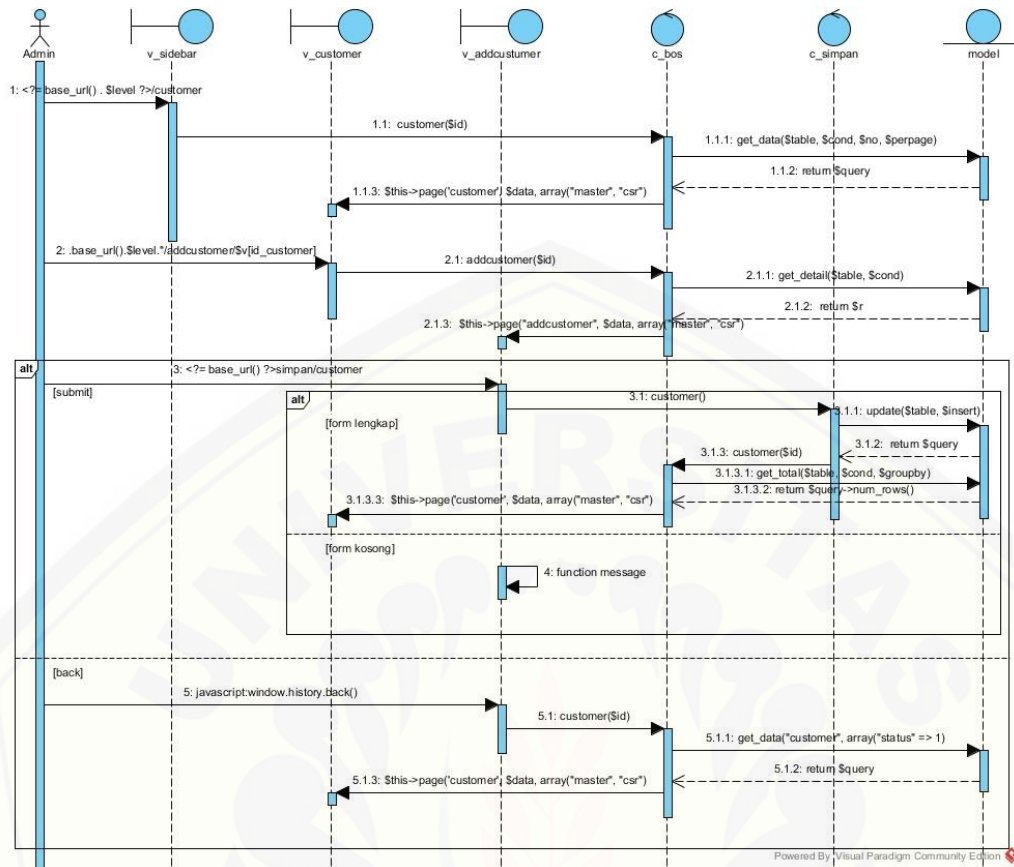
b. Sequence Diagram Ubah Data Karyawan



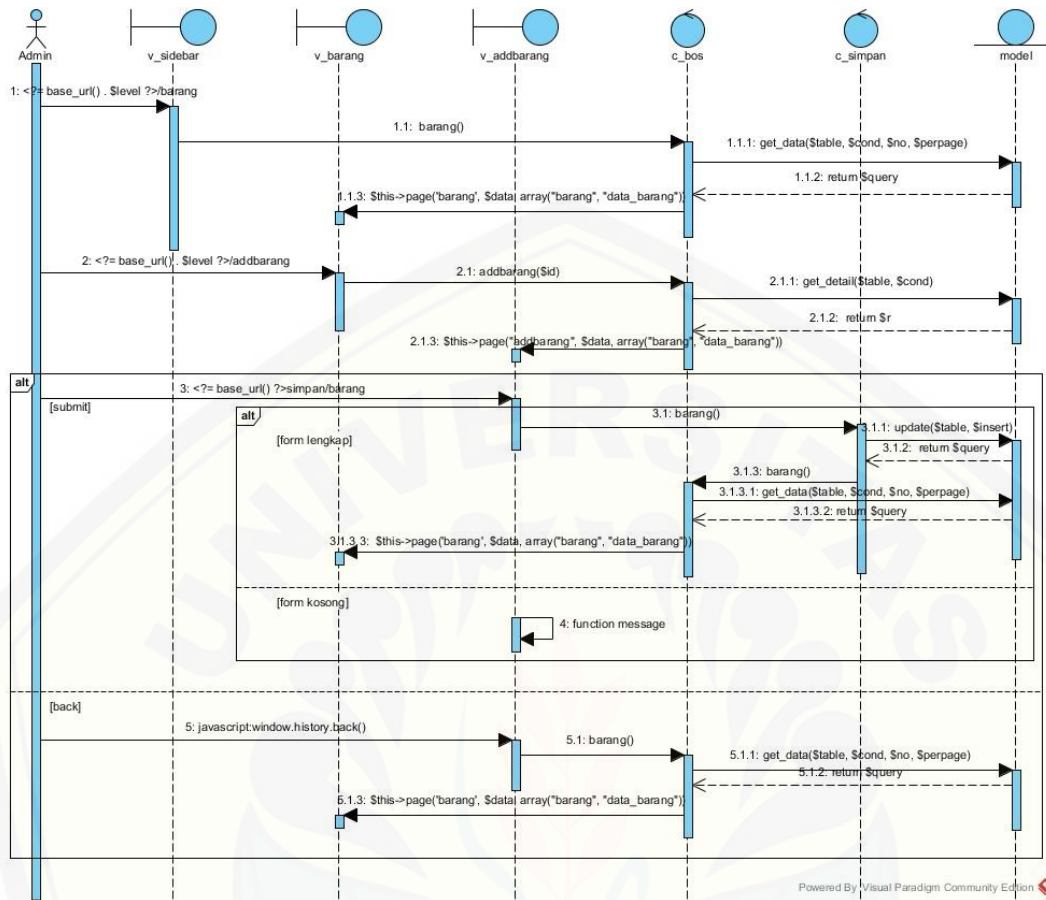
2. Sequence Diagram Mengelola Data Customer  
 a. Sequence Diagram Tambah Data Customer



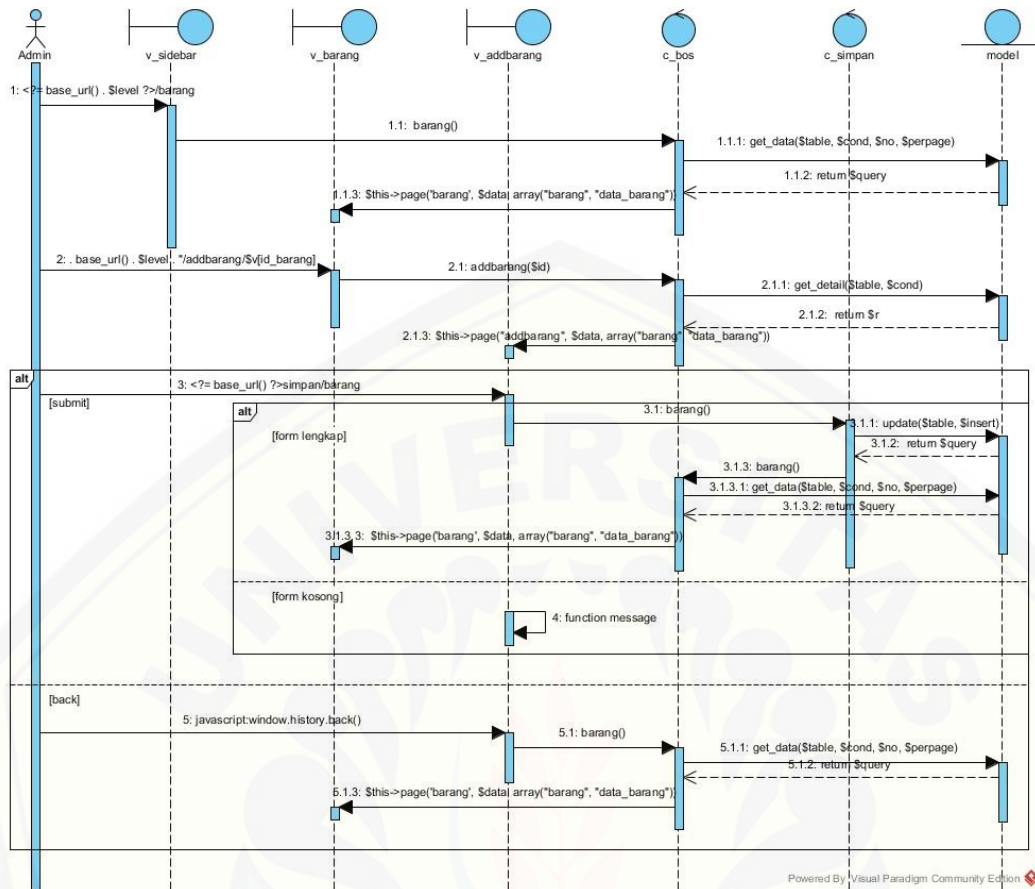
b. Sequence Diagram Ubah Data Customer



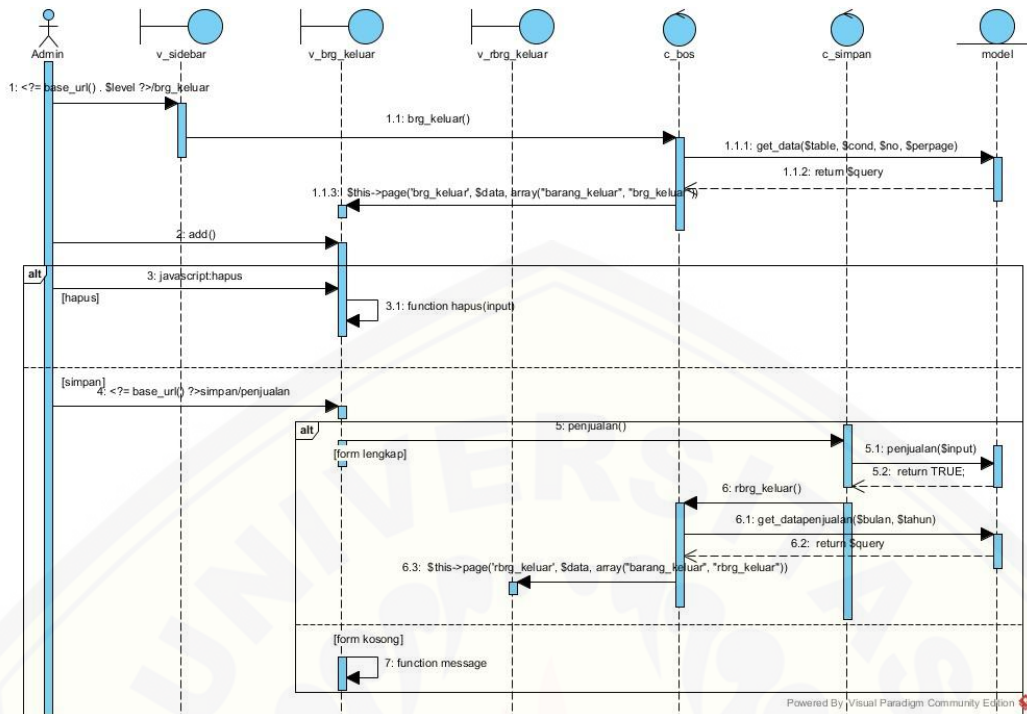
3. Sequence Diagram Mengelola Data Barang  
 a. Sequence Diagram Tambah Data Barang



b. Sequence Diagram Ubah Data Barang

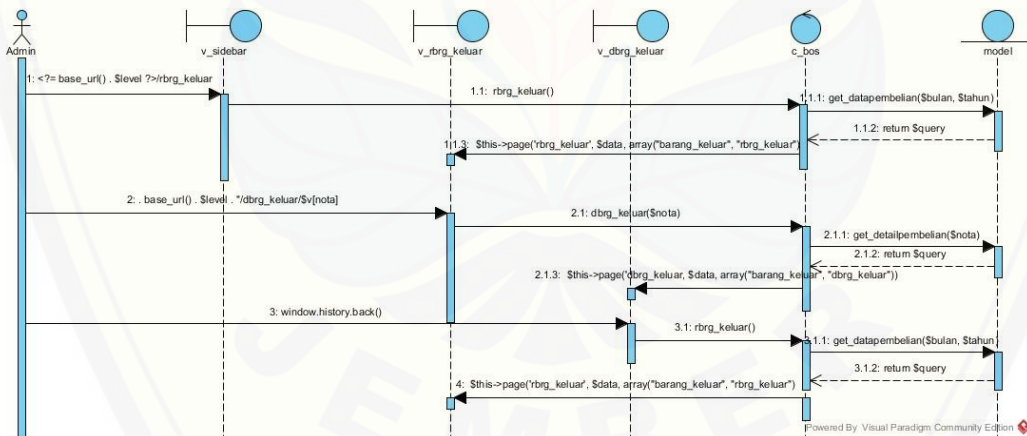


#### 4. Sequence Diagram Menambah Data Barang Keluar



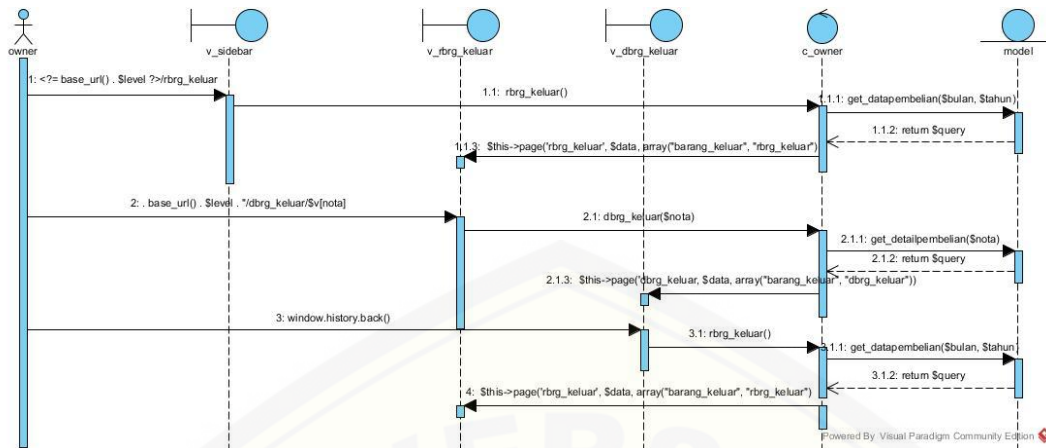
#### 5. Sequence Diagram Melihat Data Barang Keluar

##### a. Sequence Diagram Melihat Data Barang Keluar – Admin

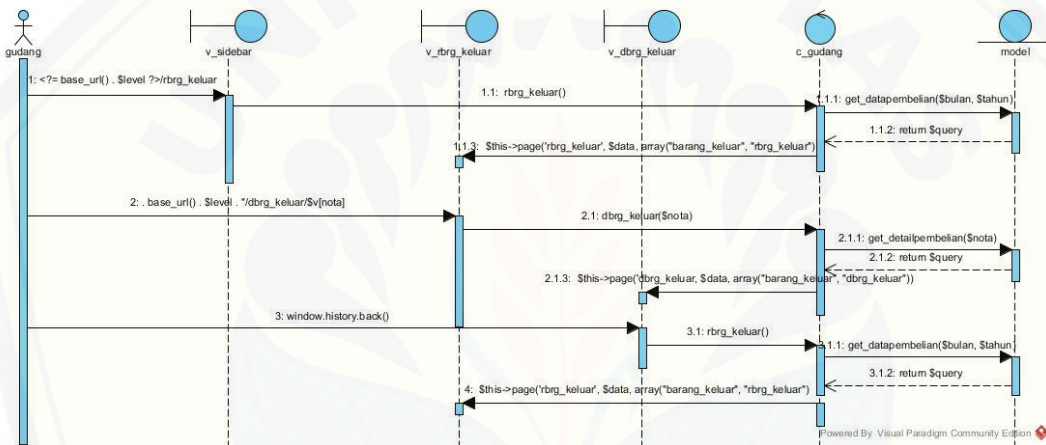




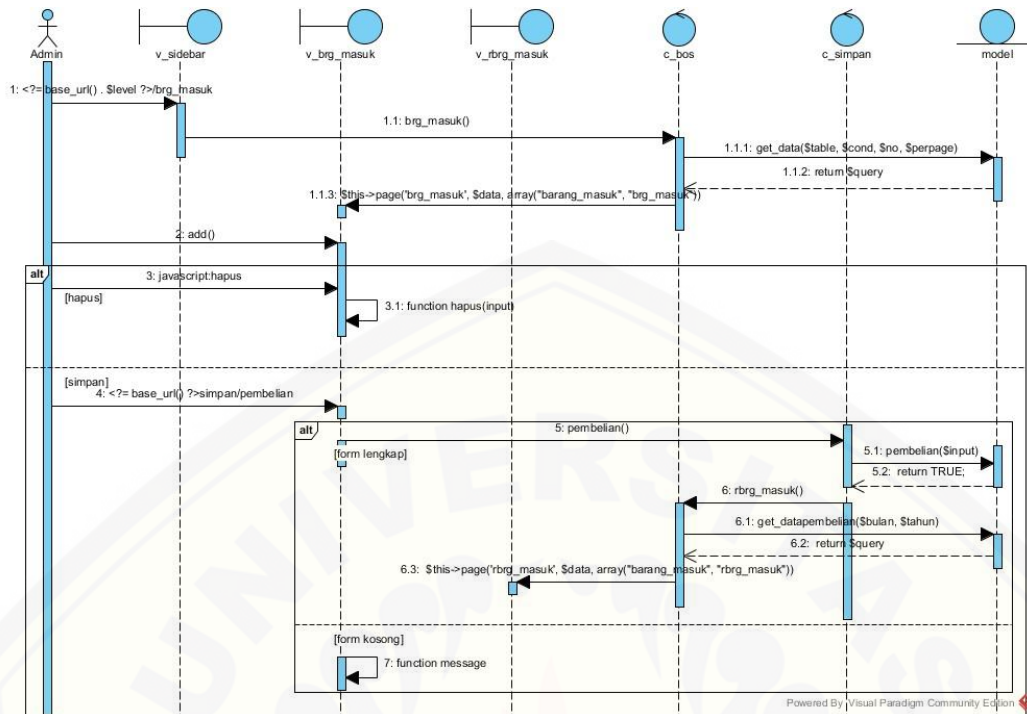
b. Sequence Diagram Melihat Data Barang Keluar – Owner



c. Sequence Diagram Melihat Data Barang Keluar – Bagian Gudang

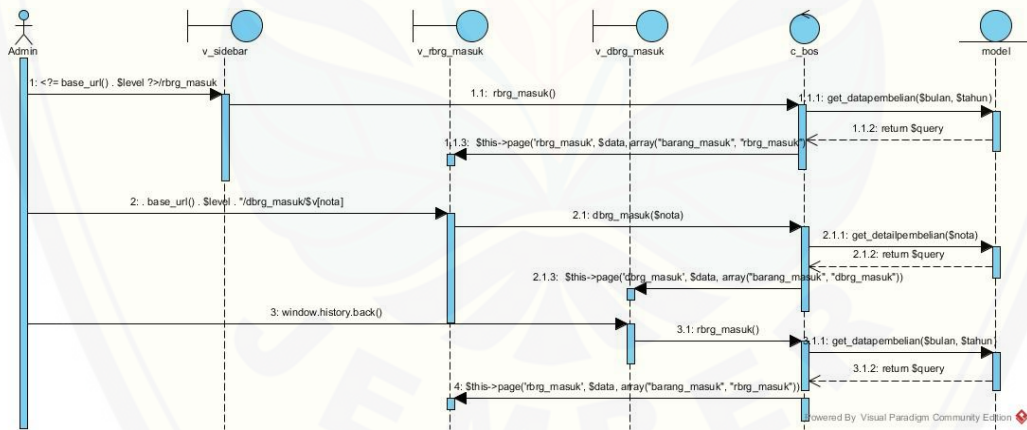


### 6. Sequence Diagram Menambah Data Barang Masuk

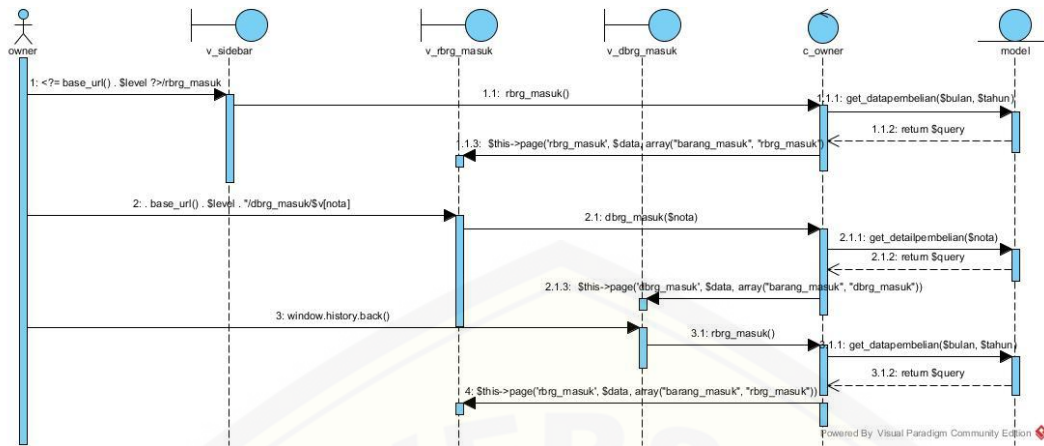


### 7. Sequence Diagram Melihat Data Barang Masuk

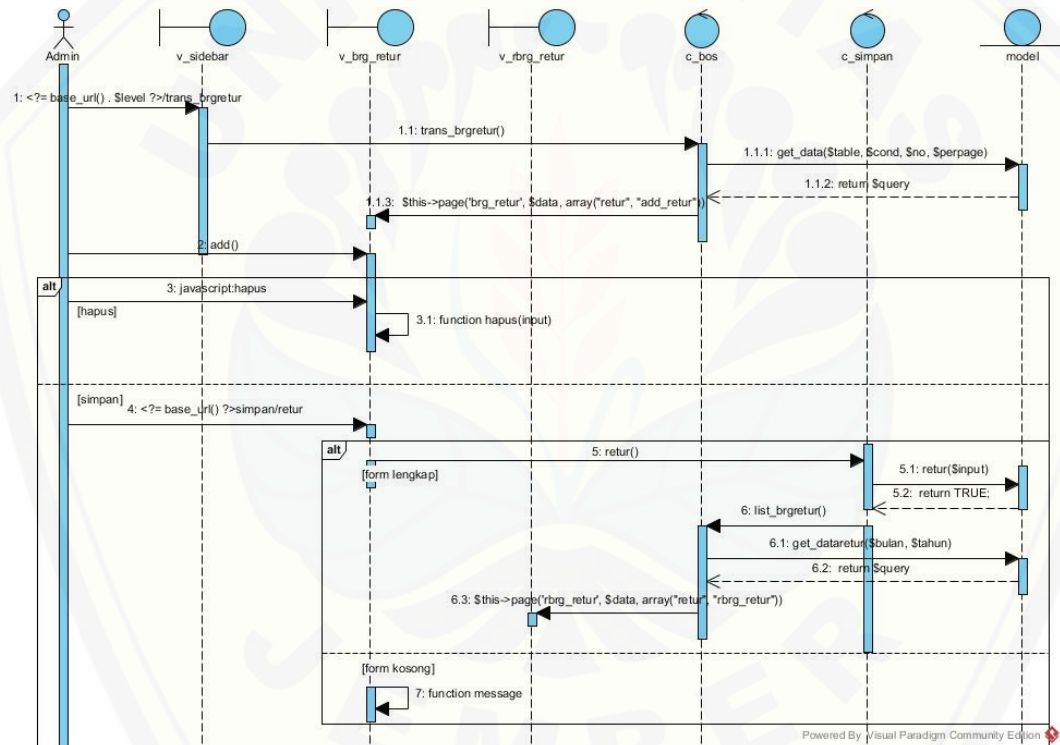
#### a. Sequence Diagram Melihat Data Barang Masuk – Admin



b. *Sequence Diagram* Melihat Data Barang Masuk – Owner

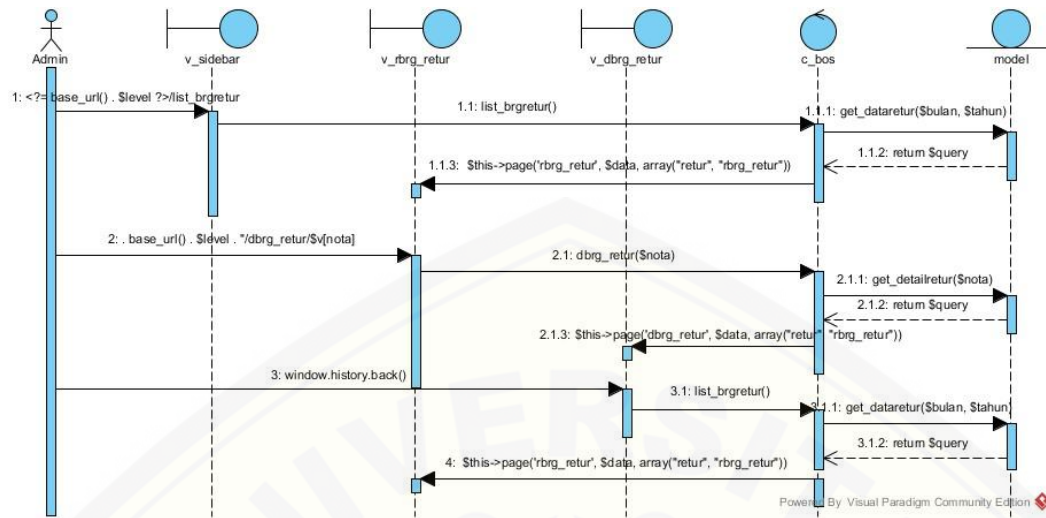


8. *Sequence Diagram* Menambah Data Barang Retur

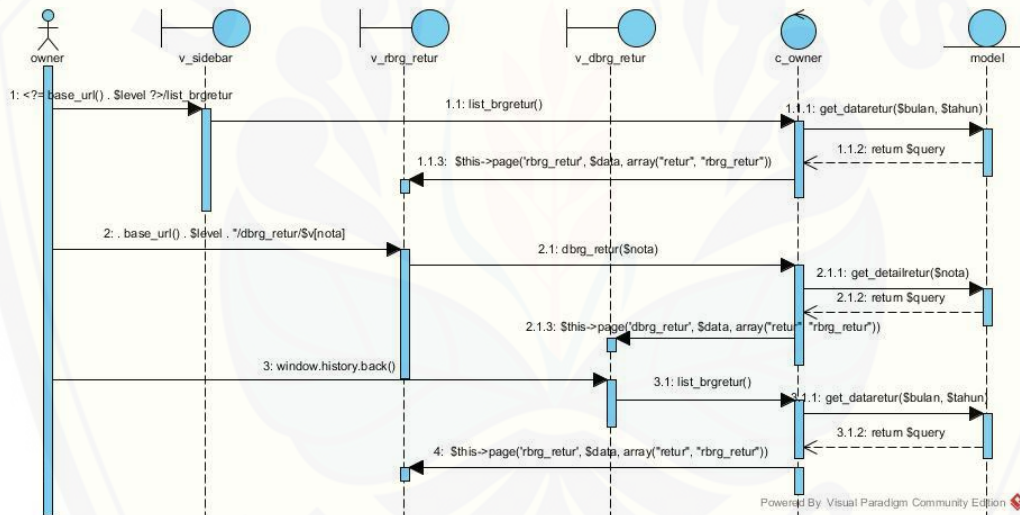


9. Sequence Diagram Melihat Data Barang Retur

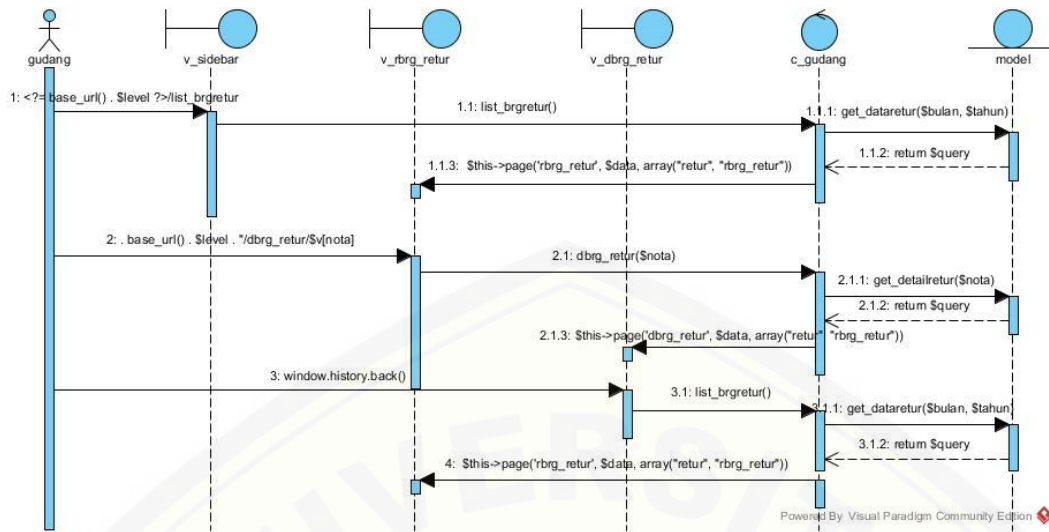
a. Sequence Diagram Melihat Data Barang Retur – Admin



b. Sequence Diagram Melihat Data Barang Retur – Owner

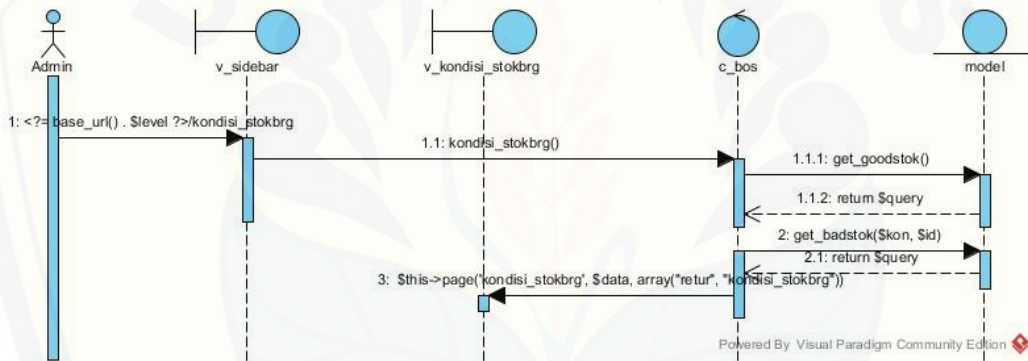


c. *Sequence Diagram* Melihat Data Barang Retur – Bagian Gudang

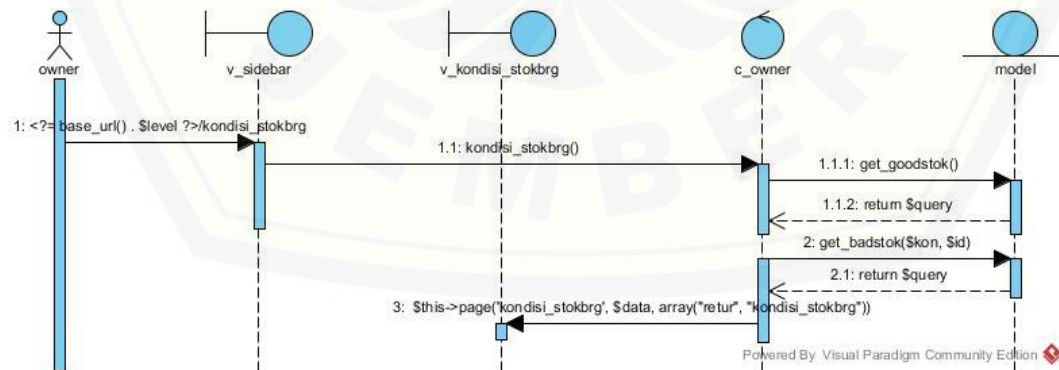


10. *Sequence Diagram* Melihat Kondisi Stok

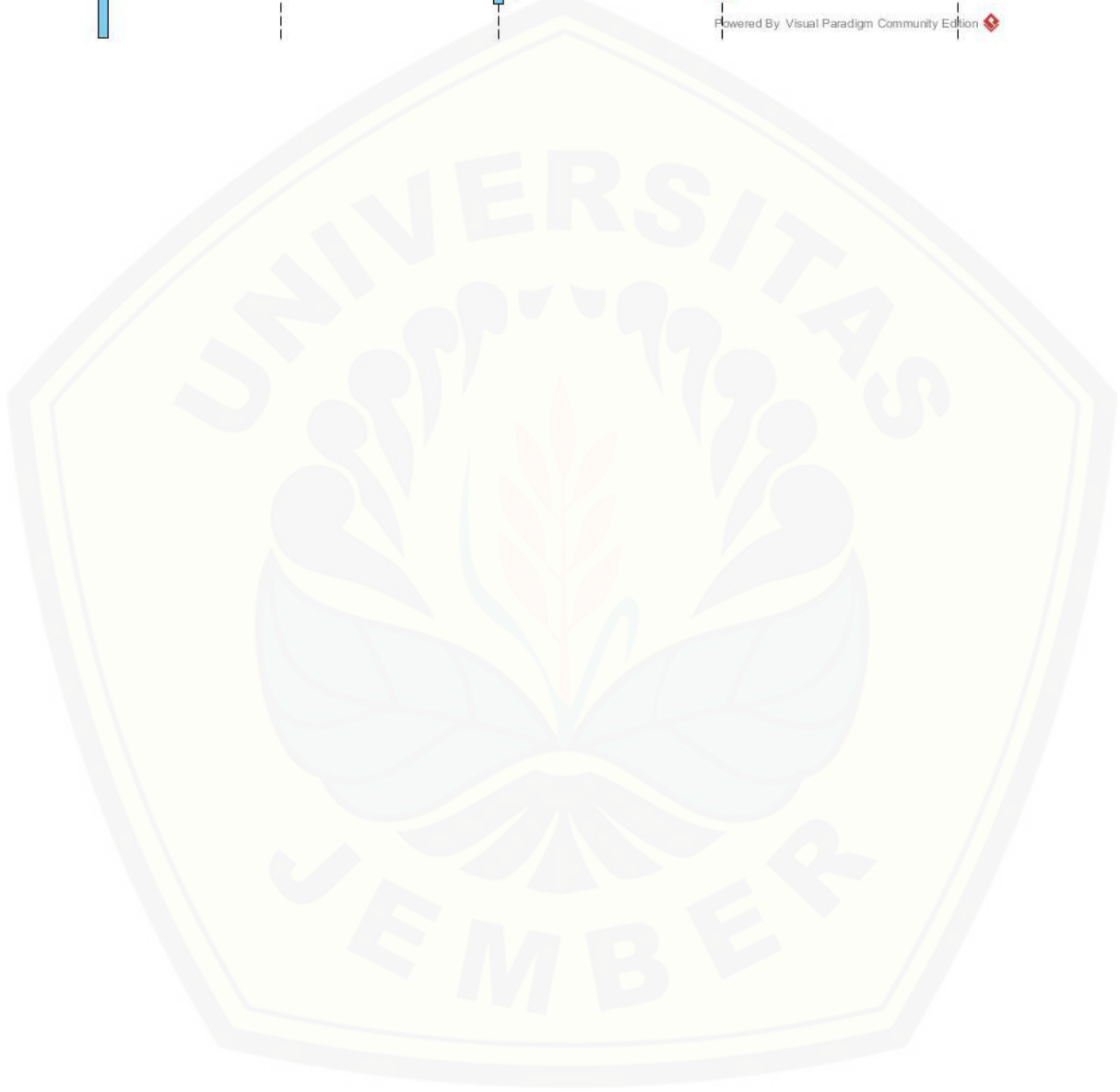
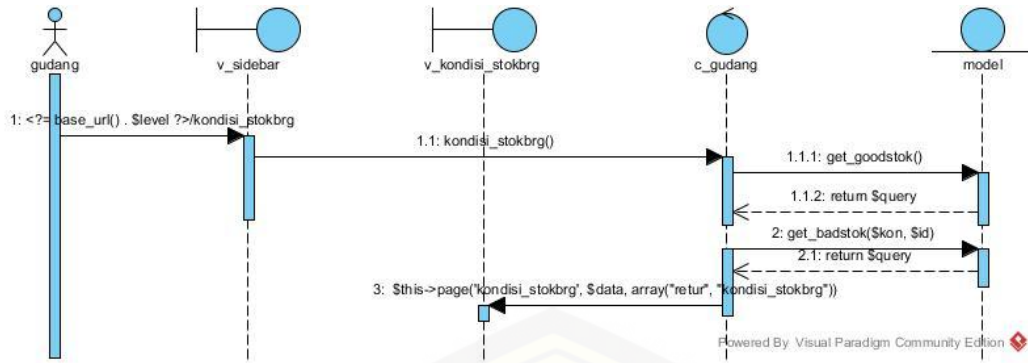
a. *Sequence Diagram* Melihat Kondisi Stok – Admin



b. *Sequence Diagram* Melihat Kondisi Stok – Owner



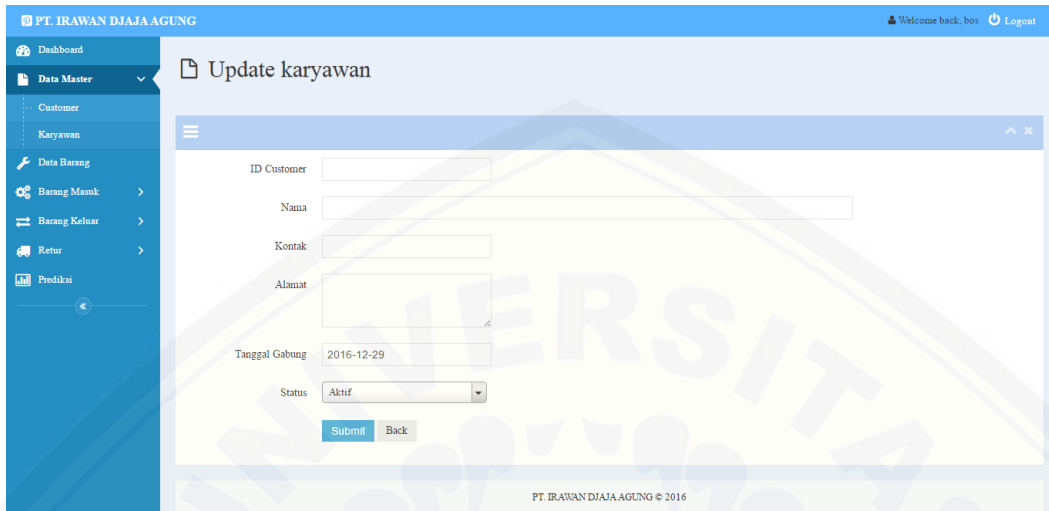
c. *Sequence Diagram* Melihat Kodisi Stok – Bagian Gudang



## LAMPIRAN B. FITUR APLIKASI

### B1. Fitur Karyawan

#### 1. Tambah Data Karyawan



PT. IRAWAN DJAJA AGUNG

Welcome back, bos Logout

Dashboard

Data Master

Customer

Karyawan

Data Barang

Barang Masuk

Barang Keluar

Retur

Prediksi

### Update karyawan

ID Customer

Nama

Kontak

Alamat

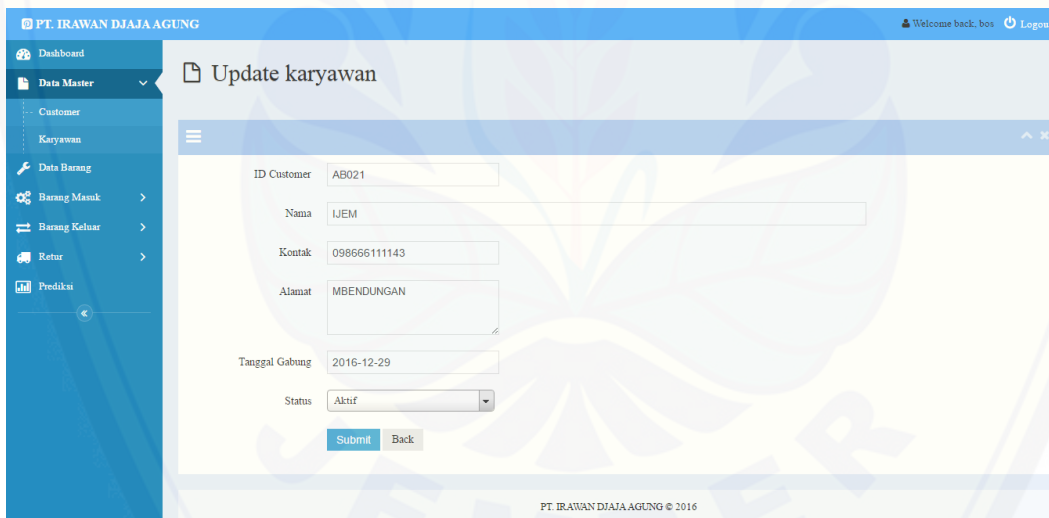
Tanggal Gabung 2016-12-29

Status Aktif

Submit Back

PT. IRAWAN DJAJA AGUNG © 2016

#### 2. Ubah Data Karyawan



PT. IRAWAN DJAJA AGUNG

Welcome back, bos Logout

Dashboard

Data Master

Customer

Karyawan

Data Barang

Barang Masuk

Barang Keluar

Retur

Prediksi

### Update karyawan

ID Customer AB021

Nama IJEM

Kontak 09866611143

Alamat MBENDUNGAN

Tanggal Gabung 2016-12-29

Status Aktif

Submit Back

PT. IRAWAN DJAJA AGUNG © 2016

## B2. Fitur Customer

### 1. Tambah Data Customer

PT. IRAWAN DJAJA AGUNG

Welcome back, bos Logout

Dashboard

Data Master

- Customer
- Karyawan

Data Barang

- Barang Masuk
- Barang Keluar
- Retur
- Prediksi

Update Customer

ID Customer

Nama

Kontak

Alamat

Status

Submit Back

PT. IRAWAN DJAJA AGUNG © 2016

### 2. Ubah Data Customer

PT. IRAWAN DJAJA AGUNG

Welcome back, bos Logout

Dashboard

Data Master

- Customer
- Karyawan

Data Barang

- Barang Masuk
- Barang Keluar
- Retur
- Prediksi

Update Customer

ID Customer

Nama

Kontak

Alamat

Status

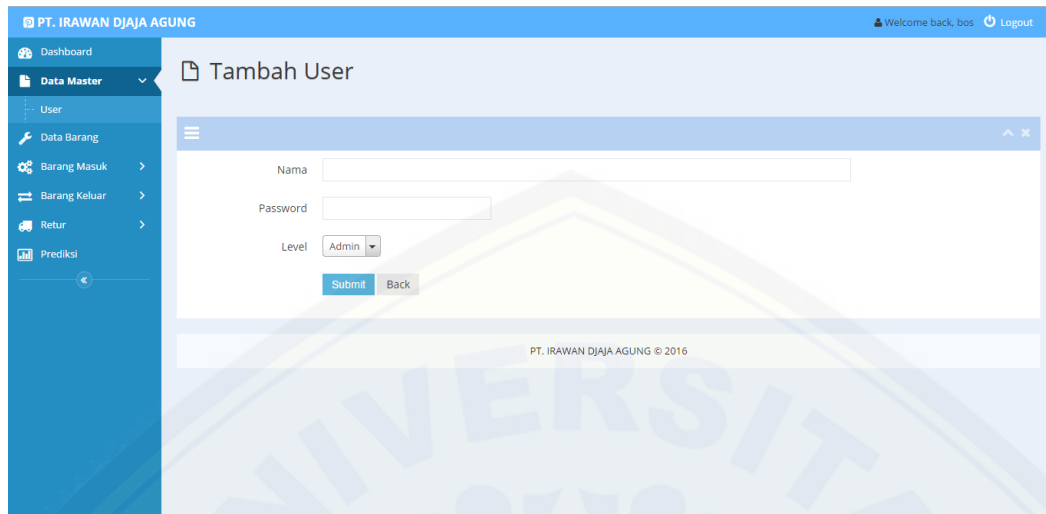
Submit Back

PT. IRAWAN DJAJA AGUNG © 2016

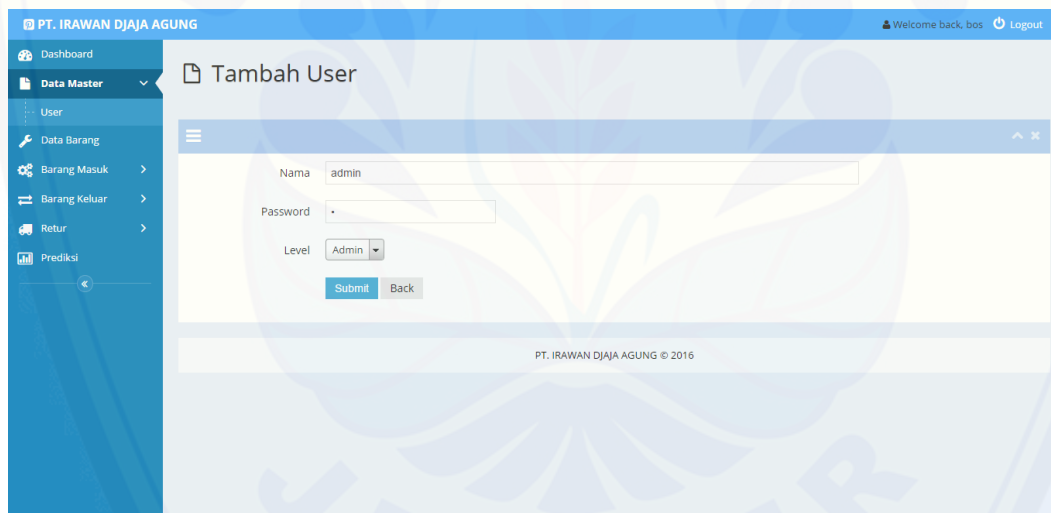


### B3. Fitur Data User

#### 1. Tambah Data User



#### 2. Ubah Data User



## B4. Fitur Data Barang

### 1. Tambah Data Barang

PT. IRAWAN DJAJA AGUNG

Welcome back, bos Logout

Dashboard

Data Master

Data Barang

Barang Masuk

Barang Keluar

Retur

Prediksi

Update barang

ID Barang

Nama

Satuan Pcs

Harga

Stok

Keterangan

Status Aktif

Submit Back

PT. IRAWAN DJAJA AGUNG © 2016

### 2. Ubah Data Barang

PT. IRAWAN DJAJA AGUNG

Welcome back, bos Logout

Dashboard

Data Master

Data Barang

Barang Masuk

Barang Keluar

Retur

Prediksi

Update barang

ID Barang PJ0001AD

Nama Puyer 19 Sakit Kepala Kotak

Satuan Pcs

Harga 38000

Stok 1160

Keterangan

Status Aktif

Submit Back

PT. IRAWAN DJAJA AGUNG © 2016

### B5. Fitur Detail Data Barang Masuk

PT. IRAWAN DJAJA AGUNG Welcome back, bos [Logout](#)

Dashboard

Data Master >

Data Barang >

**Barang Masuk** >

Input Barang Masuk

Data Barang Masuk

Barang Keluar >

Retur >

Prediksi <

Nota : PB161103225701

#	ID Barang	Barang	Harga	Qty	Total
1	PJ0003BD	Balsem Hijau Cap Betet Besar 17.5 gr	54.000.00	250	13.500.000.00
2	PJ0003BP	Balsem Hijau Cap Betet Besar 17.5 gr	4.500.00	1000	4.500.000.00
				1250	18.000.000.00

PT. IRAWAN DJAJA AGUNG © 2016

### B5. Fitur Detail Data Barang Keluar

PT. IRAWAN DJAJA AGUNG Welcome back, bos [Logout](#)

Dashboard

Data Master >

Data Barang >

Barang Masuk >

**Barang Keluar** >

Input Barang Keluar

Data Barang Keluar

Retur >

Prediksi <

Nota : PB161220223154

#	ID Barang	Barang	Harga	Qty	Total
1	PJ0001AD	Puyer 19 Sakrit Kepala Kotak	38.000.00	100	3.800.000.00
				1	3.800.000.00

PT. IRAWAN DJAJA AGUNG © 2016

B8. Fitur Detail Data Barang Retur

PT. IRAWAN DJAJA AGUNG Welcome back, bos [Logout](#)

Dashboard

Data Master >

Data Barang >

Barang Masuk >

Barang Keluar >

**Retur** >

Input Retur

Data Retur

Kondisi Stok Barang

Prediksi <

Nota : PB161212195841

Back

#	ID Barang	Barang	Qty	Harga	Total	Customer	Keterangan
1	PJ0001AD	Puyer 19 Saklit Kepala Kotak	200	38.000.00	7.600.000.00	DJO	Rusak/Cacat
2	PJ0002AK	Bedak Salicyl Spiritus 40ml	450	576.000.00	266.800.000.00	DJO	Rusak/Cacat

PT. IRAWAN DJAJA AGUNG © 2016