



**SISTEM PREDIKSI BARANG PERSEDIAAN MENGGUNAKAN
METODE LEAST SQUARE REGRESSION LINE (STUDI KASUS :
BADAN PENGELOLAAN KEUANGAN DAN ASET DAERAH
KABUPATEN JEMBER)**

SKRIPSI

Oleh

M Fariz Aminulloh

NIM 122410101060

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2019



**SISTEM PREDIKSI BARANG PERSEDIAAN MENGGUNAKAN
METODE LEAST SQUARE REGRESSION LINE (STUDI KASUS :
BADAN PENGELOLAAN KEUANGAN DAN ASET DAERAH
KABUPATEN JEMBER)**

PROPOSAL SKRIPSI

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

Oleh

M Fariz Aminulloh

NIM 122410101060

PROGRAM STUDI SISTEM INFORMASI

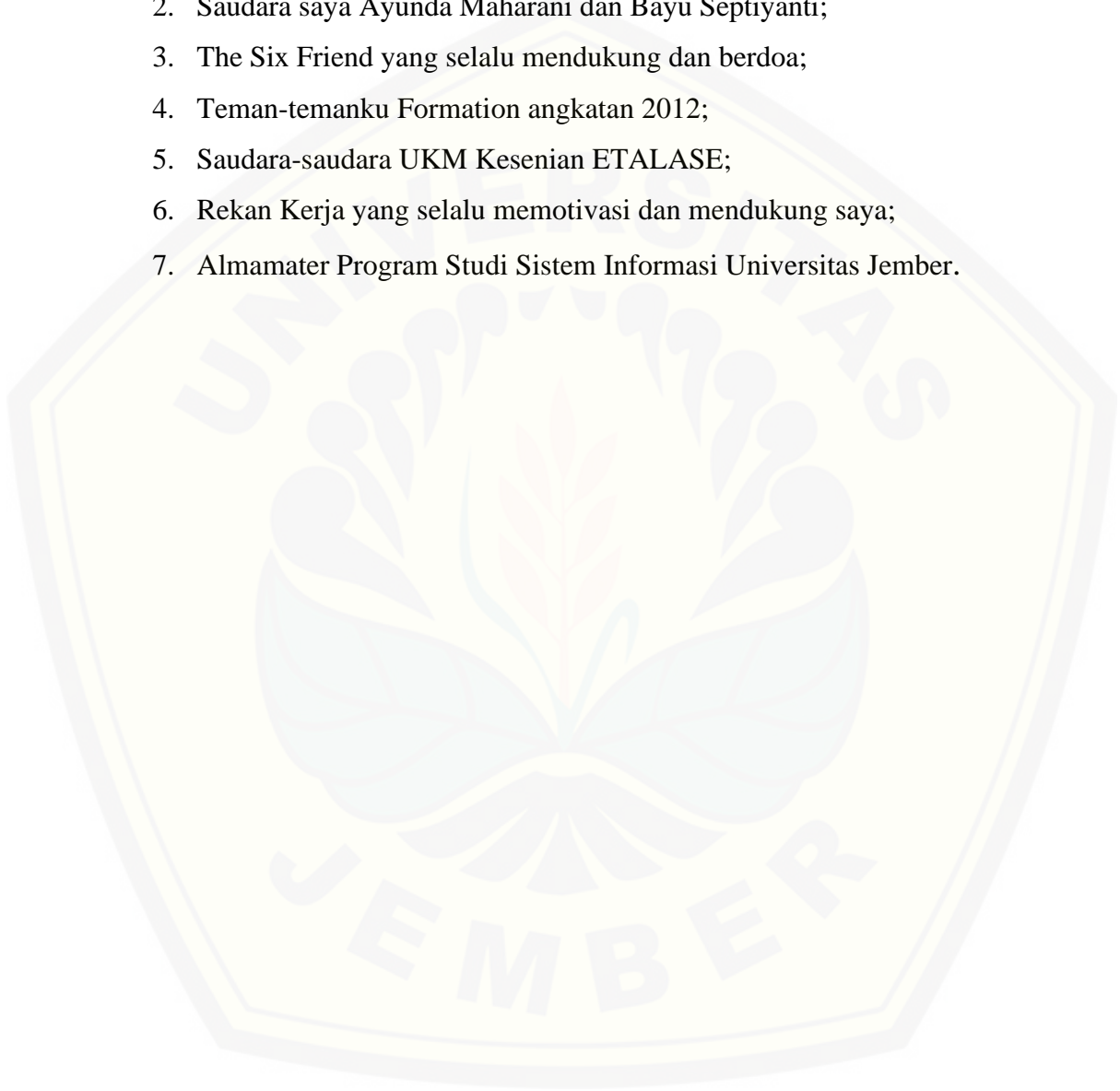
UNIVERSITAS JEMBER

2019

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Ayahanda Bambang Soepono dan Ibunda Tutik Mawarti;
2. Saudara saya Ayunda Maharani dan Bayu Septiyanti;
3. The Six Friend yang selalu mendukung dan berdoa;
4. Teman-temanku Formation angkatan 2012;
5. Saudara-saudara UKM Kesenian ETALASE;
6. Rekan Kerja yang selalu memotivasi dan mendukung saya;
7. Almamater Program Studi Sistem Informasi Universitas Jember.

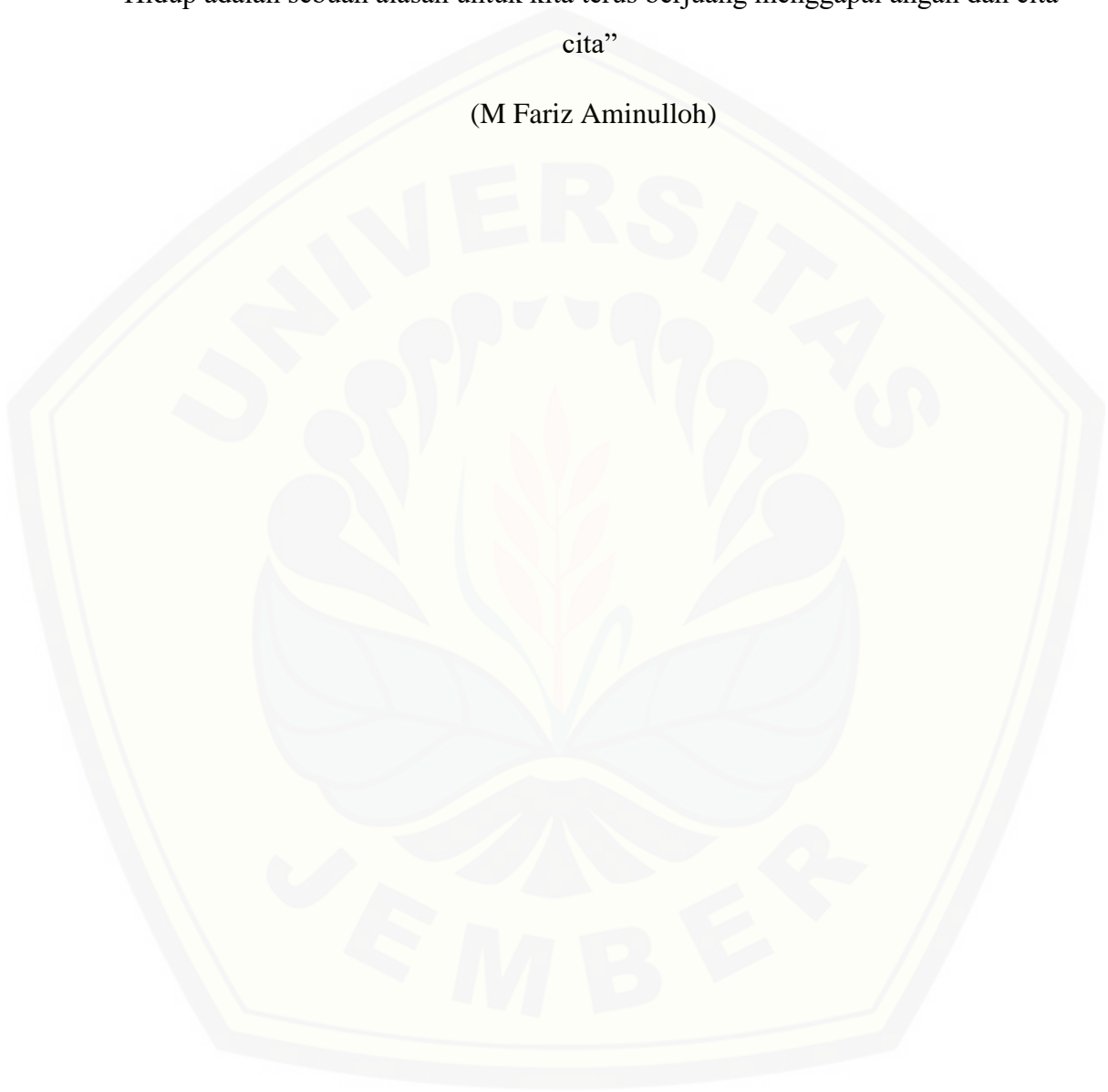


MOTO

“Belajarlh dari mimpi, karena mimpi membuatmu mewujudkan keinginanmu”

“Hidup adalah sebuah alasan untuk kita terus berjuang menggapai angan dan cita-cita”

(M Fariz Aminulloh)



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : M Fariz Aminulloh

NIM : 122410101060

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Prediksi Barang Persediaan Menggunakan *Metode Least Square Regression Line* (Studi Kasus : Badan Pengelolaan Keuangan dan Aset Daerah Kabupaten Jember)”, adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, 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 manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 15 Mei 2018

Yang menyatakan,

M Fariz Aminulloh

NIM 122410101060

SKRIPSI

**SISTEM PREDIKSI BARANG PERSEDIAAN MENGGUNAKAN
METODE LEAST SQUARE REGRESSION LINE (STUDI KASUS :
BADAN PENGELOLAAN KEUANGAN DAN ASET DAERAH
KABUPATEN JEMBER)**

Oleh

M Fariz Aminulloh

NIM 122410101060

Pembimbing :

Dosen Pembimbing Utama : Drs. Antonius Cahya P, M.App., Sc., Ph.D

Dosen Pembimbing Pendamping : Gama Wisnu Fajarianto, S.Kom.,M.Kom

PENGESAHAN PEMBIMBING

Skripsi berjudul “**Sistem Prediksi Barang Persediaan Menggunakan Metode Least Square Regression Line (Studi Kasus : Badan Pengelolaan Keuangan Dan Aset Daerah Kabupaten Jember)**” telah diuji dan disahkan pada:

hari, tanggal : Jumat, 10 Januari 2020

tempat : Fakultas Ilmu Komputer Program Studi Sistem Informasi
Universitas Jember.

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Drs. Antonius Cahya P, M.App., Sc., Ph.D

Gama Wisnu Fajarianto, S.Kom.,M.Kom

NIP. 196909281993021001

NIP. 760015717

PENGESAHAN PENGUJI

Skripsi berjudul “**Sistem Prediksi Barang Persediaan Menggunakan Metode Least Square Regression Line (Studi Kasus : Badan Pengelolaan Keuangan Dan Aset Daerah Kabupaten Jember)**” telah diuji dan disahkan pada:

hari, tanggal : Jumat, 10 Januari 2020

tempat : Fakultas Ilmu Komputer Program Studi Sistem Informasi
Universitas Jember.

Tim Penguji,

Penguji I,

Penguji II,

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

NIP. 196906151997021002

Dwiretno Istiyadi S, ST.,M.Kom

NIP. 197803302003121003

Mengesahkan

Dekan Fakultas Ilmu Komputer,

Prof. Dr. Saiful Bukhori, ST., M.Kom

NIP. 196811131994121001

RINGKASAN

Prediksi merupakan alat bantu yang penting dalam perencanaan yang efektif dan efisien khususnya dalam bidang ekonomi. Dalam organisasi pemerintahan mengetahui keadaan yang akan datang tidak saja penting, tetapi juga mampu untuk mengantisipasi berlebihnya penganggaran dalam barang persediaan. Sehingga anggaran yang berlebih tersebut dapat di alokasikan ke program kerja lainnya. Badan Pengelolaan Keuangan dan Aset Daerah Kabupaten Jember adalah institusi dibawah Pemerintah Kabupaten Jember yang bertugas untuk menyusun, memantau mengevaluasi dan melaporkan pelaksanaan tugas teknis di Badan Pengelolaan Keuangan dan Aset Daerah. Badan Pengelolaan Keuangan dan aset memiliki tanggung jawab untuk menyediakan kebutuhan barang persediaan untuk setiap bidang yang terdapat di Badan Pengelolaan Keuangan dan Aset Daerah. Proses pengadaan barang persediaan memiliki 4 tahapan mulai dari perencanaan, penganggaran, realisasi, dan pencatatan. Proses perencanaan barang persediaan berpatokan pada perencanaan yang sudah dilaksanakan pada tahun sebelumnya. Hasil dari proses perencanaan akan digunakan untuk melakukan penganggaran yang akan menjadi usulan untuk APBD. Setelah mendapat pengesahan proses realisasi dijalankan sesuai kebutuhan setiap bulan. Proses pencatatan dilakukan setiap semester berdasarkan aktifitas pemasukan dan pengeluaran barang persediaan pada setiap bulan. Pada Badan Pengelolaan Keuangan dan aset daerah kabupaten jember untuk penganggaran barang persediaan sendiri masih menganut data dari penganggaran sebelumnya. Hal tersebut mengakibatkan anggaran untuk Barang Persediaan dapat menjadi lebih atau bahkan kurang. Terdapat banyak metode untuk melakukan prediksi salah satunya adalah metode Least Square Regression Line. Pada proyek ini metode Least Square Regression Line (LSRL) di implementasikan di dalam sistem persediaan barang yang berbasis website. Barang persediaan yang keluar yang tercatat akan di hitung dengan menggunakan metode Least Square Regression Line, dan hasil dari perhitungan tersebut akan menunjukkan kebutuhan barang persediaan untuk bulan selanjutnya.

PRAKATA

Puji syukur kepada Tuhan Allah SWT, atas segala limpahan rahmat, hidayat dan karuniaNya maka penulis dapat menyelesaikan skripsi yang berjudul **“Sistem Prediksi Barang Persediaan Menggunakan Metode Least Square Regression Line (Studi Kasus : Badan Pengelolaan Keuangan Dan Aset Daerah Kabupaten 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 dukungan dan bantuan berbagai pihak. Oleh karena itu penulis menyampaikan terima kasih kepada:

1. Prof. Dr. Saiful Bukhori, ST., M.Kom selaku Dekan Fakultas Ilmu Komputer Universitas Jember;
2. Drs. Antonius Cahya P, M.App., Sc., Ph.D., selaku Dosen Pembimbing Utama dan Gama Wisnu Fajarianto, S.Kom.,M.Kom., selaku Dosen Pembimbing Pendamping yang telah meluangkan waktu, pikiran dan perhatian dalam penulisan skripsi;
3. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
4. Ayahanda Bambang Soepono dan Ibunda Tutik Mawarti;
5. Saudara saya Ayunda Maharani dan Bayu Septiyanti;
6. Teman-teman seperjuangan FORMATION angkatan 2012 dan semua mahasiswa Program Studi Sistem Informasi yang telah menjadi keluarga kecil bagi penulis selama menempuh pendidikan S1;
7. The Six Friendly yang selalu mendukung dan mendoakan, Bayu, Taufiq, Azhar, Spica, dan Rizky;
8. Keluarga besar Unit Kegiatan Mahasiswa Kesenian (UKMK) ETALASE periode 2014-2015, Ocha, Bagus, Glend, Yohan, Dinda, Alfi, dan anggota lain yang tidak dapat disebutkan satu persatu;
9. Semua pihak yang tidak dapat disebutkan satu persatu;

Dengan harapan bahwa penelitian ini nantinya akan terus berlanjut dan berkembang kelak, penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, 15 Mei 2019

Penulis



DAFTAR ISI

PERSEMBAHAN.....	i
MOTO.....	ii
PERNYATAAN	iii
SKRIPSI.....	iv
PENGESAHAN PEMBIMBING	v
PENGESAHAN PENGUJI.....	vi
RINGKASAN.....	vii
PRAKATA.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xv
BAB. 1 PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Tujuan dan Manfaat	3
1.3.1. Tujuan	3
1.3.2. Manfaat	3
1.4. Batasan Masalah	3
1.5. Sistematika Penulisan	4
BAB 2. TINJAUAN PUSTAKA	5
2.1. Penelitian Terdahulu	5
2.2. Persediaan	6
2.3. Peramalan (<i>Forecasting</i>).....	7
2.4. Metode <i>Least Square Regresion Line</i> (LSRL).....	8
2.5. <i>Data Time Series</i>	9
2.6. Autokorelasi	9
2.7. Pola Data.....	10
BAB 3. METODOLOGI PENELITIAN	13
3.1 Jenis Penelitian.....	13
3.2 Metode Penelitian	13
3.3 Waktu dan Tempat Penelitian.....	13

3.4	Tahapan Penelitian.....	13
3.4.1	Tahapan Analisis Kebutuhan (<i>Requirment Definition</i>).....	14
3.4.2	Tahapan Desain Sistem (<i>System and Software Design</i>).....	14
3.4.3	Tahapan Implementasi Sistem (<i>System Implementation</i>)	17
3.4.4	Tahapan Pengujian Sistem (<i>System Testing</i>)	17
3.4.5	Tahapan Pemeliharaan Sistem (<i>Maintanance</i>)	17
3.5	Gambaran Umum Sistem.....	18
BAB 4. PENGEMBANGAN SISTEM.....		19
4.1	Tahapan Analisis Kebutuhan	19
4.1.1	Penerapan <i>Least Square Regression Line</i>	19
4.1.2	SOP (<i>statement of purpose</i>)	19
4.1.3	Kebutuhan Fungsional	20
4.1.4	Kebutuhan Non-Fungsional	20
4.2	Desain Sistem.....	21
4.2.1	<i>Business Proces</i>	21
4.2.2	<i>Usecase Diagram</i>	22
4.2.3	<i>Scenario Diagram</i>	24
4.2.4	<i>Activity Diagram</i>	34
4.2.5	<i>Sequence Diagram</i>	37
4.2.6	<i>Class Diagram</i>	41
4.2.7	<i>Entity Relationship Diagram (ERD)</i>	42
4.3	Implementasi.....	43
4.3.1.	Kode Program Peramalan	43
4.3.2.	Kode Program Melihat prediksi kebutuhan barang persediaan (<i>class controllers/DataPakai</i>)	45
4.4	Pengujian.....	46
4.4.1	Metode <i>White Box</i>	47
4.4.2	Metode <i>Black Box</i>	50
BAB 5. HASIL DAN PEMBAHASAN		53
5.1	Hasil Pengembangan Sistem.	53
5.1.1	Tampilan Awal Sistem.....	53
5.1.2	Fitur Mengelola Data Pemasukan	54
5.1.3	Fitur Mengelola Data Pengeluaran	56
5.1.4	Fitur Melihat Prediksi Barang.....	58

5.1.5	Fitur Melihat Rekap Keseluruhan	58
5.2	Implementasi Metode <i>Least Square Regression Line</i>	59
BAB 6. PENUTUP		62
6.1	Kesimpulan	62
6.2	Saran	63
DAFTAR PUSTAKA		64
LAMPIRAN.....		65
LAMPPIRAN A. <i>ACTIVITY DIAGRAM</i>		65
A.1	<i>Activity Diagram</i> Mengelola Data Pemasukan	65
A.2	<i>Activity Diagram</i> Mengelola Data Pengeluaran.....	66
A.3	<i>Activity Diagram</i> Melihat Rekap Keseluruhan	67
A.4	<i>Activity Diagram</i> Logout	67
LAMPPIRAN B. <i>SEQUENCE DIAGRAM</i>		68
B.1	<i>Sequence Diagram</i> Mengelola Data Transaksi Pemasukan.....	68
B.2	<i>Sequence Diagram</i> Mengelola Data Transaksi Pengeluaran	69
B.3	<i>Sequence Diagram</i> Melihat Prediksi Kebutuhan Barang Persediaan	70
LAMPPIRAN C. KODE PROGRAM		71
C.1	Kode Program Login (<i>class controllers/login</i>)	71
C.2	Kode Program Login (<i>class models/m_login</i>)	72
C.3	Kode Program Mengelola Data Transaksi Pemasukan (<i>class controllers/c_barang_masuk</i>)	73
C.4	Kode Program Mengelola Data Transaksi Pemasukan (<i>class modelss/m_masuk</i>)	76
C.5	Kode Program Mengelola Data Transaksi Pengeluaran (<i>class controllerss/c_barang_keluar</i>).....	79
C.6	Kode Program Mengelola Data Transaksi Pengeluaran (<i>class modelss/m_keluar</i>)	80
C.7	Kode Program Melihat Prediksi Kebutuhan Barang Persediaan (<i>class controllers/c_regresi</i>)	82
C.8	Kode Program Melihat Rekap Keseluruhan (<i>class controllers/c_rekap</i>)	83
C.9	Kode Program Melihat Rekap Keseluruhan (<i>class models/m_rekap</i>)	84

DAFTAR GAMBAR

Gambar 3. 1 Tahapan metode <i>Waterfall</i> (Basuki, 2006)	14
Gambar 4. 1 Business Proses <i>Sistem</i>	22
Gambar 4. 2 Usecase diagram.....	23
Gambar 4. 3 <i>Activity Diagram Login</i>	35
Gambar 4. 4 Activity Diagram Memprediksi Kebutuhan Barang Persediaan	36
Gambar 4. 5 Sequence Diagram Login	38
Gambar 4. 6 <i>Sequence Diagram</i> melihat rekap keseluruhan	40
Gambar 4. 7 <i>Sequence diagram Logout</i>	41
Gambar 4. 8 <i>Class Diagram</i>	42
Gambar 4. 9 <i>Entity Relationship Diagram</i>	42
Gambar 4. 10 Diagram alir <i>fuction</i> hitung(\$id)	49
Gambar 5. 1 Tampilan halaman <i>login</i>	53
Gambar 5. 2 Tampilan halaman Pemasukan.....	54
Gambar 5. 3 Fitur <i>tambah barang masuk</i>	54
Gambar 5. 4 Edit Data Barang Masuk	55
Gambar 5. 5 Detail Barang Masuk.....	55
Gambar 5. 6 Restock Barang Persediaan	56
Gambar 5. 7 Edit Detail Stock Barang.....	56
Gambar 5. 8 Tampilan Mengelola Data Pengeluaran	57
Gambar 5. 9 Tambah Barang Keluar	57
Gambar 5. 10 Tambah Detail Barang Keluar.....	57
Gambar 5. 11 Memilih Barang Yang Ingin di Prediksi	58
Gambar 5. 12 Hasil Prediksi Barang Persediaan	58
Gambar 5. 13 Tampilan Rekap Keseluruhan	59
Gambar 5. 14 Pola data barang persediaan bolpoint tahun 2019.....	59
Gambar A. 1 <i>Activity Diagram</i> Mengelola Data Pemasukan	65
Gambar A. 2 <i>Activity Diagram</i> Mengelola Data Pengeluaran.....	66
Gambar A. 3 <i>Activity Diagram</i> Melihat Rekap Keseluruhan	67
Gambar A. 4 <i>Activity Diagram Logout</i>	67

Gambar B. 1 *Sequence Diagram* Mengelola Data Transaksi Pemasukan 68
Gambar B. 2 *Sequence Diagram* Mengelola Data Transaksi Pengeluaran..... 69
Gambar B. 3 *Sequence Diagram* Melihat Prediksi Kebutuhan Barang Persediaan
..... 70



DAFTAR TABEL

Tabel 4. 1 Deskripsi pembagian aktor sistem	23
Tabel 4. 2 Deskripsi <i>use case</i> sistem.....	24
Tabel 4. 3 <i>Scenario Login</i>	25
Tabel 4. 4 <i>Scenario</i> Mengelola Data Transaksi Pemasukan	26
Tabel 4. 5 Skenario Mengelola Data Transaksi Pengeluaran	30
Tabel 4. 6 <i>Scenario</i> melihat prediksi kebutuhan barang persediaan.....	32
Tabel 4. 7 <i>Scenario</i> Melihat Rekap Keseluruhan.....	33
Tabel 4. 8 <i>Scenario logout</i>	34
Tabel 4. 9 Kode Metode <i>Least Square Regression Line</i> hitung(\$id).....	43
Tabel 4. 10 <i>Listing program function</i> hitung(\$id)	47
Tabel 4. 11 Pengujian <i>Black Box</i> Prediksi Kebutuhan Barang Persediaan	50
Tabel 5. 1 Perhitungan nilai analisis metode <i>least square</i>	60
Tabel C. 1 Kode Program <i>Login (class controllers/login)</i>	71
Tabel C. 2 Kode Program <i>Login (class models/m_login)</i>	72
Tabel C. 3 Kode Program Mengelola Data Transaksi Pemasukan (<i>class controllers/c_barang_masuk</i>)	73
Tabel C. 4 Kode Program Mengelola Data Transaksi Pemasukan (<i>class modelss/m_masuk</i>)	76
Tabel C. 5 Kode Program Mengelola Data Transaksi Pengeluaran (<i>class controllerss/c_barang_keluar</i>).....	79
Tabel C. 6 Kode Program Mengelola Data Transaksi Pengeluaran(<i>class modelss/m_keluar</i>)	80
Tabel C. 7 Kode Program Melihat Prediksi Kebutuhan Barang Persediaan(<i>class controllers/c_regresi</i>).....	82
Tabel C. 8 Kode Program Melihat Rekap Keseluruhan (<i>class controllers/c_rekap</i>)	83
Tabel C. 9 Kode Program Melihat Rekap Keseluruhan (<i>class models/m_rekap</i>). 84	

BAB. 1 PENDAHULUAN

Bab ini merupakan bab awal dari laporan tugas akhir. Pada bab ini akan dibahas tentang latar belakang, perumusan masalah, tujuan dan manfaat, batasan masalah, dan sistematika penulisan.

1.1. Latar Belakang

Pada era komputerisasi ini kebutuhan manusia akan informasi memacu pesatnya perkembangan teknologi di bidang informasi dan teknologi telekomunikasi. Teknologi yang semakin meningkat didukung dengan sarana dan prasarana yang memadai, membuktikan bahwa kini informasi telah menjadi kebutuhan pokok dalam kehidupan manusia. Sistem informasi dan teknologi informasi berfungsi sebagai pendukung untuk mengambil keputusan yang tepat berdasarkan informasi yang tersedia. Pada perkembangan teknologi informasi yang ada saat ini dapat melakukan pengolahan data dengan mudah, dapat menghasilkan suatu informasi yang dibutuhkan dengan akurat dan mengefektifkan waktu, serta biaya yang di keluarkan lebih efisien.

Persediaan barang ialah suatu aktiva lancar yang meliputi barang-barang yang merupakan milik perusahaan (Sofyan Assauri, 2005). Persediaan adalah bahan atau barang yang disimpan dan digunakan untuk memenuhi kebutuhan tertentu, misalnya untuk digunakan dalam proses produksi atau untuk dijual kembali. Pengendalian persediaan merupakan sebuah aset penting dalam sebuah perusahaan. Tujuan utama dari pengendalian persediaan adalah untuk menjaga tingkat persediaan suatu barang dan menjaga tingkat ketersediaan barang.

Badan Pengelolaan Keuangan dan Aset Daerah Kabupaten Jember adalah institusi dibawah Pemerintah Kabupaten Jember yang bertugas untuk menyusun, memantau mengevaluasi dan melaporkan pelaksanaan tugas teknis di Badan Pengelolaan Keuangan dan Aset Daerah. Badan Pengelolaan Keuangan dan aset memiliki tanggung jawab untuk menyediakan kebutuhan barang persediaan untuk setiap bidang yang terdapat di Badan Pengelolaan Keuangan dan Aset Daerah.

Proses pengadaan barang persediaan memiliki 4 tahapan mulai dari perencanaan, penganggaran, realisasi, dan pencatatan. Proses perencanaan barang persediaan berpatokan pada perencanaan yang sudah dilaksanakan pada tahun sebelumnya. Hasil dari proses perencanaan akan digunakan untuk melakukan penganggaran yang akan menjadi usulan untuk APBD. Setelah mendapat pengesahan proses realisasi dijalankan sesuai kebutuhan setiap bulan. Proses pencatatan dilakukan setiap semester berdasarkan aktifitas pemasukan dan pengeluaran barang persediaan pada setiap bulan. Selama ini Badan Pengelolaan Keuangan dan Aset Daerah Kabupaten Jember melakukan kegiatan pencatatan ketersediaan barang secara manual. Sehingga mengakibatkan barang yang ada sering kali tidak sama dengan barang yang tercatat. Hal ini menimbulkan data yang tidak selaras antara kenyataan dengan pencatatan dan membuat pencatatan akhir tahun menjadi tidak sesuai. Untuk menjaga keselarasan tersebut diperlukan sebuah sistem yang mampu mengelola pencatatan sekaligus memprediksi kebutuhan barang persediaan serta mempermudah pelaporan akhir tahun.

Pengembangan sistem ini menggunakan metode *Least Square Regression Line*. Pemilihan metode *Least Square Regression Line* didasarkan karena metode ini dinilai cukup baik untuk diterapkan pada Proses Peramalan Persediaan Barang serta memiliki tingkat keakuratan yang baik. Prediksi permintaan atau penjualan menggunakan metode (LSRL) akan lebih akurat apabila data yang digunakan memiliki pola tren (Setiawan, 2013). Metode *Least Square* dinilai cukup baik karena metode ini memanfaatkan data yang diperoleh dari masa lalu untuk menentukan kemungkinan kebutuhan pada periode selanjutnya, serta dimungkinkan untuk mengolah kebutuhan data yang memiliki rentan waktu yang berbeda. Penggunaan pola data tren semakin menunjang tingkat keakuratan dari penggunaan metode *Least Square Regression Line*.

1.2. Rumusan Masalah

Bedasarkan latar belakang yang ada, maka didapatkan perumusan masalah sebagai berikut.

1. Bagaimana menerapkan metode *Least Square Regression Line* untuk prediksi barang persediaan pada Badan Pengelolaan Keuangan dan Aset Kabupaten Jember?
2. Bagaimana merancang dan membangun sebuah sistem prediksi barang persediaan dengan Metode *Least Square Regression Line*?

1.3. Tujuan dan Manfaat

Berdasarkan rumusan masalah yang telah disampaikan sebelumnya, maka tujuan dan manfaat yang ingin dicapai sebagai berikut:

1.3.1. Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah :

- a. Memprediksi barang persediaan untuk periode selanjutnya.
- b. Mempermudah pelaporan barang persediaan saat akhir tahun..

1.3.2. Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

- a. Bagi akademis

Hasil penelitian ini diharapkan dapat memberikan pandangan dan masukan bagi siapa saja yang membutuhkan informasi yang berhubungan dengan judul penelitian ini. Selain itu, hasil penelitian ini merupakan upaya untuk menambah perbendaharaan judul penelitian yang ada di fakultas ilmu komputer Universitas Jember.

- b. Bagi peneliti

Mengetahui bagaimana proses penerapan *Least Square Regression Line* pada sistem prediksi barang persediaan Badan Pengelolaan Keuangan dan Aset Daerah Kabupaten Jember

- c. Bagi pihak lain

Penelitian ini dapat dijadikan bahan referensi bagi peneliti lain yang ingin mengembangkan hasil penelitian ini di kemudian hari.

1.4. Batasan Masalah

Terdapat beberapa batasan masalah yang diangkat dalam penelitian ini, antara lain sebagai berikut :

- a. Lingkup penelitian adalah Badan Pengelolaan Keuangan dan Aset Daerah Kabupaten Jember,
- b. Sistem ini menggunakan metode *Least Square Regression Line*.

1.5. Sistematika Penulisan

Adapun sistematika penulisan skripsi setelah Bab 1. Pendahuluan adalah sebagai berikut:

- a. Bab 2. Tinjauan Pustaka

Bab ini memaparkan tinjauan terhadap hasil-hasil penelitian terdahulu berkaitan dengan masalah yang dibahas, landasan materi dan konsep prediksi, dan kajian teori metode analisis data yang berkaitan dengan masalah dalam penelitian.

- b. Bab 3. Metodologi Penelitian

Bab ini menguraikan tentang tempat dan waktu penelitian, metode penelitian, metode pengumpulan data, metode analisis data, dan teknik pengembangan sistem yang digunakan dalam penelitian.

- c. Bab 4. Pengembangan Sistem

Bab ini berisi uraian tentang langkah-langkah yang dilakukan dalam proses menganalisis dan merancang sistem yang hendak dibangun meliputi desain, kode program, dan pengujian sistem.

- d. Bab 5. Hasil dan Pembahasan

Bab ini memaparkan secara rinci pemecahan masalah melalui analisis yang disajikan dalam bentuk deskripsi dibantu dengan ilustrasi berupa tabel dan gambar untuk memperjelas hasil penelitian.

- e. Bab 6. Penutup

Bab ini terdiri atas kesimpulan atas penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya.

BAB 2. TINJAUAN PUSTAKA

Pada bagian ini dipaparkan tinjauan yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah, dan juga penelitian-penelitian terdahulu.

2.1. Penelitian Terdahulu

Penelitian terdahulu mengangkat permasalahan mengenai ketidakpastian dalam merencanakan persediaan obat pada tahun berikutnya yang menyebabkan tidak terkontrolnya pemesanan untuk persediaan obat. Permasalahan tersebut berusaha diselesaikan dengan membangun sebuah aplikasi untuk memprediksi tingkat persediaan obat-obatan pada tahun berikutnya berdasarkan data penjualan tahun sebelumnya (Rambe, 2014). Metode yang digunakan untuk memprediksi tingkat persediaan obat-obatan yaitu metode *Least Square* dan untuk meminimumkan kesalahan meramal (*forcast error*) menggunakan *Mean Absolute Deviation* (MAD). Penelitian yang dilakukan mendapatkan hasil prediksi pada tahun 2013 penjualan kalpanax cair sebesar 149. Hasil tersebut didapatkan dari data penjualan obat kalpanax cair pada tahun 2012 sebesar 131. Hasil dari penelitian ini adalah dapat memprediksi tingkat penjualan obat-obatan pada tahun berikutnya berdasarkan data penjualan tahun sebelumnya dengan kesalahan meramal yang minimum. Kekurangan pada penelitian ini adalah tidak menguji pola data terlebih dahulu dan hanya dapat memprediksi hasil produksi pertahun. Metode yang digunakan belum tentu sesuai dengan pola data yang dimiliki. Perbaikan pada penelitian selanjutnya adalah melakukan uji data terlebih dahulu sebelum menentukan metode prediksi yang tepat dan dapat memprediksi perhari.

Penelitian lain mencoba menyelesaikan permasalahan yang sering terjadi yaitu mengenai jumlah persediaan tidak sesuai dengan permintaan yang menyebabkan kekurangan atau kelebihan persediaan. Permasalahan tersebut juga menyebabkan kerugian pada perusahaan karena biaya yang dikeluarkan tidak terencana secara pasti. Permasalahan tersebut diselesaikan menggunakan metode

Least Square Regresion Line untuk memprediksi permintaan dan menggunakan metode *Economic Order Quantity* untuk menentukan kuantitas pesanan optimal dan salah satu asumsi pada metode tersebut adalah jumlah permintaan atau penjualan harus diketahui (Setiawan, 2013). Dari penelitian tersebut dikemukakan bahwa biaya persediaan dapat di minimalisir sebesar 20.27% pada data penjualan tahun 2012. Hasil dari penelitian ini menunjukkan bahwa metode LSRL dapat menentukan jumlah pemesanan secara optimal. Prediksi permintaan atau penjualan menggunakan metode LSRL akan lebih akurat apabila data yang digunakan memiliki pola tren. Kekurangan pada penelitian ini adalah prediksi yang dihasilkan hanya pertahun, sehingga tidak bisa memprediksi hasil produksi perhari maupun perbulan. Perbaikan pada penelitian selanjutnya adalah aplikasi dapat memprediksi hasil produksi perhari serta dapat menampilkan grafik penjualan perbulan.

Penelitian lain menjelaskan bahwa mengolah Data surat pesanan yang berasal dari apotik dijadikan input yang kemudian diprediksi dengan data penjualan yang dimiliki oleh distributor. Hasil dari penelitian tersebut disimpulkan bahwa persediaan *albothyl Concentrate* 10ml untuk bulan yang akan datang diprediksi dengan angka di bawah atau di atas 19.662. Namun jika persamaan garis regresi bernilai minus maka hasil prediksinya akan dianggap 0. (Sulardi, dkk 2017).

2.2. Persediaan

Persediaan barang ialah suatu aktiva lancar yang meliputi barang-barang yang merupakan milik perusahaan dengan sebuah maksud supaya dijual dalam suatu periode usaha normal ataupun persediaan barang-barang yang masih dalam pekerjaan sebuah proses produksi maupun persediaan bahan baku yang juga menunggu penggunaannya di dalam suatu proses produksi. (Sofyan Assauri, 2005).

Menurut kebijakan akuntansi barang persediaan adalah aset lancar dalam bentuk barang atau perlengkapan yang dimaksudkan untuk mendukung kegiatan operasional pemerintah daerah dan barang-barang yang dimaksudkan untuk dijual dan atau/ diserahkan dalam rangka pelayanan kepada masyarakat.

Barang persediaan diklasifikasikan menjadi 4 jenis yaitu:

1. Barang atau perlengkapan yang digunakan dalam rangka kegiatan operasional pemerintah daerah, misalnya barang pakai habis seperti alat tulis kantor, barang tak habis pakai seperti komponen peralatan dan pida, dan barang bekas pakai seperti komponen bekas.
2. Bahan atau perlengkapan yang digunakan dalam proses produksi, misalnya bahan baku pembuatan alat pertanian, dll
3. Barang dalam proses produksi yang dimaksudkan untuk dijual atau diserahkan kepada masyarakat, misalnya alat-alat pertanian setengah jadi, benih yang belum cukup umur.
4. Barang yang disimpan untuk dijual atau diserahkan kepada masyarakat dalam rangka kegiatan pemerintahan, misalnya hewan dan bibit tanaman.

2.3. Peramalan (*Forecasting*)

Peramalan (*forecasting*) merupakan bagian vital bagi setiap organisasi bisnis dan untuk setiap pengambilan keputusan manajemen yang sangat signifikan. Peramalan merupakan hal yang sangat penting dalam perencanaan jangka panjang pada suatu perusahaan. Hampir semua peramalan didasarkan pada asumsi bahwa masa lampau akan berulang (Padang et al, 2013).

Berdasarkan teknik yang digunakan, metode peramalan dapat digolongkan menjadi dua kategori, yaitu: (Hakimah, Muhima, & Yustina, 2015)

1. Metode peramalan kualitatif

Peramalan kualitatif adalah metode peramalan yang berdasarkan dari data kualitatif masa lalu. Hasil peramalan yang dibuat bergantung kepada orang yang menyusun.

2. Metode peramalan kuantitatif

Permalan kuantitatif adalah peramalan yang berdasarkan atas data kuantitatif masa lalu. Hasil peramalan yang dibuat bergantung pada metode yang digunakan. Dengan metode yang berbeda maka hasil yang diramalkan akan berbeda. Semakin

kecil nilai *error* pada suatu metode semakin baik metode tersebut digunakan untuk peramalan.

Metode kuantitatif dapat diterapkan apabila terdapat kondisi sebagai berikut :

- a. Tersedia informasi (data) tentang masa lalu.
- b. Informasi tersebut dapat dikuantitatifkan dalam bentuk angka.

Dapat diasumsikan pola data masa lalu akan terjadi kembali pada masa mendatang.

2.4. Metode *Least Square Regresion Line* (LSRL)

Metode *Least Square Regresion Line* (LSRL) merupakan metode peramalan penjualan analisa tren linier. Pola data masa lalu yang dimiliki harus berpola tren untuk mendapatkan prediksi yang akurat. Terdapat 2 (dua) cara yaitu untuk jumlah data genap dan untuk jumlah data ganjil. Penerapan rumus metode LSRL terdapat perbedaan yang besar untuk jumlah data genap dan ganjil, perbedaan terletak pada penentuan nilai X (periode) (Soepono, 2012:26). Rumus untuk perhitungan peramalan penjualan ditunjukkan pada persamaan 1.

$$Y = a + bX \quad \text{Persamaan (1)}$$

Keterangan :

X = periode (waktu)

Y = jumlah penjualan pada periode X

a = bilangan konstan

b = koefisien kecondongan garis tren

nilai a dan b diperoleh dari persamaan pembantu yang ditunjukkan pada persamaan 2 dan 3.

$$a = \Sigma Y / N \quad \text{Persamaan (2)}$$

$$b = \Sigma XY / \Sigma X^2 \quad \text{Persamaan (3)}$$

Data periode dengan jumlah data genap dari jumlah data dibagi 2 (dua), selanjutnya nomor pertengahan diberi angka permulaan -1, penetapan angka berikutnya untuk data di atasnya (nomor lebih kecil) + (ditambah) dengan angka – 2 (minus dua) dan untuk nomor dibawahnya (nomor lebih besar) ditambah + 2 (

dua). Data periode dengan jumlah data ganjil dari jumlah data +1 lalu dibagi 2 (dua), selanjutnya nomor pertengahan diberi angka permulaan 0 (nol), penetapan angka berikutnya untuk data di atasnya (nomor lebih kecil) + (ditambah) dengan angka – 1 (minus satu) dan untuk nomor dibawahnya (nomor lebih besar) ditambah 1 (satu) (Soepono, 2012:26). Pola data pada penelitian ini adalah pola tren, sehingga metode LSRL cocok digunakan untuk memprediksi hasil produksi dan hasil prediksi lebih akurat.

2.5. Data Time Series

Data time series adalah data yang dikumpulkan dari waktu ke waktu pada satu obyek dengan tujuan untuk menggambarkan perkembangan (Siswanto, 2012:56). Ada 4 faktor yang mempengaruhi *data time series*. Dalam data ekonomi biasanya kita mendapatkan adanya fluktuasi/ variasi dari waktu ke waktu atau disebut dengan variasi *time series*. Variasi ini biasanya disebabkan oleh adanya faktor Trend (*trend factor*), Fluktuasi siklis (*cyclical fluktuation*), Variasi musiman (*seasonal variation*), dan pengaruh random (*irregular/random influences*).

2.6. Autokorelasi

Autokorelasi digunakan untuk mengidentifikasi apakah data tersebut termasuk dalam kategori stasioner, acak, trend, dan musiman (Suliyanto, 2008). Perhitungan dengan autokorelasi akan berpatokan pada nilai lag (selisih antara sampel ke-i dengan sampel ke-i-1). Koefisien autokorelasi dapat dihitung dengan menggunakan persamaan berikut:

$$r_k = \frac{\sum_{t=1}^{n-k} (Y_t - \bar{Y})(Y_{t-k} - \bar{Y})}{\sum_{t=1}^n (Y_t - \bar{Y})^2} \quad \text{Persamaan (4)}$$

Keterangan :

- rk : Koefisien autokorelasi tingkat k
- Yt : Nilai observasi pada waktu t
- Yt-1 : Nilai observasi pada k periode sebelumnya (t-k)
- \bar{Y} : Nilai rata-rata serial data
- y : Yt- \bar{Y}

Berdasarkan perhitungan koefisien autokorelasi, pola data dapat diidentifikasi sebagai berikut:

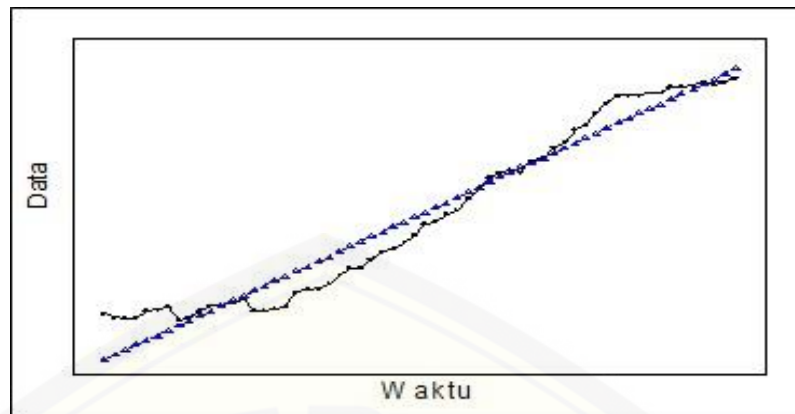
1. Apabila nilai autokorelasi pada lag 1 berbeda dengan nol (signifikan) akan tetapi pada time lag ke 2 dan ke 3 tidak berbeda dengan nol (tidak signifikan) maka data tersebut stasioner.
2. Apabila koefisien autokorelasi pada beberapa time lag pertama berbeda dengan nol (signifikan) maka data tersebut menunjukkan pola tren.
3. Apabila koefisien autokorelasi pada beberapa time lag pertama tidak berbeda dengan nol (tidak signifikan) maka data tersebut menunjukkan pola acak.
4. Apabila koefisien autokorelasi pada beberapa time lag yang mempunyai jarak secara sistematis berbeda dengan nol (signifikan) maka data tersebut menunjukkan pola musiman.

2.7. Pola Data

Pola data digunakan untuk menentukan metode atau teknik prediksi yang akan digunakan. Pola data menurut Soepono (2011:4). dikelompokkan sebagai berikut:

a. Pola Data Tren

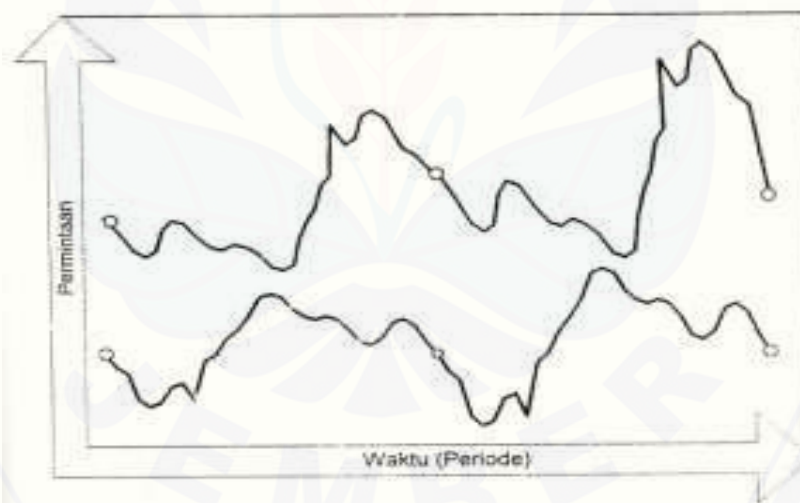
Data tren merupakan data dalam periode proyeksi menunjukkan adanya pola pertumbuhan atau penurunan. Jika dalam periode proyeksi data menunjukkan adanya pola pertumbuhan maka pola data tersebut memiliki tren positif. Begitu juga sebaliknya, periode proyeksi data menunjukkan adanya pola penurunan maka pola data tersebut dikatakan memiliki tren negatif. Grafik pola data tren dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Contoh Pola Data Tren

b. Pola Data Musiman

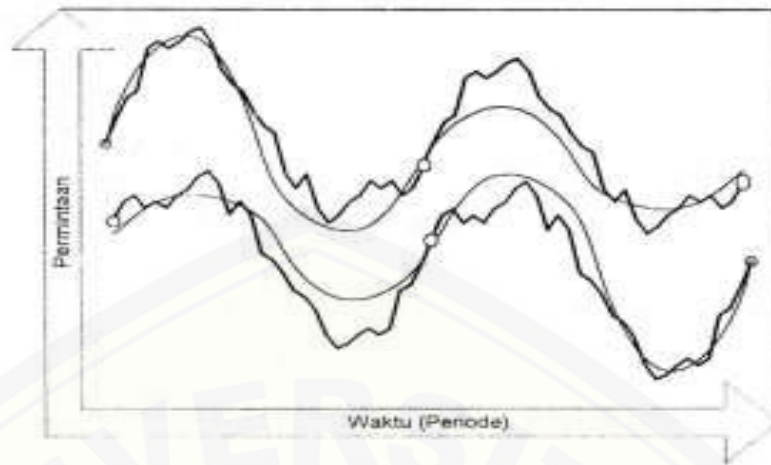
Data Musiman merupakan data runtut waktu yang memiliki pola perubahan yang berulang secara tahunan. Contoh grafik pola data musiman dapat dilihat pada Gambar 2.2



Gambar 2. 2 Contoh Pola Data Musiman

c. Pola Data Siklus

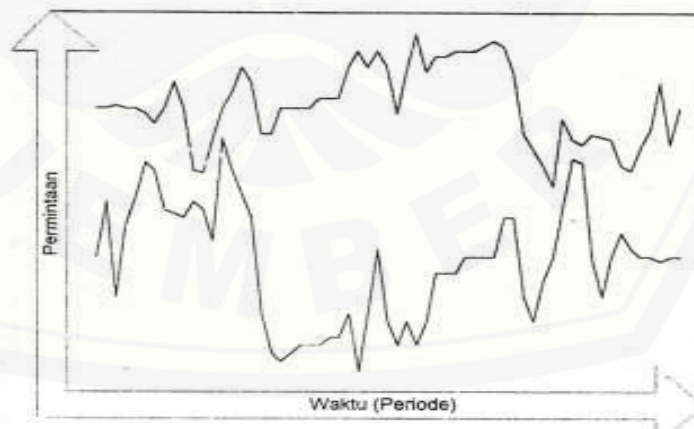
Data Siklus merupakan data runtut waktu yang memiliki fluktuasi di sekitar garis tren. Jika data memiliki pola siklus maka akan terjadi keadaan yang berulang dalam periode siklus. Contoh grafik pola data siklus dapat dilihat pada Gambar 2.3



Gambar 2. 3 Contoh Pola Data Siklus

d. Pola Data Acak

Permintaan suatu produk dapat mengikuti pola variasi secara acak karena faktor-faktor adanya bangkrutnya perusahaan, promosi khusus, dan kejadian-kejadian lainnya yang tidak mempunyai pola tertentu. Variasi acak ini diperlukan dalam rangka menentukan persediaan pengaman untuk mengantisipasi kekurangan persediaan bila terjadi lonjakan permintaan. Contoh grafik pola data acak dapat dilihat pada Gambar 2.4



Gambar 2. 4 Contoh Pola Data Acak

BAB 3. METODOLOGI PENELITIAN

Bab ini menjelaskan tentang metode yang digunakan dalam merancang dan membangun sistem menggunakan metode pengembangan.

3.1 Jenis Penelitian

Penelitian ini menggunakan jenis penelitian kualitatif dan kuantitatif. Jenis penelitian kualitatif digunakan karena penelitian ini menganalisa studi kasus pada barang persediaan Badan Pengelolaan Keuangan dan Aset Daerah Kabupaten Jember dan jenis penelitian kuantitatif digunakan karena penelitian ini menerapkan serta mengkaji teori yang sudah ada sebelumnya.

3.2 Metode Penelitian

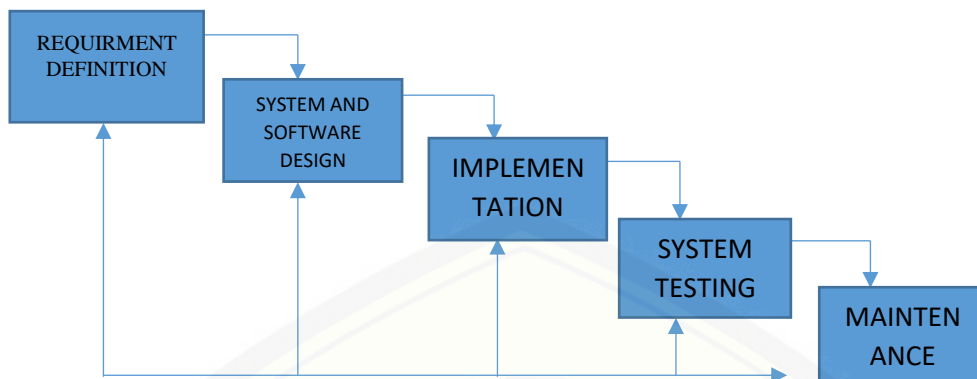
Metode penelitian yang digunakan pada penelitian ini adalah metode penelitian historikal karena pada penelitian ini membutuhkan data masa lalu yang digunakan untuk memprediksi periode mendatang.

3.3 Waktu dan Tempat Penelitian

Penelitian ini dilakukan di Badan Pengelolaan Keuangan dan Aset Kabupaten Jember. Waktu penelitian dilakukan selama 2 (dua) bulan, dimulai pada bulan November 2019 sampai dengan bulan Desember 2019.

3.4 Tahapan Penelitian

Tahap penelitian untuk pengembangan sistem prediksi barang persediaan pada Badan Pengelolaan Keuangan dan Aset Daerah Kabupaten jember ini menggunakan model *waterfall*. Model *waterfall* merupakan model pengembangan sistem yang dilakukan terurut mulai dari mengumpulkan kebutuhan, mendisain sistem, mengimplementasikan sistem dan *testing* hingga terakhir sistem akan digunakan dan siap untuk di *maintenance* (Basuki, 2006). Sebagaimana ditunjukkan pada Gambar 3.1



Gambar 3. 1 Tahapan metode *Waterfall* (Basuki, 2006)

3.4.1 Tahapan Analisis Kebutuhan *Requirment Definition*)

Tahap analisa merupakan tahapan pertama yang dilakukan dalam pengembangan sistem. Tahap ini dilakukan dengan wawancara, studi pustaka dan pengolahan data sebagai berikut :

1. Studi Pustaka

Studi literatur yang digunakan berasal dari jurnal, internet dan buku serta dokumen yang terkait dengan pola data, autokorelasi, dan data time series

2. Wawancara

Wawancara dilakukan pada “Sub Bagian Kesekretariatan umum dan kepegawaian Badan Pengelolaan Keuangan dan Aset Daerah Kabupaten Jember” untuk menentukan periode prediksi sesuai dengan ketentuan yang digunakan

3. Pengolahan Data

Tahap ini dimulai dengan menelaah data secara keseluruhan, pada tahap ini juga dilakukan penentuan kriteria apa saja yang layak untuk dijadikan parameter dari data yang sudah dikumpulkan, dan menentukan kebutuhan sistem berdasarkan studi pustaka dan wawancara yang telah dilakukan.

3.4.2 Tahapan Desain Sistem (*System and Software Design*)

Pembuatan desain sistem dalam penelitian ini menggunakan *Unified Modeling Language* (UML) yang dirancang dengan konsep *Object-Oriented*

Programming (OOP). Dengan menggunakan UML akan dibangun 7 diagram desain sistem yaitu:

1. *Business Process*
2. *Use Case Diagram*
3. *Scenario Diagram*
4. *Activity Diagram*
5. *Sequence Diagram*
6. *Class Diagram*
7. *Entity Relationship Diagram*

a. *Business Process*

Business process merupakan diagram yang menggambarkan proses dari sebuah sistem yang meliputi data apa yang diperlukan lalu data diolah untuk menghasilkan *output* yang diinginkan.

b. *Use Case Diagram*

Use case diagram adalah dokumentasi yang menggambarkan fitur dan aktor yang dapat mengakses fitur – fitur pada sistem. Gambaran yang ditekankan pada *use case* adalah “apa” yang dapat dilakukan oleh sistem, bukan “bagaimana” sistem bekerja. Aktor yang dapat mengakses sistem ini adalah pihak Gudang sebagai admin dan Manajer yang berperan untuk memantau transaksi berlangsung. Diagram *use case* dapat sangat membantu dalam penyusunan *requirements* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua fitur yang ada pada sistem.

c. *Scenario Diagram*

Scenario diagram berfungsi untuk menjelaskan alur sistem dari fitur yang ada pada *use case diagram*, skenario juga berfungsi menjelaskan reaksi yang akan terjadi pada sistem setelah menerima aksi dari aktor. *Scenario diagram*

menggambarkan keadaan normal saat setiap aktor mengakses sistem dan keadaan alternatif yang terjadi pada kondisi tertentu.

d. *Activity Diagram*

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, meliputi bagaimana masing-masing aktivitas dimulai, *decision* yang mungkin terjadi, dan bagaimana aktivitas berakhir. *Activity diagram* juga menggambarkan aktivitas sistem setelah menerima aksi dari aktor.

e. *Sequence Diagram*

Sequence diagram menggambarkan aliran logika dan interaksi yang terjadi di dalam sistem sesuai urutan dijalankannya sistem tersebut. Interaksi tersebut meliputi, aktor yang mengakses fitur sesuai dengan *usecase diagram*, tampilan sistem, *controller*, model, dan pesan yang disampaikan jika terjadi suatu kondisi tertentu.

f. *Class Diagram*

Class diagram merupakan sebuah spesifikasi yang menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan struktur dan deskripsi *class* serta hubungan antar *class*, sehingga memudahkan dalam proses pengkodean.

g. *Entity Relationship Diagram (ERD)*

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan data yang mempunyai hubungan antar relasi.

3.4.3 Tahapan Implementasi Sistem (*System Implementation*)

Setelah desain sistem selesai dilakukan maka akan dilanjutkan ketahap pembuatan system dan implementasi. Pembuatan system meliputi penulisan kode program dan pembuatan basis data. Penulisan kode dilakukan menggunakan *tools* Sublime Text 3 sebagai *editor* dengan bahasa pemograman HTML, PHP, CSS yang terintegrasi dengan DBMS MySQL sebagai manajemen basis data.

3.4.4 Tahapan Pengujian Sistem (*System Testing*)

Pengujian digunakan untuk mengetahui sejauh mana sistem ini dapat berjalan. Serta untuk mengetahui letak kekurangan yang ada pada sistem ini. Terdapat dua metode yang digunakan untuk pengujian yaitu *White Box* dan *Black Box*.

a. *White Box Testing*

White box adalah pengujian yang dilakukan dengan membandingkan aplikasi dengan modul yang telah dibuat. Jika hasil yang dikeluarkan sistem tidak sesuai dengan modul yang telah ditetapkan maka perlu ada baris kode, paramater, atau variabel yang perlu dicek dan diperbaiki ulang

b. *Black Box Testing*

Berbeda dengan *white box testing*, *Black box testing* merupakan metode pengujian perangkat lunak yang memeriksa fungsionalitas aplikasi. Pengujian ini lebih difokuskan kepada kerja program. Pengujian dilakukan dengan melibatkan mahasiswa secara acak sebagai pengguna.

3.4.5 Tahapan Pemeliharaan Sistem (*Maintanance*)

Pemeliharaan merupakan proses perawatan sistem setelah sistem digunakan oleh pengguna. Pada tahap pemeliharaan ini dilakukan untuk mengatasi masalah sistem yang ada saat melakukan *testing White box* dan *Black box*. Jika hasil *testing*

telah didapatkan, pihak *developer* segera memperbaiki kekurangan dan kesalahan sesuai hasil *testing*.

3.5 Gambaran Umum Sistem

Pada penelitian ini sistem informasi yang akan dibangun berbasis *web application*. Sistem ini akan menerapkan Metode *Least Square Regression Line* berfungsi sebagai prediksi untuk kebutuhan barang persediaan bulan selanjutnya dan untuk memprediksi kebutuhan anggaran untuk tahun selanjutnya dari hasil rekap perbulan serta akan menghasilkan *output* berupa laporan akhir persediaan. Di dalam system ini juga memungkinkan untuk melakukan penambahan dan pengurangan persediaan sehingga menunjukkan sisa dari tiap bulan yang akan menjadi acuan awal bulan berikutnya.

BAB 4. PENGEMBANGAN SISTEM

Bab ini menguraikan mengenai analisis kebutuhan, desain, implementasi, dan pengujian sistem yang digunakan dalam proses pengembangan sistem prediksi barang persediaan menggunakan metode *Least Square Regression Line* pada Badan Pengelolaan Keuangan dan Aset Daerah Kabupaten Jember. Di mana tahapan analisis kebutuhan hingga pengujian yang dilakukan sesuai dengan metode pengembangan *waterfall*.

4.1 Tahapan Analisis Kebutuhan

Tahap analisis kebutuhan merupakan tahap awal dalam membangun sistem berdasarkan model *waterfall*. Analisis kebutuhan yang dilakukan sesuai dengan data yang diperoleh pada tahapan yang telah dijelaskan pada bab metodologi penelitian.

Berdasarkan hasil studi pustaka, wawancara dan pengolahan data, tahap pertama yang dilakukan adalah mengelola data transaksi yang didapat dari hasil wawancara di Badan Pengelolaan Keuangan dan Aset Daerah Kabupaten Jember. Tahap kedua yaitu menghitung prediksi kebutuhan Barang Persediaan periode selanjutnya menggunakan metode *Least Square Regression Line*. Tahap ketiga yaitu mendeskripsikan SOP dari sistem yang akan dibangun. Tahap keempat menentukan kebutuhan fungsional dari sistem yang akan dibangun. Tahap kelima menentukan kebutuhan non-fungsional sistem yang akan dibangun.

4.1.1 Penerapan *Least Square*

Berdasarkan hasil studi pustaka yang diperoleh, metode *Least Square Regression Line* digunakan untuk melakukan prediksi untuk kebutuhan produk di periode selanjutnya.

4.1.2 SOP (*statement of purpose*)

Sistem informasi peramalan kebutuhan barang persediaan ini menggunakan metode *Least Square Regression Line* merupakan sistem yang mampu

menampilkan hasil peramalan berdasarkan data kebutuhan pada masa lalu yang diproses dengan perhitungan menggunakan metode *Least Square Regression Line*. Data yang digunakan adalah data kebutuhan barang persediaan yang di dapat melalui pengumpulan data tiap bulan mulai tahun 2019. Data tersebut kemudian dihitung sehingga menghasilkan angka prediksi kebutuhan barang persediaan untuk periode selanjutnya. Hasil prediksi tersebut akan menjadi pendukung dan bahan pertimbangan dalam menentukan jumlah penganggaran dan pembelian pada periode selanjutnya.

4.1.3 Kebutuhan Fungsional

Kebutuhan fungsional sistem berisi fitur-fitur inti yang harus dipenuhi dalam sistem agar sistem mampu difungsikan sesuai dengan tujuan dan kebutuhan pengguna terhadap sistem itu sendiri. Kebutuhan fungsional dari Sistem Informasi prediksi barang persediaan menggunakan Metode *Least Square Regression Line* yaitu:

- a. Sistem mampu mengelola data barang persediaan (*view, insert, edit*).
- b. Sistem mampu mengelola data transaksi pemasukan dan pengeluaran barang persediaan (*view, insert*).
- c. Sistem mampu mengelola data prediksi kebutuhan barang persediaan (*view, insert*).
- d. Sistem mampu menampilkan rekap
- e. Sistem mampu menampilkan pengeluaran

4.1.4 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan fitur-fitur yang dimiliki untuk mendukung sistem dalam memenuhi fungsionalitasnya untuk dapat memenuhi kebutuhan dari pengguna. Kebutuhan non-fungsional dari sistem informasi peramalan kebutuhan barang persediaan pada Badan Pengelolaan Keuangan dan Aset Daerah Kabupaten Jember Menggunakan Metode *Least Square Regression Line* yaitu:

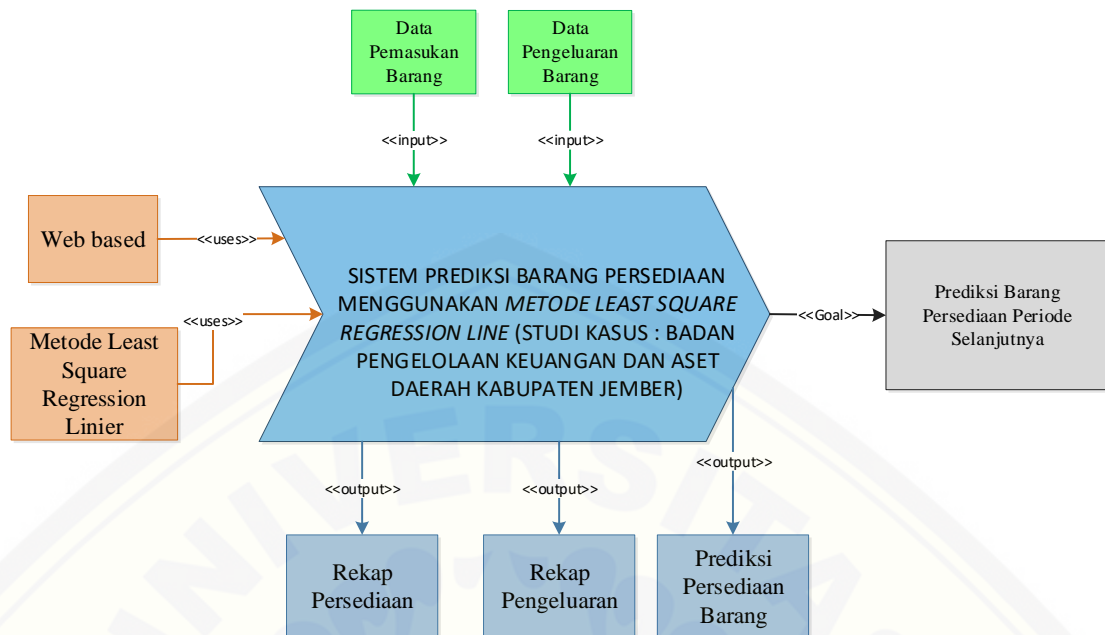
- a. Sistem bekerja sesuai dengan fungsinya dan dapat dijalankan pada semua komputer dan semua *browser* yang berbeda.
- b. Tampilan dan bahasa komunikasi sistem mudah dimengerti oleh pengguna untuk memberikan kenyamanan pemakaian dan memudahkan pengoperasian.
- c. Sistem menggunakan *username* dan *password* untuk autentifikasi akses admin terhadap sistem.

4.2 Desain Sistem

Tahapan yang dilakukan setelah melakukan analisis kebutuhan sistem yaitu tahap perencanaan pembangunan sistem yang dapat digambarkan dengan desain sistem. Desain sistem yang dibuat untuk Sistem Prediksi Barang Persediaan menggunakan metode *least square regression line* meliputi *business proses*, *use case diagram*, *scenario diagram*, *activity diagram*, *sequence diagram*, *class diagram*, dan *entity relationship diagram*.

4.2.1 Business Proses

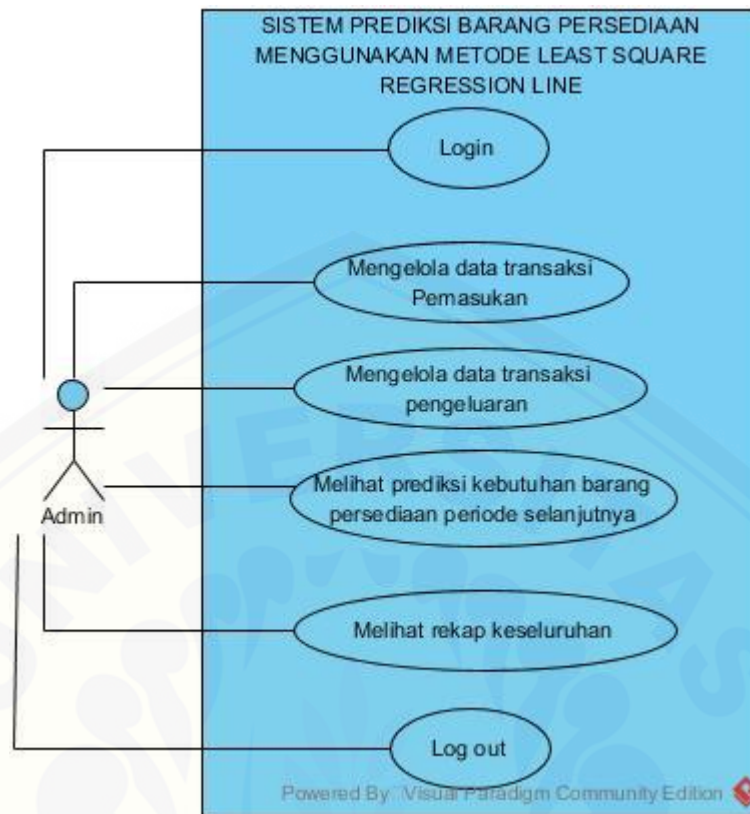
Selain dapat dideskripsikan dalam sebuah SOP (*Statement Of Purpose*), gambaran umum Sistem Prediksi Barang Persediaan menggunakan metode *least square regression line* dapat digambarkan melalui sebuah *business proses*. Seperti yang dapat kita lihat pada Gambar 4.1 menggambarkan data-data yang digunakan sebagai masukan, data keluaran, *uses* sistem yang dibangun hingga *goal* dari dibangunnya sistem sendiri.



Gambar 4. 1 Business Proses Sistem

4.2.2 Usecase Diagram

Use case diagram digunakan untuk dapat menggambarkan interaksi antara aktor dengan sistem prediksi barang persediaan menggunakan metode *Least Square Regression Line* yang akan dibangun. Melalui *use case diagram* dapat diketahui interaksi yang dapat dilakukan aktor terhadap sistem sesuai dengan hak akses yang dimiliki oleh masing-masing aktor atau pengguna. Pada Gambar 4.2 digambarkan *use case diagram* yang terdiri atas satu aktor dengan enam *use case*.



Gambar 4. 2 Usecase diagram

Tabel 4. 1 Deskripsi pembagian aktor sistem

No	Aktor	Deskripsi
1	Admin	Aktor ini memiliki hak untuk mengelola data transaksi pemasukan, data transaksi pengeluaran, prediksi barang persediaan, melihat rekap pemasukan dan pengeluaran dan rekap keseluruhan transaksi.

Terdapat 6 *usecase* dalam *usecase diagram*. Deskripsi dari *usecase* tersebut dapat dilihat pada tabel 4.2

Tabel 4. 2 Deskripsi *use case* sistem

No	<i>Use Case</i>	Deskripsi
1	2	3
1	<i>Login</i>	<i>Use case</i> yang menggambarkan proses masuk ke dalam Sistem.
2	Mengelola data transaksi pemasukan	<i>Use case</i> yang menggambarkan proses pengolahan data transaksi pemasukan meliputi menambahkan, merubah, dan menghapus data.
3	Mengelola data transaksi pengeluaran	<i>Use case</i> yang menggambarkan proses pengolahan data transaksi pengeluaran meliputi menambahkan, dan merubah data.
4	Melihat prediksi kebutuhan barang persediaan periode selanjutnya	<i>Use case</i> yang menggambarkan proses menampilkan hasil prediksi kebutuhan barang persediaan periode selanjutnya
5	Melihat rekap keseluruhan	<i>Use case</i> yang menggambarkan proses menampilkan rekap keseluruhan transaksi.
6	Logout	<i>Use case</i> yang menggambarkan proses keluar dari sistem.

4.2.3 *Scenario Diagram*

Scenario diagram merupakan penjelasan alur kerja sistem secara rinci meliputi penjelasan prekondisi dan prakondisi juga aksi yang akan dikerjakan sistem ketika pengguna menggunakan sistem. Diagram ini juga menjelaskan *alternatif flow* atau kondisi alternatif ketika pengguna melakukan kesalahan dalam menggunakan sistem, diagram ini dibuat sesuai dengan fitur yang ada pada *use case diagram*.

a. *Scenario Login*

Scenario login merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *use case scenario login*. Pada skenario tersebut dapat dilihat bagaimana sebuah alur *login* berjalan setelah

mendapat aksi dari aktor dan reaksi dari sistem untuk merespon reaksi tersebut sehingga aktor telah melakukan *login* pada sistem tersebut dan berhasil masuk sistem sebagai user yang dapat mengoperasikan sistem. Proses lengkap *scenario login* dapat dilihat pada Tabel 4.3.

Tabel 4. 3 *Scenario Login*

ID	USC 01
Nama Usecase	<i>Login</i>
Aktor	<i>Admin</i>
Pre Condition	Memasukan Username dan Password untuk masuk ke dalam sistem
Post Condition	Aktor telah memasuki sistem
Skenario Normal <i>Login</i>	
Aksi Aktor	Reaksi Sistem
1. Memasukan username dan password	
2. Klik tombol “ <i>Sign in</i> ”	
	3. Eksekusi validasi username dan password di dalam database
	4. Menampilkan halaman <i>Home</i>
Skenario Alternatif Username atau Password salah	
2. Klik tombol “ <i>Sign in</i> ”	
	3. Menampilkan <i>form login</i> dan Menampilkan pesan “gagal <i>login</i> ”

b. *Scenario Diagram* Mengelola Data Transaksi Pemasukan

Scenario diagram mengelola data transaksi pemasukan merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *scenario diagram* mengelola data transaksi pemasukan. Pada *scenario* tersebut dapat dilihat bagaimana sebuah alur melihat data transaksi pemasukan berjalan setelah mendapat aksi dari aktor dan reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah mengelola data

transaksi pemasukan. Proses lengkap *scenario* mengelola data transaksi pemasukan dapat dilihat pada Tabel 4.4.

Tabel 4. 4 *Scenario* Mengelola Data Transaksi Pemasukan

Nomor Use case	USC 02
Nama Use case	Mengelola Data Transaksi Pemasukan
Aktor	<i>Admin</i>
Precondition	<i>Admin</i> memilih menu transaksi Pemasukan
Postcondition	<i>Admin</i> berhasil menambah, mengubah, menghapus, dan melihat data transaksi pemasukan
Skenario Normal	
Tambah data Barang Masuk	
Aktor	Sistem
1. Memilih menu pemasukan	
	2. Menampilkan tabel data barang masuk yang berisi : a. No b. Nama barang c. Kategori d. Total masuk e. Harga satuan f. Total g. <i>Action</i> h. Tambah barang
3. Klik Tombol “Tambah Barang”	
	4. Menampilkan Form <i>Input</i> Barang Masuk yang berisi : a. Nama barang b. Jumlah satuan c. Harga satuan d. Satuan e. Tanggal masuk f. kategori
5. Mengisi Form <i>Input</i> Barang Sesuai dengan Atribut yang di butuhkan	
6. Klik tombol “Simpan”	
	7. Menyimpan form <i>Input</i> Barang Masuk

	8. Menampilkan tabel data barang masuk yang berisi : <ol style="list-style-type: none"> No Nama barang Kategori Total masuk Harga satuan Total Action Tambah barang
Skenario Alternatif Isi form kosong atau tidak sesuai	
6a. Klik tombol "Simpan"	
	7a. Menampilkan pesan "Please fill out this Field"
	8a. Menampilkan Form <i>Input</i> Barang Masuk yang berisi : <ol style="list-style-type: none"> Nama barang Jumlah satuan Harga satuan Satuan Tanggal masuk kategori
Skenario Alternatif Edit data	
9. Klik tombol "Edit"	
	10. Menampilkan <i>form edit</i> Data Barang yang berisi : <ol style="list-style-type: none"> Nama barang Harga satuan Satuan kategori
11. Mengisi <i>form Edit</i>	
12. Klik tombol "Simpan"	
	13. Menyimpan <i>form edit</i>
	14. Menampilkan tabel data barang masuk yang berisi : <ol style="list-style-type: none"> No Nama barang Kategori Total masuk Harga satuan Total Action Tambah barang

Skenario Alternatif Edit data tidak sesuai	
16a. Klik tombol “Simpan”	18a. Menampilkan pesan “ <i>Form</i> yang anda isi salah”
Skenario Normal Tambah Stock Barang	
15. Klik Tombol “ <i>Detail</i> ”	16. Menampilkan Tabel <i>Detail</i> dari barang yang dipilih dengan atribut : a. Nama barang b. Kategori c. Saldo awal d. Harga satuan e. Total f. Tanggal masuk g. No BAST h. <i>Action</i> i. <i>Restock</i>
17. Klik “ <i>Restock</i> ”	18. Menampilkan <i>form Restock</i> Barang yang berisi : a. Nama barang b. Jumlah satuan c. Harga satuan d. Satuan e. Tanggal restock f. No BAST
19. Mengisi <i>form Restock</i> Barang	
20. Klik tombol “Simpan”	
	21. Menyimpan <i>form Restock</i> Barang
	22. Menampilkan tabel data barang masuk yang berisi : a. No b. Nama barang c. Kategori d. Total masuk e. Harga satuan f. Total g. <i>Action</i> h. Tambah barang

Skenario Alternatif	
Isi Form Restock Barang tidak sesuai atau kosong	
20a. Klik tombol “Simpan”	
	21a. Menampilkan Pesan “Please Fill out This Field”
	22a. Menampilkan form <i>Restock</i> Barang yang berisi : a. Nama barang b. Jumlah satuan c. Harga satuan d. Satuan e. Tanggal restock f. No BAST
Skenario Normal	
Edit detail Stock Barang	
23. Klik tombol “Edit”	
	24. Menampilkan Form <i>Edit Detail Stock</i> Barang yang berisi : a. Nama barang b. Harga satuan c. Jumlah masuk d. No BAST e. Tanggal masuk
25. Mengisi form <i>Edit Detail Stock</i> Barang	
26. Klik tombol “Simpan”	
	27. Menyimpan form <i>edit Detail Stock</i> Barang
	28. Menampilkan tabel data barang masuk yang berisi : a. No b. Nama barang c. Kategori d. Total masuk e. Harga satuan f. Total g. <i>Action</i> h. Tambah barang
Skenario Alternatif	
Isi Form edit detail stock barang tidak Sesuai	
26a. Klik tombol “Simpan”	
	27a. Menampilkan pesan “Please fill out this Field”

	<p>28a. Menampilkan Form <i>Edit Detail Stock</i> Barang yang berisi :</p> <ol style="list-style-type: none"> Nama barang Harga satuan Jumlah masuk No BAST Tanggal masuk
--	--

c. Skenario Mengelola Data Transaksi Pengeluaran

Skenario mengelola data transaksi pengeluaran merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *use case* skenario mengelola data transaksi pengeluaran. Pada skenario tersebut dapat dilihat bagaimana sebuah alur melihat data transaksi pengeluaran berjalan setelah mendapat aksi dari aktor dan reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah mengelola data transaksi pemasukan. Proses lengkap skenario mengelola data transaksi pengeluaran dapat dilihat pada Tabel 4.5.

Tabel 4. 5 Skenario Mengelola Data Transaksi Pengeluaran

Nomor Usecase	USC 03
Nama Use Case	Mengelola Data Transaksi Pengeluaran
Aktor	<i>Administrator</i>
Precondition	<i>Admin</i> memilih menu Pengeluaran
Postcondition	<i>Admin</i> berhasil menambah dan melihat data transaksi pengeluaran
Skenario Normal Tambah data pengeluaran	
Aktor	Sistem
1. Memilih menu pengeluaran	
	<p>2. Menampilkan Tabel Pengeluaran yang berisi :</p> <ol style="list-style-type: none"> No Nama barang Kategori Stock Total keluar Harga satuan Total <i>Action</i>

3. Menekan tombol “Keluar”	
	4. Menampilkan <i>form</i> Pengeluaran yang berisi : a. Nama barang b. Stock barang c. Jumlah keluar d. Harga satuan e. Satuan f. Tanggal keluar g. No surat keluar
5. Mengisi <i>form</i> pengeluaran	
6. Klik tombol “Simpan”	
	7. Menyimpan <i>form</i> pengeluaran
	8. Menampilkan Tabel Pengeluaran yang berisi : a. No b. Nama barang c. Kategori d. Stock e. Total keluar f. Harga satuan g. Total h. <i>Action</i>
Skenario Alternatif Isi <i>form</i> tidak sesuai	
8a. Klik tombol “Simpan”	
	11a. Menampilkan pesan “Form yang anda isi salah”
Skenario Alternatif Melihat rincian data pengeluaran	
5. Klik tombol “Detail”	
	6. Menampilkan tabel detail transaksi pengeluaran setiap barang yang berisi : a. No b. Nama barang c. Kategori d. <i>Stock</i> e. Harga satuan f. Nilai pengeluaran g. Tanggal keluar h. No surat keluar i. Satuan j. Total

d. Skenario Melihat Prediksi Kebutuhan Barang Persediaan

Skenario melihat prediksi kebutuhan barang persediaan merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *use case* skenario melihat melihat prediksi kebutuhan barang persediaan. Pada skenario tersebut dapat dilihat bagaimana sebuah alur melihat melihat prediksi kebutuhan barang persediaan berjalan setelah mendapat aksi dari aktor dan reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah melihat melihat prediksi kebutuhan barang persediaan. Proses lengkap skenario melihat prediksi kebutuhan barang persediaan dapat dilihat pada Tabel 4. 6.

Tabel 4. 6 *Scenario* melihat prediksi kebutuhan barang perseediaan

ID	USC 04
Nama Use Case	Memprediksi Kebutuhan Barang Persediaan
Aktor	Administrator
Pre Condition	Aktor memilih menu prediksi Least Square Regression Line
Post Condition	Menampilkan Hasil Prediksi Barang Persediaan
Skenario Normal <i>Memprediksi barang persediaan</i>	
Aksi Aktor	Reaksi Sistem
1. Klik menu Prediksi <i>Least Square Regression Line</i>	
	2. Menampilkan halaman Prediksi yang berisi nama barang yang akan diprediksi
4. Memilih barang yang di prediksi	
	5. Menghitung Prediksi Barang
	6. Menampilkan hasil prediksi barang persediaan yang berisi : a. No b. Bulan , tahun keluar c. Item terjual d. Mean e. X^2 f. $X*Y$ g. Hasil prediksi

Skenario Alternatif Tidak memilih barang yang diprediksi	
3a. Tidak memilih barang yang diprediksi	
	4a. Sistem tidak melakukan perhitungan dan tidak menampilkan hasil perhitungan

e. Skenario Melihat Rekap Keseluruhan

Skenario melihat rekap keseluruhan merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *use case* skenario melihat rekap keseluruhan. Proses lengkap skenario melihat rekap keseluruhan dapat dilihat pada Tabel 4.7.

Tabel 4. 7 *Scenario* Melihat Rekap Keseluruhan

ID	USC 05
Nama Use Case	Melihat Rekap Keseluruhan
Aktor	<i>Administrator</i>
Pre Condition	<i>Admin</i> memilih menu Rekap
Post Condition	<i>Admin</i> Berhasil Melihat Rekap Keseluruhan
Skenario Normal Menampilkan Rekap Keseluruhan	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan halaman Beranda
2. Memilih menu Rekap Keseluruhan	
	3. Menampilkan rekap Keseluruhan yang berisi : a. No b. Nama barang c. Kategori d. Satuan e. Pemasukan f. Pengeluaran g. Stock opname h. Harga satuan i. Total

f. Skenario *Logout*

Skenario *logout* merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *use case* skenario *logout*. Pada skenario tersebut dapat dilihat bagaimana sebuah alur *logout* berjalan setelah mendapat aksi dari aktor dan reaksi dari sistem untuk merespon reaksi tersebut sehingga aktor telah melakukan. Proses lengkap skenario *logout* dapat dilihat pada Tabel 4.8.

Tabel 4. 8 *Scenario logout*

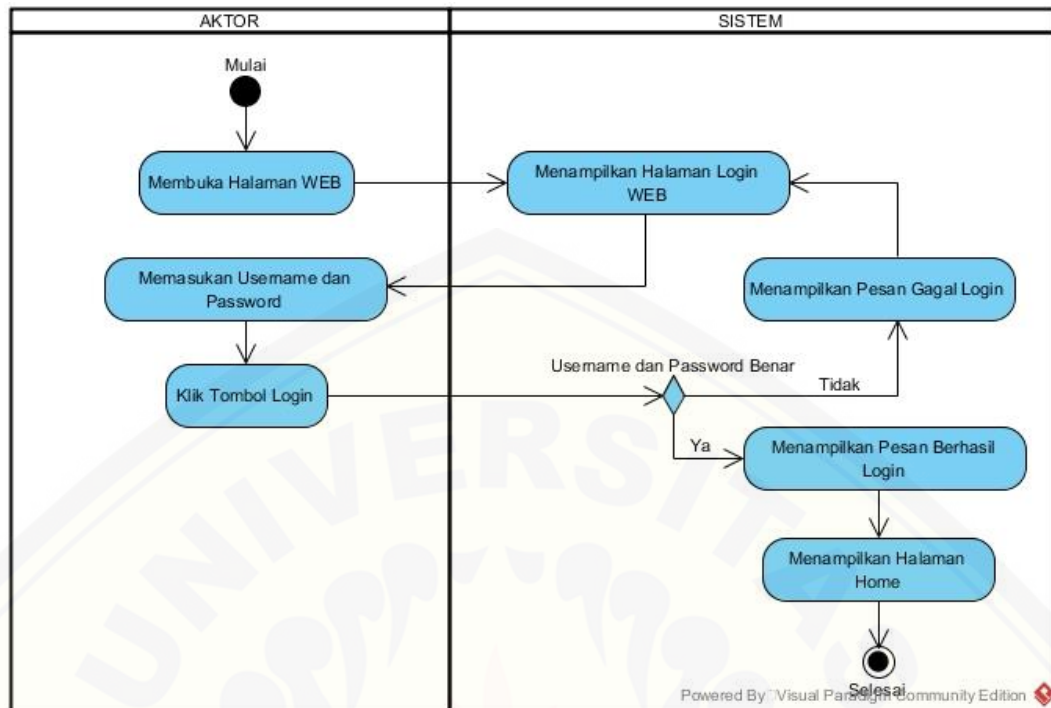
ID	USC 06
Nama Usecase	<i>Logout</i>
Aktor	<i>Admin</i>
<i>Pre Condition</i>	Sistem dalam keadaan <i>Logout</i>
<i>Post Condition</i>	Sistem Menampilkan Halaman <i>Logout</i>
Skenario Normal <i>Logout</i>	
Aksi Aktor	Reaksi Sistem
1. Klik tombol " <i>Logout</i> "	2. Sistem Melakukan <i>Break Session</i>
	3. Menampilkan halaman <i>Login</i>

4.2.4 *Activity Diagram*

Activity Diagram menggambarkan aliran aktivitas Sistem Prediksi Barang Persediaan menggunakan metode *Least Square Regression Line* yang akan dibangun. Sistem Prediksi Barang Persediaan menggunakan metode *Least Square Regression Line* memiliki delapan *activity diagram* yaitu sebagai berikut:

a. *Activity Diagram Login*

Activity diagram Login dilakukan oleh admin gudang dan manajer Laboratorium Klinik. *Activity diagram login* menjelaskan tentang bagaimana aktor memasukan data username dan password untuk mendapatkan akses sistem dan sistem merespon dengan menjalankan fungsi melakukan autentifikasi hak akses semua aktor. Proses lengkapnya dapat dilihat pada Gambar 4.3.



Gambar 4. 3 Activity Diagram Login

b. Activity Diagram Mengelola Data Transaksi Pemasukan

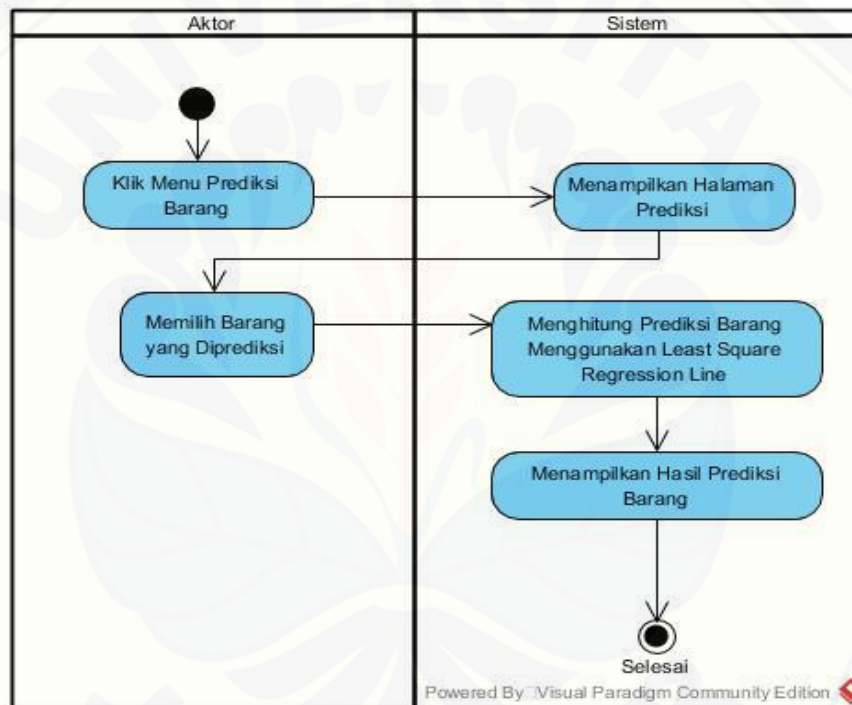
Activity diagram Mengelola data transaksi pemasukan dapat dilakukan oleh admin pengurus barang Badan Pengelolaan Keuangan dan Aset. Activity diagram Mengelola data transaksi pemasukan berisi alur bagaimana aktor dapat mengelola data pemasukan pada sistem meliputi menambahkan, merubah, dan menghapus data. Proses lengkapnya dapat dilihat pada Lampiran A Gambar A.1.

c. Activity Diagram Mengelola Data Transaksi Pengeluaran

Activity diagram Diagram Mengelola Data Transaksi Pengeluaran dapat dilakukan oleh admin. Activity diagram Diagram Mengelola Data Transaksi Pengeluaran menjelaskan alur bagaimana admin dapat mengelola data pengeluaran pada sistem meliputi menambahkan, dan melihat rincian pengeluaran. Ketika aktor memilih tombol pengeluaran maka sistem akan menampilkan *form* pengisian data pengeluaran, sedangkan jika memilih ikon *detail* maka sistem akan menampilkan rincian pengeluaran barang sesuai nama barang. Proses lengkapnya dapat dilihat pada Lampiran A Gambar A.2.

d. *Activity Diagram Memprediksi Kebutuhan Barang Persediaan*

Activity diagram Memprediksi Kebutuhan Barang Persediaan dapat dilakukan oleh admin. *Activity diagram Memprediksi Kebutuhan Barang Persediaan* berisi alur diagram yang menjelaskan ketika aktor memilih menu prediksi *Least Square Regression Line* maka sistem akan menampilkan *form* prediksi yang wajib diisi oleh aktor sehingga sistem dapat memproses perhitungan metode *Least Square Regression Line* dan menampilkan halaman hasil prediksi. Proses lengkapnya dapat dilihat pada Gambar 4.4.



Gambar 4. 4 Activity Diagram Memprediksi Kebutuhan Barang Persediaan

e. *Activity Diagram Melihat Rekap Keseluruhan*

Activity diagram melihat rekap keseluruhan dapat dilakukan oleh admin. *Activity diagram melihat rekap keseluruhan* berisi alur diagram yang menjelaskan ketika aktor memilih menu rekap dan sub menu keseluruhan masuk maka sistem dapat menampilkan data transaksi keseluruhan setiap tahun. Proses lengkapnya dapat dilihat pada Lampiran A Gambar A.3.

f. *Activity Diagram Logout*

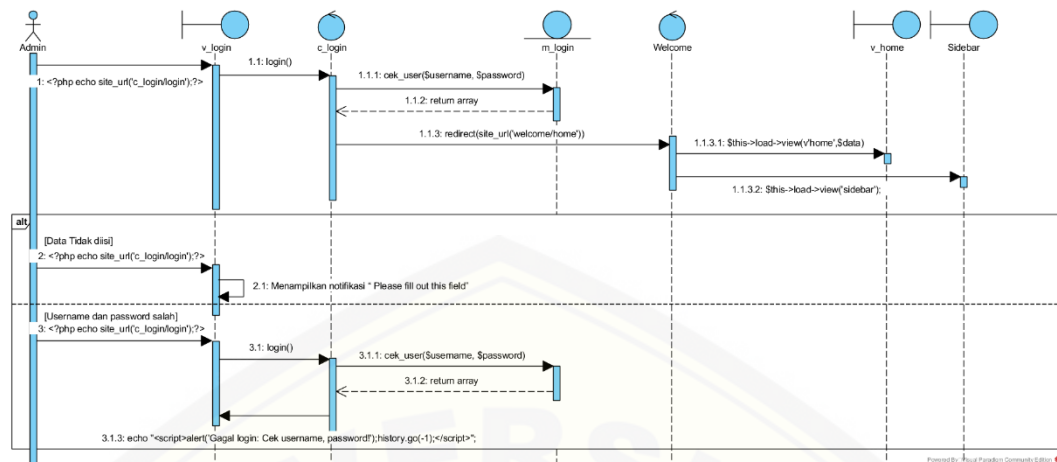
Activity diagram Logout dilakukan oleh admin. *Activity diagram logout* menjelaskan tentang bagaimana aktor melakukan logout dari sistem dan sistem merespon dengan menjalankan fungsi logout . Proses lengkapnya dapat dilihat pada Lampiran A Gambar A.4.

4.2.5 *Sequence Diagram*

Sequence diagram adalah dokumentasi berupa diagram yang menampilkan interaksi di dalam sistem sesuai dengan urutan dijalankannya sistem tersebut. Interaksi tersebut meliputi aktor yang mengakses fitur tersebut sesuai dengan *use case diagram*, tampilan sistem, *controller*, *model*, dan pesan yang disampaikan jika terjadi kondisi tertentu. *Sequence diagram* digunakan untuk menggambarkan skenario dan aliran logika dalam sistem sehingga tahapan yang dibangun jelas dan teratur. *Sequence diagram* dari Sistem Prediksi Barang Persediaan Menggunakan Metode *Least Square Regression Line* pada Badan Pengelolaan Keuangan dan Aset Daerah adalah sebagai berikut:

a. *Sequence Diagram Login*

Penggambaran *sequence diagram login* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *usecase login*. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. Pada *Sequence Login* dapat dilihat alur perjalanan fitur *Login* yang dilakukan oleh aktor. Aktor mengisi username, password dan menekan tombol *login*. Selanjutnya *controller Login* akan melakukan pengecekan terhadap database yang diproses pada *model Login* apakah username dan password dari aktor benar atau tidak, jika benar sistem akan menampilkan halaman *Home*, jika tidak maka akan kembali ke halaman *login* dengan menampilkan peringatan *Login Gagal*. *Sequence diagram login* lebih lengkap akan dijelaskan pada Gambar 4.5.



Gambar 4. 5 Sequence Diagram Login

b. *Sequence Diagram* Mengelola data transaksi Pemasukan

Penggambaran *sequence* Mengelola data transaksi Pemasukan digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *usecase* Mengelola data transaksi Pemasukan. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. Diagram ini menggambarkan alur dimana aktor memilih menu Pemasukan dan selanjutnya *controller* barang_masuk memanggil tampilan *pemasukan*. Selanjutnya aktor memilih kategori barang persediaan, maka *controler* barang_masuk memanggil barang sesuai dengan kategori melalui *model* m_masuk. Selanjutnya aktor menekan tombol tambah dan *controler* akan memanggil view form tambah_barang yang kemudian dilanjutkan aktor untuk mengisi form dan menekan tombol simpan, maka *controler* barang_masuk akan memproses data tersebut kedalam *database* melalui model m_masuk. Selanjutnya *controler* barang_masuk menampilkan halaman pemasukan kembali sebagai tanda penambahan data berhasil, sedangkan jika pengisian kurang tepat atau kosong maka *controler* akan menampilkan pesan gagal menambahkan. Diagram ini juga menggambarkan alur ketika Aktor memilih ikon ubah data maka *conttroller* akan memanggil data yang akan diubah sesuai id di *database* melalui model, selanjutnya *controler* menampilkan *form* berisi data yang telah dipilih untuk diubah. Selanjutnya *controler* barang_masuk menampilkan halaman pemasukan kembali sebagai tanda penambahan data berhasil, sedangkan jika pengisian kurang tepat

atau kosong maka *controller* akan menampilkan view gagal menambahkan data. Alur ketika aktor memilih ikon Hapus data maka *controller* akan memproses id data yang akan di hapus melalui model dan menampilkan halaman Pemasukan sebagai tanda bahwa data berhasil dihapus. *Sequence* diagram Mengelola data transaksi Pemasukan lebih lengkap akan dijelaskan pada Lampiran B Gambar B.1.

c. *Sequence* Diagram Mengelola Data Transaksi Pengeluaran

Penggambaran *sequence* Mengelola data transaksi Pengeluaran digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *usecase* Mengelola data transaksi Pengeluaran. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. Diagram ini menggambarkan alur dimana aktor memilih menu Peeluaran dan selanjutnya *controller* barang_keluar memanggil halaman *pengeluaran*. Selanjutnya aktor menekan tombol tambah dan *controler* akan memanggil view form tambah_pengeluaran yang kemudian dilanjutkan aktor untuk mengisi form dan menekan tombol simpan, maka *controler* barang_keluar akan memproses data tersebut kedalam *database* melalui model m_keluar. Selanjutnya *controler* barang_keluar menampilkan halaman pengeluaran kembali sebagai tanda penambahan data berhasil, sedangkan jika pengisian kurang tepat atau kosong maka *controller* akan menampilkan pesan gagal menambahkan. Diagram ini juga menggambarkan alur ketika Aktor memilih tombol detail maka *controller* akan memanggil data pengeluaran sesuai dengan id barang yang dipilih. *Sequence* diagram Mengelola data transaksi Pengeluaran lebih lengkap akan dijelaskan pada Lampiran B Gambar B.2.

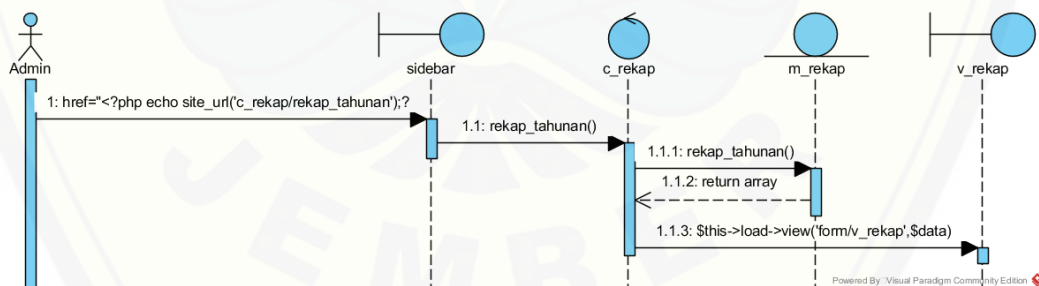
d. *Sequence* Diagram Memprediksi Kebutuhan Barang Persediaan

Penggambaran *sequence* diagram memprediksi kebutuhan barang persediaan digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* memprediksi kebutuhan barang persediaan. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. Pada *sequence* diagram ini fitur bisa

diakses oleh admin. Diagram ini menggambarkan ketika aktor memilih tombol prediksi maka *controller* prediksi akan memproses data pengeluaran barang persediaan melalui model *m_prediksi* dan akan ditampilkan oleh *controller* dalam wujud tampilan prediksi. *Sequence* diagram memprediksi kebutuhan regaen lebih lengkap akan dijelaskan pada Lampiran B Gambar B.3.

e. *Sequence* Diagram Melihat Rekap Keseluruhan

Penggambaran *sequence* diagram melihat rekap pengeluaran digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *usecase* melihat rekap keseluruhan. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. Diagram ini menggambarkan alur ketika aktor memilih menu rekap Keseluruhan maka *controller* rekap akan memanggil data di database melalui model *m_rekap*. Selanjutnya *controller* rekap menampilkan halaman rekap keseluruhan dengan menampilkan data transaksi pemasukan dan pengeluaran yang sudah diambil melalui model. *Sequence* diagram Melihat Rekap Pengeluaran lebih lengkap akan dijelaskan pada Gambar 4.6.

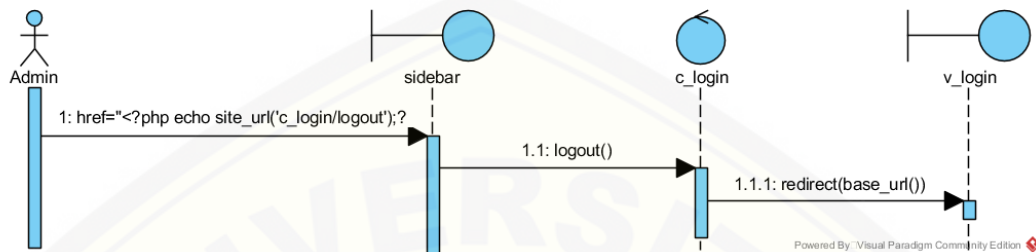


Gambar 4. 6 *Sequence* Diagram melihat rekap keseluruhan

f. *Sequence* Diagram Logout

Penggambaran *sequence* Logout digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *usecase* Logout. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. Diagram ini menggambarkan alur ketika

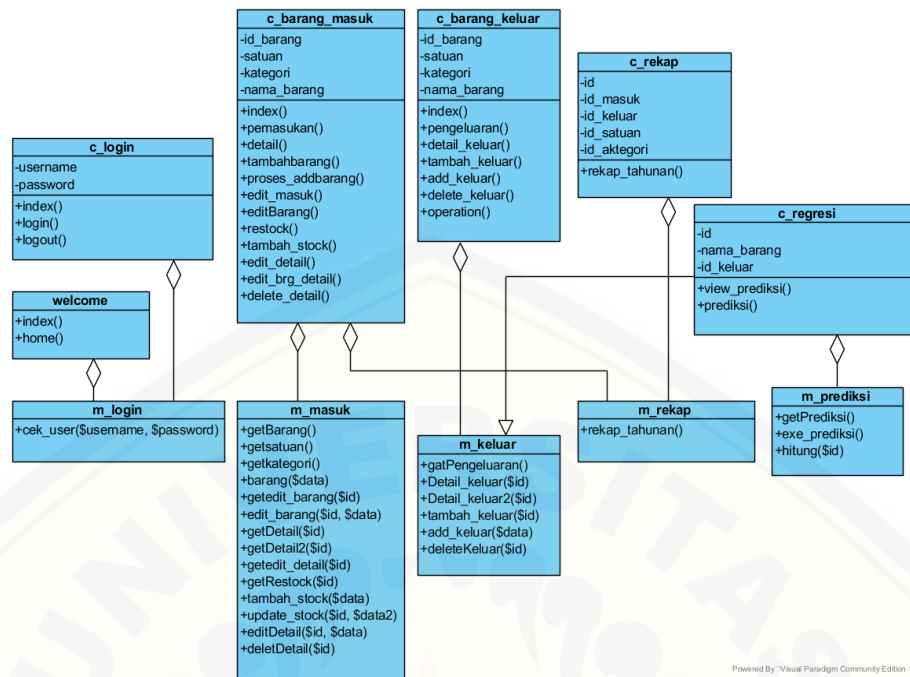
aktor memilih Logout maka *controller* Logout akan melakukan pemutusan pada Session dan akan menampilkan halaman *Login*. *Sequence* diagram Melihat Rekap Pengeluaran lebih lengkap akan dijelaskan pada Gambar 4.7.



Gambar 4.7 *Sequence diagram Logout*

4.2.6 Class Diagram

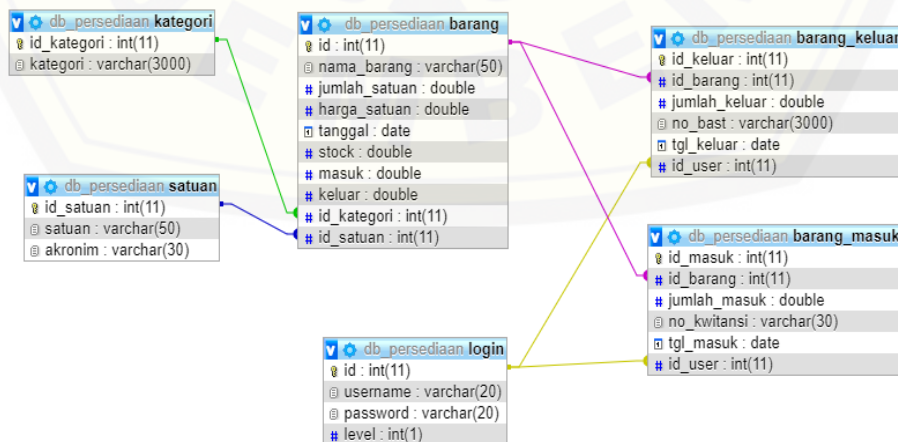
Class diagram menggambarkan hubungan antara kelas yang digunakan untuk membangun suatu sistem. Dalam paradigma OOP (*Object Oriented Programming*) terdapat 3 jenis kelas yaitu *model*, *view* dan *controller*. Berdasarkan *sequence* diagram yang telah dibangun, *class* diagram sistem Prediksi Barang Persediaan Menggunakan *Least Square Regression Line* pada Badan Pengelolaan Keuangan dan Aset Daerah ini terdiri dari 5 *class* model dan 6 *class* controller. Pada *class* tersebut memiliki fungsi masing-masing, *class controller* berfungsi untuk mengakses data yang ada pada *class* Model. Jika kita lihat garis *controller* prediksi terhubung dengan model *m_prediksi* dan *m_keluar* yang artinya ada transaksi data antara controller dan model tersebut. *Class* diagram sistem dapat dilihat pada Lampiran Gambar 4.5.



Gambar 4. 8 Class Diagram

4.2.7 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) Sistem Prediksi Barang Persediaan Menggunakan Metode *Least Square Regression Line* pada Badan Pengelolaan Keuangan dan Aset Daerah Kabupaten Jember merupakan gambaran komponen, relasi dan struktur *database* yang digunakan dalam pembuatan sistem. ERD yang diimplementasikan pada sistem ini terdiri dari 6 entitas yang dapat dilihat pada Gambar 4.6.



Gambar 4. 9 Entity Relationship Diagram

4.3 Implementasi

Tahap implementasi merupakan tahap penulisan kode program sesuai desain yang telah dirancang sebelumnya. Bahasa pemrograman yang digunakan dalam sistem ini adalah bahasa Pemrograman PHP dengan *framework Code Igniter* (CI) sedangkan DBMS yang digunakan adalah MySQL. Penulisan kode program ini ditulis dalam 3 bagian kelas yaitu model, view dan controller. Penulisan kode program dijelaskan pada sub sub bab berikut.

4.3.1. Kode Program Peramalan

Tabel 4. 9 Kode Metode *Least Square Regression Line* hitung(\$id)

Kode	
<pre>function hitung(\$id) { \$this->db->where('tgl_keluar<=',date('Y-m-d')); \$this->db->where('id_barang',\$id); \$query = \$this->db->get('barang_keluar'); \$this->db->where('id_barang',\$id); \$data['barang'] = \$this->db->get('barang_keluar')->row_array(); \$data['penjualan'] = \$this->db->get('barang')->row_array(); // HITUNG JUMLAH KOLOM \$jumlah=\$query->num_rows(); // HITUNG PREDIKSI if(\$jumlah>0){ \$genap = \$jumlah%2==0?true:false; \$stengah = (\$genap?(((\$jumlah)/2)+1:(((\$jumlah+1)/2))-1); \$datas = \$query->result_array(); \$data[\$stengah]['x'] = \$genap?-1:0; \$x = \$data[\$stengah]['x']; \$sx2 = pow(\$x, 2); \$jumlah = 0; \$i = 0; \$data['x'] = array(); \$data['jumlahx'] = 0; \$data['jumlahx2'] = 0; \$data['jumlahxy'] = 0; \$data['xprediksi'] = 0; foreach (\$datas as \$a) { if(\$genap){ if(\$i<\$stengah){ \$data['x'][\$i] = -((2*(\$stengah-\$i))-1); \$data['x2'][\$i] = pow(\$data['x'][\$i],2); \$data['xy'][\$i] = \$data['x'][\$i]*\$a['jumlah_keluar']; \$data['jumlahx'] = \$data['jumlahx']+\$data['x'][\$i]; \$data['jumlahx2'] = \$data['jumlahx2']+\$data['x2'][\$i];</pre>	Lanjut

<pre> \$data['jumlahxy'] = \$data['jumlahxy']+\$data['xy'][\$i]; }else{ \$data['x'][\$i] = (\$i-\$stengah)+1; \$data['x2'][\$i] = pow(\$data['x'][\$i],2); \$data['xy'][\$i] = \$data['x'][\$i]*\$a['jumlah_keluar']; \$data['jumlahx'] = \$data['jumlahx']+\$data['x'][\$i]; \$data['jumlahx2'] = \$data['jumlahx2']+\$data['x2'][\$i]; \$data['jumlahxy'] = \$data['jumlahxy']+\$data['xy'][\$i]; \$stengah --; } }else{ \$data['x'][\$i] = \$i-\$stengah; \$data['x2'][\$i] = pow(\$data['x'][\$i],2); \$data['xy'][\$i] = \$data['x'][\$i]*\$a['jumlah_keluar']; \$data['jumlahx'] = \$data['jumlahx']+\$data['x'][\$i]; \$data['jumlahx2'] = \$data['jumlahx2']+\$data['x2'][\$i]; \$data['jumlahxy'] = \$data['jumlahxy']+\$data['xy'][\$i]; } \$jumlah +=\$a['jumlah_keluar']; \$data['tgl_keluar'][\$i] = \$a['tgl_keluar']; \$data['nilai'][\$i] = \$a['jumlah_keluar']; \$i++; } \$data['totaly'] = \$jumlah; \$data['total'] = \$i; \$genap?\$data['xprediksi'] = \$data['x'][\$i-1]+2:\$data['xprediksi']=\$data['x'][\$i-1]+1; \$data['nilaia'] = \$jumlah/\$i; \$data['nilaib'] = \$data['jumlahxy']/\$data['jumlahx2']; \$data['y'] = \$data['nilaia']+((\$data['nilaib']*\$data['xprediksi'])); return \$data; }else{ return array();}} </pre>	Dilanjutkan
--	-------------

Pada Tabel 4.9 merupakan kode perhitungan metode *Least Square Regression Line* pada package *model* prediksi. Pada kode program tersebut menjelaskan tahapan atau langkah – langkah proses perhitungan metode *Least Square Regression Line* yang menghasilkan perhitungan kriteria yang ada. Hasil perhitungan tersebut adalah prediksi Barang Persediaan periode selanjutnya, hasil perhitungan dapat dilihat dari data Pengeluaran barang setiap ada barang keluar.

4.3.2. Kode Program Melihat prediksi kebutuhan barang persediaan(class *controllers/DataPakai*)

Tabel 4. 10 Kode Program Melihat prediksi kebutuhan barang persediaan(class *controllers/DataPakai*)

KODE
<pre> function getPrediksi(){ \$this->db->select('*'); \$this->db->join('barang', 'barang.id=barang_keluar.id_barang'); \$this->db->from('barang_keluar'); \$this->db->group_by('id_barang'); \$penjualan=\$this->db->get(); return \$penjualan->result(); } function exe_prediksi(){ \$this->db->select('*'); \$this->db->join('barang', 'barang.id=barang_keluar.id_barang','left'); \$this->db->from('barang_keluar'); \$penjualan=\$this->db->get(); return \$penjualan->result(); } function hitung(\$id) { \$this->db->where('tgl_keluar<=',date('Y-m-d')); \$this->db->where('id_barang',\$id); \$query = \$this->db->get('barang_keluar'); \$this->db->where('id_barang',\$id); \$data['barang'] = \$this->db->get('barang_keluar')->row_array(); \$data['penjualan'] = \$this->db->get('barang')->row_array(); // HITUNG JUMLAH KOLOM \$jumlah=\$query->num_rows(); // HITUNG PREDIKSI if(\$jumlah>0){ \$genap = \$jumlah%2==0?true:false; \$tengah = (\$genap?(((\$jumlah)/2)+1:(((\$jumlah+1)/2))-1); \$datas = \$query->result_array(); \$data[\$tengah]['x'] = \$genap?-1:0; \$x = \$data[\$tengah]['x']; \$sx2 = pow(\$x, 2); \$jumlah = 0; \$i = 0; \$data['x'] = array(); \$data['jumlahx'] = 0; \$data['jumlahx2'] = 0; \$data['jumlahxy'] = 0; \$data['xprediksi'] = 0; foreach (\$datas as \$a) { if(\$genap){ if(\$i<\$tengah){ \$data['x'][\$i] = -((2*(\$tengah-\$i))-1); } } } } </pre>

```

    $data['x2'][$i] = pow($data['x'][$i],2);
    $data['xy'][$i] = $data['x'][$i]*$a['jumlah_keluar'];
    $data['jumlahx'] = $data['jumlahx']+$data['x'][$i];
    $data['jumlahx2'] = $data['jumlahx2']+$data['x2'][$i];
    $data['jumlahxy'] = $data['jumlahxy']+$data['xy'][$i];
  }else{
    $data['x'][$i] = ($i-$stengah)+1;
    $data['x2'][$i] = pow($data['x'][$i],2);
    $data['xy'][$i] = $data['x'][$i]*$a['jumlah_keluar'];
    $data['jumlahx'] = $data['jumlahx']+$data['x'][$i];
    $data['jumlahx2'] = $data['jumlahx2']+$data['x2'][$i];
    $data['jumlahxy'] = $data['jumlahxy']+$data['xy'][$i];
    $stengah --;
  }
}
}else{
  $data['x'][$i] = $i-$stengah;
  $data['x2'][$i] = pow($data['x'][$i],2);
  $data['xy'][$i] = $data['x'][$i]*$a['jumlah_keluar'];
  $data['jumlahx'] = $data['jumlahx']+$data['x'][$i];
  $data['jumlahx2'] = $data['jumlahx2']+$data['x2'][$i];
  $data['jumlahxy'] = $data['jumlahxy']+$data['xy'][$i];
}
}
$jumlah +=$a['jumlah_keluar'];
$data['tgl_keluar'][$i] = $a['tgl_keluar'];
$data['nilai'][$i] = $a['jumlah_keluar'];
$i++;
}
}
$data['totaly'] = $jumlah;
$data['total'] = $i;
$genap?$data['xprediksi'] = $data['x'][$i-1]+2:$data['xprediksi']=$data['x'][$i-1]+1;
$data['nilaia'] = $jumlah/$i;
$data['nilaib'] = $data['jumlahxy']/$data['jumlahx2'];
$data['y'] = $data['nilaia']+($data['nilaib']*$data['xprediksi']);
return $data;
}else{
  return array();
}
}
}

```

4.4 Pengujian

Tahapan pengujian sistem merupakan suatu tahapan yang dilakukan secara sistematis untuk menguji dan mengevaluasi sistem dengan menggunakan sebuah metode pengujian sistem untuk mengetahui apakah kebutuhan sistem telah terpenuhi dan sistem layak untuk digunakan oleh pengguna. Agar pengujian yang

dilakukan lebih valid, maka tahap pengujian Sistem Prediksi Barang Persediaan Menggunakan Metode *Least Square Regression Line* ini dilakukan dengan menggunakan dua metode, yaitu *white box* dan *black box*.

4.4.1 Metode *White Box*

White Box Testing merupakan pengujian pada modul pengkodean program untuk menjamin kode program bebas dari kesalahan sintaks maupun logika. Dalam pengujian white box terdapat beberapa tahapan pembuatan dokumentasi pengujian yaitu *cyclomatic complexity (CC)*, *listing program*, dan *test case*. Pengujian untuk fitur prediksi barang persediaan.

b. *Listing Program*

Listing program merupakan baris-baris kode yang nantinya akan diuji. Setiap langkah dari kode-kode yang ada diberi nomor baik menjalankan *statement* biasa atau penggunaan kondisi dalam program.

Tabel 4. 11 *Listing program function hitung(\$id)*

Kode	
<pre>function hitung(\$id) { \$this->db->where('tgl_keluar<=',date('Y-m-d')); \$this->db->where('id_barang',\$id); \$query = \$this->db->get('barang_keluar'); \$this->db->where('id_barang',\$id); \$data['barang'] = \$this->db->get('barang_keluar')->row_array(); \$data['penjualan'] = \$this->db->get('barang')->row_array(); // HITUNG JUMLAH KOLOM \$jumlah=\$query->num_rows(); // HITUNG PREDIKSI if(\$jumlah>0){ \$genap = \$jumlah%2==0?true:false; \$stengah = (\$genap?(((\$jumlah)/2)+1:(((\$jumlah+1)/2))-1); \$datas = \$query->result_array(); \$data[\$stengah]['x'] = \$genap?-1:0; \$x = \$data[\$stengah]['x']; \$sx2 = pow(\$x, 2); \$jumlah = 0; \$i = 0; \$data['x'] = array();</pre>	Lanjut


```

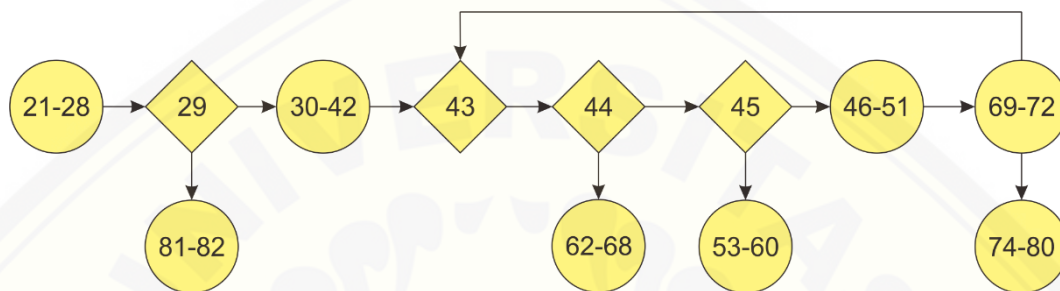
$data['jumlahx'] = 0;
$data['jumlahx2'] = 0;
$data['jumlahxy'] = 0;
$data['xprediksi'] = 0;
foreach ($datas as $a) {
    if($genap){
        if($i<$tengah){
            $data['x'][$i] = -((2*($tengah-$i))-1);
            $data['x2'][$i] = pow($data['x'][$i],2);
            $data['xy'][$i] = $data['x'][$i]*$a['jumlah_keluar'];
            $data['jumlahx'] = $data['jumlahx']+$data['x'][$i];
            $data['jumlahx2'] = $data['jumlahx2']+$data['x2'][$i];
            $data['jumlahxy'] = $data['jumlahxy']+$data['xy'][$i];
        }else{
            $data['x'][$i] = ($i-$tengah)+1;
            $data['x2'][$i] = pow($data['x'][$i],2);
            $data['xy'][$i] = $data['x'][$i]*$a['jumlah_keluar'];
            $data['jumlahx'] = $data['jumlahx']+$data['x'][$i];
            $data['jumlahx2'] = $data['jumlahx2']+$data['x2'][$i];
            $data['jumlahxy'] = $data['jumlahxy']+$data['xy'][$i];
            $tengah --;
        }
    }else{
        $data['x'][$i] = $i-$tengah;
        $data['x2'][$i] = pow($data['x'][$i],2);
        $data['xy'][$i] = $data['x'][$i]*$a['jumlah_keluar'];
        $data['jumlahx'] = $data['jumlahx']+$data['x'][$i];
        $data['jumlahx2'] = $data['jumlahx2']+$data['x2'][$i];
        $data['jumlahxy'] = $data['jumlahxy']+$data['xy'][$i];
    }
    $jumlah +=$a['jumlah_keluar'];
    $data['tgl_keluar'][$i] = $a['tgl_keluar'];
    $data['nilai'][$i] = $a['jumlah_keluar'];
    $i++;
}
$data['totaly'] = $jumlah;
$data['total'] = $i;
$genap?$data['xprediksi'] = $data['x'][$i-1]+2:$data['xprediksi']=$data['x'][$i-1]+1;
$data['nilaia'] = $jumlah/$i;
$data['nilaib'] = $data['jumlahxy']/$data['jumlahx2'];
$data['y'] = $data['nilaia']+($data['nilaib']*$data['xprediksi']);
return $data;
}else{
    return array();
}
}

```

Dilanjutkan

c. Diagram alir

Diagram alir merupakan notasi yang digunakan untuk merepresentasikan aliran kontrol. Aliran kontrol yang digambarkan merupakan hasil penomoran dari listing program. Diagram alir digambarkan dengan node-node (simpul) yang dihubungkan dengan *edge-edge* (garis) yang menggambarkan alur jalannya program.



Gambar 4. 10 Diagram alir *function* hitung(\$id)

d. Kompleksitas siklomatik (*cyclomatic complexity*)

Kompleksitas siklomatik merupakan metrik perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logis suatu program. Bila digunakan dalam konteks teknik pengujian jalur dasar, nilai yang dihitung untuk kompleksitas siklomatik mendefinisikan jumlah jumlah jalur independen dalam basis set suatu program. Perhitungan kompleksitas siklomatik menggunakan persamaan berikut ini.

$$V(G) = E - N + 2$$

Keterangan :

V(G) : Kompleksitas siklomatik

E : Jumlah *edge* (garis)

N : Jumlah *node* (simpul)

function hitung()

$$V(G) = E - N + 2$$

$$V(G) = 60 - 59 + 2$$

$$V(G) = 3$$

4.4.2 Metode *Black Box*

Dokumentasi hasil pengujian fitur prediksi kebutuhan barang persediaan dapat dilihat pada Tabel 4.7. Sedangkan untuk fitur yang lain dapat dilihat pada Lampiran E (Pengujian *Black Box*).

Tabel 4. 12 Pengujian *Black Box* Prediksi Kebutuhan Barang Persediaan

No	Fitur	Kasus	Hasil	Keterangan	
				Berhasil	Tidak
1.	<i>Login</i>	Ketika klik tombol <i>login</i> dan semua <i>field</i> pada <i>form login</i> sudah terisi dengan benar	Menampilkan halaman awal admin	V	
		Ketika klik tombol <i>login</i> dan belum mengisi <i>form</i> secara lengkap / <i>field</i> masih ada yang kosong	Menampilkan <i>Error</i> message username atau password salah	V	
2.	Tambah data Barang Pemasukan	Ketika klik tombol simpan dan semua <i>field</i> pada <i>form</i> tambah barang sudah terisi dengan benar	Menambah data barang sesuai yang di <i>inputkan</i>	V	
		Ketika klik tombol simpan dan belum mengisi <i>form</i> tambah secara lengkap / <i>field</i> masih ada yang kosong	Menampilkan pesan <i>Please fill out this field</i>	V	

3.	<i>Edit data barang masuk</i>	Ketika klik tombol <i>simpan</i> dan semua <i>field</i> pada <i>form edit</i> data barang sudah terisi dengan benar	<i>Update</i> data barang sesuai yang <i>diinputkan</i>	V	
		Ketika klik tombol <i>simpan</i> dan belum mengisi secara lengkap / <i>field</i> masih ada yang kosong	Menampilkan pesan <i>Please fill out this field</i>	V	
4.	Hapus data <i>detail stock</i> barang	Ketika klik tombol <i>delete</i> data yang akan dihapus	<i>Delete</i> data <i>detail stock</i> sesuai yang dipilih	V	
5.	Tambah <i>stock</i> barang masuk	Ketika klik tombol <i>simpan</i> dan semua <i>field</i> pada <i>form</i> restock barang masuk sudah terisi dengan benar	Menambah data <i>stock</i> barang masuk sesuai yang <i>diinputkan</i>	V	
		Ketika klik tombol <i>simpan</i> dan belum mengisi <i>form</i> restock barang secara lengkap / <i>field</i> masih ada yang kosong	Menampilkan pesan <i>Please fill out this field</i>	V	
6.	<i>Edit detail stock</i> barang	Ketika klik tombol <i>simpan</i> dan semua <i>field</i> pada <i>form edit detail stock</i> barang	<i>Update</i> data <i>detail stock</i> barang sesuai yang <i>diinputkan</i> oleh admin	V	

		sudah terisi dengan benar			
		Ketika klik tombol simpan dan belum mengisi <i>form edit detail stock</i> barang secara lengkap / <i>field</i> masih ada yang kosong	Menampilkan pesan data <i>Please fill out this field</i>	V	
7.	Tambah barang keluar	Ketika klik tombol simpan dan semua <i>field</i> pada <i>form</i> tambah barang sudah terisi dengan benar	Menambah data barang keluar sesuai yang <i>diinputkan</i>	V	
		Ketika klik tombol simpan dan belum mengisi <i>form</i> tambah barang keluar secara lengkap / <i>field</i> masih ada yang kosong	Menampilkan pesan <i>please fill out this field</i>	V	
	Prediksi Kebutuhan Barang Persediaan	Ketika pilih produk yang di prediksi.	Menampilkan Hasil Prediksi Barang Persediaan berdasarkan Pilihan dari pengguna	V	
8.	Hapus data Barang keluar	Ketika klik tombol delete data yang akan dihapus	<i>Delete</i> data barang keluar sesuai yang dipilih	V	

BAB 6. PENUTUP

Bab ini berisi kesimpulan dan saran dari penelitian yang telah dilakukan. Kesimpulan dan saran ini diharapkan mampu menjadi acuan untuk melakukan penelitian selanjutnya.

6.1 Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan adalah:

1. Penerapan metode *least square regression line* untuk prediksi barang persediaan dilakukan melalui tiga tahapan. Pertama melakukan pengolahan data barang persediaan yang sudah ada mulai dari tahun 2018 sampai tahun 2019 dengan cara melakukan pengelompokan data barang persediaan sesuai dengan nama barang, bulan pengeluaran dan tahun pengeluaran barang. Data hasil pengelompokan tersebut kemudian diolah kembali untuk menentukan pola data dari barang persediaan, setelah mendapatkan pola data trend data barang persediaan dapat dihitung menggunakan metode *least square regression line*. Proses perhitungan dengan metode *least square regression line* dilakukan menurut nama barang dan pengeluaran tiap bulan pada tahun yang sama untuk prediksi kebutuhan barang persediaan pada bulan selanjutnya.
2. Perancangan dan pembangunan Sistem Prediksi Barang Persediaan Menggunakan Metode *Least Square Regression Line* (Studi Kasus : Badan Pengelolaan Keuangan dan Aset Daerah Kabupaten Jember) ini menggunakan metode SDLC model *waterfall*. Perancangan dilakukan dengan cara melakukan tahapan pertama pada pemodelan *waterfall* yaitu analisis kebutuhan sistem. Pada tahapan ini data diperoleh melalui studi pustaka, wawancara, pengolahan data, mencari kebutuhan fungsional dan non fungsional untuk sistem prediksi barang persediaan. Dari tahapan analisis kebutuhan tersebut dikembangkan menuju tahapan desain sistem, dimana desain sistem menganut *unified modeling language* dan *object oriented programming*. Tahap selanjutnya yaitu implementasi desain untuk dijadikan sebuah sistem. Tahapan

implementasi dilakukan hingga sistem dapat dioperasikan dan akan diuji untuk menemukan kendala yang didapatkan saat sistem dijalankan. Pengujian sistem dilakukan dalam 2 tahap yaitu *white box* dan *black box*. Dalam pengujian sistem didapatkan hasil prediksi barang persediaan untuk bulan selanjutnya yang menganut perhitungan dari *least square regression line*. Data pengeluaran dihitung berdasarkan banyak pengeluaran per tanggal pengeluaran per barang. Jumlah row dari barang keluar dihitung untuk dicari nilai tengah untuk data genap nomor tengah diberi angka permulaan -2 untuk data di atasnya ditambah -2 dan untuk data dibawahnya ditambah +2. Untuk data ganjil nomor tengah diberi angka permulaan 0, data dengan nomor yang lebih kecil ditambah -1 dan untuk nomor yang lebih besar ditambah +1. Jumlah barang keluar ditotal untuk menentukan nilai Y, dan dibagi dengan jumlah row keluar untuk mencari nilai a, untuk mencari nilai b dilakukan perkalian antara $X*Y$ dibagi dengan X^2 . Dimana nilai X didapat berdasarkan penomoran dan Y jumlah barang persediaan keluar pertanggal/pertahun. Nilai a dan b akan dihitung untuk mencari prediksi periode selanjutnya. Tahapan selanjutnya ialah *mintanance* yaitu penyesuaian sistem ketika terdapat masalah pada pengujian *white box* atau *black box*.

6.2 Saran

Adapun saran yang ditujukan untuk memberikan masukan yang lebih baik yaitu :

1. Sistem prediksi barang persediaan diharapkan pada pengembang selanjutnya memberikan fitur yang lebih lengkap hingga mencakup pengelolaan prediksi barang seluruh OPD yang ada di jember.
2. Proses peramalan jumlah kebutuhan barang persediaan dilakukan dengan metode peramalan yang berbeda atau mengkombinasikan metode *Least Square Regression Line* dengan metode lainnya.

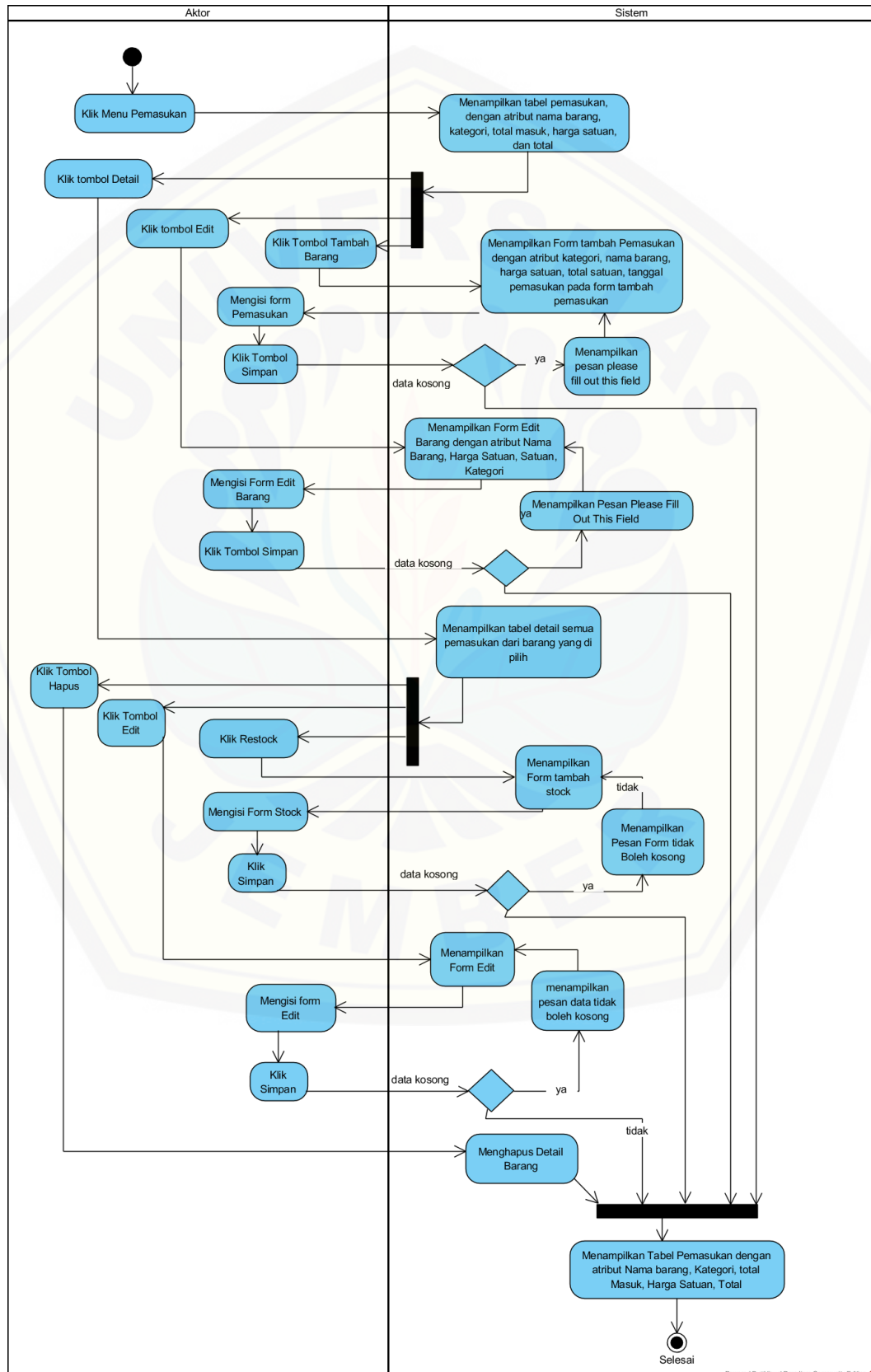
DAFTAR PUSTAKA

- Jogiyanto, H. 2005. *Analisis dan Desain Sistem Informasi*. Yogyakarta: C.V Andi Offset.
- Rambe, M. I. 2014. *Perancangan Aplikasi Peramalan Persediaan Obat-obatan Menggunakan Metode Least Square*. *Pelita informatika budi darma*, volume : VI, nomor : 1, Maret 2014 .
- Setiawan, H., Mulyanto, A., Hadjaratie, L. 2013. *Penerapan Meotde Least Square Regression Line dan Economic Order Quantity Pada Sistem Pengendalian Persediaan*.
- Siahaan,D. 2012. *Analisis Kebutuhan dalam Rekayasa Perangkat Lunak*. Yogyakarta: C.V Andi Offset.
- Siswanto, V. A. 2012. *Strategi dan Langkah - Langkah Penelitian*. Yogyakarta: Graha ilmu.
- Soepono, B. 2012. *Modul Peramalan Penjualan*. Malang: Kementerian Pendidikan dan Kebudayaan Politeknik Negeri Malang.

LAMPIRAN

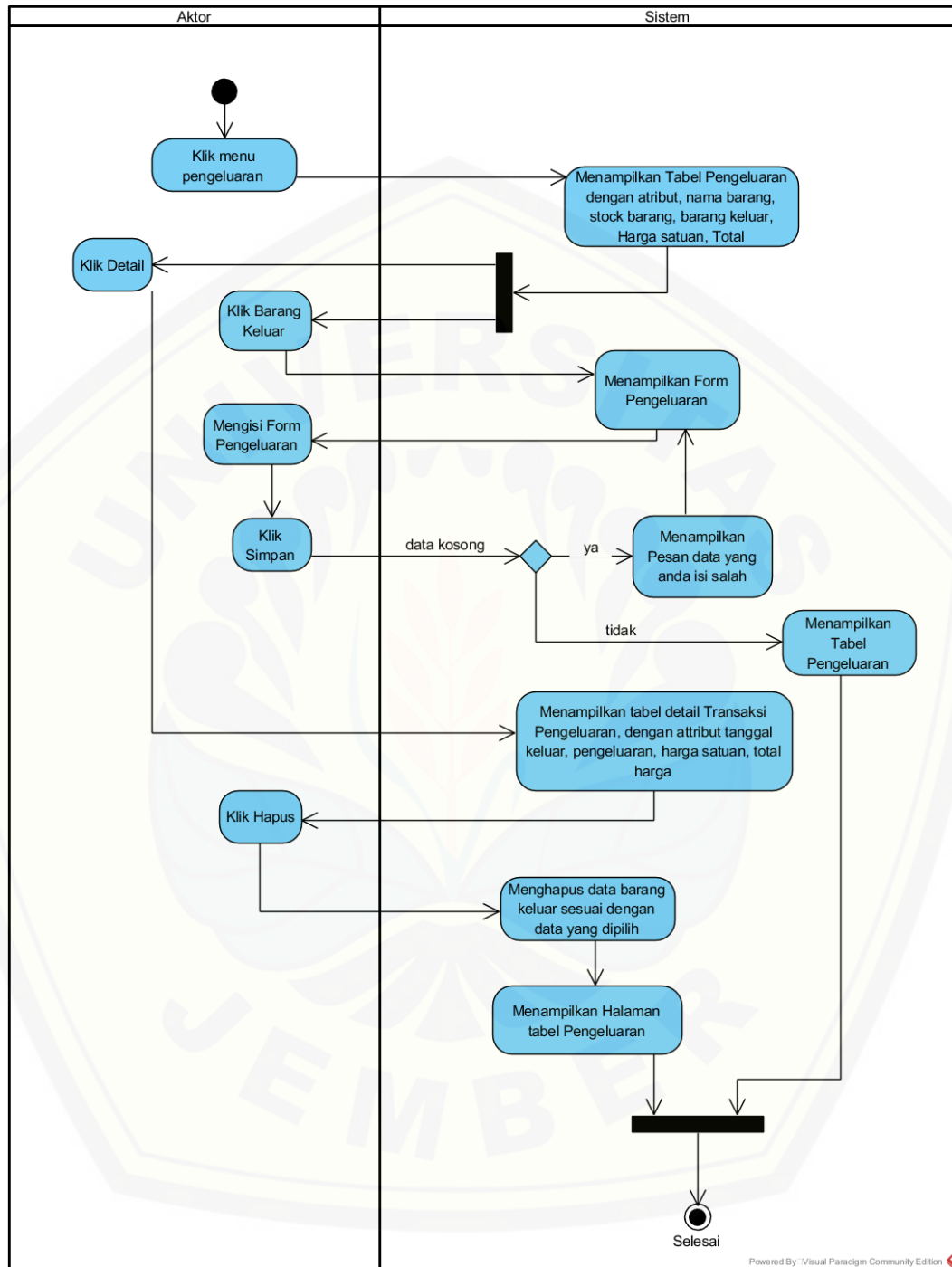
LAMPIRAN A. ACTIVITY DIAGRAM

A.1 Activity Diagram Mengelola Data Pemasukan



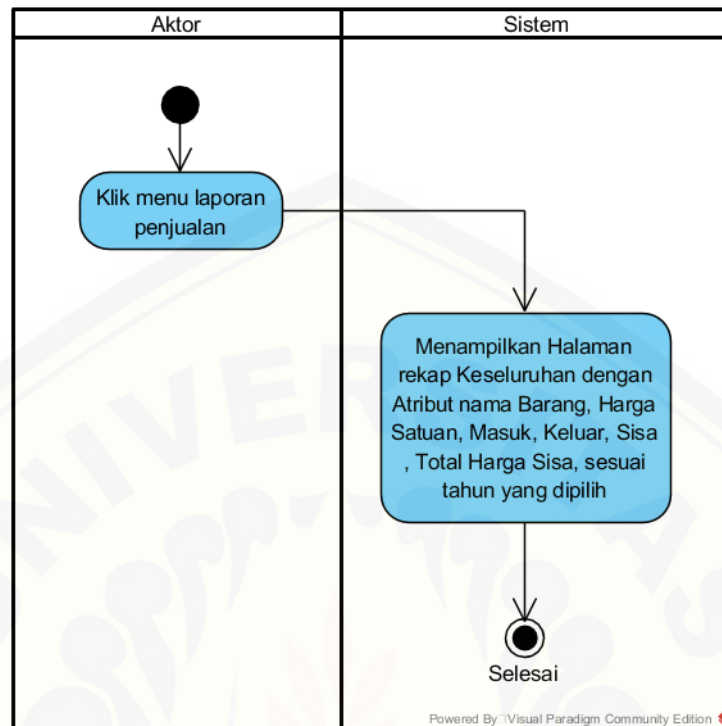
Gambar A. 1 Activity Diagram Mengelola Data Pemasukan

A.2 Activity Diagram Mengelola Data Pengeluaran



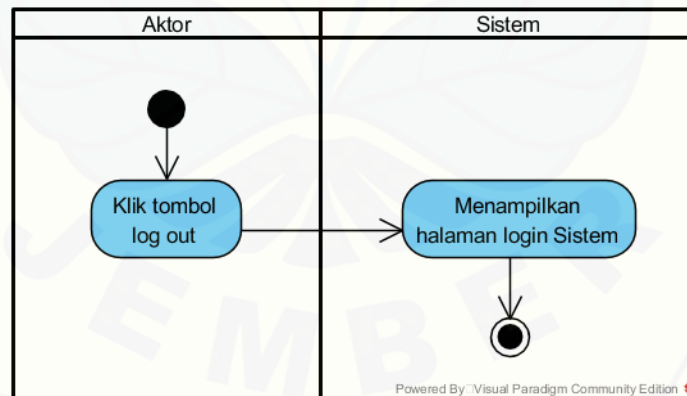
Gambar A. 2 Activity Diagram Mengelola Data Pengeluaran

A.3 Activity Diagram Melihat Rekap Keseluruhan



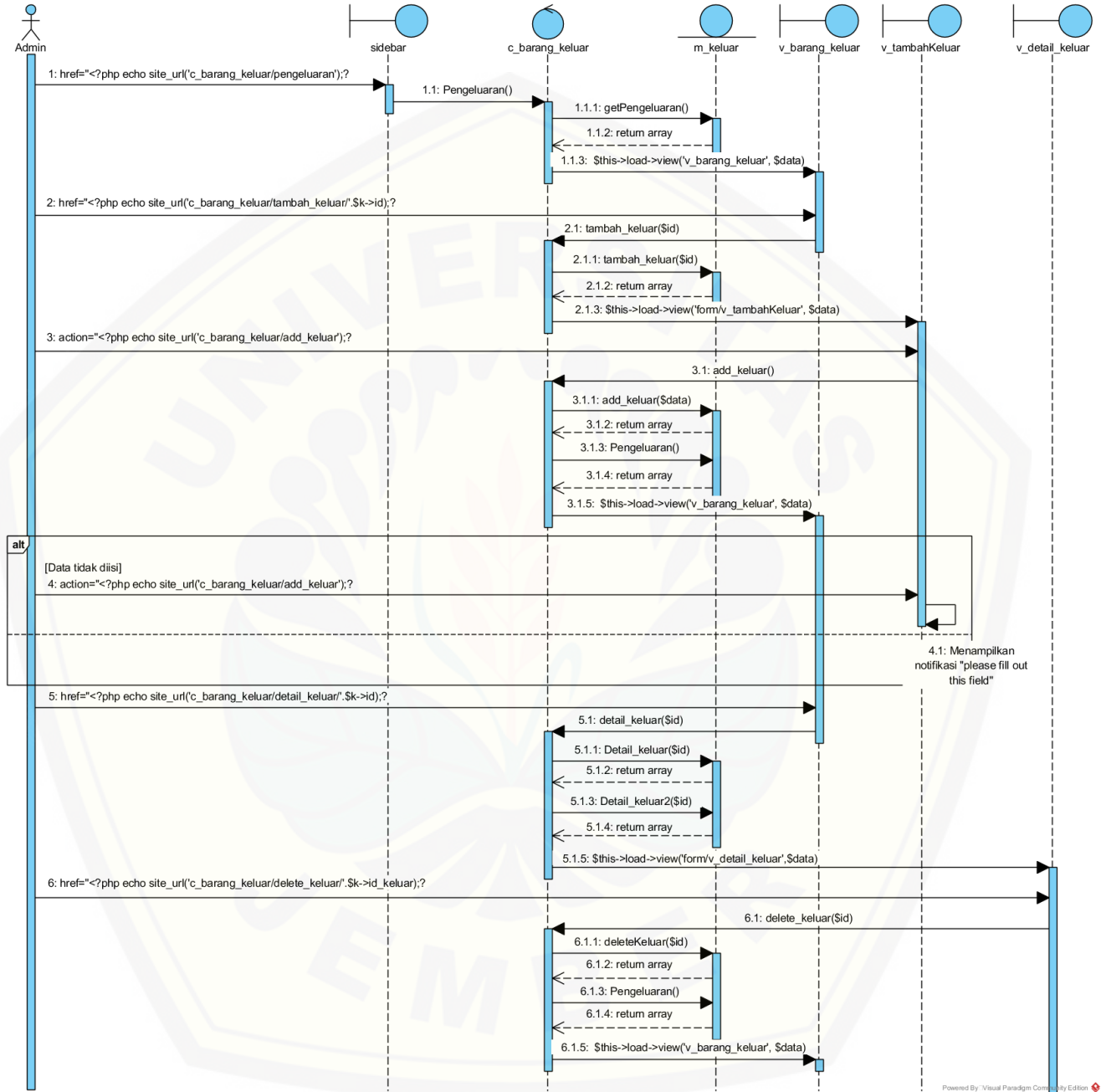
Gambar A. 3 Activity Diagram Melihat Rekap Keseluruhan

A.4 Activity Diagram Logout



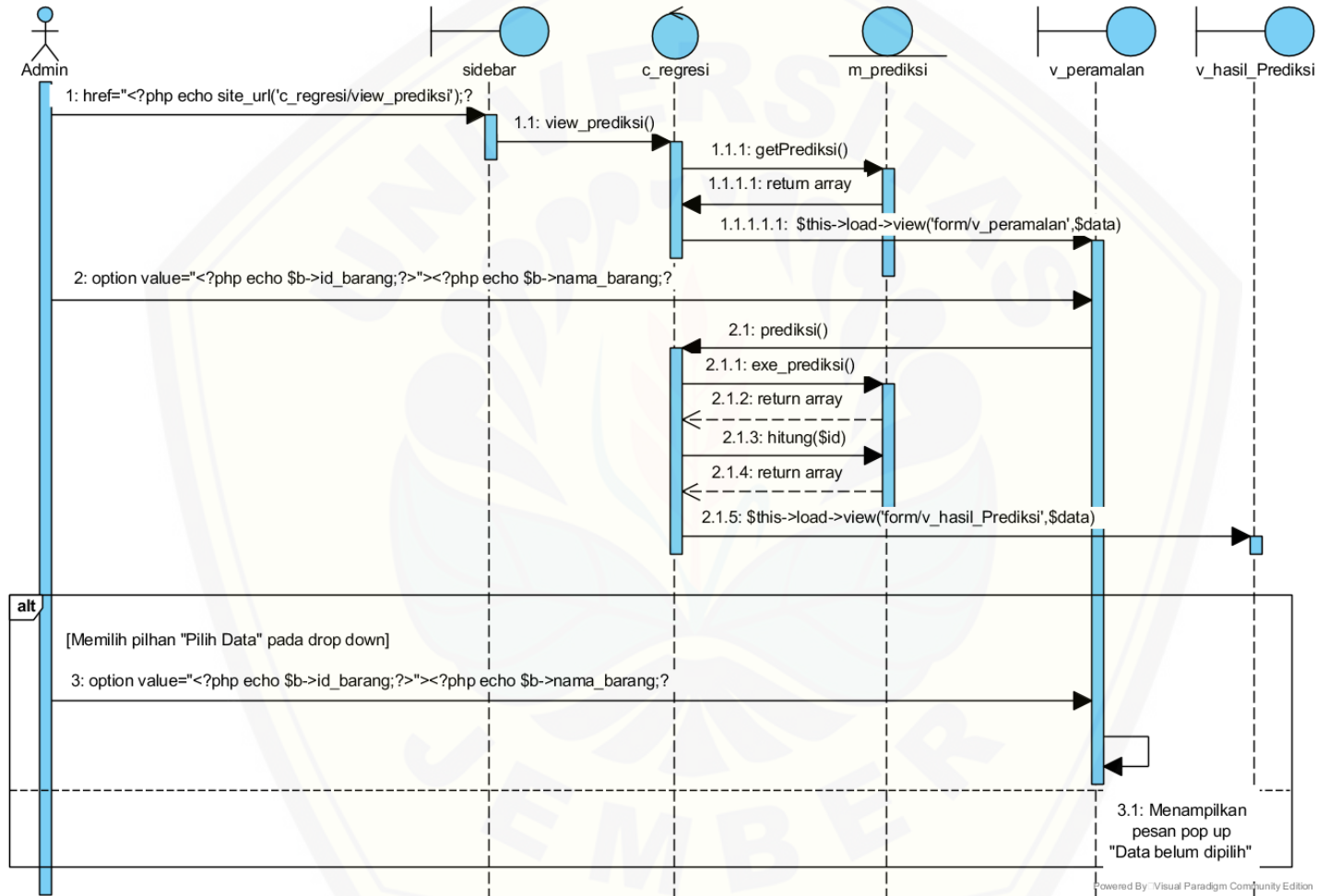
Gambar A. 4Activity Diagram Logout

B.2 Sequence Diagram Mengelola Data Transaksi Pengeluaran



Gambar B. 2 Sequence Diagram Mengelola Data Transaksi Pengeluaran

B.3 Sequence Diagram Melihat Prediksi Kebutuhan Barang Persediaan



Gambar B. 3 Sequence Diagram Melihat Prediksi Kebutuhan Barang Persediaan

LAMPIRAN C. KODE PROGRAMC.1 Kode Program *Login* (class controllers/login)Tabel C. 1 Kode Program *Login* (class controllers/login)

KODE
<pre>public function __construct(){ parent::__construct(); \$this->load->helper('url'); \$this->load->library('session'); \$this->load->model('M_login'); } public function login(){ \$username = \$this->input->post('username'); \$password = \$this->input->post('password'); \$this->load->model('m_login'); \$hasil = \$this->M_login->cek_user(\$username, \$password); if (\$hasil->num_rows() == 1) { foreach (\$hasil->result() as \$sess) { \$sess_data['level'] = \$sess->level; \$sess_data['id_user'] = \$sess->id_user; \$sess_data['username'] = \$sess->username; \$this->session->set_userdata(\$sess_data); } if (\$this->session->userdata('level')==1) { redirect(site_url('welcome/home')); } } else { echo "<script>alert('Gagal login: Cek username, password!');history.go(- 1);</script>"; } } public function logout(){ \$this->session->sess_destroy(); redirect(base_url()); } }</pre>

C.2 Kode Program Login (*class models/m_login*)Tabel C. 2 Kode Program *Login* (*class models/m_login*)

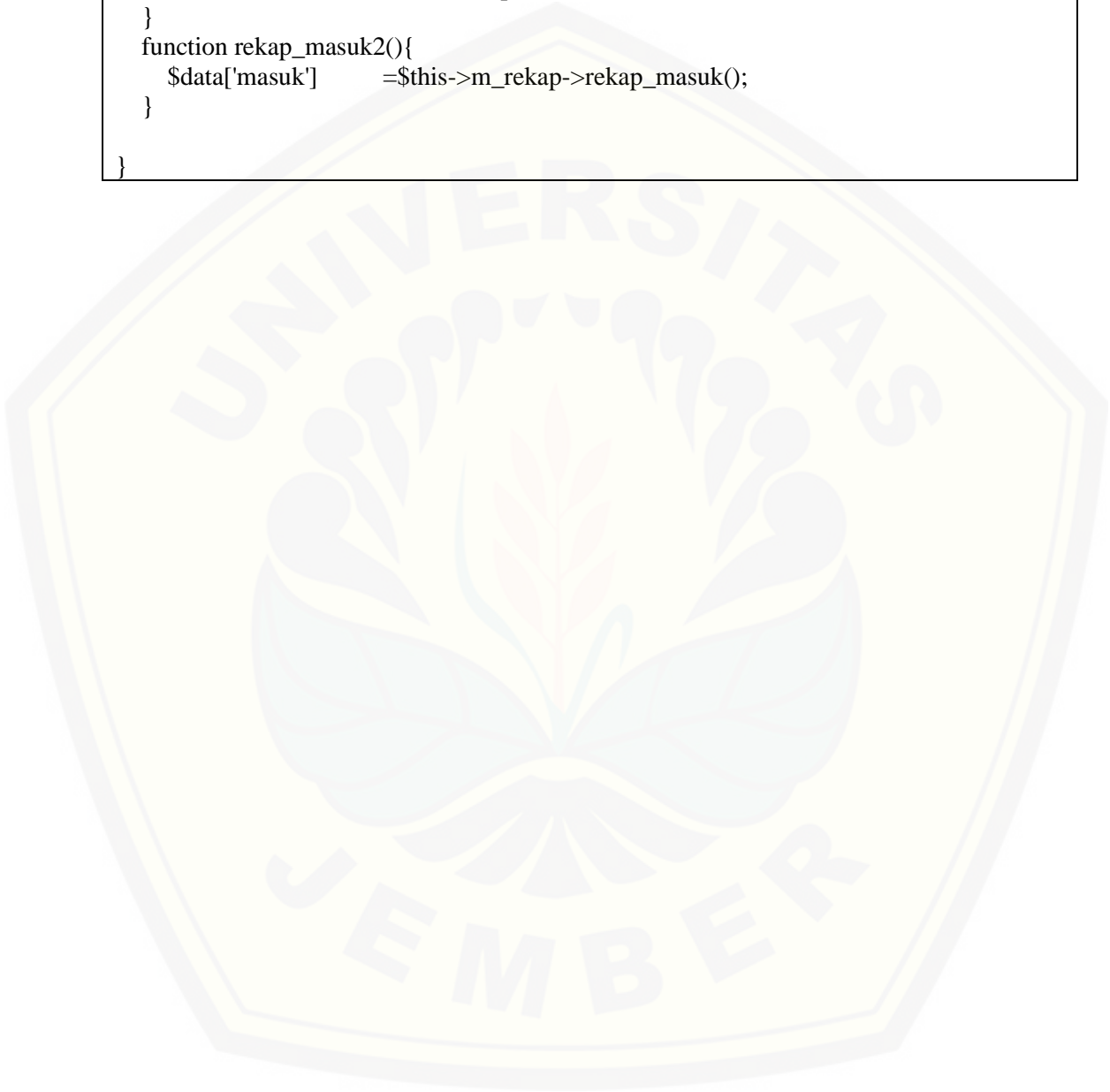
KODE
<pre><?php class m_login extends CI_Model { function __construct(){ parent::__construct(); } function cek_user(\$username, \$password){ \$this->db->select('*'); \$this->db->from('login'); \$this->db->where('username', \$username); \$this->db->where('password', \$password); return \$this->db->get(); } } }</pre>

C.3 Kode Program Mengelola Data Transaksi Pemasukan (*class**controllers/c_barang_masuk*)Tabel C. 3 Kode Program Mengelola Data Transaksi Pemasukan (*class*
controllers/c_barang_masuk)

KODE
<pre> public function index(){ \$this->load->view('sidebar'); } public function pemasukan(){ \$data['barang'] = \$this->m_masuk->getBarang(); \$this->load->view('sidebar'); \$this->load->view('v_barang_masuk', \$data); } public function detail(\$id){ \$data['barang'] = \$this->m_masuk->getDetail2(\$id); \$data['barang_masuk'] = \$this->m_masuk->getDetail(\$id); \$this->load->view('sidebar'); \$this->load->view('v_detail_masuk', \$data); } public function tambahbarang(){ \$data['satuan'] = \$this->m_masuk->getsatuan(); \$data['kategori'] = \$this->m_masuk->getkategori(); \$this->load->view('sidebar'); \$this->load->view('form/v_tambah_barang',\$data); } public function proses_addBarang(){ \$data['nama_barang'] = \$this->input->post('name'); \$data['jumlah_satuan'] = \$this->input->post('jml_satuan'); \$data['harga_satuan'] = \$this->input->post('hrga_satuan'); \$data['id_satuan'] = \$this->input->post('idsatuan'); \$data['tanggal'] = \$this->input->post('tgl_masuk'); \$data['id_kategori'] = \$this->input->post('idkategori'); \$data['stock'] = \$this->input->post('jml_satuan'); \$this->m_masuk->barang(\$data); \$this->session->set_flashdata('sukses', 'Data berhasil diproses'); redirect(site_url('c_barang_masuk/pemasukan')); } public function edit_masuk(\$id){ \$this->load->view('sidebar'); \$data['satuan'] = \$this->m_masuk->getsatuan(); \$data['kategori'] = \$this->m_masuk->getkategori(); \$data['barang'] = \$this->m_masuk->getedit_barang(\$id)->row(); \$this->load->view('form/V_edit_barang', \$data); } </pre>

```
public function editBarang($id){
    $data['nama_barang']    = $this->input->post('name');
    $data['harga_satuan']   = $this->input->post('hrg_satuan');
    $data['id_satuan']      = $this->input->post('idsatuan');
    $data['id_kategori']    = $this->input->post('idkategori');
    $this->m_masuk->editBarang($id,$data);
    $this->session->set_flashdata('sukses', 'Data berhasil diproses');
    redirect(site_url('c_barang_masuk/pemasukan'));
}
public function restock($id){
    $this->load->view('sidebar');
    $data['barang']        = $this->m_masuk->getRestock($id)->row();
    $this->load->view('form/v_restock', $data);
}
public function tambah_stock($id){
    $data['id_barang']     = $this->input->post('id');
    $data['jumlah_masuk']  = $this->input->post('jml_satuan');
    $data['tgl_masuk']     = $this->input->post('tanggal');
    $data['no_kwitansi']  = $this->input->post('nobast');
    $data['id_user']       = $this->input->post('id_user');
    $this->m_masuk->tambah_stock($data);
    $this->session->set_flashdata('sukses', 'Data berhasil diproses');
    redirect(site_url('c_barang_masuk/pemasukan'));
}
public function edit_detail($id){
    $this->load->view('sidebar');
    $data['satuan'] = $this->m_masuk->getsatuan();
    $data['kategori'] = $this->m_masuk->getkategori();
    $data['barang'] = $this->m_masuk->getedit_detail($id)->row();
    $this->load->view('form/V_edit_detail', $data);
}
public function edit_brg_detail($id){
    $data['no_kwitansi']    = $this->input->post('bast');
    $data['tgl_masuk']      = $this->input->post('tanggal');
    $this->m_masuk->editDetail($id,$data);
    $this->session->set_flashdata('sukses', 'Data berhasil diproses');
    redirect(site_url('c_barang_masuk/pemasukan'));
}
public function delete_detail($id){
    $this->m_masuk->deleteDetail($id);
    $this->session->set_flashdata('sukses', 'Data berhasil dihapus');
    redirect(site_url('c_barang_masuk/pemasukan'));
}
```

```
function rekap_masuk(){
    $data['masuk']      =$this->m_rekap->rekap_masuk();
    $this->load->view('sidebar');
    $this->load->view('form/v_rekapMasuk',$data);
}
function rekap_masuk2(){
    $data['masuk']      =$this->m_rekap->rekap_masuk();
}
}
```

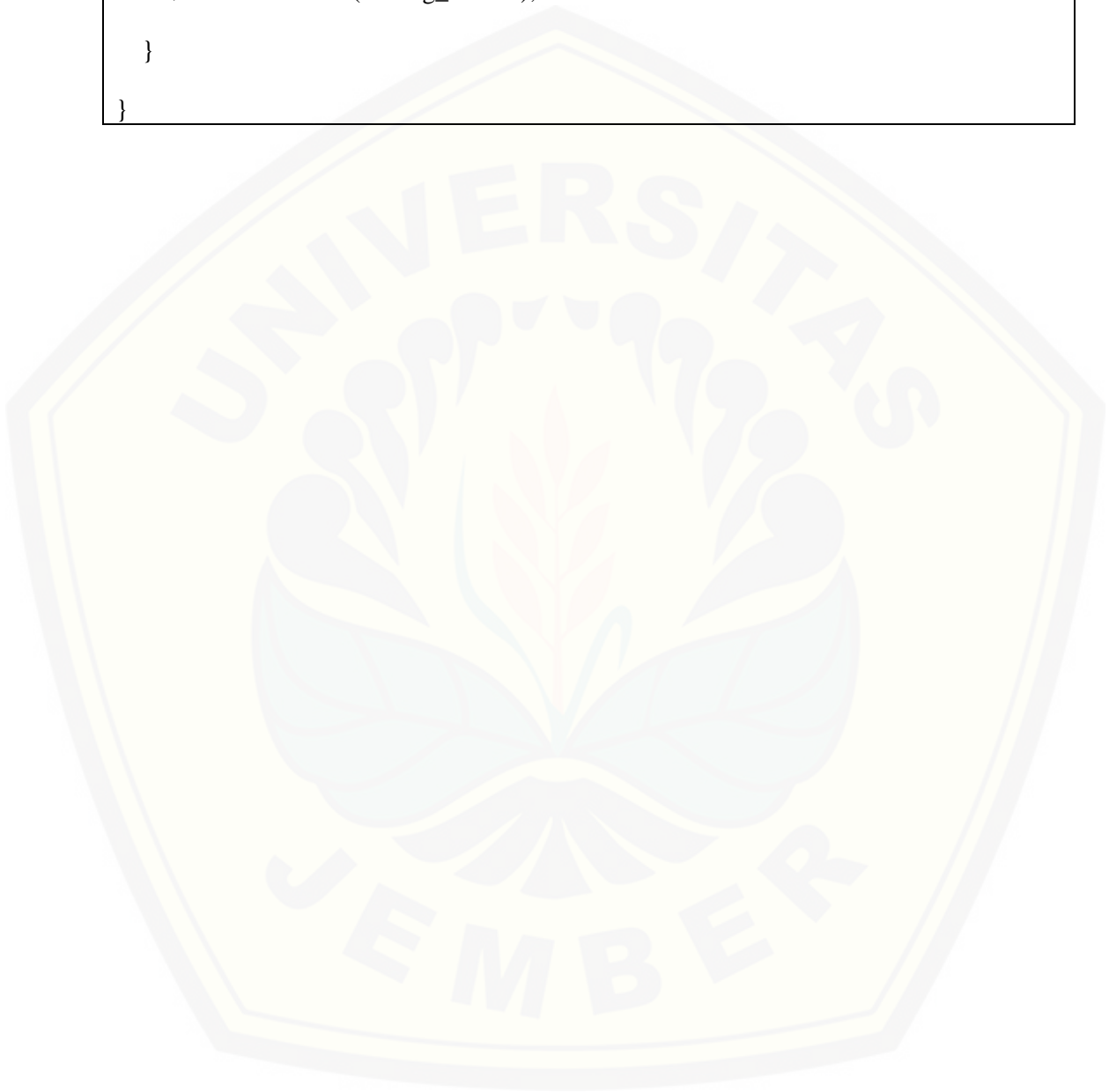


C.4 Kode Program Mengelola Data Transaksi Pemasukan (*class**models/m_masuk*)Tabel C. 4 Kode Program Mengelola Data Transaksi Pemasukan (*class models/m_masuk*)

KODE
<pre> function getBarang(){ \$this->db->select('*'); \$this->db->join('barang_masuk' , 'barang_masuk.id_barang=id_barang'); \$this->db->join('satuan' , 'satuan.id_satuan=barang.id_satuan'); \$this->db->join('kategori' , 'kategori.id_kategori=barang.id_kategori'); \$this->db->from('barang'); \$this->db->group_by('barang.id'); \$barang=\$this->db->get(); return \$barang->result(); } function getsatuan(){ \$this->db->select('*'); \$this->db->from('satuan'); \$satuan=\$this->db->get(); return \$satuan->result(); } function getkategori(){ \$this->db->select('*'); \$this->db->from('kategori'); \$kategori=\$this->db->get(); return \$kategori->result(); } function barang(\$data){ \$this->db->insert('barang' , \$data); } function getedit_barang(\$id){ \$this->db->select('*'); \$this->db->join('kategori' , 'kategori.id_kategori=barang.id_kategori'); \$this->db->join('satuan' , 'satuan.id_satuan=barang.id_satuan'); \$this->db->from('barang'); \$this->db->where('id' , \$id); return \$this->db->get(); } function editBarang(\$id,\$data){ \$this->db->where('id' , \$id); \$this->db->update('barang' , \$data); } function getDetail(\$id){ </pre>

```
$this->db->select('*');
$this->db->where('id',$id);
$this->db->join('barang_masuk', 'barang.id=barang_masuk.id_barang');
$this->db->join('satuan', 'barang.id_satuan=satuan.id_satuan');
$this->db->join('kategori', 'barang.id_kategori=kategori.id_kategori');
$this->db->from('barang');
$barang=$this->db->get();
return $barang->result();
}
function getDetail2($id){
    $this->db->select('*');
    $this->db->join('satuan', 'satuan.id_satuan=barang.id_satuan');
    $this->db->join('kategori', 'kategori.id_kategori=barang.id_kategori');
    $this->db->from('barang');
    $this->db->where('id', $id);
    $barang=$this->db->get();
    return $barang->result();
}
function getedit_detail($id){
    $this->db->select('*');
    $this->db->from('barang_masuk');
    $this->db->join('barang', 'barang.id=barang_masuk.id_barang');
    $this->db->join('kategori', 'kategori.id_kategori=barang.id_kategori');
    $this->db->join('satuan', 'satuan.id_satuan=barang.id_satuan');
    $this->db->where('id_masuk', $id);
    return $this->db->get();
}
function getRestock($id){
    $this->db->select('*');
    $this->db->join('kategori', 'kategori.id_kategori=barang.id_kategori');
    $this->db->join('satuan', 'satuan.id_satuan=barang.id_satuan');
    $this->db->from('barang');
    $this->db->where('id', $id);
    return $this->db->get();
}
function tambah_stock($data){
    $this->db->insert('barang_masuk', $data);
}
function update_stock($id, $data2){
    $this->db->update('barang', $data2);
    $this->db->where('id', $id);
}
function editDetail($id,$data){
    $this->db->where('id_masuk', $id);
    $this->db->update('barang_masuk', $data);
```

```
}  
function deleteDetail($id){  
    $this->db->where('id_masuk',$id);  
    $this->db->delete('barang_masuk');  
  
}  
}
```



C.5 Kode Program Mengelola Data Transaksi Pengeluaran (*class**controllers/c_barang_keluar*)Tabel C. 5 Kode Program Mengelola Data Transaksi Pengeluaran (*class controllers/c_barang_keluar*)

KODE
<pre> public function Pengeluaran(){ \$data['keluar'] = \$this->m_keluar->getPengeluaran(); \$this->load->view('sidebar'); \$this->load->view('v_barang_keluar', \$data); } public function detail_keluar(\$id){ \$data['keluar'] = \$this->m_keluar->Detail_keluar(\$id); \$data['barang_keluar'] = \$this->m_keluar->Detail_keluar2(\$id); \$this->load->view('sidebar'); \$this->load->view('form/v_detail_keluar',\$data); } public function tambah_keluar(\$id){ \$data['keluar'] = \$this->m_keluar->tambah_keluar(\$id)->row(); \$this->load->view('sidebar'); \$this->load->view('form/v_tambahKeluar', \$data); } function add_keluar(){ \$data['id_barang'] = \$this->input->post('id'); \$data['jumlah_keluar'] = \$this->input->post('jml_keluar'); \$data['tgl_keluar'] = \$this->input->post('tanggal'); \$data['no_bast'] = \$this->input->post('nobast'); \$data['id_user'] = \$this->input->post('id_user'); \$this->m_keluar->add_keluar(\$data); \$this->session->set_flashdata('sukses', 'Data berhasil diproses'); redirect(site_url('c_barang_keluar/Pengeluaran')); } function delete_keluar(\$id){ \$this->m_keluar->deleteKeluar(\$id); \$this->session->set_flashdata('sukses', 'Data berhasil dihapus'); redirect(site_url('c_barang_keluar/Pengeluaran')); } </pre>

C.6 Kode Program Mengelola Data Transaksi Pengeluaran(*class*
modelss/m_keluar)

Tabel C. 6 Kode Program Mengelola Data Transaksi Pengeluaran(*class*
modelss/m_keluar)

KODE
<pre> function getPengeluaran(){ \$this->db->select('*'); \$this->db->from('barang'); \$this->db->join('satuan', 'satuan.id_satuan=barang.id_satuan'); \$this->db->join('kategori', 'kategori.id_kategori=barang.id_kategori'); \$keluar=\$this->db->get(); return \$keluar->result(); } function Detail_keluar(\$id){ \$this->db->select('*'); \$this->db->from('barang'); \$this->db->join('satuan', 'satuan.id_satuan=barang.id_satuan'); \$this->db->join('kategori', 'kategori.id_kategori=barang.id_kategori'); \$this->db->where('id', \$id); \$keluar=\$this->db->get(); return \$keluar->result(); } function Detail_keluar2(\$id){ \$this->db->select('*'); \$this->db->from('barang'); \$this->db->join('satuan', 'satuan.id_satuan=barang.id_satuan'); \$this->db->join('kategori', 'kategori.id_kategori=barang.id_kategori'); \$this->db->join('barang_keluar', 'barang_keluar.id_barang=barang.id'); \$this->db->where('id', \$id); \$barang_keluar=\$this->db->get(); return \$barang_keluar->result(); } function tambah_keluar(\$id){ \$this->db->select('*'); \$this->db->from('barang'); \$this->db->join('satuan', 'satuan.id_satuan=barang.id_satuan'); \$this->db->join('kategori', 'kategori.id_kategori=barang.id_kategori'); \$this->db->where('id', \$id); return \$this->db->get(); } function add_keluar(\$data){ \$this->db->insert('barang_keluar', \$data); } function deleteKeluar(\$id){ </pre>


```
$this->db->where('id_keluar',$id);  
$this->db->delete('barang_keluar');  
}  
}
```



C.7 Kode Program Melihat Prediksi Kebutuhan Barang Persediaan(*class controllers/c_regresi*)

Tabel C. 7 Kode Program Melihat Prediksi Kebutuhan Barang Persediaan(*class controllers/c_regresi*)

KODE
<pre>public function view_prediksi(){ \$data['penjualan'] = \$this->m_prediksi->getPrediksi(); \$this->load->view('sidebar'); \$this->load->view('form/v_peramalan',\$data); } public function prediksi() { \$data['penjualan'] = \$this->m_prediksi->exe_prediksi(); \$id = \$this->input->get_post('id'); \$data = \$this->m_prediksi->hitung(\$id); \$this->load->view('form/v_hasil_Prediksi',\$data); } }</pre>

C.8 Kode Program Melihat Rekap Keseluruhan (*class controllers/c_rekap*)Tabel C. 8 Kode Program Melihat Rekap Keseluruhan (*class controllers/c_rekap*)

KODE
<pre>function rekap_tahunan(){ // \$this->m_rekap->rekap_masuk(); \$data['barang'] =\$this->m_rekap->rekap_tahunan(); \$this->load->view('sidebar'); \$this->load->view('form/v_rekap',\$data); } }</pre>

C.9 Kode Program Melihat Rekap Keseluruhan (*class models/m_rekap*)Tabel C. 9 Kode Program Melihat Rekap Keseluruhan (*class models/m_rekap*)

KODE
<pre>function rekap_tahunan(){ \$this->db->select('*'); \$this->db->from('barang'); \$this->db->join('kategori', 'barang.id_kategori=kategori.id_kategori'); \$this->db->join('satuan', 'barang.id_satuan=satuan.id_satuan'); \$masuk=\$this->db->get(); return \$masuk->result(); } }</pre>