



**SISTEM PENGENDALIAN PERSEDIAAN STOK BARANG
MENGUNAKAN METODE DOUBLE EXPONENTIAL SMOOTHING
DAN ECONOMY ORDER QUANTITY (EOQ)
(STUDI KASUS UD JASMINE)**

SKRIPSI

Oleh
Deasy Wulansari
NIM 122410101006

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER
2016**



**SISTEM PENGENDALIAN PERSEDIAAN STOK BARANG
MENGUNAKAN METODE DOUBLE EXPONENTIAL SMOOTHING
DAN ECONOMY ORDER QUANTITY (EOQ)
(STUDI KASUS UD JASMINE)**

SKRIPSI

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

Oleh
Deasy Wulansari
NIM 122410101006

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER
2016**

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Allah SWT, yang telah memberikan kelancaran dan kemudahan dalam menyelesaikan skripsi ini;
2. Ibunda Ninik Indahwati serta Ayahanda Sukarnomo atas segala kesabaran, keikhlasan, limpahan kasih sayang, doa, dan motivasi serta dukungan yang luar biasa;
3. Kakak dan adik saya tercinta Ners Ria Monikasari S.Kep dan adikku Ilham Gustri Putra;
4. Saudara-saudaraku berserta seluruh keluarga besar;
5. Sahabat kontrakan Puri Bunga Nirwana yaitu Lintang, Puput, Sefty, Kokom, Ara, Gau, Dwi, Jeje, Esa dan Delia yang selalu memberikan dukungan;
6. Marceli Aditya yang telah membimbing pembuatan skripsi;
7. Teman-teman seperjuangan kuliah di PSSI yang bersama berjuang menuntut ilmu;
8. Guru-guruku sejak taman kanak-kanak hingga perguruan tinggi;
9. Almamater Program Studi Sistem Informasi Universitas Jember.

MOTO

“Sesungguhnya bersama kesulitan ada kemudahan. Maka apabila engkau telah selesai (dari sesuatu urusan), tetaplah bekerja keras (untuk urusan orang lain). Dan hanya kepada Tuhanmulah engkau berharap”.

(QS. Al-Insyirah : 6-8)

”Manusia tidak akan mengetahui kekuatan maksimalnya, sampai ia berada dalam kondisi dimana ia dipaksa kuat untuk bertahan.”

(Merry Riana)

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama : Deasy Wulansari

NIM : 122410101006

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Pengendalian Persediaan Stok Barang Menggunakan Metode Double Exponential Smoothing dan Economy Order Quantity (EOQ) (Studi Kasus UD Jasmine)”, 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 adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 15 Juni 2016

Yang menyatakan,

Deasy Wulansari

NIM 122410101006

SKRIPSI

SISTEM PENGENDALIAN PERSEDIAAN STOK BARANG MENGUNAKAN METODE DOUBLE EXPONENTIAL SMOOTHING DAN ECONOMY ORDER QUANTITY (EOQ) (Studi Kasus UD Jasmine)

Oleh

Deasy Wulansari

NIM 122410101006

Pembimbing :

Dosen Pembimbing Utama : Dr. Saiful Bukhori, S.T., M.Kom

Dosen Pembimbing Pendamping : Muhammad Arief Hidayat, S.Kom, M.Kom

PENGESAHAN PEMBIMBING

Skripsi berjudul “Sistem Pengendalian Persediaan Stok Barang Menggunakan Metode Double Exponential Smoothing dan Economy Order Quantity (EOQ) (Studi Kasus UD Jasmine)”, telah diuji dan disahkan pada:

hari, tanggal : Rabu, 15 Juni 2016

tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Dr. Saiful Bukhori, S.T., M.Kom

NIP 196811131994121001

M. Arief Hidayat, S.Kom., M.Kom

NIP 198101232010121003

PENGESAHAN PENGUJI

Skripsi berjudul “Sistem Pengendalian Persediaan Stok Barang Menggunakan Metode Double Exponential Smoothing dan Economy Order Quantity (EOQ) (Studi Kasus UD Jasmine)”, telah diuji dan disahkan pada:

hari, tanggal : Rabu, 15 Juni 2016

tempat : Program Studi Sistem Informasi Universitas Jember

Tim Penguji,

Penguji I,

Penguji II,

Anang Andrianto, ST., MT

NIP. 196906151997021002

Fahrobby Adnan, S.Kom., M.MSI

NIP. 198706192014041001

Mengesahkan

Ketua Program Studi,

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

NIP. 19670420 1992011001

SKRIPSI

**SISTEM PENGENDALIAN PERSEDIAAN STOK BARANG
MENGUNAKAN METODE DOUBLE EXPONENTIAL SMOOTHING
DAN ECONOMY ORDER QUANTITY (EOQ)
(Studi Kasus UD Jasmine)**

Oleh

Deasy Wulansari

NIM 122410101006

Pembimbing :

Dosen Pembimbing Utama : Dr. Saiful Bukhori, S.T., M.Kom

Dosen Pembimbing Pendamping : Muhammad Arief Hidayat, S.Kom, M.Kom

ABSTRAK

Pengendalian persediaan perlu diperhatikan oleh perusahaan *retail* maupun *manufactur* karena sangat menentukan kelancaran kegiatan usaha dalam mencapai keuntungan sebesar-besarnya dengan biaya seminimal mungkin. Pengelolaan persediaan UD Jasmine masih dilakukan secara manual sehingga persediaan barang tidak akurat. Hal itu dapat diatasi dengan adanya suatu prediksi terhadap persediaan yang harus dilakukan untuk memenuhi jumlah permintaan pelanggan. Prediksi diharapkan dapat menentukan persediaan yang optimal. Pola data penjualan yang dimiliki oleh UD Jasmine adalah trend. Sehingga penggunaan metode Double Exponential Smoothing dalam prediksi permintaan periode selanjutnya merupakan pilihan yang tepat. Setelah hasil prediksi telah didapatkan maka melakukan proses perhitungan metode Economy Order Quantity (EOQ) yang digunakan untuk menentuka jumlah barang yang harus dipesan untuk setiap pemesanan, jumlah persediaan pengaman, dan jumlah stok barang minimal untuk melakukan pemesanan kembali. Sistem telah diimplementasikan menggunakan bahasa pemrograman PHP dengan *framework* CI dan database *MySQL*. Pada penelitian ini menggunakan *sample* data Beef Sausages 375 gram dimana metode ini dapat memberikan hasil yang cukup baik, dengan menggunakan nilai alpha (α) = 0,7 dan gamma (γ) = 0,1 menghasilkan nilai *MAPE* terkecil yaitu 5,2746404 %

Kata Kunci : Pengendalian Persediaan, *Double Exponential Smoothing*, *Economy Order Quantity*

RINGKASAN

Sistem Pengendalian Persediaan Stok Barang Menggunakan Metode *Double Exponential Smoothing* dan *Economy Order Quantity* (EOQ) (Studi Kasus UD Jasmine); Deasy Wulansari, 112410101006; 2016; 180 halaman; Program Studi Sistem Informasi Universitas Jember.

UD Jasmine merupakan agen yang bergerak di bidang bisnis *frozen food*. Sampai saat ini, UD Jasmine merupakan salah satu agen yang memiliki tingkat permintaan barang cukup tinggi. Pengelolaan persediaan barang masih menggunakan cara manual. UD Jasmine juga tidak memiliki acuan yang pasti tentang jumlah barang yang harus dipesan dan kapan harus melakukan pemesanan barang. Pemesanan baru dilakukan jika stok barang tertentu hampir habis, dengan jumlah yang mengacu pada penggunaan sebelumnya. Sehingga UD Jasmine sering mengalami kekurangan stok pada setiap barang. Tak jarang pula mengalami penumpukan barang digudang serta terdapat beberapa barang telah melampaui batas kadaluarsa.

Pada penelitian ini dibangun Sistem Pengendalian Persediaan Stok Barang menggunakan metode *Double Exponential Smoothing* dan *Economy Order Quantity* (EOQ), dimana sistem ini menggunakan data penjualan barang 6 bulan sebelum periode yang akan diprediksi. Metode *Double Exponential Smoothing* untuk meramalkan jumlah permintaan barang yang disediakan pada periode berikutnya. Sedangkan metode *Economy Order Quantity* (EOQ) digunakan dalam menentukan jumlah pemesanan yang optimal, jumlah persediaan pengaman, dan jumlah stok barang minimal untuk melakukan pemesanan kembali. Dalam sistem ini *output* metode *Double Exponential Smoothing* akan digunakan sebagai parameter input metode *Economy Order Quantity* (EOQ).

PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul Sistem Pengendalian Persediaan Stok Barang Menggunakan Metode Double Exponential Smoothing dan Economy Order Quantity (EOQ) (Studi Kasus UD Jasmine). Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

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

1. Prof. Drs. Slamin, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Dr. Saiful Bukhori, ST., M.Kom., selaku Dosen Pembimbing Utama dan Muhamad Arief Hidayat S.Kom., M.Kom., selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Prof. Drs. Slamin, M.Comp.Sc., Ph.D., selaku Dosen Pembimbing Akademik yang telah membimbing selama penulis menjadi mahasiswa;
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
5. Kakak tercinta Ria Monikasari dan Adik tersayang Ilham yang telah sangat mendukung baik dalam hal moril dan materil serta selalu memotivasi penulis;
6. Marceli Aditya yang telah membimbing pembuatan skripsi;
7. Sahabat kontrakan Puri Bunga Nirwana yaitu Lintang, Puput, Sefty, Kokom, Ara, Gau, Dwi, Jeje, Esa dan Delia yang selalu menghibur dan memberi dukungan;
8. keluarga besar Koperasi Mahasiswa Sistem Informasi (KOPMASI) periode 2013-2014 dan 2014-2015;

9. Keluarga besar FORMATION semua mahasiswa Program Studi Sistem Informasi angkatan 2012 yang telah menjadi keluarga kecil bagi penulis selama menempuh pendidikan S1
10. UD Jasmine selaku objek penelitian yang telah meluangkan waktu yang membantu dalam melancarkan pengerjaan skripsi ini.
11. Semua pihak yang tidak dapat disebutkan satu-persatu.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, oleh sebab itu penulis mengharapkan adanya masukan yang bersifat membangun dari semua pihak. Penulis berharap skripsi ini dapat bermanfaat bagi semua pihak.

Jember, 15 Juni 2015

Penulis

DAFTAR ISI

| | |
|--------------------------------|------|
| SKRIPSI..... | i |
| SKRIPSI..... | i |
| PERSEMBAHAN..... | ii |
| MOTO..... | iii |
| PERNYATAAN..... | iv |
| SKRIPSI..... | v |
| PENGESAHAN PEMBIMBING..... | vi |
| PENGESAHAN PENGUJI..... | vii |
| SKRIPSI..... | viii |
| ABSTRAK..... | ix |
| RINGKASAN..... | x |
| PRAKATA..... | xi |
| DAFTAR ISI..... | xiii |
| DAFTAR GAMBAR..... | xvi |
| DAFTAR TABEL..... | xx |
| 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..... | 4 |
| 1.5 Sistematika Penulisan..... | 4 |
| BAB 2. TINJAUAN PUSTAKA..... | 6 |
| 2.1 Penelitian Terdahulu..... | 6 |

| | | |
|---|--|----|
| 2.2 | Pengendalian Persediaan | 7 |
| 2.3 | Pola Data | 8 |
| 2.4 | Metode Double Exponential Smoothing | 10 |
| 2.5 | Metode Pengukuran Tingkat Kesalahan..... | 12 |
| 2.6 | Kuantitas Pemesanan Ekonomis (Economic Order Quantity/EOQ) | 12 |
| 2.6.1 | Kuantitas Pemesanan Ekonomis (EOQ) | 13 |
| 2.6.2 | Persediaan Pengaman (<i>Safety Stock</i>)..... | 13 |
| 2.6.3 | Titik Pemesanan Kembali (ROP)..... | 14 |
| BAB 3. METODOLOGI PENELITIAN..... | | 16 |
| 3.1 | Jenis Penelitian | 16 |
| 3.2 | Tempat dan Waktu Penelitian | 16 |
| 3.3 | Tahapan Penelitian | 17 |
| 3.4 | Metode Pengumpulan Data | 18 |
| 3.5 | Tahap Analisis | 18 |
| 3.6 | Tahap Pengembangan Sistem..... | 19 |
| BAB 4. DESAIN DAN PENGEMBANGAN SISTEM | | 23 |
| 4.1 | Statement Of Purpose (SOP)..... | 23 |
| 4.2 | Analisis Kebutuhan Perangkat Lunak | 24 |
| 4.3 | Desain Sistem | 25 |
| 4.3.1 | <i>Business Process</i> | 26 |
| 4.3.2 | Usecase Diagram..... | 26 |
| 4.3.3 | <i>Usecase Skenario</i> | 30 |
| 4.3.4 | <i>Activity Diagram</i> | 55 |
| 4.3.5 | <i>Sequence Diagram</i> | 75 |
| 4.3.6 | <i>Class Diagram</i> | 96 |
| 4.3.7 | <i>Entity Relationship Diagram</i> | 96 |
| 4.2 | Pengkodean Sistem..... | 98 |
| 4.3 | Pengujian Sistem | 99 |
| 4.3.1 | White Box | 99 |

| | | |
|----------------------------------|---|-----|
| 4.3.2 | Black Box..... | 99 |
| BAB 5. HASIL DAN PEMBAHASAN..... | | 107 |
| 5.1 | Hasil Implementasi Coding Pada Sistem Pengendalian Stok Barang UD Jasmine..... | 107 |
| 5.1.1 | Fitur <i>Login</i> | 107 |
| 5.1.2 | Home Admin..... | 108 |
| 5.1.3 | Manajemen User | 109 |
| 5.1.4 | Manajemen Jenis..... | 110 |
| 5.1.5 | Manajemen Supplier | 111 |
| 5.1.6 | Manajemen Barang | 113 |
| 5.1.7 | Manajemen Stok..... | 115 |
| 5.1.8 | Manajemen Retur Barang | 118 |
| 5.1.9 | Transaksi Penjualan | 119 |
| 5.1.10 | Melihat Laporan Penjualan | 119 |
| 5.1.11 | Manajemen Prediksi..... | 120 |
| 5.2 | Implementasi Metode Double Exponential Smoothing dan Economy Order Quantity Pada Sistem Pengendalian Persediaan Barang UD Jasmine | 121 |
| 5.3 | Pengujian Sistem Pengendalian Persediaan Stok Barang UD Jasmine..... | 126 |
| BAB 6. PENUTUP | | 145 |
| 6.1 | Kesimpulan..... | 145 |
| 6.2 | Saran..... | 146 |
| DAFTAR PUSTAKA | | 147 |
| LAMPIRAN..... | | 148 |

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 3.1 Tahapan Penelitian | 17 |
| Gambar 3.2 Alur Proses Perhitungan Metode Double Exponential Smoothing dan Economy Order Quantity | 19 |
| Gambar 3.3 Waterfall Model | 20 |
| Gambar 4.1 Businesss Process Diagram..... | 26 |
| Gambar 4.2 Usecase Diagram..... | 27 |
| Gambar 4.3 Activity Diagram Login | 56 |
| Gambar 4.4 Activity Diagram Manajemen Data User (view) | 57 |
| Gambar 4.5 Activity Diagram Manajemen Data User (input)..... | 57 |
| Gambar 4.6 Activity Diagram Manajemen Data User (edit) | 58 |
| Gambar 4.7 Activity Diagram Manajemen Data Jenis (view)..... | 59 |
| Gambar 4.8 Activity Diagram Manajemen Data Jenis (input) | 60 |
| Gambar 4.9 Activity Diagram Manajemen Data Jenis (edit)..... | 61 |
| Gambar 4.10 Activity Diagram Manajemen Data Supplier (view) | 62 |
| Gambar 4.11 Activity Diagram Manajemen Data Supplier (input)..... | 63 |
| Gambar 4.12 Activity Diagram Manajemen Data Supplier (edit) | 64 |
| Gambar 4.13 Activity Diagram Manajemen Data Barang (view) | 65 |
| Gambar 4.14 Activity Diagram Manajemen Data Barang (input)..... | 66 |
| Gambar 4.15 Activity Diagram Manajemen Data Barang (edit) | 67 |
| Gambar 4.16 Activity Diagram Manajemen Data Stok (view)..... | 68 |
| Gambar 4.17 Activity Diagram Manajemen Data Stok (input) | 69 |
| Gambar 4.18 Activity Diagram Melihat Data Stok | 70 |
| Gambar 4.19 Activity Diagram Manajemen Data Retur Barang (view) | 70 |
| Gambar 4.20 Activity Diagram Manajemen Data Retur Barang (input)..... | 72 |
| Gambar 4.21 Activity Diagram Menambah Data Penjualan | 73 |

| | |
|---|-----|
| Gambar 4.22 Activity Diagram Melihat Laporan Penjualan | 74 |
| Gambar 4.23 Activity Diagram Melihat Hasil Prediksi..... | 74 |
| Gambar 4.24 Activity Diagram Logout | 75 |
| Gambar 4.25 Sequence Diagram Login | 76 |
| Gambar 4.26 Sequence Diagram Manajemen Data User (view) | 78 |
| Gambar 4.27 Sequence Diagram Manajemen Data User (input)..... | 78 |
| Gambar 4.28 Sequence Diagram Manajemen Data User (edit)..... | 79 |
| Gambar 4.29 Sequence Diagram Manajemen Data Jenis (view)..... | 79 |
| Gambar 4.30 Sequence Diagram Manajemen Data Jenis (input) | 80 |
| Gambar 4.31 Sequence Diagram Manajemen Data Jenis (edit) | 81 |
| Gambar 4.32 Sequence Diagram Manajemen Data Supplier (view) | 82 |
| Gambar 4.33 Sequence Diagram Manajemen Data Supplier (input)..... | 83 |
| Gambar 4.34 Sequence Diagram Manajemen Data Supplier (edit)..... | 84 |
| Gambar 4.35 Sequence Diagram Manajemen Data Barang (view) | 85 |
| Gambar 4.36 Sequence Diagram Manajemen Data Barang (input)..... | 86 |
| Gambar 4.37 Sequence Diagram Manajemen Data Barang (edit)..... | 87 |
| Gambar 4.38 Sequence Diagram Manajemen Data Stok (view) | 88 |
| Gambar 4.39 Sequence Diagram Manajemen Data Stok (input)..... | 89 |
| Gambar 4.40 Sequence Diagram Melihat Data Stok | 90 |
| Gambar 4.41 Sequence Diagram Manajemen Data Retur Barang (view) | 90 |
| Gambar 4.42 Sequence Diagram Manajemen Data Retur (input) | 93 |
| Gambar 4.43 Sequence Diagram Menambah Data Penjualan | 94 |
| Gambar 4.44 Sequence Diagram Melihat Laporan Penjualan | 94 |
| Gambar 4.45 Sequence Diagram Melihat Hasil Prediksi..... | 95 |
| Gambar 4.46 Sequence Diagram Logout | 95 |
| Gambar 4.47 Class Diagram | 97 |
| Gambar 4.48 Entity Relationship Diagram..... | 98 |
| Gambar 4.49 Kode program function hitung() | 100 |
| Gambar 4.50 Diagram alir function hitung() | 101 |

| | |
|---|-----|
| Gambar 4.51 Kode program function hitunghari(\$bulan,\$tahun) | 104 |
| Gambar 4.52 Diagram alir function hitung() | 105 |
| Gambar 5.1 Halaman Login..... | 108 |
| Gambar 5.2 Halaman Dashboard Admin | 108 |
| Gambar 5.3 Halaman Data User | 109 |
| Gambar 5.4 Halaman Input Data User | 109 |
| Gambar 5.5 Halaman Edit Data User..... | 110 |
| Gambar 5.6 Halaman Data Jenis..... | 110 |
| Gambar 5.7 Halaman Tambah Data Jenis..... | 111 |
| Gambar 5.8 Halaman Edit Data Jenis | 111 |
| Gambar 5.9 Halaman Data Supplier | 112 |
| Gambar 5.10 Halaman Input Data Supplier | 112 |
| Gambar 5.11 Halaman Edit Data User..... | 113 |
| Gambar 5.12 Halaman Data Barang | 114 |
| Gambar 5.13 Halaman Input Data Barang..... | 114 |
| Gambar 5.14 Halaman Edit Data Barang..... | 115 |
| Gambar 5.15 Halaman Data Persediaan Barang | 116 |
| Gambar 5.16 Halaman Detail Persediaan Barang..... | 116 |
| Gambar 5.17 Halaman Detail Data Persediaan Barang | 117 |
| Gambar 5.18 Halaman Input Data Persediaan Barang..... | 117 |
| Gambar 5.19 Halaman Data Retur Barang | 118 |
| Gambar 5.20 Halaman Input Data Retur Barang | 118 |
| Gambar 5.21 Halaman Input Transaksi Penjualan..... | 119 |
| Gambar 5.22 Halaman Laporan Penjualan | 120 |
| Gambar 5.23 Halaman Form Prediksi..... | 120 |
| Gambar 5.24 Halaman Proses Perhitungan Prediksi | 121 |
| Gambar 5.25 Form ambil data tahun | 122 |
| Gambar 5.26 Kode program ambil data tahun | 122 |
| Gambar 5.27 Form ambil data bulan | 123 |

| | |
|---|-----|
| Gambar 5.28 Kode program ambil bulan..... | 123 |
| Gambar 5.29 Kode program perhitungan metode Double Exponential Smoothing dan Economy Order Quantity | 124 |
| Gambar 5.30 Grafik Plot Data Penjualan Beef Sausages \pm 2 tahun | 128 |
| Gambar 5.31 Hasil Prediksi Sistem dengan $\alpha = 0,7$ dan $\gamma = 0,1$ | 134 |
| Gambar 5.32 Hasil Prediksi Sistem dengan $\alpha = 0,1$ dan $\gamma = 0,2$ | 139 |
| Gambar 5.33 Grafik Hasil MAPE..... | 141 |

DAFTAR TABEL

| | |
|--|-----|
| Tabel 4.1 Deskripsi Pembagian Aktor | 27 |
| Tabel 4.2 Definisi Usecase..... | 28 |
| Tabel 4.3 Usecase Skenario Login..... | 30 |
| Tabel 4.4 Manajemen Data User (view) | 31 |
| Tabel 4.5 Usecase Skenario Manajemen Data User (input) | 32 |
| Tabel 4.6 Usecase Skenario Manajemen Data User (edit)..... | 34 |
| Tabel 4.7 Usecase Skenario Manajemen Data Jenis (view) | 35 |
| Tabel 4.8 Usecase Skenario Manajemen Data Jenis (input)..... | 36 |
| Tabel 4.9 Usecase Skenario Manajemen Data Jenis (edit) | 37 |
| Tabel 4.10 Usecase Skenario Manajemen Data Supplier (view)..... | 38 |
| Tabel 4.11 Usecase Skenario Manajemen Data Supplier (input) | 39 |
| Tabel 4.12 Usecase Skenario Manajemen Data Supplier (edit)..... | 41 |
| Tabel 4.13 Usecase Skenario Manajemen Data Barang (view) | 42 |
| Tabel 4.14 Usecase Skenario Manajemen Data Barang (input) | 43 |
| Tabel 4.15 Usecase Skenario Manajemen Data Barang (edit)..... | 44 |
| Tabel 4.16 Usecase Skenario Manajemen Data Stok (view) | 45 |
| Tabel 4.17 Usecase Skenario Manajemen Data Stok (input)..... | 46 |
| Tabel 4.18 Usecase Skenario Melihat Data Stok | 48 |
| Tabel 4.19 Usecase Skenario Manajemen Data Retur Barang (view) | 49 |
| Tabel 4.20 Usecase Skenario Manajemen Data Retur Barang (input)..... | 50 |
| Tabel 4.21 Usecase Skenario Menambah Data Penjualan | 51 |
| Tabel 4.22 Usecase Skenario Melihat Laporan Penjualan..... | 53 |
| Tabel 4.23 Usecase Skenario Melihat Hasil Prediksi | 54 |
| Tabel 4.24 Usecase Skenario Logout..... | 55 |
| Tabel 4.25 Test case pengujian fungsi hitung jalur 1..... | 102 |
| Tabel 4.26 Test case pengujian fungsi hitung jalur 2..... | 102 |
| Tabel 4.27 Test case pengujian fungsi get tahun hari jalur 3..... | 102 |

| | |
|--|-----|
| Tabel 4.28 Test case pengujian fungsi hitung jalur 4..... | 103 |
| Tabel 4.29 Test case pengujian fungsi hitung jalur 5..... | 103 |
| Tabel 4.30 Test case pengujian fungsi get tahun hari jalur 6..... | 103 |
| Tabel 4.31 Test case pengujian fungsi hitung jalur 7..... | 104 |
| Tabel 4.32 Test case pengujian fungsi hitung hari jalur 1 | 106 |
| Tabel 4.33 Test case pengujian fungsi hitung hari jalur 2 | 106 |
| Tabel 4.34 Test case pengujian fungsi hitung hari jalur 3 | 106 |
| Tabel 4.35 Test case pengujian fungsi hitung hari jalur 4 | 106 |
| Tabel 5.1 Data Penjualan Beef Sausages 350 gram \pm 2 tahun..... | 127 |
| Tabel 5.2 Data Penjualan Beef Sausages 350 gram selama 6 periode..... | 129 |
| Tabel 5.3 Prediksi Baris Pertama dengan $\alpha = 0,7$ dan $\gamma = 0,1$ | 129 |
| Tabel 5.4 Prediksi Baris Kedua dengan $\alpha = 0,7$ dan $\gamma = 0,1$ | 130 |
| Tabel 5.5 Prediksi Baris Ketiga dengan $\alpha = 0,7$ dan $\gamma = 0,1$ | 130 |
| Tabel 5.6 Prediksi Baris Keempat dengan $\alpha = 0,7$ dan $\gamma = 0,1$ | 131 |
| Tabel 5.7 Prediksi Baris Kelima dengan $\alpha = 0,7$ dan $\gamma = 0,1$ | 131 |
| Tabel 5.8 Prediksi Baris Keenam dengan $\alpha = 0,7$ dan $\gamma = 0,1$ | 132 |
| Tabel 5.9 Hasil Prediksi dengan $\alpha = 0,7$ dan $\gamma = 0,1$ | 133 |
| Tabel 5.10 Hasil Prediksi Manual dengan $\alpha = 0,7$ dan $\gamma = 0,1$ | 133 |
| Tabel 5.11 Prediksi Baris Pertama dengan $\alpha = 0,1$ dan $\gamma = 0,2$ | 134 |
| Tabel 5.12 Prediksi Baris Kedua dengan $\alpha = 0,1$ dan $\gamma = 0,2$ | 135 |
| Tabel 5.13 Prediksi Baris Ketiga dengan $\alpha = 0,1$ dan $\gamma = 0,2$ | 135 |
| Tabel 5.14 Prediksi Baris Keempat dengan $\alpha = 0,1$ dan $\gamma = 0,2$ | 136 |
| Tabel 5.15 Prediksi Baris Kelima dengan $\alpha = 0,1$ dan $\gamma = 0,2$ | 136 |
| Tabel 5.16 Prediksi Baris Keenam dengan $\alpha = 0,1$ dan $\gamma = 0,2$ | 137 |
| Tabel 5.17 Hasil Prediksi dengan $\alpha = 0,1$ dan $\gamma = 0,2$ | 138 |
| Tabel 5.18 Hasil Prediksi Manual dengan $\alpha = 0,1$ dan $\gamma = 0,2$ | 138 |
| Tabel 5.19 Hasil Nilai MAPE | 140 |

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

Persediaan adalah sebagai suatu aktiva yang meliputi bahan-bahan yang disediakan dalam proses yang terdapat dalam perusahaan untuk proses produksi, serta barang-barang atau produk yang digunakan untuk memenuhi permintaan dari konsumen (Freddy, 2007). Pengendalian persediaan perlu diperhatikan oleh perusahaan *retail* maupun *manufactur* karena sangat menentukan kelancaran kegiatan usaha dalam mencapai keuntungan sebesar-besarnya dengan biaya seminimal mungkin. Pengadaan barang dibutuhkan sejumlah waktu untuk proses pemesanan barang tersebut. Sehingga dengan adanya permintaan dalam suatu perusahaan, maka permintaan suatu barang yang datang diharapkan dapat dipenuhi dengan segera pada saat adanya permintaan barang yang dilakukan konsumen.

UD Jasmine merupakan salah satu perusahaan *retail* di bidang produk makanan olahan beku atau *frozen food* di wilayah jember. UD Jasmine menjual berbagai jenis *frozen food* mulai dari produk makanan olahan ikan, udang, kepiting, cumi dan ayam yang berasal dari berbagai *brand*. Pihak toko harus melakukan pemesanan barang kepada *supplier* dalam memenuhi permintaan konsumen dimana *supplier* berada di beberapa kota yang berbeda sehingga waktu order sampai dengan barang tiba memerlukan waktu tunggu. Pemesanan baru dilakukan jika stok barang tertentu hampir habis, dengan jumlah yang mengacu pada penggunaan sebelumnya.

Permasalahan yang terjadi di UD Jasmine adalah dalam pengelolaan data barang masih menggunakan cara manual dengan menulis ke dalam buku catatan. Selama ini admin melakukan pengecekan barang setiap harinya untuk memastikan jumlah stok

per item barang. Proses pengecekan seperti ini membutuhkan waktu yang cukup lama dan rentan terhadap kesalahan sehingga harus kerja dua kali atau bahkan lebih dalam perhitungan stok barang.

Permasalahan lain, tidak adanya perkiraan jumlah barang yang akan dibeli pelanggan sehingga jumlah pembelian barang dari *supplier* sering keliru. Tak jarang ada beberapa pelanggan yang kecewa karena barang yang dipesan tidak tersedia, bahkan tidak sedikit pula barang yang tersedia berlebih yang dapat menyebabkan toko harus mengeluarkan biaya tambahan untuk penyimpanan dan pemeliharaan guna menjaga kualitas barang serta meningkatnya waste akibat terbuangnya barang yang kadaluwarsa yang tidak dapat dikembalikan lagi karena melebihi tenggang waktu yang telah disepakati dengan para *supplier*.

Berdasarkan permasalahan yang dihadapi oleh UD Jasmine, diperlukan metode yang dapat disistemkan yang sesuai dengan pola data penjualan dalam pengendalian persediaan. Hasil analisis data penjualan yang digunakan sebagai data peramalan, fakta menunjukkan bahwa data penjualan kecenderungan trend. Metode yang digunakan untuk pola data *trend* yaitu dengan menggunakan metode *Double Exponential Smoothing Holt* (Markridakis, 1992). Oleh karena itu penulis akan melakukan penelitian mengenai pengendalian persediaan pada UD Jasmine menggunakan data masa lalu selama ± 2 tahun. Metode *Double Exponential Smoothing* untuk meramalkan jumlah permintaan barang yang disediakan pada periode berikutnya. Untuk menentukan kuantitas pemesanan yang optimal, persediaan pengaman (*Safety Stock*) dan titik pemesanan kembali (*reorder point*) digunakan *Economic Order Quantity* (EOQ). Harapannya sistem ini mampu menentukan persediaan yang optimal. Optimal berarti memiliki jumlah yang pas, tidak kekurangan dan tidak berlebihan.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, dapat dirumuskan permasalahan sebagai berikut :

1. Bagaimana menerapkan metode Double Exponential Smoothing dan EOQ dalam pengendalian persediaan stok barang pada UD Jasmine?
2. Bagaimana merancang dan membangun sistem pengendalian persediaan stok pada UD Jasmine menggunakan metode Double Exponential Smoothing dan EOQ?

1.3 Tujuan dan Manfaat

Berikut merupakan tujuan yang ingin dicapai dan manfaat yang ingin diperoleh dalam penelitian ini.

1.3.1 Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah :

1. Untuk mengetahui cara penerapan metode Double Exponential Smoothing dan EOQ dalam pengendalian persediaan stok barang pada UD Jasmine.
2. Untuk merancang dan membangun sistem pengendalian persediaan stok barang pada UD Jasmine menggunakan metode Double Exponential Smoothing dan EOQ

1.3.2 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Manfaat Akademis

Hasil penelitian ini diharapkan dapat memberikan kontribusi dan masukan bagi yang membutuhkan informasi yang berhubungan dengan judul penelitian ini. Selain itu, hasil penelitian ini merupakan suatu upaya untuk menambah varian judul penelitian yang ada di Program Studi Sistem Informasi Universitas Jember.

2. Manfaat bagi Peneliti

Mengetahui proses penerapan metode Double Exponential Smoothing dan *Economy Order Quantity* (EOQ) dalam pengendalian persediaan stok barang pada UD Jasmine

3. Manfaat bagi objek penelitian

- a Memberikan inovasi baru kepada perusahaan tempat penelitian mengenai pengendalian persediaan untuk setiap bulan
- b Membantu perusahann untuk menentukan persediaan barang yang optimal.

1.4 Batasan Masalah

Adapun batasan masalah dari penelitian ini adalah :

1. Sistem hanya digunakan untuk mengelolah persedian stok barang pada UD Jasmine
2. Sistem menggunakan metode Double Exponential Smoothing dan Economy Order Quantity
3. Data penelitian yang digunakan adalah data penjualan setiap barang selama ± 2 tahun.
4. Sistem dibangun berbasis web dan menggunakan framework CI.

1.5 Sistematika Penulisan

Sistematika penulisan dalam penyusunan tugas akhir ini adalah sebagai berikut:

1. Pendahuluan

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

2. Tinjauan Pustaka

Bab ini berisi tentang materi, penelitian terdahulu dan informasi apa aja yang digunakan dalam penelitian ini.

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.

4. Pengembangan Sistem

Bab ini berisi uraian tentang langkah-langkah yang ditempuh dalam proses menganalisis dan merancang sistem yang akan dibangun meliputi desain, pengkodean, dan pengujian sistem.

5. Hasil dan Pembahasan

Bab ini menjelaskan tentang hasil dan pembahasan dari penelitian yang telah dilakukan. Dengan memaparkan hasil penelitian dan hasil percobaan pengimplementasian sistem.

6. Penutup

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

BAB 2. TINJAUAN PUSTAKA

Bagian ini dipaparkan tinjauan yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah, dan juga penelitian-penelitian terdahulu. Teori-teori ini diambil dari buku literatur, jurnal dan internet.

2.1 Penelitian Terdahulu

Penelitian-penelitian mengenai pengendalian persediaan telah banyak dilakukan, seperti penelitian yang dilakukan Noeryanti (2012) yaitu "Pemulusan eksponensial dari brown dan dari holt untuk data yang memuat trend". Dari hasil analisis menunjukkan pola data aktualnya tampak adanya trend naik, dan diselesaikan menggunakan metode pemulusan eksponensial linier satu parameter dari Brown, pemulusan eksponensial linier dua parameter dari Holt dan metode pemulusan eksponensial kudratik dari Brown. Hasil yang diperoleh menunjukkan bahwa metode yang tepat dimana memiliki tingkat error paling kecil yaitu metode Double Exponential Smoothing dari Holt dengan nilai MSE dan MAPE yang terkecil untuk $\alpha = 0,2$ dan $\lambda = 0,1$ dengan nilai MSE = 172,84 dan MAPE= 5,17.

Berdasarkan hasil analisis penelitian diatas, menyebutkan bahwa penggunaan metode *Double Exponential Smoothing dari Holt* tepat untuk memprediksi data yang bersifat trend naik. Oleh karena itu, pada permasalahan yang sama yaitu untuk memprediksi permintaan barang UD Jasmine pada penelitian ini, penulis akan menerapkan metode Double Exponential Smoothing dari Holt yang mana pola data penjualan yang dimiliki oleh UD Jasmine bersifat trend naik.

Penelitian lain yang dilakukan oleh Carien Valerie Sakkung (2011) yaitu "Perbandingan Metode EOQ (Economic Order Quantity) dan JIT (Just In Time) terhadap Efisiensi Biaya Persediaan dan Kinerja Non-Keuangan (Studi Kasus Pada Pt Indoto Tirta Mulia)". Dari penelitian tersebut, dapat ditarik kesimpulan metode EOQ lebih baik dibanding JIT karena perusahaan memiliki safety stock yang mana selaras

dengan konsep EOQ, perusahaan juga dapat mengetahui jumlah bahan baku yang harus dipesan, dan mengetahui kapan seharusnya pemesanan dilakukan kembali sehingga manajemen persediaannya lebih terkontrol.

Berdasarkan hasil analisis penelitian diatas, menyebutkan bahwa penerapan metode *Economic Order Quantity* (EOQ) adalah tepat dan efisien. Oleh karena itu, dengan kebijakan yang dimiliki UD Jasmine yaitu menghendaki adanya persediaan pengaman dimana selaras dengan konsep EOQ, penulis akan menerapkan metode EOQ untuk menentukan kuantitas pemesanan yang optimal, persediaan pengaman (*Safety Stock*) dan titik pemesanan kembali (*reorder point*).

Pada penelitian ini penulis akan menggunakan metode *Double Exponential Smoothing* dan *Economic Order Quantity* didasarkan pada data penjualan serta kebijakan yang dimiliki oleh UD Jasmine. Diharapkan dengan penerapan metode ini dapat menjadi solusi terhadap permasalahan persediaan barang yang dihadapi oleh UD Jasmine.

2.2 Pengendalian Persediaan

Persediaan adalah suatu aktiva yang meliputi barang-barang milik perusahaan dengan maksud untuk dijual dalam suatu periode usaha tertentu, atau persediaan barang-barang yang masih dalam proses produksi, ataupun persediaan barang baku yang menunggu penggunaannya dalam suatu proses produksi (Freddy, 2007). Kekurangan atau kelebihan persediaan merupakan gejala yang kurang baik. Kekurangan dapat berakibat larinya pelanggan, sedangkan kelebihan persediaan dapat berakibat pemborosan atau tidak efisien. Oleh karena itu, pengendalian persediaan berusaha agar jumlah persediaan yang ada dapat menjamin kelancaran produksi.

Menurut Sofjan Assauri (1999:224), Pengendalian persediaan adalah suatu kegiatan untuk menentukan tingkat dan komposisi dari suatu persediaan, suku cadang, bahan baku, dan barang hasil atau produksi, sehingga perusahaan dapat melindungi kelancaran produksi dan penjualan serta kebutuhan pembelanjaan

perusahaan dengan efektif dan efisien. Pada pengendalian persediaan terdapat biaya-biaya yang harus dikeluarkan oleh perusahaan, diantaranya adalah (Freddy, 2007):

1. *Holding Cost* atau *Caryyng Cost*

Biaya yang dikeluarkan akibat adanya penyimpanan barang. *Caryying cost* akan bertambah besar apabila barang yang disimpan semakin banyak.

2. *Ordering Cost* atau *Set-Up Cost*

Biaya yang berhubungan dengan pemesanan dan pengadaan barang. Biaya pemesanan ini nilainya bertambah besar jika sering dilakukannya pemesanan barang, namun apabila sekali pemesanan barang dalam jumlah besar dan jumlah frekuensi pemesanan berkurang maka biaya pemesanan juga akan berkurang

3. *Stock-Out Cost*

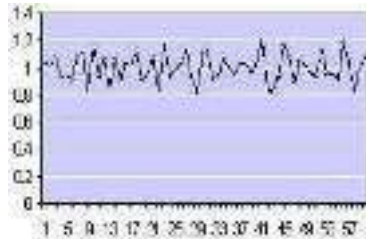
Biaya yang timbul akibat perusahaan kehabisan persediaan. Biaya ini bertambah besar ketika jumlah permintaan banyak yang tidak terpenuhi.

2.3 Pola Data

Forecasting atau peramalan adalah memperkirakan sesuatu pada waktu-waktu yang akan datang berdasarkan data masa lampau yang dianalisa secara ilmiah, khususnya menggunakan metode statistika (Supranto, 1984). Langkah penting dalam memilih metode deret berkala (*time series*) yang tepat adalah dengan mempertimbangkan jenis pola data, sehingga metode yang paling tepat dengan pola tersebut dapat diuji. Pola data dibedakan dalam empat jenis siklis (*cyclical*) dan trend yaitu sebagai berikut (Makridakis, 1992):

1. Pola Horisontal

Data yang berfluktuasi di sekitar tingkatan atau rata-rata yang konstan. Suatu proses produksi kontinyu yang tidak mengalami perubahan pada waktu tertentu termasuk jenis pola ini. Pola data horisontal atau stasioner dapat dilihat pada Gambar 2.1.

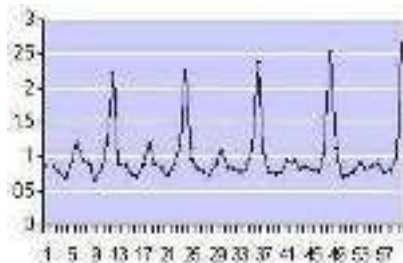


Gambar 2.1 Pola Horizontal

(Sumber: Makridakis, 1992)

2. Pola Musiman

Suatu deret data yang dipengaruhi oleh faktor musiman yang ditandai dengan adanya pola perubahan yang berulang secara otomatis (misalnya kuartal tahun tertentu, bulanan, atau hari-hari pada minggu tertentu). Penjualan produk seperti minuman ringan, es krim, dan bahan bakar pemanas ruang termasuk jenis pola ini. Pola data musiman dapat dilihat pada Gambar 2.2.



Gambar 2.2 Pola Musiman

(Sumber: Makridakis, 1992)

3. Pola Siklis

Deret data yang dipengaruhi oleh fluktuasi jangka panjang seperti yang berhubungan dengan siklus bisnis. Penjualan produk seperti mobil, baja, dan peralatan utama lain menunjukkan pola data ini. Pola data siklis dapat dilihat pada Gambar 2.3.

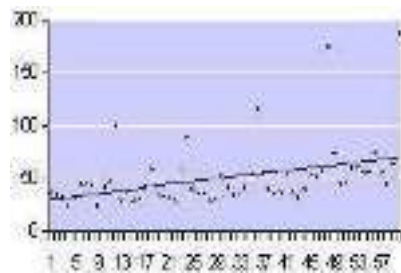


Gambar 2.3 Pola Siklis

(Sumber: Makridakis, 1992)

4. Pola Trend

Deret data menunjukkan kenaikan atau penurunan sekuler jangka panjang. Penjualan banyak perusahaan nasional (GNP) dan berbagai indikator bisnis atau ekonomi lainnya mengikuti suatu pola trend selama perubahannya sepanjang waktu. Pola data trend dapat dilihat pada Gambar 2.4.



Gambar 2.4 Pola Trend

(Sumber: Makridakis, 1992)

2.4 Metode Double Exponential Smoothing

Situasi peramalan sangat beragam dalam peristiwa horison waktu peramalan, tipe pola data, waktu peramalan dan beberapa faktor lainnya sangat berpengaruh terhadap hasil peramalan. Metode peramalan dibagi menjadi dua kategori utama yaitu metode kuantitatif dan metode kualitatif atau teknologis (Markridakis, 1992). Metode kuantitatif dapat dibagi ke dalam deret berkala (*time series*) dan metode kausal

sedangkan metode kualitatif atau teknologis dapat dibagi menjadi metode ploratoris dan normatif.

Dasar pemikiran dari metode exponential smoothing tunggal maupun ganda adalah bahwa nilai pemulusan akan terdapat pada waktu sebelum data sebenarnya apabila pada data tersebut terdapat komponen trend. Terdapat dua metode pada *Double Exponential Smoothing* yaitu *Double Exponential Smoothing* dari *Brown's* dan *Double Exponential Smoothing* dari *Holt*.

Metode yang di gunakan dalam tugas akhir ini adalah *Double Exponential Smoothing* dari *Holt*. Metode pemulusan ganda *Holt* pada prinsipnya serupa dengan brown kecuali bahwa *Holt* tidak menggunakan rumus pemulusan ganda secara langsung. Metode ini memuluskan nilai trend dengan parameter yang berbeda dari parameter yang digunakan pada deret yang asli. Ramalan dari pemulusan eksponensial linear Holt didapat dengan menggunakan dua konstanta pemulusan (dengan nilai antara 0 dan 1) yaitu α dan γ (Makridakis, 1992). Berikut ini adalah implementasi yang digunakan pada *Double Exponential Smoothing* dari *Holt*, meliputi rumus [2.1], [2.2], dan [2.3] yaitu :

$$S_t = \alpha \cdot X_t + (1 - \alpha)(S_{t-1} + b_{t-1}); \text{ untuk pemulusan} \dots\dots\dots [2.1]$$

$$b_t = \gamma \cdot (S_t - S_{t-1}) + (1 - \gamma)b_{t-1}; \text{ untuk peremajaan trend} \dots\dots\dots [2.2]$$

$$F_{t+m} = S_t + b_t m; \text{ untuk hasil peramalan} \dots\dots\dots [2.3]$$

Dimana,

S_t = Nilai pemulusan

X_t = Nilai aktual periode ke-t

b_t = Nilai trend

α = Parameter pemulusan eksponensial yang besarnya $0 < \alpha < 1$

γ = Parameter pemulusan eksponensial yang besarnya $0 < \gamma < 1$

m = Jarak periode yang akan diramalkan

F_{t+m} = hasil peramalan

2.5 Metode Pengukuran Tingkat Kesalahan

Ukuran kesalahan adalah penyimpangan antara permintaan aktual dengan hasil peramalan. Peramalan adalah hasil taksiran akan suatu nilai dimasa yang akan datang, karena masih berupa taksiran maka besar kemungkinan adanya kesalahan pada peramalan. Jika tingkat kesalahan tidak kecil, hal ini menunjukkan bahwa teknik ramalan yang digunakan salah, atau teknik tersebut perlu disesuaikan dengan mengubah parameter. Hasil proyeksi yang akurat adalah forecast yang bisa meminimalkan kesalahan meramal (forecast error). Suatu model mempunyai kinerja sangat bagus jika nilai MAPE berada di bawah 10%, dan mempunyai kinerja bagus jika nilai MAPE berada di antara 10% dan 20% (Zainun dan Majid, 2003).

Ukuran kesalahan yang di gunakan pada tugas akhir ini yaitu Mean Absolute Percentage Error (MAPE) merupakan pengukuran kesalahan yang menghitung ukuran presentase penyimpangan antara data aktual dengan data prediksi. MAPE dapat dilihat pada rumus [2.4] sebagai berikut:

$$MAPE = \frac{\sum_{t=1}^n \frac{X_t - F_{t+m}}{X_t}}{n} * 100 \dots\dots\dots[2.4]$$

Dimana,

X_t = Nilai aktual periode ke-t

F_{t+m} = Nilai ramalan periode ke-t

t = periode

n = banyaknya periode

2.6 Kuantitas Pemesanan Ekonomis (Economic Order Quantity/EOQ)

EOQ merupakan metode yang untuk menentukan jumlah pembelian bahan mentah pada setiap kali pesan dengan biaya yang paling rendah (Freddy, 2007). Hal tersebut didukung oleh Herlina (2007) yang menyatakan bahwa metode EOQ adalah metode untuk menentukan berapa jumlah pesanan yang paling ekonomis untuk satu

kali pesan. Jika persediaan bahan baku yang ada dalam perusahaan merupakan bahan baku yang dibeli dari luar dan bukan diproduksi atau dari dalam perusahaan, maka biaya yang terkait dengan persediaan diketahui sebagai biaya pemesanan (ordering costs) dan biaya penyimpanan (carrying costs).

Tujuan model persediaan ini adalah menentukan jumlah pesanan yang dapat meminimumkan biaya penyimpanan dan biaya pemesanan persediaan. Dengan menggunakan perhitungan EOQ, maka persediaan yang ada di dalam gudang tidak terlalu banyak, tapi juga tidak akan terlalu sedikit. Sehingga aktivitas perusahaan tidak akan terganggu karenanya. Metode EOQ dapat digunakan untuk menghitung kuantitas pemesanan ekonomis, safety stock, dan titik pemesanan kembali. Untuk mendapatkan nilai EOQ dibutuhkan nilai biaya pemesanan dan biaya penyimpanan.

2.6.1 Kuantitas Pemesanan Ekonomis (EOQ)

$$EOQ = \sqrt{\frac{2AD}{h}} \dots\dots\dots [2.5]$$

Dimana :

D = Permintaan per periode waktu

A = Biaya pemesanan per pesanan

h = Biaya penyimpanan per unit per periode

2.6.2 Persediaan Pengaman (*Safety Stock*)

Menurut Freddy Rangkuti (2007), Persediaan pengaman (*safety stock*) adalah persediaan yang diadakan untuk melindungi atau menjaga kemungkinan terjadinya kekurangan bahan (*stock out*). Pentingnya menghitung *safety stock* karena seringkali terjadi pesanan baru datang setelah waktu *lead time* terlampau (misalnya terlambat diperjalanan karena banjir, putusnya jembatan, atau bencana lainnya) peningkatan

permintaan produksi (peningkatan layanan) keadaan ini akan berakhir terjadinya *stock out* yang selanjutnya kan mengganguya proses produksi (Rangkuti, 2007).

Faktor-faktor yang mempengaruhi besarnya persediaan pengaman menurut Freddy Rangkuti (2007) yaitu : penggunaan bahan baku rata-rata, faktor waktu, dan biaya-biaya yang digunakan. Menghitung *Safety Stock* berdasarkan *service level*. Implementasi *Safety Stock* dapat dilihat pada rumus [2.6] dan [2.7] berikut

$$d = \frac{D}{N} \dots\dots\dots[2.6]$$

$$SS = Z \times d \times L \dots\dots\dots[2.7]$$

Dimana :

Z = *Service Level*

D = Permintaan per periode waktu

d = Penggunaan rata-rata per hari

N = lama perputaran kegiatan (hari)

L = waktu tunggu (*lead time*)

SS = Persediaan pengaman (*Safety Stock*)

2.6.3 Titik Pemesanan Kembali (ROP)

Menurut Freddy (2007), *ROP* disebut dengan batas/ titik jumlah pemesanan kembali yang harus dilakukan oleh perusahaan, sehubungan dengan adanya *leadtime* dan *safety stock*. *ROP* terjadi apabila jumlah persediaan yang terdapat dalam stok terus berkurang. Dengan demikian, perusahaan harus menentukan berapa banyak batas minimal tingkat persediaan yang harus dipertimbangkan sehingga tidak terjadi kekurangan stok. Menurut Gruendemann dan Billie (2005), Reorder Point diperoleh dari jumlah rata-rata yang diinginkan dan digunakan selama waktu jeda (*lead time*) ditambah dengan persediaan cadangan. Implementasi ROP dapat dilihat pada rumus [2.8] berikut

$$\text{ROP} = (d \times L) + \text{SS} \dots \dots \dots [2.8]$$

Dimana :

ROP = Titik pemesanan kembali (Reorder Point)

L = Waktu tunggu (Lead Time)

d = Penggunaan rata-rata per hari

SS = Persediaan cadangan (Safety Stock)

BAB 3. METODOLOGI PENELITIAN

Bab ini menjelaskan tentang jenis penelitian, tempat dan waktu penelitian, metode pengumpulan data, analisis data, dan teknik pengembangan sistem yang digunakan dalam merancang dan membangun sistem pengendalian persediaan stok barang pada UD Jasmine.

3.1 Jenis Penelitian

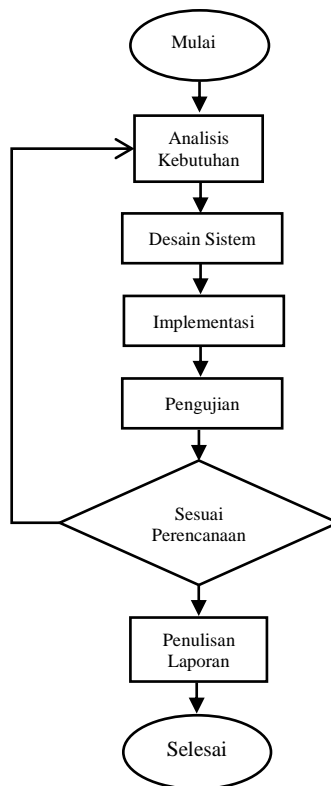
Pada penelitian ini digunakan dua jenis penelitian, yaitu penelitian kualitatif dan penelitian kuantitatif. Penelitian kualitatif adalah teknik penelitian dimana dalam pengumpulan data terjadi interaksi antara peneliti dengan yang diteliti (Sugiyono, 2012). Penggunaan metode kualitatif dalam penelitian ini adalah menganalisa studi literatur dan melakukan wawancara untuk pengumpulan sampel data yang berhubungan dengan indikator untuk menentukan persediaan barang pada UD Jasmine. Pendekatan kuantitatif adalah teknik penelitian yang berlandaskan pada sample filsafat positivisme, digunakan untuk meneliti pada populasi atau sample tertentu, pengumpulan data menggunakan instrumen penelitian, analisis data bersifat kuantitatif/ statistik, dengan tujuan untuk menguji hipotesis yang telah ditetapkan (Sugiyono, 2012). Penggunaan metode kuantitatif dalam penelitian ini adalah pengumpulan data yang digunakan dalam bentuk angka.

3.2 Tempat dan Waktu Penelitian

Tempat yang dilaksanakan untuk penelitian adalah UD Jasmine. Waktu penelitian dilakukan selama 3 (tiga) bulan, dimulai pada bulan Februari 2016 sampai dengan bulan April 2016.

3.3 Tahapan Penelitian

Penelitian ini akan dilakukan dengan beberapa tahap agar penelitian jelas, berjalan dengan baik dan teratur. Tahapan penelitian digambarkan dalam diagram alir seperti pada Gambar 3.1.



Gambar 3.1 Tahapan Penelitian

Dimana dalam gambar 3.1 menggambarkan tahapan penelitian. Pada tahapan awal penelitian, peneliti melakukan studi pustaka yang terkait dengan masalah yang ada. Wawancara kepada pihak-pihak terkait untuk mendapatkan data-data yang diperlukan dalam pembuatan sistem dilakukan setelah melakukan studi pustaka. Kemudian menganalisis data ke dalam metode yang digunakan untuk membangun dan merancang sebuah sistem. Setelah semua kebutuhan telah ditetapkan maka tahapan selanjutnya adalah pembuatan desain sistem. Tahap selanjutnya yaitu implementasi desain kedalam bahasa pemrograman. Selanjutnya tahap pengujian jika

sistem belum sesuai dengan rencana yang diharapkan maka kembali ke tahapan analisis data sedangkan jika hasil pengujian telah sesuai dengan apa yang direncanakan maka akan berlanjut ke tahap penulisan laporan dan penelitian telah selesai.

3.4 Metode Pengumpulan Data

Teknik pengumpulan data pada penelitian ini meliputi :

a. Studi Literatur

Studi literatur bertujuan untuk menyusun dasar teori yang digunakan dalam penelitian. Sumber yang dapat digunakan sebagai studi literatur seperti buku, jurnal, internet dan dokumen-dokumen yang terkait dengan pengendalian persediaan.

b. Wawancara

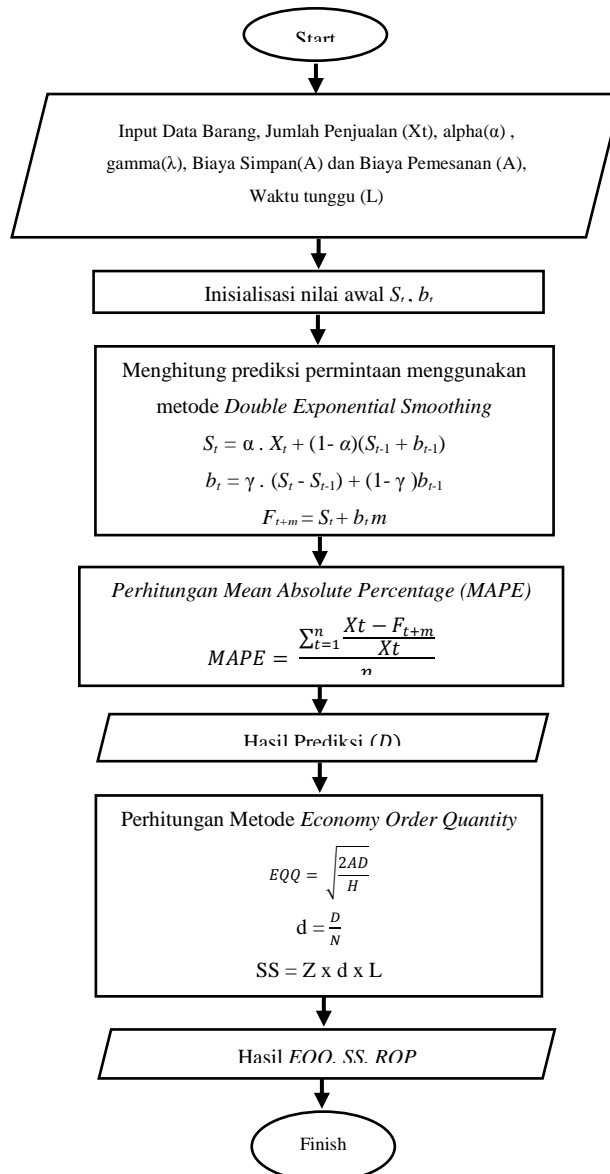
Salah satu cara untuk mengumpulkan data adalah melakukan wawancara dengan menanyakan langsung kepada narasumber. Dalam penelitian yang akan dilakukan ini penulis melakukan wawancara secara langsung pada pemilik UD Jasmine Jember.

3.5 Tahap Analisis

Tahapan analisis data dimulai dengan mengolah data-data yang telah dikumpulkan dari studi literatur dan wawancara. Data yang diperoleh akan diolah dengan metode Double Exponential Smoothing dan Economy Order Quantity yang digunakan untuk membangun sistem pengendalian persediaan stok barang.

Pada Gambar 3.2 menggambarkan tahapan-tahapan untuk melakukan perhitungan pada metode *Double Exponential Smoothing* dan *Economy Order Quantity*. Dimana data penjualan tiap bulan nantinya menjadi data acuan untuk menghitung hasil dari perhitungan prediksi. Jika data penjualan sudah di inputkan maka memulai

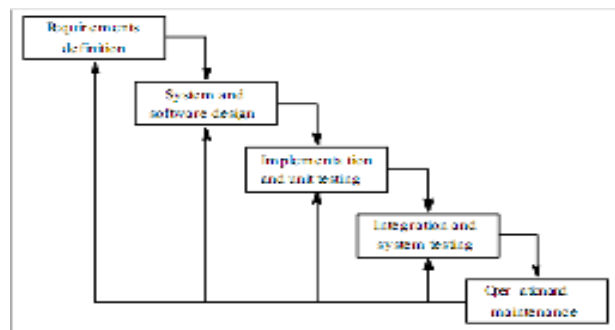
menghitung dengan rumus-rumus yang ada secara bertahap hingga ditemukan nilai tingkat kesalahannya. Setelah hasil prediksi diperoleh, maka akan memulai proses perhitungan metode *Economy Order Quantity* dengan inputan hasil prediksi, biaya penyimpanan, biaya pemesanan dan waktu tunggu.



Gambar 3.2 Alur Proses Perhitungan Metode Double Exponential Smoothing dan Economy Order Quantity

3.6 Tahap Pengembangan Sistem

Pengembangan sistem dilakukan setelah analisis pada sistem telah selesai dilakukan. Pengembangan perangkat lunak pada penelitian ini adalah metode *waterfall*. Menurut Pressman (2002) model waterfall melakukan pendekatan pengembangan perangkat lunak yang sistematis dan sekuensial dimulai pada tingkat kemajuan sistem sampai analisis, desain, kode, test, dan pemeliharaan. Sebagaimana ditunjukkan pada Gambar 3.3.



Gambar 3.3 Waterfall Model

(Sumber : Sommerville, 2003)

Penjelasan dari gambar 3.3 tahapan model Waterfall adalah sebagai berikut:

1. Analisis Kebutuhan

Analisis kebutuhan adalah proses perancangan perangkat lunak tahap pertama dimulai dari tahap merumuskan solusi dari data dan permasalahan yang diambil dari berbagai sumber yang ada. Kebutuhan yang dimaksud meliputi data kebutuhan fungsional dan data kebutuhan non fungsional. Data yang dibutuhkan merupakan data penjualan selama ± 2 tahun dan data persediaan barang.

2. Desain Sistem

Setelah semua kebutuhan telah ditetapkan maka tahapan selanjutnya adalah pembuatan desain sistem. Desain sistem yang digunakan adalah bahasa pemodelan Unified Modeling Language (UML) dan menggunakan konsep OOP. Berikut Pemodelan UML yang digunakan antara lain:

a. *Business Process*

Business Process menggambarkan proses lengkap dengan resource dan information yang dibutuhkan, event yang mendorong terjadi proses dan goal yang dituju.

b. *Usecase Diagram*

Usecase Diagram menggambarkan hubungan antara sistem dengan aktor berkaitan dengan fungsi atau tugas yang dilakukan oleh aktor.

c. *Scenario*

Scenario menjelaskan alur sistem dan keadaan yang akan terjadi ketika terjadi suatu event tertentu.

d. *Activity Diagram*

Activity Diagram mendeskripsikan aktifitas yang dibentuk dalam suatu operasi

e. *Sequence Diagram* menampilkan pesan yang dikirim dan diterima antar object

f. *Class Diagram*

Class Diagram menggambarkan relasi antar objek dan struktur sematik yang umum.

g. *Entity Relationship Diagram*

Entity Relationship Diagram menggambarkan struktur database yang akan dibangun pada system.

3. Penulisan Kode Program

Tahapan implementasi desain kedalam bahasa pemrograman dilakukan berdasarkan desain hasil analisis kebutuhan yang telah dibuat pada tahapan sebelumnya. Bahasa pemrograman yang dipakai adalah php (Hypertext Preprocessor) dengancodeigniter framework, sedangkan tool yang digunakan adalah Netbeans. DBMS yang digunakan yaitu MySQL dengan tool yang digunakan yaitu XAMPP.

4. Pengujian

Tahap ini dilakukan uji coba sistem yang telah dibuat dengan pengujian white box dan black box.

- a. *White box testing* merupakan teknik pengujian jalur dasar yang digunakan untuk menentukan kompleksitas logis dengan menentukan rangkaian dasar jalur eksekusinya. Teknik pengujian ini menggunakan pengujian jalur dasar (*basis path testing*) dimana kompleksitas dari perangkat lunak yang dibangun akan dihitung menggunakan *Cyclomatic Complexity* (Pressman, 2002). Metode siklomatik menggunakan pengujian jalur dasar (*basis path testing*) yang didalamnya terdapat beberapa tahapan pengujian antara lain pembuatan diagram alir atau grafik alir, penentuan jalur independen, penghitungan kompleksitas siklomatik jalur independen dan test case.
- b. *Black box testing* juga disebut pengujian tingkah laku, memusat pada kebutuhan fungsional perangkat lunak. Teknik pengujian *black box* memungkinkan memperoleh serangkaian kondisi masukan yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Beberapa jenis kesalahan yang dapat diidentifikasi adalah fungsi tidak benar atau hilang, kesalahan antar muka, kesalahan pada struktur data (pengaksesan basis data), kesalahan performansi, kesalahan inisialisasi dan akhir program. (Pressman, 2002)

5. Pemeliharaan

Pemeliharaan sistem diperlukan ketika sistem telah digunakan oleh user. Ketika sistem dijalankan mungkin masih terjadi kesalahan atau error yang tidak ditemukan sebelumnya. Sehingga diperlukan perbaikan pada sistem tersebut.

BAB 4. DESAIN DAN PENGEMBANGAN SISTEM

Bab ini akan menguraikan tentang proses pendesainan dan perancangan sistem untuk mengimplementasi metode Double Exponential Smoothing dan Economy Order Quantity untuk sistem pengendalian persediaan barang. Proses pendesainan dan perancangan sistem dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, dilanjutkan dengan pembuatan usecase diagram, skenario, activity diagram, sequence diagram, class diagram dan entity relation diagram (ERD).

4.1 Statement Of Purpose (SOP)

Sistem pengendalian stok barang menggunakan metode *Double Exponential Smoothing* dan *Economy Order Quantity* merupakan sebuah sistem yang mampu menampilkan hasil prediksi permintaan, jumlah pemesanan ekonomis, jumlah persediaan pengaman, serta titik pemesanan kembali pada setiap data barang untuk periode ke depan. Data yang dibutuhkan dalam membangun sistem ini, yaitu data penjualan produk setiap bulan. Tujuan dikembangkan sistem ini adalah untuk membantu bagian admin UD Jasmine dalam menentukan persediaan optimal sehingga tidak terjadi kekurangan maupun kelebihan stok barang di gudang. Sistem ini memiliki 2 hak akses yaitu admin dan kasir. Sistem ini dapat memajemen data user, data jenis, data barang, data stok barang, data pengeluaran, dan hasil prediksi. Sistem ini juga memiliki fitur *warning* data barang yang mendekati tanggal kadaluarsa.

4.2 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak dalam penelitian ini yaitu dengan cara mengidentifikasi permasalahan yang ada untuk membangun sistem pengendalian persediaan barang. Analisis kebutuhan yang dilakukan meliputi proses pengumpulan data kebutuhan fungsional dan kebutuhan non-fungsional.

4.2.1 Kebutuhan fungsional

Kebutuhan fungsional merupakan fitur-fitur pokok yang harus dapat dilakukan oleh sistem dalam menerima masukan untuk diproses sehingga menghasilkan keluaran. Kebutuhan fungsional dari sistem pengendalian stok barang UD Jasmine menggunakan metode *Double Exponential Smoothing* dan *Economy Order Quantity* yaitu:

- a Sistem dapat mengelola data user yang akan digunakan sebagai hak akses (view, input, dan edit).
- b Sistem dapat mengelola data jenis (view, input, dan edit).
- c Sistem dapat mengelola data supplier (view, input, dan edit).
- d Sistem dapat mengelola data barang (view, input, dan edit).
- e Sistem dapat mengelola data stok masuk barang berdasarkan tanggal kadaluarsa (view dan input)
- f Sistem dapat mengelola data retur barang berdasarkan tanggal kadaluarsa (view dan input).
- g Sistem dapat menginputkan jumlah pengeluaran stok barang (transaksi penjualan) yang akan digunakan sebagai data untuk melakukan prediksi.
- h Sistem dapat menampilkan hasil prediksi, jumlah kuantitas pemesanan, jumlah persediaan pengaman, dan titik pemesanan kembali setiap barang.
- i Sistem dapat menampilkan laporan penjualan barang berdasarkan interval bulan.

4.2.2 Kebutuhan non-fungsional

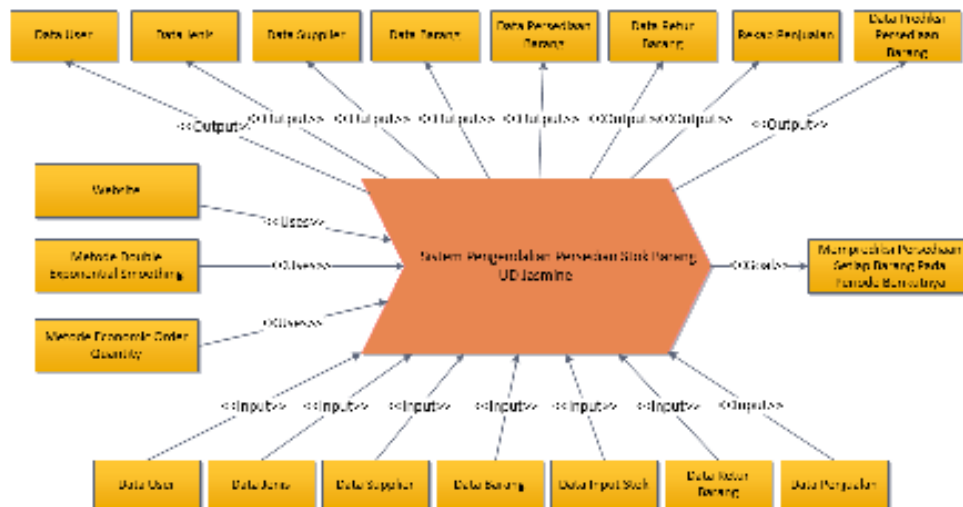
Kebutuhan non-fungsional merupakan fitur-fitur yang dimiliki untuk mendukung sistem dalam memenuhi fungsionalitasnya untuk dapat memenuhi kebutuhan dari pengguna. Kebutuhan non-fungsional dari sistem pengendalian stok barang UD Jasmine menggunakan metode *Double Exponential Smoothing* dan *Economy Order Quantity* yaitu:

- a. Sistem menggunakan username dan password untuk autentifikasi akses pengguna.
- b. Sistem menggunakan tampilan yang user friendly, sehingga pengguna tidak kesulitan untuk mengoperasikannya.

4.3 Desain Sistem

Desain perancangan sistem pengendalian persediaan stok menggunakan Unified Modeling Language (UML). Desain sistem yang akan dibuat meliputi *Bussiness Process*, *Usecase Diagram*, *Usecase Skenario*, *Activity Diagram*, *Sequence Diagram* dan *Entity Relationship Diagrams (ERD)*.

4.3.1 Business Process



Gambar 4.1 Business Process Diagram

Gambar 4.1 merupakan gambar bussiness proses dimana bussiness proses merupakan alur dari sebuah sistem informasi dimana terdapat data yang menjadi masukan (input) dan media yang digunakan sehingga akan dihasilkan data keluaran (output) dan hal yang akan dicapai oleh sebuah sistem pengendalian persediaan stok barang tersebut.

4.3.2 Usecase Diagram

Usecase diagram merupakan pemodelan yang dibuat untuk dapat menggambarkan interaksi antara aktor dengan sistem pengendalian persediaan yang akan dibangun. Usecase diagram sistem pengendalian persediaan stok barang yang terdiri dari dua aktor seperti yang dijelaskan pada Gambar 4.2.

| | | |
|----|-------|--|
| | | perhitungan hasil prediksi, manajemen data user, manajemen data jenis, manajemen data supplier, manajemen data barang, manajemen data persediaan barang, dan melihat laporan penjualan . |
| 2. | Kasir | Aktor yang hanya memiliki hak untuk menginputkan data transaksi penjualan dan melihat data stok barang |

Selain memiliki dua aktor, dalam Usecase diagram juga terdapat tujuh belas Usecase. Penjelasan dari masing – masing Usecase pada sistem pengendalian persediaan stok barang dapat di lihat pada Tabel 4.2

Tabel 4.2 Definisi Usecase

| No | Usecase | Penjelasan |
|--------|--------------------------------|---|
| USC-01 | <i>Login</i> | Fitur yang berfungsi melakukan autentifikasi hak akses aktor dalam menggunakan system |
| USC-02 | Manajemen Data User | Fitur yang dapat diakses oleh admin yang berfungsi untuk manajemen (input, edit, view) data user . |
| USC-03 | Manajemen Data Jenis | Fitur yang dapat diakses oleh admin yang berfungsi untuk manajemen (input, edit, view) data jenis barang. |
| USC-04 | Manajemen Data <i>Supplier</i> | Fitur yang dapat diakses oleh admin yang berfungsi untuk manajemen (input, edit, view) data jenis supplier. |

| | | |
|--------|--------------------------------|--|
| USC-05 | Manajemen Data Barang | Fitur yang dapat diakses oleh admin yang berfungsi untuk manajemen (input, edit, view) data barang. |
| USC-06 | Manajemen Data Stok | Fitur yang dapat diakses oleh admin yang berfungsi untuk manajemen (input, view) data stok. |
| USC-07 | Melihat Data Stok | Fitur yang dapat diakses oleh kasir yang berfungsi untuk melihat data stok. |
| USC-08 | Manajemen Data Retur Barang | Fitur yang dapat diakses oleh admin yang berfungsi untuk manajemen (input, view) data retur barang (cacat atau rusak, kadaluarsa) |
| USC-09 | Menambah Data Penjualan | Fitur yang dapat diakses oleh kasir yang berfungsi untuk menambahkan data jumlah stok yang terjual pada masing-masing barang. |
| USC-10 | Melihat Data Laporan Penjualan | Fitur yang dapat diakses oleh admin yang berfungsi untuk melihat laporan data penjualan berdasarkan barang dan interval laporan |
| USC-11 | Melihat Hasil Prediksi | Fitur yang dapat diakses oleh admin yang berfungsi untuk mengetahui hasil prediksi data persediaan menggunakan metode <i>Double Exponential Smoothing</i> dan <i>Economy Order Quantity</i> pada masing-masing barang. |

| | | |
|--------|---------------|--|
| USC-12 | <i>Logout</i> | Fitur yang berfungsi untuk keluar dari sistem. |
|--------|---------------|--|

4.3.3 *Usecase* Skenario

Usecase skenario adalah dokumentasi terhadap kebutuhan fungsional sistem. Skenario dibuat berdasarkan usecase yang telah dibuat serta berdasarkan hak akses dari sistem yang telah dideklarasikan berdasarkan *usecase* diagram. *Usecase* skenario sistem pengendalian persediaan stok barang adalah sebagai berikut.

a. *Usecase* Skenario Login

Usecase skenario ini menjelaskan alur untuk proses login untuk dapat mengakses fitur sistem. Aktor yang melakukan login adalah admin dan kasir. Setiap *user* harus memiliki username dan password yang terdaftar pada sistem. Hanya user berstatus aktif yang memiliki hak akses terhadap sistem. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif usecase skenario login dapat dilihat pada Gambar 4.3.

Tabel 4.3 Usecase Skenario Login

| | |
|-----------------------|---------------------------------------|
| Nomor Usecase | USC-01 |
| Nama | Login |
| Aktor | Admin dan Kasir |
| Pre Condition | Admin dan Kasir akan melakukan login |
| Post Condition | Admin dan Kasir telah melakukan login |

SKENARIO NORMAL "Login"

- | | |
|--------------------------|------------------------------|
| 1. Menjalankan system | 2. Menampilkan halaman login |
| 3. Inputkan username dan | |

| | |
|--|--|
| password | |
| 4. Menekan button Sign in | 5. Menampilkan halaman dashboard sesuai level masing-masing |
| SKENARIO ALTERNATIF “Salah Username dan Password” | |
| 4a. Menekan button Sign in | 5a. Menampilkan alert “Warning ! Username atau Password tidak sesuai!” |
| SKENARIO ALTERNATIF “Belum Mengisi Field” | |
| 4b. Menekan button Sign in | 5b. Menampilkan pesan “Please fill out this field” |

b. *Usecase* Skenario Manajemen Data *User (view)*

Usecase skenario ini menjelaskan alur untuk proses melihat data user. Aktor yang dapat melihat data user adalah admin. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario manajemen data *user (view)* dapat dilihat pada Tabel 4.4.

Tabel 4.4 Manajemen Data User (view)

| | |
|-----------------------|---|
| Nomor Usecase | USC-02 |
| Nama | Manajemen Data User (view) |
| Aktor | Admin |
| Pre Condition | Admin harus melakukan login ke dalam system |
| Post Condition | Admin telah melakukan view data user |

| |
|---|
| SKENARIO NORMAL “Manajemen Data User (view)” |
|---|

1. Klik menu “User”

2. Menampilkan halaman user

c. *Usecase* Skenario Manajemen Data *User* (*input*)

Usecase skenario ini menjelaskan alur untuk proses menambah data user. Aktor yang dapat menambah data *user* adalah admin. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen data *user* (*input*) dapat dilihat pada Tabel 4.5.

Tabel 4.5 *Usecase* Skenario Manajemen Data *User* (*input*)

| | |
|-----------------------|---|
| Nomor Usecase | USC-02 |
| Nama | Manajemen Data <i>User</i> (<i>input</i>) |
| Aktor | Admin |
| Pre Condition | Admin akan melakukan input data user |
| Post Condition | Aktor telah melakukan input data user |

SKENARIO NORMAL
“Manajemen Data *User* (*input*)”

1. Klik menu “user”

2. Menampilkan halaman user

3. Klik button “insert”

4. Menampilkan form insert data user

5. Mengisi seluruh isian form insert data user

6. Klik button “save”

4. Menyimpan data dalam database

5. Menampilkan pesan “Berhasil tambah data user !”

6. Menampilkan halaman user

**SKENARIO ALTERNATIF
“Cancel”**

5. Mengisi seluruh isian form
insert data user

6. Klik button “cancel”

7. Menampilkan halaman user

**SKENARIO ALTERNATIF
“Inputan field Nomor KTP tidak valid”**

6a. Klik button “save”

7a. Menampilkan pesan “KTP sudah
terdaftar !”

**SKENARIO ALTERNATIF
“Belum Mengisi Field”**

7a. Klik button “save”

7b. Menampilkan pesan “Please fill
out this field”

d. *Usecase* Skenario Manajemen Data *User (edit)*

Usecase skenario ini menjelaskan alur untuk proses mengubah data user. Aktor yang yang dapat mengubah data *user* adalah admin. Apabila terdapat user yang mengundurkan diri maka admin dapat memilih status nonaktif. Jika ingin mengaktifkan status user kembali maka harus mengubah status menjadi aktif. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif usecase skenario manajemen data *user (edit)* dapat dilihat pada Tabel 4.6.

Tabel 4.6 Usecase Skenario Manajemen Data User (edit)

| | |
|-----------------------|--------------------------------------|
| Nomor Usecase | USC-02 |
| Nama | Manajemen Data User (edit) |
| Aktor | Admin |
| Pre Condition | Admin akan melakukan edit data user |
| Post Condition | Aktor telah melakukan edit data user |

SKENARIO NORMAL
“Manajemen Data User (input)”

- | | |
|---|--|
| 1. Klik menu “user” | 2. Menampilkan halaman user |
| 3. Klik button “edit” pada salah satu kolom yang ingin diedit | 4. Menampilkan form edit data user |
| 5. Mengubah isian form data user | 7. Mengubah data dalam database |
| 6. Klik button “save” | 8. Menampilkan pesan “Berhasil ubah data user !” |
| | 9. Menampilkan halaman user |

SKENARIO ALTERNATIF
“Cancel”

- | | |
|-----------------------------------|------------------------------|
| 5a. Mengubah isian form data user | 7a. Menampilkan halaman user |
| 6a. Klik button “cancel” | |

SKENARIO ALTERNATIF
“Belum Mengisi Field”

- | | |
|------------------------|--|
| 6b. Klik button “save” | 7b. Menampilkan pesan “Please fill out this field” |
|------------------------|--|

SKENARIO ALTERNATIF
“Inputan tidak valid”

- | | |
|------------------------|--|
| 6c. Klik button “save” | |
|------------------------|--|

7c. Menampilkan pesan “Gagal
ubah data user!”

e. *Usecase* Skenario Manajemen Data Jenis (*view*)

Usecase skenario ini menjelaskan alur untuk proses melihat data jenis. Aktor yang dapat melihat data jenis adalah admin. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario manajemen data jenis (*view*) dapat dilihat pada Tabel 4.7.

Tabel 4.7 Usecase Skenario Manajemen Data Jenis (*view*)

| | |
|-----------------------|---|
| Nomor Usecase | USC-03 |
| Nama | Manajemen Data Jenis (<i>view</i>) |
| Aktor | Admin |
| Pre Condition | Admin harus melakukan login ke dalam system |
| Post Condition | Admin telah melakukan view data jenis |

SKENARIO NORMAL
“Manajemen Data Jenis (*view*)”

1. Klik menu “jenis”

2. Menampilkan halaman jenis

f. *Usecase* Skenario Manajemen Data Jenis (*input*)

Usecase skenario ini menjelaskan alur untuk proses menambah data jenis. Aktor yang dapat menambah data jenis adalah admin. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif usecase skenario manajemen data user (*input*) dapat dilihat pada Tabel 4.8.

Tabel 4.8 Usecase Skenario Manajemen Data Jenis (input)

| | |
|-----------------------|--|
| Nomor Usecase | USC-03 |
| Nama | Manajemen Data Jenis (input) |
| Aktor | Admin |
| Pre Condition | Admin akan melakukan input data jenis |
| Post Condition | Aktor telah melakukan input data jenis |

SKENARIO NORMAL
“Manajemen Data Jenis (input)”

- | | |
|---|---|
| 1. Klik menu “jenis” | 2. Menampilkan halaman jenis |
| 3. Klik button “insert” | 4. Menampilkan form insert data jenis |
| 5. Mengisi seluruh isian form insert data jenis | |
| 6. Klik button “save” | 7. Menyimpan data dalam database |
| | 8. Menampilkan pesan “Berhasil tambah data jenis !” |
| | 9. Menampilkan halaman jenis |

SKENARIO ALTERNATIF
“Cancel”

- | | |
|--|-------------------------------|
| 5a. Mengisi seluruh isian form insert data jenis | |
| 6a. Klik button “cancel” | 7a. Menampilkan halaman jenis |

SKENARIO ALTERNATIF
“Inputan tidak valid”

- | | |
|------------------------|---|
| 6b. Klik button “save” | 7b. Menampilkan pesan “Gagal tambah data jenis !” |
|------------------------|---|

| SKENARIO ALTERNATIF “Belum Mengisi Field” | |
|--|--|
| 6c. Klik button “save” | |
| | 7c. Menampilkan pesan “Please fill out this field” |

g. *Usecase* Skenario Manajemen Data Jenis (*edit*)

Usecase skenario ini menjelaskan alur untuk proses mengubah data jenis. Aktor yang dapat mengubah data jenis adalah admin. Apabila jenis barang tidak tersedia maka admin dapat memilih status *not available*. Jika jenis barang tersedia kembali maka harus mengubah status menjadi *available*. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif usecase skenario manajemen data *supplier (edit)* dapat dilihat pada Tabel 4.9.

Tabel 4.9 Usecase Skenario Manajemen Data Jenis (*edit*)

| | |
|-----------------------|---------------------------------------|
| Nomor Usecase | USC-03 |
| Nama | Manajemen Data Jenis (<i>edit</i>) |
| Aktor | Admin |
| Pre Condition | Admin akan melakukan edit data jenis |
| Post Condition | Aktor telah melakukan edit data jenis |

| SKENARIO NORMAL “Manajemen Data Jenis (input)” | |
|--|-------------------------------------|
| 1. Klik menu “jenis” | |
| | 2. Menampilkan halaman jenis |
| 3. Klik button “jenis” pada salah satu kolom yang ingin diedit | |
| | 4. Menampilkan form edit data jenis |
| 5. Mengubah isian form data jenis | |
| 6. Klik button “save” | |
| | 7. Mengubah data dalam database |

| | |
|---|--|
| | 8. Menampilkan pesan “Berhasil ubah data jenis !” |
| | 9. Menampilkan halaman jenis |
| SKENARIO ALTERNATIF “Cancel” | |
| 5a. Mengubah isian form data jenis | |
| 6a. Klik button “cancel” | |
| | 7a. Menampilkan halaman jenis |
| SKENARIO ALTERNATIF “Inputan tidak valid” | |
| 6b. Klik button “save” | |
| | 7b. Menampilkan pesan “Gagal ubah data jenis !” |
| SKENARIO ALTERNATIF “Belum Mengisi Field” | |
| 6c. Klik button “save” | |
| | 7c. Menampilkan pesan “Please fill out this field” |

h. *Usecase* Skenario Manajemen Data *Supplier* (*view*)

Usecase skenario ini menjelaskan alur untuk proses melihat data *supplier*. Aktor yang dapat melihat data *supplier* adalah admin. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal *usecase* skenario manajemen data *supplier* (*view*) dapat dilihat pada Tabel 4.10.

Tabel 4.10 *Usecase* Skenario Manajemen Data *Supplier* (*view*)

| | |
|-----------------------|--|
| Nomor Usecase | USC-04 |
| Nama | Manajemen Data <i>Supplier</i> (<i>view</i>) |
| Aktor | Admin |
| Pre Condition | Admin harus melakukan login ke dalam system |
| Post Condition | Admin telah melakukan <i>view</i> data <i>supplier</i> |

SKENARIO NORMAL
“Manajemen Data Supplier (view)”

- | | |
|-------------------------|---------------------------------|
| 1. Klik menu “supplier” | 2. Menampilkan halaman supplier |
|-------------------------|---------------------------------|

i. *Usecase* Skenario Manajemen Data *Supplier (input)*

Usecase skenario ini menjelaskan alur untuk proses menambah data *supplier*. Aktor yang yang dapat menambah data jenis adalah *supplier*. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif usecase skenario manajemen data *supplier (input)* dapat dilihat pada Tabel 4.11.

Tabel 4.11 Usecase Skenario Manajemen Data Supplier (input)

| | |
|-----------------------|---|
| Nomor Usecase | USC-04 |
| Nama | Manajemen Data Supplier (input) |
| Aktor | Admin |
| Pre Condition | Admin akan melakukan input data supplier |
| Post Condition | Aktor telah melakukan input data supplier |

SKENARIO NORMAL
“Manajemen Data Supplier (input)”

- | | |
|--|--|
| 1. Klik menu “supplier” | 2. Menampilkan halaman supplier |
| 3. Klik button “insert” | 4. Menampilkan form insert data supplier |
| 5. Mengisi seluruh isian form insert data supplier | 7. Menyimpan data dalam database |
| 6. Klik button “save” | 8. Menampilkan pesan “Berhasil tambah data supplier !” |
| | 9. Menampilkan halaman supplier |

| | |
|--|---|
| SKENARIO ALTERNATIF “Cancel” | |
| 5a. Mengisi seluruh isian form insert data supplier | |
| 6a. Klik button “cancel” | 7a. Menampilkan halaman supplier |
| SKENARIO ALTERNATIF “Inputan tidak valid” | |
| 6b. Klik button “save” | 7b. Menampilkan pesan “Gagal tambah data supplier !” |
| SKENARIO ALTERNATIF “Belum Mengisi Field” | |
| 6c. Klik button “save” | 7c. Menampilkan pesan “Please fill out this field” |

j. *Usecase* Skenario Manajemen Data *Supplier* (*edit*)

Usecase skenario ini menjelaskan alur untuk proses mengubah data *supplier*. Aktor yang dapat mengubah data *supplier* adalah admin. Apabila tidak melakukan pembelian kepada *supplier* tertentu maka admin dapat memilih status nonaktif. Jika melakukan pembelian kembali kepada *supplier* tersebut maka harus mengubah status menjadi aktif. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif usecase skenario manajemen data *supplier* (*edit*) dapat dilihat pada Tabel 4.12.

Tabel 4.12 Usecase Skenario Manajemen Data Supplier (edit)

| | |
|-----------------------|--|
| Nomor Usecase | USC-04 |
| Nama | Manajemen Data Supplier (edit) |
| Aktor | Admin |
| Pre Condition | Admin akan melakukan edit data supplier |
| Post Condition | Aktor telah melakukan edit data supplier |

SKENARIO NORMAL
“Manajemen Data Supplier (input)”

- | | |
|---|--|
| 1. Klik menu “supplier” | 2. Menampilkan halaman supplier |
| 3. Klik button “supplier” pada salah satu kolom yang ingin diedit | 4. Menampilkan form edit data supplier |
| 5. Mengubah isian form data supplier | 7. Mengubah data dalam database |
| 6. Klik button “save” | 8. Menampilkan pesan “Berhasil ubah data supplier !” |
| | 9. Menampilkan halaman supplier |

SKENARIO ALTERNATIF
“Cancel”

- | | |
|---------------------------------------|----------------------------------|
| 5a. Mengubah isian form data supplier | 7a. Menampilkan halaman supplier |
| 6a. Klik button “cancel” | |

SKENARIO ALTERNATIF
“Inputan tidak valid”

- | | |
|------------------------|---|
| 6b. Klik button “save” | 7b. Menampilkan pesan “Gagal ubah data supplier!” |
|------------------------|---|

SKENARIO ALTERNATIF
“Belum Mengisi Field”

6c. Klik button “save”

7c. Menampilkan pesan “Please fill out this field”

k. *Usecase* Skenario Manajemen Data Barang (*view*)

Usecase skenario ini menjelaskan alur untuk proses melihat data barang. Aktor yang dapat melihat data barang adalah admin. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario manajemen data barang (*view*) dapat dilihat pada Tabel 4.13.

Tabel 4.13 Usecase Skenario Manajemen Data Barang (*view*)

| | |
|-----------------------|---|
| Nomor Usecase | USC-05 |
| Nama | Manajemen Data Barang (<i>view</i>) |
| Aktor | Admin |
| Pre Condition | Admin harus melakukan login ke dalam system |
| Post Condition | Admin telah melakukan view data barang |

SKENARIO NORMAL
“Manajemen Data Supplier (*view*)”

1. Klik menu “barang”

2. Menampilkan halaman barang

1. *Usecase* Skenario Manajemen Data Barang (*input*)

Usecase skenario ini menjelaskan alur untuk proses menambah data barang. Aktor yang dapat menambah data barang adalah admin. Terdapat *field* biaya pemesanan, biaya persediaan, waktu tunggu dalam *form* insert barang yang digunakan sebagai parameter perhitungan menggunakan metode EOQ. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif usecase skenario manajemen data barang (*input*) dapat dilihat pada Tabel 4.14.

Tabel 4.14 Usecase Skenario Manajemen Data Barang (input)

| | |
|-----------------------|---|
| Nomor Usecase | USC-05 |
| Nama | Manajemen Data Barang (input) |
| Aktor | Admin |
| Pre Condition | Admin akan melakukan input data barang |
| Post Condition | Aktor telah melakukan input data barang |

SKENARIO NORMAL
“Manajemen Data Barang (input)”

- | | |
|--|--|
| 1. Klik menu “barang” | 2. Menampilkan halaman barang |
| 3. Klik button “insert” | 4. Menampilkan form insert data barang |
| 5. Mengisi seluruh isian form insert data barang | |
| 6. Klik button “save” | 7. Menyimpan data dalam database |
| | 8. Menampilkan pesan “Berhasil tambah data barang !” |
| | 9. Menampilkan halaman barang |

SKENARIO ALTERNATIF
“Cancel”

- | | |
|---|--------------------------------|
| 5a. Mengisi seluruh isian form insert data barang | |
| 6a. Klik button “cancel” | 7a. Menampilkan halaman barang |

SKENARIO ALTERNATIF
“Inputan tidak valid”

- | | |
|------------------------|--|
| 6b. Klik button “save” | 7b. Menampilkan pesan “Gagal tambah data barang !” |
|------------------------|--|

**SKENARIO ALTERNATIF
“Belum Mengisi Field”**

6c. Klik button “save”

7c. Menampilkan pesan “Please fill out this field”

m. *Usecase* Skenario Manajemen Data Barang (*edit*)

Usecase skenario ini menjelaskan alur untuk proses mengubah data barang. Aktor yang dapat mengubah data barang adalah admin. . Apabila barang tidak tersedia maka admin dapat memilih status *not available*. Jika barang tersedia atau melakukan stok barang kembali maka harus mengubah status menjadi *available*. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif usecase skenario manajemen data barang (*view*) dapat dilihat pada Tabel 4.15.

Tabel 4.15 Usecase Skenario Manajemen Data Barang (edit)

| | |
|-----------------------|--|
| Nomor Usecase | USC-05 |
| Nama | Manajemen Data Barang (edit) |
| Aktor | Admin |
| Pre Condition | Admin akan melakukan edit data barang |
| Post Condition | Aktor telah melakukan edit data barang |

**SKENARIO NORMAL
“Manajemen Data Barang (input)”**

1. Klik menu “barang”

2. Menampilkan halaman barang

3. Klik button “barang” pada salah satu kolom yang ingin diedit

4. Menampilkan form edit data barang

5. Mengubah isian form data barang

6. Klik button “save”

7. Mengubah data dalam database

| | |
|--|--|
| | 8. Menampilkan pesan “Berhasil ubah data barang !” |
| | 9. Menampilkan halaman barang |
| SKENARIO ALTERNATIF “Cancel” | |
| 5a. Mengubah isian form data barang | |
| 6a. Klik button “cancel” | |
| | 7a. Menampilkan halaman barang |
| SKENARIO ALTERNATIF “Inputan tidak valid” | |
| 6b. Klik button “save” | |
| | 7b. Menampilkan pesan “Gagal ubah data barang!” |
| SKENARIO ALTERNATIF “Belum Mengisi Field” | |
| 6c. Klik button “save” | |
| | 7c. Menampilkan pesan “Please fill out this field” |

n. *Usecase* Skenario Manajemen Data Stok (*view*)

Usecase skenario ini menjelaskan alur untuk proses melihat data stok masuk. Aktor yang dapat melihat data barang adalah admin. Ketika admin mengklik tombol detail maka akan menampilkan data detail stok masuk sesuai barang dan tanggal kadaluarsa yang pilih. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario manajemen data stok (*view*) dapat dilihat pada Tabel 4.16.

Tabel 4.16 Usecase Skenario Manajemen Data Stok (*view*)

| | |
|-----------------------|---|
| Nomor Usecase | USC-06 |
| Nama | Manajemen Data Stok (<i>view</i>) |
| Aktor | Admin |
| Pre Condition | Admin harus melakukan login ke dalam system |
| Post Condition | Admin telah melakukan view data stok |

| SKENARIO NORMAL | |
|--|---|
| “Manajemen Data Stok (view)” | |
| 1. Klik menu “persediaan barang” | |
| | 2. Menampilkan halaman persediaan barang |
| 3. Klik button “detail” pada salah satu kolom yang ingin dilihat detail persediaan barang | |
| | 4. Menampilkan data detail persediaan barang |
| 5. Klik button “detail” pada salah satu kolom yang ingin dilihat detail transaksi barang masuk | |
| | 6. Menampilkan data detail transaksi barang masuk |

o. *Usecase* Skenario Manajemen Data Stok (*input*)

Usecase skenario ini menjelaskan alur untuk proses menambah data barang. Aktor yang dapat menambah data stok masuk atau pembelian pada masing-masing barang adalah admin. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen data barang (*input*) dapat dilihat pada Tabel 4.17.

Tabel 4.17 *Usecase* Skenario Manajemen Data Stok (*input*)

| | |
|-----------------------|---------------------------------------|
| Nomor Usecase | USC-06 |
| Nama | Manajemen Data Stok (<i>input</i>) |
| Aktor | Admin |
| Pre Condition | Admin akan melakukan input data stok |
| Post Condition | Aktor telah melakukan input data stok |

SKENARIO NORMAL
“Manajemen Data Stok (input)”

| | |
|---|---|
| 1. Klik menu “persediaan barang” | 2. Menampilkan halaman persediaan barang |
| 3. Klik button “insert” | 4. Menampilkan form insert data stok masuk |
| 5. Pilih barang, input tanggal kadaluarsa barang yang dibeli, dan jumlah stok masuk | |
| 6. Klik button “add” | 7. Menampilkan list data transaksi stok masuk |
| 8. Klik button “simpan” | 9. Menyimpan data dalam database |
| | 10. Menampilkan pesan “Pembelian SUKSES!” |
| | 11. Menampilkan halaman persediaan barang |

SKENARIO ALTERNATIF
“Delete”

| | |
|--|---|
| 8a. Klik button “delete” pada list transaksi stok yang ingin dihapus | 9a. Menghapus data yang dipilih pada list data transaksi stok masuk |
| 10a. Klik button “simpan” | 11a. Menyimpan data dalam database |

SKENARIO ALTERNATIF
“Belum Mengisi Field”

| | |
|-----------------------|--|
| 6a. Klik button “add” | 7a. Menampilkan alert “Data tidak lengkap !” |
| 8a. Klik OK | |

| | |
|---|---|
| | 9a. Menampilkan form insert data stok masuk |
| SKENARIO ALTERNATIF “Inputan field jumlah stok masuk tidak valid ” | |
| 6b. Klik button “add” | |
| | 7b. Menampilkan alert “Jumlah stok masuk tidak boleh kosong!” |
| 8b. Klik OK | |
| | 9b. Menampilkan form insert data stok masuk |

p. *Usecase* Skenario Melihat Data Stok

Usecase skenario ini menjelaskan alur untuk proses melihat data stok barang. Aktor yang dapat melihat data stok barang yang dimiliki perusahaan adalah kasir. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario melihat data stok dapat dilihat pada Tabel 4.18.

Tabel 4.18 Usecase Skenario Melihat Data Stok

| | |
|-----------------------|---|
| Nomor Usecase | USC-07 |
| Nama | Melihat Data Stok |
| Aktor | Kasir |
| Pre Condition | Kasir harus melakukan login ke dalam system |
| Post Condition | Kasir telah melakukan view data stok |

| | |
|--|---|
| SKENARIO NORMAL “Melihat Data Stok” | |
| 1. Klik menu “dashboard” | |
| | 2. Menampilkan halaman data stok barang |

q. *Usecase* Skenario Manajemen Data Retur Barang (*view*)

Usecase skenario ini menjelaskan alur untuk proses melihat data retur barang. Aktor yang dapat melihat data retur barang adalah admin. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario manajemen data retur barang (*view*) dapat dilihat pada Tabel 4.19.

Tabel 4.19 Usecase Skenario Manajemen Data Retur Barang (*view*)

| | |
|-----------------------|--|
| Nomor Usecase | USC-08 |
| Nama | Manajemen Data Retur Barang (<i>view</i>) |
| Aktor | Admin |
| Pre Condition | Admin harus melakukan login ke dalam system |
| Post Condition | Admin telah melakukan view data retur barang |

SKENARIO NORMAL
“Manajemen Data Retur Barang (*view*)”

1. Klik menu “retur barang”

2. Menampilkan halaman retur barang

r. *Usecase* Skenario Manajemen Data Retur Barang (*input*)

Usecase skenario ini menjelaskan alur untuk proses menambah data retur barang. Aktor yang dapat menambah data retur barang atau stok keluar pada masing-masing barang adalah admin. Terdapat *field* keterangan digunakan untuk mengetahui kondisi barang yang akan dilakukan retur yaitu “kadaluarsa” atau “rusak atau cacat”. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif usecase skenario manajemen data barang (*input*) dapat dilihat pada Tabel 4.20.

Tabel 4.20 Usecase Skenario Manajemen Data Retur Barang (input)

| | |
|-----------------------|---|
| Nomor Usecase | USC-08 |
| Nama | Manajemen Data Retur Barang (input) |
| Aktor | Admin |
| Pre Condition | Admin akan melakukan input data retur barang |
| Post Condition | Aktor telah melakukan input data retur barang |

SKENARIO NORMAL
“Manajemen Data Retur Barang (input)”

| | |
|---|---|
| 1. Klik menu “retur barang” | 2. Menampilkan halaman retur barang |
| 3. Klik button “insert” | 4. Menampilkan form insert data retur barang |
| 5. Pilih barang, tanggal kadaluarsa barang, keterangan dan input jumlah stok keluar | |
| 6. Klik button “add” | 7. Menampilkan list data transaksi retur barang |
| 8. Klik button “simpan” | 9. Menyimpan data dalam database |
| | 10. Menampilkan pesan “Retur Barang SUKSES!” |
| | 11. Menampilkan halaman retur barang |

SKENARIO ALTERNATIF
“Delete”

| | |
|--|---|
| 8a. Klik button “delete” pada list transaksi retur barang yang ingin dihapus | 9a. Menghapus data yang dipilih pada list data transaksi retur barang |
| 10a. Klik button “simpan” | |

| | |
|--|--|
| | 11a. Menyimpan data dalam database |
| SKENARIO ALTERNATIF “Belum Mengisi Field” | |
| 6a. Klik button “add” | |
| | 7a. Menampilkan alert "Data tidak lengkap !" |
| 8a. Klik OK | |
| | 9a. Menampilkan form insert data retur barang |
| SKENARIO ALTERNATIF “Inputan field jumlah stok keluar tidak valid ” | |
| 6b. Klik button “add” | |
| | 7b. Menampilkan alert “Jumlah stok keluar tidak boleh kosong!” |
| 8b. Klik OK | |
| | 9b. Menampilkan form insert data retur barang |

s. *Usecase* Skenario Menambah Data Penjualan

Usecase skenario ini menjelaskan alur untuk proses menambah data retur barang. Aktor yang dapat menambah data penjualan pada masing-masing barang adalah kasir. Terdapat *field* jumlah stok keluar yang digunakan sebagai parameter perhitungan metode *Double Exponential Smoothing*. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif usecase skenario menambah data penjualan dapat dilihat pada Tabel 4.21.

Tabel 4.21 Usecase Skenario Menambah Data Penjualan

| | |
|-----------------------|---|
| Nomor Usecase | USC-09 |
| Nama | Menambah Data Penjualan |
| Aktor | Kasir |
| Pre Condition | Kasir akan melakukan tambah data penjualan |
| Post Condition | Kasir telah melakukan tambah data penjualan |

SKENARIO NORMAL
“Menambah Data Penjualan”

- | | |
|----------------------------------|--|
| 1. Klik menu “penjualan” | 2. Menampilkan form transaksi penjualan |
| 3. Pilih barang dan input jumlah | |
| 4. Klik button “add” | 5. Menampilkan list data transaksi penjualan |
| 6. Klik button “simpan” | 7. Menyimpan data dalam database |
| | 8. Menampilkan pesan “Penjualan SUKSES!” |
| | 9. Menampilkan form transaksi penjualan |

SKENARIO ALTERNATIF
“Delete”

- | | |
|---|--|
| 6a. Klik button “delete” pada list transaksi penjualan yang ingin dihapus | 7a. Menghapus data yang dipilih pada list data transaksi penjualan |
| 8a. Klik button “simpan” | 9a. Menyimpan data dalam database |

SKENARIO ALTERNATIF
“Belum Mengisi Field”

- | | |
|-----------------------|--|
| 4a. Klik button “add” | 5a. Menampilkan alert “Pilih Barang!” |
| 6a. Klik OK | 7a. Menampilkan form transaksi penjualan |

SKENARIO ALTERNATIF
“Inputan field jumlah stok keluar tidak valid ”

- | |
|-----------------------|
| 4b. Klik button “add” |
|-----------------------|

| | |
|-------------|--|
| | 5b. Menampilkan alert “Jumlah stok keluar tidak boleh kosong!” |
| 6b. Klik OK | |
| | 7b. Menampilkan form transaksi penjualan |

t. *Usecase* Skenario Melihat Laporan Data Penjualan

Usecase skenario ini menjelaskan alur untuk proses melihat data laporan penjualan. Aktor yang dapat melihat data laporan penjualan sesuai barang dan interval laporan adalah admin. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario melihat laporan penjualan dapat dilihat pada Tabel 4.22.

Tabel 4.22 Usecase Skenario Melihat Laporan Penjualan

| | |
|-----------------------|--|
| Nomor Usecase | USC-10 |
| Nama | Melihat Laporan Data Penjualan |
| Aktor | Admin |
| Pre Condition | Admin akan melakukan lihat data laporan penjualan |
| Post Condition | Admin telah melakukan lihat data laporan penjualan |

| SKENARIO NORMAL “Melihat Laporan Penjualan” | |
|--|---|
| 1. Klik menu “laporan penjualan” | |
| | 2. Menampilkan halaman laporan penjualan |
| 3. Pilih barang dan input interval laporan | |
| 4. Klik button “view” | |
| | 5. Menampilkan laporan sesuai barang dan interval laporan |

u. *Usecase* Skenario Melihat Hasil Prediksi

Usecase skenario ini menjelaskan alur untuk melihat proses menghitung hasil prediksi data persediaan menggunakan metode *Double Exponential Smoothing* dan

Economy Order Quantity pada masing-masing barang. Aktor yang dapat melihat proses perhitungan hasil prediksi adalah admin. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario melihat hasil prediksi dapat dilihat pada Tabel 4.23.

Tabel 4.23 Usecase Skenario Melihat Hasil Prediksi

| | |
|-----------------------|--|
| Nomor Usecase | USC-11 |
| Nama | Melihat Hasil Prediksi |
| Aktor | Admin |
| Pre Condition | Admin akan melakukan prediksi persediaan barang |
| Post Condition | Admin telah melakukan prediksi persediaan barang |

**SKENARIO NORMAL
“Melihat Hasil Prediksi”**

- | | |
|-----------------------------------|---|
| 1. Klik menu “prediksi” | 2. Menampilkan form prediksi |
| 3. Pilih barang, tahun, dan bulan | 4. Menampilkan form insert data retur barang |
| 5. Klik button “hitung” | 6. Menampilkan tabel perhitungan hasil prediksi |

**SKENARIO ALTERNATIF
“Belum Mengisi Field”**

- | | |
|--------------------------|---|
| 5a. Klik button “hitung” | 6a. Menampilkan alert “Data tidak lengkap!” |
|--------------------------|---|

v. *Usecase Skenario Logout*

Usecase skenario ini menjelaskan alur proses keluar dari sitem. Aktor yang dapat melakukan *logout* adalah admin dan kasir. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario *logout* dilihat pada tabel 4.24

Tabel 4.24 Usecase Skenario Logout

| | |
|-----------------------|---|
| Nomor Usecase | USC-12 |
| Nama | Logout |
| Aktor | Admin dan Kasir |
| Pre Condition | Admin dan Kasir harus sudah melakukan login dan masuk ke dalam system |
| Post Condition | Admin dan Kasir telah melakukan logout |

**SKENARIO NORMAL
“Logout”**

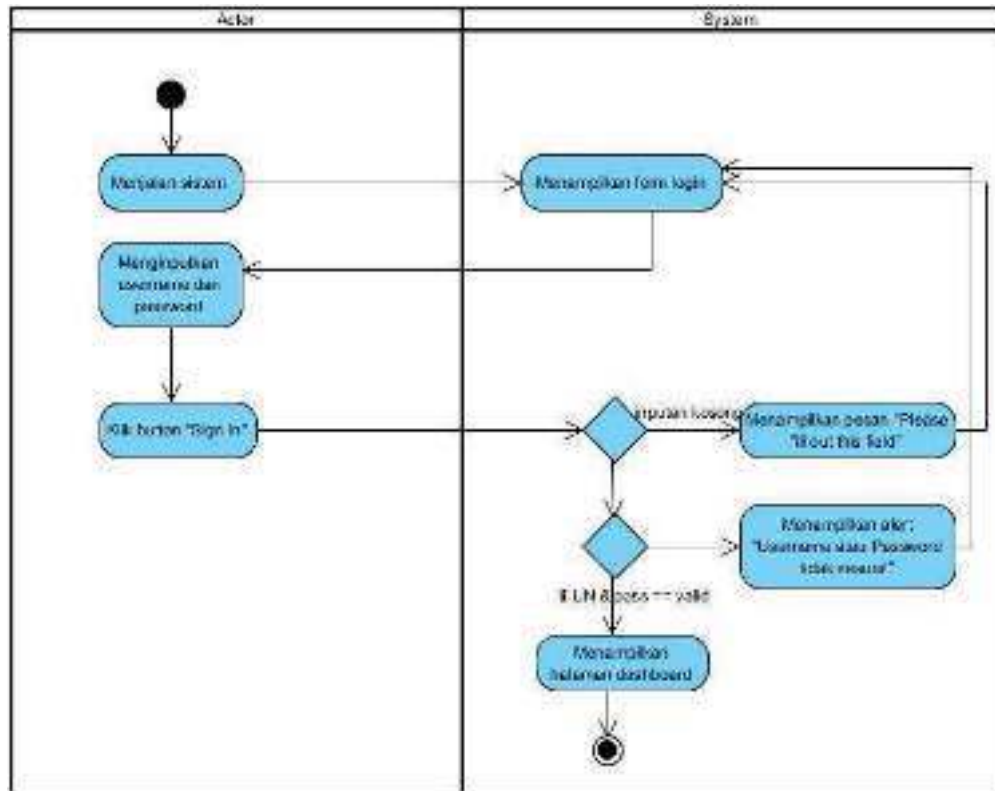
1. Klik button “Logout” dibagian pojok kanan atas halaman
2. Menghapus ijin akses system
3. Menampilkan halaman login

4.3.4 Activity Diagram

Activity diagram merupakan Gambaran dari alur aktivitas dalam sistem yang sedang dirancang, bagaimana alir tersebut berawal, decision yang mungkin terjadi dan bagaimana alur tersebut berakhir. *Activity diagram* menggambarkan aktivitas aktor dan sistem yang salingberhubungan dalam suatu aktivitas atau *event*. *Activity diagram* sistem pengendalian persediaan stok barang adalah sebagai berikut.

a. Activity Diagram Login

Activity diagram ini menggambarkan alur aktivitas dari proses login untuk dapat mengakses fitur sistem. Aktor yang melakukan login adalah admin dan kasir. Setiap *user* harus memiliki username dan password yang terdaftar pada sistem. Hanya user berstatus aktif yang memiliki hak akses terhadap sistem. Detail penjelasan alur aktivitas dari proses *login* dapat dilihat pada Gambar 4.3.



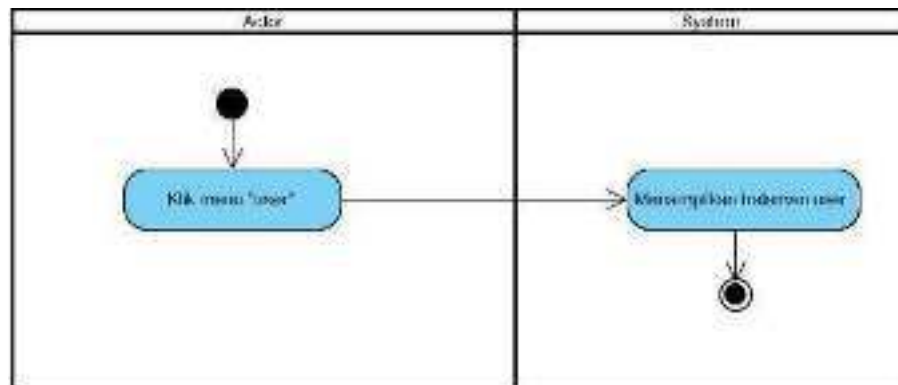
Gambar 4.3 Activity Diagram Login

b. *Activity Diagram Manajemen Data User (view)*

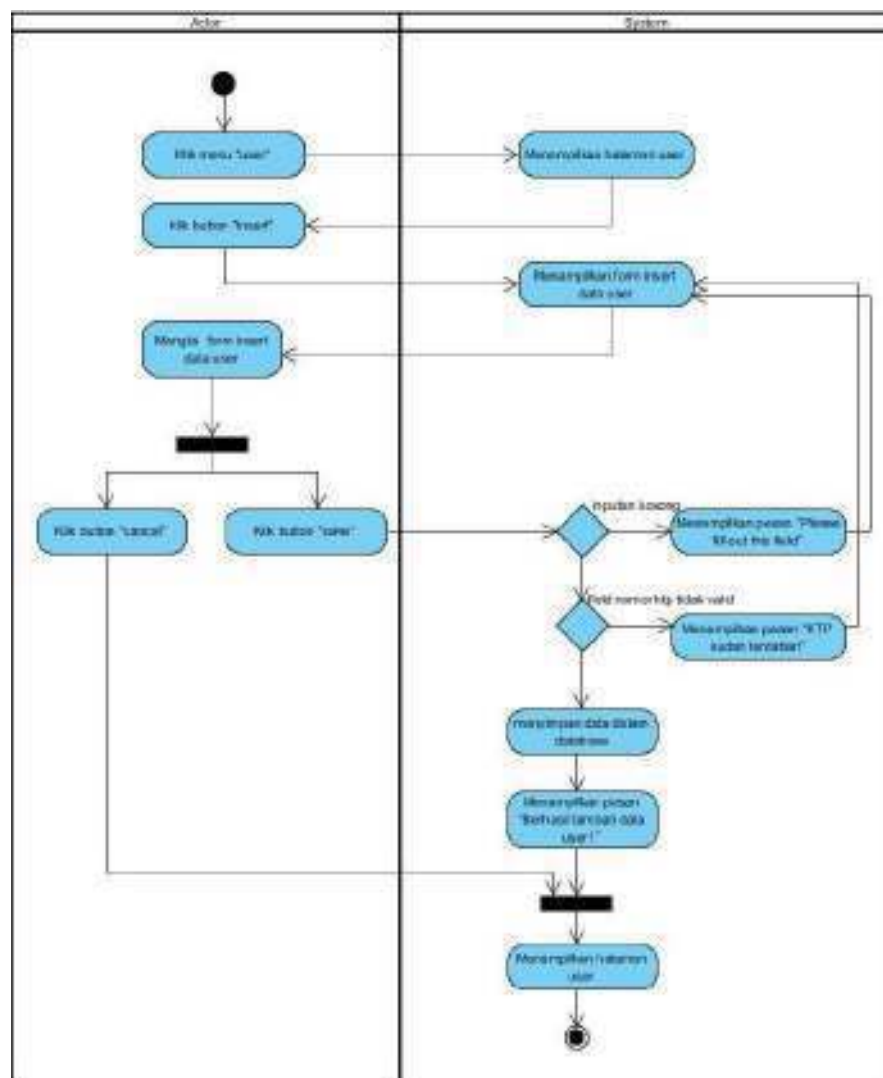
Activity diagram ini menggambarkan alur aktivitas dari proses melihat data user. Aktor yang dapat melihat data user adalah admin. Detail penjelasan alur aktivitas dari proses melihat data user dapat dilihat pada Gambar 4.4.

c. *Activity Diagram Manajemen Data User (input)*

Activity diagram ini menggambarkan alur aktivitas dari proses menambah data user. Aktor yang dapat menambah data *user* adalah admin. Detail penjelasan alur aktivitas dari proses menambah data user dapat dilihat pada Gambar 4.5.



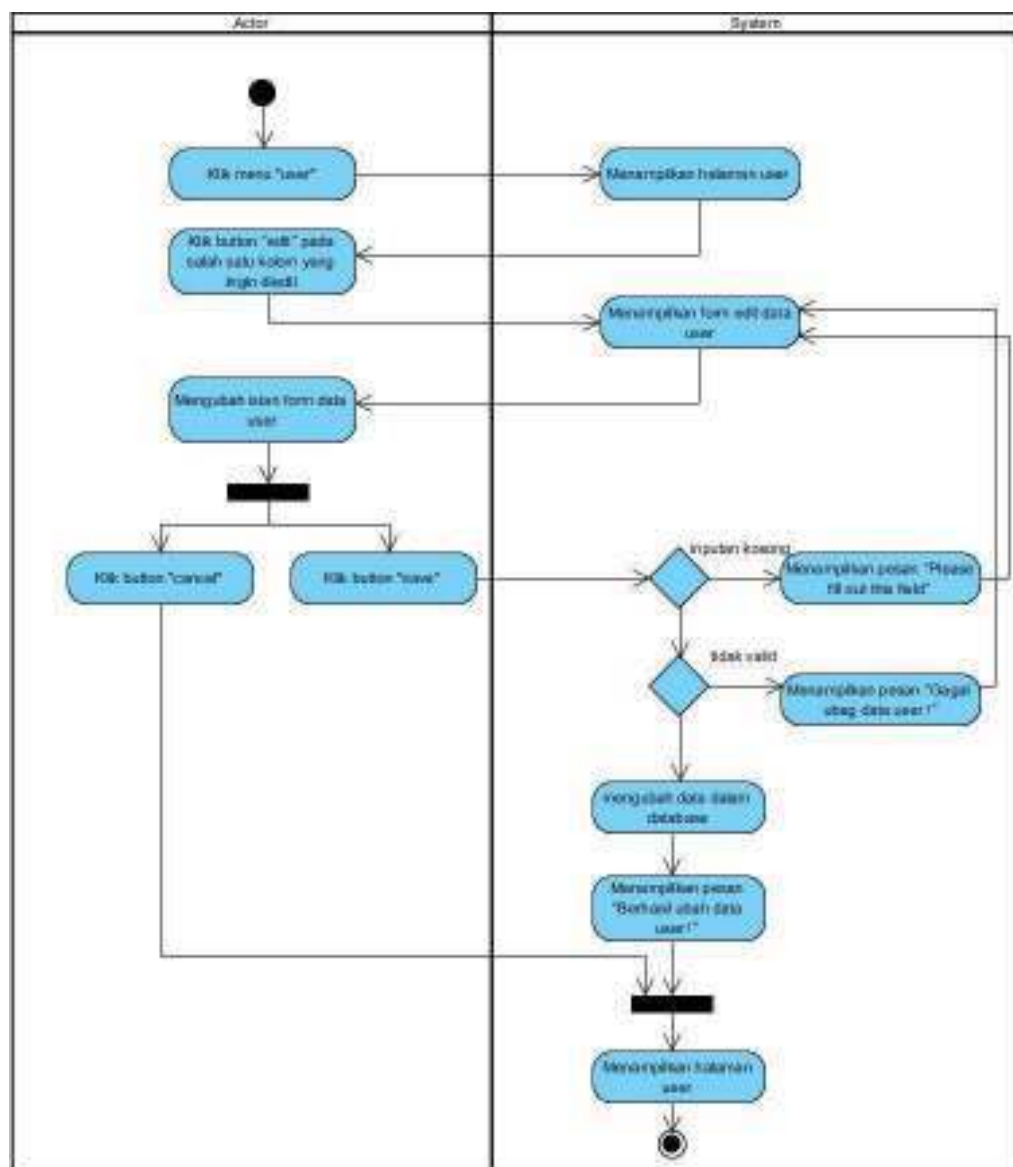
Gambar 4.4 Activity Diagram Manajemen Data User (view)



Gambar 4.5 Activity Diagram Manajemen Data User (input)

d. *Activity Diagram Manajemen Data User (edit)*

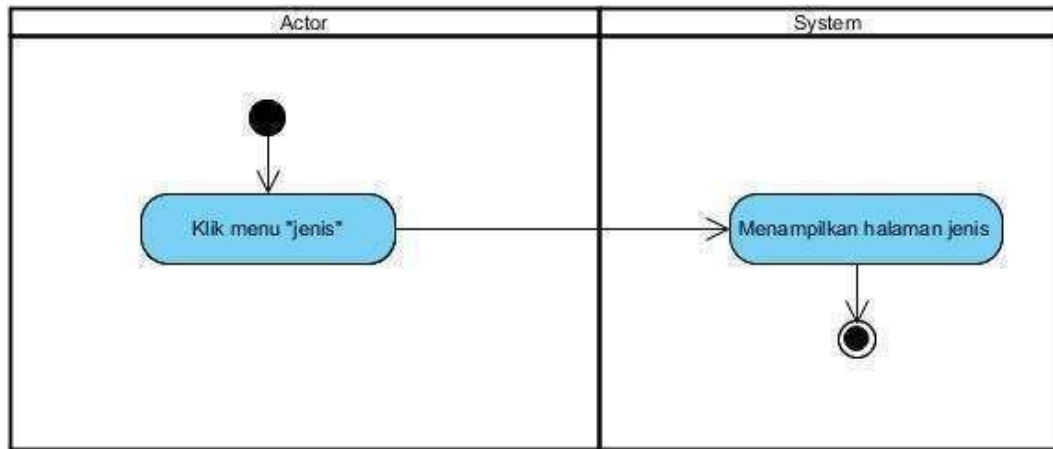
Activity diagram ini menggambarkan alur aktivitas dari proses mengubah data user. Aktor yang dapat mengubah data *user* adalah admin. Apabila terdapat user yang mengundurkan diri maka admin dapat memilih status nonaktif. Jika ingin mengaktifkan status user kembali maka harus mengubah status menjadi aktif. Detail penjelasan alur aktivitas dari proses mengubah data user dapat dilihat pada Gambar 4.6.



Gambar 4.6 Activity Diagram Manajemen Data User (edit)

e. *Activity Diagram Manajemen Data Jenis (view)*

Activity diagram ini menggambarkan alur aktivitas dari proses melihat data jenis. Aktor yang dapat melihat data jenis adalah admin. Detail penjelasan alur aktivitas dari proses melihat data jenis dapat dilihat pada Gambar 4.7.



Gambar 4.7 Activity Diagram Manajemen Data Jenis (view)

f. *Activity Diagram Manajemen Data Jenis (input)*

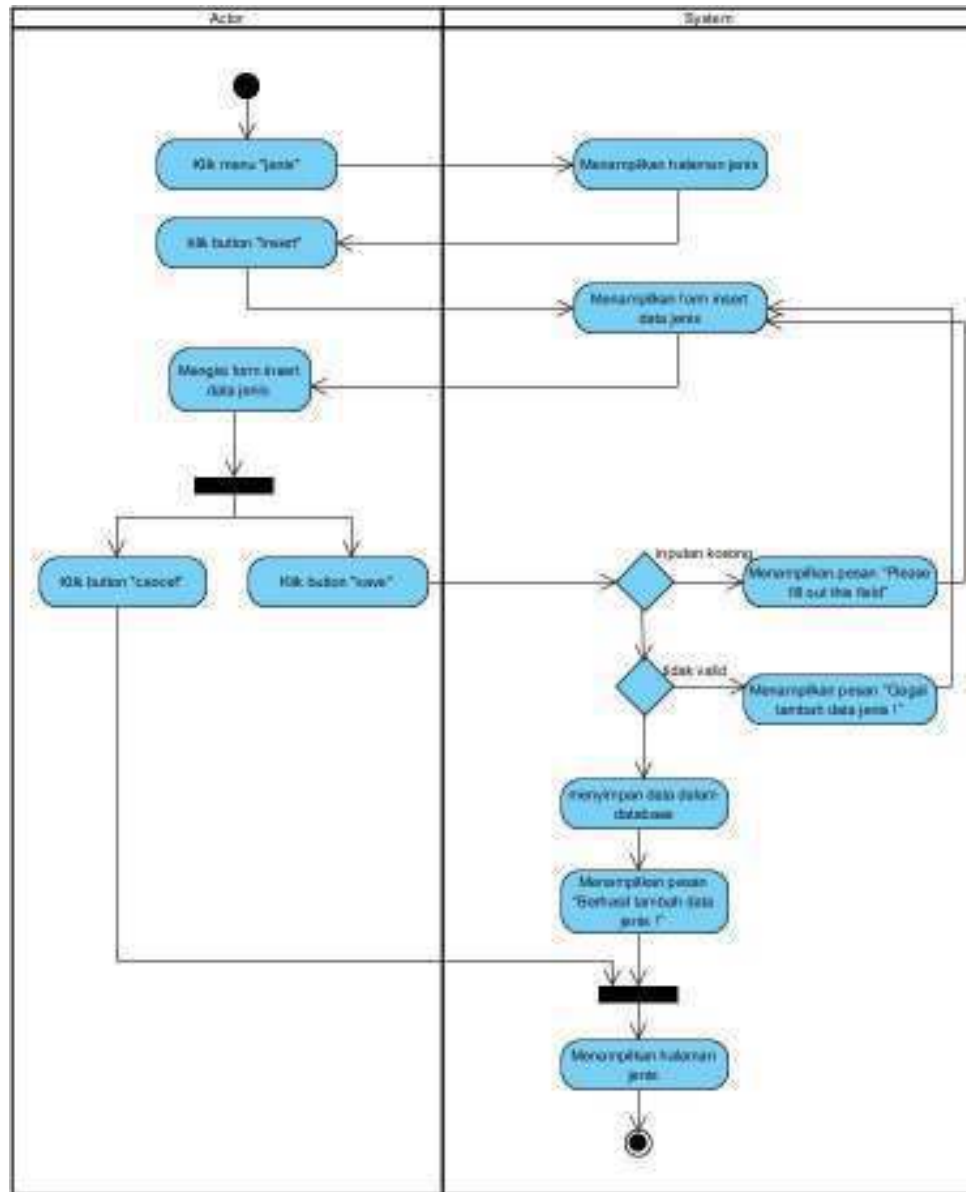
Activity diagram ini menggambarkan alur aktivitas dari proses menambah data jenis. Aktor yang dapat menambah data jenis adalah admin. Detail penjelasan alur aktivitas dari proses menambah data jenis dapat dilihat pada Gambar 4.8.

g. *Activity Diagram Manajemen Data Jenis (edit)*

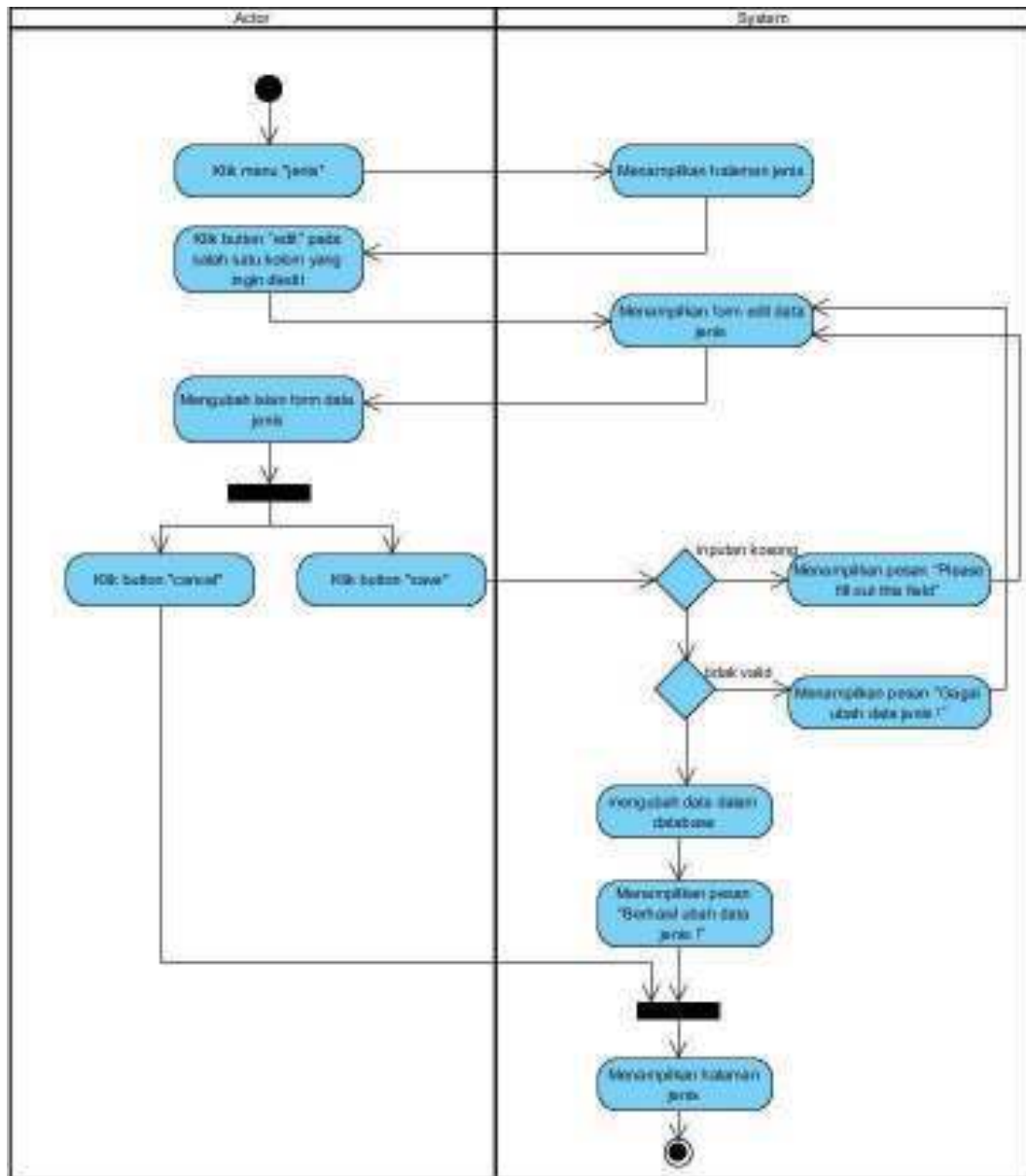
Activity diagram ini menggambarkan alur aktivitas dari proses mengubah data jenis. Aktor yang dapat mengubah data jenis adalah admin. Apabila jenis barang tidak tersedia maka admin dapat memilih status *not available*. Jika jenis barang tersedia kembali maka harus mengubah status menjadi *available*. Detail penjelasan alur aktivitas dari proses mengubah data jenis dapat dilihat pada Gambar 4.9.

h. Activity Diagram Manajemen Data *Supplier* (view)

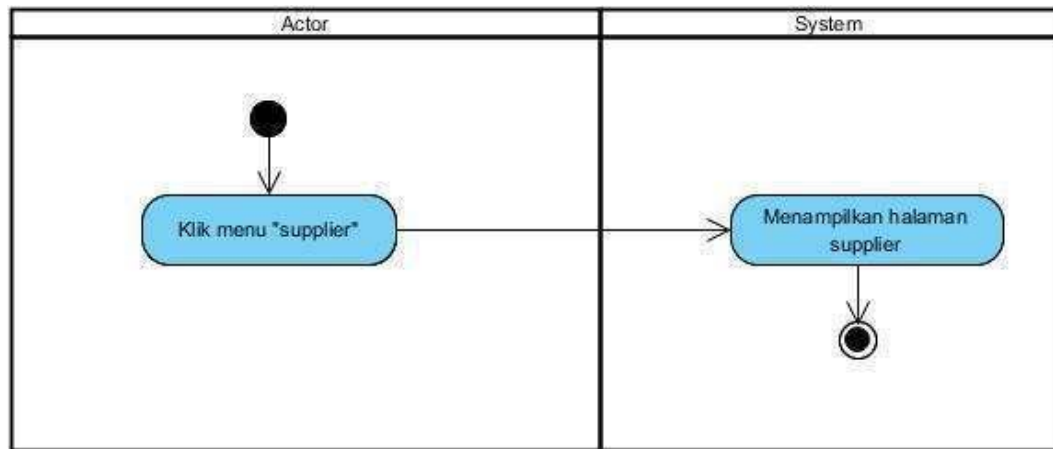
Activity diagram ini menggambarkan alur aktivitas dari proses melihat data *supplier*. Aktor yang dapat melihat data *supplier* adalah admin. Detail penjelasan alur aktivitas dari proses melihat data *supplier* dapat dilihat pada Gambar 4.10.



Gambar 4.8 Activity Diagram Manajemen Data Jenis (input)



Gambar 4.9 Activity Diagram Manajemen Data Jenis (edit)



Gambar 4.10 Activity Diagram Manajemen Data Supplier (view)

i. *Activity Diagram Manajemen Data Supplier (input)*

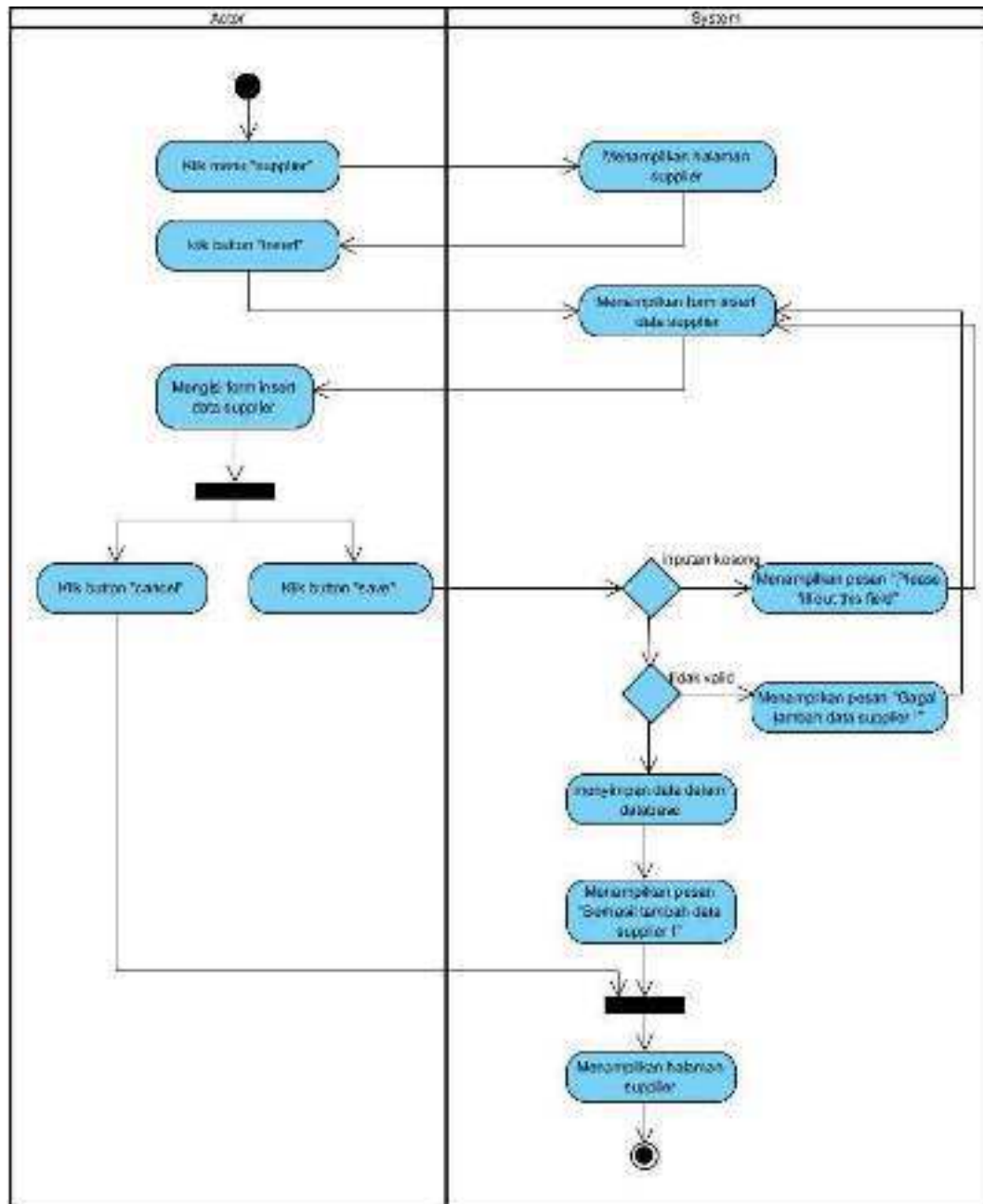
Activity diagram ini menggambarkan alur aktivitas dari proses menambah data *supplier*. Aktor yang yang dapat menambah data jenis adalah *supplier*. Detail penjelasan alur aktivitas dari proses menambah data *supplier* dapat dilihat pada Gambar 4.11.

j. *Activity Diagram Manajemen Data Supplier (edit)*

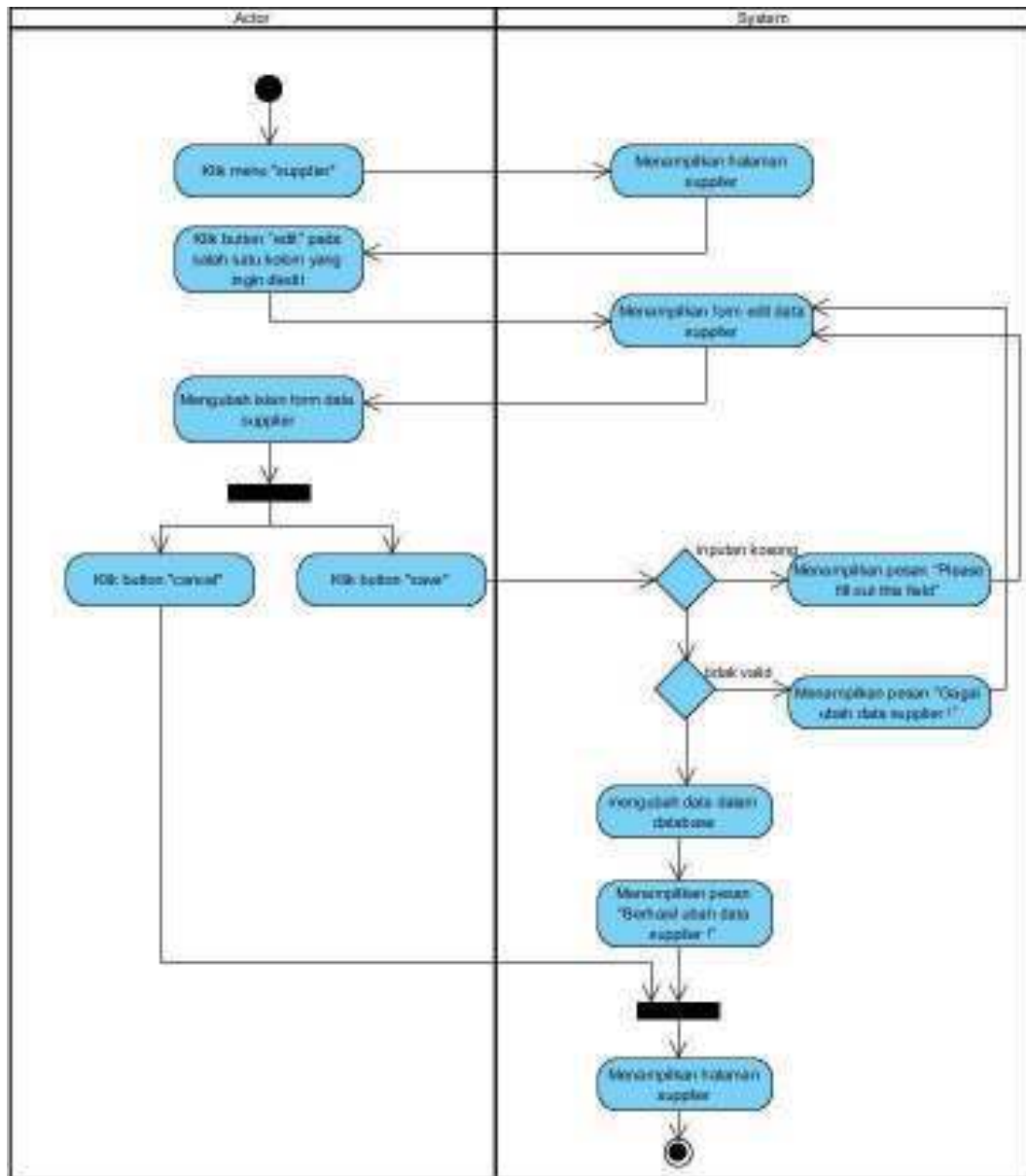
Activity diagram ini menggambarkan alur aktivitas dari proses mengubah data *supplier*. Aktor yang yang dapat mengubah data *supplier* adalah admin. Apabila tidak melakukan pembelian kepada *supplier* tertentu maka admin dapat memilih status nonaktif. Jika melakukan pembelian kembali kepada *supplier* tersebut maka harus mengubah status menjadi aktif. Detail penjelasan alur aktivitas dari proses mengubah data *supplier* dapat dilihat pada Gambar 4.12.

k. *Activity Diagram Manajemen Data Barang (view)*

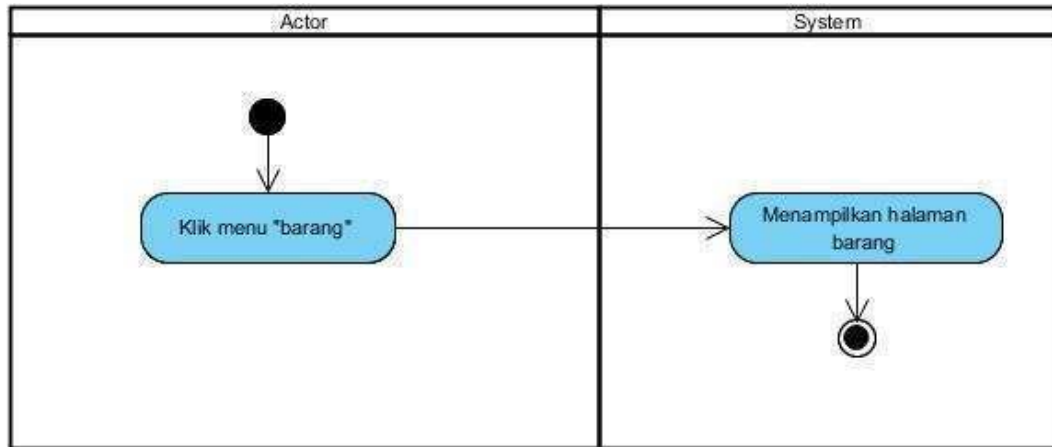
Activity diagram ini menggambarkan alur aktivitas dari proses melihat data barang. Aktor yang dapat melihat data barang adalah admin. Detail penjelasan alur aktivitas dari proses melihat data barang dapat dilihat pada Gambar 4.13.



Gambar 4.11 Activity Diagram Manajemen Data Supplier (input)



Gambar 4.12 Activity Diagram Manajemen Data Supplier (edit)



Gambar 4.13 Activity Diagram Manajemen Data Barang (view)

l. *Activity Diagram Manajemen Data Barang (input)*

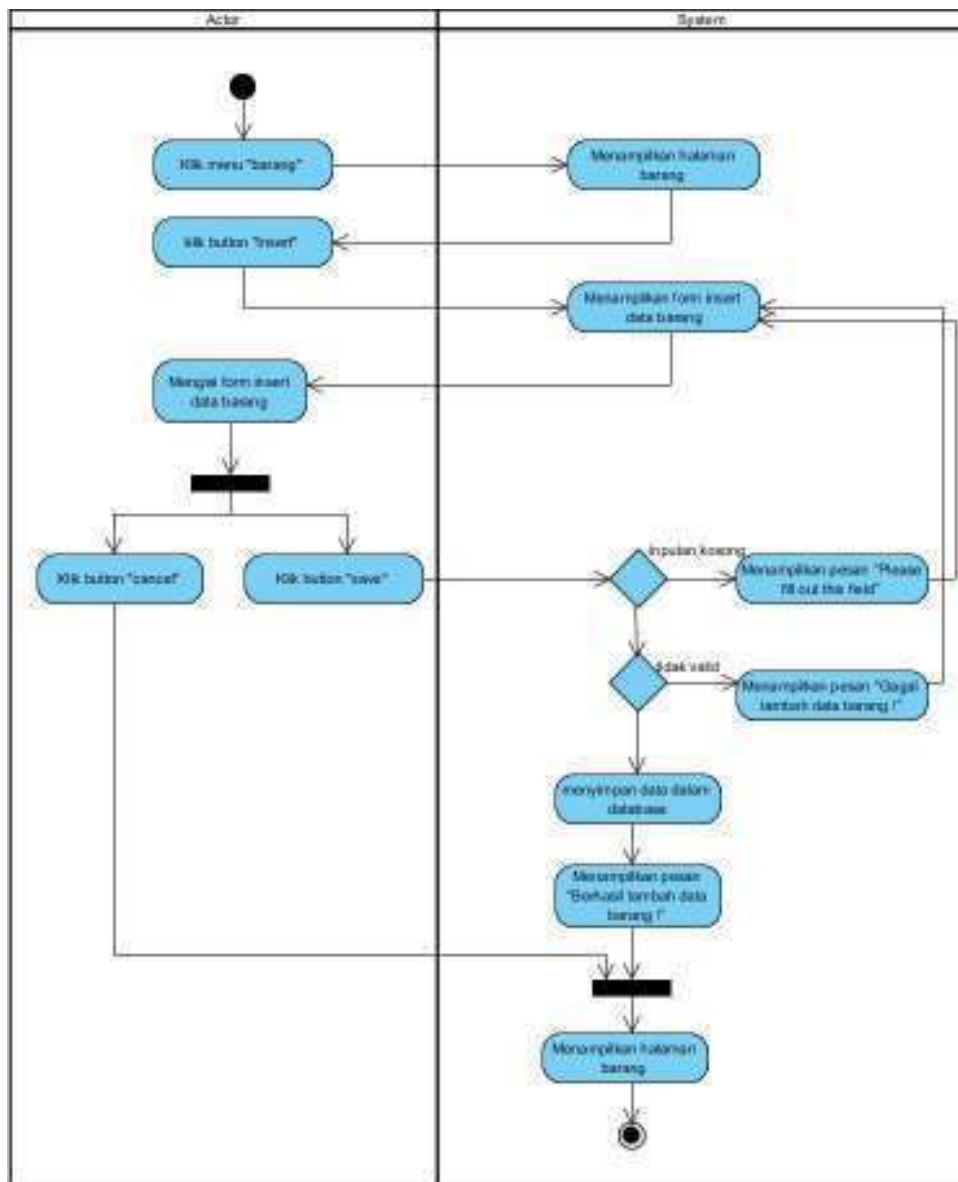
Activity diagram ini menggambarkan alur aktivitas dari proses menambah data barang. Aktor yang dapat menambah data barang adalah admin. Terdapat *field* biaya pemesanan, biaya persediaan, waktu tunggu dalam *form* insert barang yang digunakan sebagai parameter perhitungan menggunakan metode EOQ. Detail penjelasan alur aktivitas dari proses menambah data barang dapat dilihat pada Gambar 4.14.

m. *Activity Diagram Manajemen Data Barang (edit)*

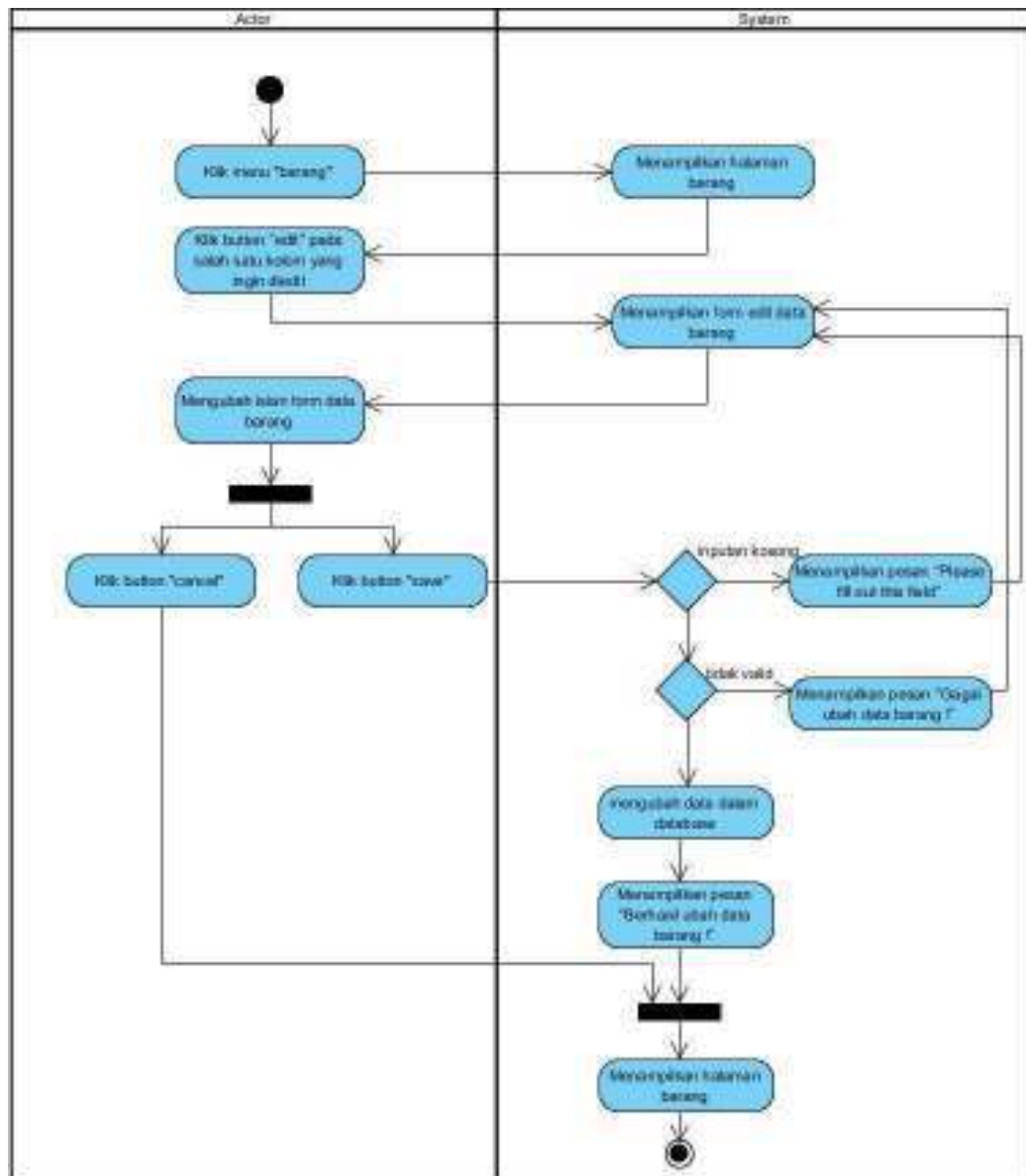
Activity diagram ini menggambarkan alur aktivitas dari proses mengubah data barang. Aktor yang dapat mengubah data barang adalah admin. Apabila barang tidak tersedia maka admin dapat memilih status *not available*. Jika barang tersedia atau melakukan stok barang kembali maka harus mengubah status menjadi *available*. Detail penjelasan alur aktivitas dari proses mengubah data barang dapat dilihat pada Gambar 4.15.

n. *Activity Diagram Manajemen Data Stok (view)*

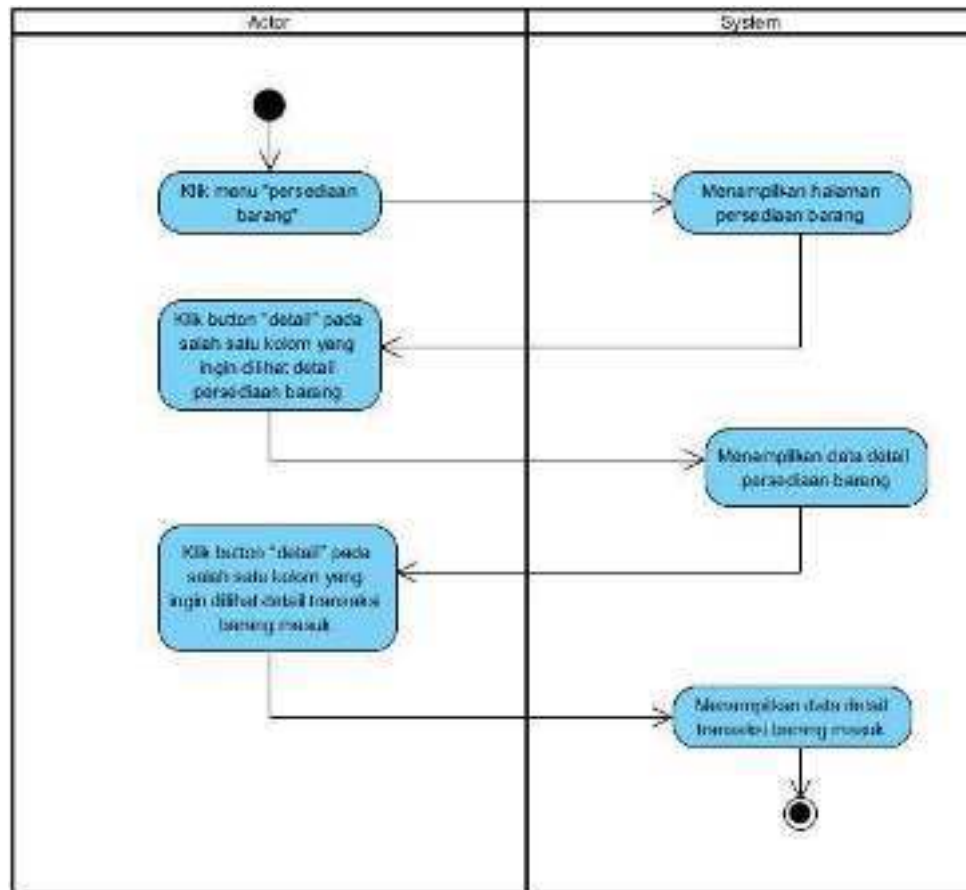
Activity diagram ini menggambarkan alur aktivitas dari proses melihat data stok masuk. Aktor yang dapat melihat data barang adalah admin. Ketika admin mengklik tombol detail maka akan menampilkan data detail stok masuk sesuai barang dan tanggal kadaluarsa yang pilih. Detail penjelasan alur aktivitas dari proses melihat data stok dapat dilihat pada Gambar 4.16.



Gambar 4.14 Activity Diagram Manajemen Data Barang (input)



Gambar 4.15 Activity Diagram Manajemen Data Barang (edit)



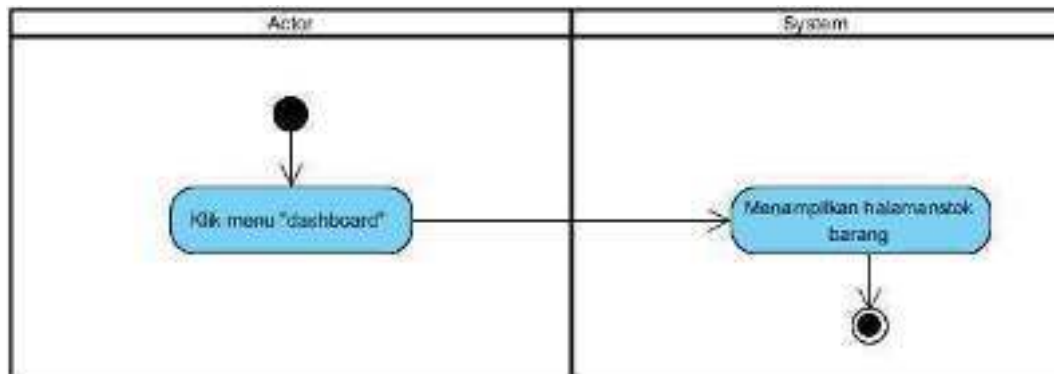
Gambar 4.16 Activity Diagram Manajemen Data Stok (view)

o. *Activity Diagram Manajemen Data Stok (input)*

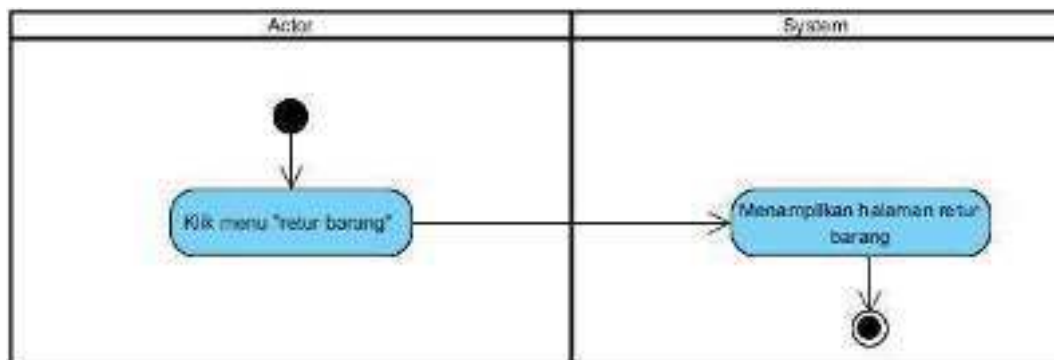
Activity diagram ini menggambarkan alur aktivitas dari proses menambah data barang. Aktor yang dapat menambah data stok masuk atau pembelian pada masing-masing barang adalah admin. Detail penjelasan alur aktivitas dari proses menambah data stok dapat dilihat pada Gambar 4.17.

p. *Activity Diagram Melihat Data Stok*

Activity diagram ini menggambarkan alur aktivitas dari proses melihat data stok barang. Aktor yang dapat melihat data stok barang adalah kasir. Detail penjelasan alur aktivitas dari proses melihat data stok dapat dilihat pada Gambar 4.18.



Gambar 4.18 Activity Diagram Melihat Data Stok



Gambar 4.19 Activity Diagram Manajemen Data Retur Barang (view)

r. *Activity Diagram Manajemen Data Retur Barang (input)*

Activity diagram ini menggambarkan alur aktivitas dari proses menambah data retur barang. Aktor yang dapat menambah data retur barang atau stok keluar pada masing-masing barang adalah admin. Terdapat *field* keterangan digunakan untuk mengetahui kondisi barang yang akan dilakukan retur yaitu “kadaluarsa” atau “rusak atau cacat”. Detail penjelasan alur aktivitas dari proses menambah data retur barang dapat dilihat pada Gambar 4.20.

s. *Activity Diagram Menambah Data Penjualan*

Activity diagram ini menggambarkan alur aktivitas dari proses menambah data retur barang. Aktor yang dapat menambah data penjualan pada masing-masing

barang adalah kasir. Terdapat *field* jumlah stok keluar yang digunakan sebagai parameter perhitungan metode *Double Exponential Smoothing*. Detail penjelasan alur aktivitas dari proses menambah data penjualan dapat dilihat pada Gambar 4.21.

t. *Activity Diagram Melihat Laporan Penjualan*

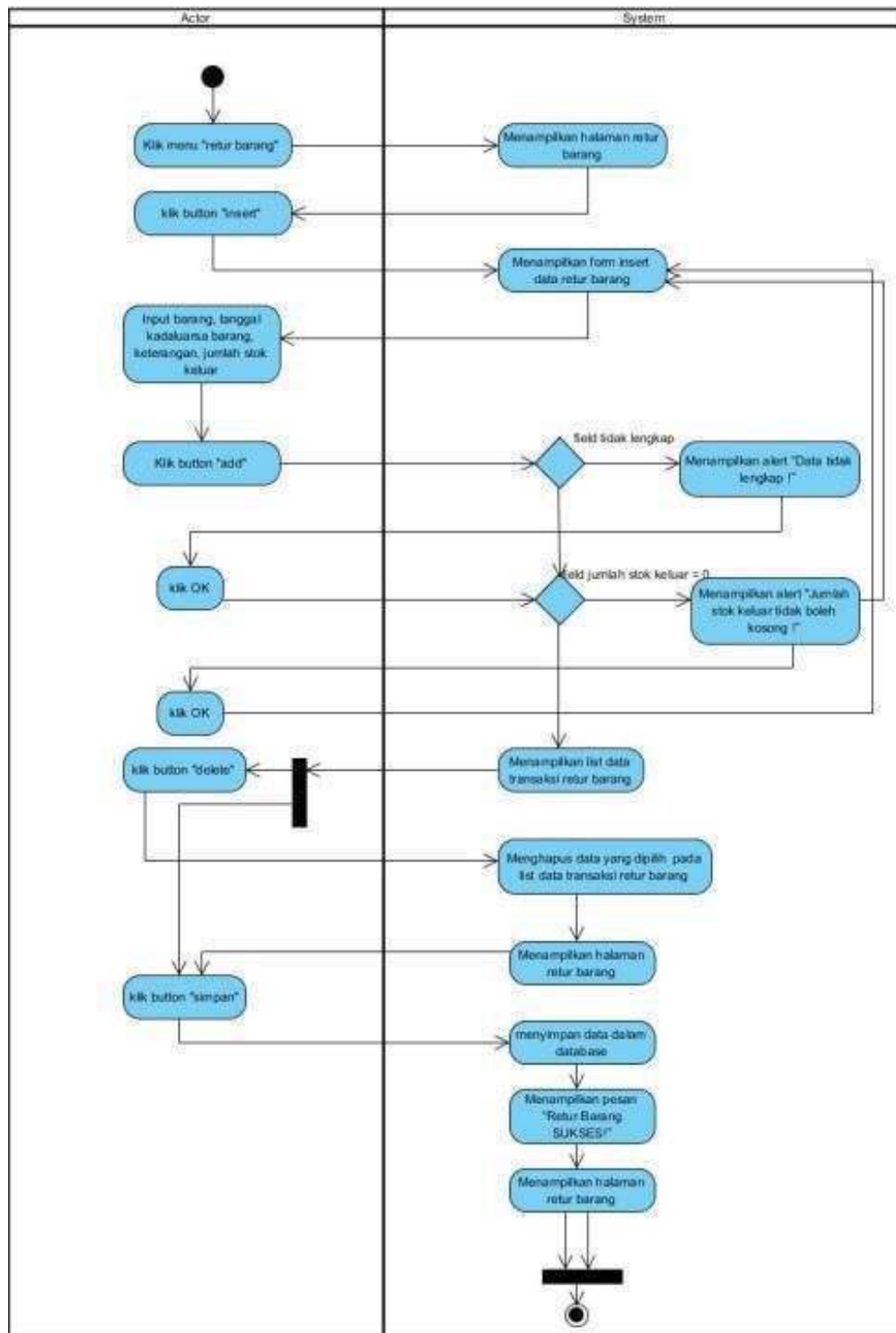
Activity diagram ini menggambarkan alur aktivitas dari proses melihat data laporan penjualan. Aktor yang dapat melihat data penjualan sesuai barang dan interval laporan adalah admin. Detail penjelasan alur aktivitas dari proses melihat laporan penjualan dapat dilihat pada Gambar 4.22.

u. *Activity Diagram Melihat Hasil Prediksi*

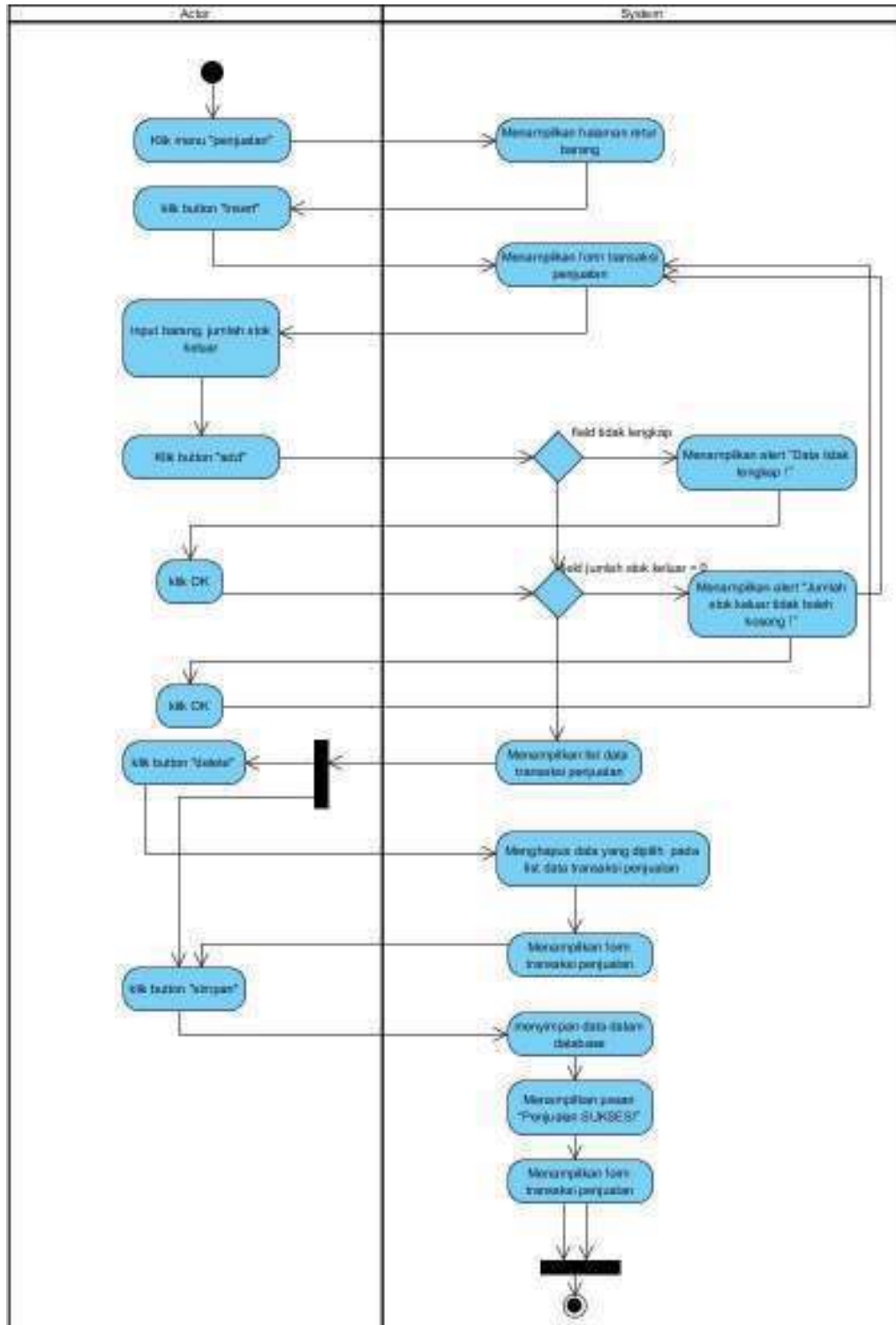
Activity diagram ini menggambarkan alur aktivitas dari proses melihat perhitungan hasil prediksi data persediaan menggunakan metode *Double Exponential Smoothing* dan *Economy Order Quantity* pada masing-masing barang. Aktor yang dapat hasil perhitungan prediksi persediaan barang adalah admin. Detail penjelasan alur aktivitas dari proses melihat laporan penjualan dapat dilihat pada Gambar 4.23.

v. *Activity Diagram Logout*

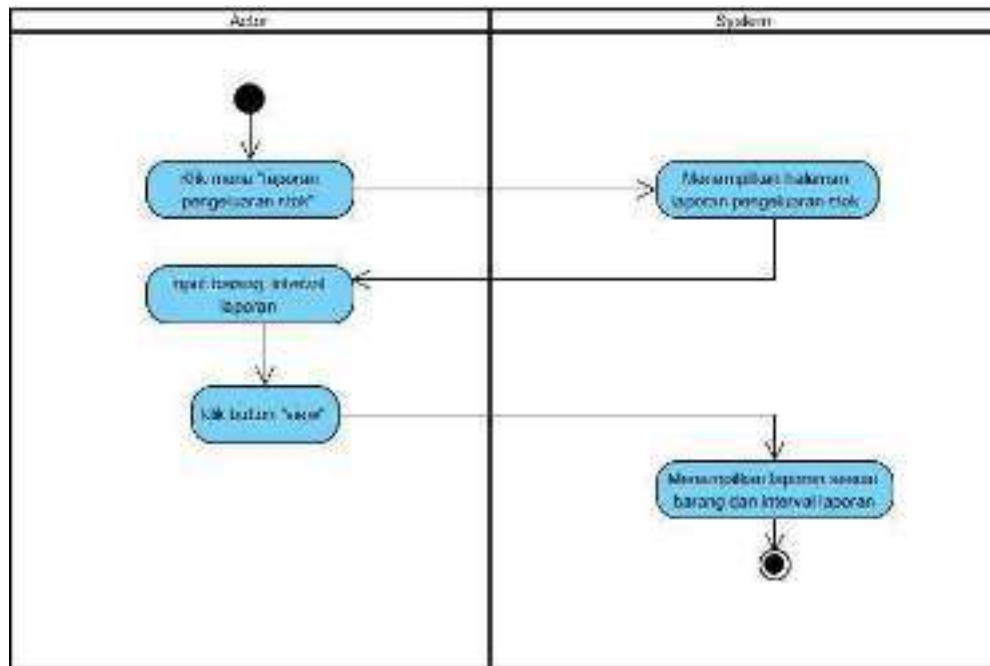
Activity diagram ini menggambarkan alur aktivitas dari proses keluar dari sitem. Aktor yang dapat melakukan *logout* adalah admin dan kasir. Detail penjelasan alur aktivitas dari proses keluar dari sitem dapat dilihat pada Gambar 4.24.



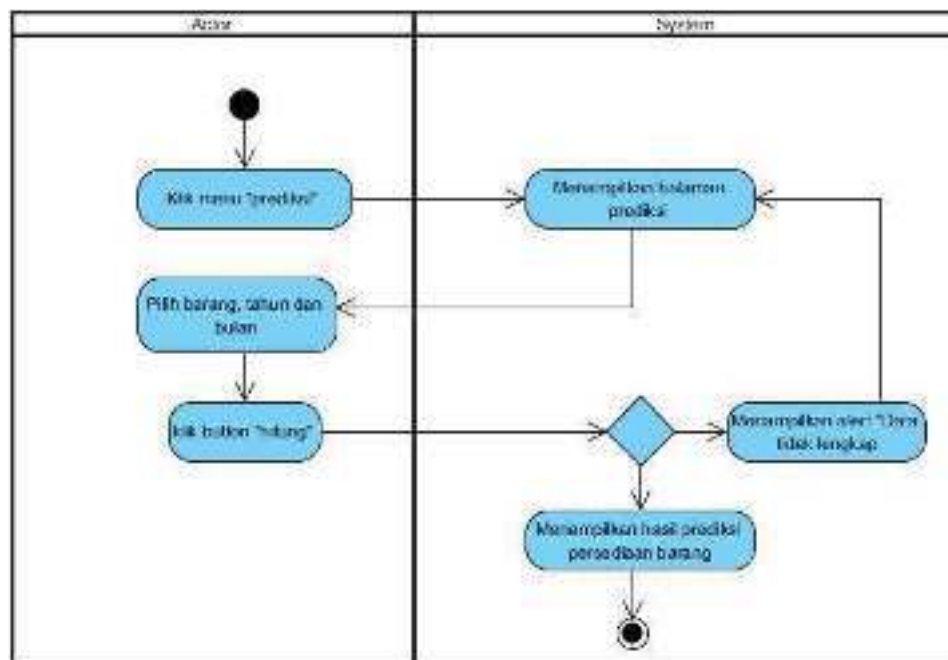
Gambar 4 20 Activity Diagram Manajemen Data Retur Barang (input)



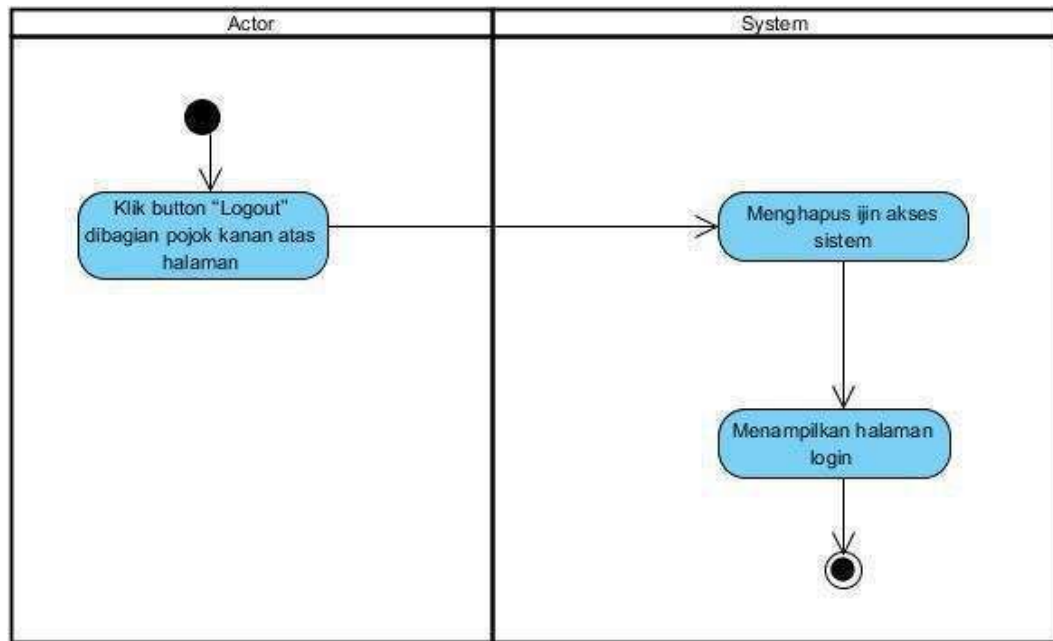
Gambar 4.21 Activity Diagram Menambah Data Penjualan



Gambar 4.22 Activity Diagram Melihat Laporan Penjualan



Gambar 4.23 Activity Diagram Melihat Hasil Prediksi



Gambar 4.24 Activity Diagram Logout

4.3.5 Sequence Diagram

Sequence diagram merupakan interaksi antar objek di dalam sistem dengan objek di sekitar system. Sequence Diagram diawali dari apa yang me-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

a. Sequence Diagram Login

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk proses *login*. Aktor yang dapat melakukan *login* adalah admin dan kasir. Pada sequence ini terdapat class view *v_login* dan *v_dashboard*, class controller *c_login* dan class model *m_login*. Sequence diagram *login* lebih lengkapnya dapat dilihat pada Gambar 4.25.

data *user*. Aktor yang dapat menambah data user adalah admin. Pada sequence ini terdapat class view *v_dashboard*, *v_user* dan *v_update_user*, class controller *c_user*, class model *m_login* dan *m_user* serta di dalam class tersebut terdapat banyak method yang dipanggil. *Sequence* diagram manajemen data *user (input)* lebih lengkapnya dapat dilihat pada Gambar 4.27.

d. *Sequence* Diagram Manajemen Data User (*edit*)

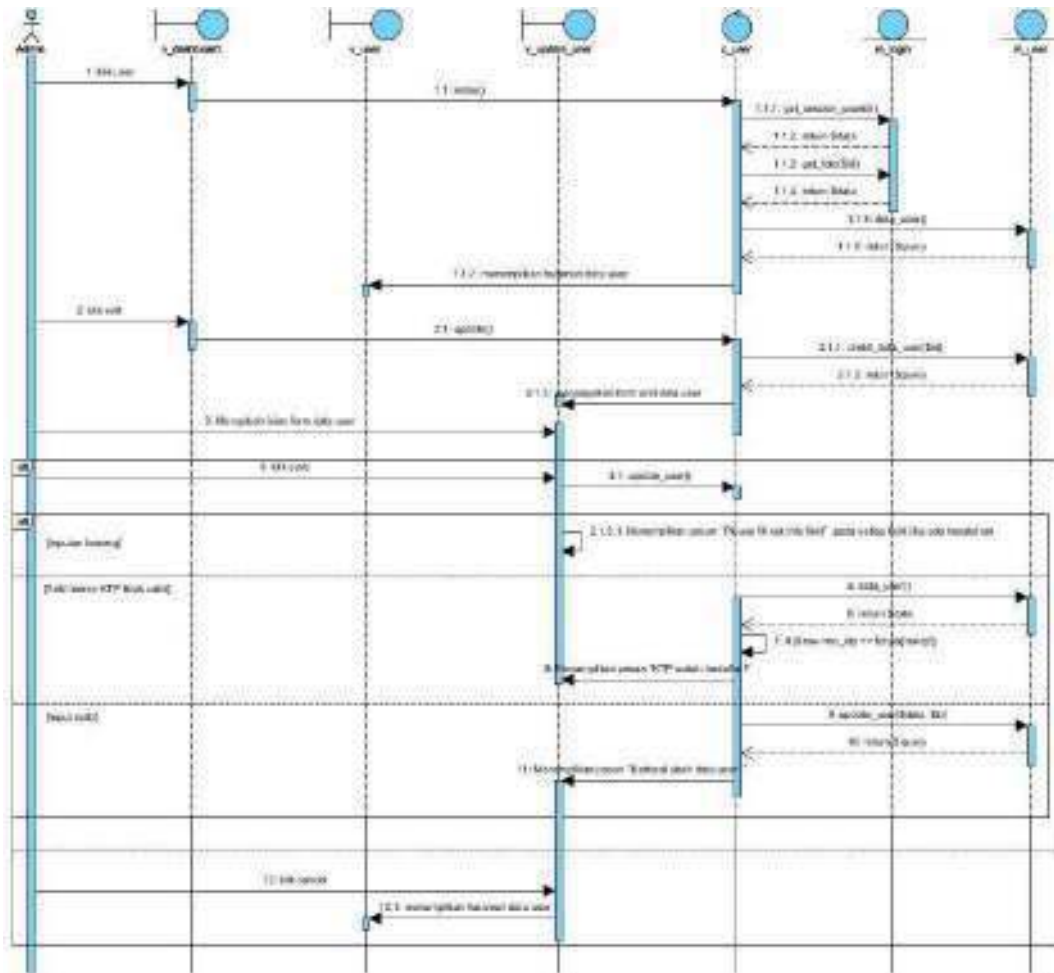
Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk mengubah data *user*. Aktor yang dapat mengubah data user adalah admin. Pada sequence ini terdapat class view *v_dashboard*, *v_user*, dan *v_edit_user*, class controller *c_user*, class model *m_login* dan *m_user* serta di dalam class tersebut terdapat banyak method yang dipanggil. *Sequence* diagram manajemen data *user (edit)* lebih lengkapnya dapat dilihat pada Gambar 4.28.

e. *Sequence* Diagram Manajemen Data Jenis (*view*)

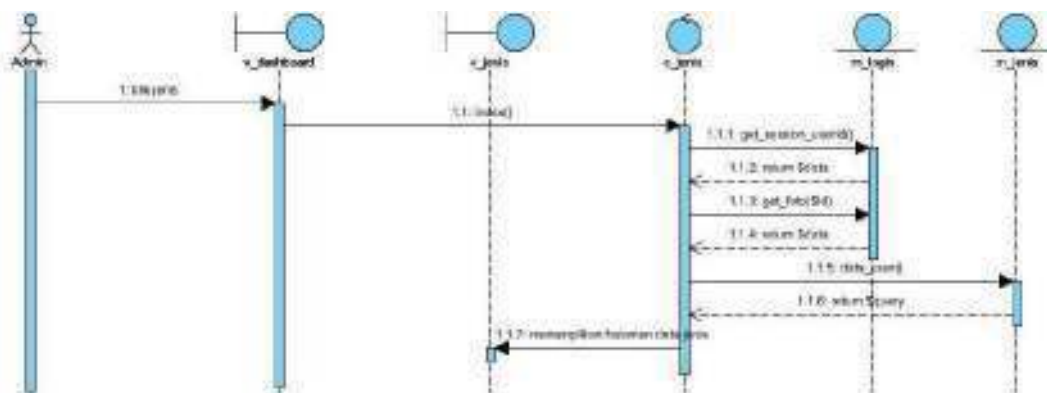
Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melihat data jenis. Aktor yang dapat melihat data jenis adalah admin. Pada sequence ini terdapat class view *v_dashboard* dan *v_jenis*, class controller *c_jenis*, class model *m_login* dan *m_jenis* serta di dalam class tersebut terdapat banyak method yang dipanggil. *Sequence* diagram manajemen data jenis (*view*) lebih lengkapnya dapat dilihat pada Gambar 4.29.

f. *Sequence* Diagram Manajemen Data Jenis (*input*)

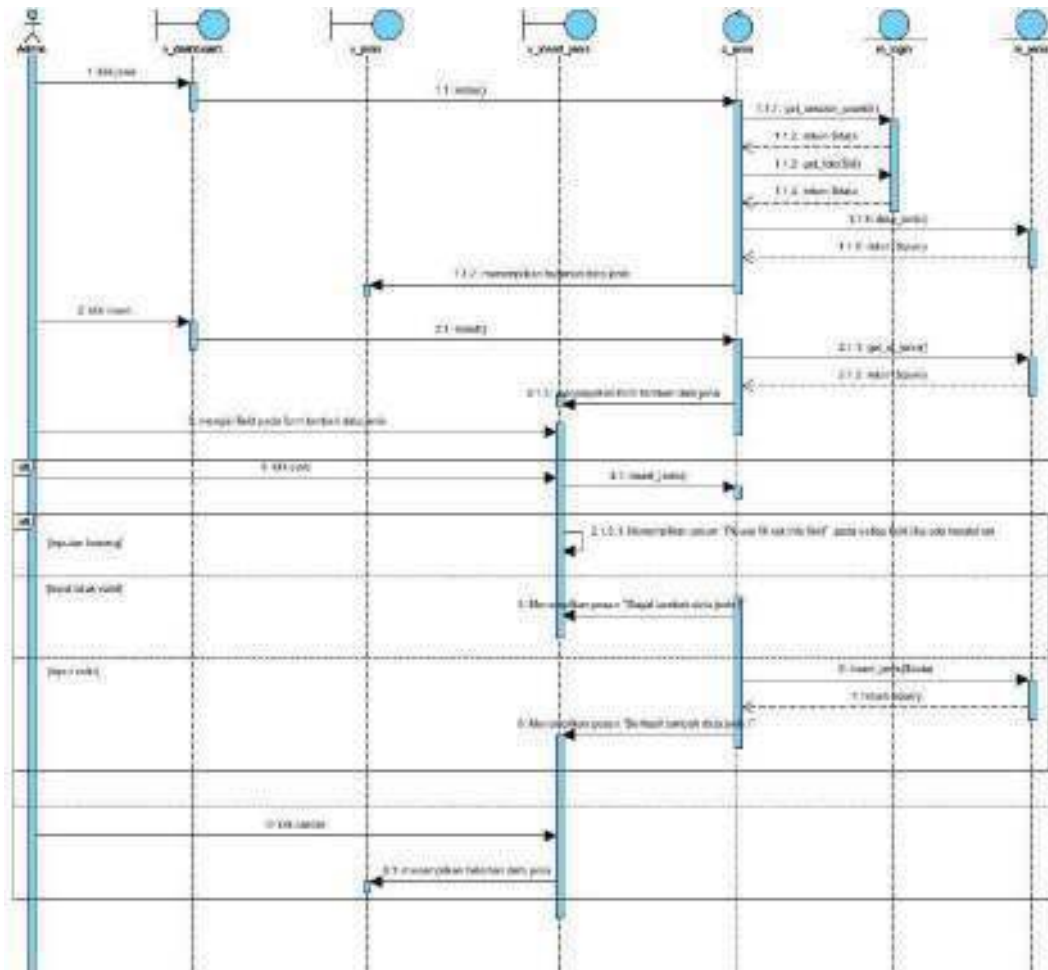
Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk menambah data jenis. Aktor yang dapat menambah data jenis adalah admin. Pada sequence ini terdapat class view *v_dashboard*, *v_jenis* dan *v_insert_jenis*, class controller *c_jenis*,



Gambar 4.28 Sequence Diagram Manajemen Data User (edit)



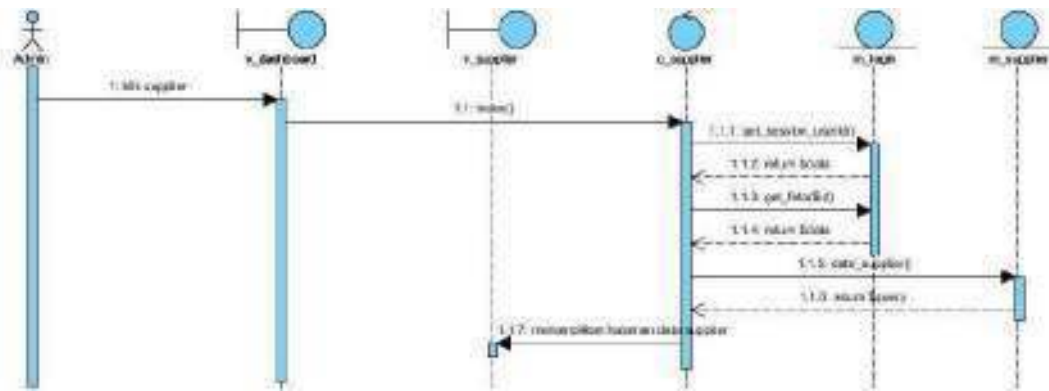
Gambar 4.29 Sequence Diagram Manajemen Data Jenis (view)



Gambar 4.30 Sequence Diagram Manajemen Data Jenis (input)

g. *Sequence Diagram Manajemen Data Jenis (edit)*

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk mengubah data jenis. Aktor yang dapat mengubah data jenis adalah admin. Pada *sequence* ini terdapat class view `v_dashboard`, `v_jenis`, dan `v_update_jenis`, class controller `c_jenis`, class model `m_login` dan `m_jenis` serta di dalam class tersebut terdapat banyak method yang dipanggil. *Sequence* diagram manajemen data jenis (*edit*) lebih lengkapnya dapat dilihat pada Gambar 4.31.



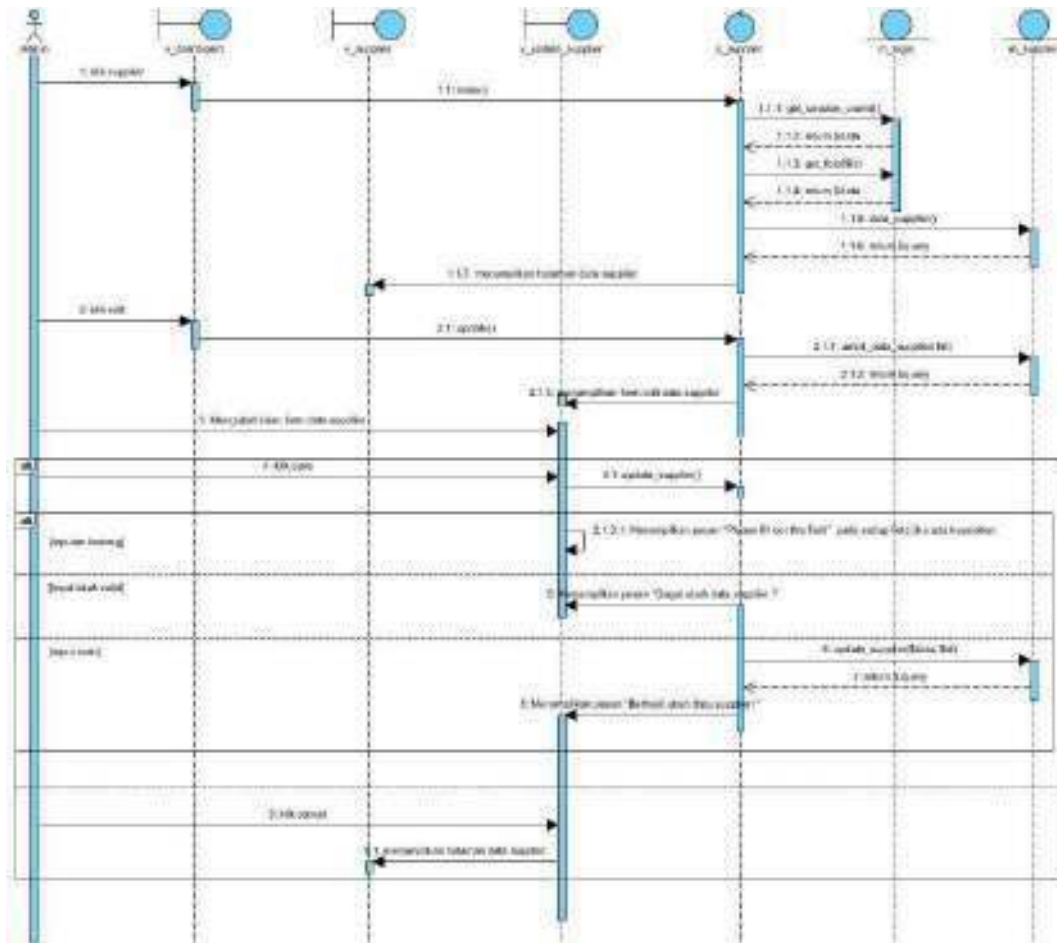
Gambar 4.32 Sequence Diagram Manajemen Data Supplier (view)

i. *Sequence* Diagram Manajemen Data Supplier (*input*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk menambah data *supplier*. Aktor yang dapat menambah data *supplier* adalah admin. Pada *sequence* ini terdapat class view *v_dashboard*, *v_supplier* dan *v_insert_supplier*, class controller *c_supplier*, class model *m_login* dan *m_supplier* serta di dalam class tersebut terdapat banyak method yang dipanggil. *Sequence* diagram manajemen data *supplier (input)* lebih lengkapnya dapat dilihat pada Gambar 4.33.

j. *Sequence* Diagram Manajemen Data Supplier (*edit*)

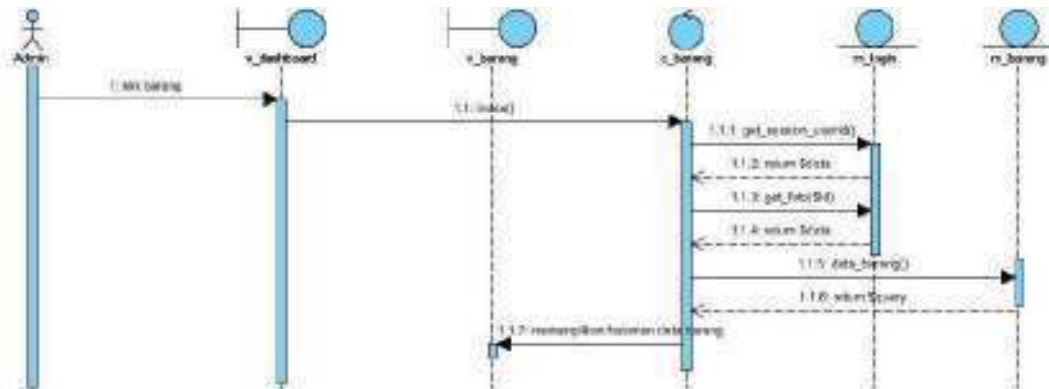
Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk mengubah data *supplier*. Aktor yang dapat mengubah data *supplier* adalah admin. Pada *sequence* ini terdapat class view *v_dashboard*, *v_supplier*, dan *v_update_supplier*, class controller *c_supplier*, class model *m_login* dan *m_supplier* serta di dalam class tersebut terdapat banyak method yang dipanggil. *Sequence* diagram manajemen data *supplier (edit)* lebih lengkapnya dapat dilihat pada Gambar 4.34.



Gambar 4.34 Sequence Diagram Manajemen Data Supplier (edit)

k. *Sequence* Diagram Manajemen Data Barang (*view*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melihat data barang. Aktor yang dapat melihat data barang adalah admin. Pada *sequence* ini terdapat class view `v_dashboard` dan `v_barang`, class controller `c_barang`, class model `m_login` dan `m_barang` serta di dalam class tersebut terdapat banyak method yang dipanggil. *Sequence* diagram manajemen data barang (*view*) lebih lengkapnya dapat dilihat pada Gambar 4.35.



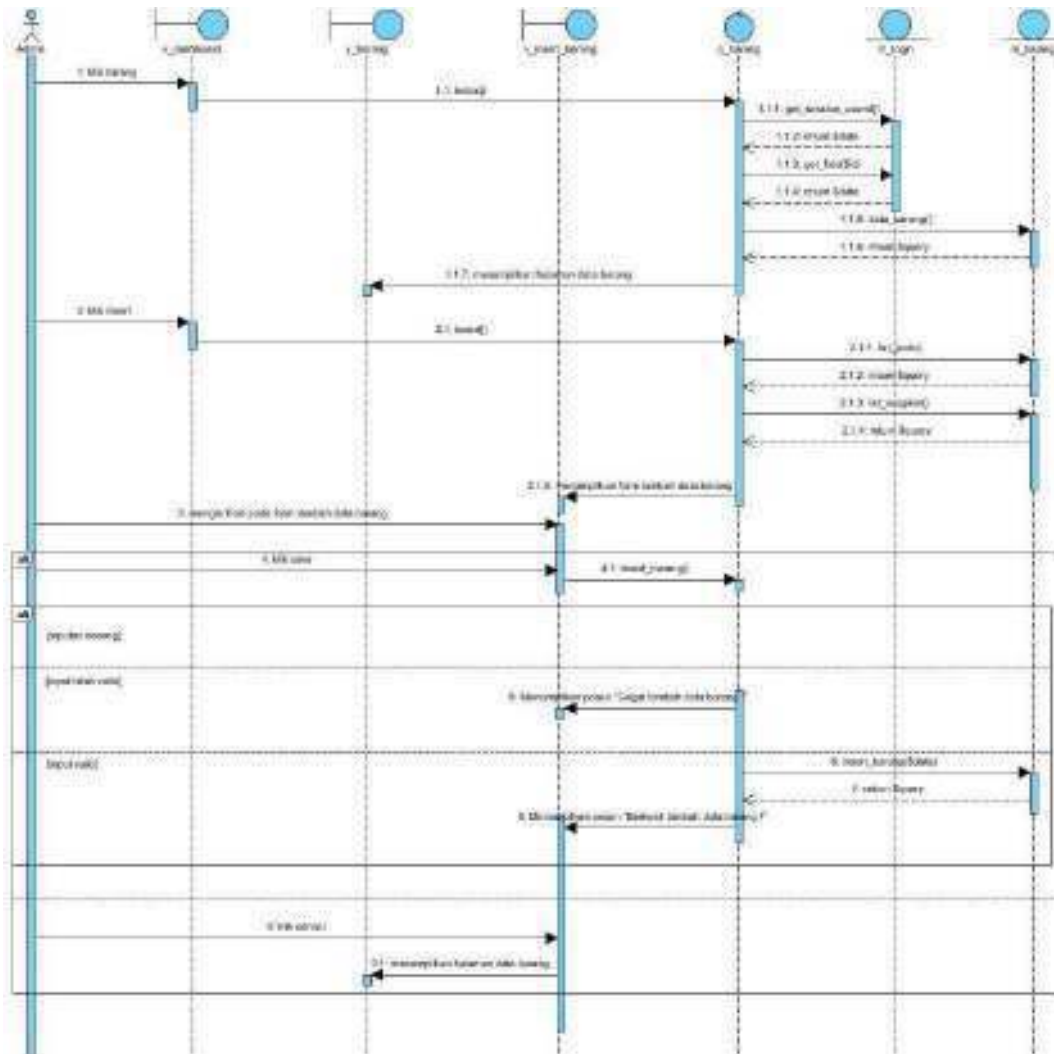
Gambar 4.35 Sequence Diagram Manajemen Data Barang (view)

l. *Sequence Diagram Manajemen Data Barang (input)*

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk menambah data barang. Aktor yang dapat menambah data barang adalah admin. Pada *sequence* ini terdapat class view *v_dashboard*, *v_barang* dan *v_insert_barang*, class controller *c_barang*, class model *m_login* dan *m_barang* serta di dalam class tersebut terdapat banyak method yang dipanggil. *Sequence* diagram manajemen data barang (*input*) lebih lengkapnya dapat dilihat pada Gambar 4.36.

m. *Sequence Diagram Manajemen Data Barang (edit)*

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk mengubah data barang. Aktor yang dapat mengubah data barang adalah admin. Pada *sequence* ini terdapat class view *v_dashboard*, *v_barang*, dan *v_update_barang*, class controller *c_barang*, class model *m_login* dan *m_barang* serta di dalam class tersebut terdapat banyak method yang dipanggil. *Sequence* diagram manajemen data barang (*edit*) lebih lengkapnya dapat dilihat pada Gambar 4.37.

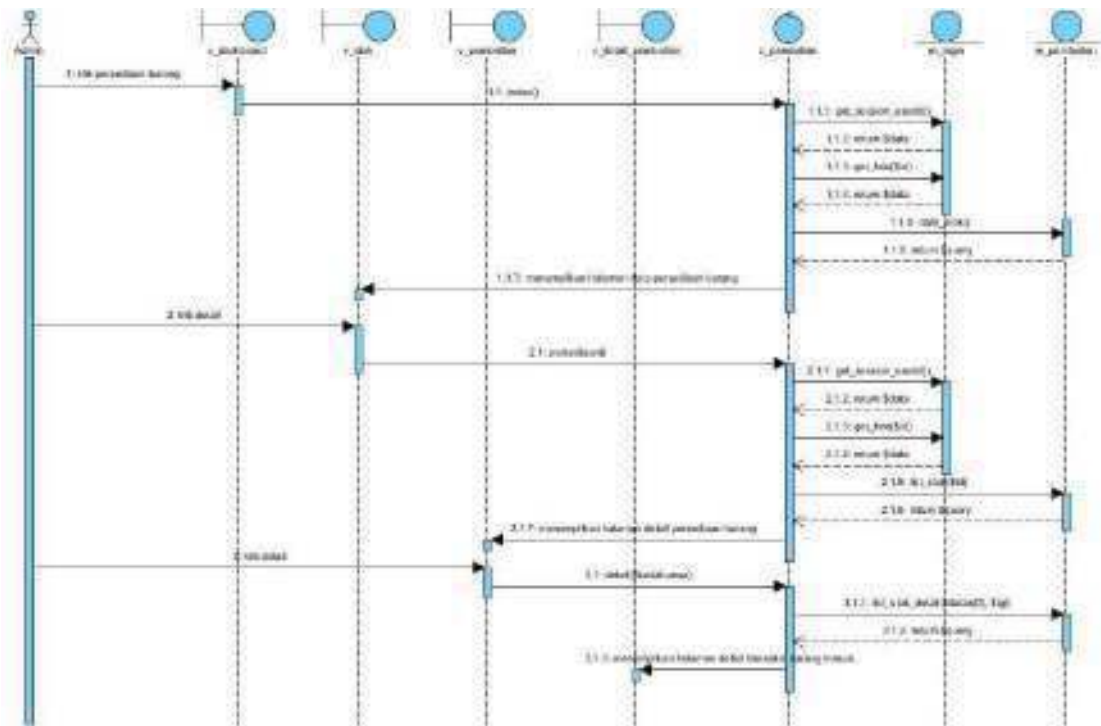


Gambar 4.37 Sequence Diagram Manajemen Data Barang (edit)

n. *Sequence* Diagram Manajemen Data Stok (*view*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melihat data stok. Aktor yang dapat melihat data stok adalah admin. Pada *sequence* ini terdapat class view *v_dashboard* dan *v_pembelian*, class controller *c_pembelian*, class model *m_login* dan *m_pembelian* serta di dalam class tersebut terdapat banyak

method yang dipanggil. *Sequence* diagram manajemen data stok (*view*) lebih lengkapnya dapat dilihat pada Gambar 4.38.



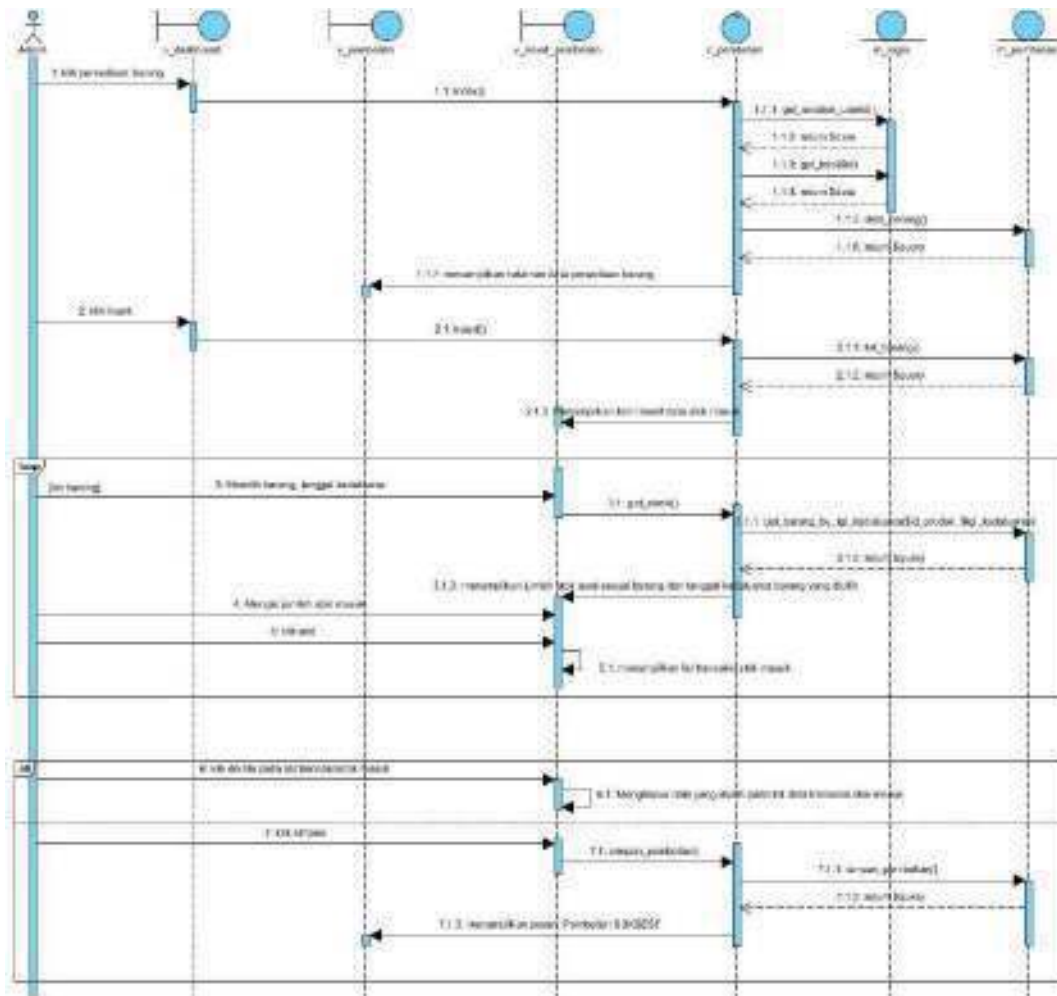
Gambar 4.38 Sequence Diagram Manajemen Data Stok (*view*)

o. *Sequence* Diagram Manajemen Data Stok (*input*)

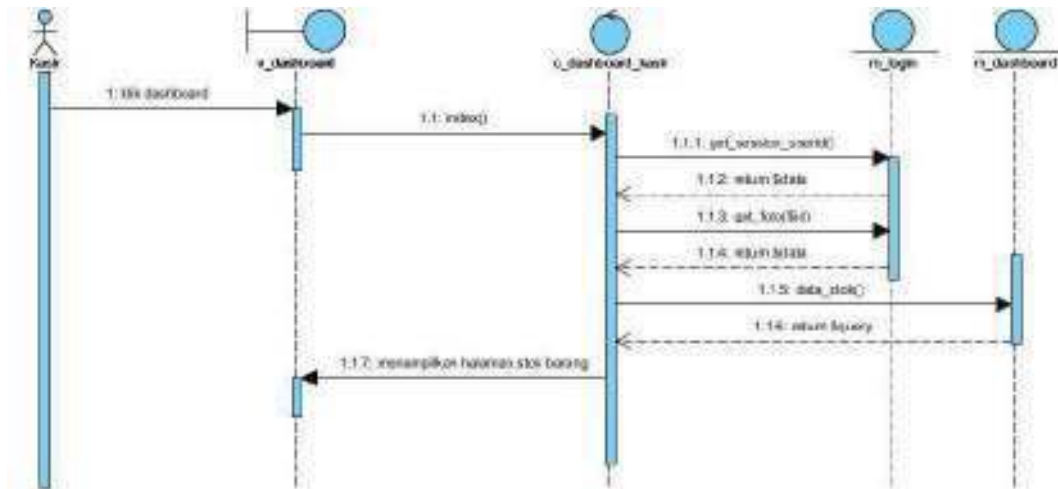
Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk menambah data stok masuk. Aktor yang dapat menambah data stok adalah admin. Pada *sequence* ini terdapat class view *v_dashboard*, *v_pembelian* dan *v_insert_pembelian*, class controller *c_pembelian*, class model *m_login* dan *m_pembelian* serta di dalam class tersebut terdapat banyak method yang dipanggil. *Sequence* diagram manajemen data stok (*input*) lebih lengkapnya dapat dilihat pada Gambar 4.39.

p. *Sequence* Diagram Melihat Data Stok

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melihat data stok. Aktor yang dapat melihat data stok adalah kasir. Pada *sequence* ini terdapat class view `v_dashboard`, class controller `c_dashboard`, class model `m_login` dan `m_dashboard` serta di dalam class tersebut terdapat banyak method yang dipanggil. *Sequence* diagram melihat data stok lebih lengkapnya dapat dilihat pada Gambar 4.40.



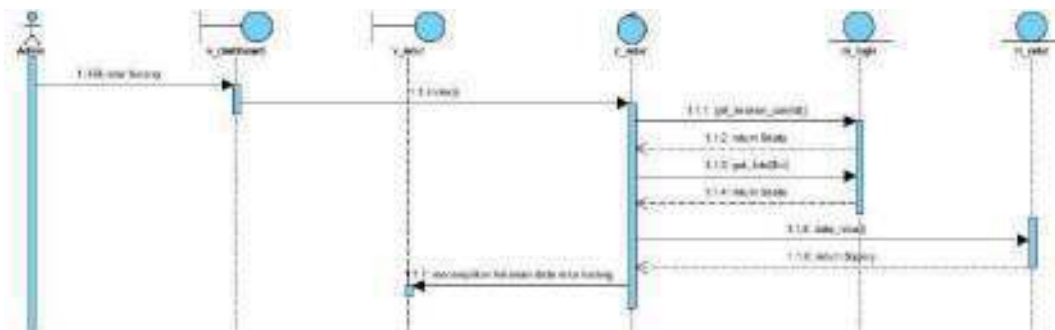
Gambar 4.39 *Sequence* Diagram Manajemen Data Stok (input)



Gambar 4.40 Sequence Diagram Melihat Data Stok

q. *Sequence* Diagram Manajemen Data Retur Barang (*view*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melihat data retur. Aktor yang dapat melihat data retur adalah admin. Pada *sequence* ini terdapat class view `v_dashboard` dan `v_retur`, class controller `c_retur`, class model `m_login` dan `m_retur` serta di dalam class tersebut terdapat banyak method yang dipanggil. *Sequence* diagram manajemen data retur barang (*view*) lebih lengkapnya dapat dilihat pada Gambar 4.41.



Gambar 4.41 Sequence Diagram Manajemen Data Retur Barang (*view*)

r. *Sequence* Diagram Manajemen Data Retur Barang (*input*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk menambah data retur barang. Aktor yang dapat menambah data retur adalah admin. Pada *sequence* ini terdapat class view v_dashboard, v_retur dan v_insert_retur, class controller c_retur, class model m_login dan m_retur serta di dalam class tersebut terdapat banyak method yang dipanggil. *Sequence* diagram manajemen data retur barang (*input*) lebih lengkapnya dapat dilihat pada Gambar 4.42.

s. *Sequence* Diagram Menambah Data Penjualan

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk menambah data stok keluar. Aktor yang dapat menambah data penjualan atau penjualan adalah kasir. Pada *sequence* ini terdapat class view v_dashboard dan v_penjualan, class controller c_penjualan, class model m_login dan m_penjualan serta di dalam class tersebut terdapat banyak method yang dipanggil. *Sequence* diagram manajemen data penjualan lebih lengkapnya dapat dilihat pada Gambar 4.43.

t. *Sequence* Diagram Melihat Laporan Penjualan

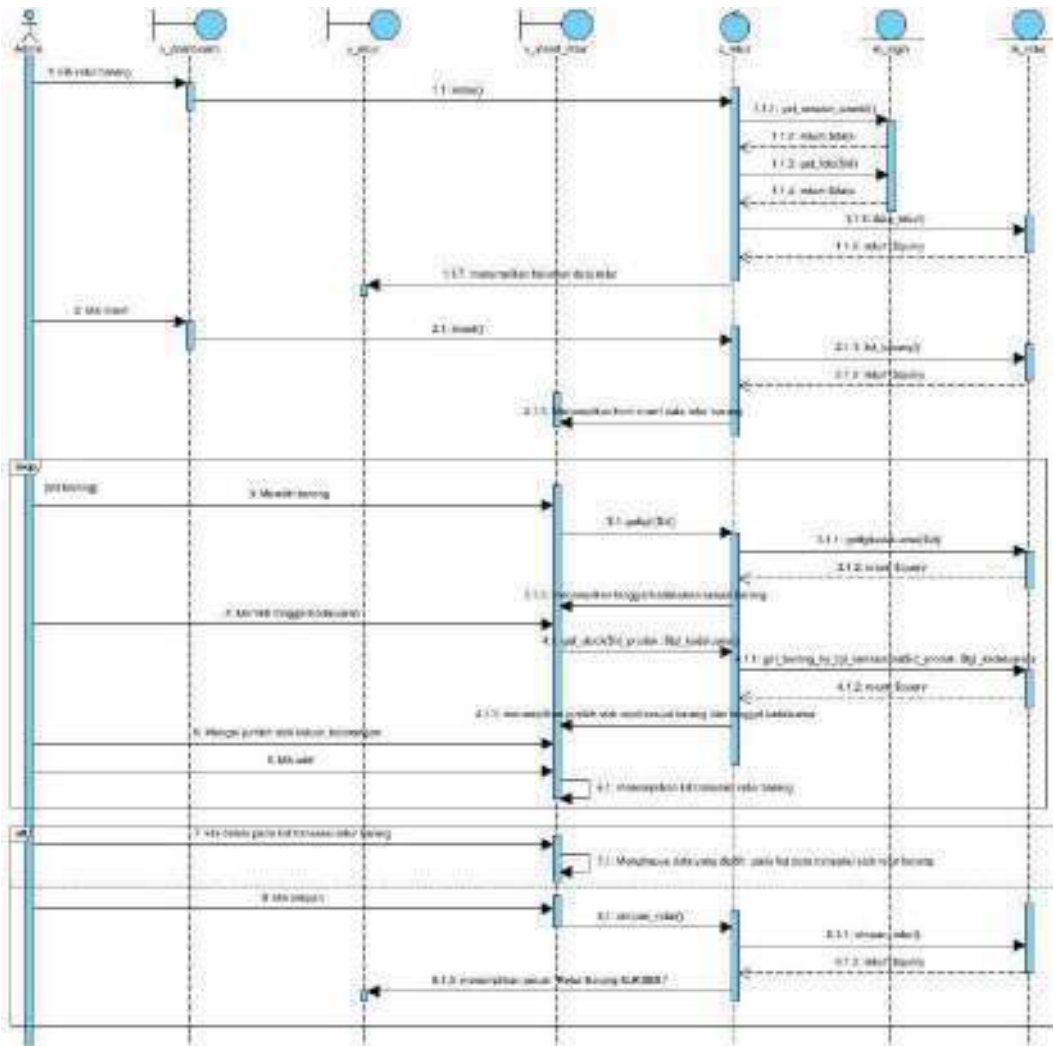
Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melihat laporan penjualan sesuai barang dan interval laporan. Aktor yang dapat melihat laporan penjualan atau penjualan sesuai nama barang dan interval yang pilih adalah admin. Pada *sequence* ini terdapat class view v_dashboard dan v_laporan, class controller c_laporan, class model m_login dan m_laporan serta di dalam class tersebut terdapat banyak method yang dipanggil. *Sequence* diagram Laporan Penjualan lebih lengkapnya dapat dilihat pada Gambar 4.44.

u. *Sequence Diagram Melihat Hasil Prediksi*

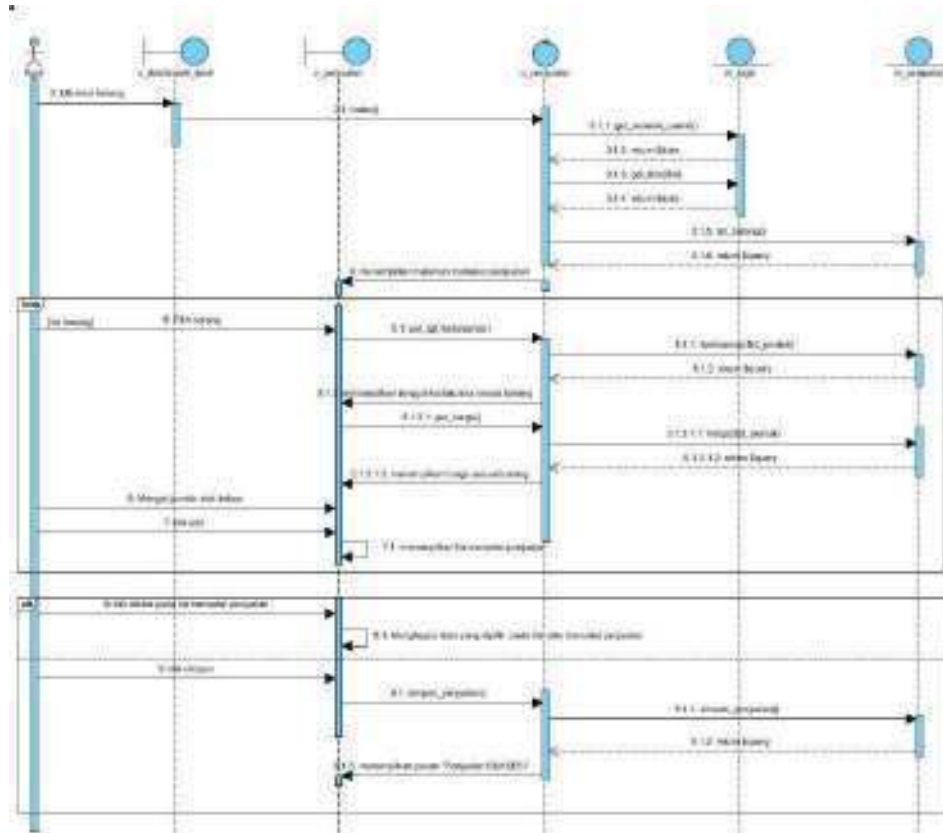
Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk keluar dari sistem. Aktor yang dapat melihat hasil prediksi adalah admin. Pada *sequence* ini terdapat class view v_dashboard dan v_prediksi, class controller c_login, class model m_login dan m_prediksi serta di dalam class tersebut terdapat banyak method yang dipanggil. *Sequence diagram logout* lebih lengkapnya dapat dilihat pada Gambar 4.47.

v. *Sequence Diagram Logout*

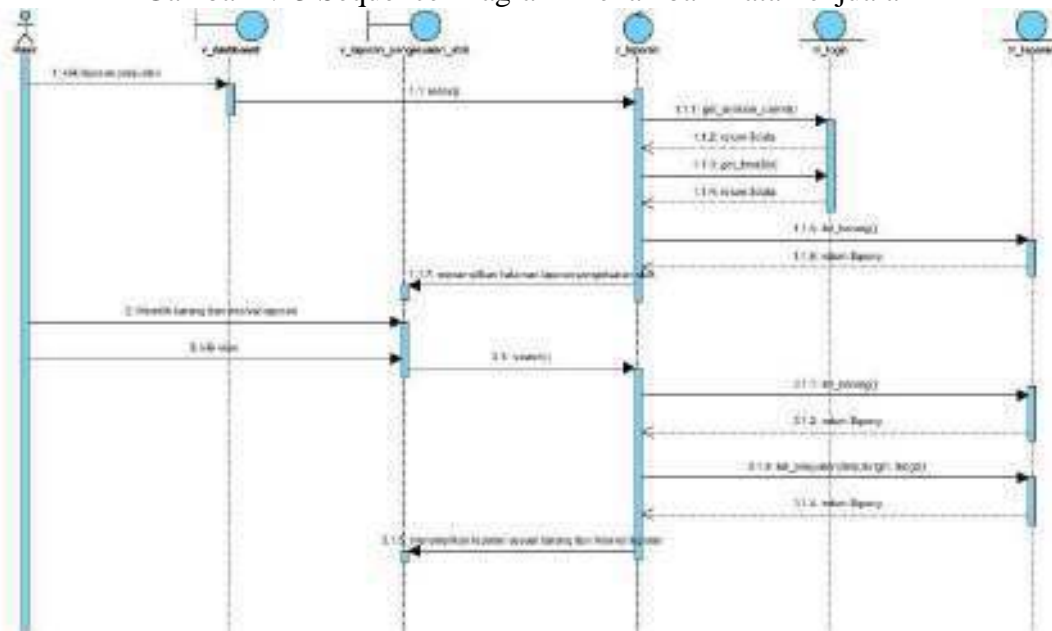
Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk keluar dari sistem. Aktor yang dapat melakukan *logout* adalah admin dan kasir. Pada *sequence* ini terdapat class view v_dashboard dan v_login, class controller c_login, class model m_login serta di dalam class tersebut terdapat banyak method yang dipanggil. *Sequence diagram logout* lebih lengkapnya dapat dilihat pada Gambar 4.46.



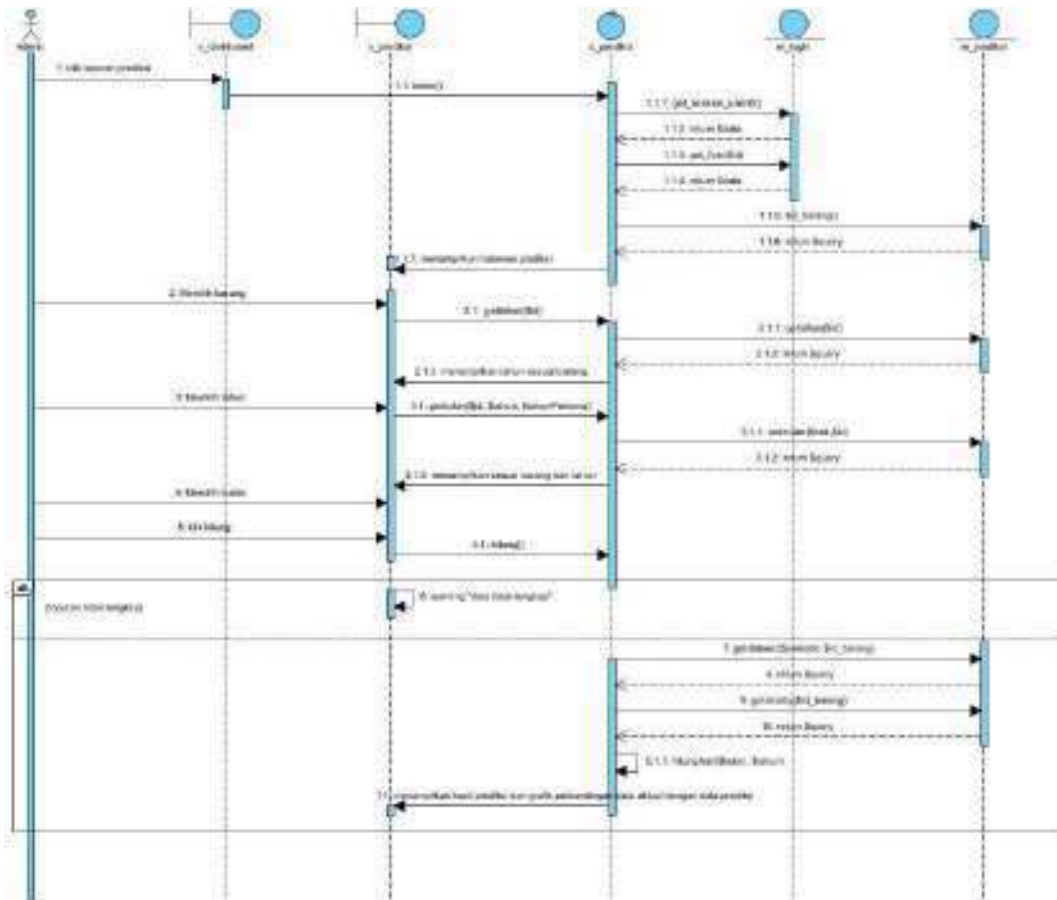
Gambar 4.42 Sequence Diagram Manajemen Data Retur (input)



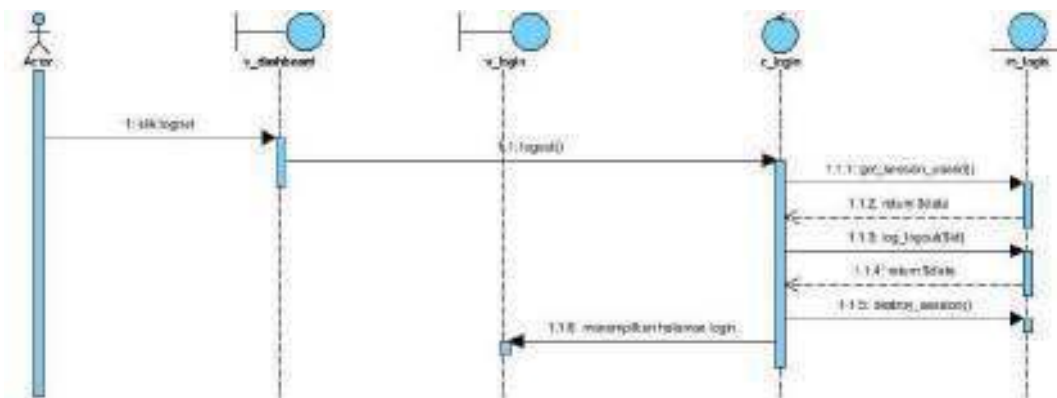
Gambar 4.43 Sequence Diagram Menambah Data Penjualan



Gambar 4.44 Sequence Diagram Melihat Laporan Penjualan



Gambar 4.45 Sequence Diagram Melihat Hasil Prediksi



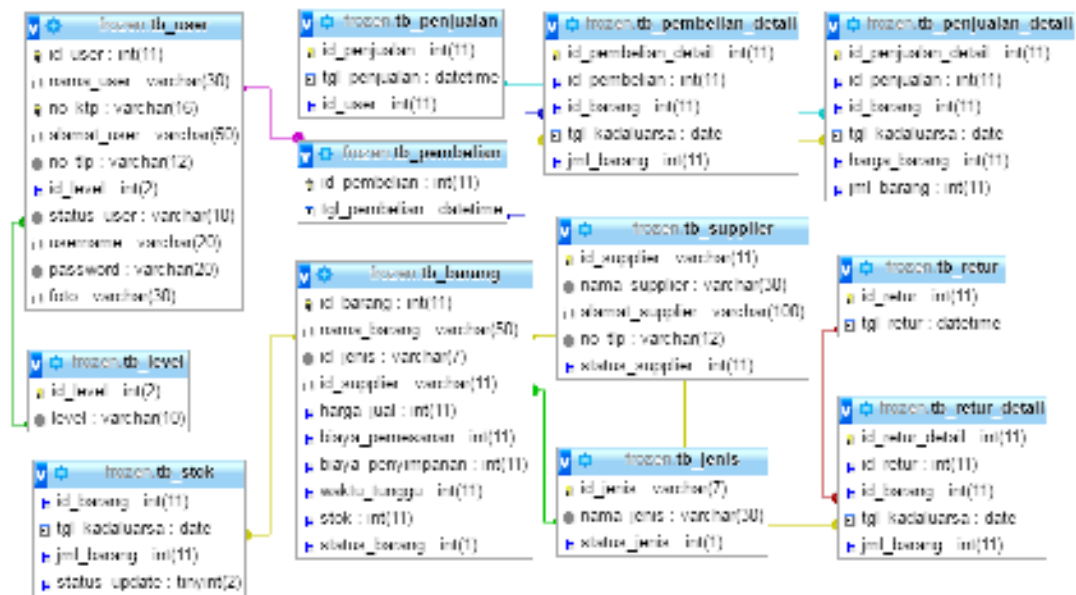
Gambar 4.46 Sequence Diagram Logout

4.3.6 *Class Diagram*

Class diagram merupakan Gambaran dari setiap class yang diimplementasikan pada program dan yang berisi nama method serta nama atribut dari setiap classnya. Class diagram ini terdapat tiga package yaitu view, model dan controller dan mempunyai berbagai macam relasi yang menggambarkan hubungan antar class. Dalam class view terdapat beberapa Tabel seperti v_login, v_dashboard, v_user, v_insert_user, v_update_user, v_jenis, v_insert_jenis, v_update_jenis, v_supplier, v_insert_supplier, v_barang, v_insert_barang, v_update_barang, v_pembelian, v_insert_pembelian, v_retur, v_insert_retur, v_laporan_pengeluaran_stok, v_prediksi, dan v_penjualan yang terbungkus ke dalam satu class yaitu loader_view. Dalam class controller juga terdapat Tabel yang sama yaitu c_login, c_dashboard, c_user, c_jenis, c_supplier, c_barang, c_pembelian, c_retur, c_laporan, c_prediksi, dan c_penjualan. Sedangkan pada class model terdapat beberapa Tabel yaitu m_login, m_dashboard, m_user, m_jenis, m_supplier, m_barang, m_pembelian, m_laporan, m_prediksi dan m_penjualan. *Class* diagram sistem pengendalian persediaan stok barang dapat dilihat pada Gambar 4.47.

4.3.7 *Entity Relationship Diagram*

Entity relationship diagram adalah suatu model untuk menjelaskan hubungan antar data dalam database berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. *Entity relationship* diagram sistem pengendalian persediaan stok barang dapat dilihat pada Gambar 4.48.



Gambar 4.48 Entity Relationship Diagram

4.2 Pengkodean Sistem

Setelah tahap desain perancangan selesai, tahap selanjutnya dalam penelitian ini yaitu tahap pengimplementasian desain perancangan ke dalam bahasa pemrograman. Bahasa pemrograman yang digunakan yaitu bahasa php, html, dan css dengan *framework Code Igniter (CI)* dan database yang digunakan adalah mysql.

Pada tahap implementasi perancangan ini menjelaskan tentang fitur – fitur yang terdapat pada sistem pengendalian persediaan stok barang . Di dalam tahap ini juga mengimplementasikan metode *Double Exponential Smoothing* dan *Economy Order Quantity* di dalam barisan kode program pada fitur manajemen prediksi persediaan barang. Detail penulisan kode program dapat dilihat pada Lampiran A.

4.3 Pengujian Sistem

Tahapan pengujian dilakukan untuk mengevaluasi sistem yang telah dibuat sistem dengan menggunakan sebuah metode pengujian sistem. Proses pengujian dilakukan dengan pengujian *white box* terlebih dahulu, kemudian akan dilanjutkan dengan pengujian *black box*.

4.3.1 White Box

Pengujian sistem dengan metode white box dilakukan untuk menguji sistem dari segi desain dan kode program. Pengujian whitebox yang dilakukan pada penelitian ini meliputi listing program, grafik alir, kompleksitas siklomatis, basis set dan test case. *Listing* program yang diujikan pada proses perhitungan metode double exponential smoothing dan economy order quantity dapat dilihat pada Gambar 4.49 dan Gambar 5.51 serta diagram alir pengujian dapat dilihat pada Gambar 4.50 dan Gambar 5.52. Sedangkan untuk *listing* program *function* lain yang terdapat dalam sistem pengendalian persediaan stok barang UD Jasmine dapat dilihat pada Lampiran B.

4.3.2 Black Box

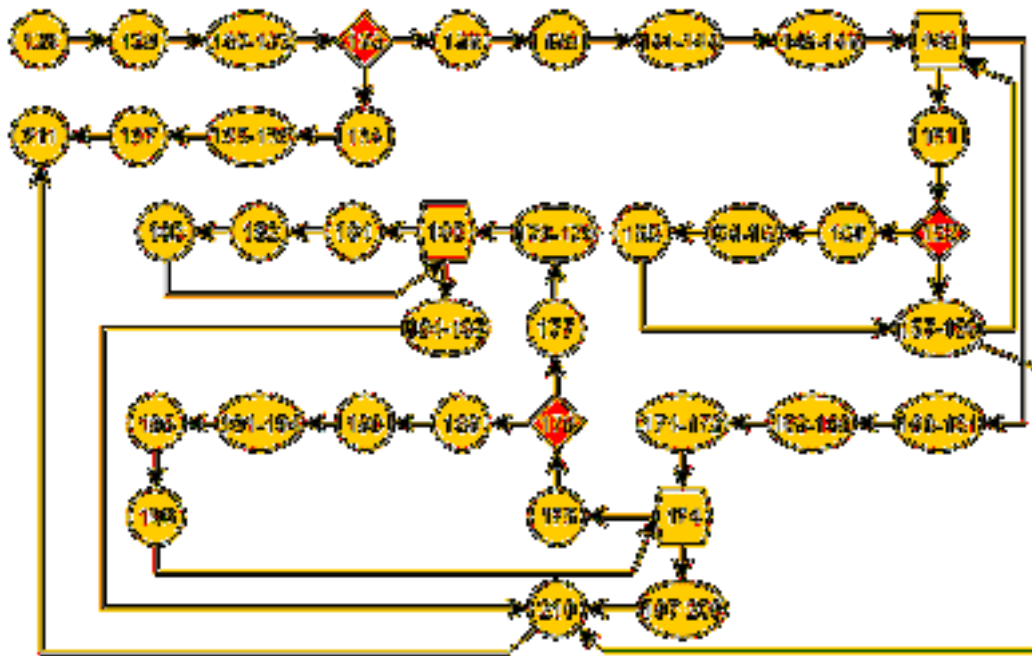
Pengujian black box berfungsi untuk menguji sistem dari segi spesifikasi fungsional sistem dengan tujuan mengetahui apakah fungsi-fungsi, inputan, dan keluaran sistem sesuai dengan spesifikasi yang dibutuhkan oleh pengguna. Pengujian dengan metode black box dilakukan oleh UD Jasmine Jember. Hasil pengujian dengan metode black box dapat dilihat pada Lampiran C.

```

128 public function hitung()
129 {
130     $id_barang = $this->input->post('id_barang');
131     $ukuran = $this->input->post('ukuran');
132     $stok = $this->input->post('stok');
133     if($id_barang == "" || $ukuran == "" || $stok == "")
134     {
135         echo "Gagal karena Data tidak lengkap!<br>";
136         redirect('admin/index');
137     }
138 }
139 else
140 {
141     // variabel
142     $maks = 0;
143     $min = 0;
144     $sum = 0;
145     $sumkuadrat = 0;
146     // perhitungan periode yang akan dijadikan dataset
147     $periode = array();
148     $stop_bulan = $ukuran;
149     $stop_tahun = $stok;
150     for($i=1;$i<=$maks;$i++)
151     {
152         $stop_bulan = 1;
153         if($stop_bulan == 0)
154         {
155             $stop_tahun = 1;
156             $stop_bulan = 1;
157         }
158         $periode[$i]['bulan'] = $stop_bulan;
159         $periode[$i]['tahun'] = $stop_tahun;
160     }
161     $dataset = $this->model->getDataset($periode, $id_barang) & result_array();
162     $kupe = $this->model->getKupe($id_barang) & result_array();
163     // inisialisasi variabel
164     $s = array();
165     $u = array();
166     $sum = array();
167     $sumk = array();
168     $sum = 0;
169     $sumk = 0;
170     // untuk data pertama (index = 0)
171     $s[0] = $dataset[0]['jumlah'];
172     $u[0] = $dataset[0]['jumlah'] * $dataset[0]['jumlah'];
173     // menghitung st, sum, map untuk index > 0
174     for($i=1;$i<=count($dataset);$i++)
175     {
176         if($i==count($dataset))
177         {
178             $sum[$i] = $st[$i-1] + $st[$i-1];
179             $sum_kuat = 0;
180             for($j=1;$j<=count($map);$j++)
181             {
182                 $sum_kuat += $map[$j];
183             }
184             $sumkuat[$i] = $sum_kuat / count($sumkuat);
185             $sum = sqrt((1 + $sumkuat * $sumkuat) * $sum[$i] + $kupe[0] * $sumkuat);
186             $sumk = $sumkuat * ($sum[$i] + $kupe[0] * $sumkuat) + $kupe[0] * $sumkuat * $sumkuat;
187             $sum = (($sum[$i] + $kupe[0] * $sumkuat) * $sumkuat) + $kupe[0] * $sumkuat * $sum;
188         }
189     }
190     else
191     {
192         $s[$i] = ($kupe + $dataset[$i]['jumlah']) * ((1 - $kupe) * ($st[$i-1] + $st[$i-1]));
193         $u[$i] = ($sumk + ($st[$i] * $st[$i-1]) * ((1 - $sumk) * $st[$i-1]));
194         $sum[$i] = $s[$i-1] + $s[$i];
195         $sumkuat[$i] = ((sum($dataset[$i]['jumlah']) * $st[$i]) / $dataset[$i]['jumlah']) * 100;
196     }
197 }
198 $session = $this->model->get_session_session();
199 $id = $session['session_session'];
200 $session['foto'] = $this->model->get_foto($id);
201 $session['barang'] = $this->model->get_barang();
202 $session['dataset'] = $dataset;
203 $session['st'] = $st;
204 $session['u'] = $u;
205 $session['sum'] = $sum;
206 $session['sumk'] = $sumk;
207 $session['sumkuat'] = $sumkuat;
208 $session['map'] = $map;
209 $this->load->view('admin/index', $session);
210 }
211 }

```

Gambar 4.49 Kode program function hitung()



Gambar 4.50 Diagram alir function hitung()

$$CC = \text{EDGE} - \text{NODE} + 2$$

$$CC = 44 - 39 + 2$$

$$CC = 7$$

Maka jalur basis set pada pengujian di atas adalah 128-129-130-131-132-133-134-135-136-137-211, 128-129-130-131-132-133-138-139-141-142-143-144-146-147-148-149-151-152-153-154-155-156-157-158-210, 128-129-130-131-132-133-138-139-141-142-143-144-146-147-148-149-151-152-157-158-210, 128-129-130-131-132-133-138-139-141-142-143-144-146-147-148-149-160-161-163-164-165-166-167-168-169-171-172-174-175-176-177-178-179-180-181-182-183-180, 128-129-130-131-132-133-138-139-141-142-143-144-146-147-148-149-160-161-163-164-165-166-167-168-169-171-172-174-175-176-177-178-179-180-184-185-186-

187-210, 128-129-130-131-132-133-138-139-141-142-143-144-146-147-148-149-160-161-163-164-165-166-167-168-169-171-172-174-175-176-189-190-191-192-193-194-195-196-174, 128-129-130-131-132-133-138-139-141-142-143-144-146-147-148-149-160-161-163-164-165-166-167-168-169-171-172-174-197-198-199-200-201-202-203-204-205-206-207-208-209-210

Pengujian kebenaran keempat jalur tersebut dapat dilihat pada Tabel 4.50 sampai 4.53

Tabel 4.25 Test case pengujian fungsi hitung jalur 1

| | |
|-------------------------------|--|
| Test Case | Jika inputan data yang akan diprediksi tidak lengkap |
| Target yang diharapkan | menampilkan alert “Data tidak lengkap” |
| Hasil Pengujian | Benar |
| Path/jalur | 128-129-130-131-132-133-134-135-136-137-211 |

Tabel 4.26 Test case pengujian fungsi hitung jalur 2

| | |
|-------------------------------|--|
| Test Case | Jika data bulan sama dengan 0 |
| Target yang diharapkan | Data tahun dikurangi 1 disimpan ke dalam variabel temp_tahun dan menyimpan nilai 12 pada variable temp_bulan |
| Hasil Pengujian | Benar |
| Path/jalur | 128-129-130-131-132-133-138-139-141-142-143-144-146-147-148-149-151-152-153-154-155-156-157-158-210 |

Tabel 4.27 Test case pengujian fungsi get tahun hari jalur 3

| | |
|-------------------------------|---|
| Test Case | Jika jumlah list periode kurang dari 6 |
| Target yang diharapkan | Jalankan proses pengambilan data periode berikutnya |

| | |
|------------------------|---|
| Hasil Pengujian | Benar |
| Path/jalur | 128-129-130-131-132-133-138-139-141-142-143- 144-146-147-148-149-151-152-157-158-210 |

Tabel 4.28 Test case pengujian fungsi hitung jalur 4

| | |
|-------------------------------|---|
| Test Case | Jika ada data awal pada baris pertama |
| Target yang diharapkan | Jalankan inisialisasi data awal pada baris pertama |
| Hasil Pengujian | Benar |
| Path/jalur | 128-129-130-131-132-133-138-139-141-142-143- 144-146-147-148-149-160-161-163-164-165-166- 167-168-169-171-172-174-175-176-177-178-179- 180-181-182-183-180 |

Tabel 4.29 Test case pengujian fungsi hitung jalur 5

| | |
|-------------------------------|---|
| Test Case | Jika jumlah baris sama dengan jumlah dataset |
| Target yang diharapkan | Menghitung hasil prediksi akhir |
| Hasil Pengujian | Benar |
| Path/jalur | 128-129-130-131-132-133-138-139-141-142-143- 144-146-147-148-149-160-161-163-164-165-166- 167-168-169-171-172-174-175-176-177-178-179- 180-184-185-186-187-210 |

Tabel 4.30 Test case pengujian fungsi get tahun hari jalur 6

| | |
|-------------------------------|--|
| Test Case | Jika jumlah baris kurang dari jumlah dataset, maka menjalankan perintah else |
| Target yang diharapkan | Memulai prediksi untuk baris berikutnya |
| Hasil Pengujian | Benar |
| Path/jalur | 128-129-130-131-132-133-138-139-141-142-143- 144-146-147-148-149-160-161-163-164-165-166- |

167-168-169-171-172-174-175-176-189-190-191-
192-193-194-195-196-174

Tabel 4.31 Test case pengujian fungsi hitung jalur 7

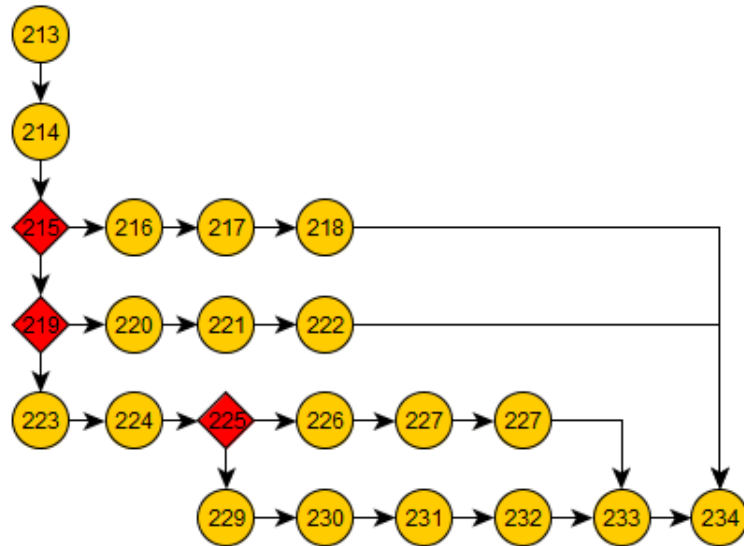
| | |
|-------------------------------|---|
| Test Case | Jika proses perhitungan prediksi selesai |
| Target yang diharapkan | Mencetak hasil prediksi |
| Hasil Pengujian | Benar |
| Path/jalur | 128-129-130-131-132-133-138-139-141-142-143- 144-146-147-148-149-160-161-163-164-165-166- 167-168-169-171-172-174-197-198-199-200-201- 202-203-204-205-206-207-208-209-210 |

```

217 public function hitunghari($bulan,$tahun)
218 {
219     if($bulan == 1 || $bulan == 2 || $bulan == 3 || $bulan == 4 || $bulan == 5 || $bulan == 6 || $bulan == 7 || $bulan == 8 || $bulan == 9 || $bulan == 10 || $bulan == 11 || $bulan == 12)
220     {
221         return 31;
222     }
223     else if($bulan == 4 || $bulan == 6 || $bulan == 9 || $bulan == 11)
224     {
225         return 30;
226     }
227     else
228     {
229         if($bulan == 2 && $tahun % 4 == 0)
230         {
231             return 29;
232         }
233         else
234         {
235             return 28;
236         }
237     }
238 }

```

Gambar 4.51 Kode program function hitunghari(\$bulan,\$tahun)



Gambar 4.52 Diagram alir function hitung()

$$CC = \text{EDGE} - \text{NODE} + 2$$

$$CC = 24 - 22 + 2$$

$$CC = 4$$

Maka jalur basis set pada pengujian di atas adalah 213-214-215-216-217-218-234, 213-214-215-219-220-221-222-234, 213-214-215-219-223-224-225-226-227-233-234 dan 213-214-215-219-223-224-225-229-230-231-232-233-234.

Pengujian kebenaran keempat jalur tersebut dapat dilihat pada Tabel 4.32 sampai 4.36

Tabel 4.32 Test case pengujian fungsi hitung hari jalur 1

| | |
|-------------------------------|---|
| Test Case | Jika data bulan yang akan diprediksi adalah Januari, Maret, Mei, Juli, Agustus, Oktober, dan Desember |
| Target yang diharapkan | Mengembalikan nilai hari yaitu 31 |
| Hasil Pengujian | Benar |
| Path/jalur | 213-214-215-216-217-218-234 |

Tabel 4.33 Test case pengujian fungsi hitung hari jalur 2

| | |
|-------------------------------|--|
| Test Case | Jika data bulan yang akan diprediksi adalah April, Juni, September, Oktober dan November |
| Target yang diharapkan | Mengembalikan nilai hari yaitu 30 |
| Hasil Pengujian | Benar |
| Path/jalur | 213-214-215-219-220-221-222-234 |

Tabel 4.34 Test case pengujian fungsi hitung hari jalur 3

| | |
|-------------------------------|---|
| Test Case | Jika data bulan yang akan diprediksi Februari dan nilai tahun habis dibagi dengan empat |
| Target yang diharapkan | Mengembalikan nilai hari yaitu 29 |
| Hasil Pengujian | Benar |
| Path/jalur | 213-214-215-219-223-224-225-226-227-233-234 |

Tabel 4.35 Test case pengujian fungsi hitung hari jalur 4

| | |
|-------------------------------|---|
| Test Case | Jika data bulan yang akan diprediksi Februari dan nilai tahun tidak habis dibagi dengan empat |
| Target yang diharapkan | Mengembalikan nilai hari yaitu 28 |
| Hasil Pengujian | Benar |
| Path/jalur | 213-214-215-219-223-224-225-229-230-231-232-233-234 |

BAB 6. PENUTUP

Pada bab ini merupakan bagian akhir di dalam penulisan skripsi, berisi tentang kesimpulan dan saran. Kesimpulan yang ditulis merupakan kesimpulan dari hasil penelitian yang telah dilakukan dan saran lanjutan untuk dilakukan pada penelitian selanjutnya

6.1 Kesimpulan

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

1. Perhitungan prediksi menggunakan metode double exponential smoothing Holt cocok digunakan dengan *sample uji* barang Beef Sausages yang memiliki pola data *trend* yaitu mengalami kenaikan tiap periodenya, didukung dengan hasil error yang cukup kecil menggunakan *MAPE* dengan rata-rata kurang dari 10 %
2. Pada penelitian ini menggunakan *sample uji* Beef Sausages 350 gram, didapatkan nilai peramalan terbaik dengan nilai α 0,7 dan γ 0,1. Akan tetapi tidak semua jenis barang yang akan diprediksi memiliki nilai error terkecil berada pada α 0,7 dan γ 0,1
3. Semakin besar nilai α yang digunakan akan diperoleh nilai *MAPE* yang lebih kecil. Sebaliknya semakin kecil nilai γ yang digunakan akan diperoleh nilai *MAPE* yang lebih kecil.
4. Menggunakan metode Economy Order Quantity menghasilkan perkiraan jumlah pemesanan yang ekonomis, jumlah persediaan pengaman dan batas stok minimal untuk melakukan pemesanan kembali sebagai informasi melakukan pengadaan persediaan.

6.2 Saran

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

1. Sistem pengendalian persediaan stok barang dapat dilengkapi dengan fitur data keuangan perusahaan dengan perincian biaya yang terkait sehingga dapat mengetahui keuntungan dan kerugian.
2. Menambah fitur perhitungan biaya persediaan untuk mengetahui besar penghematan biaya jika menggunakan metode EOQ
3. Disarankan untuk menggunakan metode lain untuk menciptakan perbandingan antar metode dan membuat sistem prediksi menggunakan platform android, iOS, atau windows phone.

DAFTAR PUSTAKA

- Makridakis. 1992. *Metode Dan Sistem Peramalan*. Jakarta: Erlangga.
- Rangkuti, Freddy. 2007. *Manajemen Persediaan Sistem Di Bidang Bisnis*. Jakarta: PT. Rajagrafindo Persada.
- Agus, Ristono. *Manajemen Persediaan*. Yogyakarta: Graha Ilmu.
- Herlina. 2007. *Manajemen Keuangan. Handout Mata Kuliah Manajemen Keuangan*. Universitas Kristen Maranatha, Bandung
- Sommerville, Ian. 2003. *Software Engineering (Rekayasa Perangkat Lunak)*. Jakarta : Erlangga.
- Noeryanti. 2012. *Sistem Pemulusan Eksponensial Dari Brown Dan Dari Holt Untuk Data Yang Memuat Trend*. Yogyakarta: Institut Sains & Teknologi AKPRIND. Akurat Jurnal Ilmiah Akuntansi Nomor 05 Tahun ke-2 Mei-Agustus 2011.
- Valerie, Carien. 2011. *Perbandingan Metode EOQ (Economic Order Quantity) dan JIT (Just In Time) terhadap Efisiensi Biaya Persediaan dan Kinerja Non-Keuangan (Studi Kasus Pada Pt Indoto Tirta Mulia)*. Jurnal Ilmiah Akuntansi Nomor 05 Tahun ke-2 Mei-Agustus 2011.
- Sugiyono. 2012. *Memahami Penelitian Kualitatif*. Bandung: ALFABETA
- Pressman, R. S. 2002. *Rekayasa Perangkat Lunak Pendekatan Praktisi*. Yogyakarta: Andi.
- Assauri, Sofjan. 2004. *Manajemen Produksi dan Operasi*. Lembaga Penerbit. Fakultas Ekonomi Universitas Indonesia.
- Gruendemann, B. J. Dan Billie F. 2005. *Buku Ajar Keperawatan Periopatif Volume 1 Prinsip*. Jakarta: EGC.

LAMPIRAN

A. Kode Program

A.1 Controller user

```

class C_User extends CI_Controller {

    public function __construct() {
        parent::__construct();
        $this->load->helper('url');
        $this->load->library('session');
        $this->load->model('m_login');
        $this->load->model('m_user');
        $isLoggedIn = $this->session->userdata('session_level');
        if ($isLoggedIn == null || $isLoggedIn != "admin") {
            redirect("/");
        }
    }

    public function index() {
        $session = $this->m_login->get_session_userid();
        $id = $session['session_userid'];
        $bantu['foto'] = $this->m_login->get_foto($id);
        $bantu['user'] = $this->m_user->data_user();
        $this->load->view('v_user', $bantu);
    }

    public function insert_user() {
        $config['upload_path'] = './img/';
        $config['allowed_types'] = 'gif|jpg|png';
        $this->load->library('upload', $config);
        $this->upload->do_upload();
        $upload_data = $this->upload->data();
        $data ['nama'] = $this->input->post('nama');
        $data ['noktp'] = $this->input->post('noktp');
        $data ['alamat'] = $this->input->post('alamat');
        $data ['notelp'] = $this->input->post('notelp');
        $data ['level'] = $this->input->post('level');
        $data ['status'] = $this->input->post('status');
        $data ['username'] = $this->input->post('username');
        $data ['password'] = $this->input->post('password');
        $data ['foto'] = $upload_data['file_name'];
        $data2['daftar_user'] = $this->m_user->data_user()->result();
        $input = true;
        foreach ($data2['daftar_user'] as $row) {
            if ($row->no_ktp == $data['noktp']) {
                $input = false;
                break;
            }
        }
        if ($input == false) {
            redirect(site_url('c_user/insert') . "?pesan=Gagaltambah");
        } else {
            $this->m_user->insert_user($data);
            redirect(site_url('c_user') . "?pesan=Suksestambah");
        }
    }

    public function update() {
        $session = $this->m_login->get_session_userid();
        $id = $session['session_userid'];
        $bantu['foto'] = $this->m_login->get_foto($id);
        $id = $this->input->get('id');
        $bantu['user'] = $this->m_user->ambil_data_user($id);
        $this->load->view('v_update_user', $bantu);
    }

    public function update_user() {
        $id = $this->input->get('id');
        $config['upload_path'] = './img/';
        $config['allowed_types'] = 'gif|jpg|png';
        $this->load->library('upload', $config);
        $this->upload->do_upload();
        $upload_data = $this->upload->data();
        $data ['nama'] = $this->input->post('nama');
        $data ['noktp'] = $this->input->post('noktp');
        $data ['alamat'] = $this->input->post('alamat');
        $data ['notelp'] = $this->input->post('notelp');
        $data ['level'] = $this->input->post('level');
        $data ['status'] = $this->input->post('status');
        $data ['username'] = $this->input->post('username');
        $data ['password'] = $this->input->post('password');
        $data ['foto'] = $upload_data['file_name'];
        if ($this->m_user->update_user($data, $id)) {
            redirect(site_url('c_user') . "?pesan=Suksestambah");
        }else{
            redirect(site_url('c_user/update') . "?pesan=Gagalubah");
        }
    }
}
}

```

A.2 Controller jenis

```

<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class C_Jenis extends CI_Controller {

    public function __construct() {
        parent::__construct();
        $this->load->helper('url');
        $this->load->library('session');
        $this->load->model('m_login');
        $this->load->model('m_jenis');
        $isLoggedIn = $this->session->userdata('session_level');
        if ($isLoggedIn == null || $isLoggedIn != "admin") {
            redirect("/");
        }
    }

    public function index() {
        $session = $this->m_login->get_session_userid();
        $id = $session['session_userid'];
        $bantu['foto'] = $this->m_login->get_foto($id);
        $bantu['jenis'] = $this->m_jenis->data_jenis();
        $this->load->view('v_jenis', $bantu);
    }

    public function insert() {
        $session = $this->m_login->get_session_userid();
        $id = $session['session_userid'];
        $bantu['foto'] = $this->m_login->get_foto($id);
        $this->load->view('v_insert_jenis', $bantu);
    }

    public function insert_jenis() {
        $bantu ['jenis'] = $this->input->post('jenis');
        $posisi1 = strtoupper(substr($bantu ['jenis'], 0, 2));
        $bantu['id_jenis'] = $this->m_jenis->get_id_jenis($posisi1);
        $bantu ['status'] = $this->input->post('status');
        $data2['daftar_jenis'] = $this->m_jenis->data_jenis()->result();
        $input = true;
        foreach ($data2['daftar_jenis'] as $row) {
            if ($row->nama_jenis == $bantu['jenis']) {
                $input = false;
                break;
            }
        }
        if ($input == false) {
            redirect(site_url('c_jenis/insert') . "?pesan=Gagaltambah");
        } else {
            $this->m_jenis->insert_jenis($bantu);
            redirect(site_url('c_jenis') . "?pesan=Suksestambah");
        }
    }

    public function update() {
        $session = $this->m_login->get_session_userid();
        $id = $session['session_userid'];
        $bantu['foto'] = $this->m_login->get_foto($id);
        $id = $this->input->get('id');
        $bantu['jenis'] = $this->m_jenis->ambil_data_jenis($id);
        $this->load->view('v_update_jenis', $bantu);
    }

    public function update_jenis() {
        $id = $this->input->get('id');
        $data ['jenis'] = $this->input->post('jenis');
        $data ['status'] = $this->input->post('status');
        if ($this->m_jenis->update_jenis($data, $id) {
            $this->session->set_userdata('pesan_sistem', 'Berhasil ubah data jenis !');
            $this->session->set_userdata('tipe_pesan', 'Suksestambah');
            redirect('c_jenis');
        } else {
            $this->session->set_userdata('pesan_sistem', 'Gagal ubah data jenis !');
            $this->session->set_userdata('tipe_pesan', 'Gagalubah');
            redirect('update');
        }
    }
}

/* End of file welcome.php */
/* Location: ./application/controllers/welcome.php */

```

A.3 Controller supplier

```

<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class C_Supplier extends CI_Controller {

    public function __construct() {
        parent::__construct();
        $this->load->helper('url');
        $this->load->library('session');
        $this->load->model('m_login');
        $this->load->model('m_supplier');
        $isLoggedIn = $this->session->userdata('session_level');
        if($isLoggedIn == null || $isLoggedIn != "admin"){
            redirect("/");
        }
    }

    public function index() {
        $session = $this->m_login->get_session_userid();
        $id = $session['session_userid'];
        $bantu['foto'] = $this->m_login->get_foto($id);
        $bantu['supplier'] = $this->m_supplier->data_supplier();
        $this->load->view('v_supplier', $bantu);
    }

    public function insert() {
        $session = $this->m_login->get_session_userid();
        $id = $session['session_userid'];
        $bantu['foto'] = $this->m_login->get_foto($id);
        $bantu['id_supplier'] = $this->m_supplier->get_id_supplier();
        $this->load->view('v_insert_supplier', $bantu);
    }

    public function insert_supplier() {
        $data ['id'] = $this->input->post('id');
        $data ['nama_supplier'] = $this->input->post('nama_supplier');
        $data ['alamat'] = $this->input->post('alamat');
        $data ['notelp'] = $this->input->post('notelp');
        $data ['status'] = $this->input->post('status');
        if ($this->m_supplier->insert_supplier($data)) {
            $this->session->set_userdata('pesan_sistem', 'Berhasil tambah data supplier !');
            $this->session->set_userdata('tipe_pesan', 'SuksesTambah');
            redirect('c_supplier');
        } else {
            $this->session->set_userdata('pesan_sistem', 'Gagal tambah data supplier !');
            $this->session->set_userdata('tipe_pesan', 'GagalTambah');
            redirect('c_supplier/insert');
        }
    }

    public function update() {
        $session = $this->m_login->get_session_userid();
        $id = $session['session_userid'];
        $bantu['foto'] = $this->m_login->get_foto($id);
        $id = $this->input->get('id');
        $bantu['supplier'] = $this->m_supplier->ambil_data_supplier($id);
        $this->load->view('v_update_supplier', $bantu);
    }

    public function update_supplier() {
        $data ['id'] = $this->input->post('id');
        $data ['nama_supplier'] = $this->input->post('nama_supplier');
        $data ['alamat'] = $this->input->post('alamat');
        $data ['notelp'] = $this->input->post('notelp');
        $data ['status'] = $this->input->post('status');
        if ($this->m_jenis->update_supplier($data, $id)) {
            $this->session->set_userdata('pesan_sistem', 'Berhasil ubah data supplier !');
            $this->session->set_userdata('tipe_pesan', 'Suksesubah');
            redirect('c_jenis');
        } else {
            $this->session->set_userdata('pesan_sistem', 'Gagal ubah data supplier !');
            $this->session->set_userdata('tipe_pesan', 'Gagalubah');
            redirect('c_jenis/update');
        }
    }
}
}

```

A.4 Controller barang

```

<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class C_Barang extends CI_Controller {

    public function __construct() {
        parent::__construct();
        $this->load->helper('url');
        $this->load->library('session');
        $this->load->model('m_login');
        $this->load->model('m_barang');
        $isLoggedIn = $this->session->userdata('session_level');
        if($isLoggedIn == null || $isLoggedIn != "admin"){
            redirect("/");
        }
    }

    public function index() {
        $session = $this->m_login->get_session_userid();
        $id = $session['session_userid'];
        $bantu['foto'] = $this->m_login->get_foto($id);
        $bantu['barang'] = $this->m_barang->data_barang();
        $this->load->view('v_barang', $bantu);
    }

    public function insert() {
        $session = $this->m_login->get_session_userid();
        $id = $session['session_userid'];
        $bantu['foto'] = $this->m_login->get_foto($id);
        $bantu['jenis'] = $this->m_barang->list_jenis();
        $bantu['supplier'] = $this->m_barang->list_supplier();
        $this->load->view('v_insert_barang', $bantu);
    }

    public function insert_barang() {
        $data ['nama_brng'] = $this->input->post('barang');
        $data ['jenis'] = $this->input->post('jenis');
        $data ['nama_splr'] = $this->input->post('supplier');
        $data ['harga_jual'] = $this->input->post('harga_jual');
        $data ['biaya_pmsnan'] = $this->input->post('biaya_pesanan');
        $data ['biaya_pympnan'] = $this->input->post('biaya_simpan');
        $data ['wktu_tgggu'] = $this->input->post('waktu_tunggu');
        $data ['status'] = $this->input->post('status');
        if ($this->m_barang->insert_barang($data) {
            $this->session->set_userdata('pesan_sistem', 'Berhasil tambah data barang !');
            $this->session->set_userdata('tipe_pesanan', 'Suksesstambah');
            redirect('c_barang');
        } else {
            $this->session->set_userdata('pesan_sistem', 'Gagal tambah data barang !');
            $this->session->set_userdata('tipe_pesanan', 'Gagaltambah');
            redirect('c_barang/insert');
        }
    }

    public function update() {
        $session = $this->m_login->get_session_userid();
        $id = $session['session_userid'];
        $bantu['foto'] = $this->m_login->get_foto($id);
        $id = $this->input->get('id');
        $bantu['jenis'] = $this->m_barang->list_jenis();
        $bantu['supplier'] = $this->m_barang->list_supplier();
        $bantu['barang'] = $this->m_barang->ambil_data_barang($id);
        $this->load->view('v_update_barang', $bantu);
    }

    public function update_barang() {
        $id = $this->input->get('id');
        $data ['nama_brng'] = $this->input->post('barang');
        $data ['jenis'] = $this->input->post('jenis');
        $data ['nama_splr'] = $this->input->post('supplier');
        $data ['harga_jual'] = $this->input->post('harga_jual');
        $data ['biaya_pmsnan'] = $this->input->post('biaya_pesanan');
        $data ['biaya_pympnan'] = $this->input->post('biaya_simpan');
        $data ['wktu_tgggu'] = $this->input->post('waktu_tunggu');
        $data ['status'] = $this->input->post('status');
        if ( $this->m_barang->update_barang($data, $id) {
            $this->session->set_userdata('pesan_sistem', 'Berhasil ubah data barang !');
            $this->session->set_userdata('tipe_pesanan', 'Suksesubah');
            redirect('c_barang');
        } else {
            $this->session->set_userdata('pesan_sistem', 'Gagal ubah data barang !');
            $this->session->set_userdata('tipe_pesanan', 'Gagalubah');
            redirect('c_barang/update');
        }
    }
}
}

```

A.5 Controller pembelian

```

<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class C_Pembelian extends CI_Controller {

    public function __construct() {
        parent::__construct();
        $this->load->helper('url');
        $this->load->library('session');
        $this->load->model('m_login');
        $this->load->model('m_pembelian');
        $isLoggedIn = $this->session->userdata('session_level');
        if($isLoggedIn == null || $isLoggedIn != "admin"){
            redirect("/");
        }
    }

    public function index() {
        $session = $this->m_login->get_session_userid();
        $id = $session['session_userid'];
        $bantu['foto'] = $this->m_login->get_foto($id);
        $bantu['stok'] = $this->m_pembelian->data_stok();
        $this->load->view('v_pembelian', $bantu);
    }

    public function detail($kadaluarsa) {
        $session = $this->m_login->get_session_userid();
        $id = $session['session_userid'];
        $bantu['foto'] = $this->m_login->get_foto($id);
        $datas = explode("oo", $kadaluarsa);
        $tgl = str_replace('%20', ' ', $datas[1]);
        $bantu['stokdetail'] = $this->m_pembelian->list_stok_detail($datas[0], $tgl);
        $this->load->view('v_detail_pembelian', $bantu);
    }

    public function insert() {
        $session = $this->m_login->get_session_userid();
        $id = $session['session_userid'];
        $bantu['foto'] = $this->m_login->get_foto($id);
        $bantu['barang'] = $this->m_pembelian->list_barang();
        $this->load->view('v_insert_pembelian', $bantu);
    }

    public function get_stock(){
        $id_produk = $this->input->post("id_produk");
        $tgl_kadaluarsa = $this->input->post("tgl_kadaluarsa");
        echo $this->m_pembelian->get_barang_by_tgl_kadaluarsa($id_produk, $tgl_kadaluarsa);
    }

    function simpan_pembelian() {
        if ($this->m_pembelian->simpan_pembelian()) {
            $this->session->set_userdata('pesan_sistem', 'Pembelian SUKSES!');
            $this->session->set_userdata('tipe_pesan', 'Sukses');
            redirect('c_pembelian');
        } else {
            $this->session->set_userdata('pesan_sistem', 'Maaf! Pembelian, GAGAL!');
            $this->session->set_userdata('tipe_pesan', 'Gagal');
            redirect('c_pembelian/insert');
        }
    }
}

```


A.6 Controller penjualan

```

class C_Penjualan extends CI_Controller {

    public function __construct() {
        parent::__construct();
        $this->load->helper('url');
        $this->load->library('session');
        $this->load->model('m_login');
        $this->load->model('m_penjualan');
        $isLoggedIn = $this->session->userdata('session_level');
        if($isLoggedIn == null || $isLoggedIn != "kasir"){
            redirect("/");
        }
    }

    public function index() {
        $session = $this->m_login->get_session_userid();
        $id = $session['session_userid'];
        $bantu['foto'] = $this->m_login->get_foto($id);
        $bantu['barang'] = $this->m_penjualan->list_barang();
        $this->load->view('v_penjualan', $bantu);
    }

    public function get_tgl_kadaluarsa(){
        $id_produk = $this->input->post("id_produk");
        echo $this->m_penjualan->kadaluarsa($id_produk);
    }

    function get_harga() {
        $id_produk = $this->input->post("id_produk");
        echo $this->m_penjualan->harga($id_produk);
    }

    function simpan_penjualan() {
        if ($this->m_penjualan->simpan_penjualan()) {
            $this->session->set_userdata('pesan_sistem', 'Pembelian SUKSES!');
            $this->session->set_userdata('tipe_pesan', 'Sukses');
            redirect('c_penjualan');
        } else {
            $this->session->set_userdata('pesan_sistem', 'Maaf! Pembelian, GAGAL!');
            $this->session->set_userdata('tipe_pesan', 'Gagal');
            redirect('c_penjualan');
        }
    }
}

```

A7. Controller retur

```

class C_Retur extends CI_Controller {

    public function __construct() {
        parent::__construct();
        $this->load->helper('url');
        $this->load->library('session');
        $this->load->model('m_login');
        $this->load->model('m_retur');
        $isLoggedIn = $this->session->userdata('session_level');
        if($isLoggedIn == null || $isLoggedIn != "admin"){
            redirect("/");
        }
    }

    public function index() {
        $session = $this->m_login->get_session_userid();
        $id = $session['session_userid'];
        $bantu['foto'] = $this->m_login->get_foto($id);
        $bantu['retur'] = $this->m_retur->data_retur();
        $this->load->view('v_retur', $bantu);
    }

    public function insert() {
        $session = $this->m_login->get_session_userid();
        $id = $session['session_userid'];
        $bantu['foto'] = $this->m_login->get_foto($id);
        $bantu['barang'] = $this->m_retur->list_barang();
        $this->load->view('v_insert_retur', $bantu);
    }
}

```

```

public function gettgl ($id){
    $result=$this->m_retur->gettglkadaluarsa($id);
    $HTML="";
    if($result->num_rows() > 0){
        foreach($result->result() as $list){
            $HTML.="<option value='".$list->tgl.'">".$list->tgl."</option>";
        }
    }
    echo $HTML;
}

public function get_stock($id_produk, $tgl_kadaluarsa){
    echo $this->m_retur->get_barang_by_tgl_kadaluarsa($id_produk, $tgl_kadaluarsa);
}

function simpan_retur() {
    if ($this->m_retur->simpan_retur()) {
        $this->session->set_userdata('pesan_sistem', 'Retur Barang SUKSES!');
        $this->session->set_userdata('tipe_pesan', 'Sukses');
        redirect('c_retur');
    } else {
        $this->session->set_userdata('pesan_sistem', 'Maaf! Retur Barang, GAGAL!');
        $this->session->set_userdata('tipe_pesan', 'Gagal');
        redirect('c_retur/insert');
    }
}
}

```

A7. Controller prediksi

```

class C_Prediksi extends CI_Controller {

    public function __construct() {
        parent::__construct();
        $this->load->helper('url');
        $this->load->library('session');
        $this->load->model('m_login');
        $this->load->model('m_prediksi');

        $isLoggedIn = $this->session->userdata('session_level');
        if ($isLoggedIn == null || $isLoggedIn != "admin") {
            redirect("/");
        }
    }

    public function index() {
        $session = $this->m_login->get_session_userid();
        $id = $session['session_userid'];
        $bantu['foto'] = $this->m_login->get_foto($id);

        $bantu['barang'] = $this->m_prediksi->list_barang();
        $this->load->view('v_prediksi', $bantu);
    }

    public function gettahun($id) {
        $result = $this->m_prediksi->gettahun($id);
        $total = $result->num_rows();
        $HTML = "";
        $lastMonth = "";
        $tahun = 0;
        if ($total > 0) {
            foreach ($result->result() as $list) {
                $tahun = intval($list->tahun);
                $lastMonth = intval($list->bulan);
                $HTML.="<option value='".$tahun.'" . $tahun . ">".$tahun . "</option>";
            }
            if ($lastMonth == 12) {
                $HTML.="<option value='".$($tahun + 1) . "'>".$($tahun + 1) . "</option>";
            }
        }
        echo $HTML;
    }

    public function getbulan($id, $tahun, $tahunPertama) {
        $namaBulan = ["Januari", "Februari", "Maret", "April", "Mei", "Juni", "Juli", "Agustus", "September", "Oktober",
        "November", "Desember"];
        $maks = 12;
        $temp_bulan = $maks;
        $temp_tahun = $tahun;
        $k = 0;
        $data = array();
        $cek = array();
    }
}

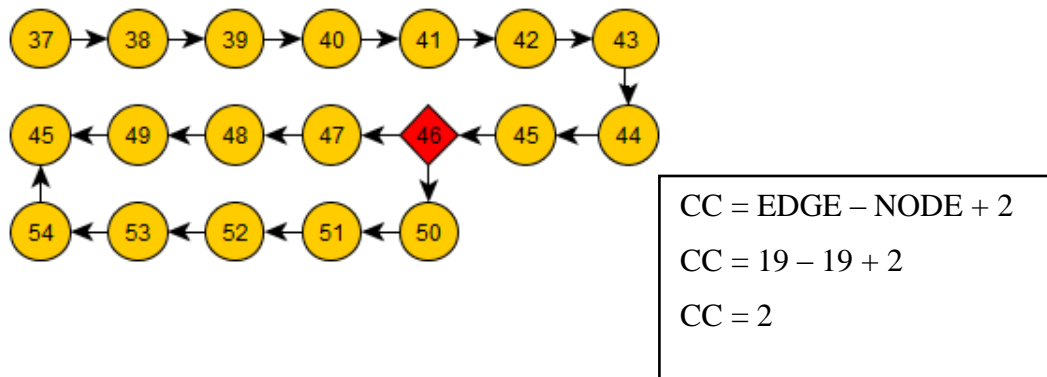
```

```

if($tahunPertama == 1)
{
    for($i=0;$i<6;$i++)
    {
        $temp_bulan -= 1;
        $data[$i]['bulan'] = $temp_bulan;
        $data[$i]['tahun'] = $temp_tahun;

        $temp_bulan_1 = $temp_bulan;
        $temp_tahun_1 = $temp_tahun;
        for($j=0;$j<6;$j++)
        {
            if($temp_bulan_1 == -1)
            {
                $temp_tahun_1 -= 1;
                $temp_bulan_1 = 11;
            }
            $cek[$j]['bulan'] = $temp_bulan_1;
            $cek[$j]['tahun'] = $temp_tahun_1;
            $temp_bulan_1 -= 1;
        }
        $res = $this->m_prediksi->cekbulan($cek,$id)->result_array();
        if(count($res) == 6)
        {
            $indeks[$k] = $data[$i]['bulan'];
            $k++;
        }
    }
}
else
{
    for($i=0;$i<12;$i++)
    {
        $temp_bulan -= 1;
        if($temp_bulan == -1)
        {
            $temp_tahun -= 1;
            $temp_bulan = 11;
        }
        $data[$i]['bulan'] = $temp_bulan;
        $data[$i]['tahun'] = $temp_tahun;
        $temp_bulan_1 = $temp_bulan;
        $temp_tahun_1 = $temp_tahun;
        for($j=0;$j<6;$j++)
        {
            if($temp_bulan_1 == 0)
            {
                $temp_tahun_1 -= 1;
                $temp_bulan_1 = 11;
            }
            $cek[$j]['bulan'] = $temp_bulan_1;
            $cek[$j]['tahun'] = $temp_tahun_1;
            $temp_bulan_1 -= 1;
        }
        $res = $this->m_prediksi->cekbulan($cek,$id)->result_array();
        if(count($res) == 6 && $data[$i]['tahun'] == $tahun)
        {
            $indeks[$k] = $data[$i]['bulan'];
            $k++;
        }
    }
}
sort($indeks);
$HTML = "";
foreach($indeks as $row)
{
    $HTML.="<option value='".$row . "'>". $namaBulan[$row] . "</option>";
}
echo $HTML;
}
public function hitung()
{
    $id_barang = $this->input->post('id_produk');
    $bulan = $this->input->post('bulan');
    $tahun = $this->input->post('tahun');
    if($id_barang == '' || $bulan == '' || $tahun == '')
    {
        echo "<script>alert('Data tidak lengkap!')</script>";
        redirect('c_prediksi','refresh');
    }
    else
    {
        // konstanta
        $maks = 6;
        $alpa = 0.1;
        $gamma = 0.2;
        $service_level = 1.645;
        // perhitungan periode yang akan dijadikan dataset
        $periode = array();
        $temp_bulan = $bulan;
        $temp_tahun = $tahun;
    }
}

```

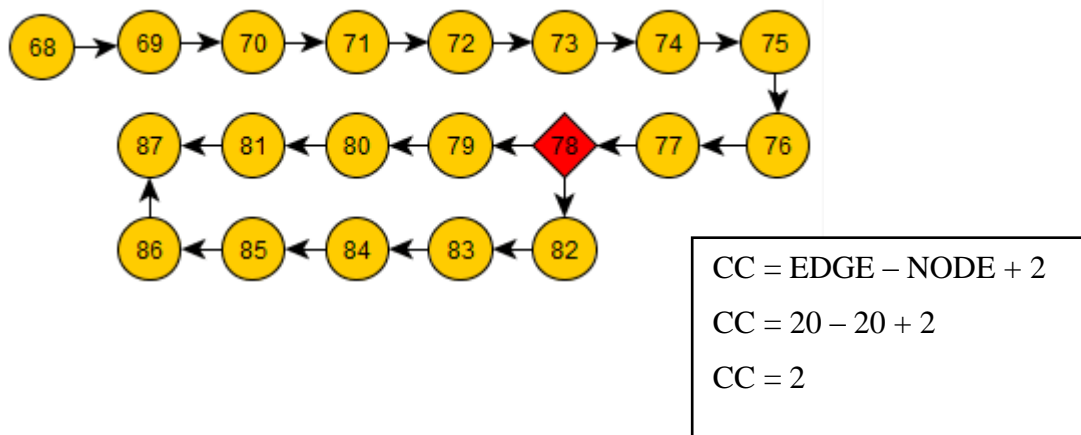



B.4 Edit Data Barang

```

05 public function update_barang() {
06     $id = $this->input->post('id');
07     $nama_barang = $this->input->post('barang');
08     $jenis = $this->input->post('jenis');
09     $nama_kolektor = $this->input->post('kolektor');
10     $harga_jual = $this->input->post('harga_jual');
11     $biaya_pemasok = $this->input->post('biaya_pemasok');
12     $biaya_promosi = $this->input->post('biaya_promosi');
13     $waktu_tinggi = $this->input->post('waktu_tinggi');
14     $status = $this->input->post('status');
15     $this->db->barang->update_barang($id, $nama_barang, $jenis, $nama_kolektor, $harga_jual, $biaya_pemasok, $biaya_promosi, $waktu_tinggi, $status);
16     $this->session->set_userdata('pesan_error', 'Berhasil ubah data barang !');
17     $this->session->set_userdata('type_pesan', 'Suksesubah');
18     redirect('index.php/barang');
19 } else {
20     $this->session->set_userdata('pesan_error', 'Gagal ubah data barang !');
21     $this->session->set_userdata('type_pesan', 'Gagalubah');
22     redirect('index.php/barang/update');
23 }
24 }

```

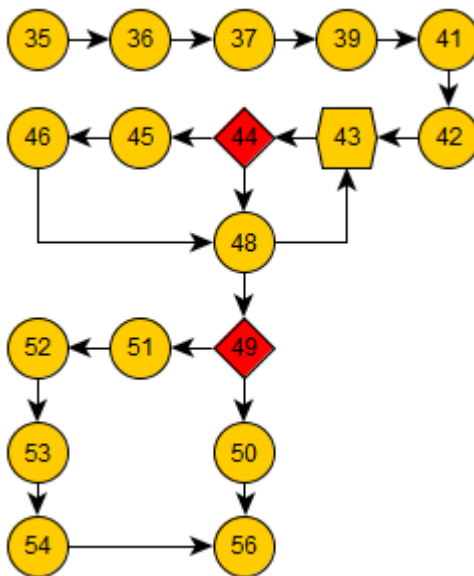


B.5 Insert Data Jenis

```

35     public function insert_jenis() {
36         $bantu ['jenis'] = $this->input->post('jenis');
37         $posisi1 = strtoupper(substr($bantu ['jenis'], 0, 2));
38         // $posisi2 = strpos($bantu ['jenis'], "2");
39         $bantu['id_jenis'] = $this->m_jenis->get_id_jenis($posisi1);
40         $bantu ['status'] = $this->input->post('status');
41         $data2['daftar_jenis'] = $this->m_jenis->data_jenis()->result();
42         $input = true;
43         foreach ($data2['daftar_jenis'] as $row) {
44             if ($row->nama_jenis == $bantu['jenis']) {
45                 $input = false;
46                 break;
47             }
48         }
49         if ($input == false) {
50             redirect(site_url('c_jenis/insert') . "?pesan=Gagaltambah");
51         } else {
52             $this->m_jenis->insert_jenis($bantu);
53             redirect(site_url('c_jenis') . "?pesan=Suksestambah");
54         }
55     }
56 }

```



$$CC = \text{EDGE} - \text{NODE} + 2$$

$$CC = 14 - 14 + 2$$

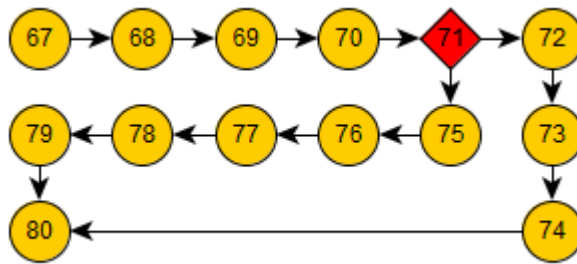
$$CC = 2$$

B.6 Edit Data Jenis

```

67     public function edit_data_jenis() {
68         $data = $this->input->post('data');
69         $jenis ['jenis'] = $this->input->post('jenis');
70         $status ['status'] = $this->input->post('status');
71         // $posisi2 = strpos($bantu ['jenis'], "2");
72         $bantu['id_jenis'] = $this->m_jenis->get_id_jenis($posisi1);
73         $bantu ['status'] = $this->input->post('status');
74         $data2['daftar_jenis'] = $this->m_jenis->data_jenis()->result();
75         $input = true;
76         foreach ($data2['daftar_jenis'] as $row) {
77             if ($row->nama_jenis == $bantu['jenis']) {
78                 $input = false;
79                 break;
80             }
81         }
82         if ($input == false) {
83             redirect(site_url('c_jenis/edit') . "?pesan=Gagaltambah");
84         } else {
85             $this->m_jenis->edit_data_jenis($bantu);
86             redirect(site_url('c_jenis') . "?pesan=Suksestambah");
87         }
88     }
89 }

```

$$CC = \text{EDGE} - \text{NODE} + 2$$

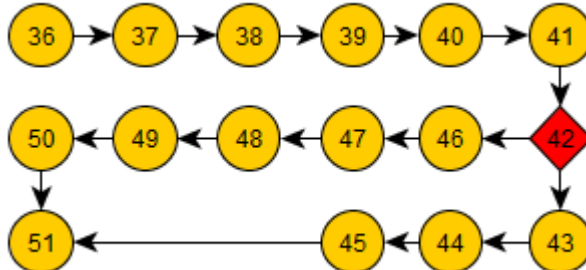
$$CC = 14 - 14 + 2$$

$$CC = 2$$

B.7 Insert Data Supplier

```

56 public function insert_supplier() {
57     $data['id'] = $this->input->post('id');
58     $data['nama_supplier'] = $this->input->post('nama_supplier');
59     $data['alamat'] = $this->input->post('alamat');
60     $data['no_telp'] = $this->input->post('no_telp');
61     $data['status'] = $this->input->post('status');
62     if ($this->supplier->insert_supplier($data)) {
63         $this->session->set_flashdata('pesan_sistem', 'Berhasil tambah data supplier !');
64         $this->session->set_flashdata('tipo_pesan', 'tambah');
65         redirect('t_supplier');
66     } else {
67         $this->session->set_flashdata('pesan_sistem', 'Gagal tambah data supplier !');
68         $this->session->set_flashdata('tipo_pesan', 'gagal');
69         redirect('t_supplier/insert');
70     }
71 }
  
```



$$CC = \text{EDGE} - \text{NODE} + 2$$

$$CC = 16 - 16 + 2$$

$$CC = 2$$

B.8 Edit Data Supplier

```

62 public function update_supplier() {
63     $data['id'] = $this->input->post('id');
64     $data['nama_supplier'] = $this->input->post('nama_supplier');
65     $data['alamat'] = $this->input->post('alamat');
66     $data['no_telp'] = $this->input->post('no_telp');
67     $data['status'] = $this->input->post('status');
68     if ($this->supplier->update_supplier($data, $id)) {
69         $this->session->set_flashdata('pesan_sistem', 'Berhasil ubah data supplier !');
70         $this->session->set_flashdata('tipo_pesan', 'ubah');
71         redirect('t_jenis');
72     } else {
73         $this->session->set_flashdata('pesan_sistem', 'Gagal ubah data supplier !');
74         $this->session->set_flashdata('tipo_pesan', 'gagal');
75         redirect('t_jenis/update');
76     }
77 }
  
```


B.11 Simpan Data Retur

```

52 void simpan_tahun() {
53     if (simpan_tahun() != 0) {
54         studi_penelitian_kegiatan_teknis_didatim, "Keturun Bangsa 1980-1981";
55         studi_penelitian_kegiatan_teknis_didatim, "1982-1983";
56         return;
57     }
58     studi_penelitian_kegiatan_teknis_didatim, "Keturun Bangsa 1984-1985";
59     studi_penelitian_kegiatan_teknis_didatim, "1986-1987";
60     return;
61 }

```



$$CC = EDGE - NODE + 2$$

$$CC = 11 - 11 + 2$$

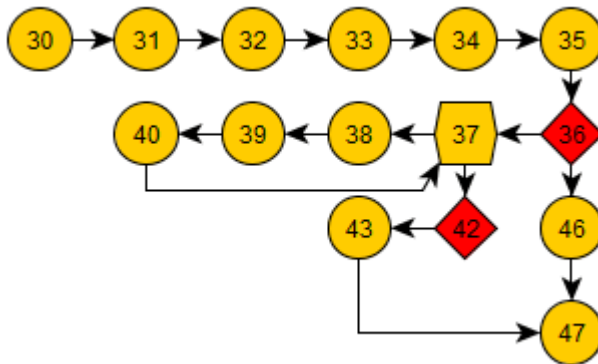
$$CC = 2$$

B.12 Mengambil data tahun

```

46 void simpan_data_tahun(int a) {
47     studi_penelitian_kegiatan_teknis_didatim(a);
48     studi_penelitian_kegiatan_teknis_didatim(a);
49     studi_penelitian_kegiatan_teknis_didatim(a);
50     studi_penelitian_kegiatan_teknis_didatim(a);
51     studi_penelitian_kegiatan_teknis_didatim(a);
52     studi_penelitian_kegiatan_teknis_didatim(a);
53     studi_penelitian_kegiatan_teknis_didatim(a);
54     studi_penelitian_kegiatan_teknis_didatim(a);
55     studi_penelitian_kegiatan_teknis_didatim(a);
56     studi_penelitian_kegiatan_teknis_didatim(a);
57     studi_penelitian_kegiatan_teknis_didatim(a);
58     studi_penelitian_kegiatan_teknis_didatim(a);
59     studi_penelitian_kegiatan_teknis_didatim(a);
60     studi_penelitian_kegiatan_teknis_didatim(a);
61     studi_penelitian_kegiatan_teknis_didatim(a);
62     studi_penelitian_kegiatan_teknis_didatim(a);
63     studi_penelitian_kegiatan_teknis_didatim(a);
64     studi_penelitian_kegiatan_teknis_didatim(a);
65     studi_penelitian_kegiatan_teknis_didatim(a);
66     studi_penelitian_kegiatan_teknis_didatim(a);
67     studi_penelitian_kegiatan_teknis_didatim(a);
68     studi_penelitian_kegiatan_teknis_didatim(a);
69     studi_penelitian_kegiatan_teknis_didatim(a);
70     studi_penelitian_kegiatan_teknis_didatim(a);
71     studi_penelitian_kegiatan_teknis_didatim(a);
72     studi_penelitian_kegiatan_teknis_didatim(a);
73     studi_penelitian_kegiatan_teknis_didatim(a);
74     studi_penelitian_kegiatan_teknis_didatim(a);
75     studi_penelitian_kegiatan_teknis_didatim(a);
76     studi_penelitian_kegiatan_teknis_didatim(a);
77     studi_penelitian_kegiatan_teknis_didatim(a);
78     studi_penelitian_kegiatan_teknis_didatim(a);
79     studi_penelitian_kegiatan_teknis_didatim(a);
80     studi_penelitian_kegiatan_teknis_didatim(a);
81     studi_penelitian_kegiatan_teknis_didatim(a);
82     studi_penelitian_kegiatan_teknis_didatim(a);
83     studi_penelitian_kegiatan_teknis_didatim(a);
84     studi_penelitian_kegiatan_teknis_didatim(a);
85     studi_penelitian_kegiatan_teknis_didatim(a);
86     studi_penelitian_kegiatan_teknis_didatim(a);
87     studi_penelitian_kegiatan_teknis_didatim(a);
88     studi_penelitian_kegiatan_teknis_didatim(a);
89     studi_penelitian_kegiatan_teknis_didatim(a);
90     studi_penelitian_kegiatan_teknis_didatim(a);
91     studi_penelitian_kegiatan_teknis_didatim(a);
92     studi_penelitian_kegiatan_teknis_didatim(a);
93     studi_penelitian_kegiatan_teknis_didatim(a);
94     studi_penelitian_kegiatan_teknis_didatim(a);
95     studi_penelitian_kegiatan_teknis_didatim(a);
96     studi_penelitian_kegiatan_teknis_didatim(a);
97     studi_penelitian_kegiatan_teknis_didatim(a);
98     studi_penelitian_kegiatan_teknis_didatim(a);
99     studi_penelitian_kegiatan_teknis_didatim(a);
100    studi_penelitian_kegiatan_teknis_didatim(a);

```



$$CC = EDGE - NODE + 2$$

$$CC = 16 - 15 + 2$$

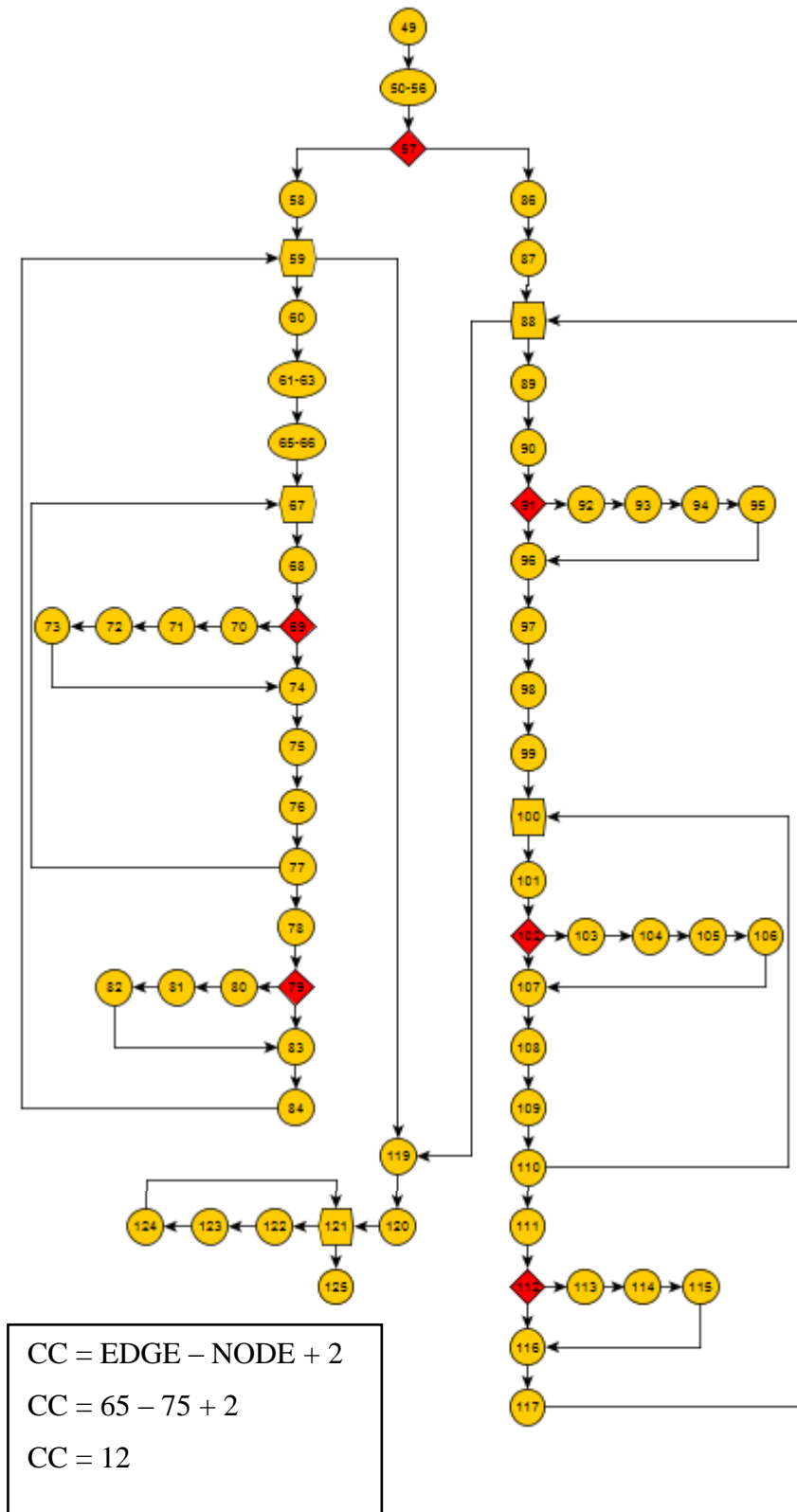
$$CC = 3$$

B.13 Mengambil data bulan

```

85     else
86     {
87         run($i);
88     }
89     {
90         $loop_bulan = 1;
91         if($loop_bulan == 1)
92         {
93             $loop_tahun = 1;
94             $loop_sahun = 1;
95         }
96         $data[$i]["bulan"] = $loop_bulan;
97         $data[$i]["tahun"] = $loop_tahun;
98         $loop_bulan_1 = $loop_bulan;
99         $loop_tahun_1 = $loop_tahun;
100         run($i,$j,$k);
101     }
102     {
103         if($loop_bulan_1 == 0)
104         {
105             $loop_tahun_1 = 1;
106             $loop_sahun_1 = 1;
107         }
108         $data[$j]["tahun"] = $loop_bulan_1;
109         $data[$j]["tahun"] = $loop_tahun_1;
110         $loop_bulan_1 = 1;
111     }
112     $res = $this->get_sudut_kal($tabel,$col,$id) (result_array());
113     if(count($res) == 0 && $data[$i]["tahun"] == $tahun)
114     {
115         $induk[$k] = $data[$i]["bulan"];
116         $k++;
117     }
118 }
119 }
120 $arr($success);
121 $HTML = "";
122 foreach($induk as $row)
123 {
124     $HTML = "option value=" . ($row[1] . "0" . $indukBulan[$row] . "option");
125 }
126 echo $HTML;
127 }
128 }
129 }
130 }
131 }
132 }
133 }
134 }
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }
143 }
144 }
145 }
146 }
147 }
148 }
149 }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }
159 }
160 }
161 }
162 }
163 }
164 }
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
201 }
202 }
203 }
204 }
205 }
206 }
207 }
208 }
209 }
210 }
211 }
212 }
213 }
214 }
215 }
216 }
217 }
218 }
219 }
220 }
221 }
222 }
223 }
224 }
225 }
226 }
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
236 }
237 }
238 }
239 }
240 }
241 }
242 }
243 }
244 }
245 }
246 }
247 }
248 }
249 }
250 }
251 }
252 }
253 }
254 }
255 }
256 }
257 }
258 }
259 }
260 }
261 }
262 }
263 }
264 }
265 }
266 }
267 }
268 }
269 }
270 }
271 }
272 }
273 }
274 }
275 }
276 }
277 }
278 }
279 }
280 }
281 }
282 }
283 }
284 }
285 }
286 }
287 }
288 }
289 }
290 }
291 }
292 }
293 }
294 }
295 }
296 }
297 }
298 }
299 }
300 }
301 }
302 }
303 }
304 }
305 }
306 }
307 }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322 }
323 }
324 }
325 }
326 }
327 }
328 }
329 }
330 }
331 }
332 }
333 }
334 }
335 }
336 }
337 }
338 }
339 }
340 }
341 }
342 }
343 }
344 }
345 }
346 }
347 }
348 }
349 }
350 }
351 }
352 }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361 }
362 }
363 }
364 }
365 }
366 }
367 }
368 }
369 }
370 }
371 }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 }
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
401 }
402 }
403 }
404 }
405 }
406 }
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
416 }
417 }
418 }
419 }
420 }
421 }
422 }
423 }
424 }
425 }
426 }
427 }
428 }
429 }
430 }
431 }
432 }
433 }
434 }
435 }
436 }
437 }
438 }
439 }
440 }
441 }
442 }
443 }
444 }
445 }
446 }
447 }
448 }
449 }
450 }
451 }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 }
460 }
461 }
462 }
463 }
464 }
465 }
466 }
467 }
468 }
469 }
470 }
471 }
472 }
473 }
474 }
475 }
476 }
477 }
478 }
479 }
480 }
481 }
482 }
483 }
484 }
485 }
486 }
487 }
488 }
489 }
490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
590 }
591 }
592 }
593 }
594 }
595 }
596 }
597 }
598 }
599 }
600 }
601 }
602 }
603 }
604 }
605 }
606 }
607 }
608 }
609 }
610 }
611 }
612 }
613 }
614 }
615 }
616 }
617 }
618 }
619 }
620 }
621 }
622 }
623 }
624 }
625 }
626 }
627 }
628 }
629 }
630 }
631 }
632 }
633 }
634 }
635 }
636 }
637 }
638 }
639 }
640 }
641 }
642 }
643 }
644 }
645 }
646 }
647 }
648 }
649 }
650 }
651 }
652 }
653 }
654 }
655 }
656 }
657 }
658 }
659 }
660 }
661 }
662 }
663 }
664 }
665 }
666 }
667 }
668 }
669 }
670 }
671 }
672 }
673 }
674 }
675 }
676 }
677 }
678 }
679 }
680 }
681 }
682 }
683 }
684 }
685 }
686 }
687 }
688 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
700 }
701 }
702 }
703 }
704 }
705 }
706 }
707 }
708 }
709 }
710 }
711 }
712 }
713 }
714 }
715 }
716 }
717 }
718 }
719 }
720 }
721 }
722 }
723 }
724 }
725 }
726 }
727 }
728 }
729 }
730 }
731 }
732 }
733 }
734 }
735 }
736 }
737 }
738 }
739 }
740 }
741 }
742 }
743 }
744 }
745 }
746 }
747 }
748 }
749 }
750 }
751 }
752 }
753 }
754 }
755 }
756 }
757 }
758 }
759 }
760 }
761 }
762 }
763 }
764 }
765 }
766 }
767 }
768 }
769 }
770 }
771 }
772 }
773 }
774 }
775 }
776 }
777 }
778 }
779 }
780 }
781 }
782 }
783 }
784 }
785 }
786 }
787 }
788 }
789 }
790 }
791 }
792 }
793 }
794 }
795 }
796 }
797 }
798 }
799 }
800 }
801 }
802 }
803 }
804 }
805 }
806 }
807 }
808 }
809 }
810 }
811 }
812 }
813 }
814 }
815 }
816 }
817 }
818 }
819 }
820 }
821 }
822 }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
830 }
831 }
832 }
833 }
834 }
835 }
836 }
837 }
838 }
839 }
840 }
841 }
842 }
843 }
844 }
845 }
846 }
847 }
848 }
849 }
850 }
851 }
852 }
853 }
854 }
855 }
856 }
857 }
858 }
859 }
860 }
861 }
862 }
863 }
864 }
865 }
866 }
867 }
868 }
869 }
870 }
871 }
872 }
873 }
874 }
875 }
876 }
877 }
878 }
879 }
880 }
881 }
882 }
883 }
884 }
885 }
886 }
887 }
888 }
889 }
890 }
891 }
892 }
893 }
894 }
895 }
896 }
897 }
898 }
899 }
900 }
901 }
902 }
903 }
904 }
905 }
906 }
907 }
908 }
909 }
910 }
911 }
912 }
913 }
914 }
915 }
916 }
917 }
918 }
919 }
920 }
921 }
922 }
923 }
924 }
925 }
926 }
927 }
928 }
929 }
930 }
931 }
932 }
933 }
934 }
935 }
936 }
937 }
938 }
939 }
940 }
941 }
942 }
943 }
944 }
945 }
946 }
947 }
948 }
949 }
950 }
951 }
952 }
953 }
954 }
955 }
956 }
957 }
958 }
959 }
960 }
961 }
962 }
963 }
964 }
965 }
966 }
967 }
968 }
969 }
970 }
971 }
972 }
973 }
974 }
975 }
976 }
977 }
978 }
979 }
980 }
981 }
982 }
983 }
984 }
985 }
986 }
987 }
988 }
989 }
990 }
991 }
992 }
993 }
994 }
995 }
996 }
997 }
998 }
999 }
1000 }

```



C. Pengujian *Black Box*

| No | Menu | Fungsi | Kasus | Hasil | Ket |
|----|------|--|--|--|-----|
| 1 | User | Menu ini memiliki fitur untuk menginsert, mengedit, dan melihat data user. | Ketika admin memilih user dan terdapat data user yang tersimpan | Menampilkan data user dalam list tabel | OK |
| | | | Ketika admin mengklik button insert | Menampilkan form insert user | OK |
| | | | Ketika admin mengklik button save sebelum mengisi seluruh field | Menampilkan pesan disebelah kolom yang belum terisi "Please fill out this field" | OK |
| | | | Ketika admin mengklik button save, inputan nomor ktp tidak valid (data yang diinputkan sama yang sudah ada dalam | Menampilkan pesan "KTP sudah terdaftar !" | OK |

| | | | |
|--|---|--|----|
| | | database) | |
| | Ketika admin mengklik button save dan seluruh field sudah terisi dan data valid | Menyimpan inputan data ke dalam database , menampilkan pesan “Berhasil tambah data user !” dan kembali halaman data user | OK |
| | Ketika admin mengklik button cancel pada form insert user | Menampilkan halaman data user | OK |
| | Ketika admin mengklik button edit | Menampilkan form edit user | OK |
| | Ketika admin mengklik button save sebelum mengisi seluruh field | Menampilkan pesan disebelah kolom yang belum terisi “Please fill out this field” | OK |
| | Ketika admin mengklik button save, inputan tidak valid | Menampilkan pesan “Gagal ubah data user!” | OK |

| | | | | | | |
|---|-------|---|--|---|---|----|
| | | | | Ketika admin mengklik button save dan seluruh field sudah terisi dan data valid | Mengubah inputan data ke dalam database , menampilkan pesan “Berhasil ubah data user !” dan kembali halaman data user | OK |
| 2 | Jenis | Menu ini memiliki fitur untuk menginsert, mengedit, dan melihat jenis | | Ketika admin memilih jenis terdapat data jenis yang tersimpan | Menampilkan data jenis dalam list tabel | OK |
| | | | | Ketika admin mengklik button insert | Menampilkan form insert jenis | OK |
| | | | | Ketika admin mengklik button save sebelum mengisi seluruh field | Menampilkan pesan disebelah kolom yang belum terisi “Please fill out this field” | OK |
| | | | | Ketika admin menyimpan data jenis telah | Menampilkan pesan “Gagal tambah data jenis !” | OK |

| | | | |
|--|---|--|----|
| | | diinputkan dan data tidak valid | |
| | Ketika admin mengklik button save dan seluruh field sudah terisi dan data valid | Menyimpan inputan data ke dalam database , menampilkan pesan “Berhasil tambah data jenis !” dan kembali halaman data jenis | OK |
| | Ketika admin mengklik button cancel pada form insert jenis | Menampilkan halaman data jenis | OK |
| | Ketika admin mengklik button edit | Menampilkan form edit jenis | OK |
| | Ketika admin mengklik button save sebelum mengisi seluruh field | Menampilkan pesan disebelah kolom yang belum terisi “Please fill out this field” | OK |
| | Ketika admin mengubah data jenis yang | Menampilkan pesan “Gagal ubah data jenis | OK |

| | | | | | | | | |
|---|----------|----------|---|---|---|---|----|--|
| | | | | telah diinputkan dan data tidak valid | !” | | | |
| | | | | Ketika admin mengklik button save dan seluruh field sudah terisi dan data valid | Mengubah inputan data ke dalam database , menampilkan pesan “Berhasil ubah data jenis !” dan kembali halaman data jenis | OK | | |
| 3 | Supplier | Menu ini | memiliki fitur untuk menginsert, mengedit, dan melihat data supplier | memilih menu supplier dan terdapat data supplier yang tersimpan | admin | Menampilkan data supplier dalam list tabel | OK | |
| | | | | Ketika admin mengklik button insert | admin | Menampilkan form insert supplier | OK | |
| | | | | Ketika admin mengklik button save sebelum mengisi seluruh field | admin | Menampilkan pesan disebelah kolom yang belum terisi “Please fill out this field” | OK | |
| | | | | Ketika admin | admin | Menampilkan | OK | |

| | | | |
|---|--|----|--|
| menyimpan data suplieryang telah diinputkan dan data tidak valid | pesan “Gagal tambah data supplier !” | | |
| Ketika admin mengklik button save dan seluruh field sudah terisi dan data valid | Menyimpan inputan data ke dalam database , dan menampilkan pesan “Berhasil tambah data supplier !” dan kembali halaman data supplier | OK | |
| Ketika admin mengklik button cancel pada form insert supplier | Menampilkan halaman data supplier | OK | |
| Ketika admin mengklik button edit | Menampilkan form edit supplier | OK | |
| Ketika admin mengklik button save sebelum mengisi | Menampilkan pesan disebelah kolom yang belum terisi “Please fill out | OK | |

| | | | | | | |
|---|--------|---|-----|---|---|----|
| | | | | seluruh field | this field” | |
| | | | | Ketika admin mengubah data supplier yang telah diinputkan dan data tidak valid | Menampilkan pesan “Gagal ubah data supplier !” | OK |
| | | | | Ketika admin mengklik button save dan seluruh field sudah terisi dan data valid | Mengubah inputan data ke dalam database , dan seluruh field sudah terisi dan data valid | OK |
| 4 | Barang | Menu ini memiliki fitur untuk menginsert, mengedit, dan melihat data supplier | ini | Ketika admin memilih menu barang dan terdapat data barang yang tersimpan | Menampilkan data barang dalam list tabel | OK |
| | | | | Ketika admin mengklik button insert | Menampilkan form insert barang | OK |
| | | | | Ketika admin mengklik | Menampilkan pesan disebelah | OK |

| | | | |
|------------------|------------------|----|--|
| button save | kolom yang | | |
| sebelum | belum terisi | | |
| mengisi | “Please fill out | | |
| seluruh field | this field” | | |
| Ketika admin | Menampilkan | OK | |
| menyimpan | pesan “Gagal | | |
| data barang | tambah data | | |
| yang telah | barang !” | | |
| diinputkan dan | | | |
| data tidak valid | | | |
| Ketika admin | Menyimpan | OK | |
| mengklik | inputan data ke | | |
| button save | dalam database , | | |
| dan seluruh | menampilkan | | |
| field sudah | pesan “Berhasil | | |
| terisi dan data | tambah data | | |
| valid | barang !” dan | | |
| | kembali | | |
| | halaman data | | |
| | barang | | |
| Ketika admin | Menampilkan | OK | |
| mengklik | halaman data | | |
| button cancel | barang | | |
| pada form | | | |
| insert barang | | | |
| Ketika admin | Menampilkan | OK | |
| mengklik | form edit barang | | |
| button edit | | | |
| Ketika admin | Menampilkan | OK | |

| | | | | | | |
|---|----------------------|---|--|---|--|----|
| | | | | mengklik button save sebelum mengisi seluruh field | pesan disebelah kolom yang belum terisi “Please fill out this field” | |
| | | | | Ketika admin mengubah data barang yang telah diinputkan dan data tidak valid | Menampilkan pesan “Gagal ubah data barang !” | OK |
| | | | | Ketika admin mengklik button save dan seluruh field sudah terisi dan data valid | Mengubah inputan data ke dalam database , menampilkan pesan “Berhasil ubah data barang !” dan kembali halaman data barang | OK |
| 5 | Persediaan Barang | Menu ini memiliki fitur untuk menginsert dan melihat data persediaan barang | | Ketika admin memilih menu persediaan barang dan terdapat data persediaan barang yang tersimpan | Menampilkan data persediaan barang dalam list tabel | OK |

| | | | |
|--|---|-----------------------------------|----|
| Ketika admin mengklik button detail pada persediaan barang | Menampilkan data transaksi masuk nama dan kadaluarsa | detail stok sesuai barang tanggal | OK |
| Ketika admin mengklik button insert | Menampilkan form insert stok masuk | data | OK |
| Ketika admin mengklik button inputan stok tidak valid (jumlah stok keluar = 0) | Menampilkan pesan "Jumlah stok tidak boleh kosong !" (jumlah stok keluar = 0) | add, stok masuk kosong !" | OK |
| Ketika admin mengklik button add sebelum mengisi seluruh field | Menampilkan alert "Data tidak lengkap !" sebelum mengisi seluruh field | add lengkap !" | OK |
| Ketika admin mengklik button add setelah seluruh field terisi dan | Menjalankan java script dan menampilkan list data transaksi stok | java script dan data stok | OK |

| | | | | | | | |
|---|--------------|---|--|--|--|--|----|
| | | | | data valid | masuk | | |
| | | | | Ketika admin mengklik simpsan | Menyimpan inputan data ke dalam database , menampilkan pesan “Pembelian SUKSES !” dan kembali halaman data persediaan barang | OK | |
| 6 | Retur Barang | Menu ini memiliki fitur untuk menginsert dan melihat retur barang | Menu ini memiliki fitur untuk menginsert dan melihat data retur barang | Ketika admin memilih retur dan data barang tersimpan | admin menu barang terdapat retur yang | Menampilkan data retur barang dalam list tabel | OK |
| | | | | Ketika admin mengklik button insert | Menampilkan form insert data retur barang | OK | |
| | | | | Ketika admin mengklik button add, inputan stok tidak | admin pesan “Jumlah stok keluar tidak boleh kosong !” keluar valid | OK | |

| | | | | | | |
|---|-------------------|---|-----|--|--|----|
| | | | | (jumlah stok keluar = 0) | | |
| | | | | Ketika admin mengklik button add sebelum mengisi seluruh field | Menampilkan alert “Data tidak lengkap !” | OK |
| | | | | Ketika admin mengklik button add setelah seluruh field terisi dan data valid | Menjalankan java script dan menampilkan list data transaksi retur barang | OK |
| | | | | Ketika admin mengklik simpan | Menyimpan inputan data ke dalam database , menampilkan pesan “Retur Barang SUKSES !” dan kembali halaman data retur barang | OK |
| 8 | Laporan Penjualan | Menu ini memiliki fitur untuk melihat laporan | ini | Ketika admin memilih laporan penjualan | Menampilkan halaman laporan penjualan | OK |

| penjualan | | | |
|-----------|----------|--|---|
| | | Ketika admin mengklik button view sebelum mengisi seluruh field | Menampilkan alert “Data tidak lengkap !” OK |
| | | Ketika admin mengklik button view seluruh field sudah terisi , data valid dan tidak terdapat data penjualan yang tersimpan | Menampilkan pesan “no data available in table” OK |
| | | Ketika admin mengklik button view seluruh field sudah terisi , data valid dan terdapat data penjualan yang tersimpan | Menampilkan laporan sesuai barang dan interval laporan OK |
| 8 | Prediksi | Menu ini memiliki manajemen data prediksi | Ketika admin memilih menu prediksi Menampilkan halaman form prediksi OK |

| | | | | | | | |
|---|---------------------|--|-----|---|--|--|--|
| | | persediaan barang | | | | | |
| | | | | Ketika admin mengklik button hitung view sebelum mengisi seluruh field | Kembali ke OK form prediksi dan tidak melakukan perhitungan prediksi | | |
| | | | | Ketika admin mengklik button hitung seluruh field sudah terisi , data valid | Menampilkan OK hasil prediksi dan grafik perbandingan data aktual dengan data prediksi | | |
| 6 | Transaksi Penjualan | Menu ini memiliki fitur untuk menginsert transaksi penjualan | ini | Ketika kasir memilih transaksi penjualan | Menampilkan OK form insert transaksi penjualan | | |
| | | | | Ketika admin mengklik button add, inputan jumlah stok keluar tidak valid (jumlah stok keluar = 0) | Menampilkan OK pesan “Jumlah stok keluar tidak boleh kosong !” | | |

| | | | |
|--|--|---|----|
| Ketika mengklik button sebelum mengisi seluruh field | kasir add | Menampilkan alert “Data tidak lengkap !” | OK |
| Ketika mengklik button setelah field data valid | kasir add seluruh data terisi dan data valid | Menjalankan java script dan menampilkan list data transaksi penjualan | OK |
| Ketika mengklik simpan | kasir | Menyimpan inputan data ke dalam database , menampilkan pesan “Penjualan SUKSES !” | OK |