



**SISTEM INFORMASI PERAMALAN PENJUALAN PROL TAPE DAN
PIA TAPE MENGGUNAKAN METODE *WINTER EXPONENTIAL
SMOOTHING*
(STUDI KASUS PUSAT OLEH-OLEH PURNAMA JATI)**

SKRIPSI

Oleh

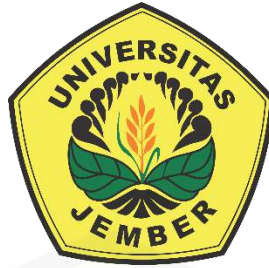
Putri Damayanti

132410101090

Program Studi Sistem Informasi

Universitas Jember

2017



**SISTEM INFORMASI PERAMALAN PENJUALAN PROL TAPE DAN
PIA TAPE MENGGUNAKAN METODE *WINTER EXPONENTIAL
SMOOTHING*
(STUDI KASUS PUSAT OLEH-OLEH PURNAMA JATI)**

SKRIPSI

Diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan pendidikan di Program Studi Sistem Informasi Universitas
Jember dan mendapat gelar Sarjana Sistem Informasi

Oleh

Putri Damayanti

132410101090

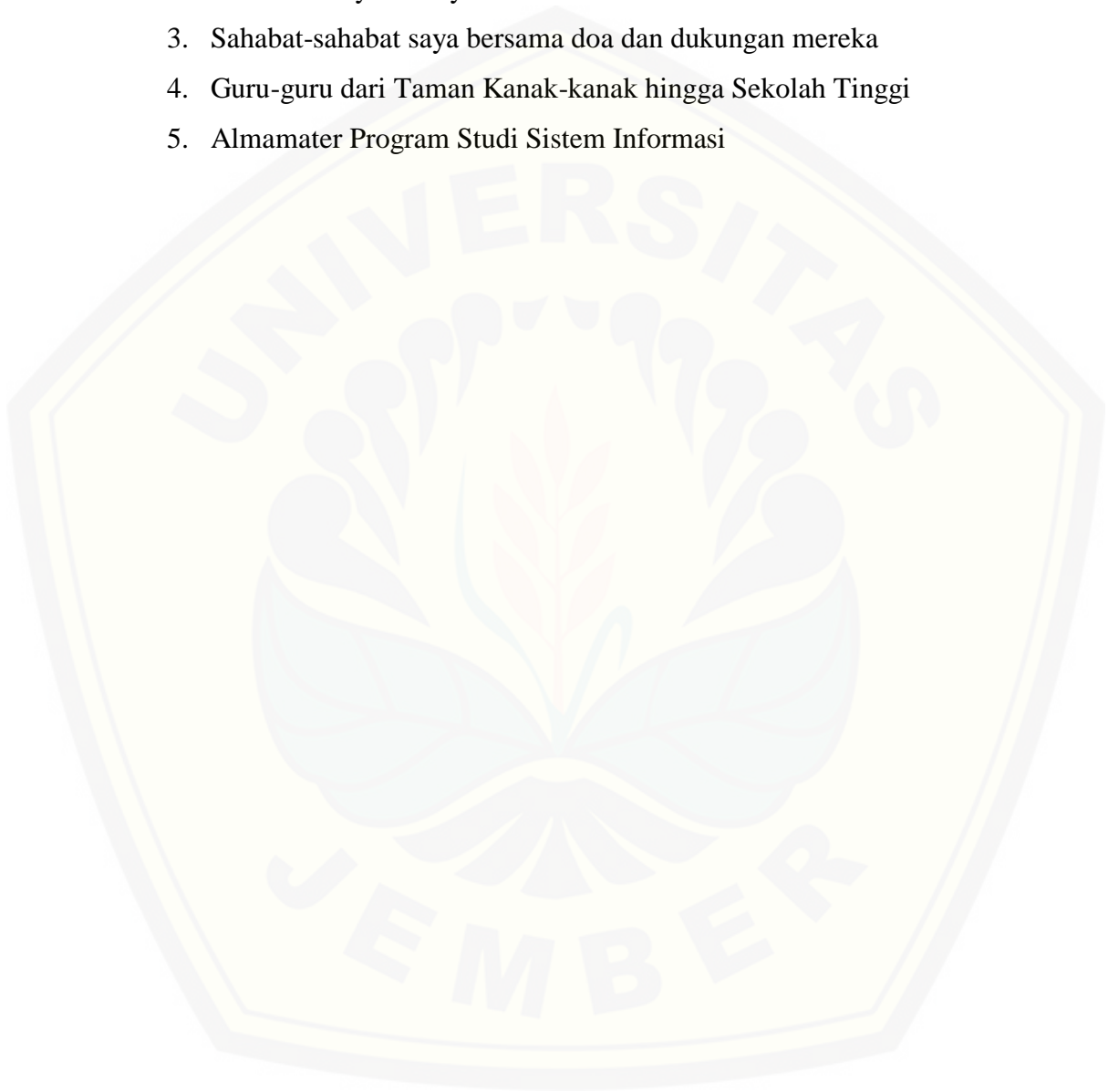
Program Studi Sistem Informasi

Universitas Jember

2017

PERSEMBAHAN

1. Ayah Sutrisno dan Ibu Indrayani
2. Adik Dwi Ayu Noviyanti
3. Sahabat-sahabat saya bersama doa dan dukungan mereka
4. Guru-guru dari Taman Kanak-kanak hingga Sekolah Tinggi
5. Almamater Program Studi Sistem Informasi



MOTO

"Mereka berkata bahwa setiap orang membutuhkan tiga hal yang akan membuat mereka berbahagia di dunia ini, yaitu; seseorang untuk dicintai, sesuatu untuk dilakukan, dan sesuatu untuk diharapkan."

(Tom Bodett)



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Putri Damayanti

NIM : 132410101090

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Informasi Peramalan Penjualan Prol Tape dan Pia Tape menggunakan Metode *Winter Exponential Smoothing* (Studi Kasus Pusat Oleh-Oleh Purnama Jati)”, adalah hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada instansi 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 kemudian hari dinyatakan tidak benar.

Jember, 18 Mei 2017

Yang menyatakan,

Putri Damayanti

NIM 132410101090

SKRIPSI

**SISTEM INFORMASI PERAMALAN PENJUALAN PROL TAPE DAN
PIA TAPE MENGGUNAKAN METODE *WINTER EXPONENTIAL
SMOOTHING*
(STUDI KASUS PUSAT OLEH-OLEH PURNAMA JATI)**

Oleh

Putri Damayanti

132410101090

Pembimbing

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

Dosen Pembimbing Anggota : Fahrobby Adnan, S.Kom., MMSI.

PENGESAHAN PEMBIMBING

Skripsi berjudul “Sistem Informasi Peramalan Penjualan Prol Tape dan Pia Tape menggunakan Metode *Winter Exponential Smoothing* (Studi Kasus Pusat Oleh-Oleh Purnama Jati)” , telah diuji dan disahkan pada :

hari, tanggal : Jumat, 26 Mei 2017

tempat : Kampus Program Studi Sistem Informasi

Disetujui oleh :

Pembimbing I,

Pembimbing II,

Drs. Antonius Cahya P., M.App.Sc., Ph.D.
NIP. 196909281993021001

Fahrobby Adnan, S.Kom., MMSI.
NIP. 198706192014041001

PENGESAHAN PENGUJI

Skripsi berjudul “Sistem Informasi Peramalan Penjualan Prol Tape dan Pia Tape menggunakan Metode *Winter Exponential Smoothing* (Studi Kasus Pusat Oleh-Oleh Purnama Jati) ”, telah diuji dan disahkan pada :

hari, tanggal : Jumat, 26 Mei 2017

tempat : Kampus Program Studi Sistem Informasi

Tim Penguji :

Penguji I,

Penguji II,

Nelly Oktavia Adiwijaya S.Si.,MT.

NIP 198410242009122008

Priza Pandunata S.Kom., M.Sc.

NIP 198301312015041001

Mengesahkan

Ketua Program Studi

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

NIP. 19670420 1992011001

RINGKASAN

Sistem Informasi Peramalan Penjualan Prol Tape dan Pia Tape menggunakan Metode *Winter Exponential Smoothing* (Studi Kasus Pusat Oleh-Oleh Purnama Jati); Putri Damayanti, 132410101090; 2017: 162 halaman, Program Studi Sistem Informasi.

Perencanaan merupakan kegiatan awal dari suatu kegiatan produksi. Perencanaan jadwal produksi adalah salah satu hal yang penting dalam kegiatan produksi. Pusat Oleh-oleh Purnama Jati memiliki sistem perencanaan jadwal produksi yang reguler setiap harinya. Jumlah barang yang diproduksi didasarkan pada perkiraan. Hal ini menimbulkan sebuah permasalahan yaitu produksi harus dilakukan secara berulang-ulang tiap harinya karena dalam sehari outlet penerima melakukan permintaan beberapa kali. Penelitian ini dilakukan guna membantu Purnama Jati untuk menyelesaikan permasalahan tersebut. Dengan mengembangkan sistem peramalan penjualan produk dapat membantu menyelesaikan permasalahan tersebut. Sistem ini akan menampilkan peramalan penjualan pada keesokan harinya sehingga bagian produksi dapat memperkirakan jumlah produk yang akan diproduksi esok harinya agar produk yang dihasilkan tidak menumpuk atau mengalami kekurangan. Sistem peramalan penjualan ini memiliki 3 aktor yaitu Admin, Marketing dan Produksi. Masukan yang digunakan pada proses perhitungan peramalan adalah data penjualan periode sebelumnya. Pada proses perhitungan peramalan penjualan digunakan metode *Winter Exponential Smoothing* yang cocok untuk data Purnama Jati yang berpola musiman. Sistem yang dikembangkan dapat melakukan perhitungan peramalan data penjualan produk Prol Tape Besar, Prol Tape Kecil dan Pia Tape dengan menggunakan metode *Winter Exponential Smoothing*.

PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan hidayah-Nya, sehingga penulis mampu menyelesaikan tugas akhir dengan judul “Sistem Informasi Peramalan Penjualan Prol Tape dan Pia Tape menggunakan Metode *Winter Exponential Smoothing* (Studi Kasus Pusat Oleh-Oleh Purnama Jati)“. Skripsi ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari dukungan berbagai pihak. Oleh karena itu penulis mengucapkan terima kasih kepada :

1. Allah SWT
2. Ayah Sutrisno, Ibu Indrayani dan Adik Dwi Ayu Noviyanti atas segala dukungan dan doa-doa yang tiada henti;
3. Bapak Prof. Drs. Slamini, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
4. Bapak Drs. Antonius Cahya Prihandoko, M.App.Sc., Ph.D. selaku Dosen Pembimbing Utama dan Bapak Fahrobby Adnan, S.Kom., MMSI. yang telah memberikan arahan-arahan dalam proses pengerjaan skripsi ini;
5. Bapak Muhamad Arief Hidayat, S.Kom, M.Kom. selaku Dosen Pembimbing Akademik yang telah mendampingi penulis selama masa perkuliahan S1;
6. Seluruh bapak ibu dosen beserta staf akademik Program Studi Sistem Informasi;
7. Sahabat serta teman-teman seperjuangan yang selalu menemani dan memberi semangat serta doa;
8. Serta semua pihak yang tidak dapat disebutkan satu per satu yang telah membantu baik tenaga maupun doa dalam pelaksanaan kegiatan penelitian dan penyusunan skripsi ini.

Dengan harapan bahwa penelitian ini akan bermanfaat bagi semua pihak dan akan terus berkembang. Penulis juga mengharapkan saran dan kritik yang bersifat membangun dari semua pihak.

Jember, Mei 2017

Penulis



DAFTAR ISI

PERSEMBAHAN	III
MOTO	IV
PERNYATAAN.....	V
SKRIPSI.....	VI
PENGESAHAN PEMBIMBING.....	VII
PENGESAHAN PENGUJI.....	VIII
RINGKASAN	IX
PRAKATA.....	X
DAFTAR ISI.....	XII
DAFTAR TABEL.....	XVI
DAFTAR GAMBAR	XVIII
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan dan Manfaat	2
1.3.1 Tujuan	2
1.3.2 Manfaat	3
1.4 Batasan Masalah.....	3
1.5 Sistematika Penulisan.....	3
BAB 2 TINJAUAN PUSTAKA	5
2.1 Penelitian Terdahulu	5
2.2 Dasar Teori.....	6
2.2.1 <i>Winter Exponential Smoothing</i>	6
2.2.2 Perhitungan Kesalahan Peramalan.....	10

BAB 3 METODOLOGI PENELITIAN.....	11
3.1 Pengembangan Sistem	11
3.1.1 Analisis Kebutuhan	12
3.1.2 Desain Sistem.....	13
3.1.3 Implementasi	14
3.1.4 Pengujian.....	14
3.1.5 <i>Maintanance</i>	14
3.2 Analisis Metode <i>Winter Exponential Smoothing</i>	15
3.3 Gambaran Umum Sistem	15
BAB 4 PENGEMBANGAN SISTEM.....	15
4.1 Analisis Kebutuhan Sistem	16
4.1.1 Kebutuhan Fungsional	17
4.1.2 Kebutuhan Non Fungsional.....	17
4.2 Desain Sistem.....	18
4.2.1 <i>Business Process</i>	18
4.2.2 <i>Use Case Diagram</i>	18
4.2.3 <i>Use Case Scenario</i>	21
4.2.4 <i>Sequence Diagram</i>	35
4.2.5 <i>Activity Diagram</i>	47
4.2.6 <i>Class Diagram</i>	61
4.2.7 Entity Relationship Diagram (ERD)	63
4.3 Implementasi	63
4.3.1 Kode Program <i>Login</i>	63
4.3.2 Kode Program Mengelola Data Akun Pengguna	64
4.3.3 Kode Program Mengelola Data Diri	65

4.3.4	Kode Program Mengelola Data Pesanan.....	66
4.3.5	Kode Program Melihat Data Pesanan	68
4.3.6	Kode Program Mengelola Data Penjualan.....	68
4.3.7	Kode Program Mengelola Hasil Peramalan	70
4.3.8	Kode Program Melihat Data Penjualan.....	71
4.3.9	Kode Program Melihat Data Hasil Peramalan	71
4.3.10	Kode Program Mengelola Data Produk	72
4.3.11	Kode Program Melihat Data Produk.....	73
4.3.12	Kode Program Mengelola Data Produksi	73
4.3.13	Kode Program Melihat Data Produksi	74
4.4	Pengujian.....	75
4.4.1	<i>White Box</i>	75
4.4.2	<i>Black Box</i>	80
BAB 5 HASIL DAN PEMBAHASAN.....		84
5.1	Hasil Penerapan Metode Peramalan <i>Winter Exponential Smoothing</i>	84
5.1.1	Proses Peramalan dengan Menggunakan <i>Winter Exponential Smoothing</i>	84
5.1.2	Perhitungan Peramalan Data Hasil Penjualan dengan menggunakan Metode <i>Winter Exponential Smoothing</i>	84
5.2	Pembahasan Penerapan Metode <i>Winter Exponential Smoothing</i> pada Proses Peramalan Penjualan	96
5.3	Hasil Pembangunan Sistem Informasi Peramalan Penjualan Menggunakan Metode <i>Winter Exponential Smoothing</i>	97
5.3.1	Fitur <i>Login</i>	98
5.3.2	Fitur Mengelola Data Diri	99
5.3.3	Fitur Mengelola Data Akun Pengguna.....	100

5.3.4	Fitur Mengelola Data Pesanan	101
5.3.5	Fitur Melihat Data Pesanan	102
5.3.6	Fitur Mengelola Data Penjualan.....	103
5.3.7	Fitur Mengelola Data Hasil Peramalan	104
5.3.8	Fitur Melihat Data Penjualan	105
5.3.9	Fitur Melihat Data Hasil Peramalan.....	106
5.3.10	Fitur Mengelola Data Produk.....	106
5.3.11	Fitur Melihat Data Produk	107
5.3.12	Fitur Mengelola Data Produksi	108
5.3.13	Fitur Melihat Data Produksi.....	109
5.4	Pembahasan Hasil Pembangunan Sistem Informasi Peramalan Penjualan Menggunakan Metode <i>Winter Exponential Smoothing</i>	109
5.4.1	Kemampuan Sistem	110
5.4.2	Keterbatasan Sistem	110
BAB 6 PENUTUP.....		112
7.1	Kesimpulan	112
7.2	Saran.....	113
DAFTAR PUSTAKA		114
LAMPIRAN		115
A.	<i>Use Case Scenario</i>	115
B.	<i>Sequence Diagram</i>	120
C.	<i>Activity Diagram</i>	124
D.	Pengujian <i>Black Box</i>	130
E.	Pengujian White Box	135
F.	Transkrip Hasil Wawancara	140

DAFTAR TABEL

Tabel 2.1 Perbandingan Metode <i>Winter Exponential Smoothing</i>	9
Tabel 4.1 Definisi Aktor	20
Tabel 4.2 Definisi Proses	20
Tabel 4.3 Skenario Menambah Data Akun Pengguna	22
Tabel 4.4 Skenario Mengubah Data Akun Pengguna	23
Tabel 4.5 Skenario Menghapus Data Akun Pengguna.....	23
Tabel 4.6 Skenario Mengelola Data Diri (Admin).....	24
Tabel 4.7 Skenario Mengelola Data Diri (Produksi)	25
Tabel 4.8 Skenario Mengelola Data Diri (Marketing)	26
Tabel 4.9 Skenario Mengelola Data Pesanan (Input).....	27
Tabel 4.10 Skenario Mengelola Data Pesanan (Edit)	27
Tabel 4.11 Skenario Mengelola Data Pesanan (Hapus).....	28
Tabel 4.12 Skenario <i>Use Case</i> Mengelola Data Penjualan.....	29
Tabel 4.13 Skenario Mengelola Data Hasil Peramalan	30
Tabel 4.14 Skenario Mengelola Data Produk (Input)	32
Tabel 4.15 Skenario Mengelola Data Produk (Edit).....	32
Tabel 4.16 Skenario Mengelola Data Produk (Delete)	33
Tabel 4.17 Skenario Mengelola Data Produksi.....	34
Tabel 4.18 Test Case Kelas login/auth.....	75
Tabel 4.19 Test Case Kelas forecast/add	77
Tabel 4.20 Test Case Kelas sales/update	79
Tabel 4.21 Hasil Pengujian Fitur Mengelola Data Akun Pengguna	80
Tabel 4.22 Hasil Pengujian Fitur Mengelola Data Diri	82
Tabel 4.23 Hasil Pengujian Fitur Mengelola Data Penjualan	82
Tabel 4.24 Hasil Pengujian Fitur Mengelola Data Hasil Peramalan	83
Tabel 5.1 Nilai Inisial Perhitungan Peramalan	85
Tabel 5.2 Penjelasan Perhitungan Nilai Inisial Peramalan	86
Tabel 5.3 Hasil Peramalan Penjualan Prol Tape Besar bulan Oktober 2016.....	87
Tabel 5.4 Hasil Peramalan Penjualan Prol Tape Kecil bulan Oktober 2016	87
Tabel 5.5 Hasil Peramalan Penjualan Pia Tape bulan Oktober 2016	90

Tabel A. 1 Use Case Scenario Login (Admin)	115
Tabel A. 2 Use Case Scenario Login (Admin)	115
Tabel A. 3 Use Case Scenario Login (Admin)	116
Tabel A. 4 Use Case Scenario Melihat Data Pesanan.....	117
Tabel A. 5 Use Case Scenario Melihat Data Penjualan.....	117
Tabel A. 6 Use Case Scenario Melihat Data Hasil Peramalan (Admin).....	118
Tabel A. 7 Use Case Scenario Melihat Data Hasil Peramalan (Produksi).....	118
Tabel A. 8 Use Case Scenario Melihat Data Produk	119
Tabel A. 9 Use Case Scenario Melihat Data Produksi.....	119
Tabel D. 1 Hasil Pengujian Fitur Login	130
Tabel D. 2 Hasil Pengujian Fitur Mengelola Data Pesanan.....	131
Tabel D. 3 Hasil Pengujian Fitur Melihat Data Pesanan	132
Tabel D. 4 Hasil Pengujian Fitur Melihat Data Penjualan.....	132
Tabel D. 5 Hasil Pengujian Fitur Melihat Data Hasil Peramalan	132
Tabel D. 6 Hasil Pengujian Fitur Mengelola Data Produk	133
Tabel D. 7 Hasil Pengujian Fitur Melihat Data Produk	134
Tabel D. 8 Hasil Pengujian Fitur Mengelola Data Produksi.....	134
Tabel D. 9 Pengujian Fitur Melihat Data Produksi.....	134
Tabel E. 1 Test Case Kelas users/changepass	135
Tabel E. 2 Test Case Kelas dashboard/index	136
Tabel E. 3 Test Case Kelas order/edit.....	137
Tabel E. 4 Test Case Kelas product/edit	138
Tabel E. 5 Test Case Kelas product/delete.....	139
Tabel E. 6 Test Case Kelas production/edit	139
Tabel E. 7 Test Case Kelas production/update	140

DAFTAR GAMBAR

Gambar 2.1 Grafik Data Stasioner	6
Gambar 2.2 Grafik Data Trend	7
Gambar 2.3 Grafik Data Musiman.....	7
Gambar 3.1 Model SDLC Agile	11
Gambar 3.2 Flowchart Model Agile	12
Gambar 4.1 Business Process	18
Gambar 4.2 Use Case Diagram.....	19
Gambar 4.3 Sequeunce Mengelola Data Akun Pengguna (Input)	36
Gambar 4.4 Sequeunce Mengelola Data Akun Pengguna (Edit).....	37
Gambar 4.5 Sequeunce Mengelola Data Akun Pengguna (Delete)	38
Gambar 4.6 Sequeunce Mengelola Data Diri (Admin).....	39
Gambar 4.7 Sequeunce Mengelola Data Diri (Marketing)	39
Gambar 4.8 Sequeunce Mengelola Data Diri (Produksi).....	40
Gambar 4.9 Sequence Mengelola Data Pesanan (Input).....	41
Gambar 4.10 Sequence Mengelola Data Pesanan (Delete).....	41
Gambar 4.11 Sequence Mengelola Data Pesanan (Edit)	42
Gambar 4.12 Sequence Mengelola Data Penjualan	43
Gambar 4.13 Sequence Diagram Mengelola Data Hasil Peramalan.....	44
Gambar 4.14 Sequence Mengelola Data Produk	45
Gambar 4.15 Sequence Mengelola Data Produk	45
Gambar 4.16 Sequence Mengelola Data Produk	46
Gambar 4.17 Sequence Mengelola Data Produksi.....	47
Gambar 4.18 Activity Mengelola Data Akun Pengguna	48
Gambar 4.19 Activity Mengelola Data Akun Pengguna	49
Gambar 4.20 Activity Mengelola Data Akun Pengguna	49
Gambar 4.21 Activity Mengelola Data Diri.....	50
Gambar 4.22 Activity Mengelola Data Diri.....	51
Gambar 4.23 Activity Mengelola Data Diri.....	52
Gambar 4.24 Activity Mengelola Data Pesanan	53
Gambar 4.25 Activity Mengelola Data Pesanan	54

Gambar 4.26 Activity Mengelola Data Pesanan	55
Gambar 4.27 Activity Mengelola Data Penjualan	56
Gambar 4.28 Activity Diagram Mengelola Hasil Peramalan	57
Gambar 4.29 Activity Mengelola Data Produk.....	58
Gambar 4.30 Activity Mengelola Data Produk.....	59
Gambar 4.31 Activity Mengelola Data Produk.....	60
Gambar 4.32 Activity Mengelola Data Produksi	61
Gambar 4.33 <i>Class Diagram</i>	62
Gambar 4.34 Entity Relationship Diagram.....	63
Gambar 4.35 Kode Program Kelas Login/auth.....	64
Gambar 4.36 Kode Program Mengelola Data Akun Pengguna	65
Gambar 4.37 Kode Program Mengelola Data Akun Pengguna	65
Gambar 4.38 Kode Program Mengelola Data Diri	66
Gambar 4.39 Kode Program Mengelola Data Pesanan.....	67
Gambar 4.40 Kode Program Mengelola Data Pesanan.....	67
Gambar 4.41 Kode Program Mengelola Data Penjualan	69
Gambar 4.42 Kode Program Mengelola Data Penjualan	69
Gambar 4.43 Kode Program Mengelola Data Penjualan	70
Gambar 4.44 Kode Program Kelas index dan detail.....	70
Gambar 4.45 Kode Program Kelas add.....	71
Gambar 4.46 Kode Program Mengelola Data Produk	72
Gambar 4.47 Kode Program Mengelola Data Produk	73
Gambar 4.48 Kode Program Mengelola Data Produksi.....	74
Gambar 4.49 Flow Graph login/auth	75
Gambar 4.50 Flow Graph Kelas forecast/add	76
Gambar 4.51 Flow Graph Kelas sales/update	78
Gambar 5.1 Grafik perbandingan penjualan aktual dengan hasil peramalan.....	90
Gambar 5.2 Grafik perbandingan penjualan aktual dan hasil peramalan	93
Gambar 5.3 Grafik perbandingan hasil peramalan dengan penjualan aktual.....	96
Gambar 5.4 Halaman Login	98
Gambar 5.5 Halaman Login dengan Peringatan	98

Gambar 5.6 Halaman Login dengan Peringatan	99
Gambar 5.7 Halaman Dashboard dengan Peringatan	99
Gambar 5.8 Halaman Ubah Password	100
Gambar 5.9 Halaman Mengelola Data Akun Pengguna	100
Gambar 5.10 Halaman Mengelola Data Akun Pengguna	101
Gambar 5.11 Halaman Daftar Pesanan	101
Gambar 5.12 Menambah Data Pesanan	102
Gambar 5.13 Halaman Edit Data Pesanan	102
Gambar 5.14 Halaman Daftar Pesanan	103
Gambar 5.15 Halaman Daftar Penjualan	103
Gambar 5.16 Halaman Edit Penjualan	104
Gambar 5.17 Halaman Daftar Produk Untuk Peramalan.....	104
Gambar 5.18 Halaman Daftar Peramalan	105
Gambar 5.19 Halaman Melihat Data Penjualan.....	105
Gambar 5.20 Halaman Melihat Data Hasil Peramalan	106
Gambar 5.21 Halaman Daftar Produk.....	107
Gambar 5.22 Halaman Edit Data Produk.....	107
Gambar 5.23 Halaman Daftar Produk.....	108
Gambar 5.24 Halaman Mengelola Data Produksi.....	108
Gambar 5.25 Halaman Mengelola Data Produksi.....	109
Gambar 5.26 Halaman Melihat Data Produksi	109
Gambar B. 1 Sequence Diagram Login Marketing.....	120
Gambar B. 2 Sequence Diagram Login Admin	120
Gambar B. 3 Sequence Diagram Login Produksi	121
Gambar B. 4 Sequence Diagram Melihat Data Pesanan Admin.....	121
Gambar B. 5 Sequence Diagram Melihat Data Pesanan Produksi.....	121
Gambar B. 6 Sequence Diagram Melihat Data Penjualan Admin	122
Gambar B. 7 Sequence Diagram Melihat Data Peramalan (Admin)	122
Gambar B. 8 Sequence Diagram Melihat Data Peramalan (Admin)	123
Gambar B. 9 Sequence Diagram Melihat Data Produk Admin	123

Gambar B. 10 Sequence Diagram Melihat Data Produksi Admin.....	123
Gambar C. 1 Activity Diagram Login Marketing	124
Gambar C. 2 Activity Diagram Login Marketing	125
Gambar C. 3 Activity Diagram Login Produksi	126
Gambar C. 4 Activity Diagram Melihat Data Pesanan (Admin)	127
Gambar C. 5 Activity Diagram Melihat Data Pesanan (Produksi)	127
Gambar C. 6 Activity Diagram Melihat Data Penjualan	128
Gambar C. 7 Activity Diagram Melihat Data Hasil Peramalan (Admin)	128
Gambar C. 8 Activity Diagram Melihat Data Hasil Peramalan (Produksi).....	129
Gambar C. 9 Activity Diagram Melihat Data Produk.....	129
Gambar C. 10 Activity Diagram Melihat Data Produksi	130
Gambar E. 1 Flow Graph Kelas users/changepass.....	135
Gambar E. 2 Flow Graph Kelas dashboard/index.....	136
Gambar E. 3 Flow Graph Kelas order/edit.....	137
Gambar E. 4 Flow Graph Kelas product/edit.....	138
Gambar E. 5 Flow Graph Kelas product/delete	138
Gambar E. 6 Flow Graph Kelas production/edit.....	139
Gambar E. 7 Flow Graph Kelas production/update	139

BAB 1

PENDAHULUAN

Bab ini merupakan langkah awal dari penulisan tugas akhir. Bab ini berisi latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah, metodologi penelitian, dan sistematika penulisan.

1.1 Latar Belakang

Kegiatan produksi merupakan kegiatan utama dalam perekonomian perusahaan produsen. Banyak kendala yang umum terjadi pada perusahaan produsen terkait masalah produksi seperti produk yang berlebih atau kekurangan produk. Permasalahan produksi ini juga sering terjadi di Purnama Jati sebagai perusahaan produksi makanan oleh-oleh. Purnama Jati sering kali melakukan produksi berulang-ulang dikarenakan permintaan yang terus bertambah setiap harinya. Jika produk yang dibuat berlebih maka akan menyia-nyiakan masa kadaluwarsa. Sedangkan jika produk yang dibuat kurang maka industri harus melakukan proses produksi secara berulang. Hal tersebut tentu membawa kerugian untuk industri tersebut. Oleh karena itu dibutuhkan suatu ramalan jumlah produk yang terjual agar jumlah produk yang diproduksi dapat terencana dengan baik. Dan untuk meminimalisir kelebihan atau kekurangan produk.

Penelitian ini dilakukan untuk menghasilkan peramalan yang tepat dengan studi kasus di Pusat Oleh-oleh Purnama Jati. Purnama Jati merupakan tempat jajanan oleh-oleh khas Jember seperti prol tape, pia tape, brownies tape dan lain sebagainya. Hampir semua produk yang dijual di Purnama Jati berbahan dasar tape. Purnama Jati melakukan proses produksi beberapa produknya setiap hari seperti prol tape dan pia tape karena kedua produk ini yang paling laku dipasaran. Beberapa produk lainnya diproduksi pada waktu tertentu. Produk prol tape dan pia tape dipasarkan ke beberapa outlet dan outlet milik Purnama Jati sendiri setiap harinya.

Purnama Jati melakukan produksi rata-rata 200 pcs per harinya terkecuali hari-hari besar. Dan 200 pcs inilah yang didistribusikan ke outlet-outlet. Setiap hari penjualan pada setiap outlet tidak menentu sehingga Purnama Jati berulang kali

melakukan pengiriman ke outlet-outlet tersebut jika stok telah habis. Akibatnya dalam sehari Purnama Jati dapat melakukan beberapa kali produksi. Produksi yang berulang kali tentu membuang waktu dan dana. Dalam penelitian ini dibuat sebuah sistem yang dapat mengatasi masalah tersebut dengan meramalkan penjualan untuk hari berikutnya. Sehingga dapat meminimalisir kelebihan atau kekurangan produk keesokan harinya.

Sistem yang dibuat dalam penelitian ini adalah Sistem Informasi Peramalan Penjualan Menggunakan Metode *Winter Exponential Smoothing*. Sistem ini meramalkan penjualan untuk hari berikutnya. Sistem juga dapat mengelola beberapa proses lain seperti mengelola pesanan dan produksi. Dengan melakukan peramalan terlebih dahulu tentu dapat membantu Purnama Jati dalam hal jumlah produk yang akan diproduksi keesokan harinya. Dan hal ini tentu meminimalisir adanya kerugian dan waktu yang terbuang. Dalam meramalkan penjualan dalam sistem ini digunakan Metode *Winter Exponential Smoothing*. Metode ini digunakan untuk data berpola trend dan musiman. Pada penelitian kali ini data penjualan Purnama Jati yang digunakan merupakan data berpola stasioner, trend dan musiman. Sehingga metode ini cocok digunakan untuk penelitian ini.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang yang telah dipaparkan diatas, dapat diambil keputusan bahwa permasalahan yang muncul adalah sebagai berikut :

1. Bagaimana meramalkan penjualan menggunakan metode *Winter Exponential Smoothing*?
2. Bagaimana membangun sebuah sistem informasi peramalan penjualan dengan menerapkan metode *Winter Exponential Smoothing* ?

1.3 Tujuan dan Manfaat

1.3.1 Tujuan

Tujuan dari penelitian yang dilakukan adalah :

1. Meramalkan penjualan untuk periode tertentu dengan metode *Winter Exponential Smoothing*

2. Menghitung tingkat akurasi peramalan penjualan dengan metode *Winter Exponential Smoothing*
3. Membangun sebuah sistem informasi peramalan penjualan dengan Metode *Winter Exponential Smoothing*

1.3.2 Manfaat

Manfaat yang didapatkan dari penelitian ini adalah :

1. Peneliti mendapat tambahan pengetahuan tentang peramalan dan metode yang digunakan
2. Lembaga pendidikan mendapatkan tambahan penelitian
3. Membantu Purnama Jati mengatasi masalah produksinya
4. Purnama Jati dapat meningkatkan penjualan dan dapat memanajemen kegiatan produksi dengan baik

1.4 Batasan Masalah

Batasan masalah dari penelitian yang dilakukan adalah :

1. Produk yang diteliti adalah Prol Tape dan Pia Tape
2. Data yang digunakan adalah data bulan Oktober – Desember 2016
3. Tidak memberi alasan perubahan data
4. Sistem yang dikembangkan pada penelitian ini berbasis website
5. Sistem yang dibangun menggunakan bahasa pemrograman PHP dan *database* MYSQL

1.5 Sistematika Penulisan

Sistematika penulisan dan keruntutan skripsi ini disusun sebagai berikut :

1. Pendahuluan
Bab ini menjelaskan tentang latar belakang, perumusan masalah, tujuan, dan manfaat, ruang lingkup studi dan sistematika penulisan.
2. Tinjauan Pustaka
Bab ini menjelaskan tentang materi, informasi, tinjauan pustaka dan studi terdahulu menjadi kerangka pemikiran dalam penelitian.

3. Metodologi Penelitian

Bab ini menjelaskan tentang metode penelitian yang digunakan dalam penelitian.

4. Pengembangan Sistem

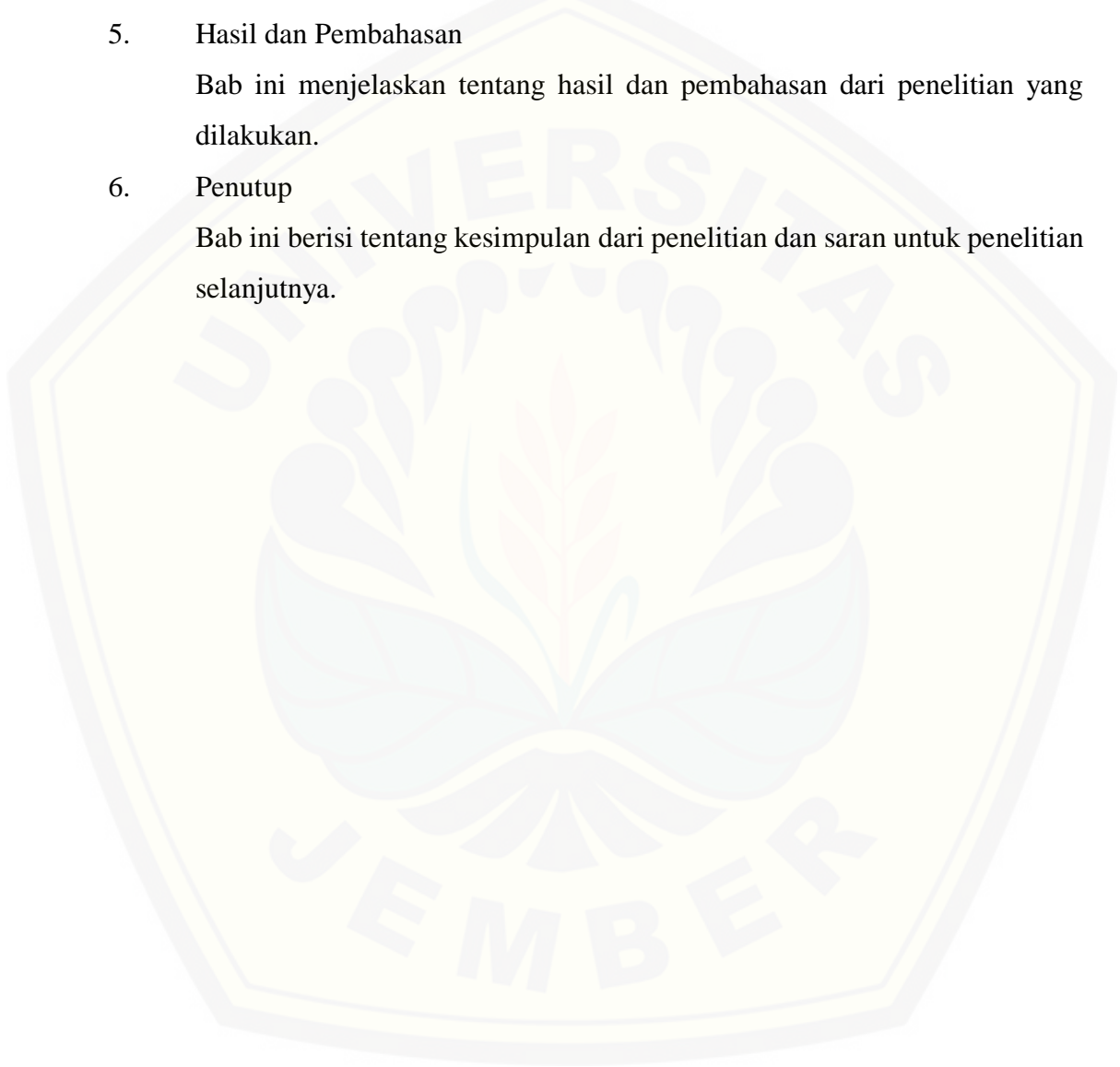
Bab ini menjelaskan tentang pengembangan sistem yang dikembangkan.

5. Hasil dan Pembahasan

Bab ini menjelaskan tentang hasil dan pembahasan dari penelitian yang dilakukan.

6. Penutup

Bab ini berisi tentang kesimpulan dari penelitian dan saran untuk penelitian selanjutnya.



BAB 2

TINJAUAN PUSTAKA

Pada bagian ini akan dijelaskan tinjauan terhadap hasil-hasil penelitian berkaitan dengan masalah yang dibahas, kajian teori berkaitan dengan masalah, kerangka pemikiran yang merupakan sintesis dari kajian teori yang dikaitkan dengan permasalahan yang dihadapi.

2.1 Penelitian Terdahulu

Perencanaan produksi merupakan akar dari suatu kegiatan produksi. Dalam merencanakan kegiatan produksi dibutuhkan suatu ramalan tentang berapa banyak jumlah produk yang akan diproduksi. Metode yang digunakan untuk meramalkan bermacam-macam. Dalam penelitian Amrit Pal singh dan kawan-kawan (Pal, Gaur, & Kumarkasdekar, 2015), peneliti membandingkan 4 metode yaitu Moving Average, Single Exponential Smoothing, Double Exponential Smoothing dan *Winter Exponential Smoothing*. Studi kasus yang diteliti adalah perusahaan sepatu dimana sering terjadi lonjakan permintaan atau permintaan yang tidak terprediksi sebelumnya. Dan kesimpulan dari penelitian ini setelah membandingkan 4 metode tersebut didapatkan bahwa metode *Winter Exponential Smoothing* yang cocok untuk data pada studi kasus. Peneliti juga menyimpulkan bahwa keakuratan suatu peramalan dilihat dari besar *error* yang dihasilkan.

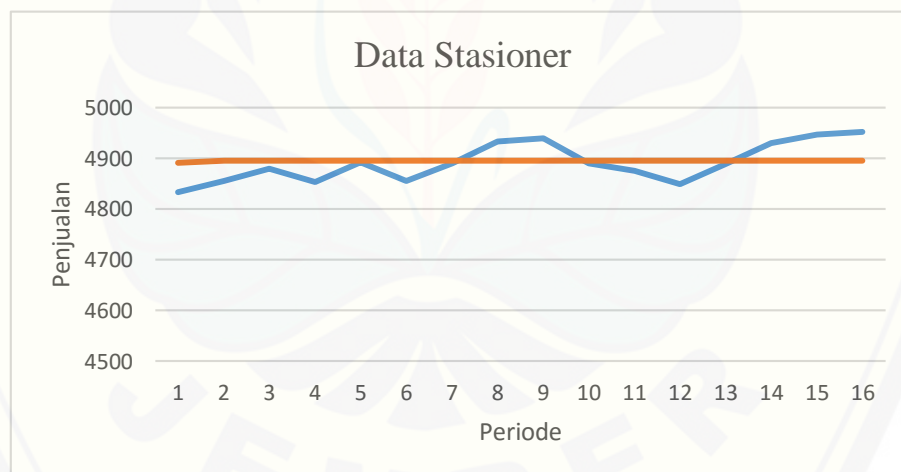
Salah satu metode dalam menghitung *error* dari suatu peramalan adalah dengan menghitung *Mean Absolute Percentage Error* (MAPE). Hasil peramalan dikatakan sangat akurat jika nilai MAPE kurang dari 10% , akurat jika kurang dari 30% dan tidak akurat jika lebih dari 30%. Hal ini dijelaskan oleh Claudimar Pereira Da Veiga (Da Veiga, Da Veiga, Catapan, Tortato, & Da Silva, 2014) pada penelitiannya tentang perbandingan metode ARIMA dan *Holt-Winter* dalam meramalkan permintaan di industri makanan. Dan hasil penelitian tersebut adalah metode *Holt-Winter* lebih cocok untuk data yang disediakan. Metode *Holt-Winter* dinilai sederhana dan dapat membuktikan keakuratan hasil peramalan daripada hasil yang diperoleh dari metode lain yang lebih rumit.

2.2 Dasar Teori

2.2.1 *Winter Exponential Smoothing*

Metode *Winter Exponential Smoothing* (WES) digunakan untuk meramalkan data yang berpola tidak teratur atau berubah-ubah sesuai dengan indikator trend dan musiman. Metode WES digunakan untuk data yang berpola stasioner, trend dan musiman.

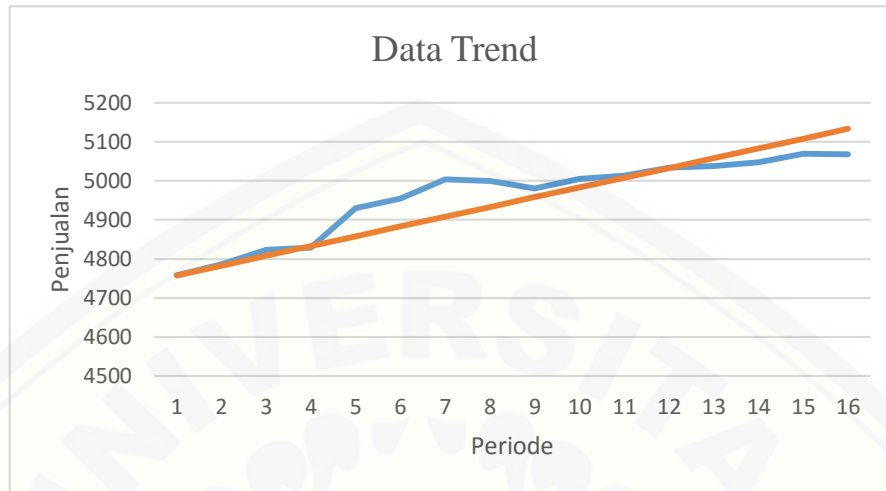
Data stasioner berkaitan erat dengan kebiasaan statistik data pada waktu tertentu dan ini sering ditandai dengan distribusi probabilitas konstan setiap waktu (Montgomery, Jennings, & Kulahci, 2011). Data stasioner bergerak pada rata-rata data. Fluktuasi yang terjadi tidak berubah jauh dari rata-rata data seperti ditampilkan pada Gambar 2.1. Data *stasioner* dapat ditentukan dengan mengambil “*snapshot*” dari proses pada titik-titik yang berbeda dan mengamati perilaku umumnya. Jika menunjukkan perilaku yang sama maka dapat diasumsikan bahwa data tersebut termasuk data *stasioner* (Montgomery et al., 2011).



Gambar 2.1 Grafik Data *Stasioner*

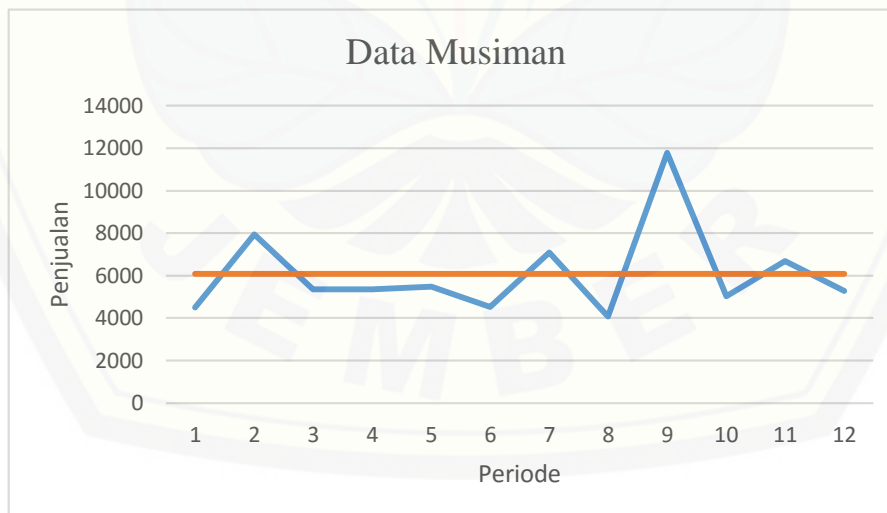
Data dapat mengalami pergeseran bertahap atau gerakan yang relatif lebih tinggi atau lebih rendah selama periode waktu yang lebih lama. Data dengan kebiasaan tersebut disebut data berpola trend seperti digambarkan pada Gambar 2.2. Data *trend* biasanya merupakan hasil faktor-faktor jangka panjang seperti

peningkatan atau penurunan karakter tertentu (Anderson, Sweeney, Williams, Camm, & Martin, 2011).



Gambar 2.2 Grafik Data *Trend*

Data musiman dapat dilihat dari adanya pola yang berulang secara berturut-turut dalam kurun waktu tertentu (Anderson et al., 2011). Pola yang dibentuk data musiman dapat berupa penurunan dan peningkatan yang signifikan serta berfluktuasi tinggi seperti digambarkan pada Gambar 2.3.



Gambar 2.3 Grafik Data Musiman

Metode *Winter Exponential Smoothing* digunakan untuk pola data dengan pergerakan yang berubah-ubah mengikuti musim atau trend. Metode ini merupakan salah satu metode yang populer digunakan untuk data berpola musiman. Selain

menggunakan faktor level/stasioner dan tren metode *Winter Exponential Smoothing* juga menggunakan faktor musiman untuk setiap musim (Winston & Albright, 2016). Pada metode yang dikemukakan oleh winter ini, didasarkan atas 3 (tiga) parameter penghalusan, yaitu satu untuk unsur , satu untuk trend, dan satu untuk musiman. Berikut adalah rumus perhitungan *winter exponential smoothing* pada persamaan 2.1, persamaan 2.2, persamaan 2.3 dan persamaan 2.4.

$$A_t = \frac{\alpha Y_t}{S_{t-L}} + (1 - \alpha)(A_{t-1} + T_{t-1}) \dots\dots\dots \text{Persamaan 2.1}$$

$$T_t = \beta(A_t - A_{t-1}) + (1 - \beta)T_{t-1} \dots\dots\dots \text{Persamaan 2.2}$$

$$S_t = \frac{\gamma Y_t}{A_t} + (1 - \gamma)S_{t-L} \dots\dots\dots \text{Persamaan 2.3}$$

$$F_{t+p} = (A_t + pY_t)S_{t-L+p} \dots\dots\dots \text{Persamaan 2.4}$$

Keterangan :

A_t = Nilai Pemulusan

α = Konstanta Pemulusan ($0 < \alpha < 1$)

Y_t = Nilai aktual pada periode t

β = Konstanta pemulusan untuk estimasi trend ($0 < \beta < 1$)

T_t = Estimasi trend

γ = Konstanta pemulusan untuk musiman ($0 < \gamma < 1$)

S_t = Estimasi musiman diukur sebagai index

L = Panjang musim

p = Jumlah periode yang akan diramal

F_{t+p} = Hasil peramalan periode p

Sebelum memutuskan menggunakan metode *Winter Exponential Smoothing* pada penelitian ini dilakukan beberapa perbandingan dengan metode-metode peramalan yang lain yaitu *Single Exponential Smoothing* (SES), *Double Exponential Smoothing* (DES) dan *Triple Exponential Smoothing* (TES). Seperti dijelaskan pada Tabel 2.1. berikut :

Tabel 2.1 Perbandingan Metode *Winter Exponential Smoothing*

	WES	SES	DES	TES
Parameter	Menggunakan 3 parameter : Stasioner, trend dan musiman	Menggunakan 1 parameter : Stasioner	Menggunakan 2 parameter : Stasioner dan trend	Menggunakan 1 parameter : Stasioner
Estimasi trend dan musiman	Memiliki estimasi trend dan musiman	Tidak memiliki estimasi trend dan musiman	Memiliki estimasi trend	Tidak memiliki estimasi trend dan musiman
Data	Untuk data berpola trend dan musiman	Untuk data statis	Untuk data trend	Untuk data musiman

Dari penjelasan Tabel 2.1 peneliti memilih metode Winter Exponential Smoothing karena metode WES lebih detail dalam perhitungan peramalannya. Metode WES memperhatikan estimasi secara detail dengan memasukkan estimasi stasioner, trend dan musiman kedalam perhitungan. Peneliti menentukan metode peramalan dengan memperhatikan pola data yang dikelola. Metode WES dan TES memiliki kesamaan pada pola data yang dikelola. Perbedaan metode WES dan TES adalah pada estimasi atau parameter perhitungannya. Metode WES memiliki 3 estimasi yaitu stasioner, trend dan musiman. Sedangkan metode TES hanya memiliki 1 parameter yaitu stasioner (Strasheim, 2012). Dalam proses perhitungan metode TES menggunakan metode SES dan DES terlebih dahulu. Sedangkan metode WES langsung melakukan perhitungan dengan mengelola 3 parameter yang telah disebut sebelumnya. Oleh karena itu, peneliti lebih melihat perlakuan metode WES dan TES kepada data. Menurut peneliti metode WES lebih detail dalam penanganan data karena mempertimbangkan tidak hanya 1 parameter dan dalam proses perhitungan metode WES dinilai lebih praktis.

2.2.2 Perhitungan Kesalahan Peramalan

Dalam menghitung kesalahan pada hasil peramalan terdapat beberapa cara yaitu dengan menghitung *Mean Absolute Percentage Error* (MAPE) dan *Mean Absolute Deviation* (MAD).

Mean Absolute Percentage Error (MAPE) merupakan perhitungan untuk mengukur akurasi dari hasil sebuah metode peramalan. Metode ini menghitung rata-rata nilai absolute dari presentase kesalahan data asli dengan data hasil peramalan seperti yang dijelaskan pada persamaan (2.5). (Camm, Cochran, Fry, Ohlmann, & Anderson, 2015)

$$MAPE = \frac{\sum \left| \left(\frac{Et}{Yt} \right) * 100 \right|}{n} \dots\dots\dots \text{Persamaan 2.5}$$

Keterangan :

Et = Nilai Absolute Kesalahan

Yt = Nilai Aktual

n = Jumlah Data

Mean Absolute Deviation (MAD) adalah perhitungan untuk mengukur kesalahan sebuah peramalan dengan menghindari permasalahan nilai positif ataupun negatif untuk menutupi kesalahan yang lain. Persamaan (2.6) menunjukkan rumus untuk menghitung MAD. (Camm et al., 2015)

$$MAD = \frac{\sum |Et|}{n} \dots\dots\dots \text{Persamaan 2.6}$$

Keterangan :

Et = Nilai *absolute* kesalahan

n = Jumlah data

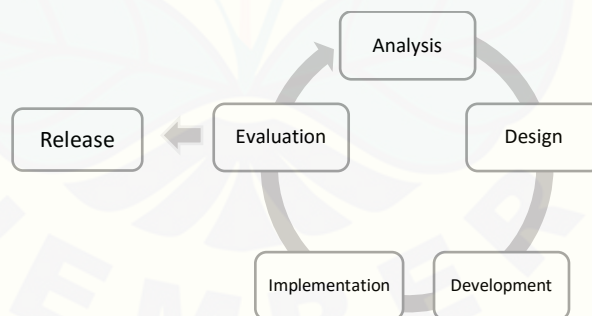
BAB 3

METODOLOGI PENELITIAN

Pada bagian ini membahas mengenai metode penelitian pembangunan Sistem Perencanaan Jadwal Produksi di Pusat Oleh-oleh Purnama Jati dan memaparkan tentang langkah atau prosedur yang dilakukan dalam mengumpulkan data atau informasi empiris guna menyelesaikan permasalahan dalam penelitian ini.

3.1 Pengembangan Sistem

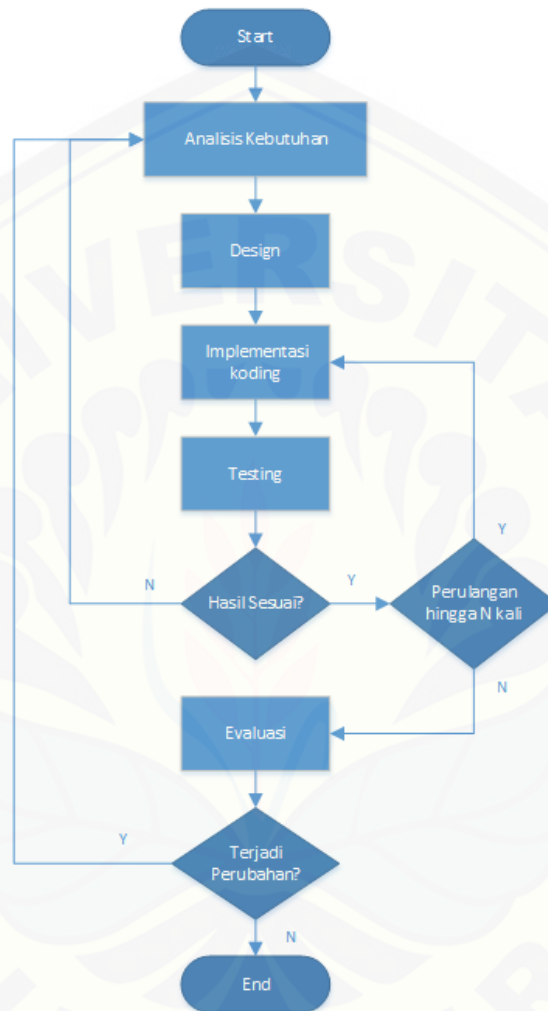
Pengembangan sistem pada penelitian ini menggunakan Model SDLC *Agile*. Model *Agile* merupakan aktifitas pembangunan perangkat lunak yang menekankan pada aktifitas konstruksi (desain dan koding). Model ini mengeliminasi sebagian besar waktu untuk melakukan perencanaan sistem dan berusaha sebisa mungkin mematuhi jadwal deliver sistem yang telah dijanjikan. *Requirements* yang dibutuhkan secara langsung di-drive oleh pelanggan itu sendiri, dan apabila terjadi perubahan terhadap requirements tersebut, pengembang dituntut mampu beradaptasi dengan perubahan yang terjadi. Gambar 3.1 merupakan bagan Model *Agile*.



Gambar 3.1 Model SDLC Agile

Model *Agile* juga dapat diartikan sekelompok metodologi pengembangan software yang didasarkan pada prinsip-prinsip yang sama atau pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dari pengembang terhadap perubahan dalam bentuk apapun. *Agile Method* digunakan untuk pengembangan proses yang membutuhkan waktu cepat dan pengerjaannya tidak rumit. Metode ini menyediakan cara untuk mengembangkan perangkat lunak yang cepat dan

mengakomodir perubahan dari bagian apapun saat proses pengemaban sistem. (Amir, Khan, Khan, & Khan, 2013). Gambar 3.2 merupakan flowchart dari model *Agile*.



Gambar 3.2 *Flowchart Model Agile*

3.1.1 Analisis Kebutuhan

1. Pengumpulan Data

Pengumpulan data penelitian ini adalah dengan wawancara dan dokumentasi. Wawancara dilakukan di Purnama Jati untuk mengetahui proses produksi dan jumlah penjualan per periodenya. Dokumentasi merupakan pencarian data-data yang dibutuhkan untuk pengembangan sistem.

2. Analisis Data

Data yang telah dikumpulkan akan dianalisis kesesuaiannya dengan kebutuhan. Dan apabila data yang diperoleh tidak sesuai atau kurang maka dilakukan wawancara lagi dengan tujuan mencari data yang tepat.

3.1.2 Desain Sistem

Design sistem dilakukan dengan berdasarkan pada data yang telah diperoleh dan dianalisis. *Design* sistem menggambarkan proses data diolah sedemikian rupa sehingga sistem dapat berjalan dengan baik dan tepat. Pembuatan desain sistem menggunakan *Unified Modeling Language* yang dirancang menggunakan konsep *Object Oriented Programming*. Berikut Pemodelan UML yang digunakan yaitu :

1. Bisnis Proses

Bisnis proses menjelaskan tentang data-data yang digunakan pada sistem. Data tersebut dikelompokkan menjadi data *input* dan data *output*.

2. Use Case

Use Case menggambarkan fitur-fitur yang dapat digunakan oleh user.

3. Scenario Use Case

Scenario Use Case merupakan penjelasan dari alur kegiatan aktor dan respon dari sistem.

4. Activity Diagram

Activity Diagram menunjukkan apa saja yang dilakukan user dengan tampilan-tampilan yang telah disediakan untuk user. *User* dapat memilih menu-menu tertentu dan melakukan hal-hal tertentu dan sistem akan merespon hasilnya.

5. Sequence Diagram

Sequence Diagram menjelaskan method-method yang diproses pada kelas tertentu ketika user melakukan suatu aktivitas.

6. Class Diagram

Class Diagram menunjukkan hubungan antar kelas yang dibuat pada sistem.

7. Entity Class Relationship

ERD menampilkan hubungan *database* tiap-tiap tabel satu sama lain adakah hubungan atau jika ada hubungan apa yang digunakan.

3.1.3 Implementasi

Dari desain yang telah dibuat diimplementasikan menjadi kode program. Beberapa hal yang dilakukan pada tahap implementasi:

1. Penulisan kode dengan menggunakan bahasa pemrograman PHP
2. Manajemen basis data menggunakan DBMS Mysql
3. Windows 8.1 64 bit

3.1.4 Pengujian

1. *White Box*

White Box Testing adalah metode pengujian yang menguji modul-modul dalam sistem. Pengujian ini menggunakan perhitungan *Cyclomatic Complexity* (CC). *Cyclomatic Complexity* adalah pengukuran untuk mengukur dan mengontrol jumlah path pada sebuah program. *Cyclomatic Complexity* dapat menemukan permasalahan-permasalahan bersarang atau permasalahan kerumitan program. Sebuah program dengan tingkat kerumitan yang tinggi akan berkemungkinan besar memiliki jumlah path yang tidak terbatas. (McCabe, 1976). Rumus CC dapat dilihat pada Persamaan (3.1).

$$CC = (E - N) + 2 \dots\dots\dots \text{Persamaan 3.1}$$

Keterangan :

E = Edge

N = Node

2. *Black Box*

Black Box digunakan untuk menemukan kesalahan dalam beberapa kategori yaitu fungsi tidak benar atau hilang, kesalahan *interface* atau antarmuka, kesalahan dalam struktur data atau akses *database* eksternal, kesalahan kinerja atau perilaku, kesalahan inisialisasi dan terminasi (Tenia Wahyuningrum, 2015).

3.1.5 *Maintanance*

Perbaikan dilakukan apabila terdapat kesalahan pada sistem saat dilakukan pengujian. Perbaikan yang dilakukan meliputi perbaikan koding, perbaikan desain dan kesalahan penghitungan.

3.2 Analisis Metode *Winter Exponential Smoothing*

Metode *Winter Exponential Smoothing* (WES) digunakan untuk meramalkan data berpola trend, musiman dan stasioner. Dengan melakukan penghitungan dengan 3 estimasi yaitu stasioner, trend dan musiman dalam 1 data. Pada metode ini akan dilihat berapa nilai estimasi yang dimiliki oleh data tersebut untuk setiap estimasi. Jika estimasi trend bernilai negatif maka trend mengalami penurunan sedangkan jika bernilai positif maka trend mengalami peningkatan. Pada estimasi musiman jika estimasi lebih dari 1 maka terpengaruh musiman dan jika kurang dari 1 tidak terpengaruh musiman.

Metode WES diuji dengan menghitung tingkat *error* yang dimiliki oleh hasil peramalan WES. Semakin tinggi *error* yang dimiliki maka semakin tidak akurat hasil peramalannya. Kesalahan peramalan dihitung dengan menggunakan MAPE (*Mean Absolute Percentage Error*) dan MAD (*Mean Absolute Deviation*).

3.3 Gambaran Umum Sistem

Sistem informasi peramalan penjualan dibuat dengan tujuan untuk mempermudah Purnama Jati dalam memprediksi jumlah produk yang terjual pada periode selanjutnya.

Sistem ini memiliki 3 user yaitu bagian produksi, bagian marketing, dan bagian admin. Bagian marketing melakukan masukan penjualan yang nantinya akan diramalkan. Bagian produksi memasukkan produk yang akan diramalkan dan harga tiap produk. Bagian admin mengawasi semua kegiatan pada sistem dan membuat akun pengguna.

BAB 4

PENGEMBANGAN SISTEM

Bab ini berisi tentang pengembangan sistem dan penerapan metode *Winter Exponential Smoothing* dalam meramalkan penjualan. Tahap pengembangan sistem dilaksanakan berdasarkan Model *Agile*, dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, pembuatan desain sistem, penulisan kode program, dan pengujian sistem.

4.1 Analisis Kebutuhan Sistem

Dalam penentuan kebutuhan sistem diperlukan hasil pengumpulan data dan analisis data yang telah dilakukan sebelumnya. Pada penelitian ini peneliti melakukan 2 cara dalam mengumpulkan data yang pertama yaitu wawancara dan yang kedua yaitu dokumentasi data. Transkrip wawancara dijelaskan pada Lampiran F. Dari hasil wawancara pada Purnama Jati, Purnama Jati mengalami sebuah permasalahan dimana setiap hari Purnama Jati harus mengirim produk berulang-ulang dalam sehari. Outlet-outlet distribusi meminta produk lagi dalam sehari jika kehabisan produk. Dan hal ini sering kali terjadi. Akibatnya Purnama Jati harus memproduksi berulang kali juga dalam sehari. Hal ini tentu membuang waktu dan biaya. Oleh sebab itu Purnama Jati memerlukan perencanaan yang matang dalam hal jumlah produksi agar tidak terjadi kekurangan maupun kelebihan produk. Peneliti kemudian merencanakan pembuatan sistem yang berfungsi meramalkan penjualan produk perhari sebagai salah satu saran dalam penentuan jumlah produk yang akan di produksi.

Penentuan jumlah produksi tidak hanya didasarkan pada penjualan tetapi pada pesanan yang telah dipesan oleh pembeli pada hari sebelumnya. Jumlah pesanan dapat mempengaruhi perencanaan jumlah produksi sehingga bagian produksi dapat memperkirakan jumlah produk yang akan diproduksi. Agar penjualan per hari dapat terpenuhi dan pesanan juga dapat terpenuhi.

Dengan demikian peneliti merumuskan apa saja yang dibutuhkan untuk dapat membangun sistem dengan tujuan membantu menyelesaikan permasalahan

Purnama Jati. Untuk dapat membangun sistem dibutuhkan data-data. Dalam permasalahan ini data utama yang dibutuhkan adalah data penjualan dan data pesanan. Untuk menunjang data-data tersebut dibutuhkan data-data lain yaitu data produk dan data produksi.

Dari hasil perumusan kebutuhan data, peneliti melakukan dokumentasi data dari Purnama Jati. Data yang diperoleh pada penelitian ini adalah data penjualan, data pesanan, data produksi dan data produk. Serta peneliti juga mendapat penjelasan tentang bagian-bagian apa saja dalam pengelolaan yang dapat mengakses sistem. Bagian yang dapat mengakses sistem adalah bagian produksi, bagian marketing dan admin (pemilik). Dari hasil analisis data didapatkan bahwa data yang diperoleh sudah sesuai kebutuhan dalam pembangunan sistem.

Setelah peneliti mendapat data dan menganalisisnya didapatkan beberapa kebutuhan sistem yang meliputi kebutuhan fungsional dan non fungsional sistem.

4.1.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan proses-proses yang dilakukan oleh sistem. Dan kebutuhan fungsional pada sistem ini adalah sebagai berikut :

1. Sistem dapat mengelola data diri
2. Sistem dapat mengelola data pengguna
3. Sistem dapat mengelola data penjualan
4. Sistem dapat mengelola data hasil peramalan penjualan
5. Sistem dapat mengelola data produk
6. Sistem dapat mengelola data produksi
7. Sistem dapat mengelola data pesanan

4.1.2 Kebutuhan Non Fungsional

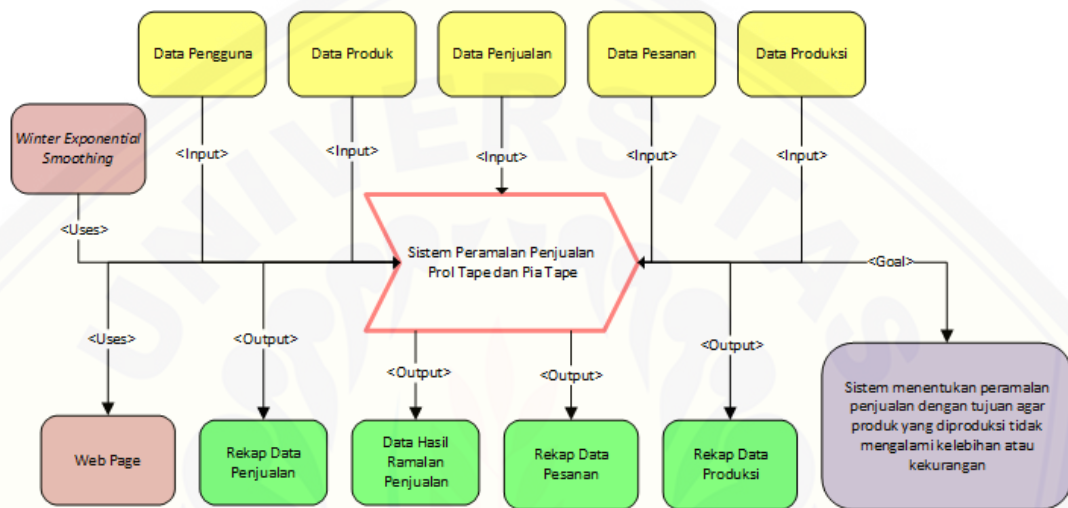
Kebutuhan non fungsional berisi tentang proses yang mendukung kebutuhan fungsional. Kebutuhan non fungsional sistem ini adalah :

1. Sistem online selama 24 jam
2. Sistem mampu menampilkan *user interface* yang *user friendly*
3. Sistem memiliki batasan hak akses menggunakan *username* dan *password*

4.2 Desain Sistem

4.2.1 Business Process

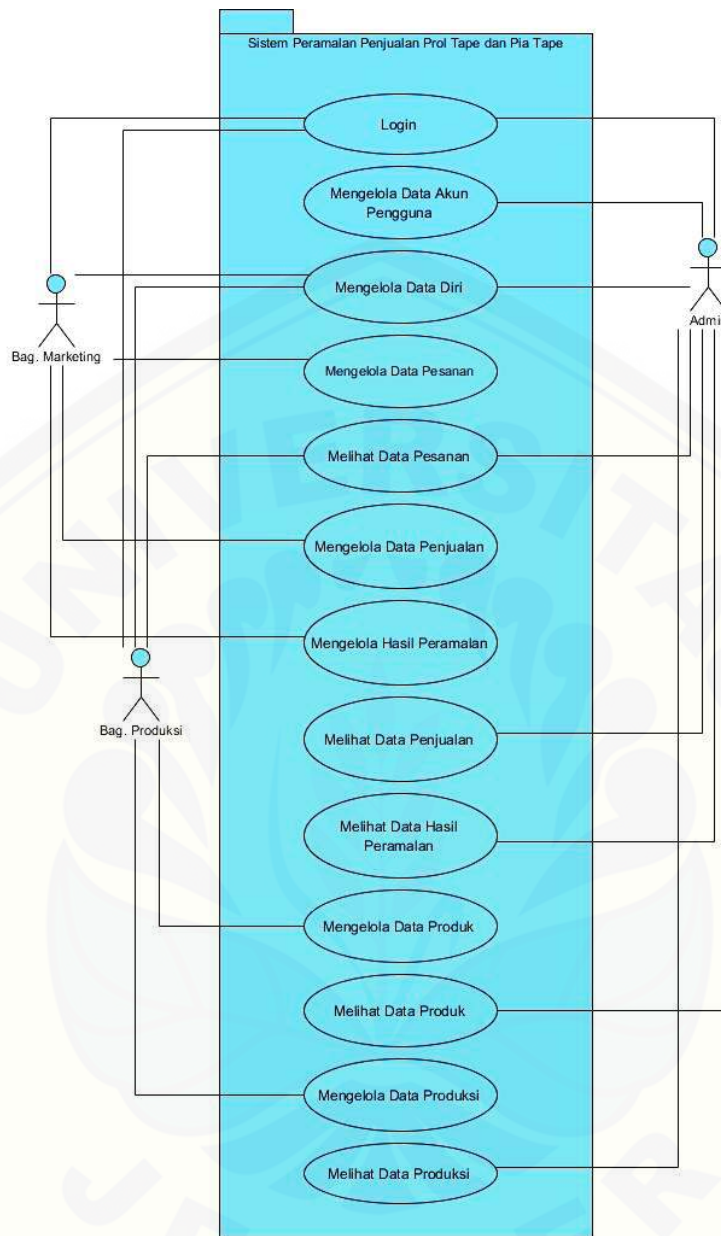
Business Process merupakan diagram yang menjelaskan tentang data-data yang menjadi masukan (*input*), keluaran (*output*), media yang digunakan (*uses*), tujuan pembuatan sistem (*goal*) dan data pendukung (*supply*) untuk sistem. Bisnis proses pada sistem ini dapat dilihat pada Gambar 4.1.



Gambar 4.1 *Business Process*

4.2.2 Use Case Diagram

Diagram *Use Case* merupakan diagram untuk menjelaskan fitur-fitur pada sistem dan aktor-aktor yang mengakses fitur tersebut. Diagram *Use Case* pada sistem ini dapat dilihat pada Gambar 4.2.



Gambar 4.2 Use Case Diagram

Penjelasan tentang *Use Case* meliputi penjelasan tentang aktor dan *Use Case* dijelaskan pada Tabel 4.1 dan Tabel 4.2.

1. Aktor

Penjelasan tentang kerja aktor pada sistem ini adalah sebagai berikut :

Tabel 4.1 Definisi Aktor

No	Aktor	Penjelasan
1	Admin	Aktor admin memiliki hak akses untuk mengelola data pengguna sistem. Meliputi membuat akun, menonaktifkan akun ataupun mengubah akun, mengawasi kegiatan yang dikelola bagian lain.
2	Marketing	Marketing memiliki hak akses yang berhubungan dengan penjualan produk meliputi pengelolaan data penjualan, mengelola data hasil peramalan penjualan, dan mengelola data pesanan.
3	Produksi	Bagian produksi memiliki hak akses untuk mengelola produk yang ada di dalam sistem, dan mengelola data produksi.

2. Use Case

Penjelasan dari masing-masing *Use Case* yang terdapat dalam sistem ini adalah sebagai berikut :

Tabel 4.2 Definisi Proses

No	Use Case	Penjelasan
1	<i>Login</i>	Proses autentifikasi masuk ke akun pengguna
2	Mengelola Akun Pengguna	Proses ini berisi pengelolaan akun yang meliputi pembuatan akun, <i>edit</i> akun dan menonaktifkan akun pengguna
3	Mengelola Data Diri	Proses ini berisi penggantian <i>password user</i> oleh setiap <i>user</i> saat pertama kali <i>login</i>
4	Mengelola Data Pesanan	Menggambarkan tentang pengelolaan data pesanan meliputi tambah, <i>edit</i> dan hapus data

Dilanjutkan

Lanjutan

No	Use Case	Penjelasan
5	Melihat Data Pesanan	Menggambarkan proses menampilkan daftar data pesanan
6	Mengelola Data Penjualan	Menggambarkan pengelolaan data penjualan yang berisi <i>edit</i> data penjualan
7	Melihat Data Penjualan	Proses ini hanya berisi tentang daftar penjualan tanpa dapat diubah-ubah isinya
8	Mengelola Hasil Peramalan	Proses ini menjelaskan tentang pengelolaan hasil peramalan yang berupa penghitungan hasil peramalan penjualan
9	Melihat Hasil Peramalan	Proses ini hanya berisi tentang daftar hasil peramalan
10	Mengelola Data Produk	Menggambarkan proses pengelolaan data produk meliputi penambahan data produk, <i>edit</i> data produk dan menghapus data produk
11	Melihat Data Produk	Proses ini hanya melihat daftar produk yang telah dibuat oleh aktor produksi
12	Mengelola Data Produksi	Menggambarkan proses pengelolaan data produksi meliputi tambah, <i>edit</i> dan <i>delete</i> data
13	Melihat Data Produksi	Melihat daftar produksi dan masa kadaluwarsa produk

4.2.3 Use Case Scenario

1. Skenario Use Case Login

Skenario *login* menjelaskan tentang proses autentikasi pengguna untuk dapat masuk kedalam sistem. Proses ini dilakukan oleh semua pengguna dan hanya berisi masukan *username* dan *password* pengguna. Pada skenario dijelaskan alur *login* dan alternatif apa saja yang terjadi saat melakukan *Login*. Untuk detail skenario *login* akan ditampilkan pada Tabel A.1 pada Lampiran A.

2. Skenario Use Case Mengelola Data Akun Pengguna

Skenario mengelola data pengguna dijalankan oleh admin. Yang berisi proses pembuatan akun pengguna meliputi *input username* dan *password* untuk pengguna baru. Serta mengubah akun pengguna. Akan tetapi fitur *edit* pengguna hanya terbatas pada mengubah status akun. Hanya dapat mengubah akun aktif atau non aktif. Selain itu admin juga dapat menghapus akun pengguna. Detail skenario dapat dilihat pada Tabel 4.3, Tabel 4.4, dan Tabel 4.5.

Tabel 4.3 Skenario Menambah Data Akun Pengguna

Nomor Use Case	Use Case 2
Nama Use Case	Mengelola Akun Pengguna – Input – Admin
Aktor	Admin
Deskripsi	Aktor memasukkan data pengguna baru
Pre Kondisi	Aktor berada di halaman dashboard
Pra Kondisi	Aktor Menambah Akun Pengguna
Normal Flow	
Aksi Aktor	Reaksi Sistem
1. Memilih menu Akun	
	2. Mengambil data_users, data_role, dan data_status
	3. Menampilkan halaman users/index
4. Memilih tombol Tambah Akun	
5. Mengisi form informasi akun baru	
6. Memilih tombol Simpan	
	7. Menambahkan data user(username, password, role, dan status) ke database
	8. Menampilkan halaman users/index
Alternatife Flow – Blank Field	
	7. Menampilkan popover "Please Fill Out This Field"
	8. Menampilkan halaman users/index

Tabel 4.4 Skenario Mengubah Data Akun Pengguna

Nomor Use Case	Use Case 2
Nama Use Case	Mengelola Akun Pengguna – Edit – Admin
Aktor	Admin
Deskripsi	Aktor Mengubah Data Akun Pengguna
Pre Kondisi	Aktor berada di halaman dashboard
Pra Kondisi	Aktor Mengubah Data Akun Pengguna
Normal Flow	
Aksi Aktor	Reaksi Sistem
1. Memilih menu Akun	
	2. Mengambil data_users, data_role, dan data_status
	3. Menampilkan halaman users/index
4. Memilih tombol edit	
	2. Mengambil data_users, data_role, dan data_status dengan id yang dipilih
	3. Menampilkan halaman users/edit
5. Mengubah form informasi akun yang dipilih	
6. Memilih tombol Simpan	
	7. Mengubah data user(status) dengan id yang dipilih ke database
	8. Menampilkan halaman users/index

Tabel 4.5 Skenario Menghapus Data Akun Pengguna

Nomor Use Case	Use Case 2
Nama Use Case	Mengelola Akun Pengguna – Delete – Admin
Aktor	Admin
Deskripsi	Aktor mengelola data akun pengguna
Pre Kondisi	Aktor berada di halaman dashboard
Pra Kondisi	Aktor Menghapus Data Akun Pengguna
Normal Flow	
Aksi Aktor	Reaksi Sistem
1. Memilih menu Akun	

Dilanjutkan

Lanjutan

Aksi Aktor	Reaksi Sistem
	2. Mengambil data_users, data_role, dan data_status
	3. Menampilkan halaman users/index
4. Memilih tombol Delete	
	2. Menghapus data pengguna dengan id yang dipilih
	3. Menampilkan halaman users/index

3. Skenario Use Case Mengelola Data Diri

Skenario mengelola data diri dilakukan oleh semua aktor. Pada proses ini aktor disarankan untuk melakukan penggantian *password* sebagai bentuk keamanan terhadap data dan akses pengguna. Sehingga tidak ada pengguna lain yang dapat mengakses akun lain. Peringatan akan muncul ketika pengguna pertama kali melakukan *login*. Detail skenario mengelola data diri akan ditampilkan pada Tabel 4.6, Tabel 4.7, dan Tabel 4.8.

Tabel 4.6 Skenario Mengelola Data Diri (Admin)

Nomor Use Case	Use Case 3
Nama Use Case	Mengelola Data Diri – Admin
Aktor	Admin
Deskripsi	Aktor mengganti password akun
Pre Kondisi	Aktor berada di halaman dashboard, Aktor belum mengganti password
Pra Kondisi	Aktor mengganti password
Normal Flow	
Aksi Aktor	Reaksi Sistem
1. Memilih menu akun profil	
2. Memilih menu setting	
	3. Mengambil data sesuai id user
	4. Menampilkan halaman changepass
5. Mengisi form	
6. Memilih simpan	
	7. Mengubah password sesuai dengan id user
	8. Menampilkan halaman login

Dilanjutkan

Lanjutan

Alternatif Flow – Data Kosong	
	7. Menampilkan popover”Please fill out this field”
	8. Menampilkan halaman changepass
Alternatif Flow – Password Konfirmasi Salah	
	7. Menampilkan alert”Password Confirmation Doesn’t Match”
	8. Menampilkan halaman changepass

Tabel 4.7 Skenario Mengelola Data Diri (Produksi)

Nomor Use Case	Use Case 3
Nama Use Case	Mengelola Data Diri – Produksi
Aktor	Produksi
Deskripsi	Aktor mengganti password akun
Pre Kondisi	Aktor berada di halaman dashboard, Aktor belum mengganti password
Pra Kondisi	Aktor mengganti password
Normal Flow	
Aksi Aktor	Reaksi Sistem
1. Mengklik nama akun	
2. Mengklik menu setting	
	3. Mengambil data sesuai id user
	4. Menampilkan halaman changepass
5. MengInput form	
6. Mengklik simpan	
	7. Mengubah password sesuai dengan id user
	8. Menampilkan halaman Login
Alternatif Flow	
	7. Menampilkan popover”Please fill out this field”
	8. Menampilkan halaman changepass
Alternatif Flow – Password Konfirmasi Salah	
	7. Menampilkan alert”Password Confirmation Doesn’t Match”
	8. Menampilkan halaman changepass

Tabel 4.8 Skenario Mengelola Data Diri (Marketing)

Nomor Use Case	Use Case 3
Nama Use Case	Mengelola Data Diri – Marketing
Aktor	Marketing
Deskripsi	Aktor mengganti password akun
Pre Kondisi	Aktor berada di halaman dashboard, Aktor belum mengganti password
Pra Kondisi	Aktor mengganti password
Normal Flow	
Aksi Aktor	Reaksi Sistem
1. Mengklik nama akun	
2. Mengklik menu setting	
	3. Mengambil data sesuai id user
	4. Menampilkan halaman changepass
5. MengInput form	
6. Mengklik simpan	
	7. Mengubah password sesuai dengan id user
	8. Menampilkan halaman login
Alternatif Flow	
	7. Menampilkan popover”Please fill out this field”
	8. Menampilkan halaman changepass
Alternatif Flow – Password Konfirmasi Salah	
	7. Menampilkan alert”Password Confirmation Doesn’t Match”
	8. Menampilkan halaman changepass

4. Skenario Use Case Mengelola Data Pesanan

Skenario mengelola data pesanan dijalankan oleh aktor marketing. Bagian marketing menerima daftar pesanan untuk hari-hari kedepan dari pembeli dan memasukkan kedalam sistem. Proses ini berisi *input* pesanan, *edit* pesanan dan *delete* pesanan apabila pesanan dibatalkan. Detail skenario dapat dilihat pada Tabel 4.9, Tabel 4.10, dan Tabel 4.11.

Tabel 4.9 Skenario Mengelola Data Pesanan (*Input*)

Nomor Use Case	Use Case 4
Nama Use Case	Mengelola Data Pesanan – Input – Marketing
Aktor	Marketing
Deskripsi	Aktor mengelola data pesanan
Pre Kondisi	Aktor berada di halaman dashboard
Pra Kondisi	Aktor Menambah data pesanan
Normal Flow	
Aksi Aktor	Reaksi Sistem
1. Memilih menu pesanan	
	2. Mengambil data_order
	3. Menampilkan halaman order/index
4. Memilih tombol Tambah	
5. Mengisi form pesanan	
6. Memilih tombol submit	
	7. Menambahkan data_order(product, date, amount) ke database
	8. Menampilkan halaman order/index
Alternatife Flow – Blank Field	
	7. Menampilkan popover "Please Fill Out This Field"
	8. Menampilkan halaman order/index

Tabel 4.10 Skenario Mengelola Data Pesanan (*Edit*)

Nomor Use Case	Use Case 4
Nama Use Case	Mengelola Data Pesanan – Edit – Marketing
Aktor	Marketing
Deskripsi	Aktor mengelola data pesanan
Pre Kondisi	Aktor berada di halaman dashboard
Pra Kondisi	Aktor Mengubah Data Pesanan
Normal Flow	

Dilanjutkan

Lanjutan

Aksi Aktor	Reaksi Sistem
1. Mengklik menu pesanan	
	2. Mengambil data_order
	3. Menampilkan halaman order/index
4. Mengklik tombol edit	
	5. Mengambil data_order
	6. Menampilkan halaman order/edit
7. Mengubah form	
8. Mengklik tombol submit	
	9. Mengubah data_order dengan id yang dipilih ke database
	10. Menampilkan halaman order/index
Alternatife Flow – Blank Field	
	9. Menampilkan popover "Please Fill Out This Field"
	10. Menampilkan halaman order/edit

Tabel 4.11 Skenario Mengelola Data Pesanan (Hapus)

Nomor Use Case	Use Case 4
Nama Use Case	Mengelola Data Pesanan – Delete – Marketing
Aktor	Marketing
Deskripsi	Aktor mengelola data pesanan
Pre Kondisi	Aktor berada di halaman dashboard
Pra Kondisi	Aktor Menghapus Data Pesanan
Normal Flow	
Aksi Aktor	Reaksi Sistem
1. Mengklik menu pesanan	
	2. Mengambil data_order
	3. Menampilkan halaman order/index
4. Mengklik tombol delete	
	2. Menghapus data_order dengan id yang dipilih
	3. Menampilkan halaman order/index

5. Skenario *Use Case* Melihat Data Pesanan

Skenario melihat data pesanan dilakukan oleh admin dan produksi. Admin melihat daftar pesanan sebagai bentuk pengawasan terhadap aktifitas bagian lain. Dan produksi melihat data pesanan untuk merencanakan jumlah produk yang akan diproduksi pada hari dimana ada pesanan. Detail skenario melihat data pesanan dapat dilihat pada Tabel A.2 di Lampiran A.

6. Skenario *Use Case* Mengelola Data Penjualan

Skenario mengelola data penjualan dijalankan oleh marketing. Pada proses ini marketing melakukan *edit* data yang telah dibuat sebelumnya pada fitur mengelola hasil peramalan. Dan marketing memasukkan jumlah penjualan setiap hari saat semua kegiatan telah usai. Detail skenario mengelola data penjualan dapat dilihat pada Tabel 4.12.

Tabel 4.12 Skenario *Use Case* Mengelola Data Penjualan

Nomor Use Case	Use Case 6
Nama Use Case	Mengelola Data Penjualan – Marketing
Aktor	Marketing
Deskripsi	Aktor mengelola data penjualan
Pre Kondisi	Aktor berada di halaman dashboard
Pra Kondisi	Aktor Mengubah Data Penjualan
Normal Flow	
Aksi Aktor	Reaksi Sistem
1. Mengklik menu penjualan	
	2. Mengambil data_sales
	3. Menampilkan halaman sales/index
4. Mengklik tombol edit	
	5. Mengambil data_sales
	6. Menampilkan halaman sales/edit
7. Mengubah form	
8. Mengklik tombol submit	

Dilanjutkan

Lanjutan

Aksi Aktor	Reaksi Sistem
	9. Mengubah data_sales dengan id yang dipilih ke database
	10. Menampilkan halaman sales/index
Alternatife Flow – Blank Field	
	9. Menampilkan popover "Please Fill Out This Field"
	10. Menampilkan halaman sales/Edit

7. Skenario Use Case Mengelola Hasil Peramalan

Skenario mengelola hasil peramalan dijalankan oleh marketing. Proses ini berisi perhitungan peramalan yang dilakukan ketika aktor marketing menekan tombol "Hitung Peramalan" pada tampilan sistem. Sistem akan melakukan perhitungan peramalan untuk periode selanjutnya. Akan tetapi sistem akan menampilkan peringatan apabila data penjualan aktual pada hari sebelumnya belum diinputkan. Detail skenario mengelola data penjualan dapat dilihat pada Tabel 4.13.

Tabel 4.13 Skenario Mengelola Data Hasil Peramalan

Nomor Use Case	Use Case 7
Nama Use Case	Mengelola Data Hasil Peramalan
Aktor	Marketing
Deskripsi	Melakukan perhitungan hasil peramalan penjualan
Pre Kondisi	Aktor berada di halaman dashboard, Aktor belum mempunyai hasil peramalan
Pra Kondisi	Aktor Mendapatkan hasil penghitungan peramalan
Normal Flow	
Aksi Aktor	Reaksi Sistem
1. Mengklik menu Peramalan	
	2. Mengambil data_product

Dilanjutkan

Lanjutan

Aksi Aktor	Reaksi Sistem
	3. Menampilkan halaman sales/index
4. Mengklik produk	
	5. Mengambil data produk dengan id tertentu
	6. Menampilkan halaman detail produk
7. Mengklik hitung peramalan	
	9. Menambahkan peramalan kemudian mengambil data_sales
	10. Menampilkan halaman detail
Alternatife Flow – Data Sebelumnya Belum Diinputkan	
9. Menampilkan alert "Data Sebelumnya Belum Diinputkan"	
10. Menampilkan halaman detail	

8. Skenario *Use Case* Melihat Data Penjualan

Skenario melihat data penjualan dilakukan oleh admin. Proses ini digunakan sebagai bentuk pengawas admin terhadap kegiatan yang dilakukan bagian lain. Proses ini hanya menampilkan daftar penjualan setiap hari untuk setiap produk. Detail skenario dapat dilihat pada Tabel A.3 di Lampiran A.

9. Skenario *Use Case* Melihat Hasil Peramalan

Skenario melihat data penjualan dilakukan oleh admin. Proses ini digunakan sebagai bentuk pengawas admin terhadap kegiatan yang dilakukan bagian lain. Proses ini hanya menampilkan daftar peramalan setiap hari untuk setiap produk. Detail skenario dapat dilihat pada Tabel A.4 di Lampiran A.

10. Skenario *Use Case* Mengelola Data Produk

Skenario mengelola data produk dilakukan oleh aktor produksi. Bagian produksi bertugas memasukkan produk apa saja yang dikelola oleh sistem. Dan bagian produksi dapat menghapus produk atau mengubah produk. Detail skenario dapat dilihat pada Tabel 4.14, Tabel 4.15, dan Tabel 4.16.

Tabel 4.14 Skenario Mengelola Data Produk (*Input*)

Nomor Use Case	Use Case 10
Nama Use Case	Mengelola Data Produk – Input – Produksi
Aktor	Produksi
Deskripsi	Aktor melakukan pengelolaan data produk
Pre Kondisi	Aktor berada di halaman dashboard
Pra Kondisi	Aktor Menambah data produk
Normal Flow	
Aksi Aktor	Reaksi Sistem
1. Mengklik menu produk	
	2. Mengambil data_product
	3. Menampilkan halaman product/index
4. Mengklik tombol Tambah	
5. Mengisi form produk	
6. Mengklik tombol submit	
	7. Menambahkan data_product ke database
	8. Menampilkan halaman product/index
Alternatife Flow – Blank Field	
	7. Menampilkan popover "Please Fill Out This Field"
	8. Menampilkan halaman product/index

Tabel 4.15 Skenario Mengelola Data Produk (*Edit*)

Nomor Use Case	Use Case 10
Nama Use Case	Mengelola Data Produk – Edit – Produksi
Aktor	Produksi
Deskripsi	Aktor melakukan pengelolaan data produk
Pre Kondisi	Aktor berada di halaman dashboard
Pra Kondisi	Aktor Mengubah Data Produk

Dilanjutkan

Lanjutan

Normal Flow	
Aksi Aktor	Reaksi Sistem
1. Mengklik menu produk	
	2. Mengambil data_product
	3. Menampilkan halaman product/index
4. Mengklik tombol edit	
	5. Mengambil data_product
	6. Menampilkan halaman product/edit
7. Mengubah form	
8. Mengklik tombol submit	
	9. Mengubah data_product dengan id yang dipilih ke <i>database</i>
	10. Menampilkan halaman product/index
Alternatife Flow – Blank Field	
	9. Menampilkan popover "Please Fill Out This Field"
	10. Menampilkan halaman product/Edit

Tabel 4.16 Skenario Mengelola Data Produk (*Delete*)

Nomor Use Case	Use Case 10
Nama Use Case	Mengelola Data Produk – Delete – Produksi
Aktor	Produksi
Deskripsi	Aktor melakukan pengelolaan data produk
Pre Kondisi	Aktor berada di halaman dashboard
Pra Kondisi	Aktor Menghapus Data Product
Normal Flow	
Aksi Aktor	Reaksi Sistem
1. Mengklik menu produk	
	2. Mengambil data_product
	3. Menampilkan halaman product/index
4. Mengklik tombol delete	

Dilanjutkan

Lanjutan

Aksi Aktor	Reaksi Sistem
	2. Menghapus data_product dengan id yang dipilih
	3. Menampilkan halaman product/index

11. Skenario Use Case Melihat Data Produk

Skenario melihat data penjualan dilakukan oleh admin. Proses ini digunakan sebagai bentuk pengawasan admin dan mengetahui produk apa saja yang dikelola oleh sistem tersebut. Proses ini hanya menampilkan daftar produk yang diinputkan bagian produksi. Detail skenario dapat dilihat pada Tabel A.5 di Lampiran A.

12. Skenario Use Case Mengelola Data Produksi

Skenario mengelola produk yang diproduksi dilakukan oleh bagian produksi. Pada proses ini bagian produksi memasukkan jumlah produk yang diproduksi setiap harinya. Detail skenario mengelola data produksi dapat dilihat pada Tabel 4.17 di Lampiran A.

Tabel 4.17 Skenario Mengelola Data Produksi

Nomor Use Case	Use Case 12
Nama Use Case	Mengelola Data Produksi – Produksi
Aktor	Produksi
Deskripsi	Aktor melakukan pengelolaan data produksi
Pre Kondisi	Aktor berada di halaman dashboard
Pra Kondisi	Aktor Mengubah Data Produksi
Normal Flow	
Aksi Aktor	Reaksi Sistem
1. Mengklik menu penjualan	
	2. Mengambil data_production
	3. Menampilkan halaman production /index
4. Mengklik tombol edit	
	5. Mengambil data_ production
	6. Menampilkan halaman production /edit

Dilanjutkan

Lanjutan

7. Mengubah form	
8. Mengklik tombol submit	
	9. Mengubah data_production dengan id yang dipilih ke database
	10. Menampilkan halaman production /index
Alternatife Flow – Blank Field	
	9. Menampilkan popover "Please Fill Out This Field"
	10. Menampilkan halaman production /Edit

13. Skenario Use Case Melihat Data Produksi

Skenario melihat data produksi dilakukan oleh bagian admin. Proses ini sebagai bentuk pengawasan admin terhadap kegiatan yang dilakukan sistem. Detail skenario melihat data produksi dapat dilihat pada Tabel A.6 di Lampiran A.

4.2.4 Sequence Diagram

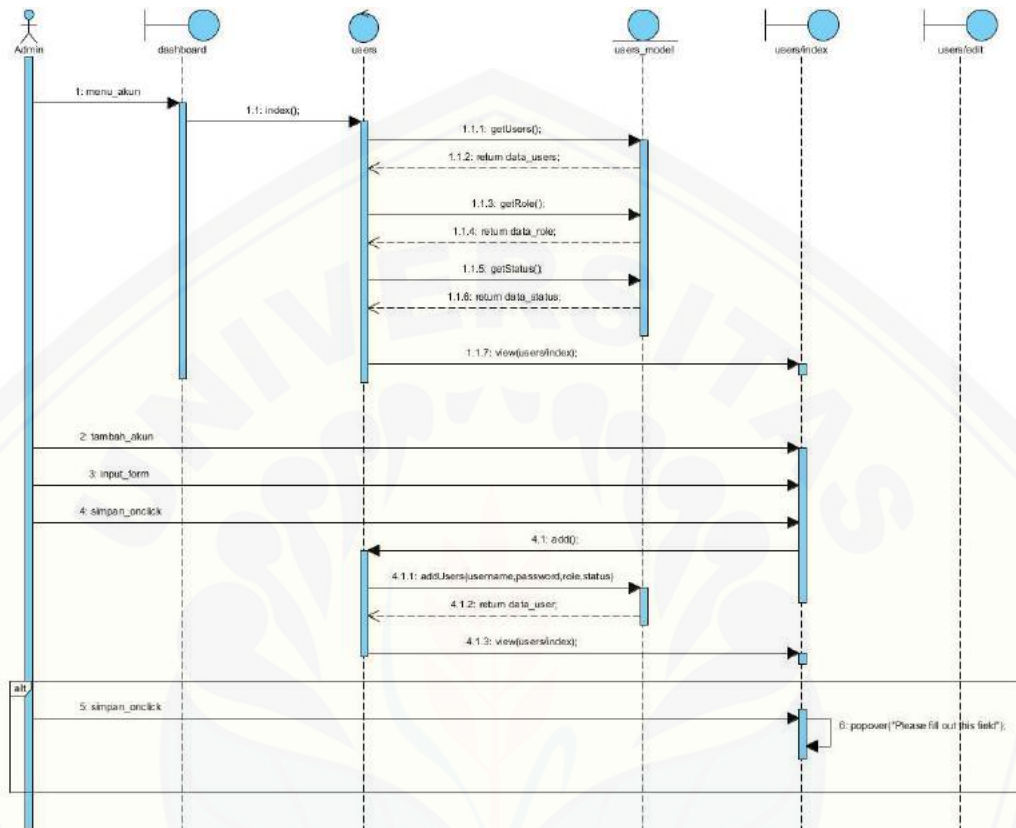
1. Sequence Diagram Login

Sequence Diagram Login menjelaskan tentang proses autentifikasi pengguna untuk dapat masuk ke dalam sistem. Method-method apa saja yang digunakan pada *controller login*, *view login* dan *model login*. Serta proses alur kode program berjalan antar method. Sehingga menghasilkan suatu proses autentifikasi. Dan pengguna berhasil masuk ke dalam sistem. Proses ini dilakukan oleh semua pengguna dan hanya berisi *input username* dan *password* pengguna. Pada *sequence* dijelaskan alur *login* dan alternatif apa saja yang terjadi saat melakukan *login*. Untuk detail *sequence diagram login* akan ditampilkan pada Gambar B.1, Gambar B.2, dan Gambar B.3 pada Lampiran B.

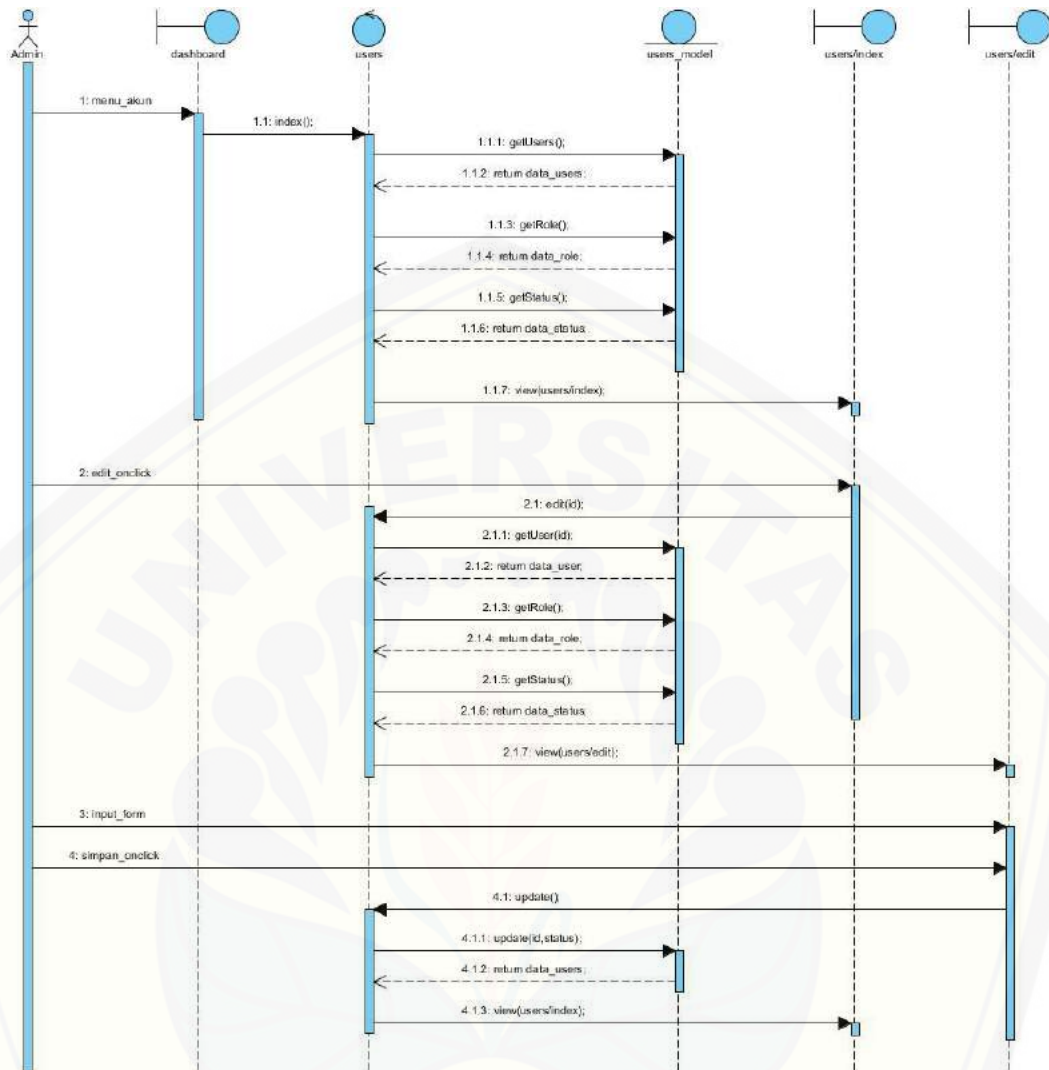
2. Sequence Diagram Mengelola Data Akun Pengguna

Sequence Diagram mengelola data akun pengguna menjelaskan tentang proses pengelolaan akun pengguna oleh admin. Method-method apa saja yang digunakan pada *controller users*, *view users* dan *model users*. Serta proses alur kode program berjalan antar method. Sehingga menghasilkan suatu proses pengelolaan data akun. Proses yang dijalankan meliputi *input* akun pengguna, *edit* akun

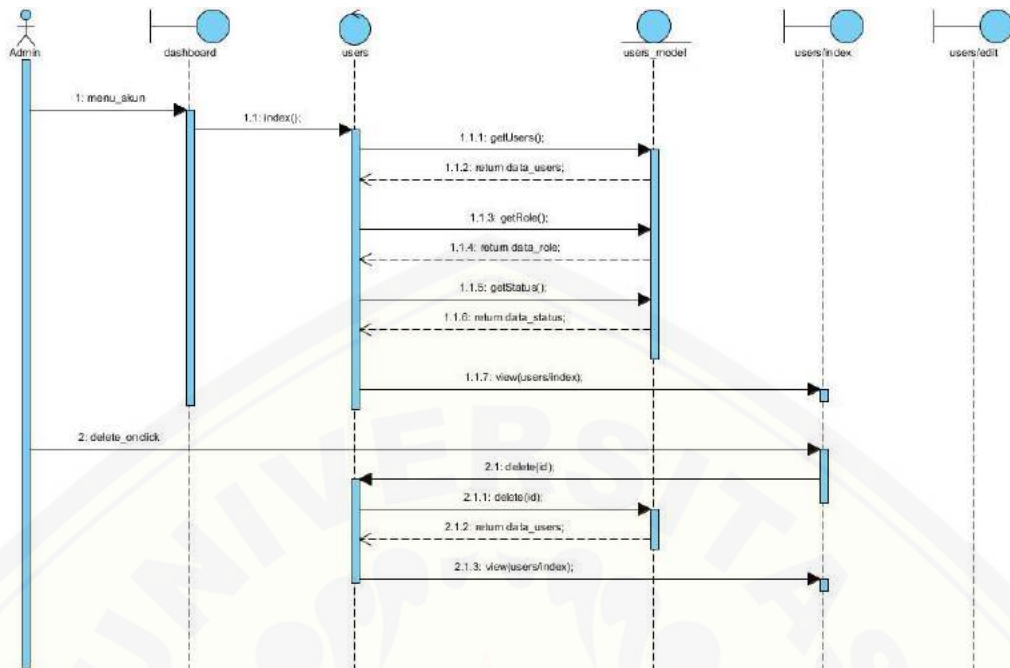
pengguna dan *delete* akun pengguna. Detail *sequence diagram* mengelola data akun pengguna akan ditampilkan pada Gambar 4.3, Gambar 4.4 dan Gambar 4.5.



Gambar 4.3 Sequeunce Mengelola Data Akun Pengguna (*Input*)



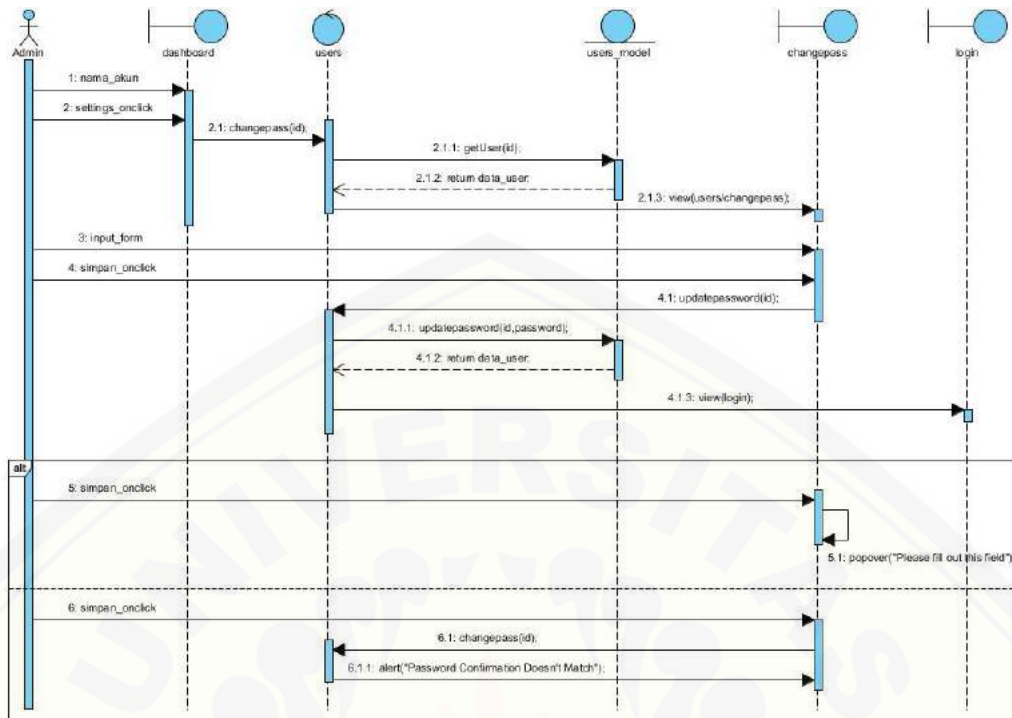
Gambar 4.4 Sequeunce Mengelola Data Akun Pengguna (*Edit*)



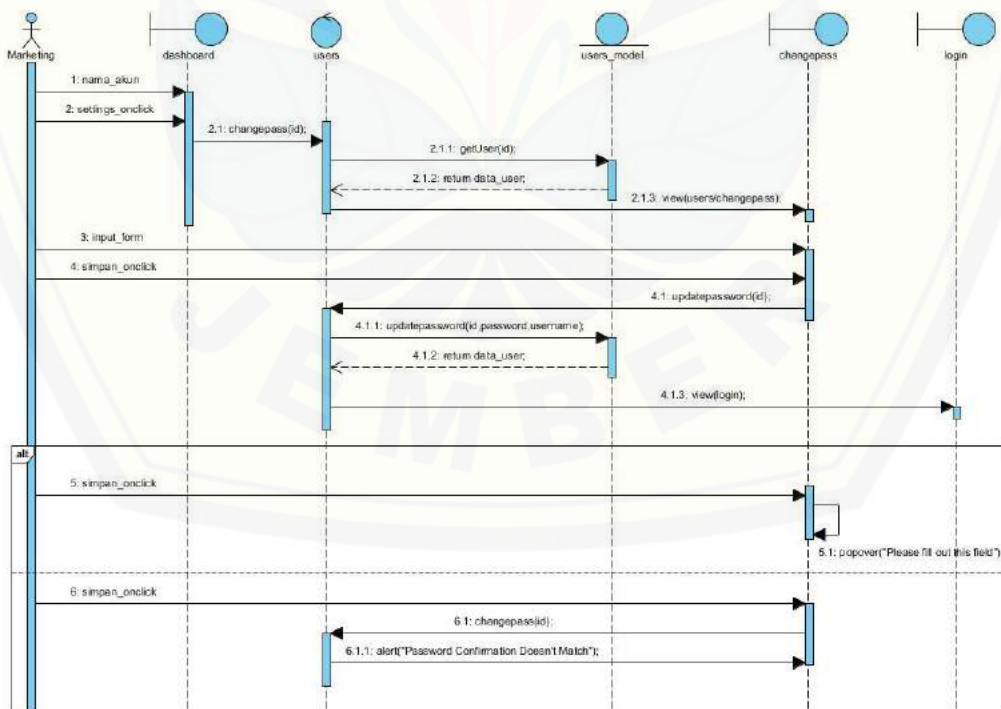
Gambar 4.5 Sequeunce Mengelola Data Akun Pengguna (*Delete*)

3. *Sequence Diagram* Mengelola Data Diri

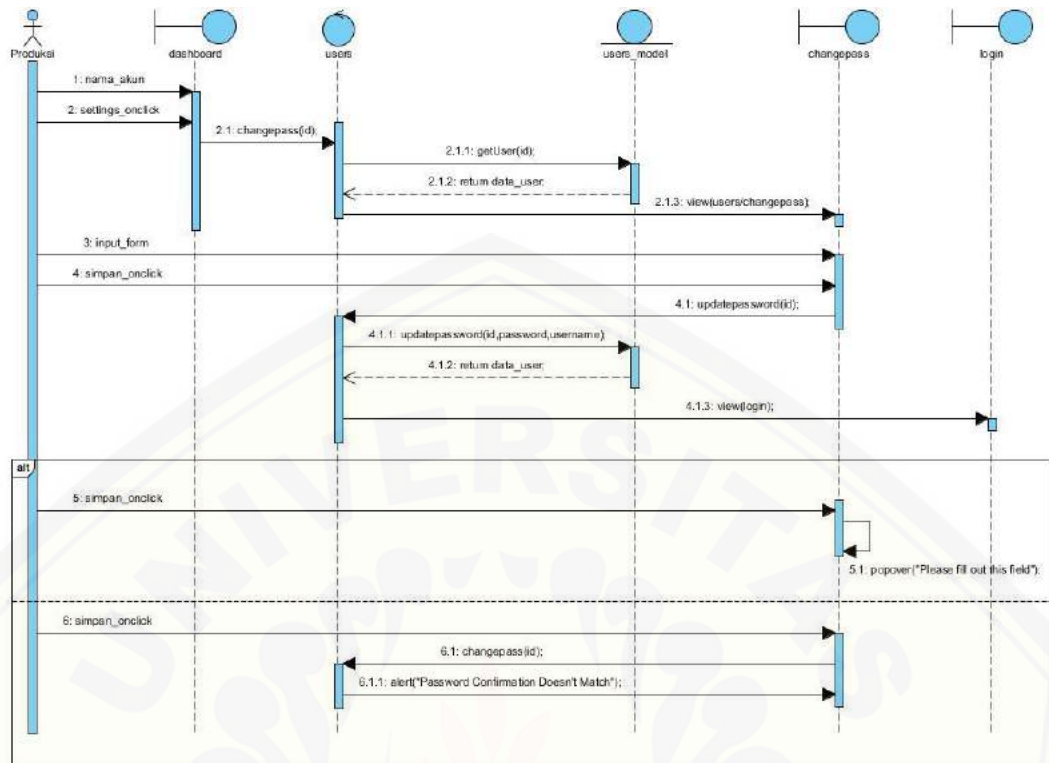
Sequence diagram mengelola data diri menjelaskan tentang proses perubahan *password*. Saat pertama kali *login* pengguna disarankan mengubah *password* untuk keamanan data dan akses ke sistem. Pada diagram *sequence* menjelaskan tentang method-method pada controller, view dan model yang digunakan untuk mengubah *password*. Detail *sequence diagram* mengelola data diri akan ditampilkan pada Gambar 4.6, Gambar 4.7 dan Gambar 4.8.



Gambar 4.6 *Sequeunce* Mengelola Data Diri (Admin)



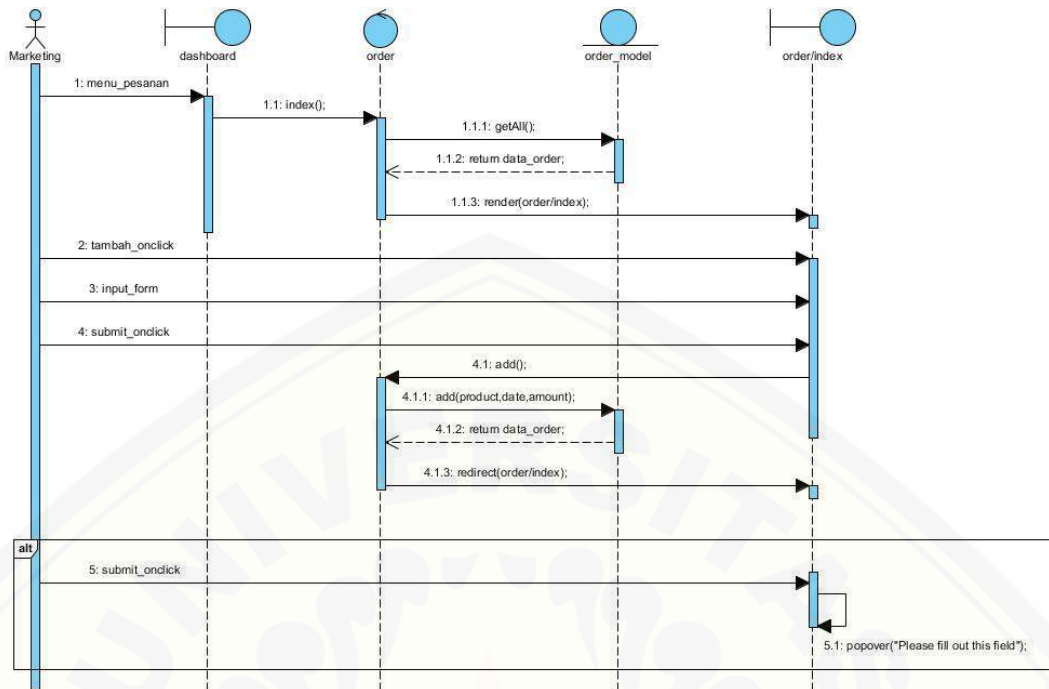
Gambar 4.7 *Sequeunce* Mengelola Data Diri (Marketing)



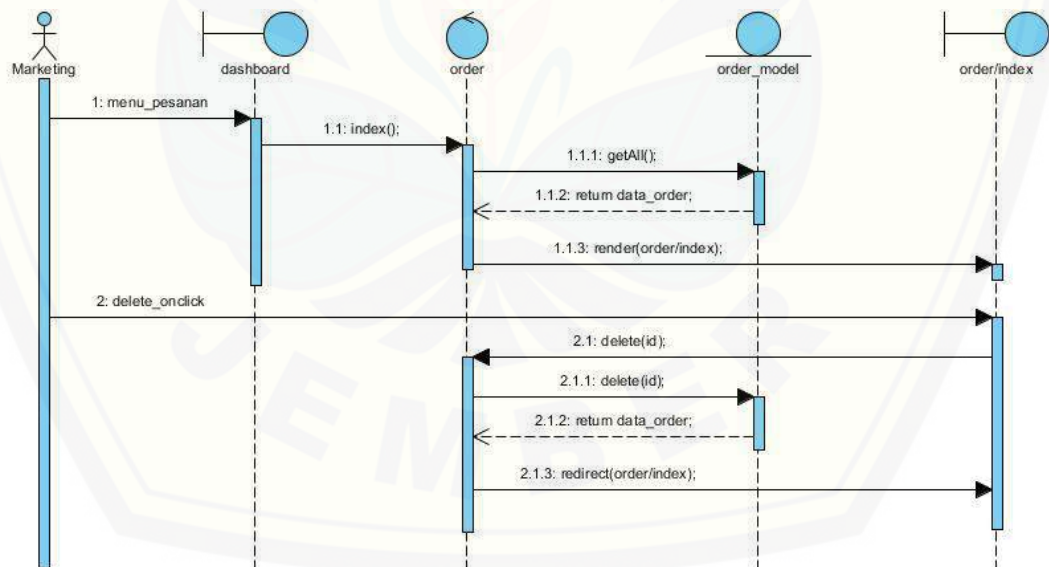
Gambar 4.8 *Sequence* Mengelola Data Diri (Produksi)

4. *Sequence Diagram* Mengelola Data Pesanan

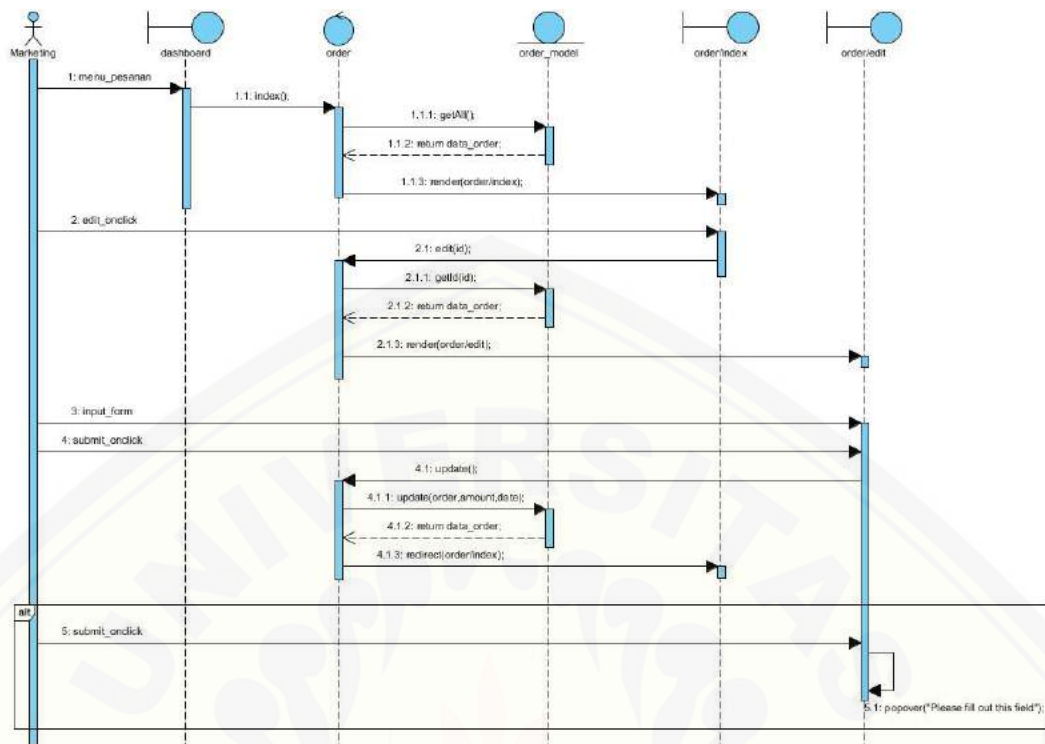
Sequence diagram mengelola data pesanan menjelaskan tentang proses pengelolaan data pesanan oleh bagian marketing. Pada *sequence* dijelaskan method apa saja yang digunakan pada controller, model dan view serta alur setiap method sehingga dapat terjadi proses *input*, *update* dan *delete* data pesanan. Detail *sequence diagram* mengelola data pesanan akan ditampilkan pada Gambar 4.9, Gambar 4.10 dan Gambar 4.11.



Gambar 4.9 Sequence Mengelola Data Pesanan (Input)



Gambar 4.10 Sequence Mengelola Data Pesanan (Delete)



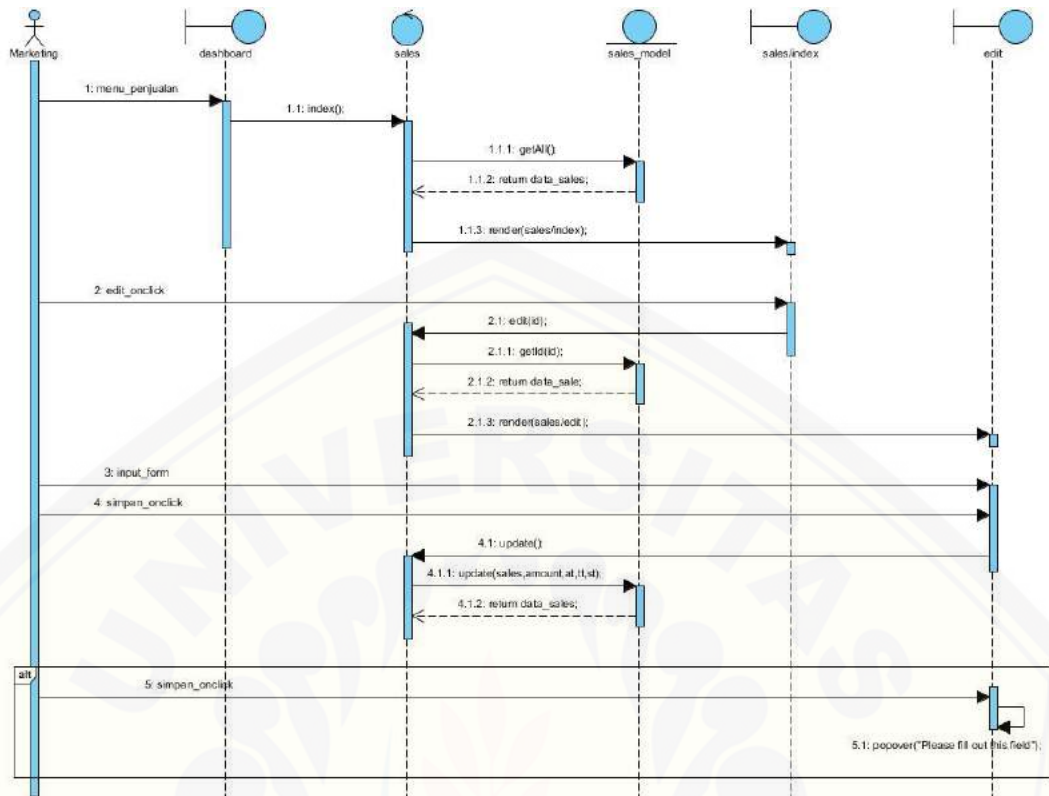
Gambar 4.11 *Sequence* Mengelola Data Pesanan (*Edit*)

5. *Sequence Diagram* Melihat Data Pesanan

Sequence Diagram melihat data pesanan menjelaskan tentang proses menampilkan data pesanan oleh admin dan bagian produksi. Pada *sequence* dijelaskan *method* apa saja yang digunakan pada *controller order*, *model orders* dan *view* pada folder *order* serta alur setiap *method* sehingga dapat terjadi proses menampilkan daftar pesanan. Detail *sequence diagram* melihat data pesanan akan ditampilkan pada Gambar B.4 dan Gambar B.5 pada Lampiran B.

6. *Sequence Diagram* Mengelola Data Penjualan

Sequence Diagram mengelola data penjualan menjelaskan tentang pengelolaan data penjualan yang berupa *input* data penjualan per hari. Pada *sequence* ditampilkan *method* yang digunakan untuk *update* data penjualan. Nilai data penjualan ini akan digunakan untuk menghitung peramalan penjualan pada hari berikutnya. Detail *sequence diagram* mengelola data penjualan akan ditampilkan pada Gambar 4.12.



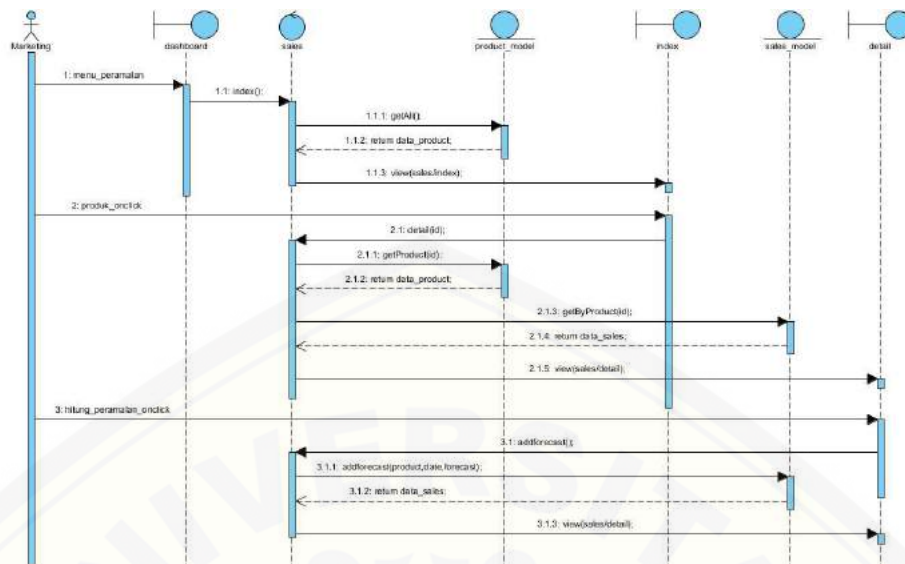
Gambar 4.12 Sequence Mengelola Data Penjualan

7. Sequence Diagram Melihat Data Penjualan

Sequence Diagram melihat data penjualan menjelaskan tentang proses menampilkan data penjualan yang telah dimasukkan bagian marketing oleh admin. Pada *sequence* dijelaskan method-method dan kelas-kelas yang digunakan dalam proses menampilkan data penjualan. Detail *sequence diagram* mengelola data penjualan akan ditampilkan pada Gambar B.6 pada Lampiran B.

8. Sequence Diagram Mengelola Hasil Peramalan

Sequence diagram mengelola hasil peramalan menjelaskan tentang proses perhitungan peramalan penjualan. Pada *sequence* dijelaskan tentang *method* dan kelas yang digunakan dalam perhitungan peramalan. Dalam kelas dan *method* tersebut berisi proses perhitungan peramalan. Detail *sequence diagram* mengelola data hasil peramalan akan ditampilkan pada Gambar 4.13.



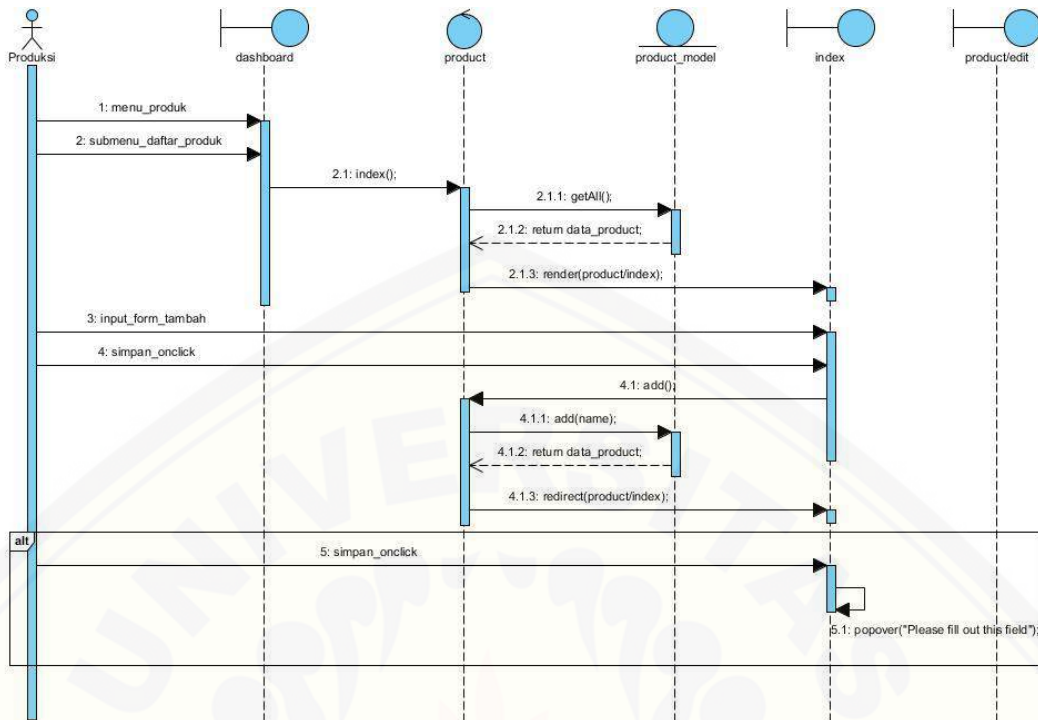
Gambar 4.13 *Sequence Diagram* Mengelola Data Hasil Peramalan

9. *Sequence Diagram* Melihat Data Hasil Peramalan

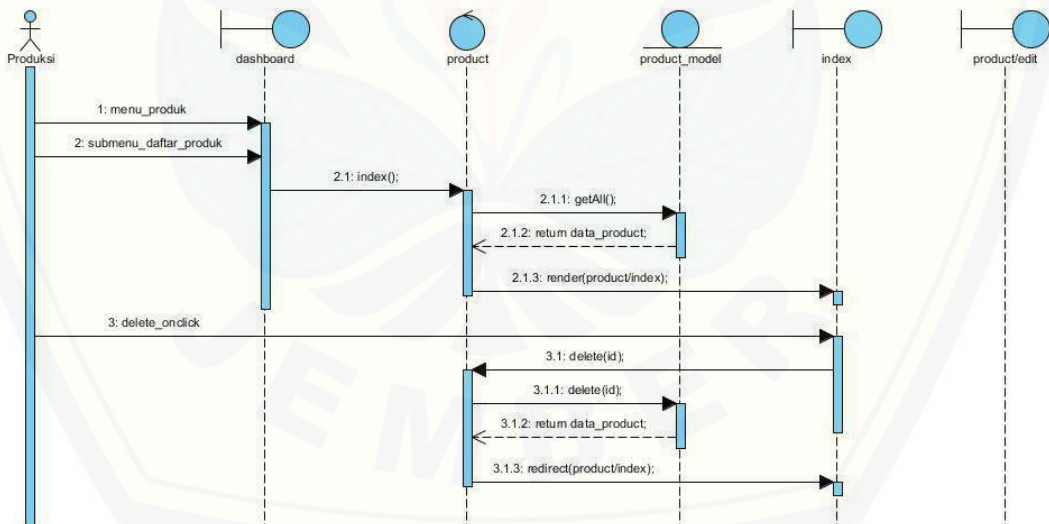
Sequence diagram melihat data hasil peramalan menjelaskan tentang proses menampilkan hasil peramalan yang dilakukan oleh aktor admin dan produksi. Admin dapat mengakses fitur ini dengan tujuan untuk mengawasi kegiatan sistem. Sedangkan produksi bertujuan untuk merencanakan jumlah produk yang akan diproduksi dengan saran dari hasil peramalan penjualan. Pada *sequence* menjelaskan tentang *method* dan kelas yang digunakan untuk menampilkan data hasil peramalan. Detail *sequence diagram* melihat data hasil peramalan akan ditampilkan pada Gambar B.7 pada Lampiran B.

10. *Sequence Diagram* Mengelola Data Produk

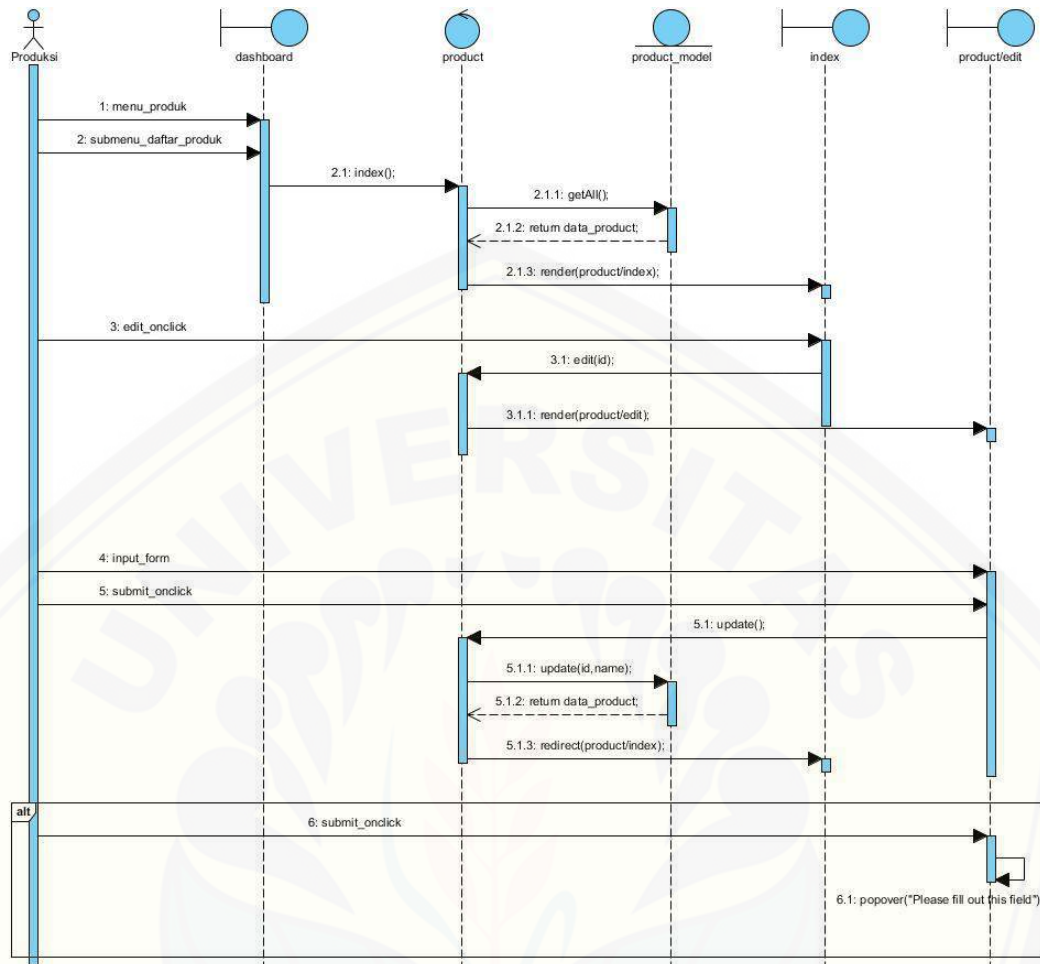
Sequence diagram mengelola data produk menjelaskan tentang proses pengelolaan data produk. Pada *sequence* dijelaskan *method* dan kelas yang digunakan untuk *input*, *update*, dan *delete* data produk. Serta alur penggunaan *method* dan kelas tersebut sehingga proses berhasil dijalankan. Detail *sequence diagram* mengelola data produk akan ditampilkan pada Gambar 4.14, Gambar 4.15 dan Gambar 4.16.



Gambar 4.14 Sequence Mengelola Data Produk



Gambar 4.15 Sequence Mengelola Data Produk



Gambar 4.16 *Sequence* Mengelola Data Produk

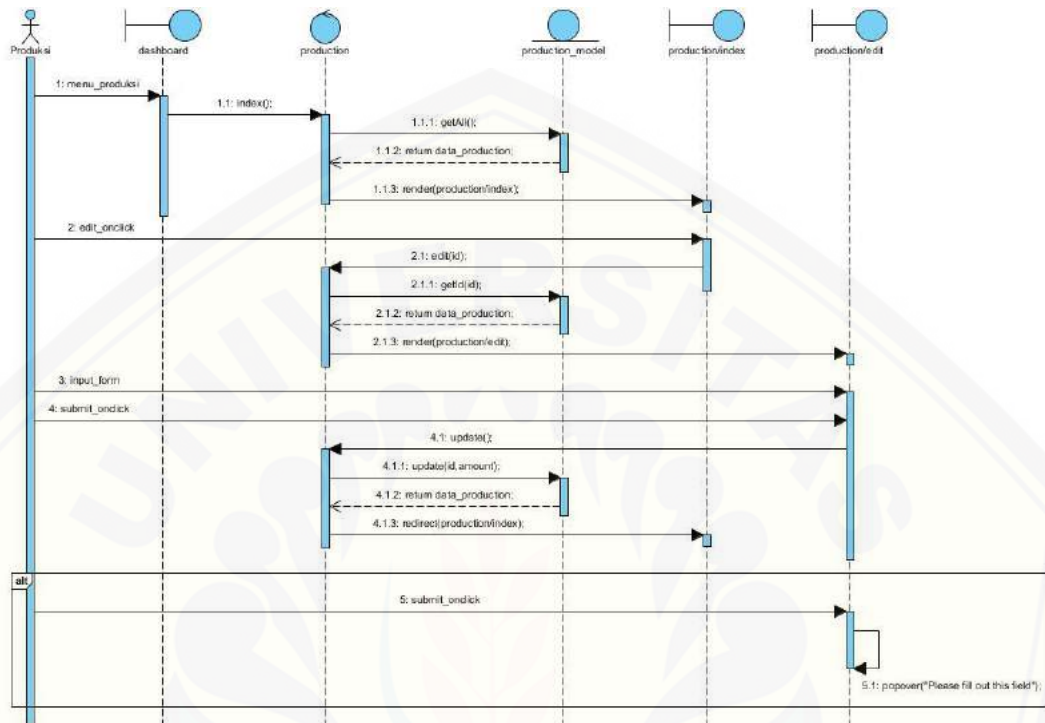
11. *Sequence Diagram* Melihat Data Produk

Sequence diagram melihat data produk menjelaskan tentang proses menampilkan data produk. Proses ini dijalankan oleh admin dan marketing. Admin melakukan kegiatan ini berfungsi sebagai pengawasan sedangkan marketing berfungsi untuk melihat produk apa saja yang dikelola oleh sistem. Detail *sequence diagram* melihat data produk akan ditampilkan pada Gambar B.8 pada Lampiran B.

12. *Sequence Diagram* Mengelola Data Produksi

Sequence diagram mengelola data produksi menjelaskan tentang proses pengelolaan data produksi. Proses ini dikelola oleh bagian produksi. Pada *sequence* dijelaskan *method* dan kelas yang digunakan untuk memasukkan jumlah produksi

dan menghitung tanggal kadaluwarsa produk. Detail *sequence diagram* mengelola data produksi akan ditampilkan pada Gambar 4.17.



Gambar 4.17 *Sequence* Mengelola Data Produksi

13. *Sequence Diagram* Melihat Data Produksi

Sequence Diagram mengelola data diri menjelaskan tentang proses menampilkan data produksi. Pada *sequence* dijelaskan method dan kelas serta alur jalannya program sehingga dapat menampilkan data produksi. Proses ini dijalankan oleh aktor admin. Detail *sequence diagram* melihat data produksi akan ditampilkan pada Gambar B.9 pada Lampiran B.

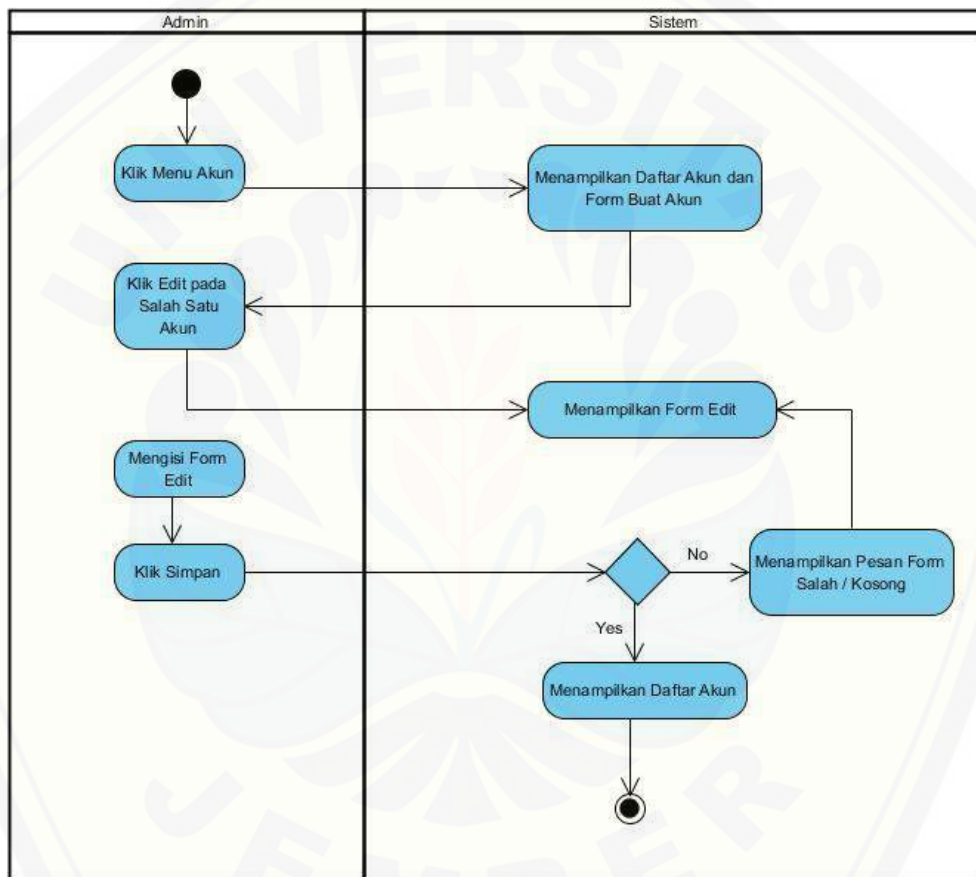
4.2.5 *Activity Diagram*

1. *Activity Diagram Login*

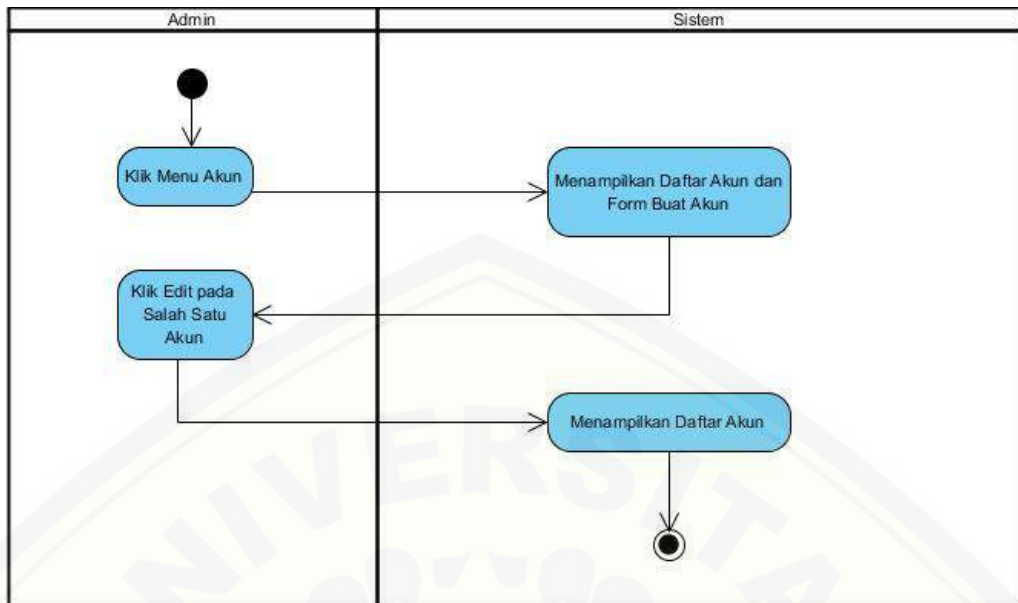
Activity diagram login menjelaskan tentang alur jalannya fitur yang dilakukan oleh aktor. Aktor melakukan *input password* dan *username* lalu menekan tombol *login*. Untuk detail *activity diagram login* dapat dilihat pada Gambar 15, Gambar C.1, Gambar C.2, dan Gambar C.3 di lampiran C.

2. Activity Diagram Mengelola Data Akun Pengguna

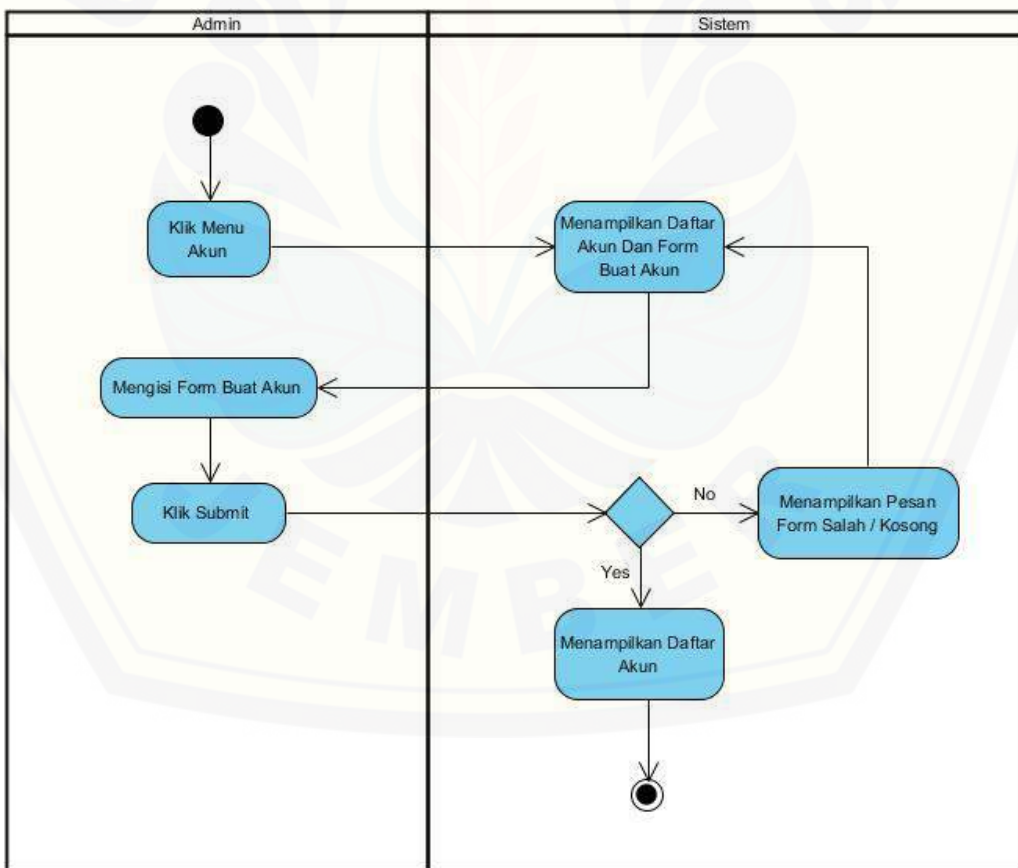
Activity diagram mengelola data akun pengguna menjelaskan tentang alur kegiatan aktor admin ketika melakukan pengelolaan data akun pengguna. Aktor melakukan 3 kegiatan yaitu *input* data akun, *edit* data akun dan *delete* data akun. Untuk detail activity mengelola data akun pengguna dapat dilihat pada Gambar 4.18, Gambar 4.19, dan Gambar 4.20.



Gambar 4.18 Activity Mengelola Data Akun Pengguna



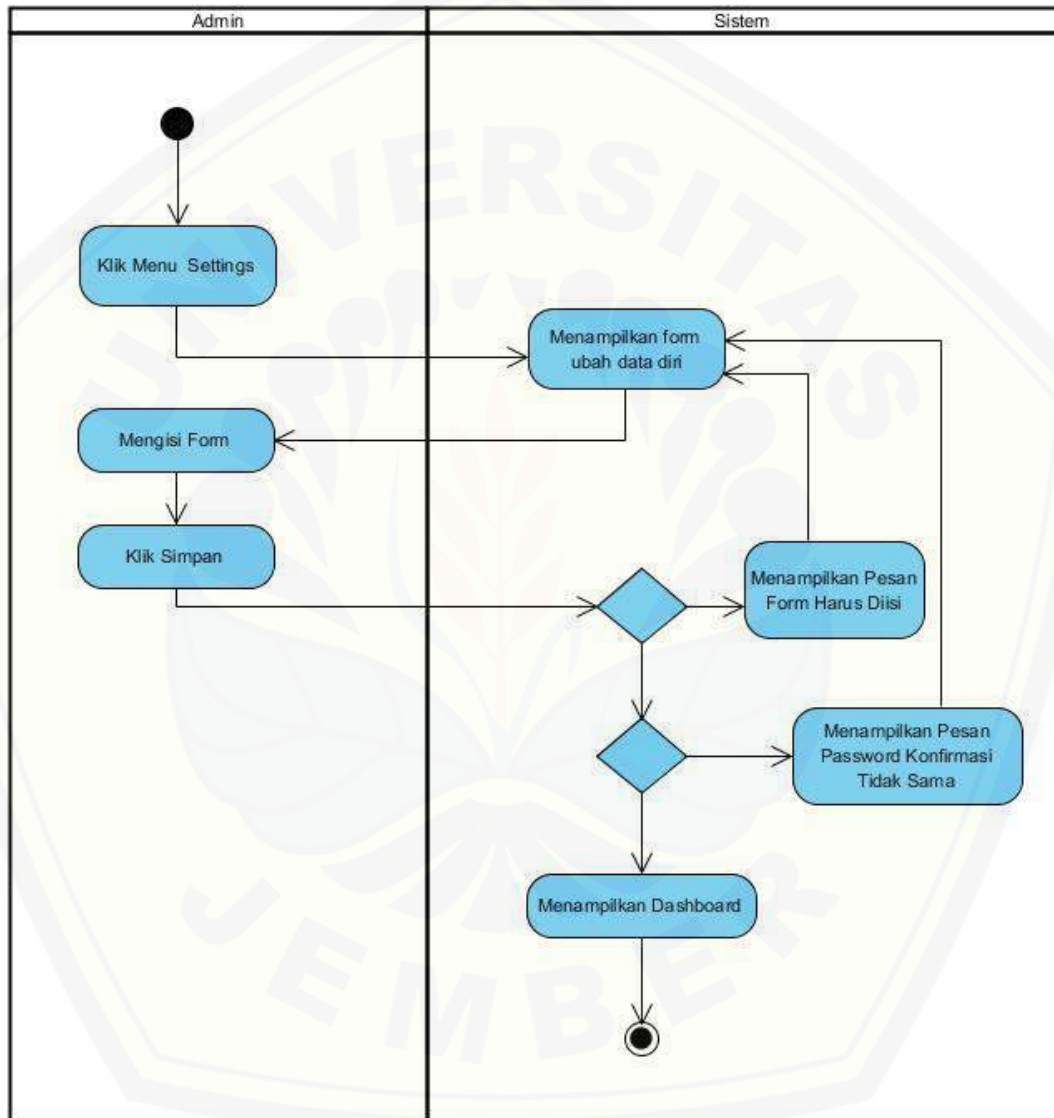
Gambar 4.19 Activity Mengelola Data Akun Pengguna



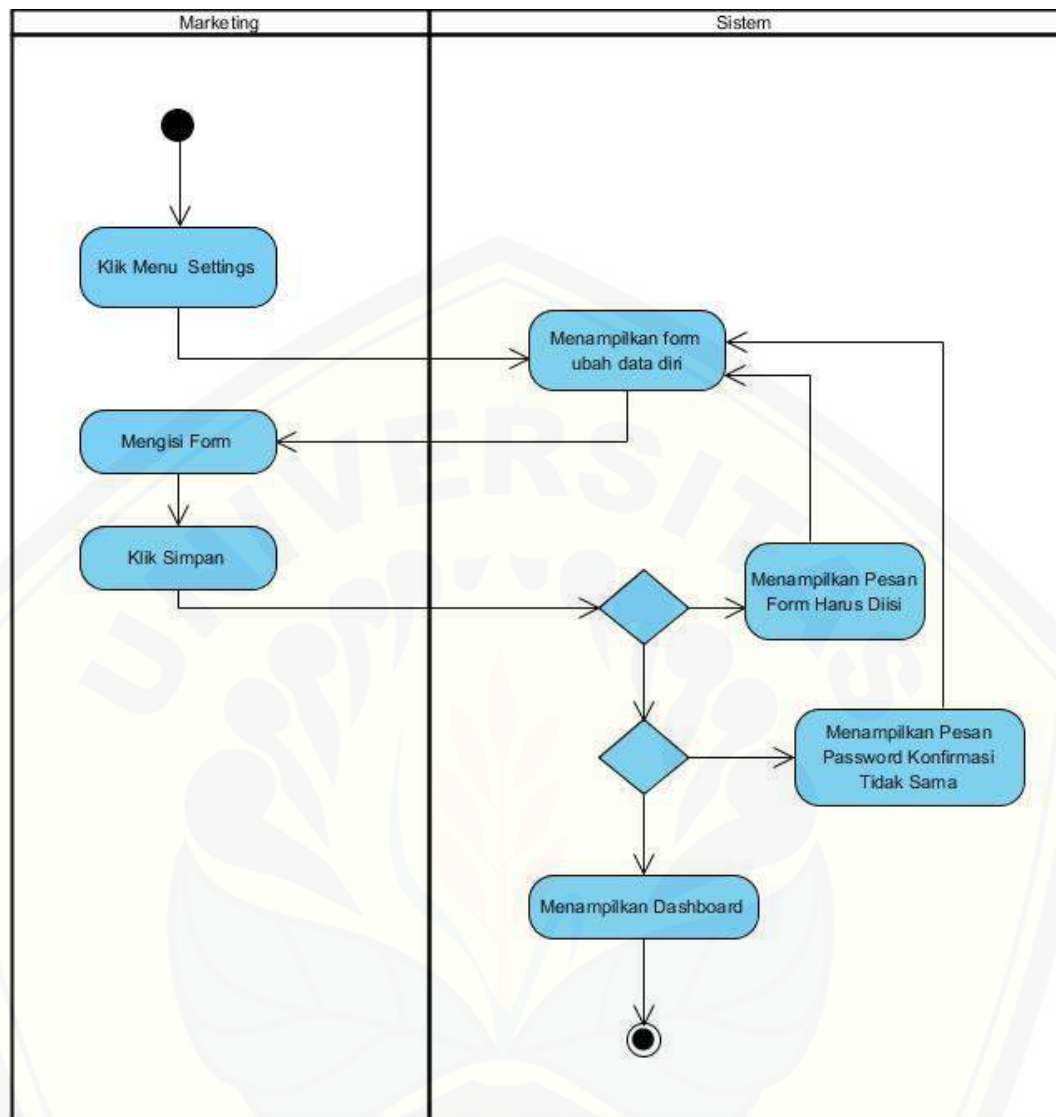
Gambar 4.20 Activity Mengelola Data Akun Pengguna

3. Activity Diagram Mengelola Data Diri

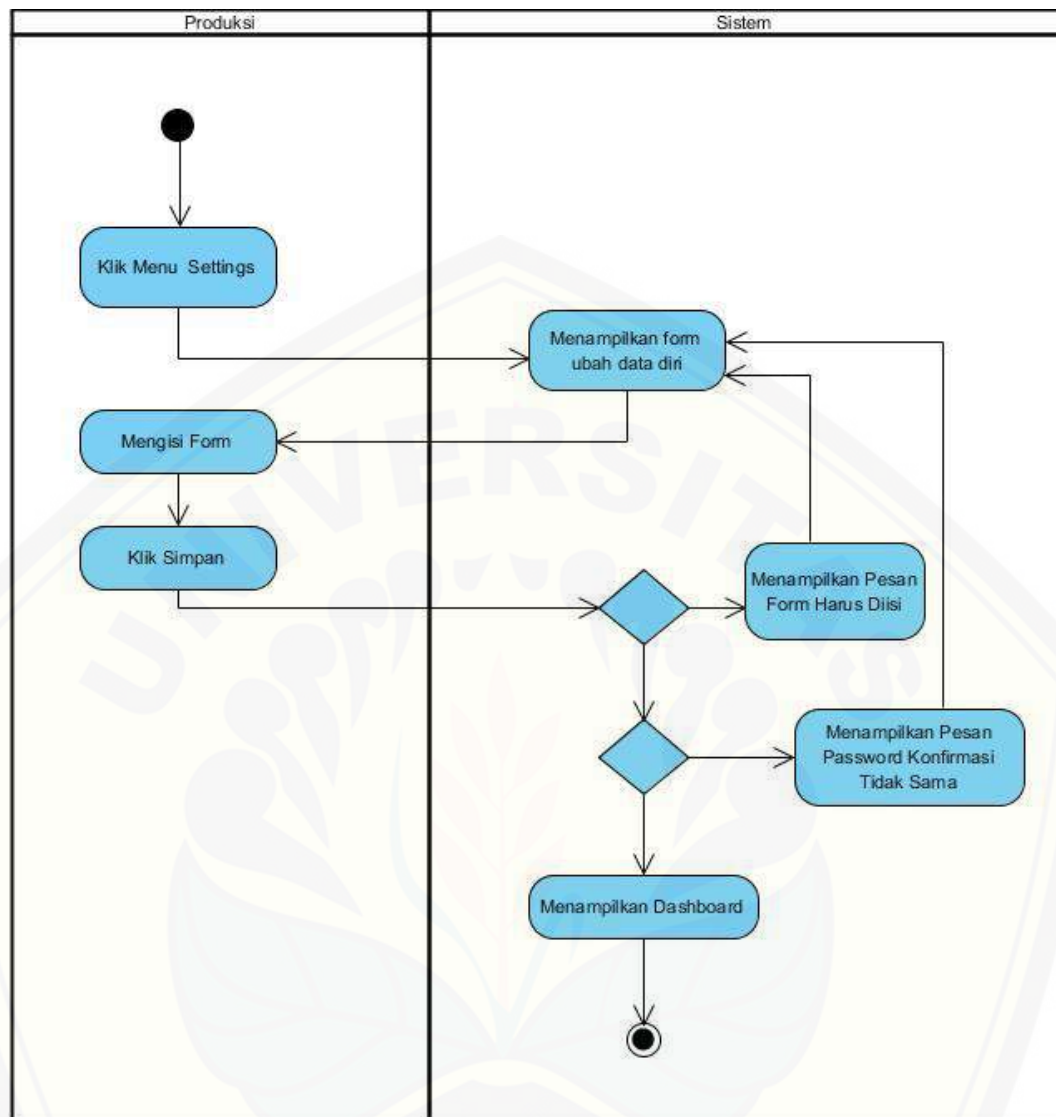
Activity Diagram mengelola data diri menjelaskan tentang alur kegiatan aktor ketika melakukan pengelolaan data diri. Fitur ini meliputi perubahan *password* ketika pertama kali user *login*. Untuk detail activity mengelola data diri dapat dilihat pada Gambar 4.21, Gambar 4.22 dan Gambar 4.23.



Gambar 4.21 Activity Mengelola Data Diri



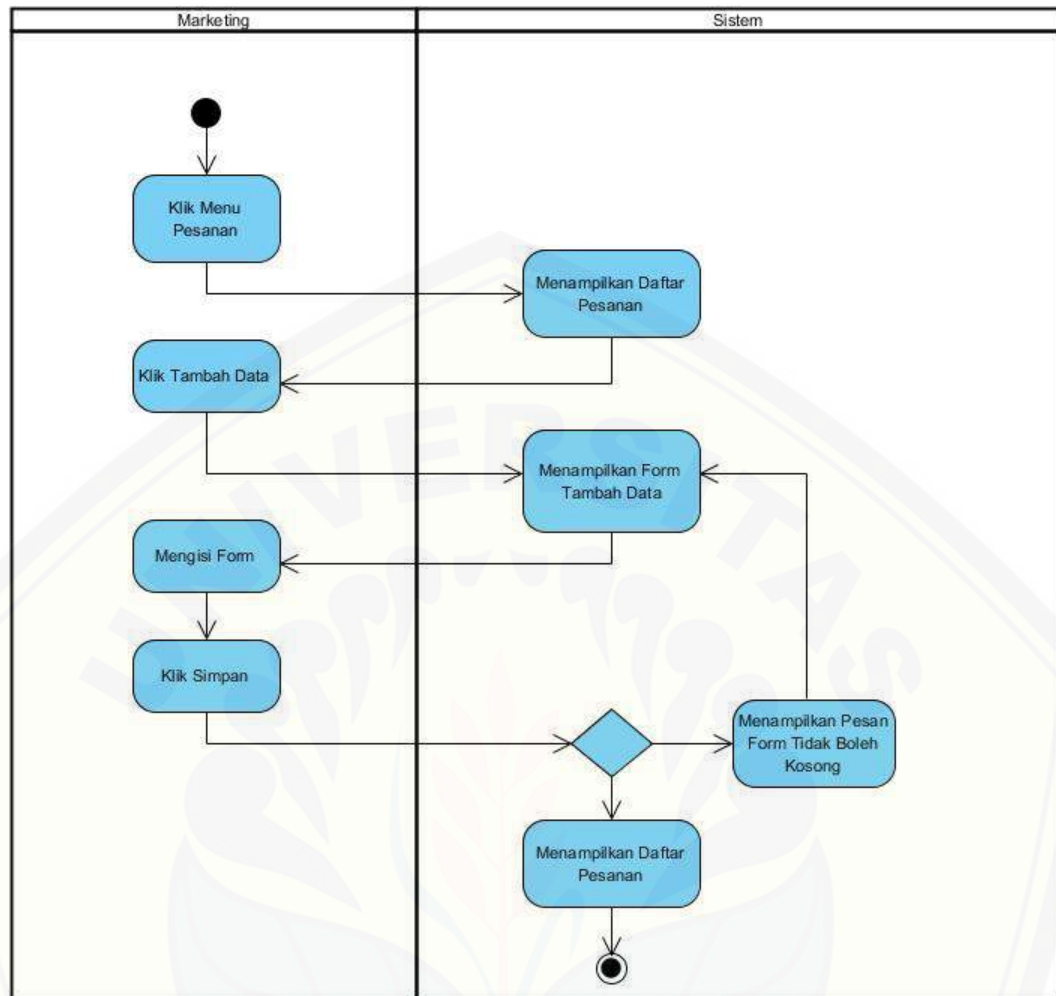
Gambar 4.22 Activity Mengelola Data Diri



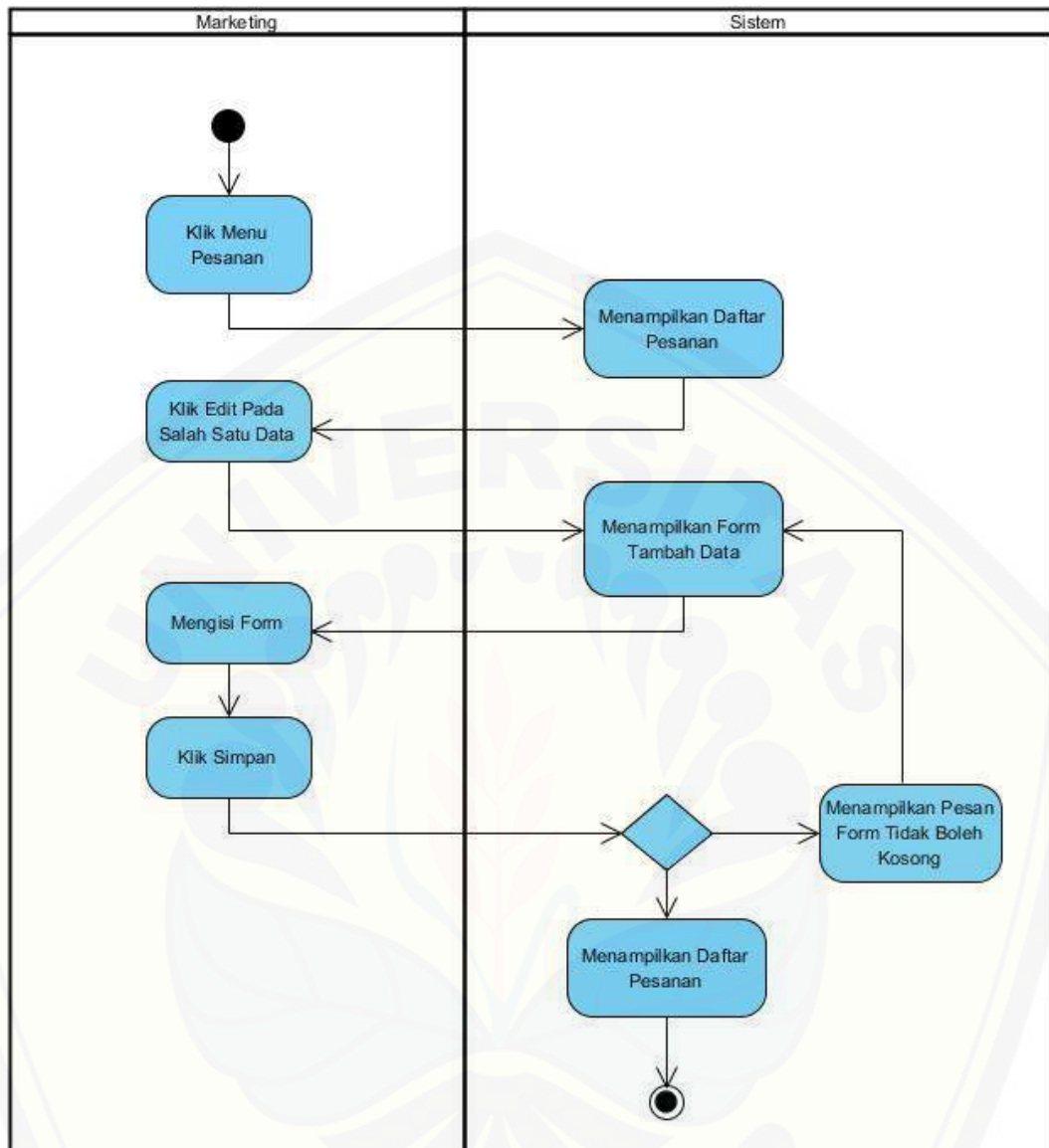
Gambar 4.23 Activity Mengelola Data Diri

4. Activity Diagram Mengelola Data Pesanan

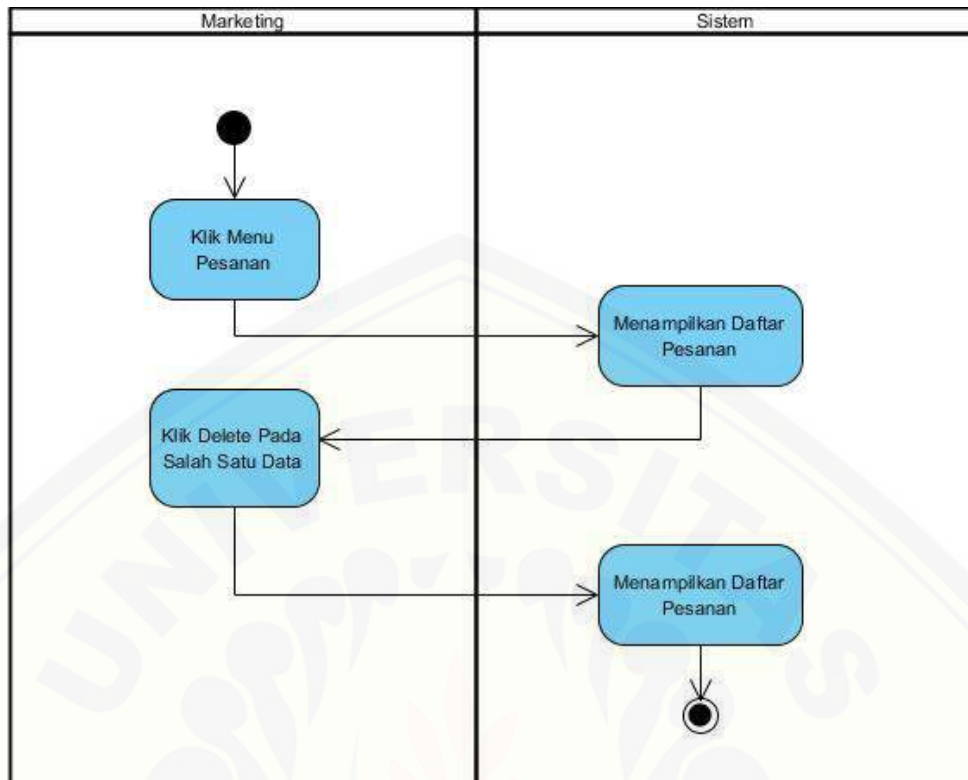
Activity diagram mengelola data pesanan menjelaskan tentang alur kegiatan aktor marketing ketika melakukan pengelolaan data pesanan. Aktor marketing melakukan *input*, *edit* dan hapus data pesanan. Untuk detail activity mengelola data pesanan dapat dilihat pada Gambar 4.24, Gambar 4.25 dan Gambar 4.26.



Gambar 4.24 Activity Mengelola Data Pesanan



Gambar 4.25 Activity Mengelola Data Pesanan



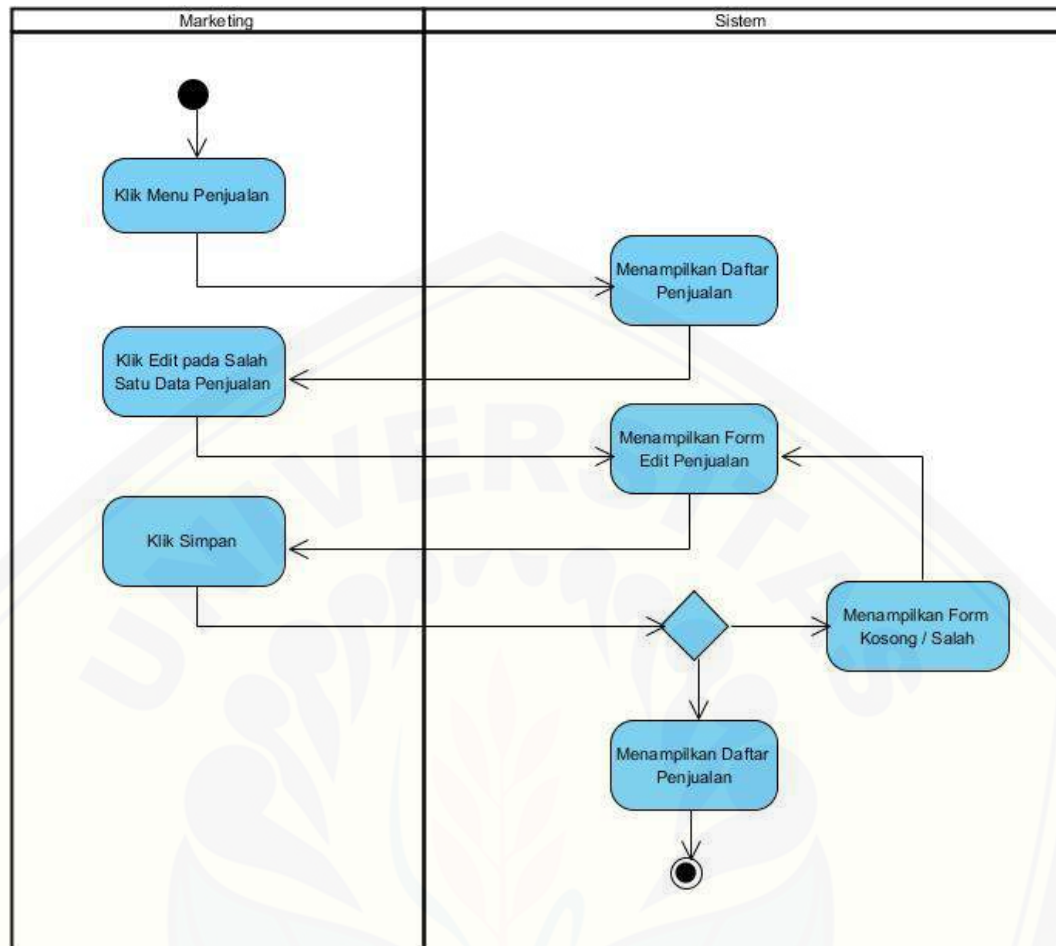
Gambar 4.26 Activity Mengelola Data Pesanan

5. Activity Diagram Melihat Data Pesanan

Activity diagram melihat data pesanan menjelaskan tentang alur kegiatan admin dan produksi ketika melihat data pesanan. Sistem akan menampilkan daftar pesanan yang telah diinputkan bagian marketing. Untuk detail activity melihat data pesanan dapat dilihat pada Gambar C.4 dan Gambar C.5 lampiran C.

6. Activity Diagram Mengelola Data Penjualan

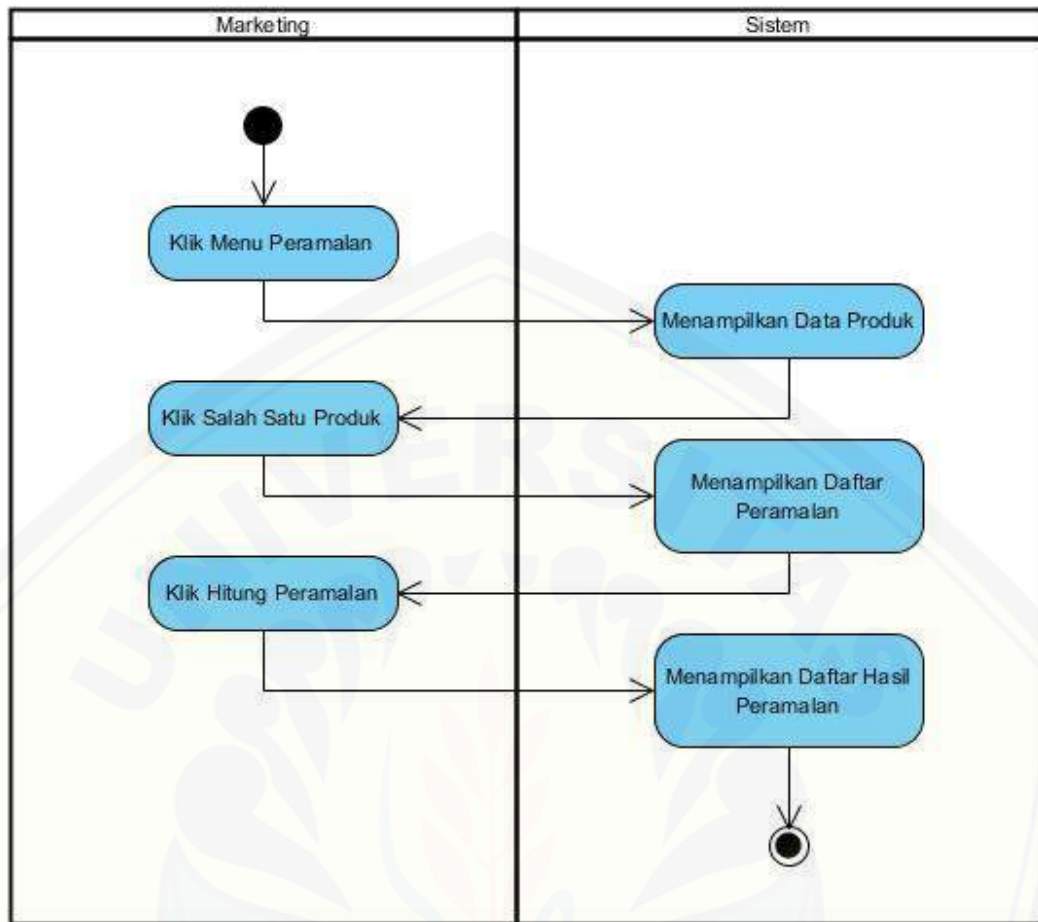
Activity diagram mengelola data penjualan menjelaskan tentang alur kegiatan aktor marketing ketika mengelola data penjualan yang berupa *input* data penjualan. Data penjualan perhari dimasukkan oleh bagian marketing perharinya. Untuk detail activity mengelola data penjualan dapat dilihat pada Gambar 4.27.



Gambar 4.27 Activity Mengelola Data Penjualan

7. Activity Diagram Mengelola Hasil Peramalan

Activity Diagram mengelola data hasil peramalan menjelaskan tentang alur kegiatan aktor marketing ketika mengelola data hasil peramalan yang berupa *input* data penjualan. Setelah marketing memasukkan data penjualan, bagian marketing melakukan perhitungan peramalan dengan menekan tombol “Hitung Peramalan” pada view. Untuk detail *activity* mengelola data hasil peramalan dapat dilihat pada Gambar 4.28.



Gambar 4.28 *Activity Diagram* Mengelola Hasil Peramalan

8. *Activity Diagram* Melihat Data Penjualan

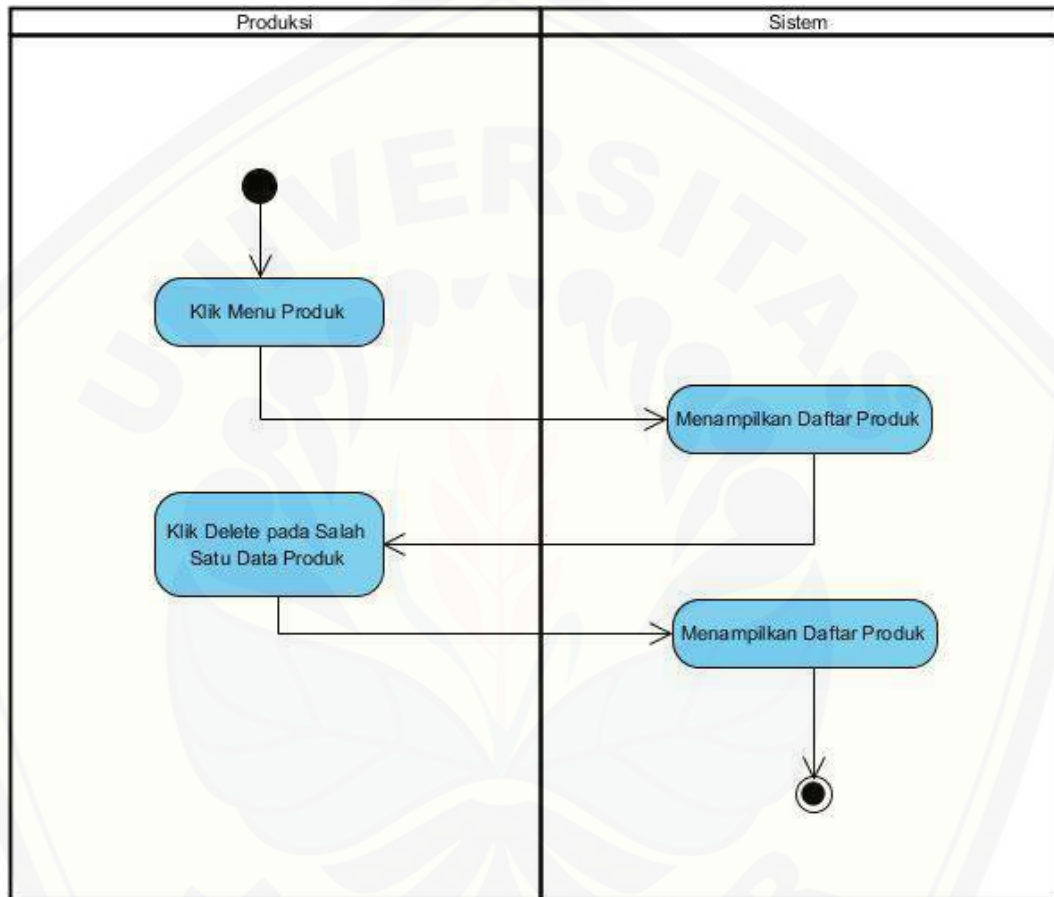
Activity Diagram melihat data penjualan menjelaskan alur kegiatan aktor admin ketika melihat data penjualan. Sistem akan menampilkan data penjualan yang sudah diinputkan bagian marketing. Untuk detail *activity* melihat data penjualan dapat dilihat pada Gambar C.6 lampiran C.

9. *Activity Diagram* Melihat Data Peramalan

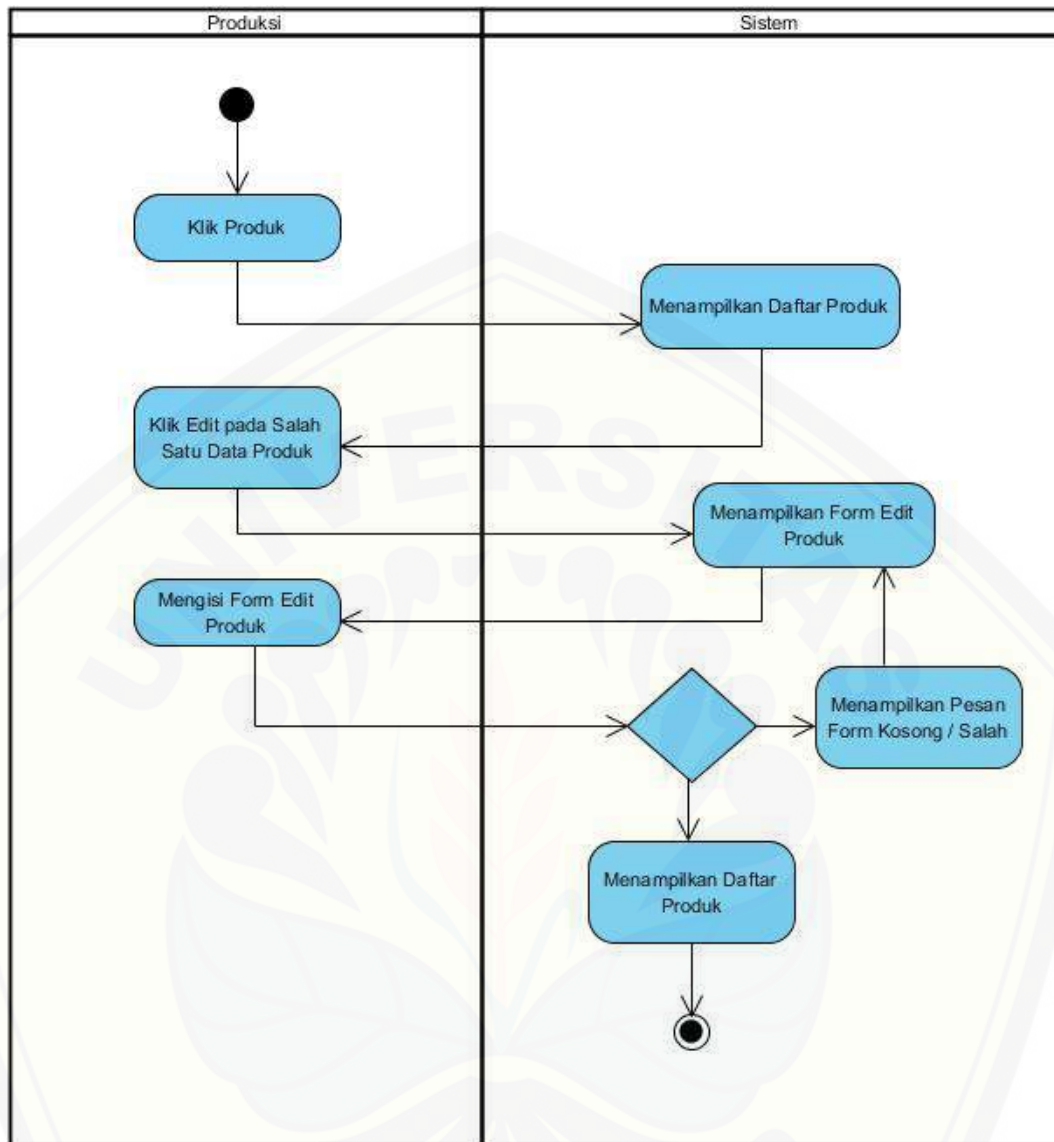
Activity Diagram melihat data peramalan menjelaskan alur ketika aktor admin dan produksi melihat data peramalan. Aktor admin melihat data peramalan guna mengawasi kegiatan sistem. Dan untuk aktor produksi berfungsi sebagai saran dalam menentukan jumlah produksi. Untuk detail *activity* melihat data peramalan dapat dilihat pada Gambar C.7 dan Gambar C.8 lampiran C.

10. Activity Diagram Mengelola Data Produk

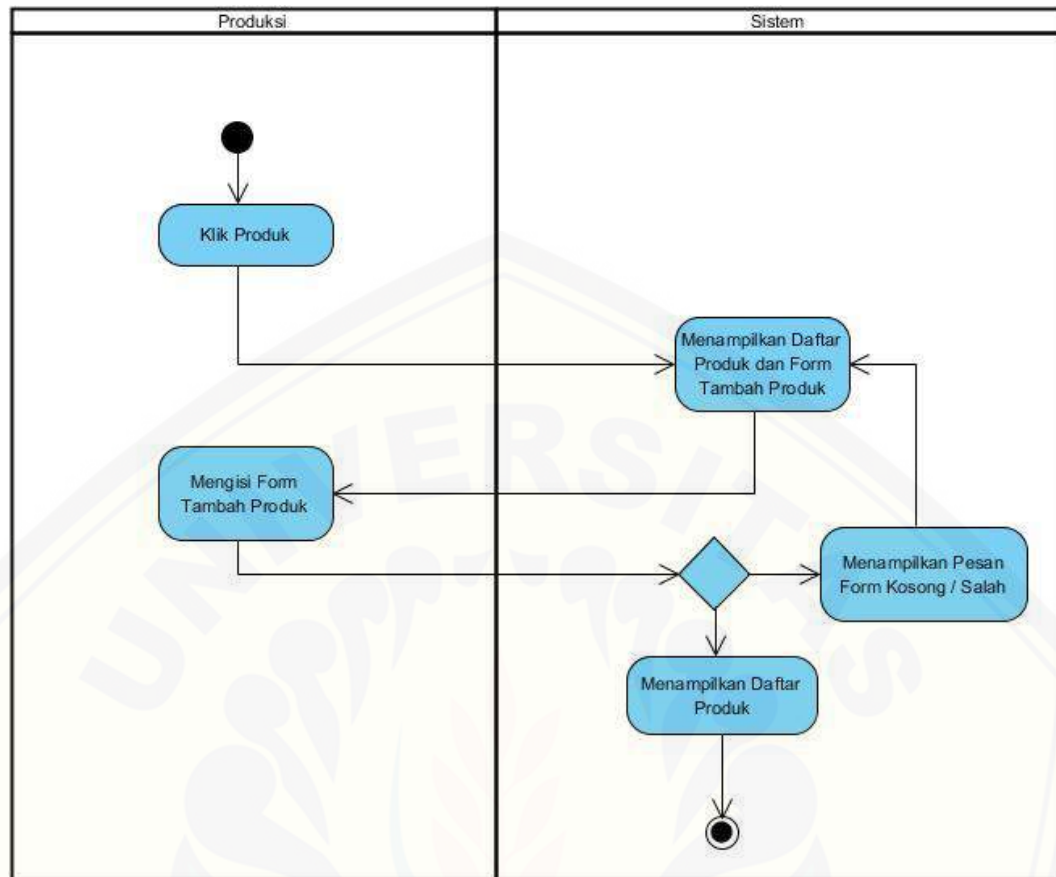
Activity Diagram mengelola data produk menjelaskan tentang alur kegiatan aktor produksi serta respon dari sistem ketika melakukan pengelolaan data produk. Proses ini meliputi *input*, *edit* dan *delete* produk. Untuk detail *activity* mengelola data produk dapat dilihat pada Gambar 4.29, Gambar 4.30 dan Gambar 4.31.



Gambar 4.29 Activity Mengelola Data Produk



Gambar 4.30 Activity Mengelola Data Produk



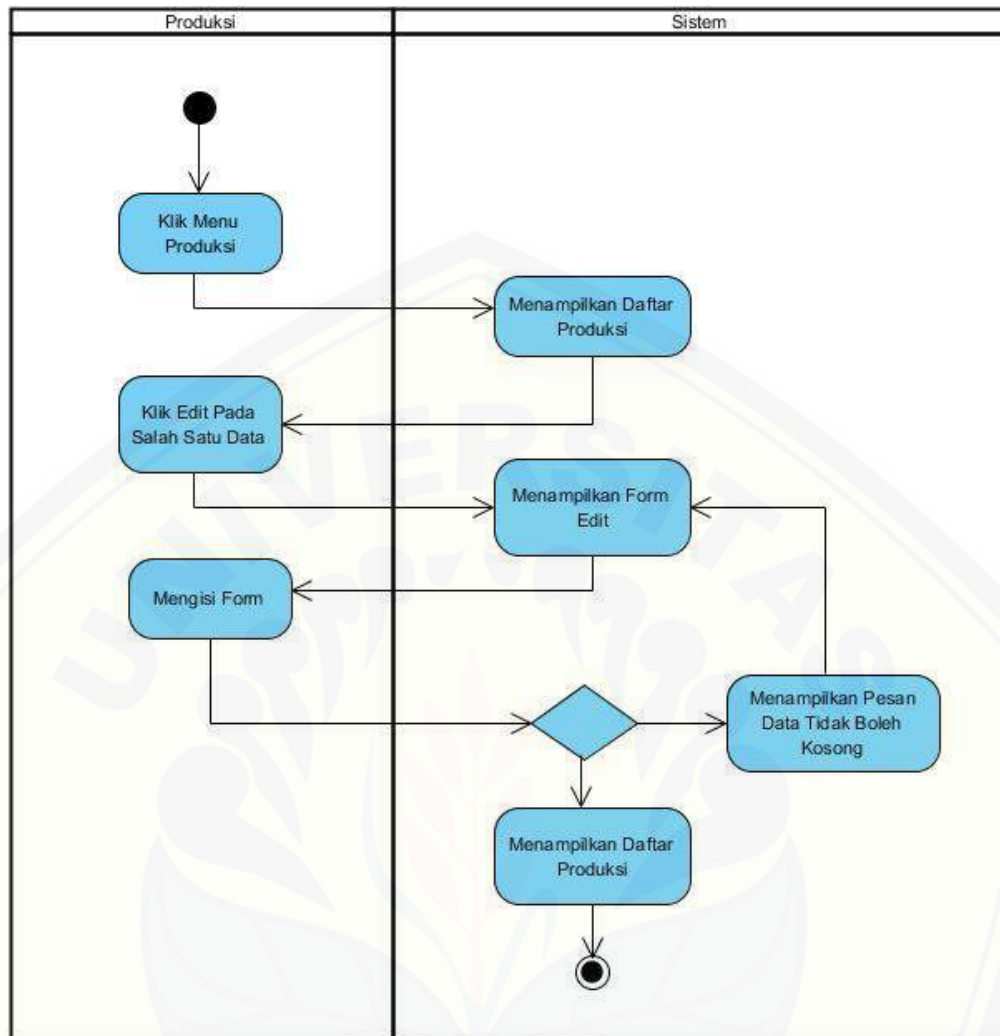
Gambar 4.31 *Activity* Mengelola Data Produk

11. *Activity Diagram* Melihat Data Produk

Activity Diagram melihat data produk menjelaskan tentang kegiatan aktor admin dan marketing serta respon sistem saat menampilkan data produk. Untuk detail activity melihat data produk dapat dilihat pada Gambar C.9 lampiran C.

12. *Activity Diagram* Mengelola Data Produksi

Activity Diagram mengelola data produksi menjelaskan tentang kegiatan aktor produksi dan respon sistem saat melakukan proses pengelolaan data produksi. Proses ini berisi *input* data produksi perhari. Untuk detail activity mengelola data produksi dapat dilihat pada Gambar 4.32.



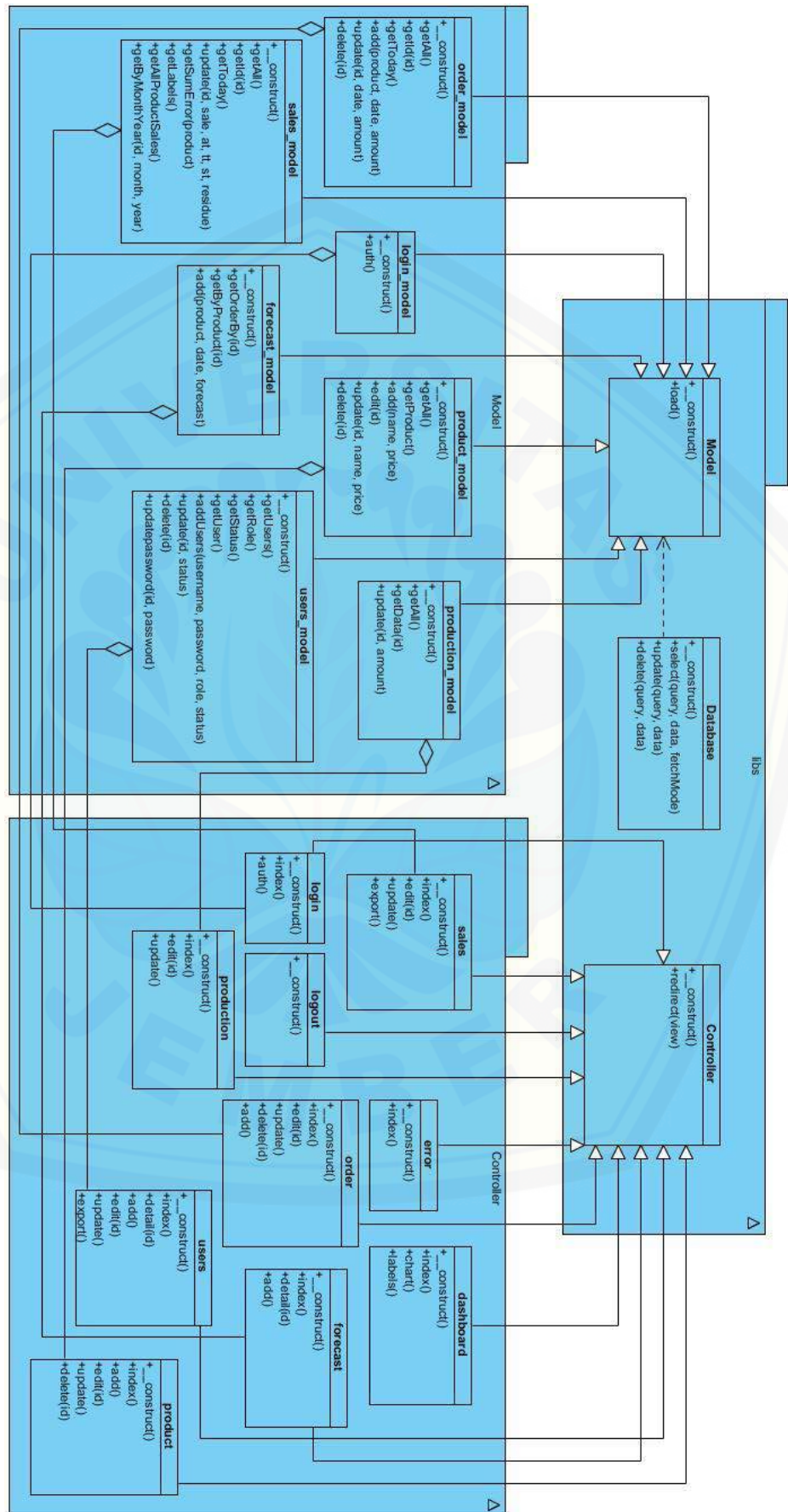
Gambar 4.32 Activity Mengelola Data Produksi

13. Activity Diagram Melihat Data Produksi

Activity Diagram melihat data produksi menjelaskan tentang alur kegiatan aktor admin dapat melihat data produksi sebagai bentuk pengawasan terhadap kerja sistem. Untuk detail activity melihat data produksi dapat dilihat pada Gambar C.10 lampiran C.

4.2.6 Class Diagram

Class Diagram pada sistem ini menjelaskan tentang hubungan antar method dan antar kelas sehingga dapat membentuk suatu kegiatan dalam sistem. Kegiatan-kegiatan tersebut dibagi dalam bentuk fitur-fitur. Dan setiap fitur memiliki hubungan dengan method lain pada kelas lain.

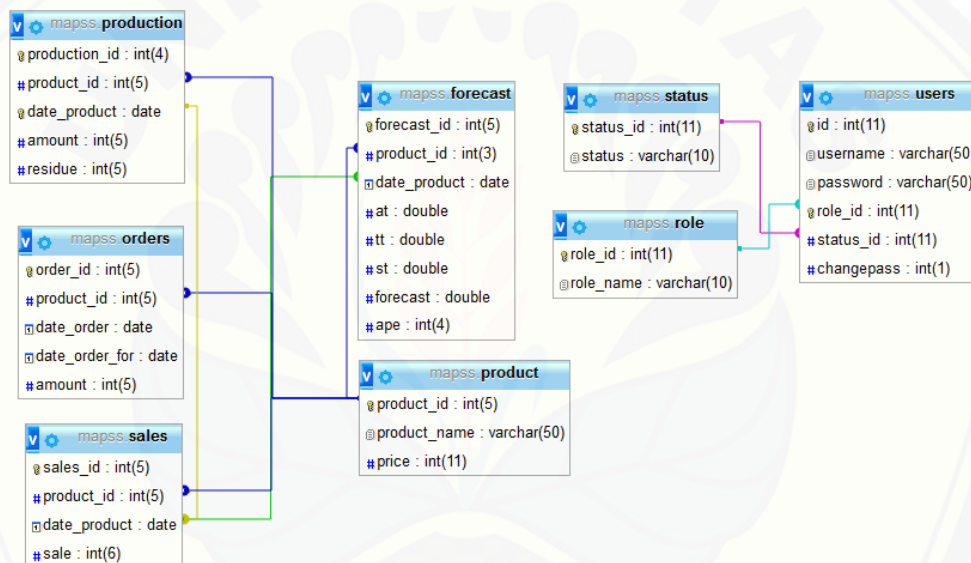


Gambar 4.33 Class Diagram

Pada Gambar 4.33 dijelaskan bahwa semua kelas pada model membutuhkan kelas Model sebagai induk. Dan kelas pada controller membutuhkan kelas Controller sebagai induk. Kelas-kelas pada controller dan model saling terhubung untuk membentuk suatu proses dimana kelas pada controller memberi perintah dan model memanggil query sesuai perintah dari controller lalu melanjutkan lagi ke controller.

4.2.7 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan *design database*. Yang berisi hubungan antar tabel pada *database*. ERD pada sistem ini dapat dilihat pada Gambar 4.34.



Gambar 4.34 Entity Relationship Diagram

4.3 Implementasi

Desain yang telah dibuat kemudian diimplementasikan ke dalam kode program. Bahasa pemrograman yang digunakan dalam sistem ini adalah Bahasa Pemrograman PHP. Untuk intrface menggunakan html, css, dan javascript. Sedangkan *database* yang digunakan adalah MYSQL.

4.3.1 Kode Program Login

Kode program *Login* terletak pada kelas *view/login*, *controller/login*, dan *login_model*. Pengguna memasukkan *username* dan *password* pada *view login*.

Dan ketika pengguna menekan tombol “login” maka controller *login* akan mengambil *username* dan *password* yang telah diinputkan. Dan akan dilakukan pengecekan pada model *login* apakah sesuai dengan data pada *database* atau tidak. Jika tidak sesuai maka akan muncul peringatan dan jika sesuai maka akan mengarahkan view ke dashboard. Penulisan kode program ini dapat dilihat pada Gambar 4.35.

```
24 function auth() {
25
26     $username = $_POST['username'];
27     $password = $_POST['password'];
28     $this->loginmodel->auth($username, $password);
29     if(Session::get('loggedIn') == TRUE){
30         $this->redirect( view: 'dashboard');
31     } elseif (Session::get('nonaktif') == TRUE) {
32         $this->redirect( view: '');
33         Session::set('error', "Akun Anda Telah Dinonaktifkan!");
34     } else {
35         Session::set('error', "Username atau password salah");
36         $this->redirect( view: '');
37     }
}
```

Gambar 4.35 Kode Program Kelas *Login/auth*

4.3.2 Kode Program Mengelola Data Akun Pengguna

Kode program mengelola data akun pengguna terletak pada kelas *view/users/index*, *view/users/edit*, *controller/users*, dan *users_model*. Aktor admin yang bertugas membuat akun pengguna akan melakukan *input* akun baru dengan menekan tombol “Tambah” pada view *users/index*. Sistem akan menampilkan form tambah akun. Dan jika admin menekan tombol “Simpan” maka semua masukan pada view akan dibaca oleh controller *users* dan diteruskan ke model *users* untuk ditambahkan pada *database*. Penulisan kode program *input* akun dapat dilihat pada Gambar 4.36 dan 4.37. Selain itu admin dapat mengubah akun pengguna dengan menekan tombol *edit* dan akan menampilkan form *edit* data pengguna. Jika admin telah menyimpan data maka controller *users* akan membaca data dan menruskan pada model *users* untuk selanjutnya di *update* pada *database*. Admin juga dapat menghapus data pengguna dengan menekan hapus pada view *users/index* dan diteruskan ke controller kemudian ke model untuk menghapus data pada *database*.

Penulisan kode program *edit* dan *delete* akun dapat dilihat pada Gambar 4.36 dan Gambar 4.37.

```
38     function add() {
39         $username = $_POST['username'];
40         $password = $_POST['password'];
41         $role     = $_POST['role_id'];
42         $status   = $_POST['status_id'];
43
44         $this->user->addUsers($username, $password, $role, $status);
45         $this->redirect( view: '/users' );
46     }
47
48     function edit($id) {
49         if(Session::get('role') == 1) {
50             $this->view->user = $this->user->getUser($id);
51             $this->view->role = $this->user->getRole();
52             $this->view->status = $this->user->getStatus();
53             $this->view->subtitle = "Edit Akun";
54             $this->view->render( view: 'users/edit' );
55         } else {
56             $this->view->rendererror();
57         }
58     }
```

Gambar 4.36 Kode Program Mengelola Data Akun Pengguna

```
60     function update() {
61         $status = $_POST['status'];
62         $id     = $_POST['id'];
63
64         $this->user->update($id, $status);
65         header( string: 'location: '.URL.'/users' );
66     }
67
68     function delete($id) {
69         $this->user->delete($id);
70         header( string: 'location: '.URL.'/users' );
71     }
```

Gambar 4.37 Kode Program Mengelola Data Akun Pengguna

4.3.3 Kode Program Mengelola Data Diri

Kode program mengelola data diri terletak pada kelas `view/users/changepass`, `controller/users`, dan `users_model`. Pengguna mengganti data diri yang meliputi perubahan *password* saat pertama kali *login*. Pengguna akan diarahkan ke halaman form ubah *password*. Dan jika selesai mengisi form lalu menekan tombol simpan maka controller users akan menjalankan method

changepass untuk mengirim *password* baru ke model users dan mengubah data ke *database*. Penulisan kode program ini dapat dilihat pada Gambar 4.38.

```
73 function changepass($id) {
74     $this->view->title = 'Akun';
75     $this->view->index = URL.'/dashboard';
76     $this->view->user = $this->user->getUser($id);
77     $this->view->subtitle = "Ubah Password";
78     $this->view->render( view: 'users/changepass');
79 }
80
81 function updatepassword() {
82     $id = $_POST['id'];
83     $password = $_POST['password'];
84     $confirm = $_POST['confirm'];
85     if($password != $confirm) {
86         $this->redirect( view: 'users/changepass/'.$id);
87     } else {
88         $this->user->updatepassword($id, $password);
89         Session::destroy();
90         header( string: 'location: '.URL.'/');
91     }
92 }
93 }
```

Gambar 4.38 Kode Program Mengelola Data Diri

4.3.4 Kode Program Mengelola Data Pesanan

Kode program mengelola data pesanan terletak pada kelas *view/orders/index*, *view/orders/edit*, *controller/order*, dan *order_model*. Aktor marketing yang bertugas mengelola data pesanan. Saat membuat data baru pengguna akan menekan tombol data baru dan program akan menampilkan form tambah data. Jika aktor selesai mengisi form dan menekan simpan maka *inputan* yang telah diisi akan dibaca oleh controller order dan diteruskan ke model order untuk diinputkan ke dalam *database*. Ketika aktor marketing memilih tombol *edit* program akan menampilkan form *edit*. Dan saat pengguna memilih simpan maka controller order akan membaca data yang telah diubah dan meneruskan ke model order untuk di *update* pada *database*. Saat proses *Delete* pengguna hanya menekan tombol *delete* dan controller order akan membaca id order yang akan di hapus dan meneruskannya ke model order untuk dihapus dari *database*. Penulisan kode program ini dapat dilihat pada Gambar 4.39 dan Gambar 4.40.

```
25 function index() {
26     $this->view->listproduct = $this->product->getAll();
27     $this->view->listorder = $this->order->getAll();
28     $this->view->title = "Pesanan";
29     $this->view->subtitle = "Daftar Pesanan";
30     $this->view->index = URL.'/order';
31     $this->view->render( view: 'orders/index');
32 }
33
34 function add() {
35     $product = $_POST['product'];
36     $date = $_POST['date'];
37     $amount = $_POST['amount'];
38
39     $this->order->add($product, $date, $amount);
40     $this->redirect( view: 'order');
41 }
42
43 function edit($id) {
44     if(Session::get('role') == 2) {
45         $this->view->order = $this->order->getId($id);
46         $this->view->title = "Pesanan";
47         $this->view->subtitle = "Edit Pesanan";
48         $this->view->index = URL.'/order';
49         $this->view->render( view: 'orders/edit');
50     } else {
51         $this->view->rendererror();
52     }
53 }
```

Gambar 4.39 Kode Program Mengelola Data Pesanan

```
55 function update() {
56     $id = $_POST['id'];
57     $date = $_POST['date'];
58     $amount = $_POST['amount'];
59
60     $this->order->update($id, $date, $amount);
61     $this->redirect( view: 'order/index');
62 }
63
64 function delete($id) {
65     $this->order->delete($id);
66     $this->redirect( view: 'order/index');
67 }
```

Gambar 4.40 Kode Program Mengelola Data Pesanan

4.3.5 Kode Program Melihat Data Pesanan

Kode program melihat data pesanan terletak pada kelas `view/orders/index`, `controller/order`, dan `order_model`. Aktor produksi dan admin yang melakukan proses ini dan hanya dapat menampilkan daftar pesanan. Aktor memilih menu pesanan dan controller order akan memanggil model order, dan model order mengambil data dari *database* kemudian dilanjutkan ke controller order untuk ditampilkan pada view order. Sistem akan menampilkan daftar pesanan yang belum dikerjakan. Penulisan kode program ini dapat dilihat pada Gambar 4.39 dalam method `index`.

4.3.6 Kode Program Mengelola Data Penjualan

Kode program mengelola data penjualan terletak pada kelas `view/sales/index`, `view/sales/edit`, `controller/sales`, dan `sales_model`. Aktor marketing yang bertugas mengelola data penjualan. Untuk memasukkan data penjualan baru aktor marketing memilih tombol *edit* program akan menampilkan form *edit*. Dan saat pengguna memilih simpan maka controller sales akan membaca data yang telah diubah dan meneruskan ke model sales untuk di *update* pada *database*. Penulisan kode program ini dapat dilihat pada Gambar 4.41, Gambar 4.42, dan Gambar 4.43.

```

56 function update() {
57     $sale      = $_POST['sale'];
58     $product   = $_POST['product_id'];
59     $date      = $_POST['date_product'];
60     $value     = $this->forecast->getByProduct($product);
61     $data = [];
62     $pe       = [];
63     $At = null; $Tt = null; $St = null;
64     $alpha = 0;
65     $beta = 0;
66     $gamma = 0;
67     $ad = null;
68     $residue = null; $date0 = null; $residue0 = null;
69
70     if($product == 1){
71         $alpha = 0.069;
72         $beta = 0.78;
73         $gamma = 0.01;
74     } elseif ($product == 3) {
75         $alpha = 0.23;
76         $beta = 0.15;
77         $gamma = 0.01;
78     } elseif ($product == 4) {
79         $alpha = 0.03;
80         $beta = 0.99;
81         $gamma = 0.01;
82     }

```

Gambar 4.41 Kode Program Mengelola Data Penjualan

```

84 for ($i = 0; $i < count($value); $i++) {
85     if ($i < count($value)-1) {
86         $At = $value[$i]['at'];
87         $data[$i] = $At;
88         $Tt = $value[$i]['tt'];
89         $St = $value[$i]['st'];
90     } else {
91         $At = $alpha * ($sale / $value[$i - 7]['st']) + (1 - $alpha) * ($value[$i - 1]['at'] + $value[$i-1]['tt']);
92         $data[$i] = $At;
93         $Tt = ($beta * ($data[$i] - $value[$i - 1]['at'])) + ((1 - $beta) * $value[$i - 1]['tt']);
94         $St = $gamma * ($sale / $data[$i]) + (1 - $gamma) * $value[$i - 7]['st'];
95         if($sale != 0) {
96             $pe = (abs( number: $sale - $value[$i]['forecast'] )/$sale) * 100;
97         } else {
98             $pe = 100;
99         }
100         $ad = abs( number: $sale - $value[$i]['forecast'] );
101     }
102 }
103
104 $this->sales->update($product, $sale, $At, $Tt, $St, $pe, $date, $ad);
105 $this->redirect( view: 'sales');
106 }

```

Gambar 4.42 Kode Program Mengelola Data Penjualan

```

34 function index() {
35     $this->view->listsales = $this->sales->getAll();
36     $this->view->listproduct = $this->product->getAll();
37     $this->view->title = 'Penjualan';
38     $this->view->subtitle = "Daftar Penjualan";
39     $this->view->index = URL.' /sales';
40     $this->view->render( view: 'sales/index');
41 }
42
43 function edit($id) {
44     if(Session::get('role') == 2) {
45         $this->view->sales = $this->sales->getId($id);
46         $this->view->listproduct = $this->product->getAll();
47         $this->view->title = 'Penjualan';
48         $this->view->subtitle = "Edit Penjualan";
49         $this->view->index = URL.' /sales';
50         $this->view->render( view: 'sales/edit');
51     } else {
52         $this->view->rendererror();
53     }
54 }

```

Gambar 4.43 Kode Program Mengelola Data Penjualan

4.3.7 Kode Program Mengelola Hasil Peramalan

Kode program mengelola hasil peramalan terletak pada kelas view/forecast/index, view/forecast/detail, view/forecast/edit, controller/forecast, dan forecast_model. Aktor marketing yang bertugas mengelola data data hasil peramalan. Untuk menghitung data peramalan aktor marketing memilih tombol hitung peramalan dan controller forecast akan menghitung peramalan kemudian meneruskan hasilnya ke model forecast untuk diinputkan pada *database*. Penulisan kode program ini dapat dilihat pada lampiran D.

```

19 public function index() {
20     $this->view->product = $this->product->getAll();
21     $this->view->title = 'Peramalan';
22     $this->view->subtitle = "Daftar Produk";
23     $this->view->index = URL.' /forecast';
24     $this->view->render( view: 'forecast/index');
25 }
26
27 function detail($id) {
28     $this->view->list = $this->forecast->getOrderBy($id);
29     $this->view->product = $this->product->getProduct($id);
30     $this->view->title = 'Peramalan';
31     $this->view->subtitle = "Daftar Peramalan";
32     $this->view->index = URL.' /forecast';
33     $this->view->render( view: 'forecast/detail');
34 }

```

Gambar 4.44 Kode Program Kelas index dan detail

```

36 function add()
37 {
38     $product = $_POST['product_id'];
39     $value = $this->forecast->getByProduct($product);
40     $tanggal = NULL;
41
42     for ($i = 0; $i < count($value); $i++) {
43         if ($i < count($value)-1) {
44             $Yt = $value[$i]['forecast'];
45             $tanggal = $value[$i]['date_product'];
46         } else {
47             if($value[$i]['at'] == 0) {
48                 Session::set('errorforecast', true);
49             } else {
50                 $date_sales = new DateTime($value[$i]['date_product']);
51                 $date_sales->add(new DateInterval( interval_spec: 'P01D'));
52                 $tanggal = $date_sales->format( format: 'Y-m-d');
53                 $Yt = ($value[$i]['at'] + ($value[$i]['tt'])) * $value[$i - 6]['st'];
54                 $this->forecast->add($product, $tanggal, $Yt);
55             }
56         }
57     }
58     $this->redirect( view: 'forecast/detail/'.$product);
59 }

```

Gambar 4.45 Kode Program Kelas add

4.3.8 Kode Program Melihat Data Penjualan

Kode program melihat data penjualan terletak pada kelas view/sales/index, controller/sales, dan sales_model. Aktor admin yang melakukan proses ini dan hanya dapat menampilkan daftar penjualan. Aktor memilih menu penjualan dan controller sales akan memanggil model sales, dan model sales mengambil data dari *database* kemudian dilanjutkan ke controller sales untuk ditampilkan pada view sales. Sistem akan menampilkan daftar penjualan yang telah diinputkan oleh marketing. Penulisan kode program ini dapat dilihat pada Gambar 4.43 dalam method index.

4.3.9 Kode Program Melihat Data Hasil Peramalan

Kode program melihat data hasil peramalan terletak pada kelas view/forecast/index, view/forecast/detail, controller/forecast, dan forecast_model. Aktor produksi dan admin yang melakukan proses ini dan hanya dapat menampilkan daftar hasil peramalan. Aktor memilih menu peramalan dan controller forecast akan memanggil model forecast, dan model forecast mengambil data dari *database* kemudian dilanjutkan ke controller forecast untuk ditampilkan pada view forecast. Sistem akan menampilkan daftar hasil peramala. Penulisan kode program ini dapat dilihat pada Gambar 4.44 dalam method index.

4.3.10 Kode Program Mengelola Data Produk

Kode program mengelola data produk terletak pada kelas `view/product/index`, `view/product/edit`, `controller/product`, dan `product_model`. Aktor produksi yang bertugas mengelola data pesanan. Saat membuat data baru pengguna akan menekan tombol data baru dan program akan menampilkan form tambah data. Jika aktor selesai mengisi form dan menekan simpan maka masukan yang telah diisi akan dibaca oleh controller product dan diteruskan ke model product untuk diinputkan ke dalam *database*. Ketika aktor marketing memilih tombol *edit* program akan menampilkan form *edit*. Dan saat pengguna memilih simpan maka controller product akan membaca data yang telah diubah dan meneruskan ke model product untuk di *update* pada *database*. Saat proses *delete* pengguna hanya menekan tombol *delete* dan controller product akan membaca id product yang akan di hapus dan meneruskannya ke model product untuk dihapus dari *database*. Penulisan kode program ini dapat dilihat pada lampiran D.

```
26 function index() {
27     $this->view->listproduct = $this->product->getAll();
28     $this->view->title = 'Produk';
29     $this->view->subtitle = 'Daftar Produk';
30     $this->view->render( view: 'product/index');
31 }
32
33 function add() {
34     $name = $_POST['name'];
35     $price = $_POST['price'];
36     $this->product->add($name, $price);
37     $this->redirect( view: 'product');
38 }
39
40 function edit($id) {
41     if(Session::get('role') == 3) {
42         $this->view->product = $this->product->getProduct($id);
43         $this->view->title = 'Produk';
44         $this->view->subtitle = 'Edit Produk';
45         $this->view->render( view: 'product/edit');
46     } else {
47         $this->view->rendererror();
48     }
49 }
```

Gambar 4.46 Kode Program Mengelola Data Produk

```
51 function update() {  
52     $id     = $_POST['id'];  
53     $name   = $_POST['name'];  
54     $price  = $_POST['price'];  
55     $this->product->update($id, $name, $price);  
56     $this->redirect( view: 'product');  
57 }  
58  
59 function delete($id) {  
60     $this->product->delete($id);  
61     $this->redirect( view: 'product');  
62 }
```

Gambar 4.47 Kode Program Mengelola Data Produk

4.3.11 Kode Program Melihat Data Produk

Kode program melihat data produk terletak pada kelas `view/product/index`, `controller/product`, dan `product_model`. Aktor marketing dan admin yang melakukan proses ini dan hanya dapat menampilkan daftar produk. Aktor memilih menu produk dan controller product akan memanggil model product, dan model product mengambil data dari *database* kemudian dilanjutkan ke controller product untuk ditampilkan pada view product. Sistem akan menampilkan daftar produk yang telah diinputkan bagian produksi. Penulisan kode program ini dapat dilihat pada Gambar 4.46 dalam method `index`.

4.3.12 Kode Program Mengelola Data Produksi

Kode program mengelola data produksi terletak pada kelas `view/production/index`, `view/production/edit`, `controller/production`, dan `production_model`. Aktor produksi yang bertugas mengelola data pesanan. Untuk memasukkan data produksi aktor marketing memilih tombol *edit* program akan menampilkan form *edit*. Dan saat pengguna memilih simpan maka controller production akan membaca data yang telah diubah dan menruskan ke model production untuk di *update* pada *database*. Penulisan kode program ini dapat dilihat pada lampiran D.

```
26 function index(){
27     $production = $this->production->getAll();
28     $this->view->listproduction = $production;
29     $tanggal = null;
30     for($i = 0; $i < count($production); $i++) {
31         $date[$i] = new DateTime($production[$i]['date_product']);
32         $date[$i]->add(new DateInterval( interval_spec: 'P06D'));
33         $tanggal = $date[$i]->format('Y m d');
34     }
35
36     $this->view->expired = $tanggal;
37     $this->view->title = 'Produksi';
38     $this->view->subtitle = 'Daftar Produksi';
39     $this->view->render( view: 'production/index');
40 }
41
42 function edit($id) {
43     $this->view->listproduction = $this->production->getData($id);
44     $this->view->title = 'Produksi';
45     $this->view->subtitle = 'Edit Data Produksi';
46     $this->view->render( view: 'production/edit');
47 }
48
49 function update() {
50     $id = $_POST['id'];
51     $amount = $_POST['amount'];
52     $this->production->update($id, $amount);
53     $this->redirect( view: 'production');
54 }
```

Gambar 4.48 Kode Program Mengelola Data Produksi

4.3.13 Kode Program Melihat Data Produksi

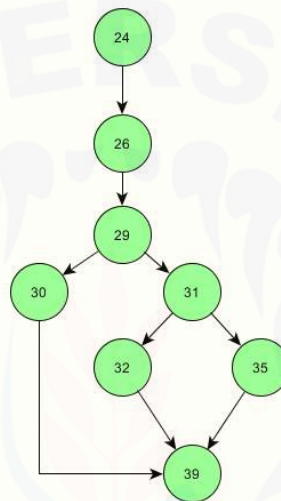
Kode program melihat data produksi terletak pada kelas view/production/index, controller/production, dan production_model. Aktor marketing dan admin yang melakukan proses ini dan hanya dapat menampilkan daftar produk. Aktor memilih menu produk dan controller production akan memanggil model production, dan model production mengambil data dari *database* kemudian dilanjutkan ke controller production untuk ditampilkan pada view production. Sistem akan menampilkan daftar produk yang telah diinputkan bagian produksi. Penulisan kode program ini dapat dilihat pada Gambar 4.48 dalam method index.

4.4 Pengujian

4.4.1 White Box

Pengujian *White Box* pada sistem perencanaan jadwal produksi yang dikembangkan penelitian ini adalah dengan menghitung *Cyclomatic Complexity* (CC) pada setiap fitur. Perhitungan CC untuk beberapa fitur dapat dilihat pada penjelasan dibawah ini. Untuk fitur yang lain dapat dilihat pada Lampiran E.

1. Kelas login/auth



Gambar 4.49 Flow Graph login/auth

Gambar 4.49 merupakan flow graph dari kelas login dan method auth yang dijalankan saat pengguna melakukan login.

$$CC = E - N + 2 = 9 - 8 + 2 = 3$$

Tabel 4.18 Test Case Kelas login/auth

Test Case 1	Jika <i>username</i> atau <i>password</i> salah
Target Yang Diharapkan	Tidak berhasil <i>login</i> karena <i>username</i> atau <i>password</i> salah
Hasil Pengujian	Benar
Path / Jalur	24, 25, 26, 29, 32, 35, 39
Test Case 2	Jika akun telah dinonaktifkan
Target Yang Diharapkan	Tidak berhasil <i>login</i> karena akun telah dinonaktifkan
Hasil Pengujian	Benar
Path / Jalur	24, 25, 26, 29, 32, 31, 39

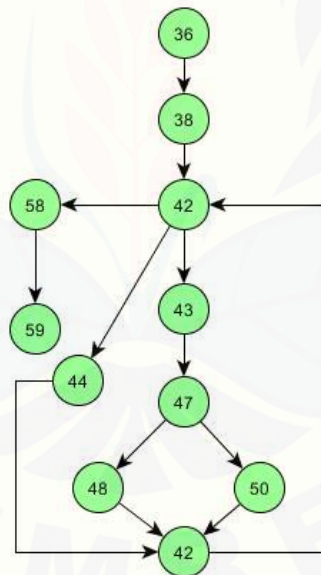
Dilanjutkan

Lanjutan

Test Case 3	Jika berhasil melakukan <i>login</i>
Target Yang Diharapkan	Berhasil melakukan <i>login</i>
Hasil Pengujian	Benar
Path / Jalur	24, 25, 26, 29, 30, 39

Tabel 4.18 menjelaskan tentang test case dari flow graph yang telah digambarkan sebelumnya. Test case menjelaskan tentang kemungkinan-kemungkinan yang terjadi dan hasil yang didapatkan serta menjelaskan path apa saja yang dikerjakan saat kode program dijalankan. Dari hasil perhitungan CC didapatkan bahwa kode program untuk kelas login dan method auth tidak rumit karena masih ada batas path yang dijalankan.

2. Kelas forecast/add (Menghitung Peramalan)



Gambar 4.50 *Flow Graph* Kelas forecast/add

Gambar 4.50 merupakan flow graph dari kelas forecast dan method add yang dijalankan saat pengguna menghitung peramalan untuk periode selanjutnya.

$$CC = E - N + 2 = 13 - 11 + 2 = 4$$

Tabel 4.19 Test Case Kelas forecast/add

Test Case 1	Jika berhasil menambahkan data ke <i>database</i> hasil peramalan penjualan untuk periode berikutnya
Target Yang Diharapkan	Berhasil menambahkan data ke <i>database</i> hasil peramalan penjualan untuk periode berikutnya
Hasil Pengujian	Benar
Path / Jalur	36, 38, 42, 43, 47, 50, 42, 42
Test Case 2	Jika tidak berhasil menambahkan data ke <i>database</i> hasil peramalan penjualan untuk periode berikutnya
Target Yang Diharapkan	Tidak berhasil menambahkan data ke <i>database</i> hasil peramalan penjualan untuk periode berikutnya
Hasil Pengujian	Benar
Path / Jalur	36, 38, 42, 43, 47, 48, 42, 42
Test Case 3	Jika berhasil menghitung perulangan sampai pada jumlah data dikurangi 1 maka data tersebut bernilai data itu sendiri
Target Yang Diharapkan	Berhasil menghitung perulangan sampai pada jumlah data dikurangi 1 maka data tersebut bernilai data itu sendiri
Hasil Pengujian	Benar
Path / Jalur	36, 38, 42, 44, 42, 42

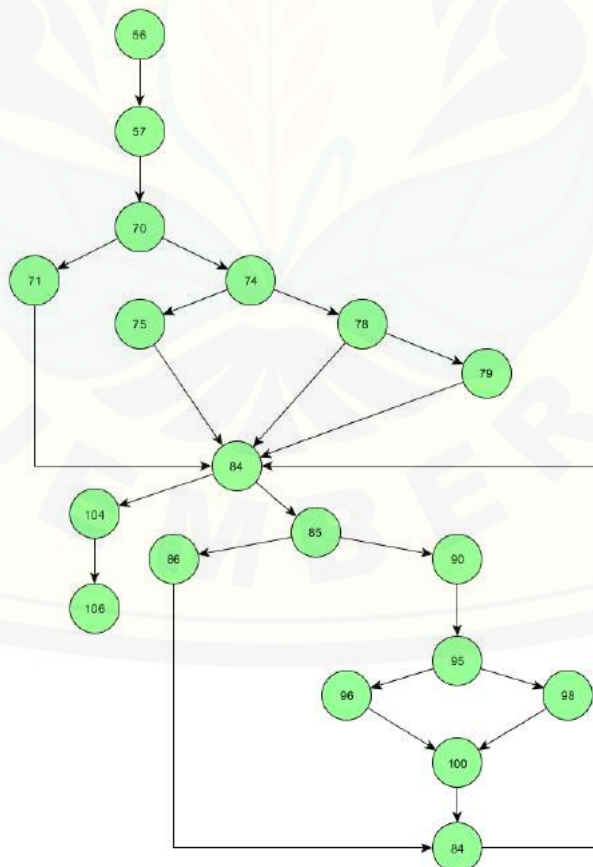
Dilanjutkan

Lanjutan

Test Case 4	Jika perulangan lebih dari jumlah data dikurangi 1 maka menghitung peramalan pada periode berikutnya
Target Yang Diharapkan	Berhasil menghitung peramalan pada periode berikutnya
Hasil Pengujian	Benar
Path / Jalur	36, 38, 42, 43, 47, 50, 42, 42

Tabel 4.19 menjelaskan tentang test case dari flow graph yang telah digambarkan sebelumnya. Dari hasil perhitungan CC didapatkan bahwa kode program untuk kelas forecast dan method add tidak rumit karena masih ada batas path yang dijalankan.

3. Kelas sales/update



Gambar 4.51 Flow Graph Kelas sales/update

Gambar 4.51 merupakan flow graph dari kelas sales dan method update yang dijalankan saat pengguna menginputkan data penjualan atau mengubah data penjualan dan system menghitung parameter-parameter yang dibutuhkan saat perhitungan peramalan.

$$CC = E - N + 2 = 24 - 19 + 2 = 7$$

Tabel 4.20 Test Case Kelas sales/update

Test Case 1	Jika produk yang diubah adalah produk dengan id 1 dan berhasil mengubah data
Target Yang Diharapkan	Berhasil mengubah data dengan produk id 1
Hasil Pengujian	Benar
Path / Jalur	56, 57, 70, 71, 84, 85, 90, 95, 96, 100, 84, 84, 104, 106
Test Case 2	Jika produk yang diubah adalah produk dengan id 2 dan berhasil mengubah data
Target Yang Diharapkan	Berhasil mengubah data dengan produk id 2
Hasil Pengujian	Benar
Path / Jalur	56, 57, 70, 74, 75, 84, 85, 90, 95, 96, 100, 84, 84, 104, 106
Test Case 3	Jika produk yang diubah adalah produk dengan id 3 dan berhasil mengubah data
Target Yang Diharapkan	Berhasil mengubah data dengan produk id 3
Hasil Pengujian	Benar
Path / Jalur	56, 57, 70, 74, 78, 79, 84, 85, 90, 95, 96, 100, 84, 84, 104, 106
Test Case 4	Jika perulangan kurang dari jumlah data dikurangi 1 maka nilai parameter peramalan adalah tetap dan tidak mengubah data
Target Yang Diharapkan	Tidak mengubah data
Hasil Pengujian	Benar
Path / Jalur	56, 57, 70, 74, 78, 79, 84, 85, 86, 100, 84, 84, 104, 106
Test Case 5	Jika perulangan lebih dari jumlah data dikurangi 1 maka mengubah data parameter peramalan dan penjualan
Target Yang Diharapkan	Berhasil mengubah data parameter peramalan dan penjualan

Dilanjutkan

Lanjutan

Hasil Pengujian	Benar
Path / Jalur	56, 57, 70, 74, 78, 79, 84, 85, 90, 95, 96, 100, 84, 84, 104, 106
Test Case 6	Jika penjualan tidak sama dengan 0 maka menghitung nilai <i>error</i>
Target Yang Diharapkan	Berhasil menghitung nilai <i>error</i>
Hasil Pengujian	Benar
Path / Jalur	56, 57, 70, 74, 78, 79, 84, 85, 90, 95, 96, 100, 84, 84, 104, 106
Test Case 7	Jika penjualan tidak sama dengan 0 maka nilai <i>error</i> adalah 100
Target Yang Diharapkan	Nilai <i>error</i> sama dengan 100
Hasil Pengujian	Benar
Path / Jalur	56, 57, 70, 74, 78, 79, 84, 85, 90, 95, 98, 100, 84, 84, 104, 106

Tabel 4.20 menjelaskan tentang test case dari flow graph yang telah digambarkan sebelumnya. Dari hasil perhitungan CC didapatkan bahwa kode program untuk kelas sales dan method update tidak rumit karena masih ada batas path yang dijalankan.

4.4.2 Black Box

Black Box Testing adalah cara menguji fungsionalitas sistem dengan menjalankan sistem dan memeriksa ada tidaknya bug atau kesalahan-kesalahan pada sistem. Pengujian dilakukan pada proses inputan, output yang dihasilkan, kesesuaian hasil peramalan dengan perhitungan manual dan beberapa fungsi-sungsi pendukung yang ada pada sistem. Pengujian *Black Box* untuk beberapa fitur dapat dilihat pada penjelasan di bawah ini. Untuk fitur lain dapat dilihat pada Lampiran F.

Tabel 4.21 Hasil Pengujian Fitur Mengelola Data Akun Pengguna

No	Fitur	Aksi	Hasil	Kesimpulan	
				Berhasil	Tidak
1	Melihat data akun	Memilih menu akun	Menampilkan daftar akun pengguna	✓	

Dilanjutkan

Lanjutkan

No	Fitur	Aksi	Hasil	Kesimpulan	
				Berhasil	Tidak
2	Menambah akun	Memilih tombol “Tambah Akun”	Menampilkan form tambah akun	✓	
		Memilih tombol “Simpan”	<ul style="list-style-type: none"> • Menyimpan data ke <i>database</i> • Menampilkan daftar akun 	✓	
3	Mengubah data akun	Memilih tombol “ <i>Edit</i> ” pada salah satu data	Menampilkan form <i>Edit</i> akun	✓	
		Memilih tombol “Simpan”	<ul style="list-style-type: none"> • Menyimpan data ke <i>database</i> • Menampilkan form data akun 	✓	
4	Menghapus data akun	Memilih tombol “ <i>Delete</i> ” pada salah satu akun	Menghapus data dari <i>database</i>	✓	

Pada Tabel 4.21 didapatkan hasil pengujian black box untuk fitur mengelola data akun pengguna berhasil dijalankan tanpa ada kesalahan. Setiap proses pada fitur dapat dijalankan dengan baik.

Hasil pengujian black box pada fitur mengelola data diri adalah berhasil. Semua proses dapat dijalankan dengan baik. Hal ini dijelaskan pada Tabel 4.22.

Tabel 4.22 Hasil Pengujian Fitur Mengelola Data Diri

No	Fitur	Aksi	Hasil	Kesimpulan	
				Berhasil	Tidak
1	Mengubah Data Diri	<ul style="list-style-type: none"> Memilih tombol “Ubah <i>Password</i>” pada dashboard Atau memilih menu “Settings” 	Menampilkan form ubah data diri	✓	
		<ul style="list-style-type: none"> Mengisi form Memilih tombol simpan 	<ul style="list-style-type: none"> Menyimpan data ke <i>database</i> Menampilkan halaman <i>Login</i> 	✓	

Hasil pengujian black box pada fitur mengelola data penjualan adalah berhasil. Semua proses dapat dijalankan dengan baik. Hal ini dijelaskan pada Tabel 4.23.

Tabel 4.23 Hasil Pengujian Fitur Mengelola Data Penjualan

No	Fitur	Aksi	Hasil	Kesimpulan	
				Berhasil	Tidak
1	Melihat data penjualan	Memilih tombol penjualan	Menampilkan data penjualan	✓	
2	Menambah data penjualan	Memilih tombol <i>Edit</i>	Menampilkan form <i>Edit</i> data penjualan	✓	
		<ul style="list-style-type: none"> Mengisi form <i>Edit</i> data penjualan Memilih simpan 	<ul style="list-style-type: none"> Menyimpan data penjualan ke <i>database</i> Manampilkan daftar data penjualan 	✓	

Hasil pengujian black box pada fitur mengelola data hasil peramalan adalah berhasil. Semua proses dapat dijalankan dengan baik dan hasil peramalan sama dengan perhitungan manual. Hal ini dijelaskan pada Tabel 4.24.

Tabel 4.24 Hasil Pengujian Fitur Mengelola Data Hasil Peramalan

No	Fitur	Aksi	Hasil	Kesimpulan	
				Berhasil	Tidak
1	Melihat data peramalan	Memilih menu penjualan	Menampilkan data produk	✓	
2		Memilih produk	<ul style="list-style-type: none">• Menampilkan data peramalan	✓	
3	Menghitung peramalan	Memilih tombol “Hitung Peramalan”	<ul style="list-style-type: none">• Melakukan perhitungan peramalan• Menampilkan data peramalan dan penjualan	✓	

BAB 6

PENUTUP

6.1 Kesimpulan

Dari penelitian yang telah dilakukan didapatkan beberapa kesimpulan untuk menjawab pertanyaan-pertanyaan pada penelitian ini.

1. Dalam melakukan perhitungan peramalan penjualan dengan metode *Winter Exponential Smoothing* diperlukan 3 tahap yaitu menentukan jumlah musim, menentukan nilai α , β , γ dan melakukan perhitungan dengan berdasarkan pada nilai aktual penjualan. Pada penelitian ini jumlah musim yang digunakan adalah 7 karena data yang digunakan merupakan data harian. Dalam proses menentukan nilai α , β , γ peneliti menggunakan aplikasi Solver pada MS. Excel. Perhitungan peramalan penjualan dihitung dengan menggunakan rumus yang telah dijelaskan pada Persamaan 2.1 sampai dengan Persamaan 2.4 dibuat dalam bentuk Excel sebelum diterapkan pada sistem. Dengan data sebanyak 3 bulan didapatkan nilai *Error* sebesar 24% untuk Prol Tape Besar, 24% untuk Prol Tape Kecil dan 21% untuk Pia Tape. Dimana nilai MAPE yang rata-rata dibawah 30% dikatakan cukup akurat dan dapat dijadikan acuan untuk menentukan jumlah produk yang akan diproduksi periode selanjutnya. Akan tetapi terdapat beberapa hari yang nilai kesalahannya lebih besar dari 30% dan dikatakan tidak akurat. Hal ini dikarenakan data pada hari tersebut mengalami perubahan yang tidak sesuai dengan pola data.

2. Sistem dibangun dengan menggunakan model SDLC *Agile*. Proses diawali dari pencarian data dan analisis data. Pencarian data dilakukan dengan wawancara dan dokumentasi data. Data yang digunakan pada penelitian ini adalah data penjualan, data pesanan, dan data produksi. Dengan data tersebut dibuat suatu sistem dengan beberapa fitur utama yaitu pengelolaan penjualan, pengelolaan pesanan dan pengelolaan produksi. Dalam membangun sistem hal yang dilakukan setelah menentukan fitur adalah membuat desain sistem. Pada penelitian ini peneliti menggunakan Unified Modelling Language (UML) sebagai media desain. Pembuatan kode program didasarkan pada desain yang telah dibuat. Kode program

dibuat dengan berdasar perhitungan manual peramalan penjualan yang telah dibuat sebelumnya. Program yang telah dibuat kemudian diuji dan diperbaiki apabila terdapat ketidaksesuaian dengan desain.

6.2 Saran

Dari kekurangan dan permasalahan pada penelitian ini, didapatkan beberapa saran sebagai berikut :

1. Penentuan nilai konstanta α , β , dan γ sebaiknya disistemkan. Sehingga sistem dapat mencari nilai α , β , dan γ yang memiliki nilai MAPE minimum. Dengan begitu nilai α , β , dan γ dapat sesuai dengan penambahan data dan bersifat dinamis.
2. Penentuan musim pada perhitungan peramalan menggunakan metode *Winter Exponential Smoothing* lebih baik dihitung berdasarkan pola musim pada data yang diteliti.
3. Adanya penanganan khusus untuk data yang bernilai 0 agar tidak mempengaruhi proses perhitungan peramalan selanjutnya.

DAFTAR PUSTAKA

- Amir, M., Khan, K., Khan, A., & Khan, M. (2013). An Appraisal of Agile Software Development Process. *International Journal of Advanced Science and Technology*, 58, 75–86. <https://doi.org/10.14257/ijast.2013.58.07>
- Anderson, D., Sweeney, D., Williams, T., Camm, J., & Martin, R. (2011). Time Series Analysis and Forecasting. *An Introduction to Management Science: Quantitative Approaches to Decision Making*, 1–896.
- Camm, J., Cochran, J., Fry, M., Ohlmann, J., & Anderson, D. (2015). Time Series Analysis and Forecasting. In *Essentials of Business Analytics* (1st ed., pp. 202–250). Stamford: Cengage Learning.
- Da Veiga, C. P., Da Veiga, C. R. P., Catapan, A., Tortato, U., & Da Silva, W. V. (2014). Demand forecasting in food retail: A comparison between the Holt-Winters and ARIMA models. *WSEAS Transactions on Business and Economics*, 11(1), 608–614.
- Mccabe, J. (1976). A Complexity Measure. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, SE-2(4), 308–320.
- Montgomery, D. C., Jennings, C. L., & Kulahci, M. (2011). *Introduction to Time Series Analysis and Forecasting*. *Journal of Chemical Information and Modeling* (Vol. 53). <https://doi.org/10.1017/CBO9781107415324.004>
- Onlinehelp.tableau.com. (n.d.). How Forecasting Works in Tableau. Retrieved May 7, 2017, from http://onlinehelp.tableau.com/current/pro/desktop/en-us/forecast_how_it_works.html
- Pal, A., Gaur, M. K., & Kumarkasdekar, D. (2015). A Study of Time Series Model for Forecasting of Boot in Shoe Industry, 8(8), 143–152.
- Strasheim, J. J. (2012). Demand Forecasting for Motor Vehicle Spare Parts. *The South African Journal of Industrial Engineering*, 6(2), 18–29. <https://doi.org/10.7166/6-2-411>
- Winston, W. L., & Albright, S. C. (2016). Regression And Forecasting Models. In *Practical Management Science* (5th ed., pp. 785–846). Boston: Cengage Learning.

LAMPIRAN

A. Use Case Scenario

1. Use Case Scenario Login

Tabel A. 1 Use Case Scenario Login (Admin)

Nomor Use Case	Use Case 1
Nama Use Case	Login – Admin
Aktor	Admin
Deskripsi	Aktor melakukan proses autentifikasi masuk ke sistem
Pre Kondisi	Aktor masuk kehalaman <i>Login</i> sistem
Pra Kondisi	Aktor melakukan <i>Login</i> dan masuk ke halaman dashboard Admin
Normal Flow	
Aksi Aktor	Reaksi Sistem
1. <i>Input username dan password</i>	
2. Memilih button <i>Login</i>	
	3. Melakukan cek <i>username</i> dan <i>password</i> di <i>database</i> sistem
	4. Menampilkan halaman dashboard Admin
Alternative Flow jika Akun Nonaktif	
	3. Melakukan cek <i>username</i> dan <i>password</i> di <i>database</i> sistem
	4. Menampilkan alert “Akun Anda Telah Dinonaktifkan”
Alternative Flow jika <i>username</i> atau <i>password</i> Salah	
	3. Melakukan cek <i>username</i> dan <i>password</i> di <i>database</i> sistem
	4. Menampilkan alert “ <i>Username</i> atau <i>Password</i> Salah”

Tabel A. 2 Use Case Scenario Login (Admin)

Nomor Use Case	Use Case 1
Nama Use Case	Login – Marketing
Aktor	Marketing

Dilanjutkan

Lanjutan

Deskripsi	Aktor melakukan proses autentifikasi masuk ke sistem
Pre Kondisi	Aktor masuk kehalaman <i>Login</i> sistem
Pra Kondisi	Aktor melakukan <i>Login</i> dan masuk ke halaman dashboard Marketing
Normal Flow	
Aksi Aktor	Reaksi Sistem
1. <i>Input username dan password</i>	
2. Mengklik button <i>Login</i>	
	3. Mengecek <i>username dan password</i> di <i>database</i> sistem
	4. Menampilkan halaman dashboard
Alternative Flow jika Akun Nonaktif	
	3. Mengecek <i>username dan password</i> di <i>database</i> sistem
	4. Menampilkan alert “Akun Anda Telah Dinonaktifkan”
Alternative Flow jika <i>Password</i> Salah	
	3. Mengecek <i>username dan password</i> di <i>database</i> sistem
	4. Menampilkan alert “ <i>Username</i> atau <i>Password</i> Salah”

Tabel A. 3 Use Case Scenario *Login* (Admin)

Nomor <i>Use Case</i>	<i>Use Case 1</i>
Nama <i>Use Case</i>	<i>Login – Produksi</i>
Aktor	Produksi
Deskripsi	Aktor melakukan proses autentifikasi masuk ke sistem
Pre Kondisi	Aktor masuk kehalaman <i>Login</i> sistem
Pra Kondisi	Aktor melakukan <i>Login</i> dan masuk ke halaman dashboard Produksi
Normal Flow	
Aksi Aktor	Reaksi Sistem
1. <i>Input username dan password</i>	
2. Mengklik button <i>Login</i>	

Dilanjutkan

Lanjutan

	3. Mengecek <i>username</i> dan <i>password</i> di <i>database</i> sistem
	4. Menampilkan halaman dashboard
Alternative Flow jika Akun Nonaktif	
	3. Mengecek <i>username</i> dan <i>password</i> di <i>database</i> sistem
	4. Menampilkan alert “Akun Anda Telah Dinonaktifkan”
Alternative Flow jika <i>Password</i> Salah	
	3. Mengecek <i>username</i> dan <i>password</i> di <i>database</i> sistem
	4. Menampilkan alert “ <i>Username</i> atau <i>Password</i> Salah”

2. Use Case Scenario Melihat Data Pesanan

Tabel A. 4 Use Case Scenario Melihat Data Pesanan

Nomor Use Case	Use Case 5
Nama Use Case	Melihat Data Pesanan – Admin
Aktor	Admin
Deskripsi	Aktor melihat data pesanan
Pre Kondisi	Aktor berada di halaman dashboard
Pra Kondisi	Aktor melihat data pesanan
Normal Flow	
Aksi Aktor	Reaksi Sistem
1. Mengklik menu pesanan	
	2. Mengambil <i>data_order</i>
	3. Menampilkan halaman <i>order/index</i>

3. Use Case Scenario Melihat Data Penjualan

Tabel A. 5 Use Case Scenario Melihat Data Penjualan

Nomor Use Case	Use Case 8
Nama Use Case	Melihat Data Penjualan – Admin
Aktor	Admin
Deskripsi	Aktor melihat data penjualan
Pre Kondisi	Aktor berada di halaman dashboard, Aktor melihat list penjualan

Dilanjutkan

Lanjutan

Pra Kondisi	Aktor melihat detail dari penjualan
Normal Flow	
Aksi Aktor	Reaksi Sistem
1. Mengklik menu penjualan	
	2. Mengambil data_sales
	3. Menampilkan halaman sales/index

4. *Use Case* Scenario Melihat Data Hasil Peramalan

Tabel A. 6 *Use Case* Scenario Melihat Data Hasil Peramalan (Admin)

Nomor <i>Use Case</i>	<i>Use Case 9</i>
Nama <i>Use Case</i>	Melihat Hasil Peramalan – Admin
Aktor	Admin
Deskripsi	Aktor melihat data hasil peramalan
Pre Kondisi	Aktor berada di halaman dashboard, Aktor melihat list produk
Pra Kondisi	Aktor melihat detail dari peramalan
Normal Flow	
Aksi Aktor	Reaksi Sistem
1. Mengklik menu peramalan	
	2. Mengambil data_product
	3. Menampilkan halaman forecast/index
4. Mengklik salah satu produk	
	5. Mengambil detail data_sales dengan id produk yang dipilih
	6. Menampilkan halaman forecast/detail

Tabel A. 7 *Use Case* Scenario Melihat Data Hasil Peramalan (Produksi)

Nomor <i>Use Case</i>	<i>Use Case 9</i>
Nama <i>Use Case</i>	Melihat Hasil Peramalan – Admin
Aktor	Admin
Deskripsi	Aktor melihat data hasil peramalan
Pre Kondisi	Aktor berada di halaman dashboard, Aktor melihat list produk
Pra Kondisi	Aktor melihat detail dari peramalan
Normal Flow	

Dilanjutkan

Lanjutan

Aksi Aktor	Reaksi Sistem
1. Mengklik menu peramalan	
	2. Mengambil data_product
	3. Menampilkan halaman forecast/index
4. Mengklik salah satu produk	
	5. Mengambil detail data_sales dengan id produk yang dipilih
	6. Menampilkan halaman forecast/detail

5. *Use Case* Scenario Melihat Data Produk

Tabel A. 8 *Use Case* Scenario Melihat Data Produk

Nomor <i>Use Case</i>	<i>Use Case</i> 11
Nama <i>Use Case</i>	Melihat Data Produk – Admin
Aktor	Admin
Deskripsi	Aktor melihat data produk
Pre Kondisi	Aktor berada di halaman dashboard
Pra Kondisi	Aktor melihat daftar produk
Normal Flow	
Aksi Aktor	Reaksi Sistem
1. Mengklik menu produk	
	2. Mengambil data_product
	3. Menampilkan halaman product/index

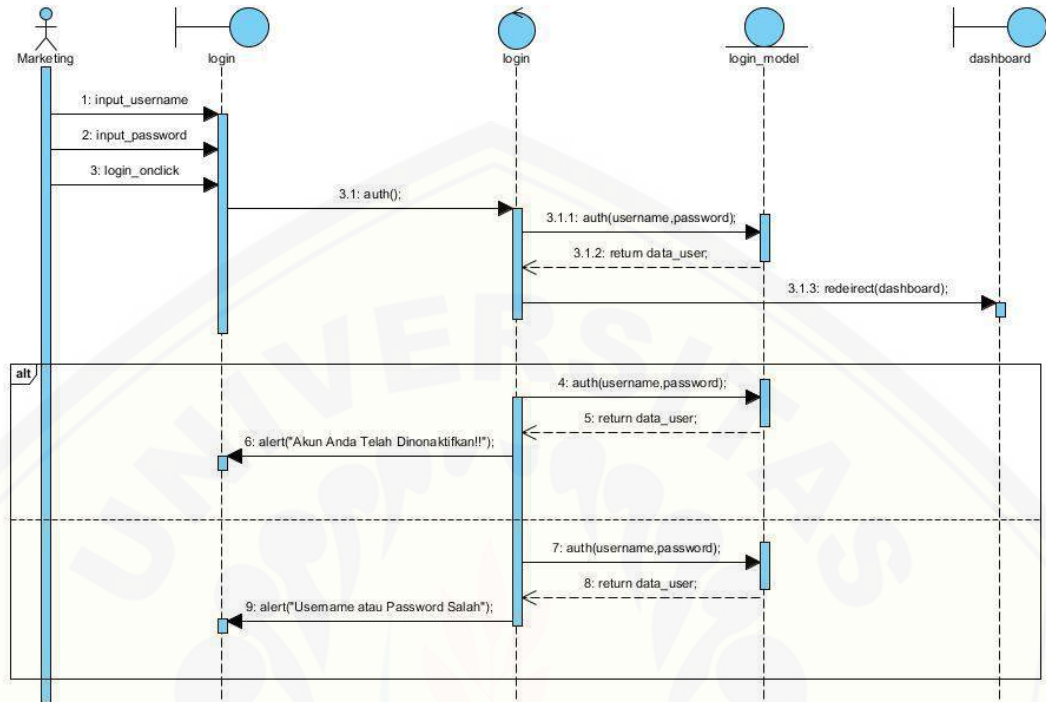
6. *Use Case* Scenario Melihat Data Produksi

Tabel A. 9 *Use Case* Scenario Melihat Data Produksi

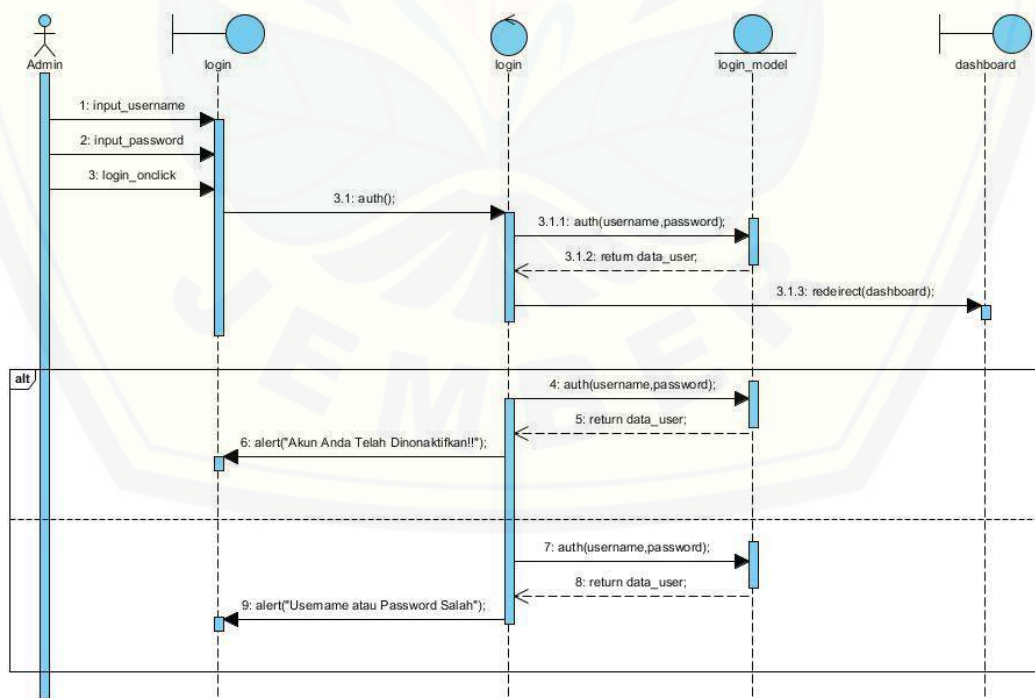
Nomor <i>Use Case</i>	<i>Use Case</i> 13
Nama <i>Use Case</i>	Melihat Data Produksi – Admin
Aktor	Admin
Deskripsi	Aktor melihat data produksi
Pre Kondisi	Aktor berada di halaman dashboard
Pra Kondisi	Aktor melihat daftar produksi
Normal Flow	
Aksi Aktor	Reaksi Sistem
1. Mengklik menu produksi	
	2. Mengambil data_production
	3. Menampilkan halaman production/index

B. Sequence Diagram

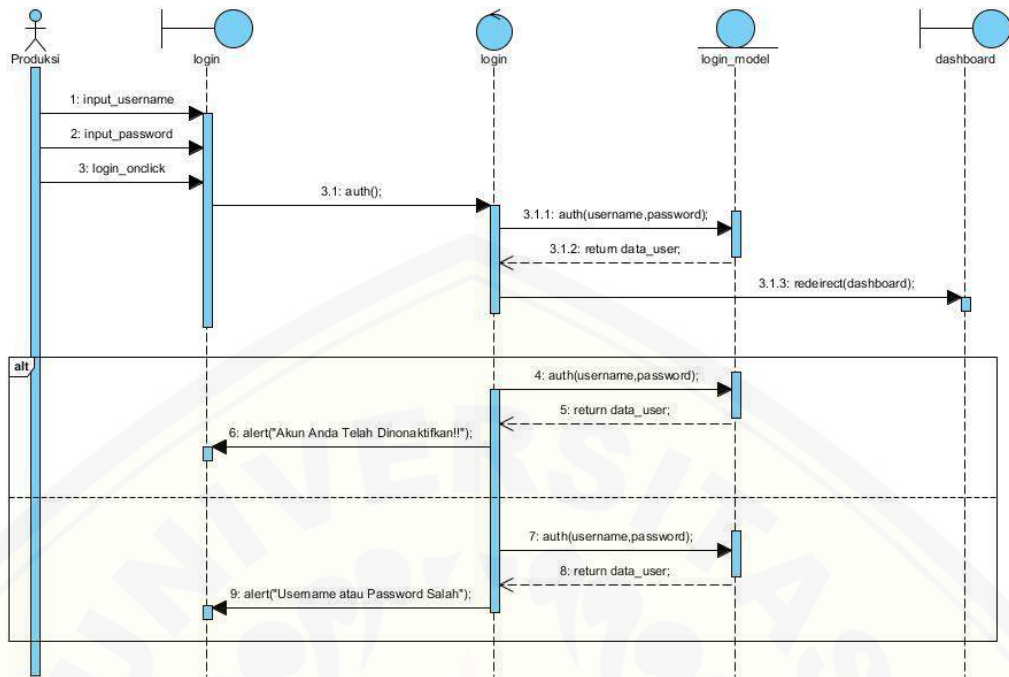
1. Sequence Diagram Login



Gambar B. 1 Sequence Diagram Login Marketing

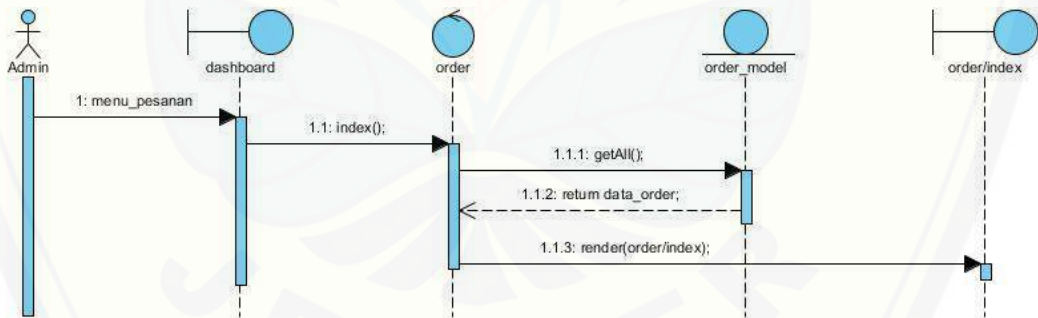


Gambar B. 2 Sequence Diagram Login Admin

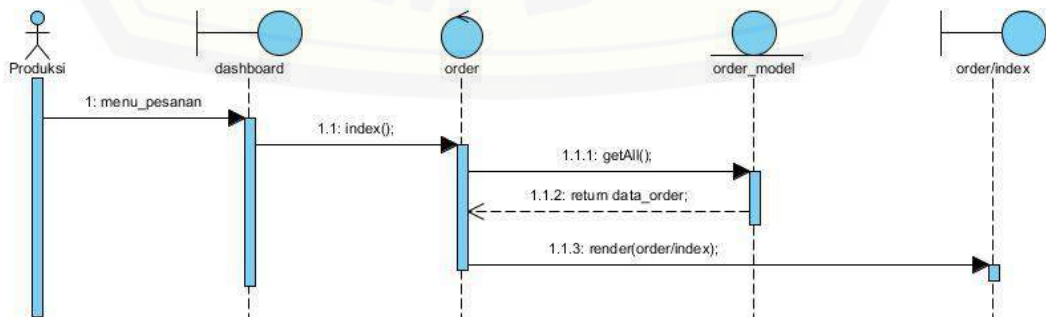


Gambar B. 3 *Sequence Diagram Login Produksi*

2. *Sequence Diagram Melihat Data Pesanan*

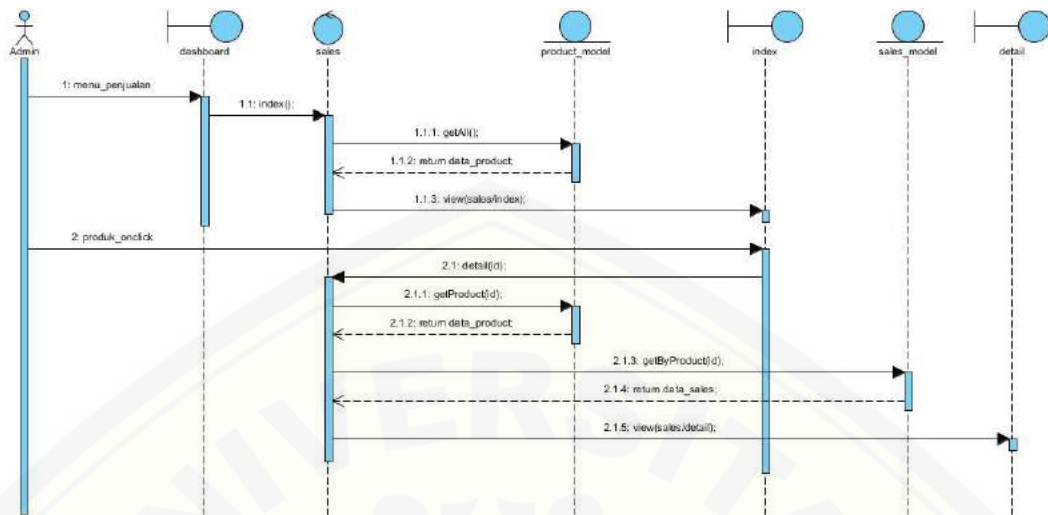


Gambar B. 4 *Sequence Diagram Melihat Data Pesanan Admin*



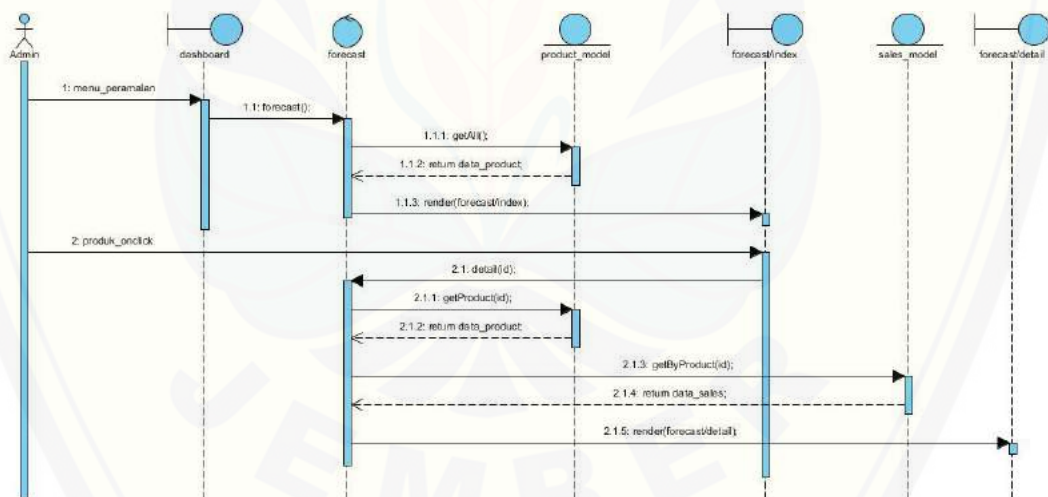
Gambar B. 5 *Sequence Diagram Melihat Data Pesanan Produksi*

3. *Sequence Diagram* Melihat Data Penjualan

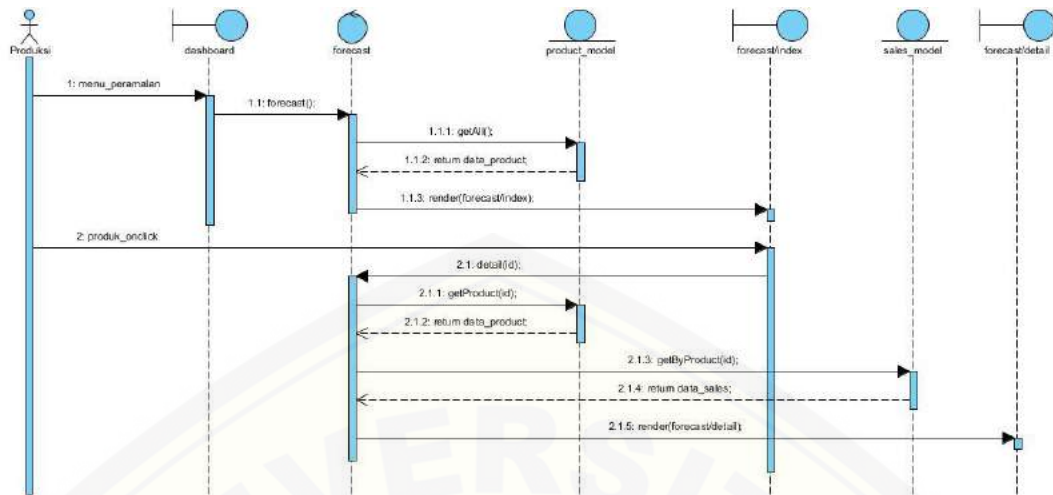


Gambar B. 6 *Sequence Diagram* Melihat Data Penjualan Admin

4. *Sequence Diagram* Melihat Data Hasil Peramalan

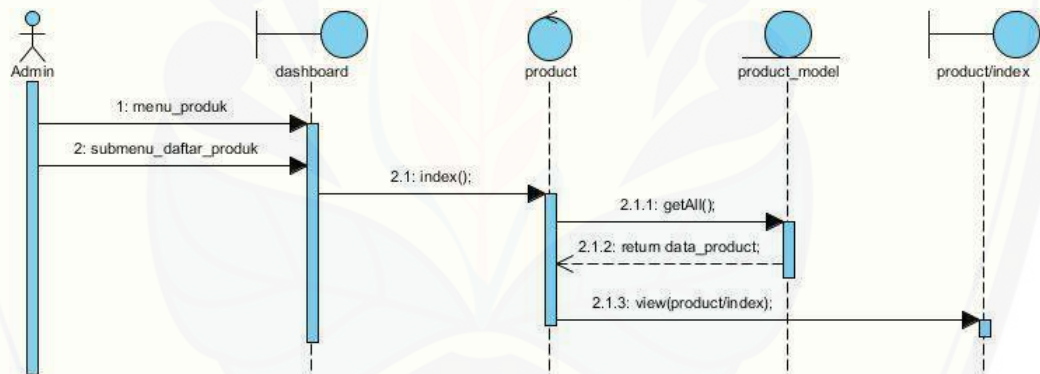


Gambar B. 7 *Sequence Diagram* Melihat Data Peramalan (Admin)



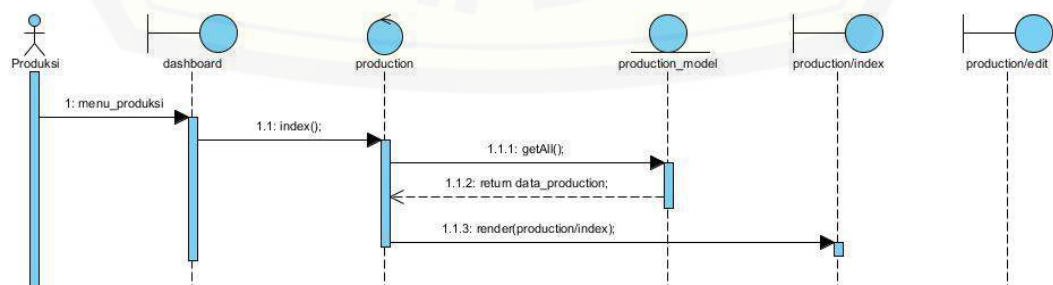
Gambar B. 8 *Sequence Diagram* Melihat Data Peramalan (Admin)

5. *Sequence Diagram* Melihat Data Produk



Gambar B. 9 *Sequence Diagram* Melihat Data Produk Admin

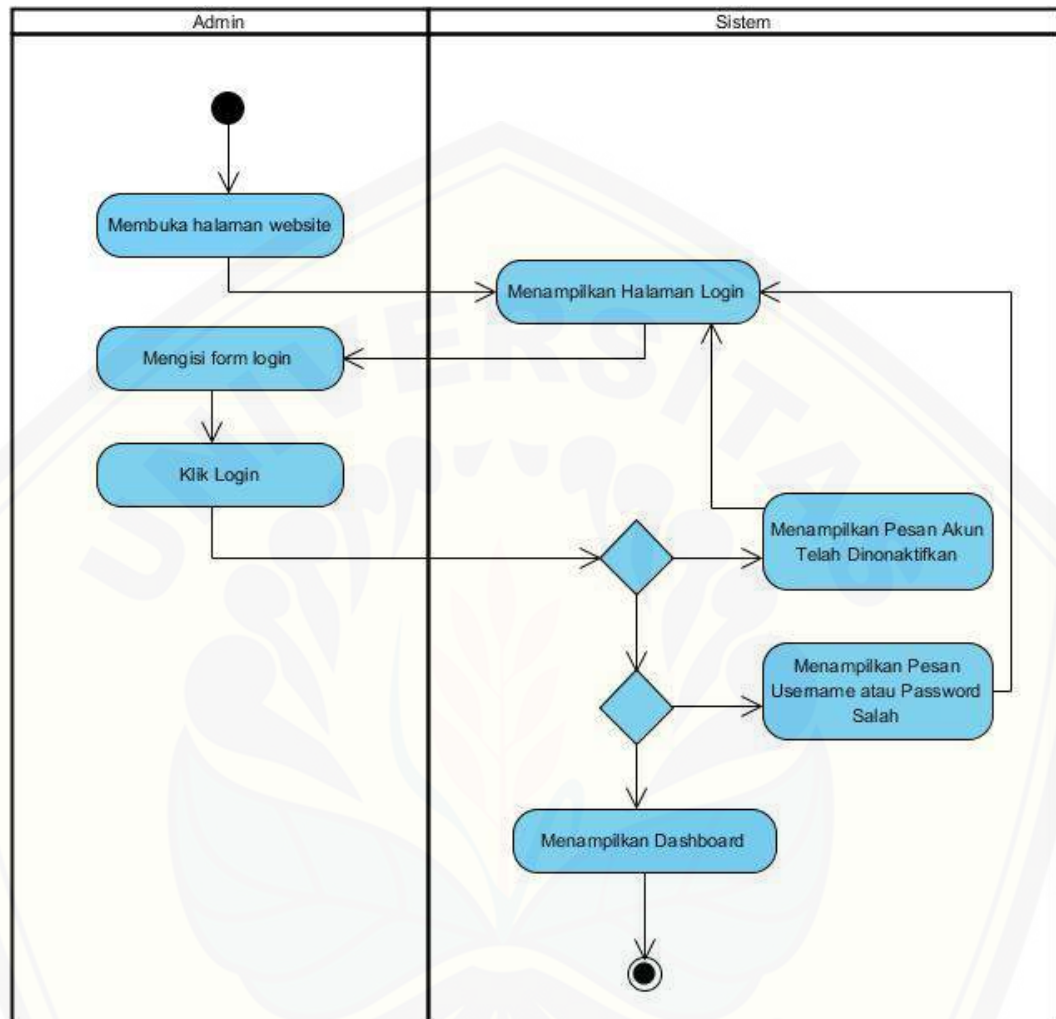
6. *Sequence Diagram* Melihat Data Produksi



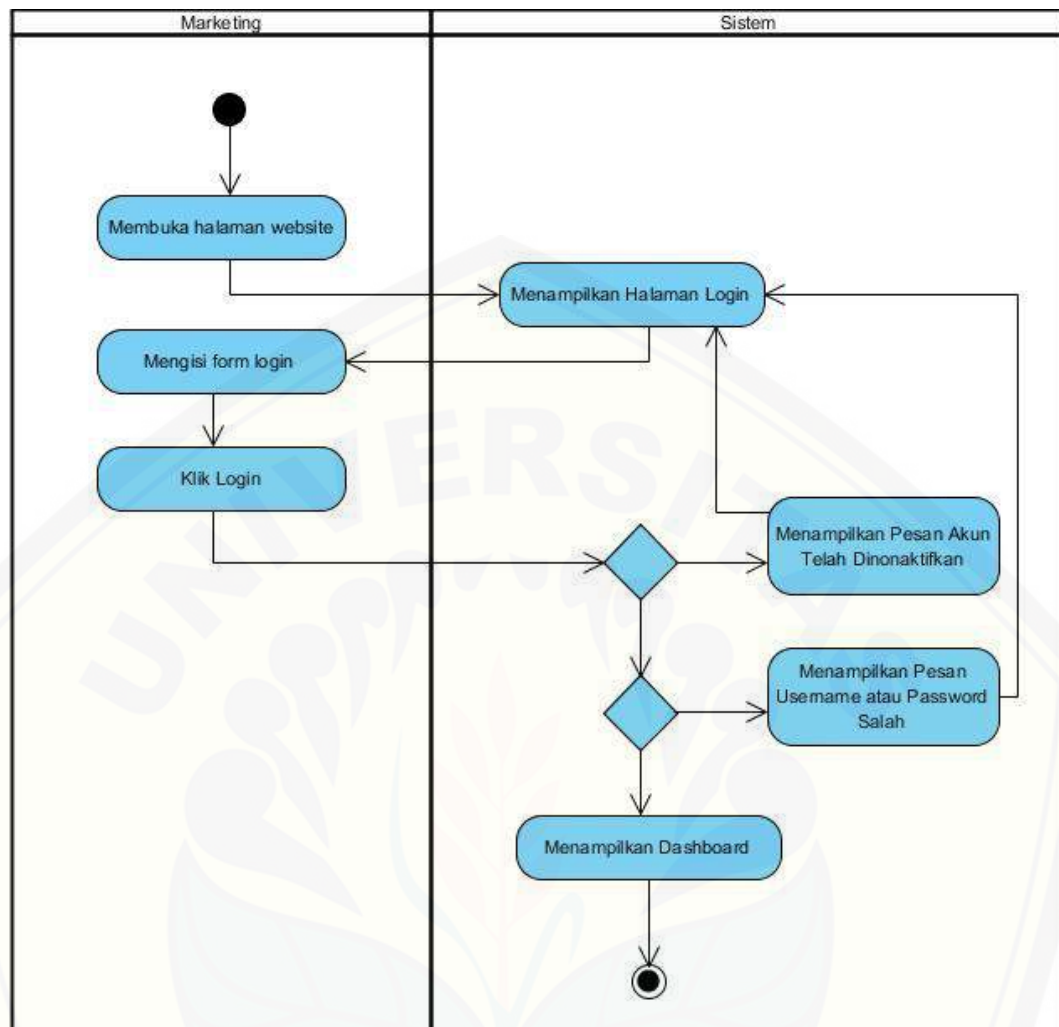
Gambar B. 10 *Sequence Diagram* Melihat Data Produksi Admin

C. Activity Diagram

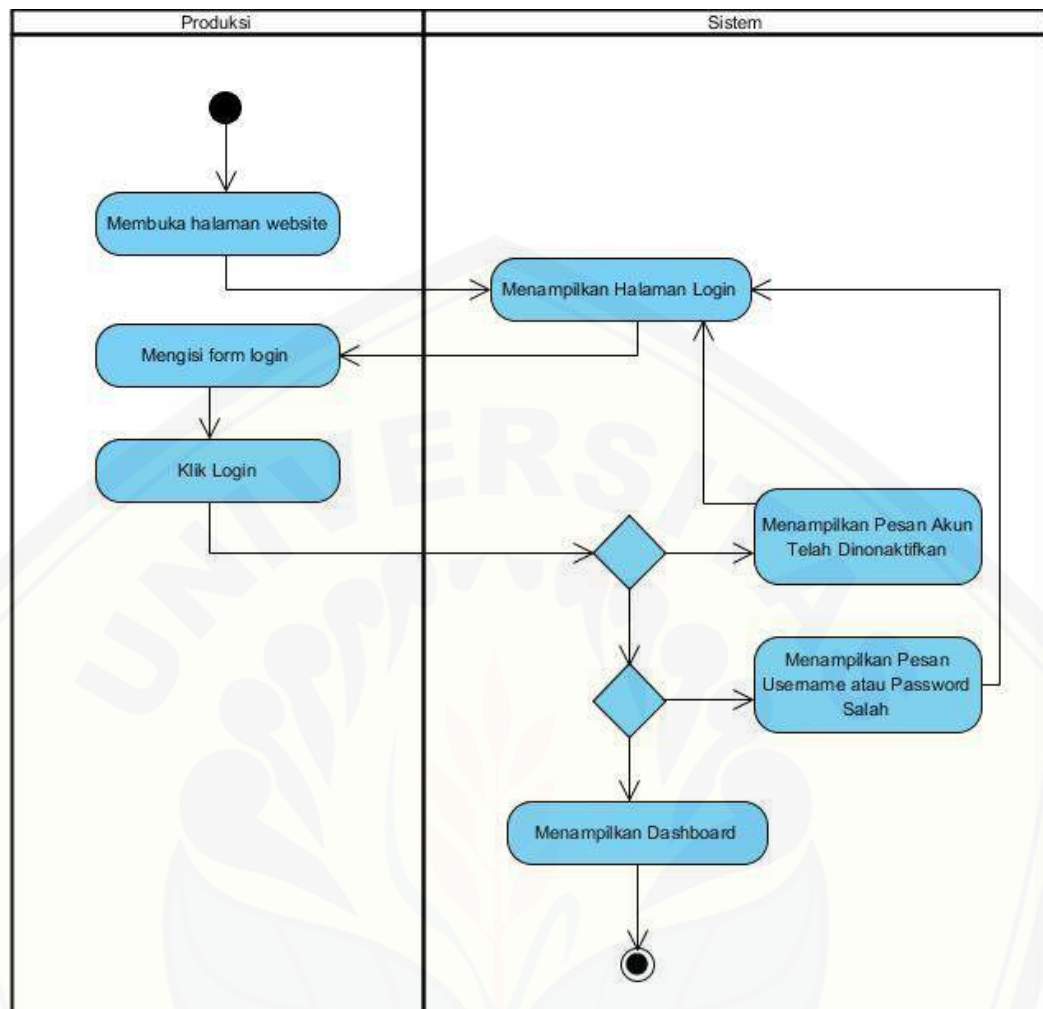
1. Activity Diagram Login



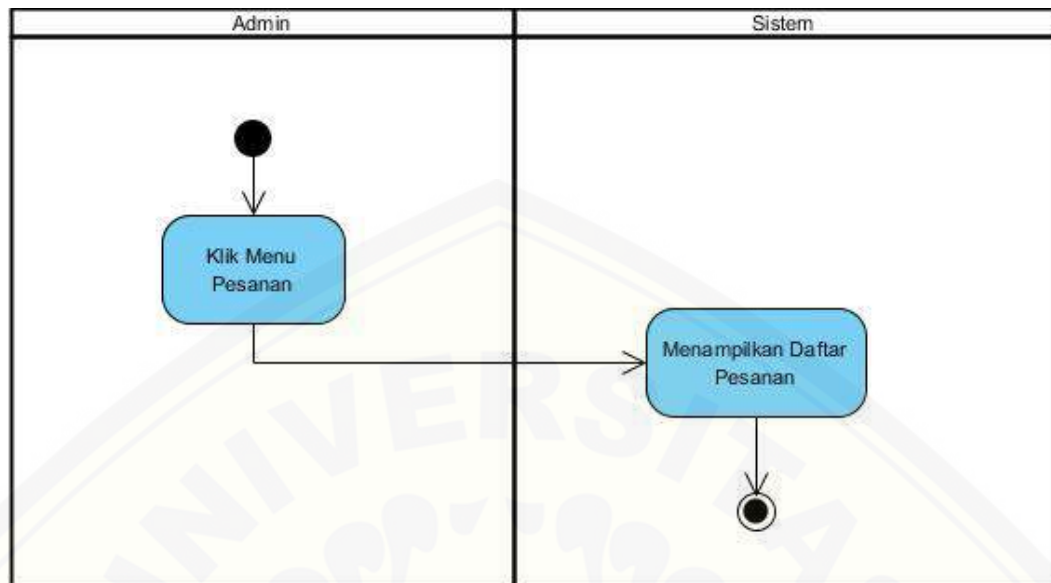
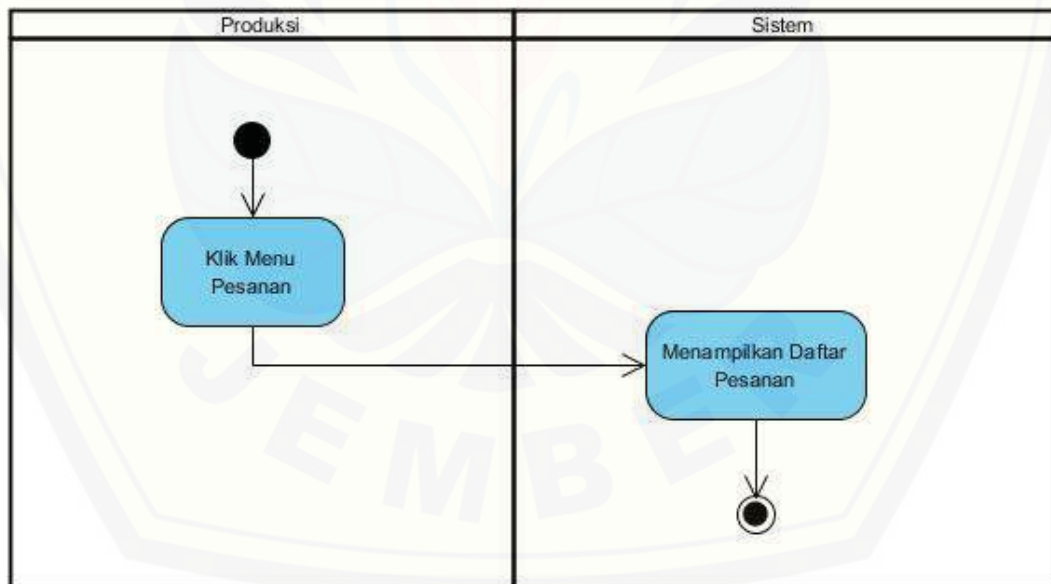
Gambar C. 1 Activity Diagram Login Marketing



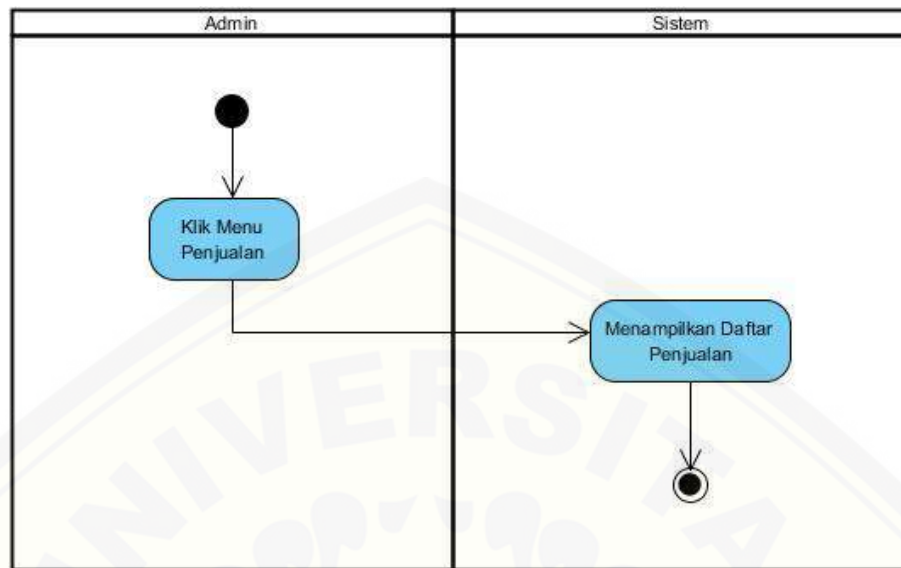
Gambar C. 2 Activity Diagram Login Marketing



Gambar C. 3 Activity Diagram Login Produksi

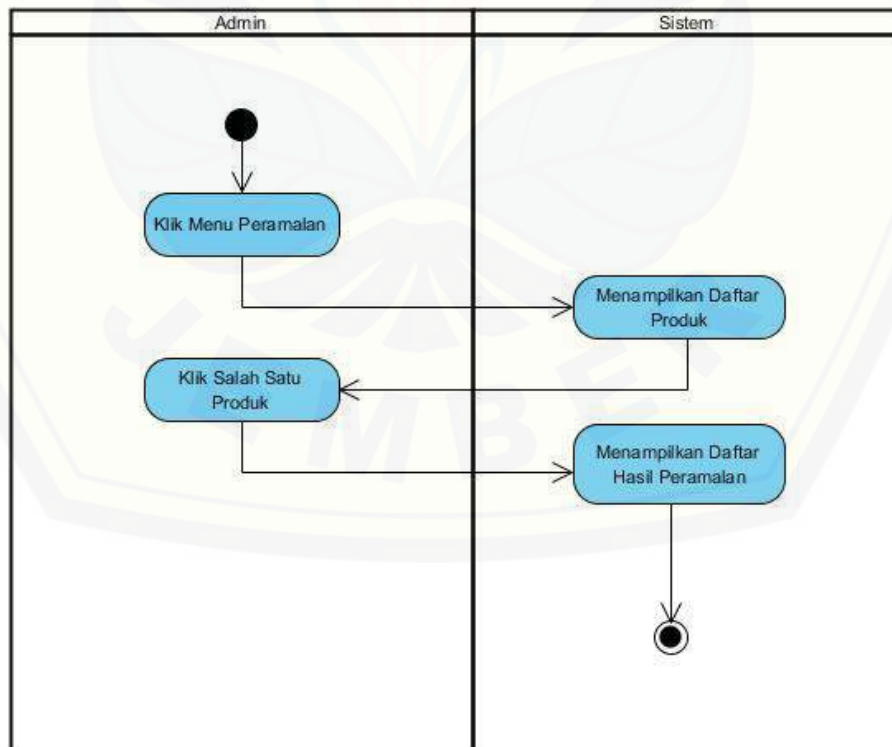
2. *Activity Diagram* Melihat Data PesananGambar C. 4 *Activity Diagram* Melihat Data Pesanan (Admin)Gambar C. 5 *Activity Diagram* Melihat Data Pesanan (Produksi)

3. *Activity Diagram* Melihat Data Penjualan

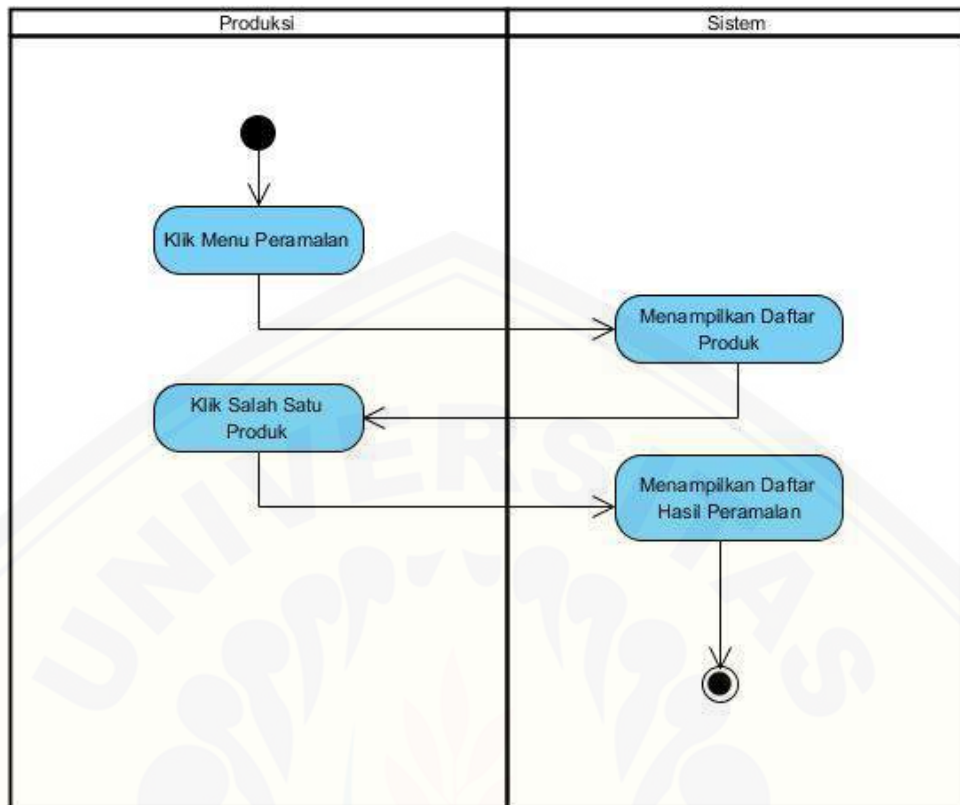


Gambar C. 6 *Activity Diagram* Melihat Data Penjualan

4. *Activity Diagram* Melihat Data Hasil Peramalan

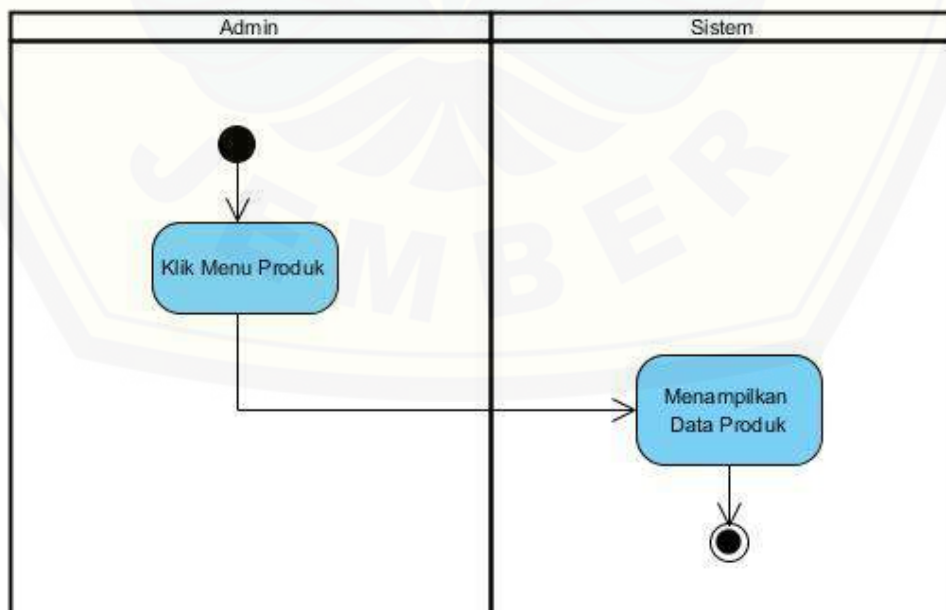


Gambar C. 7 *Activity Diagram* Melihat Data Hasil Peramalan (Admin)

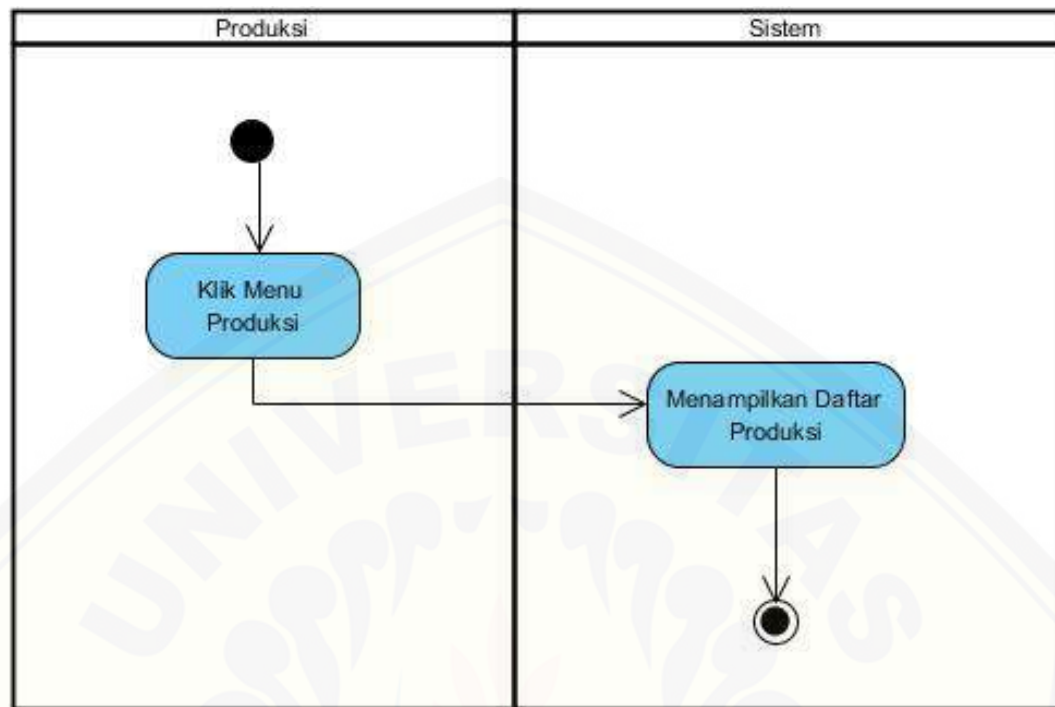


Gambar C. 8 Activity Diagram Melihat Data Hasil Peramalan (Produksi)

5. Activity Diagram Melihat Data Produk



Gambar C. 9 Activity Diagram Melihat Data Produk

6. *Activity Diagram* Melihat Data ProduksiGambar C. 10 *Activity Diagram* Melihat Data Produksi**D. Pengujian *Black Box***1. Pengujian Fitur *Login*Tabel D. 1 Hasil Pengujian Fitur *Login*

No	Fitur	Aksi	Hasil	Kesimpulan	
				Berhasil	Tidak
1	Melakukan autentikasi <i>login</i>	Memasukkan <i>username</i> dan <i>password</i> lalu klik <i>login</i>	Menampilkan halaman "Dashboard"	✓	

2. Pengujian Fitur Mengelola Data Pesanan

Tabel D. 2 Hasil Pengujian Fitur Mengelola Data Pesanan

No	Fitur	Aksi	Hasil	Kesimpulan	
				Berhasil	Tidak
1	Melihat data pesanan	<ul style="list-style-type: none"> Memilih menu pesanan 	Menampilkan data pesanan	✓	
2	Menambah data pesanan	<ul style="list-style-type: none"> Mengisi form tambah pesanan Memilih tombol “Simpan” 	<ul style="list-style-type: none"> Menyimpan data baru ke <i>database</i> Menampilkan data pesanan 	✓	
3	Mengubah data pesanan	Memilih <i>edit</i> pada salah satu data	Menampilkan form <i>edit</i> pesanan	✓	
		<ul style="list-style-type: none"> Mengisi form Memilih tombol “Simpan” 	<ul style="list-style-type: none"> Menyimpan data produk ke <i>database</i> Menampilkan daftar pesanan 	✓	
4	Menghapus data pesanan	Memilih tombol hapus	Menghapus data pesanan dari <i>database</i>	✓	

3. Pengujian Fitur Melihat Data Pesanan

Tabel D. 3 Hasil Pengujian Fitur Melihat Data Pesanan

No	Fitur	Aksi	Hasil	Kesimpulan	
				Berhasil	Tidak
1	Melihat data pesanan	• Memilih menu pesanan	Menampilkan daftar pesanan	✓	

4. Pengujian Fitur Melihat Data Penjualan

Tabel D. 4 Hasil Pengujian Fitur Melihat Data Penjualan

No	Fitur	Aksi	Hasil	Kesimpulan	
				Berhasil	Tidak
1	Melihat data penjualan	Memilih menu penjualan	Menampilkan daftar penjualan	✓	

5. Pengujian Fitur Melihat Data Hasil Peramalan

Tabel D. 5 Hasil Pengujian Fitur Melihat Data Hasil Peramalan

No	Fitur	Aksi	Hasil	Kesimpulan	
				Berhasil	Tidak
1	Melihat data hasil peramalan	Memilih menu peramalan	Menampilkan daftar produk	✓	
2		Memilih salah satu produk	Menampilkan data peramalan	✓	

6. Pengujian Fitur Mengelola Data Produk

Tabel D. 6 Hasil Pengujian Fitur Mengelola Data Produk

No	Fitur	Aksi	Hasil	Kesimpulan	
				Berhasil	Tidak
1	Melihat data produk	<ul style="list-style-type: none"> Memilih menu produk 	Menampilkan data produk	✓	
2	Menambah data produk	<ul style="list-style-type: none"> Mengisi form tambah produk Memilih tombol “Simpan” 	<ul style="list-style-type: none"> Menyimpan data baru ke <i>database</i> Menampilkan data produk 	✓	
3	Mengubah data produk	Memilih <i>edit</i> pada salah satu data	Menampilkan form <i>edit</i> produk	✓	
		<ul style="list-style-type: none"> Mengisi form Memilih tombol “Simpan” 	<ul style="list-style-type: none"> Menyimpan data produk ke <i>database</i> Menampilkan daftar produk 	✓	
4	Menghapus data produk	Memilih tombol hapus	Menghapus data produk dari <i>database</i>	✓	

7. Pengujian Fitur Melihat Data Produk

Tabel D. 7 Hasil Pengujian Fitur Melihat Data Produk

No	Fitur	Aksi	Hasil	Kesimpulan	
				Berhasil	Tidak
1	Melihat data produk	<ul style="list-style-type: none"> Memilih menu produk 	Menampilkan daftar produk	✓	

8. Pengujian Fitur Mengelola Data Produksi

Tabel D. 8 Hasil Pengujian Fitur Mengelola Data Produksi

No	Fitur	Aksi	Hasil	Kesimpulan	
				Berhasil	Tidak
1	Melihat data penjualan	Memilih tombol penjualan	Menampilkan data penjualan	✓	
2	Menambah data penjualan	Memilih tombol <i>edit</i>	Menampilkan form <i>edit</i> data penjualan	✓	
		<ul style="list-style-type: none"> Mengisi form <i>edit</i> data penjualan Memilih <i>simpan</i> 	<ul style="list-style-type: none"> Menyimpan data penjualan ke <i>database</i> Manampilkan daftar data penjualan 	✓	

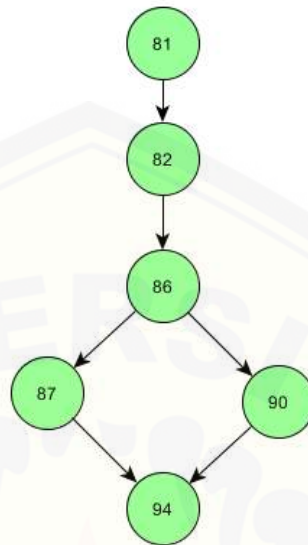
9. Pengujian Fitur Melihat Data Produksi

Tabel D. 9 Pengujian Fitur Melihat Data Produksi

No	Fitur	Aksi	Hasil	Kesimpulan	
				Berhasil	Tidak
1	Melihat data produksi	<ul style="list-style-type: none"> Memilih menu produksi 	Menampilkan daftar produksi	✓	

E. Pengujian White Box

1. Kelas users/changepass



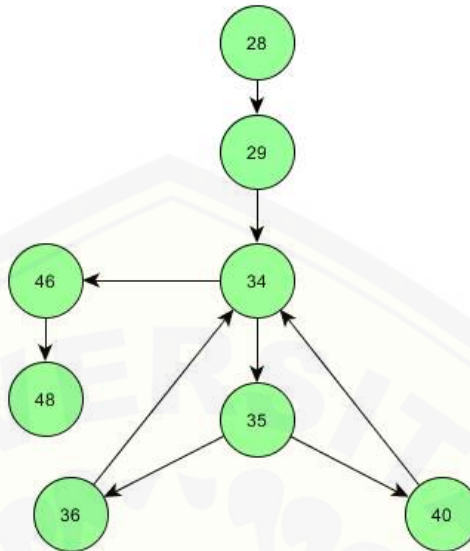
Gambar E. 1 Flow Graph Kelas users/changepass

$$CC = E - N + 2 = 6 - 6 + 2 = 2$$

Tabel E. 1 Test Case Kelas users/changepass

Test Case 1	Jika berhasil mengubah password
Target Yang Diharapkan	Berhasil mengubah password
Hasil Pengujian	Benar
Path / Jalur	81, 82, 86, 87, 94
Test Case 2	Jika gagal mengubah password karena konfirmasi password gagal
Target Yang Diharapkan	Tidak berhasil mengubah password
Hasil Pengujian	Benar
Path / Jalur	81, 82, 86, 90, 94

2. Kelas dashboard/index



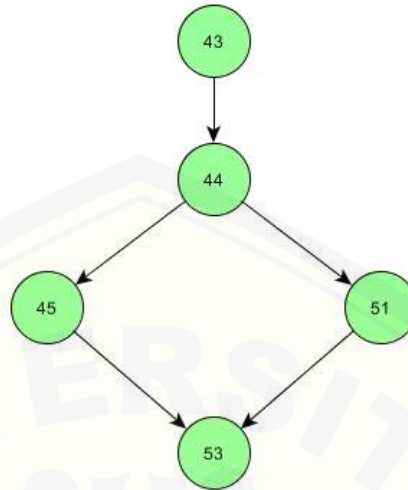
Gambar E. 2 Flow Graph Kelas dashboard/index

$$CC = E - N + 2 = 9 - 8 + 2 = 3$$

Tabel E. 2 Test Case Kelas dashboard/index

Test Case 1	Jika penjualan dan produksi tidak kosong maka penjualan dan produksi bernilai tetap
Target Yang Diharapkan	Berhasil menampilkan penjualan dan produksi dengan nilai tetap
Hasil Pengujian	Benar
Path / Jalur	28, 29, 34, 35, 36, 34, 46, 48
Test Case 2	Jika penjualan dan produksi kosong maka penjualan dan produksi bernilai 0
Target Yang Diharapkan	Berhasil menampilkan penjualan dan produksi yang bernilai 0
Hasil Pengujian	Benar
Path / Jalur	28, 29, 34, 35, 40, 34, 46, 48
Test Case 3	Jika berhasil menampilkan halaman dashboard
Target Yang Diharapkan	Berhasil menampilkan halaman dashboard
Hasil Pengujian	Benar
Path / Jalur	28, 29, 34, 46, 48

3. Kelas order/edit



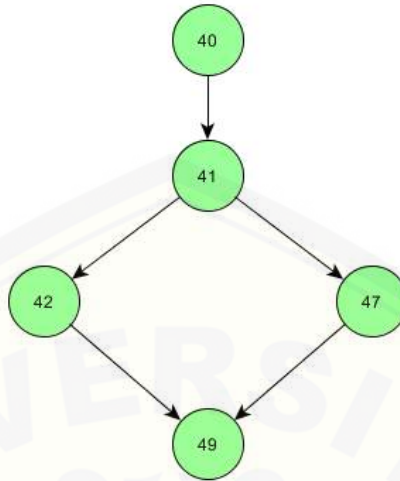
Gambar E. 3 Flow Graph Kelas order/edit

$$CC = E - N + 2 = 5 - 5 + 2 = 2$$

Tabel E. 3 Test Case Kelas order/edit

Test Case 1	Jika pengguna bukan bagian marketing
Target Yang Diharapkan	Pengguna bukan bagian marketing
Hasil Pengujian	Benar
Path / Jalur	43, 44, 51, 53
Test Case 2	Jika pengguna adalah bagian marketing
Target Yang Diharapkan	Pengguna adalah bagian marketing
Hasil Pengujian	Benar
Path / Jalur	43, 44, 45, 53

4. Kelas product/edit



Gambar E. 4 Flow Graph Kelas product/edit

$$CC = E - N + 2 = 5 - 5 + 2 = 2$$

Tabel E. 4 Test Case Kelas product/edit

Test Case 1	Jika pengguna bukan bagian produksi
Target Yang Diharapkan	Pengguna bukan bagian produksi
Hasil Pengujian	Benar
Path / Jalur	40, 41, 47, 49
Test Case 2	Jika pengguna adalah bagian produksi
Target Yang Diharapkan	Pengguna adalah bagian produksi
Hasil Pengujian	Benar
Path / Jalur	40, 41, 47, 49

5. Kelas product/delete



Gambar E. 5 Flow Graph Kelas product/delete

$$CC = E - N + 2 = 2 - 3 + 2 = 1$$

Tabel E. 5 Test Case Kelas product/delete

Test Case 1	Berhasil menghapus data produk
Target Yang Diharapkan	Berhasil menghapus data produk
Hasil Pengujian	Benar
Path / Jalur	59, 60, 62

6. Kelas production/edit



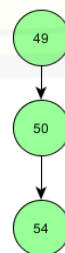
Gambar E. 6 Flow Graph Kelas production/edit

$$CC = E - N + 2 = 2 - 3 + 2 = 1$$

Tabel E. 6 Test Case Kelas production/edit

Test Case 1	Berhasil menampilkan halaman edit
Target Yang Diharapkan	Berhasil menampilkan halaman edit
Hasil Pengujian	Benar
Path / Jalur	42, 43, 47

7. Kelas production/update



Gambar E. 7 Flow Graph Kelas production/update

$$CC = E - N + 2 = 2 - 3 + 2 = 1$$

Tabel E. 7 Test Case Kelas production/update

Test Case 1	Berhasil menghapus data produksi
Target Yang Diharapkan	Berhasil menghapus data produksi
Hasil Pengujian	Benar
Path / Jalur	49, 50, 54

F. Transkrip Hasil Wawancara

Hasil wawancara peneliti dengan Purnama Jati dengan tujuan mendapat informasi dan data untuk penelitian ini.

Pewawancara : Produk apa saja yang diproduksi oleh Purnama Jati?

Narasumber : Prol tape besar, prol tape kecil, pia tape, pia edamame, brownies tape besar, brownies tape kecil, suwar-suwir. Itu adalah produk yang paling sering diproduksi.

Pewawancara : Produk apa saja yang diproduksi setiap hari?

Narasumber : Prol tape besar, prol tape kecil, pia tape brownies tape kecil dan brownies tape besar

Pewawancara : Berapa harga untuk setiap produknya?

Narasumber : Prol tape besar Rp 28,000 , prol tape kecil Rp 19,000 dan pia tape Rp 19,000

Pewawancara : Berapa jumlah produk yang diproduksi setiap harinya?

Narasumber : Rata-rata 200 produk untuk prol tape besar , prol tape kecil 150 dan pia tape dibawah 100 tergantung perkiraan dan permintaan dari outlet.

Pewawancara : Bagaimana sistem penjualan produk Purnama Jati?

Narasumber : Produk Purnama Jati tidak hanya dijual di outlet Purnama Jati sendiri tetapi juga didistribusikan ke beberapa outlet setiap harinya.

Pewawancara : Berapa jumlah yang dikirim ke outlet setiap harinya?

Narasumber : Rata-rata 20 perhari dan jumlah outlet ada sekitar 12 outlet.

Pewawancara : Bagaimana jika ternyata produk habis terjual dalam sehari saat di outlet?

Narasumber : Biasanya pemilik outlet akan menghubungi kami lagi untuk meminta kiriman lagi.

Pewawancara : Menentukan jumlah produk yang diproduksi berdasarkan apa?

Narasumber : Biasanya menggunakan perkiraan. Jika ternyata kurang kami melakukan produksi lagi dan jika berlebih untuk besok produksi dikurangi.

Pewawancara : Apakah ada saat dimana penjualan sangat tinggi atau penjualan sangat rendah?

Narasumber : Ada. Jika penjualan tinggi biasanya saat Hari Raya atau saat liburan. Penjualan rendah biasanya saat tahun ajaran.

