



**SISTEM INFORMASI DISTRIBUSI CABAI DENGAN METODE
DISTRIBUTION REQUIREMENTS PLANNING (DRP)
DI PROVINSI JAWA TIMUR**

SKRIPSI

oleh

Nur Kholis Mansur

122410101019

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS JEMBER**

2019



**SISTEM INFORMASI DISTRIBUSI CABAI DENGAN METODE
DISTRIBUTION REQUIREMENTS PLANNING (DRP)
DI PROVINSI JAWA TIMUR**

SKRIPSI

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

oleh
Nur Kholis Mansur
122410101019

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS JEMBER
2019**

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

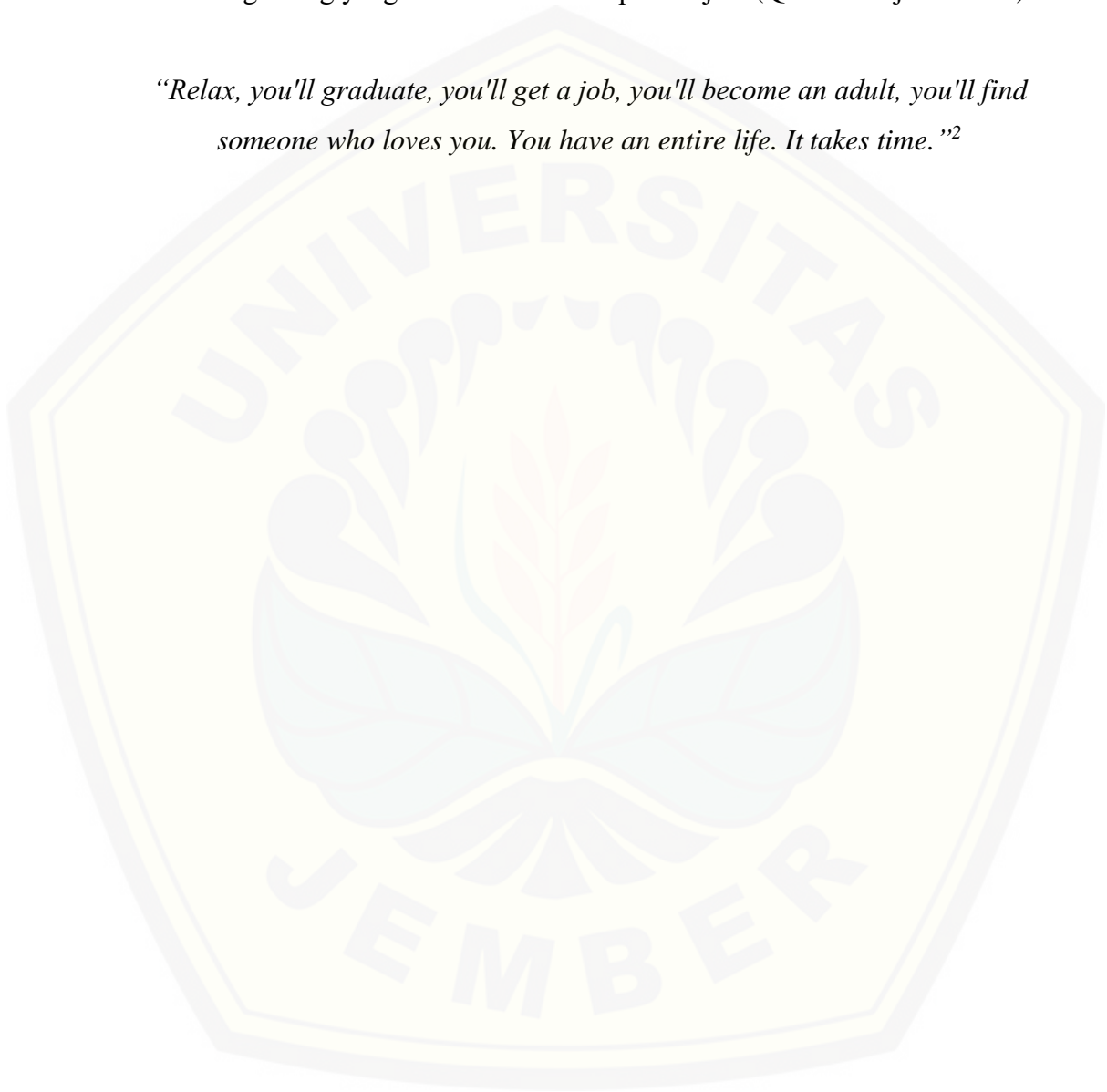
1. Menjadi sarana ibadah kepada Allah subhanahu wa ta'ala, yang telah memberikan kemudahan dan kelancaran dalam mengerjakan skripsi ini;
2. Menunaikan amanah pendidikan dari Ayahanda Hamzah dan Ibunda Tosiya agar ilmu yang didapatkan selama ini bisa bermanfaat untuk orang lain;
3. Kakak Aminullah, Kakak Suhairiyah, Kakak Naufal Rozin, yang senantiasa memberikan dukungan moral untuk menyelesaikan skripsi ini;
4. Guru-guru sejak sekolah dasar, madrasah tsanawiyah, dan sekolah menengah atas yang telah menyalurkan ilmu dan memberikan pelajaran hidup.
5. Dosen-dosen di Fakultas Ilmu Komputer dan Universitas Jember yang telah mengajarkan banyak ilmu.
6. Tim kerja dalam grup FORMATION LULUS yang saling memberikan semangat dan memberikan solusi untuk menyelesaikan skripsi ini;

Semoga keberkahan hidup senantiasa meliputi orang-orang yang telah berjasa memberikan ilmu dan pelajaran hidup untuk saya.

MOTTO

“Niscaya Allah akan mengangkat (derajat) orang-orang yang beriman diantaramu dan orang-orang yang diberi ilmu beberapa derajat” (QS. Al-Mujadalah:11)¹

“Relax, you'll graduate, you'll get a job, you'll become an adult, you'll find someone who loves you. You have an entire life. It takes time.”²



¹ Kementerian Agama Republik Indonesia. 2014. *Al Qur'an dan terjemahannya*. Jakarta: OASIS Terrace Resident.

² Johanna de Silentio

PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : Nur Kholis Mansur

NIM : 122410101019

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Informasi Distribusi Cabai Dengan Metode *Distribution Requirements Planing (DRP)* di Provinsi Jawa Timur”, adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

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

Jember, 10 Juni 2019

Yang menyatakan,

Nur Kholis Mansur

NIM 122410101019

SKRIPSI

**SISTEM INFORMASI DISTRIBUSI CABAI DENGAN METODE
DISTRIBUTION REQUIREMENTS PLANNING (DRP)
DI PROVINSI JAWA TIMUR**

Oleh

Nur Kholis Mansur

122410101019

PEMBIMBING

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

Dosen pembimbing Pendamping : Oktalia Juwita, S. Kom., M.MT

PENGESAHAN PEMBIMBING

Skripsi berjudul "Sistem Informasi Distribusi Cabai Dengan Metode *Distribution Requirements Planing (DRP)* di Provinsi Jawa Timur", telah diuji dan disahkan pada:

hari, tanggal : Selasa, 12 Juli 2019

tempat : Fakultas Ilmu Komputer Informasi Universitas Jember

Disetujui oleh:

Dosen Pembimbing Utama,

Dosen Pembimbing Pendamping,

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

Oktalia Juwita, S. Kom., M.MT.

NIP. 196811131994121001

NIP. 198110202014042001

PENGESAHAN PENGUJI

Skripsi berjudul "Sistem Informasi Distribusi Cabai Dengan Metode *Distribution Requirements Planing (DRP)* di Provinsi Jawa Timur", telah diuji dan disahkan pada:

hari, tanggal : Selasa, 16 Juli 2019

tempat : Fakultas Ilmu Komputer Universitas Jember

Tim Penguji:

Penguji I,

Penguji II,

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

Yudha Alif Auliya, S.Kom., M.Kom.

NIP. 196906151997021002

NIP. 760018031

Mengesahkan

a.n Dekan

Wakil Dekan I Fakultas Ilmu Komputer,

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

NIP. 196909281993021001

RINGKASAN

Sistem Informasi Distribusi Cabai Dengan Metode *Distribution Requirements Planning (DRP)* di Provinsi Jawa Timur; Nur Kholis Mansur, 122410101019; 2019: 203 halaman; Program Studi Sistem Informasi, Fakultas Ilmu Komputer Universitas Jember

Pada tahun 2016 berdasarkan data Badan Pusat Statistika Jawa Timur produksi cabai besar dan cabai rawit hampir tembus 2 juta ton dengan rincian cabai rawit 1 juta ton dan cabai besar 939 ribu ton. Hasil produksi cabai tersebut tersebar di Pulau Jawa, dimana provinsi Jawa Timur menjadi sentra penghasil cabai terbesar di Indonesia. Meskipun kontribusi jumlah produksi yang tinggi, cabai masih menjadi penyebab inflasi di Jawa Timur. Faktor inflasi disebabkan oleh beberapa faktor, salah satunya jumlah produksi cabai melimpah namun terjadi pada bulan tertentu atau penyebab lainnya adalah pendistribusian yang tidak merata.

Mengatasi permasalahan tersebut diperlukan model optimasi distribusi cabai dengan menggunakan beberapa parameter model yakni pendataan jumlah produksi, perencanaan distribusi, dan pendataan nilai akurasi permintaan cabai di wilayah Provinsi Jawa Timur.

Salah satu alternatif teknologi yang digunakan adalah dengan menerapkan sistem informasi menggunakan metode *Distribution Requirements Planning (DRP)*. Metode *DRP* adalah sebuah metode yang digunakan untuk menentukan permintaan dan memenuhi persediaan pada pusat-pusat distribusi, menggabungkan permintaan historis sebagai input untuk sistem. Sistem akan mampu mengelola data cabai, mengelola jumlah permintaan, jumlah produksi, dan yang utama menghitung pola optimal dalam pendistribusian cabai besar. Sistem informasi yang dibangun diharapkan bisa membantu pemerintah dalam mengatur jumlah produksi dan melakukan pendistribusian cabai di Jawa Timur dan petani dalam memantau perkembangan produktivitas dan permintaan cabai di Jawa Timur.

PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul Sistem Informasi Distribusi Cabai Dengan Metode *Distribution Requirements Planing (DRP)* di Provinsi Jawa Timur. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Jember.

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

1. Prof. Dr Saiful Bukhori, ST., M. Kom., selaku Dekan Fakultas Ilmu Komputer Universitas Jember;
2. Prof. Dr Saiful Bukhori, ST., M. Kom., selaku Dosen Pembimbing Utama dan Oktalia Juwita, S. Kom., M.MT., selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Seluruh Bapak dan Ibu dosen beserta staf karyawan program studi sistem informasi di Fakultas Ilmu Komputer Universitas Jember;
4. Ayahanda Hamzah dan Ibunda Tosiya yang senantiasa memberikan doa dan upaya terbaik untuk kesuksesan anak-anaknya, semoga Allah memberikan balasan surga;
5. Seluruh keluarga besar yang telah memberikan dukungan untuk menyelesaikan amanah mengerjakan tugas akhir ini;
6. Bangkesbangpol Provinsi Jawa Timur, Dinas Pertanian Provinsi Jawa Timur dan Dinas Perdagangan dan Perindustrian Provinsi Jawa Timur;
7. Keluarga besar FORMATION angkatan 2012 yang telah menjadi keluarga selama menempuh pendidikan S1;
8. Tim kerja dalam grup FORMATION LULUS yang saling memberikan semangat dan memberikan solusi setiap permasalahan untuk menyelesaikan skripsi ini;
9. Semua pihak yang tidak dapat disebutkan satu-persatu;

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

Jember, 24 April 2019

Penulis



DAFTAR ISI

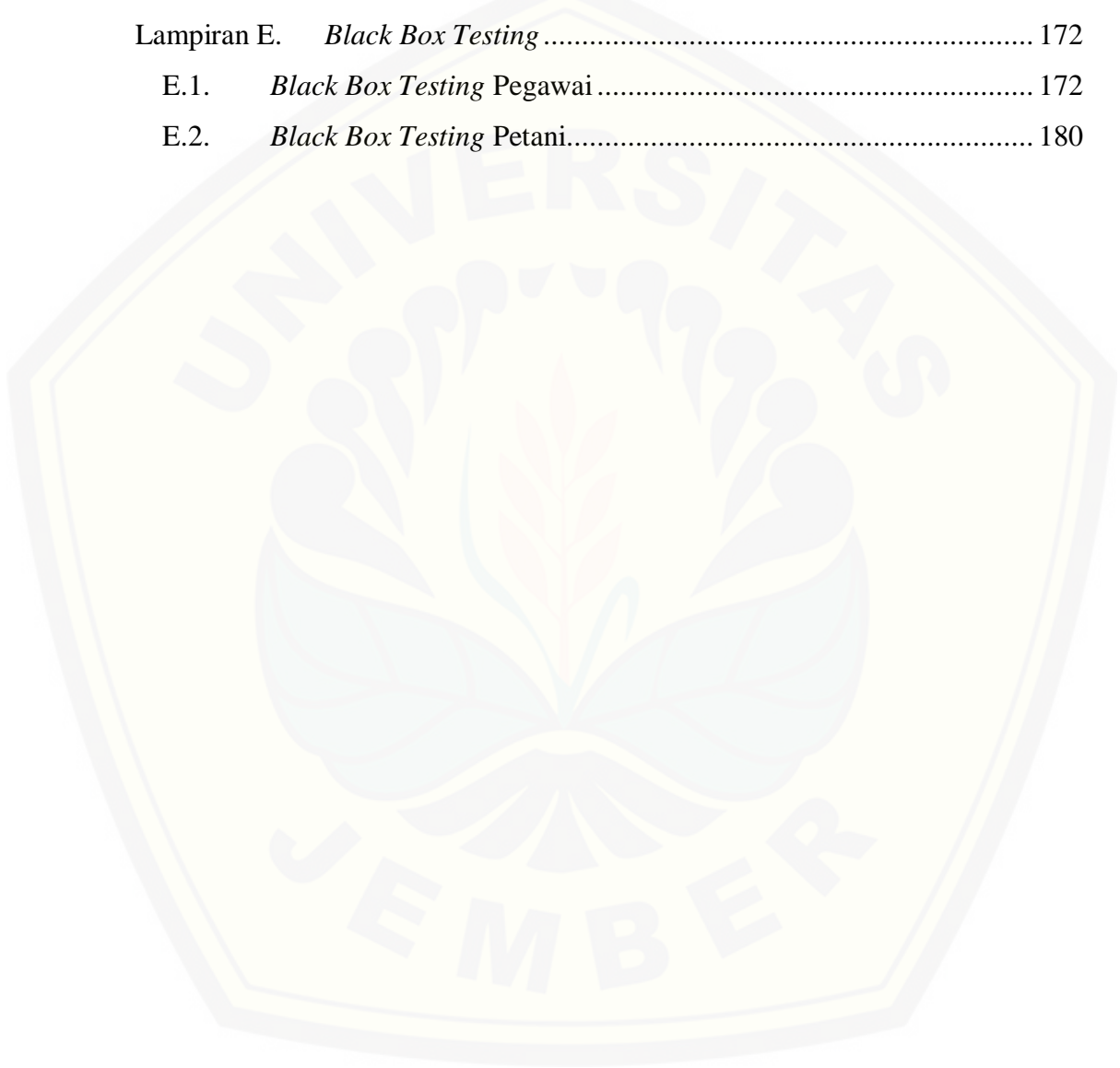
HALAMAN SAMPUL	i
PERSEMBAHAN.....	ii
MOTTO.....	iii
PERNYATAAN	iv
PEMBIMBING.....	v
PENGESAHAN PEMBIMBING	vi
PENGESAHAN PENGUJI.....	vii
RINGKASAN.....	viii
PRAKATA	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xvi
DAFTAR TABEL	xviii
BAB 1. PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	3
1.3. Tujuan	3
1.4. Manfaat	4
1.5. Batasan Masalah.....	4
1.6. Sistematika Penulisan	5
BAB 2. TINJAUAN PUSTAKA.....	6
2.1. Penelitian Terdahulu	6
2.2. Optimalisasi.....	7
2.3. Alur Distribusi Cabai.....	7
2.4. Tanaman Cabai.....	8
2.5. <i>Supply Chain Management</i>	11
2.6. Waktu Tunggu (<i>Lead Time</i>).....	11
2.7. Persediaan Pengaman (<i>Safety Stock</i>)	12
2.8. Ukuran Lot (<i>Lot Sizing</i>)	12
2.9. <i>Distribution Requirements Planning</i> (DRP)	14

2.10.	Metode Perbandingan Eksponensial (MPE).....	17
BAB 3. METODE PENELITIAN		18
3.1.	Tempat Penelitian	18
3.2.	Jenis Penelitian	18
3.3.	Tahapan Penelitian.....	18
3.4.1.	Analisis Kebutuhan (<i>Requirements Definition</i>)	19
3.4.2.	Sistem dan Desain Software (<i>System and Software Design</i>)	22
3.4.3.	Implementasi Sistem (<i>Implementation and Unit Testing</i>)	24
3.4.4.	Pengujian Sistem (<i>Integration and System Testing</i>).....	24
3.4.5.	Pemeliharaan (<i>Operation and Maintenance</i>)	25
BAB 4. PENGEMBANGAN SISTEM.....		26
4.1.	Analisis Kebutuhan.....	26
4.1.1.	Pemilihan Distributor Cabai.....	26
4.1.2.	Data Produksi Cabai Jawa Timur	29
4.1.3.	Data Konsumsi Cabai Jawa Timur	30
4.1.4.	Data Distribusi Cabai Jawa Timur.....	31
4.1.5.	Kebutuhan Fungsional	32
4.1.6.	Kebutuhan Non-fungsional	33
4.2.	Desain Sistem	33
4.2.1.	<i>Business Process</i>	33
4.2.2.	<i>Use case Diagram</i>	34
4.2.3.	<i>Use Case Scenario</i>	38
4.2.4.	<i>Activity Diagram</i>	47
4.2.5.	<i>Sequence Diagram</i>	53
4.2.6.	<i>Class Diagram</i>	60
4.2.7.	<i>Entity Relationship Diagram</i>	61
4.3.	Penulisan Kode Program.....	61
4.3.1.	Kode <i>Login</i>	62
4.3.2.	Kode Pengguna.....	62
4.3.3.	Kode Data Cabai.....	62
4.3.4.	Kode Kab/Kota.....	63

4.3.5.	Kode Produksi Cabai	63
4.3.6.	Kode Konsumsi Cabai	63
4.3.7.	Kode <i>Distribution Requirements Planning</i>	63
4.4.	Pengujian Sistem	64
4.4.1.	<i>Black Box Testing</i> Admin	64
4.4.2.	<i>Black Box Testing</i> Pegawai	72
4.4.3.	<i>Black Box Testing</i> Petani.....	72
BAB 5.	HASIL DAN PEMBAHASAN	73
5.1.	Hasil Pengolahan Data Produksi Cabai	73
5.1.1.	Produksi Cabai Besar.....	73
5.1.2.	Produksi Cabai Rawit	75
5.2.	Hasil Pengolahan Data Konsumsi Cabai	77
5.2.1.	Konsumsi Cabai Besar.....	78
5.2.2.	Konsumsi Cabai Rawit	79
5.3.	Hasil Perhitungan DRP	81
5.3.1.	Waktu Tunggu (<i>Lead Time</i>)	82
5.3.2.	Standar Deviasi.....	84
5.3.3.	Persediaan Pengaman (<i>Safety Stock</i>)	87
5.3.4.	<i>Distribution Requirements Planning</i> (DRP)	94
5.4.	Hasil Pengembangan Sistem	99
5.4.1.	Fitur login.....	99
5.4.2.	Halaman Beranda Pegawai	100
5.4.3.	Fitur Edit Profil.....	101
5.4.4.	Fitur Data Cabai	101
5.4.5.	Fitur Pengguna	102
BAB 6.	PENUTUP	103
6.1.	Kesimpulan.....	103
6.2.	Saran	104
DAFTAR PUSTAKA	105
LAMPIRAN	107
Lampiran A.	<i>Use Case Scenario</i>	107

A.1.	<i>Use Case Scenario</i> Mengelola Data Konsumsi Cabai.....	107
A.2.	<i>Use Case Scenario</i> Mengelola Data Produktivitas Cabai.....	111
A.3.	<i>Use Case Scenario</i> Mengelola Data Pengguna	117
A.4.	<i>Use Case Scenario</i> Mengelola Data Cabai	121
A.5.	<i>Use Case Scenario</i> Mengelola Data Kab/Kota	125
A.6.	<i>Use Case Scenario</i> Melihat Data Produktivitas Cabai	128
A.7.	<i>Use Case Scenario</i> Melihat Data Konsumsi Cabai	130
A.8.	<i>Use Case Scenario</i> Melihat Data Jenis Cabai	132
A.9.	<i>Use Case Scenario</i> Melihat Data Kab/Kota.....	133
Lampiran B.	<i>Activity Diagram</i>	134
B.1.	<i>Activity Diagram</i> Mengelola Data Konsumsi Cabai	134
B.2.	<i>Activity Diagram</i> Mengelola Data Produktivitas Cabai	135
B.3.	<i>Activity Diagram</i> Mengelola Data Pengguna.....	136
B.4.	<i>Activity Diagram</i> Mengelola Data Cabai	137
B.5.	<i>Activity Diagram</i> Mengelola Data Kab/Kota.....	138
B.6.	<i>Activity Diagram</i> Melihat Data Produktivitas Cabai	139
B.7.	<i>Activity Diagram</i> Melihat Data Konsumsi Cabai.....	139
B.8.	<i>Activity Diagram</i> Melihat Data Jenis Cabai.....	140
B.9.	<i>Activity Diagram</i> Melihat Data Kab/Kota	141
Lampiran C.	<i>Sequence Diagram</i>	143
C.1.	<i>Sequence Diagram</i> Mengelola Data Konsumsi Cabai.....	143
C.2.	<i>Sequence Diagram</i> Mengelola Data Produktivitas Cabai.....	144
C.3.	<i>Sequence Diagram</i> Mengelola Data Pengguna	145
C.4.	<i>Sequence Diagram</i> Mengelola Data Cabai	146
C.5.	<i>Sequence Diagram</i> Mengelola Data Kab/Kota	147
C.6.	<i>Sequence Diagram</i> Melihat Data Produktivitas Cabai	148
C.7.	<i>Sequence Diagram</i> Melihat Data Konsumsi Cabai	148
C.8.	<i>Sequence Diagram</i> Melihat Data Jenis Cabai	149
C.9.	<i>Sequence Diagram</i> Melihat Data Kab/Kota.....	149
Lampiran D.	Kode Program	150
D.1.	Kode Login.....	150

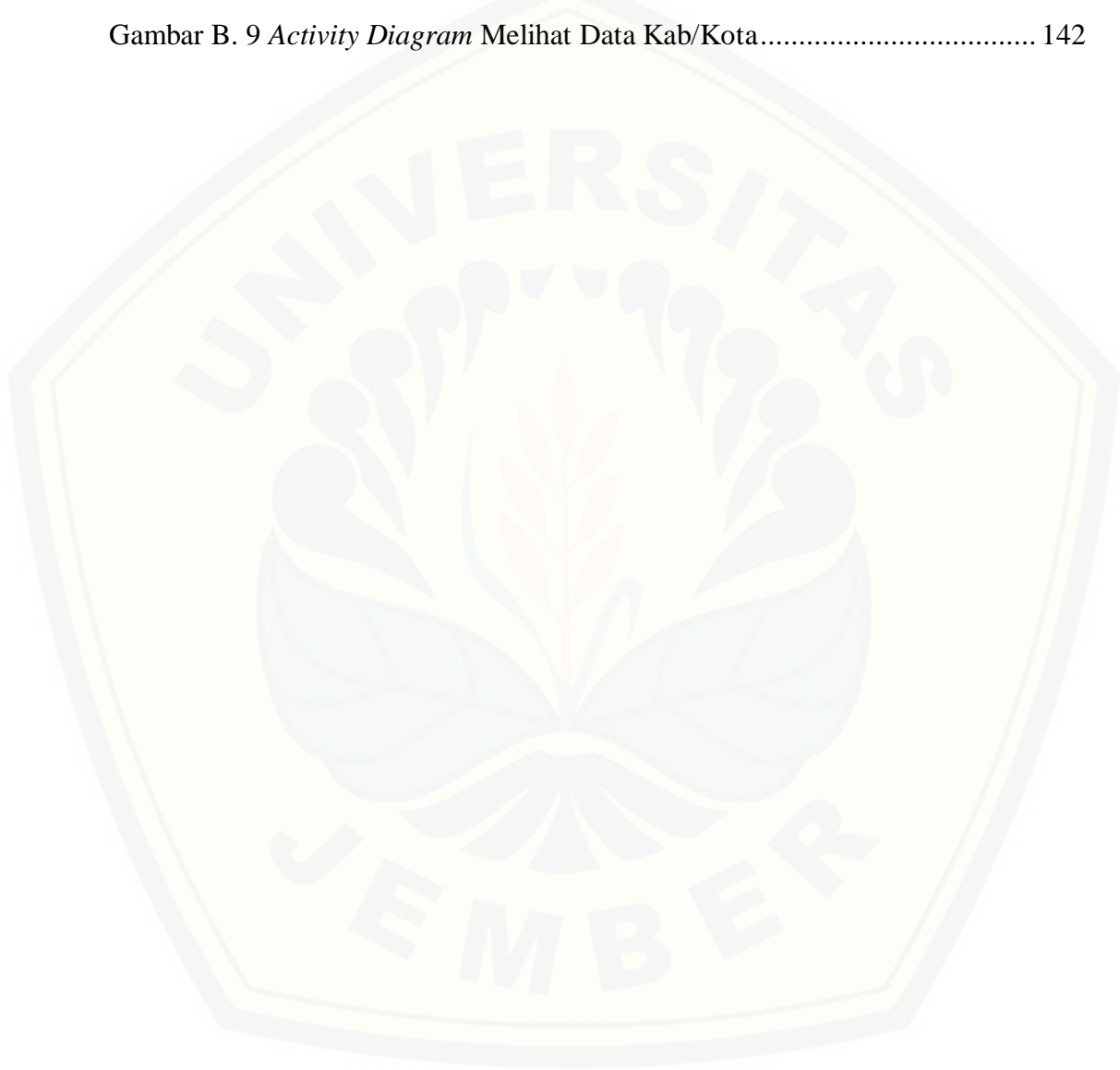
D.2.	Kode Pengguna.....	152
D.3.	Kode Data Cabai.....	158
D.4.	Kode Produksi Cabai	161
D.5.	Kode Konsumsi Cabai	165
D.6.	Kode <i>Distribution Requirements Planning</i>	169
Lampiran E.	<i>Black Box Testing</i>	172
E.1.	<i>Black Box Testing</i> Pegawai	172
E.2.	<i>Black Box Testing</i> Petani.....	180



DAFTAR GAMBAR

Gambar 2. 1. Peta Kebutuhan Cabai Jawa Timur (BPS Jatim, 2015)	7
Gambar 2. 2. Peta Distribusi Perdagangan Cabai di Jatim (BPS Jatim, 2015)	8
Gambar 2. 3 Cabai Besar	9
Gambar 2. 4 Cabai Rawit	10
Gambar 2. 5 Karakteristik <i>Lot Sizing Models</i> (Nadjib & Dauzere, 2017)	13
Gambar 3. 1 <i>Model Waterfall</i> (Pressman, 2001)	19
Gambar 3. 2 <i>Flowchart</i> Penelitian	20
Gambar 4. 1 Statistika Produksi Cabai Besar di Jawa Timur Tahun 2015	28
Gambar 4. 2 Statistika Produksi Cabai Rawit di Jawa Timur Tahun 2015	29
Gambar 4. 3 <i>Business Process</i> Sistem Informasi Distribusi Cabai	33
Gambar 4. 4 <i>Use case Diagram</i>	34
Gambar 4. 5 <i>Activity Diagram</i> Kalkulasi Distribusi Cabai	48
Gambar 4. 6 <i>Sequence Diagram</i> Kalkulasi Distribusi Cabai Jawa Timur	54
Gambar 4. 7 <i>Class Diagram</i> Sistem Informasi Distribusi Cabai	60
Gambar 4. 8 <i>Entity Relationship Diagram</i> Sistem Informasi Distribusi Cabai	61
Gambar 5. 1 Fitur Produksi Cabai Besar	74
Gambar 5. 2 Fitur Produksi Cabai Rawit	76
Gambar 5. 3 Fitur Konsumsi Cabai Besar	78
Gambar 5. 4 Fitur Konsumsi Cabai Rawit	80
Gambar 5. 5 Fitur Kalkulasi Distribusi	81
Gambar 5. 6 Tabulasi Kalkulasi Metode DRP	82
Gambar 5. 7 Fitur Login Admin	100
Gambar 5. 8 Halaman Beranda Pegawai	100
Gambar 5. 9 Fitur Edit Profil	101
Gambar 5. 10 Fitur Data Cabai	101
Gambar 5. 11 Fitur Pengguna	102
Gambar B. 1 <i>Activity Diagram</i> Mengelola Data Konsumsi Cabai	134
Gambar B. 2 <i>Activity Diagram</i> Mengelola Data Produktivitas Cabai	135
Gambar B. 3 <i>Activity Diagram</i> Mengelola Data Pengguna	136

Gambar B. 4 <i>Activity Diagram</i> Mengelola Data Cabai	137
Gambar B. 5 <i>Activity Diagram</i> Mengelola Data Kab/Kota	138
Gambar B. 6 <i>Activity Diagram</i> Melihat Data Produktivitas Cabai	139
Gambar B. 7 <i>Activity Diagram</i> Melihat Data Konsumsi Cabai	140
Gambar B. 8 <i>Activity Diagram</i> Melihat Data Jenis Cabai	141
Gambar B. 9 <i>Activity Diagram</i> Melihat Data Kab/Kota.....	142

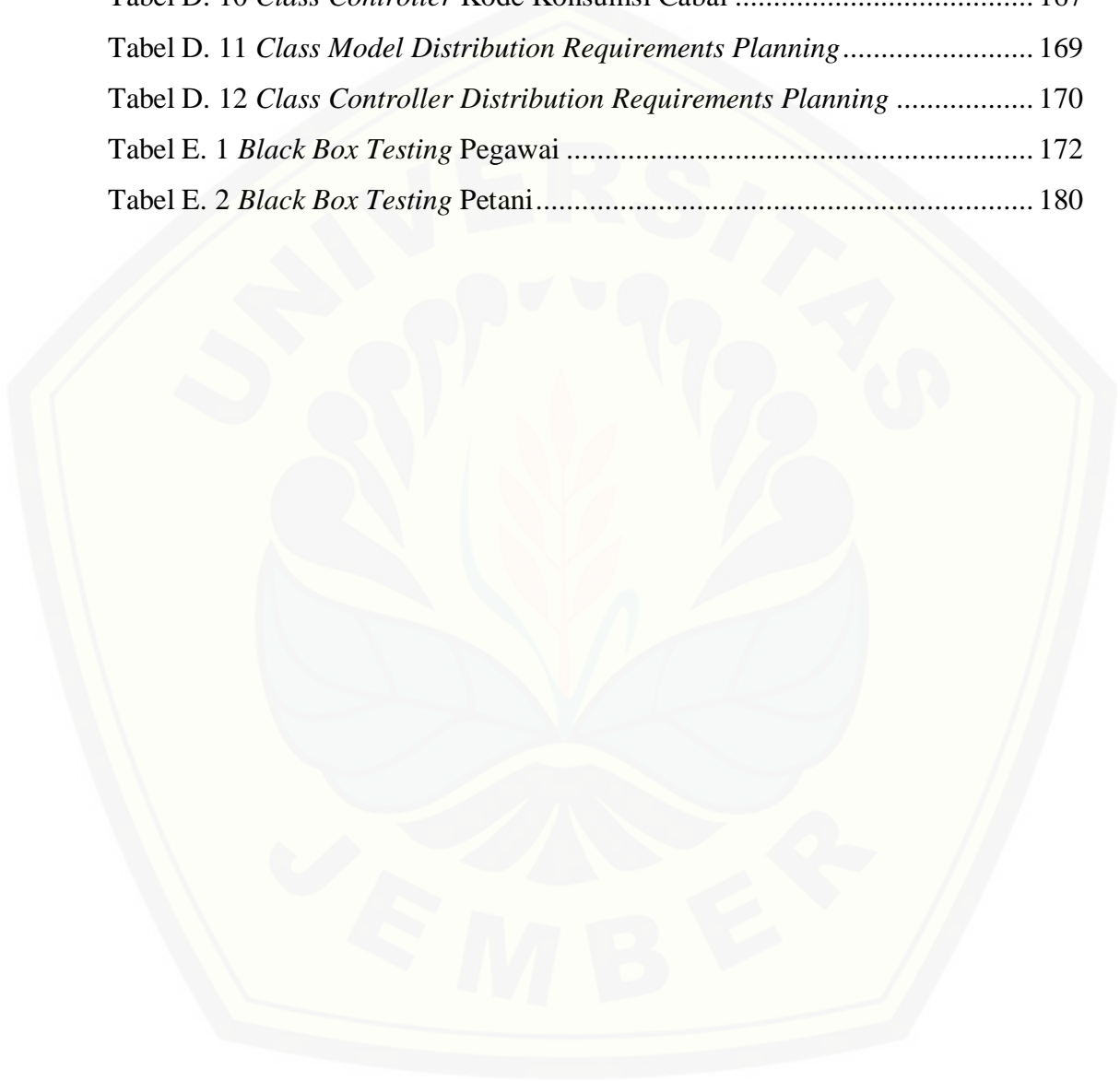


DAFTAR TABEL

Tabel 2. 1 Tabel <i>Distribution Requirement Planning</i>	16
Tabel 4. 1 Distributor Cabai Besar Terpilih di Jawa Timur Tahun 2015	27
Tabel 4. 2 Distributor Cabai Rawit Terpilih di Jawa Timur Tahun 2015	28
Tabel 4. 3 Data Konsumsi Cabai Besar dengan Data Produksi Cabai Besar	29
Tabel 4. 4 Data Konsumsi Cabai Rawit dengan Data Produksi Cabai Rawit	30
Tabel 4. 5 Jumlah Konsumsi Cabai Besar di Jawa Timur	30
Tabel 4. 6 Jumlah Konsumsi Cabai Rawit di Jawa Timur	31
Tabel 4. 7 Data Distribusi Cabai Jawa Timur	31
Tabel 4. 8 Definisi Aktor	35
Tabel 4. 9 Definisi <i>Use case</i>	36
Tabel 4. 10 <i>Use Case Scenario</i> Kalkulasi Distribusi Cabai.....	38
Tabel 4. 11 <i>Black Box Testing</i> Admin	64
Tabel 5. 1 Hasil Pengolahan Data Produksi Cabai Besar	74
Tabel 5. 2 Hasil Pengolahan Data Produksi Cabai Rawit.....	76
Tabel 5. 3 Hasil Pengolahan Data Konsumsi Cabai Besar	79
Tabel 5. 4 Hasil Pengolahan Data Konsumsi Cabai Rawit	80
Tabel 5. 5 <i>Lead Time</i> Cabai Besar.....	83
Tabel 5. 6 <i>Lead Time</i> Cabai Rawit	83
Tabel 5. 7 Perhitungan Standar Deviasi Cabai Besar	85
Tabel 5. 8 Perhitungan Standar Deviasi Cabai Rawit.....	86
Tabel 5. 9 Perhitungan <i>Safety Stock</i> Cabai Besar di Bojonegoro.....	88
Tabel 5. 10 Perhitungan <i>Safety Stock</i> Cabai Besar di Tuban	89
Tabel 5. 11 Perhitungan <i>Safety Stock</i> Cabai Besar di Blitar	89
Tabel 5. 12 Perhitungan <i>Safety Stock</i> Cabai Besar di Malang	89
Tabel 5. 13 Perhitungan <i>Safety Stock</i> Cabai Besar di Kediri	90
Tabel 5. 14 Perhitungan <i>Safety Stock</i> Cabai Besar di Banyuwangi.....	90
Tabel 5. 15 Hasil Perhitungan <i>Safety Stok</i> Cabai Besar.....	91
Tabel 5. 16 Perhitungan <i>Safety Stock</i> Cabai Rawit di Blitar	91
Tabel 5. 17 Perhitungan <i>Safety Stock</i> Cabai Rawit di Kediri.....	92

Tabel 5. 18 Perhitungan <i>Safety Stock</i> Cabai Rawit di Sampang	92
Tabel 5. 19 Perhitungan <i>Safety Stock</i> Cabai Rawit di Jember.....	93
Tabel 5. 20 Perhitungan <i>Safety Stock</i> Cabai Besar di Malang	93
Tabel 5. 21 Perhitungan <i>Safety Stock</i> Cabai Besar di Lumajang	93
Tabel 5. 22 Hasil Perhitungan <i>Safety Stok</i> Cabai Rawit	94
Tabel 5. 23 Perhitungan DRP Cabai Besar Bojonegoro	95
Tabel 5. 24 Perhitungan DRP Cabai Besar Tuban	95
Tabel 5. 25 Perhitungan DRP Cabai Besar Blitar	95
Tabel 5. 26 Perhitungan DRP Cabai Besar Malang	96
Tabel 5. 27 Perhitungan DRP Cabai Besar Kediri	96
Tabel 5. 28 Perhitungan DRP Cabai Besar Banyuwangi.....	96
Tabel 5. 29 Perhitungan DRP Cabai Rawit Blitar	97
Tabel 5. 30 Perhitungan DRP Cabai Rawit Kediri	97
Tabel 5. 31 Perhitungan DRP Cabai Rawit Sampang	98
Tabel 5. 32 Perhitungan DRP Cabai Rawit Jember.....	98
Tabel 5. 33 Perhitungan DRP Cabai Rawit Malang	98
Tabel 5. 34 Perhitungan DRP Cabai Rawit Lumajang	99
Tabel A. 1 <i>Use Case Scenario</i> Mengelola Data Konsumsi Cabai.....	107
Tabel A. 2 <i>Use Case Scenario</i> Mengelola Data Produktivitas Cabai.....	111
Tabel A. 3 <i>Use Case Scenario</i> Mengelola Data Pengguna	117
Tabel A. 4 <i>Use Case Scenario</i> Mengelola Data Cabai	121
Tabel A. 5 <i>Use Case Scenario</i> Mengelola Data Kab/Kota	125
Tabel A. 6 <i>Use Case Scenario</i> Melihat Data Produktivitas Cabai	128
Tabel A. 7 <i>Use Case Scenario</i> Melihat Data Konsumsi Cabai	130
Tabel A. 8 <i>Use Case Scenario</i> Melihat Data Jenis Cabai	132
Tabel A. 9 <i>Use Case Scenario</i> Melihat Data Kab/Kota.....	133
Tabel D. 1 <i>Class Model</i> Kode Login	150
Tabel D. 2 <i>Class Controller</i> Kode Login.....	151
Tabel D. 3 <i>Class Model</i> Kode Pengguna	152
Tabel D. 4 <i>Class Controller</i> Kode Pengguna.....	155
Tabel D. 5 <i>Class Model</i> Kode Data Cabai	158

Tabel D. 6 <i>Class Controller</i> Kode Data Cabai.....	159
Tabel D. 7 <i>Class Model</i> Kode Produksi Cabai.....	161
Tabel D. 8 <i>Class Controller</i> Kode Produksi Cabai	163
Tabel D. 9 <i>Class Model</i> Kode Konsumsi Cabai.....	165
Tabel D. 10 <i>Class Controller</i> Kode Konsumsi Cabai	167
Tabel D. 11 <i>Class Model Distribution Requirements Planning</i>	169
Tabel D. 12 <i>Class Controller Distribution Requirements Planning</i>	170
Tabel E. 1 <i>Black Box Testing</i> Pegawai	172
Tabel E. 2 <i>Black Box Testing</i> Petani.....	180



BAB 1. PENDAHULUAN

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

1.1. Latar Belakang

Cabai (*Capsicum Annuum L.*) merupakan salah satu komoditas sayuran yang memiliki nilai ekonomis penting dan tingkat konsumsi tinggi di Indonesia. Jumlah konsumsi cabai dan hasil produksi dari tahun 2013 hingga 2016 terus meningkat. Pada tahun 2016 berdasarkan data Badan Pusat Statistika Jawa Timur produksi cabai besar dan cabai rawit hampir tembus 2 juta ton dengan rincian cabai rawit 1 juta ton dan cabai besar 939 ribu ton. Hasil produksi cabai tersebut tersebar di Pulau Jawa, dimana provinsi Jawa Timur menjadi sentra penghasil cabai terbesar di Indonesia.

Jawa Timur menjadi produksi cabai rawit peringkat pertama selama tahun 2013-2015. Pada tahun 2013 kontribusinya sebesar 31,88%, sedangkan pada tahun 2015 mengalami penurunan menjadi 28,73%. Meskipun kontribusi jumlah produksi yang tinggi, cabai masih menjadi penyebab inflasi di Jawa Timur. Faktor inflasi disebabkan oleh beberapa faktor, salah satunya jumlah produksi cabai melimpah namun terjadi pada bulan tertentu atau penyebab lainnya adalah pendistribusian yang tidak merata. Beberapa kota di Jawa Timur mengalami inflasi yang disebabkan oleh cabai pada bulan februari 2015 yaitu kabupaten Banyuwangi sebesar 0,0211; kota Kediri sebesar 0,0152; kabupaten Sumenep sebesar 0,08; kota Malang sebesar 0,0967; kota Probolinggo sebesar 0,0226. Secara keseluruhan akibat kenaikan harga cabai rawit menyumbang kenaikan inflasi di Jawa Timur sebesar 0,0234 (BPS Jatim, 2015). Oleh karena itu pendistribusian dan kontrol fluktuasi harga cabai harus mendapat perhatian khusus dari pemerintah pusat maupun daerah.

Permasalahan tersebut membutuhkan solusi yang efektif dan optimal, agar jumlah produksi cabai tiap daerah bisa didistribusikan dan jumlah permintaan bisa terpenuhi dengan baik. Untuk mengatasinya diperlukan model optimasi distribusi cabai dengan menggunakan beberapa parameter model yakni pendataan jumlah produksi, perencanaan distribusi, dan pendataan nilai akurasi permintaan cabai di wilayah Provinsi Jawa Timur.

Optimalisasi distribusi produk cabai akan lebih efisien dan terkoordinasi lebih efektif menggunakan metode *Distribution Requirements Planning* (DRP). DRP adalah sebuah metode yang digunakan untuk menentukan permintaan dan memenuhi persediaan pada pusat-pusat distribusi, menggabungkan permintaan historis sebagai input untuk sistem produksi dan material (Bowersox, Closs, & Cooper, 2012). DRP mampu merumuskan kemungkinan permintaan berkaitan dengan kebutuhan untuk pengiriman dari titik produksi menuju tujuan akhir pendistribusian. Hal ini akan membantu pengambilan keputusan untuk melakukan tindakan-tindakan korektif sebelum kejadian-kejadian yang tidak diinginkan berkembang menjadi krisis.

Penelitian sebelumnya metode DRP ini digunakan untuk menghasilkan solusi dengan melakukan optimasi persediaan produk dan permintaan. Distribusi produk menggunakan metode DRP untuk permintaan yang akan datang akan bekerja lebih efisien. Penggunaan DRP menjalankan koordinasi dengan yang baik secara berkala agar pengiriman produk untuk tiap-tiap tujuan berjalan lebih efektif dan optimal dalam memenuhi permintaan (Abdillah, 2009). Metode DRP menjadi salah satu cara untuk membantu merencanakan aktivitas distribusi dan penjadwalan dimana dengan metode ini dapat menetapkan kebutuhan alokasi persediaan dan memastikan bahwa pemenuhan sumber akan dapat memenuhi permintaan. DRP juga memiliki fungsi objektif dalam pengelolaan stok dan operasi pemenuhan permintaan.

Penelitian ini bertujuan untuk melakukan analisis perencanaan dan penjadwalan yang optimal sesuai jumlah kebutuhan cabai pada wilayah terpilih di Jawa Timur. Analisis dilakukan menggunakan metode DRP, dimana metode tersebut akan diterapkan dalam sistem informasi berbasis *website* yang bisa

menampilkan hasil analisa perhitungan distribusi cabai. Sistem yang dibangun akan mampu mengelola data cabai, mengelola jumlah permintaan, jumlah produksi, dan menghasilkan analisa optimal distribusi cabai. Sistem informasi yang dibangun diharapkan bisa membantu pemerintah melakukan analisa jumlah produksi dan perhitungan data kebutuhan konsumsi cabai di Jawa Timur. Sistem juga diharapkan bisa membantu petani untuk memantau perkembangan produktivitas dan konsumsi cabai di Jawa Timur.

1.2. Rumusan Masalah

Berdasarkan masalah yang telah diuraikan diatas, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana menganalisa pendistribusian cabai di Provinsi Jawa Timur menggunakan metode *Distribution Requirements Planing*?
2. Bagaimana cara menguji efektifitas metode *Distribution Requirements Planing* diterapkan dalam analisa pendistribusian cabai di Jawa Timur?
3. Bagaimana mengimplementasikan sistem distribusi cabai di Provinsi Jawa Timur menggunakan metode *Distribution Requirements Planing*?

1.3. Tujuan

Tujuan dari penelitian yang dilakukan adalah sebagai berikut:

1. Menganalisa pendistribusian cabai di Provinsi Jawa Timur dengan metode *Distribution Requirements Planing*.
2. Menguji efektifitas penerapan metode *Distribution Requirements Planing* untuk analisa pendistribusian cabai.
3. Merancang dan membangun sistem informasi distribusi *berbasis web* untuk membantu pemerintah Jawa Timur melakukan pemantauan secara digital terkait analisa pendistribusian cabai menggunakan metode *Distribution Requirements Planing*.

1.4. Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Diharapkan dapat membantu pihak pemerintah dalam menentukan kebutuhan cabai pada periode yang telah ditentukan dan petani mendapatkan informasi jumlah produktivitas dan jumlah permintaan cabai pada tahun tertentu.
2. Dapat mengetahui bagaimana proses penerapan metode *Distribution Requirement Planning* pada pendistribusian cabai di Jawa Timur.
3. Diharapkan dapat menjadi dasar bagi peneliti pada penelitian selanjutnya untuk pengembangan sistem informasi prediksi yang lebih baik.

1.5. Batasan Masalah

Pembahasan yang dilakukan dalam skripsi ini memiliki batasan masalah sebagai berikut:

1. Objek dari penelitian ini adalah jenis cabai besar dan cabai rawit.
2. Wilayah yang digunakan dalam analisa data maksimal menggunakan 6 kota karena adanya keterbatasan dalam mengakses data.
3. Optimalisasi distribusi hanya terbatas pada hasil analisa kebutuhan dan analisa rencana pendistribusian cabai dalam bentuk rekomendasi jumlah produksi dan jumlah kebutuhan.
4. Distribusi tidak mencakup pada proses bisnis dari titik produksi menuju konsumen akhir.
5. Menggunakan data *history* jumlah produksi dan jumlah konsumsi cabai besar dan cabai rawit selama 5 tahun.
6. Data 4 tahun dijadikan sebagai data *test* dan data 1 tahun dijadikan sebagai data training untuk analisa pendistribusian cabai.
7. Data *history* yang digunakan sebagai data *training* terbatas pada data cabai besar dan cabai rawit tahun 2015.
8. Aplikasi yang dibangun adalah sistem informasi berbasis *website*.

1.6. Sistematika Penulisan

Sistematika penulisan dan urutan skripsi ini disusun sebagai berikut:

1. Pendahuluan

Bab ini menjelaskan tentang latar belakang, rumusan masalah, tujuan dan manfaat, ruang lingkup studi dan sistematika penulisan.

2. Tinjauan Pustaka

Bab ini menjelaskan tentang materi dari hasil penelitian terdahulu, informasi alur distribusi cabai, dan teori penggunaan metode DRP yang menjadi kerangka pemikiran dalam penelitian.

3. Metodologi Penelitian

Bab ini menjelaskan tentang metode penelitian yang digunakan meliputi desain penelitian, jenis penelitian, tempat penelitian, dan tahapan penelitian yang digunakan untuk mengembangkan sistem.

4. Pengembangan Sistem

Bab ini menjelaskan tentang gambaran umum pengembangan sistem, terkait penerapan hasil data analisa kebutuhan cabai yang mencakup data distributor cabai terpilih hingga penentuan kebutuhan fungsional, desain sistem, penulisan kode program, dan pengujian.

5. Hasil dan Pembahasan

Bab ini menjelaskan tentang hasil dan pembahasan terkait perhitungan manual yang dibutuhkan untuk melengkapi kebutuhan data metode DRP dan perhitungan yang diterapkan dalam sistem informasi.

6. Penutup

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

BAB 2. TINJAUAN PUSTAKA

Bab ini dipaparkan teori-teori dan pustaka yang akan dipakai dalam penelitian. Teori-teori ini berupa teori dari buku literatur dan jurnal. Berikut merupakan teori-teori yang dibahas dalam penelitian.

2.1. Penelitian Terdahulu

Penelitian yang dilakukan oleh Hidayat, dkk (2016) menggunakan metode *DRP* menghasilkan kesimpulan berupa kapan dan berapa jumlah persediaan yang *over stock*. Analisa pengaruh terhadap penghematan biaya jika persediaan yang berlebih dijual dengan harga lebih murah (*cost saving*), harga jual minimum (*minimum salvage value*) serta solusi mengenai keputusan apakah yang harus diambil.

Penelitian ini dilakukan berdasarkan teori dan studi yang pernah dilakukan Abdillah (2009) menyebutkan Distribusi produk menggunakan metode *DRP* untuk permintaan yang akan datang akan lebih efisien karena terkoordinasi dengan baik secara berkala untuk tiap-tiap tujuan dan lebih efektif karena pengiriman produk yang optimal dalam memenuhi permintaan masing-masing *warehouse* sesuai kapasitas kontainer yang digunakan.

Penelitian yang dilakukan oleh Bowersox, Closs, & Cooper (2012) menyebutkan bahwa optimalisasi distribusi produk cabai akan lebih efisien dan terkoordinasi lebih efektif menggunakan metode *Distribution Requirements Planning* (*DRP*). *DRP* adalah sebuah metode yang digunakan untuk menentukan permintaan dan memenuhi persediaan pada pusat-pusat distribusi, menggabungkan permintaan historis sebagai input untuk sistem produksi dan material.

Penelitian ini bertujuan untuk melakukan analisis perencanaan dan penjadwalan yang optimal sesuai jumlah kebutuhan cabai pada wilayah terpilih di Jawa Timur. Analisis dilakukan menggunakan *DRP*, dimana metode tersebut akan diterapkan dalam sistem informasi berbasis *website* yang bisa menampilkan hasil analisa perhitungan distribusi cabai. Sistem yang dibangun akan mampu mengelola

data cabai, mengelola jumlah permintaan, jumlah produksi, dan menghasilkan analisa optimal distribusi cabai. Sistem informasi yang dibangun diharapkan bisa membantu pemerintah melakukan analisa jumlah produksi dan perhitungan data kebutuhan konsumsi cabai di Jawa Timur.

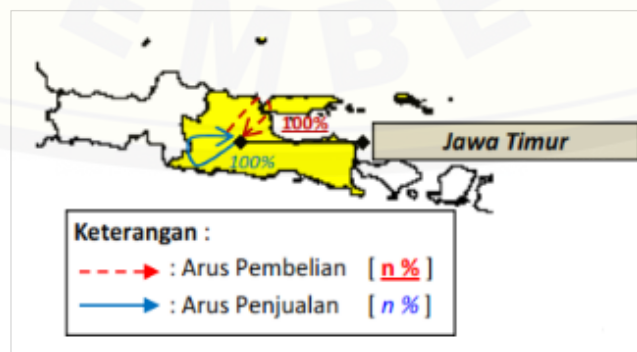
2.2. Optimalisasi

Optimasi berasal dari kata dasar optimal yang berarti terbaik, tertinggi, paling menguntungkan, menjadikan paling baik, dan perbuatan mengoptimalkan (menjadikan paling baik, paling tinggi, dan sebagainya). Sedangkan secara istilah optimasi adalah proses untuk mencapai hasil ideal atau optimal (nilai efektif yang dapat dicapai). (Departemen Pendidikan Indonesia, 2008)

Optimasi distribusi cabai merupakan sebuah proses untuk mengoptimalkan pengiriman produksi cabai ke seluruh area yang menjadi cakupan distribusi di Jawa Timur. Optimasi distribusi cabai juga dapat diartikan sebagai suatu bentuk mengoptimalkan perencanaan distribusi dan menjadikannya lebih efektif.

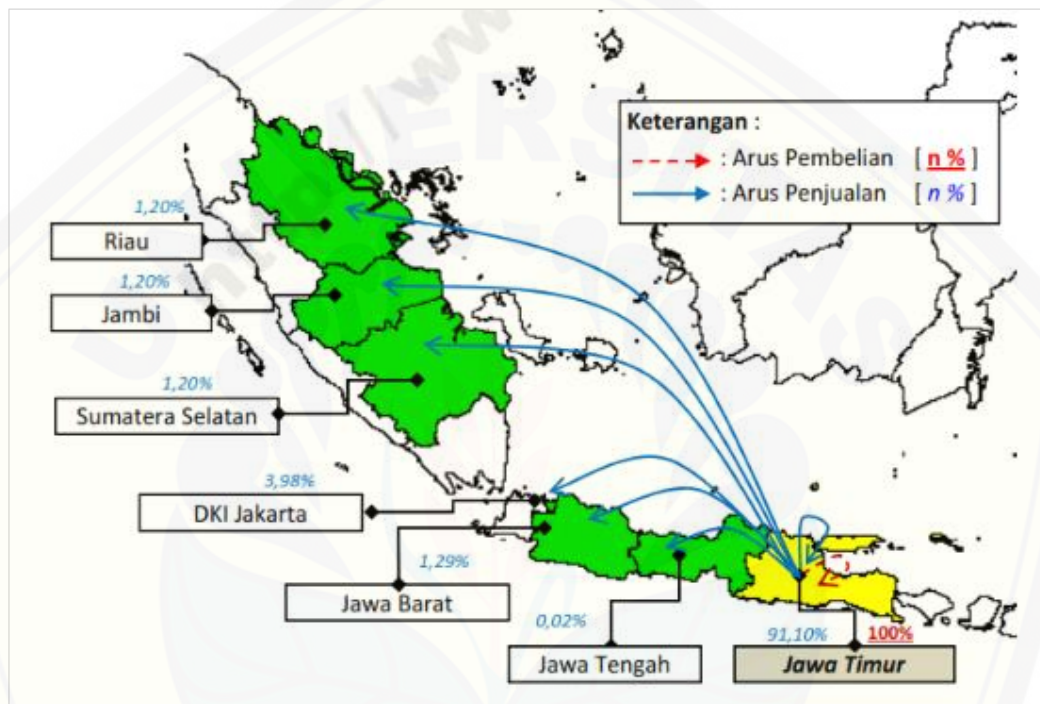
2.3. Alur Distribusi Cabai

Alur distribusi tanaman cabai di Jawa Timur berdasarkan penjualan produksi cabai besar, didapatkan bahwa produsen cabai di Provinsi Jawa Timur mendapatkan bahan baku seluruhnya dari wilayahnya sendiri (100,00%). Secara umum hasil bahan baku yang dihasilkan digunakan untuk memenuhi kebutuhan wilayahnya sendiri. Peta kebutuhan bahan baku cabai dapat dilihat pada gambar 2.1.



Gambar 2. 1. Peta Kebutuhan Cabai Jawa Timur (BPS Jatim, 2015)

Menurut wilayah penjualannya, produsen cabai di Provinsi Jawa Timur pun hanya menjual cabai ke wilayahnya sendiri 91,10%. Sedangkan sebagian lainnya penjualan cabai keluar Provinsi Jawa Timur yaitu Riau 1,20%, Jambi (1,20%), Sumatera Selatan (1,20%), DKI Jakarta 3,98%, Jawa Barat 1,29%, dan Jawa Tengah (0,02%). Alur distribusi cabai Jawa Timur saat memenuhi kebutuhan kota lain dapat dilihat pada gambar 2.2.



Gambar 2. 2. Peta Distribusi Perdagangan Cabai di Jatim (BPS Jatim, 2015)

2.4. Tanaman Cabai

Tanaman cabai (*Capsicum annum L*) merupakan tanaman perdu yang berasal dari Benua Amerika, tepatnya daerah Peru yang kemudian terus menyebar ke seluruh Amerika, Eropa, dan Asia termasuk Indonesia. Pembuktian yang dilakukan ilmuwan memberikan hasil bahwa budidaya cabai pertama kali ditemukan dalam penelusuran fakta sejarah Peru dan terdapat sisa-sisa biji yang telah berumur lebih dari 5000 tahun SM didalam gua di Tehuacan, Meksiko. Penyebaran cabai ke seluruh dunia termasuk negara-negara di Asia, seperti Indonesia dilakukan oleh pedagang Spanyol dan Portugis (Dermawan, 2010).

Dalam sistem klasifikasi, tanaman cabai dapat digolongkan sebagai berikut:

- Kingdom : Plantae
- Divisi : Magnoliophyta
- Famili : Solanaceae
- Spesies : *Capsicum annum*, *Capsicum frutescens*, *Capsicum chinense*

Dalam penelitian ini cabai besar (*Capsicum annum*) dan cabai rawit (*Capsicum frutescens*) yang menjadi objek penelitian.

1. Cabai Besar

Menurut Setiadi (2012) cabai besar (*Capsicum annum*) atau lombok besar dihasilkan dari proses pemuliaan (*varietas*). Tanaman cabai besar yang matang dapat dilihat pada gambar 2.3.



Gambar 2. 3 Cabai Besar

Cabai besar yang merupakan varietas hasil budidaya memiliki ciri umum sebagai berikut:

- a. Batang yang tegak dengan ketinggian 50 cm-100 cm.
- b. Tangkai daun horizontal atau miring dengan panjang 1,5 - 4,5 cm.
- c. Posisi bunga menggantung memiliki mahkota berwarna putih.
- d. Panjang tangkai bunga cabai dari 1 cm-2 cm.
- e. Buahnya memanjang dengan biji buahnya berwarna putih kekuning-kuningan (krem) atau kuning kecokelat-cokelatan.
- f. Biasanya digunakan untuk pewarna masakan dan rasanya tidak pedas.

2. Cabai Rawit

Cabai rawit (*Capsicum frutescens*) memiliki ciri yang berbeda dengan ukuran daun lebih kecil dibandingkan dengan daun tanaman cabai besar. Tanaman cabai rawit yang dapat dilihat pada gambar 2.4.



Gambar 2. 4 Cabai Rawit

Menurut Cahyono (2003) secara morfologi bagian-bagian atau organ-organ penting dari tanaman cabai rawit dapat dideskripsikan sebagai berikut:

- a. Batang tanaman memiliki struktur keras dan berkayu, berwarna gelap hijau, berbentuk bulat, halus, dan bercabang banyak.
- b. Cabang tanaman beruas-ruas sehingga ruas ditumbuhi daun dan tunas (cabang).
- c. Daun berbentuk bulat telur dengan ujung runcing dan tepi daun rata (tidak bergerigi/berlekuk).
- d. Ukuran daun cabai rawit lebih kecil dibandingkan dengan daun tanaman cabai besar.
- e. Bunga tunggal yang berbentuk bintang yang tumbuh menunduk pada ketiak daun.
- f. Buah berbentuk bulat, pendek dengan ujung runcing atau berbentuk kerucut.
- g. Pada waktu masih muda, rasa buah cabai rawit kurang pedas tetapi setelah masak menjadi pedas

2.5. *Supply Chain Management*

Supply chain management adalah suatu pendekatan yang digunakan untuk mencapai pengintegrasian yang efisien dari *supplier*, produsen, distributor, *retailer* dan *customer*. Artinya barang diproduksi dalam jumlah yang tepat, pada saat yang tepat dan pada tempat yang tepat dengan tujuan untuk meminimalkan biaya sistem secara keseluruhan dan memenuhi persyaratan tingkat layanan. (Simchi-Levi & Kaminsky, 2003)

Keunggulan kompetitif dari *supply chain management* mencakup cara mengatur aliran barang atau produk dalam suatu rantai pasok, dengan kata lain model *supply chain management* mengaplikasikan bagaimana suatu jaringan kegiatan produksi dan distribusi dari satu bagian/divisi sehingga menghasilkan produk sesuai dengan tuntutan konsumen.

Seluruh elemen dalam *supply chain management* tidak bisa berjalan secara terpisah, tetapi merupakan satu kesatuan. Sehingga akan menghasilkan sinergi dan pada akhirnya menciptakan efisiensi dan efektifitas. Koordinasi dalam *supply chain* diperlukan untuk memantapkan dan mempertahankan kontinuitas operasi. Koordinasi dapat dibagi menjadi 4 bagian manajerial, yaitu: peramalan pasar produk, pengolahan pesanan, perencanaan operasi dan *procurement* (perencanaan kebutuhan material).

2.6. Waktu Tunggu (*Lead Time*)

Lead time adalah tenggang waktu yang diperlukan (yang terjadi) antara saat pemesanan barang dengan datangnya barang itu sendiri. Waktu tunggu sangat erat hubungannya dengan perencanaan pemesanan barang yang dilakukan selanjutnya. Penggunaan waktu tunggu yang tepat akan memberikan akurasi data untuk melakukan pembelian pada saat yang tepat pula, sehingga resiko penumpukan persediaan atau kekurangan persediaan dapat ditekan seminimal mungkin (Juliani & Ridwan, 2015).

Lead time merupakan waktu yang dibutuhkan sejak dilakukannya pemesanan hingga waktu diterimanya pesanan tersebut di toko, gudang penyimpanan, atau

cabang distribusi. Untuk bagian logistik, *lead time* dimulai pada saat ditentukannya kebutuhan suatu produk hingga pada saat dapat diambilnya inventori yang tersedia untuk memenuhi kebutuhan konsumen.

2.7. Persediaan Pengaman (*Safety Stock*)

Menurut Taylor & Bernard (2008) persediaan cadangan adalah persediaan yang disimpan untuk mengantisipasi permintaan pelanggan yang sulit diketahui dengan pasti. Stok cadangan ini disimpan untuk memenuhi permintaan musiman atau siklus. Persediaan pengaman (*safety stock*) merupakan suatu persediaan yang digunakan sebagai pengaman dari kelangsungan proses produksi perusahaan. Persediaan pengaman diperlukan karena dalam kenyataan jumlah bahan baku yang diperlukan untuk proses produksi tidak selalu tepat seperti yang direncanakan.

Menurut Juliani & Ridwan (2015) dalam penelitiannya mendapatkan formula persamaan untuk menentukan besarnya nilai *safety stock* dapat menggunakan beberapa cara seperti konsep pendekatan tingkat pelayanan (*service level*) dengan menggunakan persamaan sebagai berikut:

$$SS = Z\alpha \times S \times \sqrt{L} \quad \text{persamaan (1)}$$

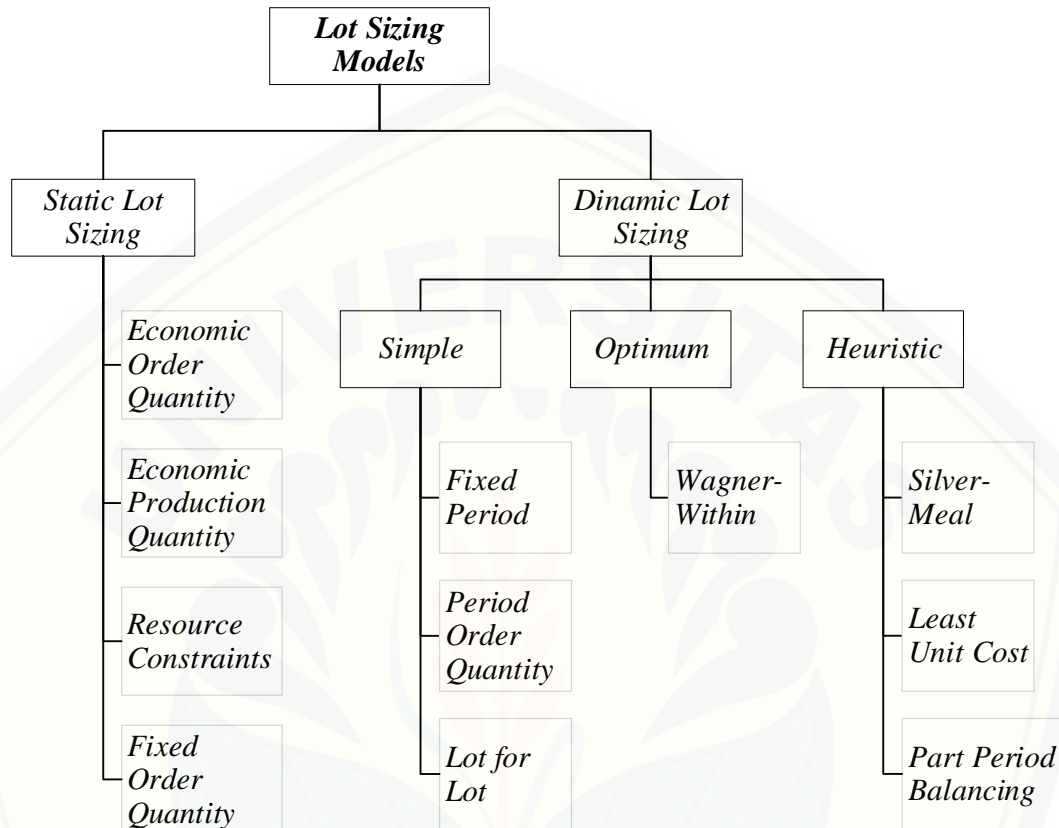
dimana:

SS	: <i>Safety stock</i>	L	: <i>Lead time</i> atau waktu <i>buffer</i>
Z α	: Tingkat pelayanan	S	: Standar deviasi permintaan

2.8. Ukuran Lot (*Lot Sizing*)

Penentuan ukuran *Lot* dalam *Distribution Requirements Planning* (DRP) merupakan masalah yang kompleks dan sulit. *Lot Sizing* diartikan sebagai kuantitas yang dinyatakan dalam penerimaan pesanan dan penyerahan pesanan dalam penjadwalan DRP. Untuk komponen yang diproduksi di dalam pabrik, *lot sizing* merupakan jumlah produksi, untuk komponen yang dibeli. *Lot Sizing* berarti jumlah yang dipesan dari supplier. Dengan demikian *lot sizing* secara umum merupakan pemenuhan kebutuhan komponen untuk satu atau lebih periode (Nadjib & Dauzere, 2017).

Metode *lot sizing* dikelompokkan berdasarkan karakteristik sifat yang diinginkan apakah statis atau dinamis. Secara singkat pengelompokan tersebut dapat dilihat pada bagan berikut:



Gambar 2. 5 Karakteristik *Lot Sizing Models* (Nadjib & Dauzere, 2017)

Teknik *lot sizing* yang menggunakan pendekatan *level by level* yang dapat menjadi pilihan untuk digunakan adalah sebagai berikut:

1. Jumlah pesanan tetap atau *Fixed Order Quantity* (FOQ).
2. Jumlah pesanan ekonomi atau *Economic Order Quantity* (EOQ)
3. Lot untuk lot atau *Lot for Lot* (LFL).
4. Kebutuhan periode tetap atau *Fixed Period Requirements* (FPR).
5. Jumlah pesanan periode atau *Period Order Quantity* (POQ).
6. Ongkos unit terkecil atau *Least Unit Cost* (LUC).
7. Ongkos total terkecil atau *Least Total Cost* (LTC).
8. Keseimbangan suatu periode atau *Part Period Balancing* (PBB).
9. Metode *Silver Meal* (SM).
10. Algoritma *Wagner Whittin* (AWW).

Lot Sizing yang digunakan pada penelitian ini adalah Teknik *Lot for Lot*. Teknik penerapan ukuran *lot* yang dilakukan atas dasar pesanan diskrit. Disamping itu teknik ini merupakan cara paling sederhana dari semua teknik ukuran *lot* yang ada. Teknik ini selalu melakukan perhitungan kembali (bersifat dinamis) terutama apabila terjadi perubahan pada kebutuhan bersih. Penggunaan teknik ini bertujuan untuk meminimumkan ongkos simpan, sehingga dengan teknik ini ongkos simpan menjadi nol. Oleh karena itu, sering sekali digunakan untuk item-item yang mempunyai biaya simpan per unit sangat mahal. Apabila dilihat dari pola kebutuhan yang mempunyai sifat diskontinu atau tidak teratur, maka teknik L-4-L ini memiliki kemampuan yang baik.

Pada teknik ini, pemenuhan kebutuhan bersih dilaksanakan disetiap periode yang membutuhkannya, sedangkan besar ukuran kuantitas pemesanan (*lot sizing*) adalah sama dengan jumlah kebutuhan bersih yang harus dipenuhi pada periode yang bersangkutan. Teknik ini biasanya digunakan untuk item-item yang mahal atau yang tingkat diskontinuitas permintaanya tinggi.

2.9. *Distribution Requirements Planning (DRP)*

Distribution Requirements Planning (DRP) mampu merumuskan kemungkinan permintaan berkaitan dengan kebutuhan untuk pengiriman dari titik produksi menuju tujuan akhir pendistribusian. Hal ini akan membantu untuk melakukan tindakan-tindakan korektif yang diambil sebelum kejadian-kejadian yang tidak diinginkan berkembang menjadi krisis (Gasper, 2004). Berdasarkan pendapat tersebut dapat disimpulkan bahwa *DRP* adalah suatu sistem yang menentukan perencanaan kebutuhan untuk mengisi kembali inventori pada pusat distribusi. *DRP* lebih tepat digolongkan ke dalam bagian dari manajemen permintaan.

DRP merupakan suatu pendekatan perencanaan yang menggunakan perencanaan permintaan pada titik yang memiliki kebutuhan untuk menetapkan peramalan permintaan kepada pusat. *DRP* berfungsi sebagai jembatan antara konsumen, manajemen permintaan, dan penjadwalan produksi. Maka *DRP* mempunyai tugas untuk mengkoordinasikan aliran produk mulai dari selesai

diproduksi, disimpan di gudang, proses pengiriman produk, penyimpanan di distribution centre sampai pada akhirnya produk tersebut sampai di tangan konsumen.

Menurut Sofyan (2013) metode DRP terdiri dari dua bagian, bagian pertama merupakan informasi deskriptif dan bagian kedua berisi informasi dari waktu ke waktu. Informasi deskriptif tersebut meliputi hal-hal sebagai berikut:

1. Persediaan Awal (*On Hand Balance*)

On hand balance adalah jumlah barang yang siap untuk dijual di lokasi persediaan atau cabang distribusi. Untuk gudang barang jadi atau cabang distribusi, *on hand balance* adalah jumlah barang yang siap untuk dikirim. Untuk gudang bahan baku dan komponen *on hand balance* adalah jumlah barang yang siap digunakan pada rantai produksi. *On hand balance* tidak termasuk jumlah barang yang masih dalam perjalanan maupun produk cacat.

2. Persediaan Pengaman (*Safety Stocks*)

Tingkat *safety stock* yang berlebihan dalam sistem akan melemahkan integritas dalam perhitungan. *Safety stock* pada DRP digunakan untuk mengantisipasi ketidakpastian permintaan relatif ramalan-ramalan yang dibuat dalam sebuah proses. Pendekatan terhadap *safety stock* adalah menghitung batas terakhir *planned orders* berdasarkan data saat *projected on hand* bernilai negatif.

3. *Lead Time*

Lead time adalah waktu yang dibutuhkan sejak dilakukannya pemesanan hingga waktu diterimanya pesanan tersebut di toko, gudang penyimpanan, atau cabang distribusi. Untuk bagian logistik, *lead time* dimulai pada saat ditentukannya kebutuhan suatu produk hingga pada saat dapat diambilnya inventori yang tersedia untuk memenuhi kebutuhan konsumen.

4. *Order Quantity*

Order Quantity adalah kuantitas order dari komponen yang harus dipesan berdasarkan *Lot Sizing* persediaan yang ada untuk memenuhi kebutuhan dalam sebuah pemesanan dimana ada beberapa jenis yang bisa digunakan pada *lot sizing* model data dinamis.

5. *Period*

Period didapatkan dari penggunaan model *push system* pada DRP. *Push System* adalah suatu sistem dimana operasi (produksi, pengadaan, pergerakan material, distribusi produk) terjadi hanya sebagai respon terhadap perencanaan penjadwalan untuk setiap operasi tanpa memperhitungkan status *real-time* dari operasi yang bersangkutan dengan tujuan untuk mengoperasikan sebuah penjadwalan.

Tabel Informasi yang digunakan untuk melakukan perhitungan dalam metode *distribution requirement planning* adalah sebagai berikut:

Tabel 2. 1 Tabel *Distribution Requirement Planning*

<i>Distribution Center X</i>									
<i>On Hand Balance:</i>					<i>Lead Time:</i>				
<i>Safety Stock:</i>					<i>Order Quantity:</i>				
	<i>Past Due</i>	<i>Period</i>							
		1	2	3	4	5	6	7	8
<i>Gross Requirement</i>									
<i>Schedule Receipts</i>									
<i>Project on Hand</i>									
<i>Net Requirements</i>									
<i>Planned Order Receipts</i>									
<i>Planned Order Release</i>									

Menurut Bowersox, Closs, dan Cooper (2012) perhitungan untuk mengisi tabel yang terdapat pada sistem *Distribution Requirements Planning (DRP)* adalah sebagai berikut:

1. Menentukan *Gross Requirement (GR)* yang diperoleh dari jumlah permintaan konsumen.
2. Dilakukan perhitungan untuk mendapatkan *Net Requirement (NR)* yang mengidentifikasi kuantitas produk bersih yang dibutuhkan untuk memenuhi permintaan konsumen dengan perhitungan:

$$NR = (GR + SS) - (SR + POH_{n-1}) \qquad \text{persamaan (2)}$$

Nilai NR yang dicatat adalah yang bernilai positif.

3. Dihasilkan sebuah *Planned Order Receipt* (PORc) sejumlah dengan NR yang diperlukan pada periode terkait.
4. Penentuan untuk kapan dilakukannya pengiriman sesuai dengan NR ditentukan dengan cara mengurangi hari PORc dengan lead time yang dibutuhkan untuk mendatangkan produk dan dikenal dengan *Planned Order Release* (PORl).
5. Besarnya PORl akan menjadi GR pada periode yang sama untuk level yang lebih tinggi dari jaringan distribusi.
6. Kemudian akan dihitung nilai *Projected on Hand* (PoH) atau *On Hand Balance* pada periode terkait dengan rumus:

$$POH = (POH_{n-1} + SR + POR_c) - GR \quad \text{persamaan (3)}$$

2.10. Metode Perbandingan Eksponensial (MPE)

Metode Perbandingan Eksponensial (MPE) adalah salah satu metode dari Sistem Pendukung Keputusan (SPK) yang digunakan untuk menentukan urutan prioritas alternatif keputusan dengan multi kriteria (Marimin, 2007). Untuk nilai skor yang dihasilkan, akan menggambarkan urutan prioritas yang menjadi besar, ini mengakibatkan urutan prioritas alternatif keputusan menjadi lebih nyata.

Prinsipnya MPE merupakan metode skoring terhadap pilihan yang ada. Dengan perhitungan secara eksponensial, perbedaan nilai antar kriteria dapat dibedakan tergantung kepada kemampuan orang yang menilai (Rangkuti, 2011). Langkah-langkah yang perlu dilakukan dalam pemilihan keputusan dengan MPE adalah sebagai berikut:

1. Penentuan alternatif keputusan;
2. Penyusunan kriteria keputusan yang akan dikaji,
3. Penentuan derajat kepentingan relatif setiap kriteria keputusan dengan menggunakan skala konversi tertentu sesuai keinginan pengambil keputusan,
4. Penentuan derajat kepentingan relatif dari setiap alternatif keputusan dan
5. Pemingkatan nilai yang diperoleh dari setiap alternatif keputusan.

BAB 3. METODE PENELITIAN

Bab ini akan dibahas mengenai metodologi penelitian, yaitu tahapan-tahapan yang dilalui oleh peneliti mulai dari analisa kebutuhan data untuk pengembangan sistem.

3.1. Tempat Penelitian

Tempat penelitian diambil secara *purposive sampling* yaitu menentukan sampel penelitian dengan beberapa pertimbangan tertentu yang bertujuan agar data yang diperoleh nantinya bisa lebih representatif (Sugiyono, 2010).

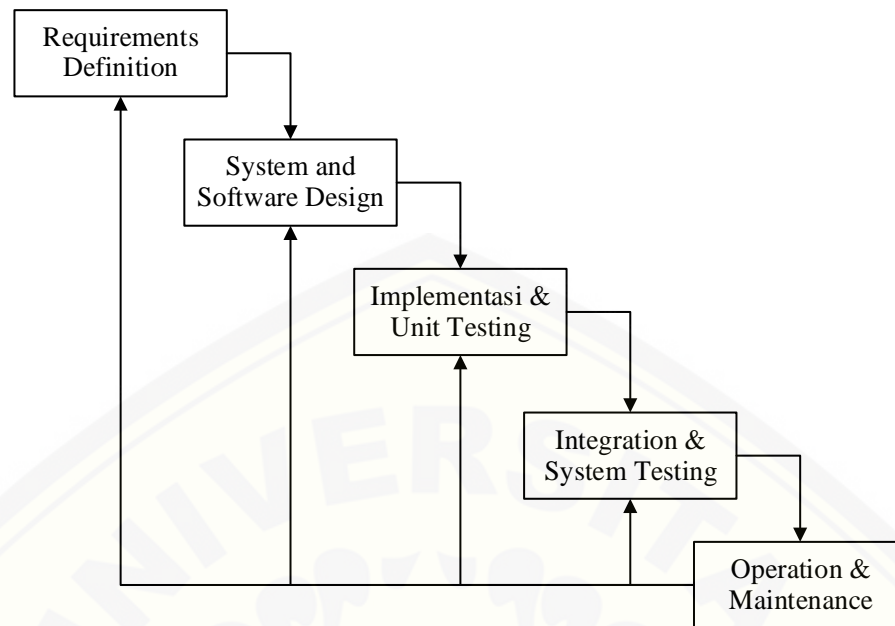
Tempat penelitian yang dipilih adalah Provinsi Jawa Timur dengan sistem pemilihan data berdasarkan keputusan dari pihak pemerintah. Penelitian dilakukan di Dinas Pertanian Jawa Timur, Dinas Perindustrian dan Perdagangan Jawa Timur, BPS Jawa Timur, BPS Jawa Timur. Waktu penelitian dilaksanakan pada bulan Januari 2018 sampai dengan Maret 2018.

3.2. Jenis Penelitian

Jenis penelitian yang digunakan adalah penelitian pengembangan sistem. Penelitian jenis ini digunakan untuk menganalisa data menggunakan metode *Distribution Requirements Planning (DRP)* yang diterapkan dalam proses perhitungan dan penulisan kode pengembangan sistem informasi. Sedangkan model yang digunakan menggunakan pendekatan melalui beberapa tahap untuk menganalisis dan merancang sistem melalui penggunaan siklus kegiatan secara spesifik dengan model *waterfall*.

3.3. Tahapan Penelitian

Model yang digunakan dalam penelitian ini adalah model *waterfall*, dimana proses yang dilakukan menjelaskan alur dari analisis kebutuhan hingga proses pengembangan dan pemeliharaan sistem. Adapun alur kerja *waterfall* ditunjukkan pada gambar 3.2.

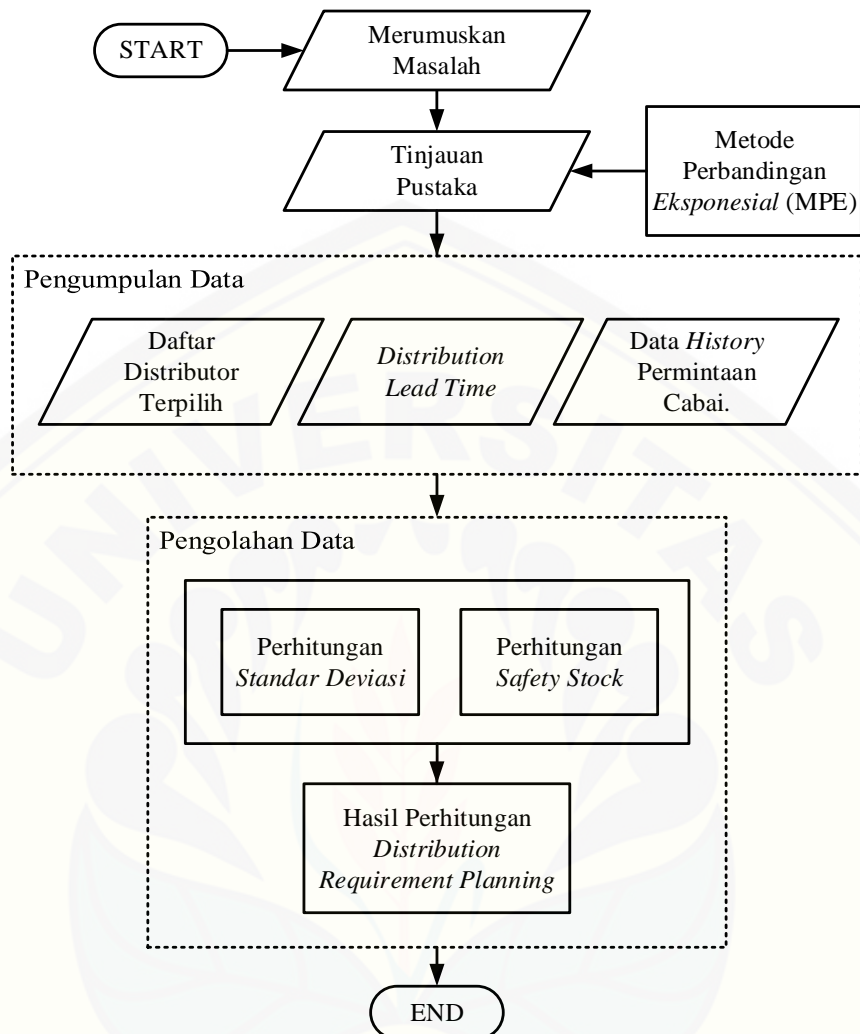


Gambar 3. 1 *Model Waterfall* (Pressman, 2001)

3.4.1. Analisis Kebutuhan (*Requirements Definition*)

Analisis kebutuhan merupakan tahap untuk mengumpulkan data dan informasi yang dibutuhkan untuk membangun sistem. Data tersebut dikelompokkan menjadi kebutuhan fungsional dan kebutuhan non fungsional. Untuk memahami sifat program yang akan dibangun, maka harus memahami informasi yang dibutuhkan untuk perangkat lunak, fungsi yang diperlukan, alur, kinerja dan *interface* dari program yang akan dibangun (Pressman, 2001).

Tahapan-tahapan yang dilakukan dalam analisis kebutuhan data cabai dapat dilihat pada gambar 3.1.



Gambar 3. 2 *Flowchart* Penelitian

Penjelasan dari gambar desain analisis kebutuhan menggunakan dua tahapan metode penelitian yang dijelaskan sebagai berikut:

1. Tahap Pengumpulan Data

Tahap pengumpulan data bertujuan untuk memperoleh data atau informasi yang dibutuhkan dalam mencapai tujuan penelitian. Pengumpulan data pada penelitian ini dilakukan melalui beberapa cara yaitu:

a. Studi Literatur

Studi literatur berisi uraian tentang teori, temuan dan bahan penelitian lain yang diperoleh dari bahan acuan untuk dijadikan landasan kegiatan

penelitian. Studi literatur berisi ulasan, rangkuman data pemikiran penulisan tentang beberapa sumber pustaka (dapat berupa artikel, buku, slide, informasi internet dan lain-lain) tentang topik yang dibahas dan biasanya ditempatkan pada bab awal (Hasibuan, 2007).

Studi literatur yang dilakukan pada penelitian ini fokus pada kajian teori dan studi terkait *Distribution Requirement Planning* (DRP) yang pernah dilakukan sebelumnya. Dalam DRP terdapat beberapa formula yang dibutuhkan untuk melanjutkan penelitian dan pengembangan sistem.

b. Wawancara

Wawancara merupakan teknik pengumpulan data survei berupa tanya jawab peneliti dengan narasumber. Wawancara tersebut berupa percakapan langsung antar dua pihak atau lebih untuk mendapatkan informasi secara lisan dengan tujuan untuk memperoleh data yang dapat menjelaskan ataupun menjawab suatu permasalahan penelitian (Hasibuan, 2007).

Peneliti melakukan wawancara kepada pihak penanganan distribusi cabai di Jawa Timur untuk mengetahui permasalahan yang ada dan memperoleh data-data cabai yang nantinya akan diolah dalam penelitian ini. Data dan informasi yang diperoleh dari tahapan ini antara lain data produktivitas cabai, data konsumsi cabai, dan data distributor terpilih dengan berdasarkan pakar dan kriteria yang mereka tentukan.

2. Tahap Pengolahan Data

Data yang telah diperoleh pada tahap pengumpulan data diolah untuk menentukan kebutuhan untuk mengembangkan sistem informasi distribusi cabai. Pengolahan data untuk perhitungan dengan metode *Distribution Requirements Planning* (DRP) dilakukan dengan beberapa cara sebagai berikut:

a. Menentukan Nilai Standar Deviasi

Standar deviasi merupakan nilai statistik yang digunakan untuk menentukan bagaimana sebaran data dalam sampel, dan seberapa dekat titik data individu ke *mean* (rata-rata) nilai sampel. Sebuah standar deviasi dari kumpulan data sama dengan nol menunjukkan bahwa semua nilai-nilai dalam

himpunan tersebut adalah sama. Sebuah nilai deviasi yang lebih besar akan memberikan makna bahwa titik data individu jauh dari nilai rata-rata.

b. Menghitung Safety Stock

Persediaan pengaman (*safety stock*) merupakan suatu persediaan yang digunakan sebagai pengaman dari kelangsungan proses produksi perusahaan. Persediaan pengaman diperlukan karena dalam kenyataan jumlah bahan baku yang diperlukan untuk proses produksi tidak selalu tepat seperti yang direncanakan.

3.4.2. Sistem dan Desain Software (*System and Software Design*)

Tahapan berikutnya adalah desain sistem menggunakan *Unified Modeling Language (UML)* yang dirancang menggunakan konsep *Object-Oriented Programming (OOP)*. Berikut pemodelan UML yang digunakan antara lain:

1. *Business Process*

Business process merupakan suatu kumpulan aktivitas atau pekerjaan terstruktur yang saling terkait untuk menyelesaikan suatu masalah tertentu atau yang menghasilkan produk atau layanan (demi meraih tujuan tertentu). Suatu proses bisnis dapat dipecah menjadi beberapa subproses yang masing-masing memiliki atribut sendiri tetapi juga berkontribusi untuk mencapai tujuan dari prosesnya. Dalam pengembangan sistem informasi distribusi cabai ini memiliki tiga jenis proses yang terlibat proses manajemen, proses operasional, dan proses pendukung.

2. *Use Case Diagram*

Use case diagram gambaran graphical dari beberapa atau semua actor, use case, dan interaksi diantaranya yang memperkenalkan suatu sistem. *Use case diagram* tidak menjelaskan secara detil tentang penggunaan use case, tetapi hanya memberi gambaran singkat hubungan antara *use case*, aktor, dan sistem. Dalam *use case* ini akan diketahui fungsi-fungsi apa saja yang berada pada sistem dan aktor bagian apa saja yang melakukan operasi pada sistem.

3. *Use Case Scenario*

Use case scenario digunakan untuk menjelaskan cara kerja sistem berdasarkan tugas pengguna yang terdapat pada *use case diagram*. *Use case scenario* terdiri dari nama *use case*, aksi aktor dan reaksi sistem. *Use case scenario* terbagi menjadi dua kondisi yakni *use case scenario normal* yang menjelaskan proses dalam sistem berjalan normal tanpa terjadi kesalahan atau *error*. Sedangkan *use case scenario alternatif* menjelaskan proses sistem tidak berjalan normal atau mengalami *error*. *Use case scenario normal* dan *use case scenario alternatif* dapat berjumlah lebih dari satu.

4. *Sequence Diagram*

Sequence Diagram digunakan untuk menunjukkan interaksi antar objek pada sebuah sistem berupa pesan yang menggambarkan kolaborasi dinamis antara sejumlah object. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antara objek. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. Secara umum *sequence* merupakan *blue print* bagi *programmer*.

5. *Activity Diagram*

Activity Diagram digunakan untuk menggambarkan urutan aktivitas dalam sebuah proses yang terjadi dalam sistem. Aktivitas tersebut sesuai dengan *use case scenario* yang berisi tugas *user* dan reaksi sistem dan digambarkan dalam bentuk diagram dan penjelasan atribut yang digunakan.

6. *Class Diagram*

Class Diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti pewarisan, asosiasi, dependensi dan lain-lain. *Class* yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem.

7. *Entity Relationship Diagram*

Entity Relationship Diagram (ERD) menjelaskan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD digunakan untuk memodelkan

struktur data dan hubungan antar data dalam database, untuk menggambarannya digunakan beberapa notasi dan simbol.

3.4.3. Implementasi Sistem (*Implementation and Unit Testing*)

Tahap implementasi dilakukan berdasarkan desain sistem yang selanjutnya diubah dalam bentuk program, yaitu:

1. Penulisan program menggunakan bahasa pemrograman *Page Hyper Text Processor (PHP)* dengan framework *Codeigniter (CI)*.
2. *Database Management System (DBMS)* yang digunakan adalah *MySQL* dengan menggunakan jaringan lokal aplikasi *XAMPP*.

3.4.4. Pengujian Sistem (*Integration and System Testing*)

Tahap pengujian sistem bertujuan untuk mengetahui sejauh mana sistem ini dapat berjalan. *Integration Testing* berfungsi untuk mengetahui apakah sistem ini dapat berfungsi dengan baik sesuai dengan yang diharapkan, serta untuk mengetahui letak kekurangan pada sistem. Untuk melakukan pengujian terdapat beberapa cara sebagai berikut:

1. *White Box Testing*

White box testing merupakan cara pengujian dengan melihat modul yang telah dibuat dan program-program yang ada. Jika ada modul yang menghasilkan output yang tidak sesuai, maka baris-baris program, variabel dan parameter yang terlibat pada unit tersebut satu persatu akan dicek dan diperbaiki, kemudian di *compile* ulang (Pressman, 2001). Pengujian ini dilakukan oleh pembuat program (*developer*).

2. *Black Box Testing*

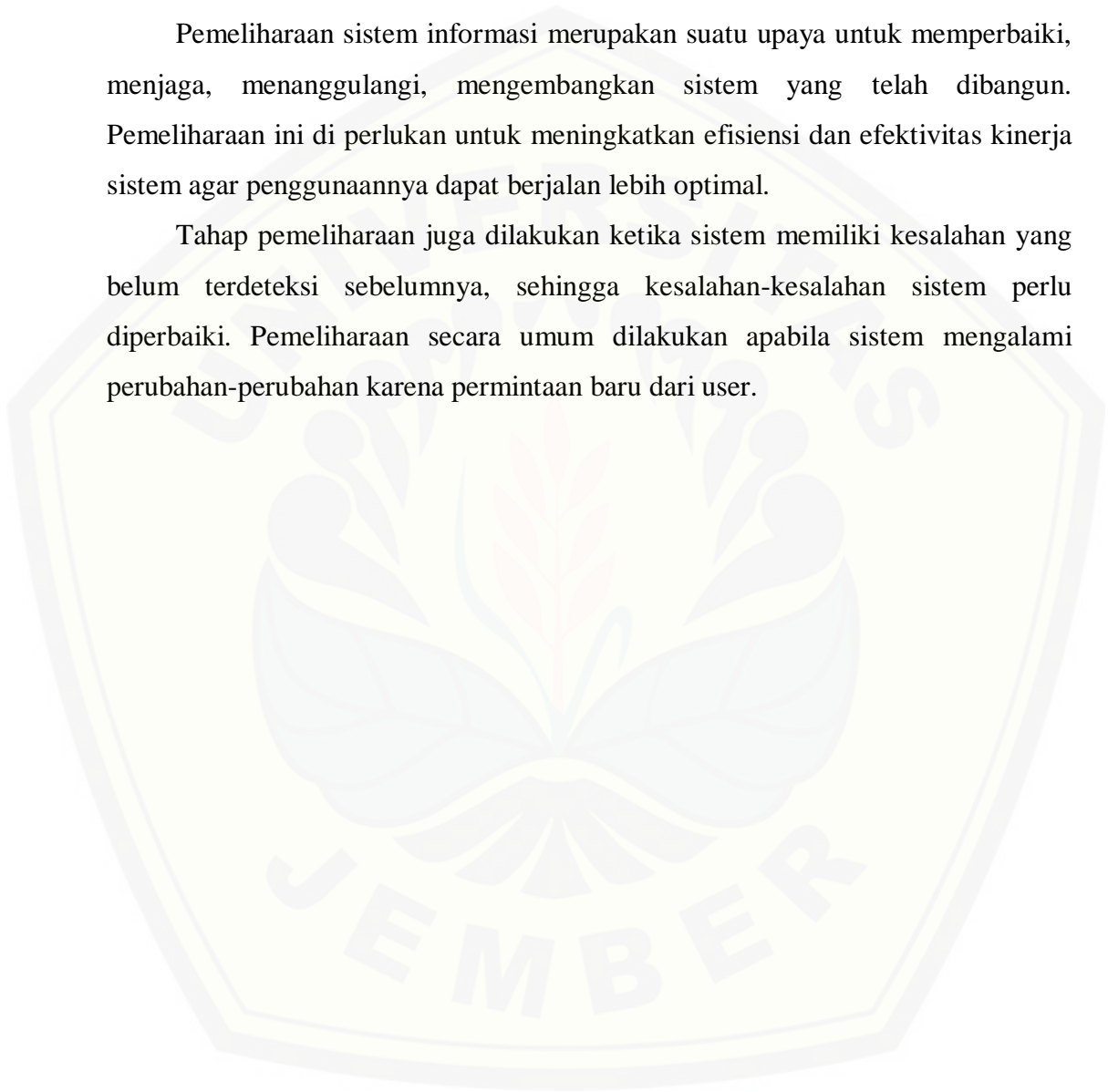
Black box testing merupakan metode pengujian perangkat lunak yang memeriksa fungsional dari aplikasi yang berkaitan dengan struktur internal atau kerja. Metode ini memfokuskan pada keperluan fungsional dari software (Pressman, 2001). Pengujian dilakukan dengan cara menjalankan semua fitur yang terdapat dalam sistem. Saat pengguna menjalankan fitur tertentu maka sistem akan

memberikan informasi sesuai dengan yang dibutuhkan. Status yang dijalankan dalam kondisi berfungsi sukses atau gagal.

3.4.5. Pemeliharaan (*Operation and Maintenance*)

Pemeliharaan sistem informasi merupakan suatu upaya untuk memperbaiki, menjaga, menanggulangi, mengembangkan sistem yang telah dibangun. Pemeliharaan ini di perlukan untuk meningkatkan efisiensi dan efektivitas kinerja sistem agar penggunaannya dapat berjalan lebih optimal.

Tahap pemeliharaan juga dilakukan ketika sistem memiliki kesalahan yang belum terdeteksi sebelumnya, sehingga kesalahan-kesalahan sistem perlu diperbaiki. Pemeliharaan secara umum dilakukan apabila sistem mengalami perubahan-perubahan karena permintaan baru dari user.



BAB 4. PENGEMBANGAN SISTEM

Pengembangan sistem ini dilakukan dengan menggunakan model *waterfall*. Model ini melakukan pendekatan kepada perangkat lunak sistematis dan sekuensial yang mulai pada tingkat kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan.

4.1. Analisis Kebutuhan

Analisis kebutuhan perangkat lunak dalam penelitian ini yaitu dengan cara mengidentifikasi permasalahan untuk dijadikan bahan membangun sistem informasi distribusi cabai. Analisis kebutuhan meliputi proses pengumpulan data kebutuhan fungsional dan kebutuhan non-fungsional. Data yang terkumpul adalah data produksi cabai, data konsumsi cabai, data riwayat distribusi cabai.

4.1.1. Pemilihan Distributor Cabai

Proses pengambilan keputusan untuk memilih data distributor yang digunakan sebagai analisa distribusi dilakukan oleh beberapa pihak antara lain instansi pemerintah daerah (Biro Perekonomian, Dinas Pertanian, Dinas Perdagangan dan Perindustrian), Bulog, BPS dan Kantor Bank Indonesia yang bertugas untuk menjaga ketersediaan pasokan, kelancaran distribusi, yang tergabung dalam Tim Pemantauan dan Pengendalian Inflasi Daerah (TPID). Setiap instansi mempunyai peran berbeda untuk menjaga kestabilan distribusi cabai. Sehingga dalam pembangunan sistem ini tidak bisa berjalan sendiri perlu adanya dukungan dari pemerintah serta dukungan berupa teknologi pertanian untuk mengatasi masalah distribusi dan permintaan cabai.

Pemilihan data distributor cabai di Provinsi Jawa Timur dilakukan secara kolektif menggunakan Metode Perbandingan Eksponensial (MPE). MPE merupakan salah satu metode pengambilan keputusan yang mengkuantifikasikan pendapat seseorang atau lebih dalam skala tertentu. Pada prinsipnya merupakan metode skoring terhadap pilihan yang ada. Dengan perhitungan secara

eksponensial, perbedaan nilai antar kriteria dapat dibedakan tergantung kepada kemampuan orang yang menilai. Dalam penerapan MPE, untuk menentukan derajat kepentingan/bobot dari setiap kriteria yang ditetapkan pada distributor cabai dilakukan secara langsung oleh ahli dan berdasarkan data laporan tahunan. Pemberian bobot bersifat subjektif yang dilakukan secara langsung tanpa melakukan perbandingan relatif terhadap kriteria lainnya.

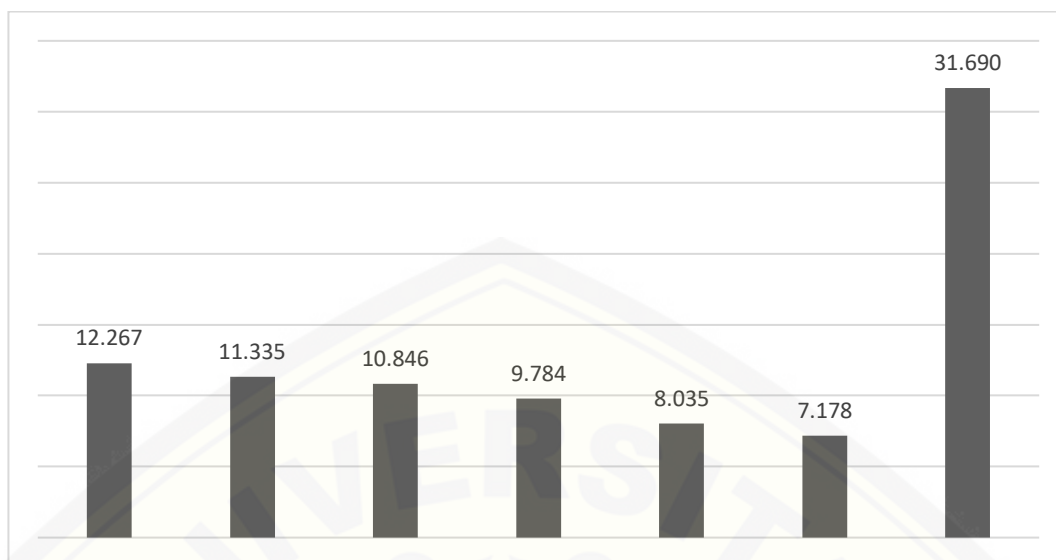
1. Data Terpilih Distributor Cabai Besar dengan Jumlah Produksi Tertinggi.

Distributor cabai besar dipilih dengan kriteria jumlah produksi yang tinggi, agar tingkat akurasi perhitungan dan pendataan mudah dilakukan. Data distributor cabai besar terpilih di Provinsi Jawa Timur dapat dilihat pada tabel 4.1.

Tabel 4. 1 Distributor Cabai Besar Terpilih di Jawa Timur Tahun 2015

Kab/Kota	Produksi (ton)	Share (%)	Rangking
Bojonegoro	12.267	13,46	1
Tuban	11.335	12,44	2
Blitar	10.846	11,90	3
Malang	9.784	10,74	4
Kediri	8.035	8,82	5
Banyuwangi	7.178	7,88	6
Lainnya	31.690	34,77	7
Total	91.135	100,00	

Produksi cabai besar di Jawa Timur terdapat di sebagian besar kabupaten, terbanyak di Bojonegoro. Pada tahun 2015 produksi cabai besar dari Bojonegoro mencapai 12,26 ribu ton atau 13,46% dari total produksi cabai besar Jawa Timur, diikuti oleh Tuban dengan kontribusi sebesar 12,44%. Adapun statistik produksi cabai besar di Jawa Timur tahun 2015 dapat dilihat pada gambar 4.1



Gambar 4. 1 Statistika Produksi Cabai Besar di Jawa Timur Tahun 2015

2. Data Terpilih Distributor Cabai Rawit dengan Jumlah Produksi Tertinggi.

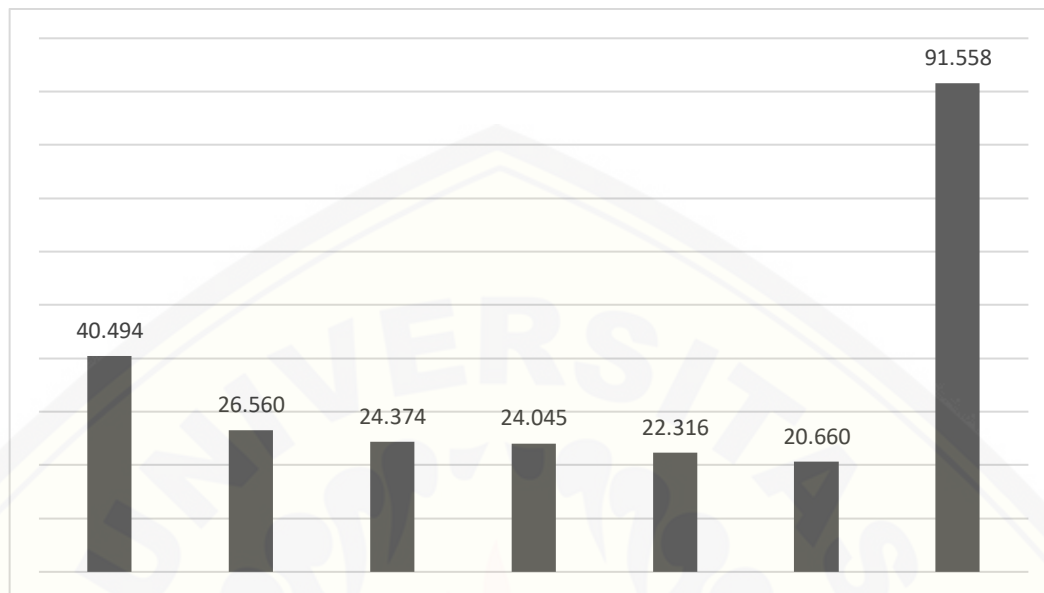
Distributor cabai rawit dipilih dengan kriteria jumlah produksi yang tinggi, agar tingkat akurasi perhitungan dan pendataan mudah dilakukan. Data distributor cabai rawit terpilih di Provinsi Jawa Timur dapat dilihat pada tabel 4.2.

Tabel 4. 2 Distributor Cabai Rawit Terpilih di Jawa Timur Tahun 2015

Kab/Kota	Produksi (ton)	Share (%)	Rangking
Blitar	40.494	16,20	1
Kediri	26.560	10,62	2
Sampang	24.374	9,75	3
Jember	24.045	9,62	4
Malang	22.316	8,93	5
Lumajang	20.660	8,26	6
Lainnya	91.558	36,62	7
Total	250.007	100,00	

Produksi cabai rawit di Jawa Timur terdapat di sebagian besar kabupaten, terutama di Blitar. Pada tahun 2015 produksi cabai rawit dari Blitar mencapai 40,49 ribu ton atau 16,20% dari total produksi cabai rawit Jawa Timur, diikuti oleh Kediri

dengan kontribusi sebesar 10,62%. Adapun statistik produksi cabai rawit di Jawa Timur tahun 2015 dapat dilihat pada gambar 4.2.



Gambar 4. 2 Statistika Produksi Cabai Rawit di Jawa Timur Tahun 2015

4.1.2. Data Produksi Cabai Jawa Timur

Pada tabel berikut dijelaskan jumlah permintaan konsumsi cabai besar dan cabai rawit di Jawa Timur.

1. Analisis Data Konsumsi Cabai Besar dengan Data Produksi Cabai Besar

Apabila jumlah konsumsi cabai rawit sama dengan jumlah produksi maka harga stabil sehingga tidak mengalami fluktuasi harga. Data konsumsi cabai besar dengan data produksi cabai besar dapat dilihat pada tabel 4.3.

Tabel 4. 3 Data Konsumsi Cabai Besar dengan Data Produksi Cabai Besar

Tahun	Konsumsi (ton)	Produksi (ton)
2010	22.680	71.565
2011	22.859	81.000,99
2012	23.019	99.676,3
2013	23.073	101.690,8
2014	24.083	111.806,7
2015	25.122	89.862,2

2. Analisis Data Konsumsi Cabai Rawit dengan Data Produksi Cabai Rawit

Data yang dihasilkan merupakan konsumsi cabai rawit dengan hasil produksi di Kabupaten terpilih di Jawa Timur. Pada tabel dapat disebutkan bahwa apabila jumlah konsumsi cabai rawit sama dengan jumlah produksi maka harga stabil sehingga tidak mengalami fluktuasi harga. Data konsumsi cabai rawit dengan data produksi cabai rawit dapat dilihat pada tabel 4.4.

Tabel 4. 4 Data Konsumsi Cabai Rawit dengan Data Produksi Cabai Rawit

Tahun	Konsumsi (ton)	Produksi (ton)
2010	28.875	145.584
2011	29.152	204.588,89
2012	29.618	324.147,39
2013	29.058	226.034,9
2014	30.223	251.383,09
2015	30.422	253.396,79

4.1.3. Data Konsumsi Cabai Jawa Timur

Data dari BPS pada tahun 2016 menunjukkan bahwa cabai besar dan cabai rawit memiliki tingkat konsumsi yang semakin meningkat. Pada tabel berikut dijelaskan jumlah konsumsi cabai besar dan cabai rawit di Jawa Timur. Data jumlah konsumsi cabai besar dapat dilihat pada tabel 4.5.

Tabel 4. 5 Jumlah Konsumsi Cabai Besar di Jawa Timur

Tahun	Konsumsi Perkapita (kg)	Jumlah Penduduk (jiwa)	Total Konsumsi (ton)
2010	0,05019	37.656.800	22.680
2011	0,05034	37.840.700	22.859
2012	0,05034	38.106.600	23.019
2013	0,05012	38.363.200	23.073
2014	0,05198	38.610.200	24.083
2015	0,05389	38.847.600	25.122

Adapun data jumlah konsumsi cabai rawit di Jawa Timur pada tahun 2010 hingga tahun 2015 mengalami peningkatan. Jumlah konsumsi cabai rawit dapat dilihat pada tabel 4.6.

Tabel 4. 6 Jumlah Konsumsi Cabai Rawit di Jawa Timur

Tahun	Konsumsi Perkapita (kg)	Jumlah Penduduk (jiwa)	Total Konsumsi (ton)
2010	0,06392	37.656.800	28.875
2011	0,06462	37.840.700	29.152
2012	0,06477	38.106.600	29.618
2013	0,06312	38.363.200	29.058
2014	0,06523	38.610.200	30.223
2015	0,06526	38.847.600	30.422

4.1.4. Data Distribusi Cabai Jawa Timur

Berdasarkan hasil survei, diperoleh informasi bahwa kategori pedagang besar cabai rata-rata memperoleh margin (rasio MPP) sebesar 27 persen dan rata-rata perolehan margin setelah dikurangi biaya transportasi (rasio MP) adalah sebesar 26,94 persen. Pedagang eceran cabai rata-rata memperoleh margin (rasio MPP) sebesar 33,88 persen, sedangkan rata-rata perolehan margin setelah dikurangi biaya transportasi (rasio MP) adalah sebesar 33,87 persen. Rata-rata perolehan margin pedagang cabai di Provinsi Jawa Timur adalah sebesar 27 persen dan rata-rata perolehan margin setelah dikurangi biaya transportasi adalah sebesar 26,94 persen. Data distribusi cabai dapat dilihat pada tabel 4.7.

Tabel 4. 7 Data Distribusi Cabai Jawa Timur

Tahun	PB	PE	PB & PE
Rata-rata Nilai Penjualan (Rp)	21.511.307,54	56.657,67	19.127.457,56
Rata-rata Nilai Pembelian (Rp)	16.937.968,58	42.318,67	15.060.674,15
Rata-rata Biaya Transportasi (Rp)	13.705,01	15,00	13.020,51
Rata-rata MPP (000 Rp)	4.573.338,96	14.339,00	4.066.783,41
Rasio MPP (%)	27,00	33,88	27,00

Tahun	PB	PE	PB & PE
Rata-rata MP (000 Rp)	4.562.489,16	14.334,00	4.057.138,59
Rasio MP (%)	26,94	33,87	26,94

4.1.5. Kebutuhan Fungsional

Berdasarkan data yang terkumpul dalam tahapan analisa, maka diperoleh kebutuhan fungsional dan non fungsional yang dibutuhkan untuk membangun aplikasi optimalisasi distribusi cabai. Kebutuhan fungsional dan non fungsional merupakan hal pokok yang harus dilakukan oleh sistem dalam menerima masukan untuk diproses sehingga menghasilkan keluaran. Kebutuhan fungsional berisi proses-proses apa saja yang nantinya dibutuhkan oleh sistem informasi distribusi cabai adalah sebagai berikut:

1. Sistem mampu mengelola data produktivitas cabai meliputi menambah, mengubah dan menghapus.
2. Sistem mampu mengelola data konsumsi cabai meliputi menambah, mengubah dan menghapus.
3. Sistem mampu mengelola data jenis cabai meliputi menambah, mengubah dan menghapus.
4. Sistem mampu menampilkan data jenis cabai.
5. Sistem mampu menampilkan data kabupaten atau kota produksi cabai.
6. Sistem mampu menampilkan hasil kalkulasi distribusi cabai menggunakan metode *Distribution Requirements Planning*.
7. Sistem mampu mengelola data tahun meliputi menambah, mengubah dan menghapus.
8. Sistem mampu mengelola data kabupaten dan kota meliputi menambah, mengubah dan menghapus.
9. Sistem mampu mengelola data user meliputi menambah, mengubah dan menghapus.

4.1.6. Kebutuhan Non-fungsional

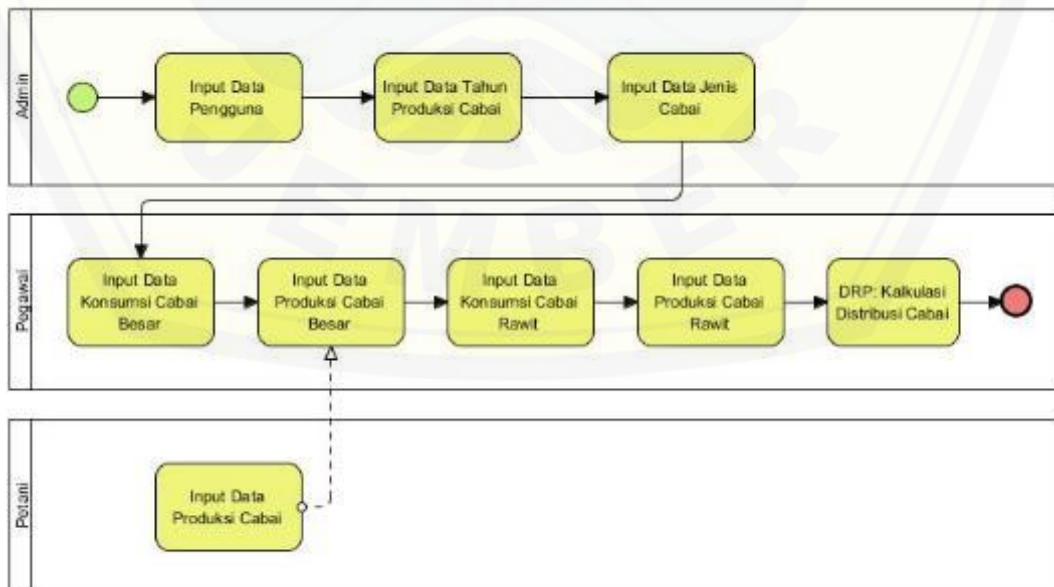
Kebutuhan non-fungsional pada sistem informasi distribusi cabai meliputi bahasa komunikasi dalam sistem mudah dimengerti oleh pengguna dan desain tampilan sistem yang *user friendly*, serta aplikasi bisa diakses beberapa *user/multiuser*.

4.2. Desain Sistem

Desain sistem yang dibuat meliputi *Business Process*, *Usecase Diagram*, *Use case scenario*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram* dan *Entity Relationship Diagram (ERD)*.

4.2.1. Business Process

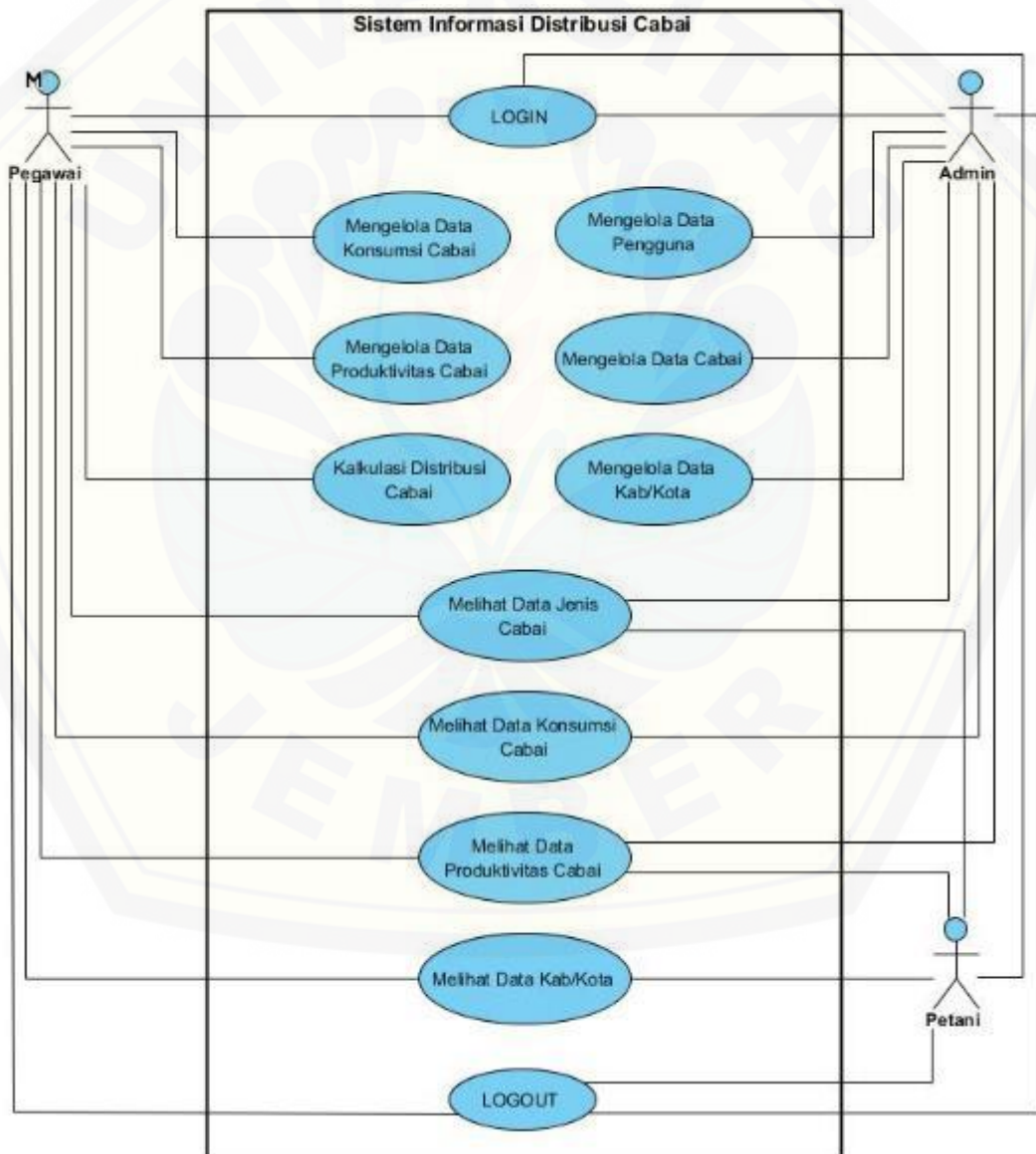
Business Process dalam pendistribusian cabai ini melibatkan 3 bagian yang menjadi tenaga ahli dan pengelola. Admin (pemerintah) memiliki hak untuk mengelola data cabai, mengelola data kabupaten. Pegawai melakukan input data produksi, input data konsumsi, kalkulasi perhitungan optimal distribusi cabai. Petani memberikan informasi produksi cabai yang didapatkan setiap panen. *Business Process* dalam manajemen proses sistem informasi distribusi cabai dapat dilihat pada gambar 4.3.



Gambar 4. 3 *Business Process* Sistem Informasi Distribusi Cabai

4.2.2. Use case Diagram

Use case diagram merupakan diagram yang digunakan untuk menjelaskan fungsi apa saja yang ada di dalam sistem dan siapa saja yang berhak mengakses fungsi-fungsi tersebut. Use case pada pengembangan sistem informasi distribusi cabai ini dijelaskan dengan 3 aktor yang dapat mengakses sistem yaitu Admin (Pemerintah), Pegawai, dan Petani. Admin memiliki akses untuk mengelola data dasar meliputi data pengguna, data cabai, dan data kota. Pegawai berhak mengelola data produktivitas dan data konsumsi cabai yang digunakan untuk kalkulasi DRP.



Gambar 4. 4 Use case Diagram

Fitur-fitur yang terdapat pada sistem beserta aktor yang berhak mengakses fitur tersebut berdasarkan *use case diagram* dijelaskan pada tabel 4.8 dan tabel 4.9.

Tabel 4. 8 Definisi Aktor

No.	Aktor	Deskripsi
1	Admin	Aktor admin memiliki hak untuk mengatur hak akses dan data dasar yang digunakan dalam aplikasi ini. Aktor admin dapat melakukan <i>login</i> dan <i>logout</i> , mengelola data user (menambah, melihat, mengubah, menghapus, dan riwayat login), mengelola data tahun (menambah, melihat, mengubah, dan menghapus), mengelola data kab/kota (menambah data kota, melihat data kota, mengubah data kota, dan menghapus data kota), dan mengelola jenis cabai (menambah data jenis cabai, melihat data jenis cabai, mengubah data jenis cabai, dan menghapus data jenis cabai), yang digunakan memprediksi jumlah konsumsi.
2	Pegawai	Aktor ini memiliki hak akses untuk mengelola data dasar yang telah disiapkan oleh admin. Aktor ini dapat melihat data pengguna, melihat data cabai, manajemen produktivitas cabai besar (menambah, melihat, mengubah, dan menghapus), manajemen konsumsi cabai besar (menambah, melihat, mengubah, dan menghapus), manajemen produktivitas cabai rawit (menambah, melihat, mengubah, dan menghapus), manajemen konsumsi cabai rawit (menambah, melihat, mengubah, dan menghapus), yang hasilnya digunakan untuk memenuhi menganalisa jumlah permintaan cabai, kemudian digunakan untuk menghitung optimalisasi

No.	Aktor	Deskripsi
		distribusi cabai dengan kategori data yang telah diolah dengan baik dalam sistem.
3	Petani	Aktor ini dapat mengelola hasil produksi cabai (menambah, melihat, dan mengubah), melihat data kabupaten kota, melihat jenis cabai, melihat data produktivitas cabai.

Tabel 4. 9 Definisi *Use case*

No.	<i>Use case</i>	Penjelasan
UC-01	Login	Merupakan <i>use case</i> yang berfungsi untuk masuk dalam sistem informasi distribusi cabai
UC-02	Mengelola Data Konsumsi Cabai	Menggambarkan proses untuk menampilkan, mengubah, menambah, dan menghapus data konsumsi cabai di Jawa Timur
UC-03	Mengelola Data Produktivitas Cabai	Merupakan <i>use case</i> yang menggambarkan proses menampilkan, mengubah, menambah, dan menghapus data produktivitas cabai di Jawa Timur
UC-04	Melihat Data Produktivitas Cabai	Merupakan <i>use case</i> yang digunakan untuk melihat data produktivitas cabai besar dan cabai rawit
UC-05	Melihat Data Konsumsi Cabai	Merupakan <i>use case</i> yang digunakan untuk melihat data konsumsi cabai besar dan cabai rawit

No.	Use case	Penjelasan
UC-06	Melihat Data Jenis Cabai	Merupakan <i>use case</i> yang digunakan untuk melihat jenis cabai yang digunakan dalam sistem.
UC-07	Kalkulasi Distribusi Cabai	Merupakan <i>use case</i> yang digunakan untuk menghitung pola optimal pendistribusian cabai di Provinsi Jawa Timur dengan contoh sejumlah kota terpilih.
UC-08	Melihat Data Kab/Kota	Merupakan <i>use case</i> yang digunakan untuk melihat kab dan kota yang ada di Jawa Timur
UC-09	Mengelola Data Pengguna	Merupakan <i>use case</i> yang menggambarkan proses menampilkan, mengubah, menambah, dan menghapus data pengguna yang berhak mengakses aplikasi ini. Setiap pengguna memiliki hak akses berbeda
UC-10	Mengelola Data Cabai	Merupakan <i>use case</i> yang menggambarkan proses menampilkan, mengubah, menambah, dan menghapus data cabai yang digunakan untuk pendistribusian
UC-11	Mengelola Data Kab/Kota	Merupakan <i>use case</i> yang menggambarkan proses menampilkan, mengubah, menambah, dan menghapus data kabupaten dan kota yang termasuk dalam wilayah provinsi Jawa Timur

No.	Use case	Penjelasan
UC-12	Logout	Merupakan <i>use case</i> yang berfungsi untuk keluar dari sistem informasi distribusi cabai

4.2.3. Use Case Scenario

Use case scenario digunakan untuk menjelaskan cara kerja sistem berdasarkan tugas user yang terdapat pada *use case diagram*. *Use case scenario* terdiri dari nama *use case*, aksi aktor dan reaksi sistem. Berikut ini *use case scenario* dari sistem informasi distribusi cabai:

1. Use Case Scenario Kalkulasi Distribusi Cabai

Penjelasan urutan reaksi aktor dan reaksi sistem pada *use case use case scenario* normal dan *use case use case scenario* alternatif *use case* melakukan kalkulasi distribusi cabai besar di Jawa Timur dapat dilihat pada tabel 4.10.

Tabel 4. 10 *Use Case Scenario* Kalkulasi Distribusi Cabai

Nomor Use case	UC-07
Nama	Kalkulasi Distribusi Cabai
Aktor	Pegawai
<i>Precondition</i>	Pegawai memilih menu Kalkulasi Distribusi
<i>Postcondition</i>	Pegawai berhasil melakukan kalkulasi distribusi cabai
USE CASE SCENARIO NORMAL	
“Kalkulasi Distribusi Cabai”	
Aktor	Sistem
1.	Memilih menu Kalkulasi Distribusi

dilanjutkan...

2. Mengambil data cabai dengan atribut yang tersedia sebagai berikut:

- Jumlah produksi pada tabel produksi.
- Jumlah konsumsi cabai pada tabel konsumsi yang berisi data pertahun.
- Data kab/kota dari tabel kabupaten berdasarkan pengaturan provinsi di *database*.

Data diakses dari *database* menggunakan *class produksi* dan *function kalkulasi*.

3. Menampilkan form seleksi dengan atribut sebagai berikut:

- Pilihan data jenis cabai
- Pilihan data kab/kota
- Pilihan data tahun produksi
- Formulir Stock Cabai
- Formulir Lot Size
- Formulir Lead Time
- Formulir Order Quantity

4. Mengisi form seleksi untuk data kalkulasi.

5. Menekan tombol Kalkulasi Cabai.

6. Menghitung distribusi cabai dengan metode *distribution requirements planning* menggunakan data jumlah produksi cabai dan jumlah konsumsi cabai di provinsi Jawa Timur.

7. Menampilkan tabel hasil perhitungan distribusi cabai yang berisi atribut sebagai berikut:

- *Gross Requirements,*
- *Schedule Receipts,*
- *Project on Hand,*
- *Net Requirements,*
- *Planned Order Receipts,*
- *Planned Order Release*

Periode data yang ditampilkan setiap kali melakukan perhitungan adalah data tahun mulai dari Januari, Februari, Maret, April, Mei, Juni, Juli, Agustus, September, Oktober, November, Desember.

8. Menekan tombol bersihkan

9. Menampilkan form seleksi dengan atribut sebagai berikut:

- Pilihan data jenis cabai
 - Pilihan data kab/kota
 - Pilihan data tahun produksi
 - Formulir Stock Cabai
-

dilanjutkan...

-
- Formulir Lot Size
 - Formulir Lead Time
 - Formulir Order Quantity

USE CASE SCENARIO ALTERNATIF

“Melihat Kalkulasi Distribusi Cabai”

Jika aktor melakukan kalkulasi distribusi cabai

Aktor

Sistem

5a. Menekan tombol Kalkulasi Cabai.

- Pilihan data jenis cabai tidak boleh kosong
 - Pilihan data kab/kota tersortir sesuai jenis cabai dan tidak boleh kosong
 - Pilihan data tahun produksi disesuaikan dengan tahun produksi yang ada
 - Formulir stock cabai terisi otomatis berdasarkan hasil perhitungan manual dan menjadi nilai ketetapan.
 - Formulir Lot Size terisi otomatis berdasarkan hasil perhitungan manual dan menjadi nilai ketetapan.
 - Formulir Lead Time terisi otomatis berdasarkan hasil perhitungan manual dan menjadi nilai ketetapan.
-

dilanjutkan...

-
- Formulir Order Quantity terisi otomatis berdasarkan hasil perhitungan manual dan menjadi nilai ketetapan.
-

6a. Menampilkan notifikasi form 'Please fill out this field'

7a. Menghitung distribusi cabai menggunakan metode *distribution requirements planning*.

8a. Menampilkan tabel hasil perhitungan distribusi cabai yang berisi atribut sebagai berikut:

- *Gross Requirements,*
- *Schedule Receipts,*
- *Project on Hand,*
- *Net Requirements,*
- *Planned Order Receipts,*
- *Planned Order Release*

Periode data yang ditampilkan setiap kali melakukan perhitungan adalah data tahun mulai dari Januari, Februari, Maret, April, Mei, Juni, Juli, Agustus, September, Oktober, November, Desember.

2. *Use Case Scenario* Mengelola Data Konsumsi Cabai

Use case scenario mengelola data konsumsi cabai menjelaskan urutan aksi admin saat menambah, menghapus, atau mengubah data produktivitas cabai dalam

sistem. Urutan aksi dan reaksi sistem pada *use case scenario* mengelola data konsumsi cabai diuraikan sebagai berikut:

- a. Pegawai memilih menu produksi dan reaksi sistem menampilkan halaman pengguna berisi tabel daftar konsumsi, tombol tambah data, tombol edit data, dan tombol hapus data.
- b. Pegawai memilih tombol tambah data dan reaksi sistem menampilkan form pada halaman tambah data dengan atribut provinsi, kabupaten, jenis cabai, tahun, dan konsumsi perkapita.
- c. Pegawai memilih tombol simpan dan reaksi sistem mengalihkan halaman tambah data ke halaman daftar konsumsi.

Penjelasan lengkap urutan aksi aktor dan reaksi sistem pada *use case scenario* normal dan *use case scenario* alternatif mengelola konsumsi cabai dapat dilihat di lampiran A.1 pada tabel A.1 *use case scenario* mengelola data konsumsi cabai.

3. *Use Case Scenario* Mengelola Data Produktivitas Cabai

Use case scenario mengelola data produktivitas cabai menjelaskan urutan aksi admin saat menambah, menghapus, atau mengubah data produktivitas cabai dalam sistem. Urutan aksi dan reaksi sistem pada *use case scenario* mengelola data produktivitas cabai diuraikan sebagai berikut:

- d. Pegawai memilih menu produksi dan reaksi sistem menampilkan halaman pengguna berisi tabel daftar produksi, tombol tambah data, tombol edit data, dan tombol hapus data.
- e. Pegawai memilih tombol tambah data dan reaksi sistem menampilkan form pada halaman tambah data dengan atribut nama, jenis kelamin, email, password, hak akses dengan tombol simpan dan tombol reset.
- f. Pegawai memilih tombol simpan dan reaksi sistem mengalihkan halaman tambah data ke halaman daftar produksi.

Penjelasan lengkap urutan aksi aktor dan reaksi sistem pada *use case scenario* normal dan *use case scenario* alternatif mengelola data produksi cabai dapat dilihat di lampiran A.2 pada tabel A.2 *use case scenario* mengelola data produksi cabai.

4. *Use Case Scenario* Mengelola Data Pengguna

Use case scenario mengelola data pengguna menjelaskan urutan aksi admin saat menambah, menghapus, atau mengubah data pengguna dalam sistem. Urutan aksi dan reaksi sistem pada *use case scenario* mengelola data pengguna diuraikan sebagai berikut:

- g. Admin memilih menu pengguna dan reaksi sistem menampilkan halaman pengguna berisi tabel daftar pengguna, tombol tambah data, tombol edit data, dan tombol hapus data.
- h. Admin memilih tombol tambah data dan reaksi sistem menampilkan form pada halaman tambah data dengan atribut nama, jenis kelamin, email, password, hak akses dengan tombol simpan dan tombol reset.
- i. Admin memilih tombol simpan dan reaksi sistem mengalihkan halaman tambah data ke halaman daftar pengguna.

Penjelasan lengkap urutan aksi aktor dan reaksi sistem pada *use case scenario* normal dan *use case scenario* alternatif mengelola data cabai dapat dilihat di lampiran A.3 pada tabel A.3 *use case scenario* mengelola data pengguna.

5. *Use Case Scenario* Mengelola Data Cabai

Use case scenario mengelola data cabai menjelaskan urutan aksi admin saat menambah, menghapus, atau mengubah data cabai yang terdapat dalam database. Urutan aksi dan reaksi sistem pada *use case scenario* mengelola data cabai diuraikan sebagai berikut:

- a. Admin memilih menu data cabai dan reaksi sistem menampilkan halaman data cabai berisi tabel jenis cabai, tombol tambah data, tombol edit data, dan tombol hapus data.
- b. Admin memilih tombol tambah data dan reaksi sistem menampilkan form pada halaman tambah data cabai dengan atribut jenis cabai dan ciri-ciri cabai dengan tombol simpan dan tombol reset.
- c. Admin memilih tombol simpan dan reaksi sistem mengalihkan halaman tambah data ke halaman daftar pengguna.

Penjelasan lengkap urutan aksi aktor dan reaksi sistem pada *use case scenario* normal dan *use case scenario* alternatif mengelola data cabai dapat dilihat di lampiran A.4 pada tabel A.4 *use case scenario* mengelola data cabai.

6. *Use Case Scenario* Mengelola Data Kab/Kota

Use case scenario mengelola data kab/kota menjelaskan urutan aksi admin saat menambah, menghapus, atau mengubah data kab/kota yang terdapat dalam database. Urutan aksi dan reaksi sistem pada *use case scenario* mengelola data kab/kota diuraikan sebagai berikut:

- a. Admin memilih menu distributor dan reaksi sistem menampilkan halaman data kab/kota dengan tabel yang memiliki atribut provinsi, kabupaten, dan *lead time*, serta memiliki tombol tambah data, tombol edit data, dan tombol hapus data.
- b. Admin memilih tombol tambah data dan reaksi sistem menampilkan form halaman tambah data kab/kota dengan atribut atribut provinsi, kabupaten, dan *lead time* yang memiliki tombol simpan dan tombol reset.
- c. Admin memilih tombol simpan dan reaksi sistem mengalihkan halaman tambah data ke halaman data kab/kota.

Penjelasan lengkap urutan aksi aktor dan reaksi sistem pada *use case scenario* normal dan *use case scenario* alternatif mengelola data kab/kota dapat dilihat di lampiran A.5 pada tabel A.5 *use case scenario* mengelola data kab/kota.

7. *Use Case Scenario* Melihat Data Produktivitas Cabai

Use case scenario melihat data produktivitas cabai menjelaskan urutan aksi petani saat melihat data produktivitas cabai yang terdapat dalam sistem. Urutan aksi dan reaksi sistem pada *use case scenario* melihat data produktivitas cabai diuraikan sebagai berikut:

- a. Admin memilih menu produksi dan reaksi sistem menampilkan halaman data produksi dengan tabel yang memiliki atribut provinsi, kabupaten, jenis cabai, tahun, dan total produksi yang tersedia tombol detail.

- b. Admin memilih tombol detail dan reaksi sistem menampilkan tabel pada halaman detail data kabupaten tertentu dengan atribut bulan dan jumlah produksi.

Penjelasan lengkap urutan aksi aktor dan reaksi sistem pada *use case scenario* normal dan *use case scenario* alternatif melihat data produktivitas cabai dapat dilihat di lampiran A.6 pada tabel A.6 *use case scenario* melihat data produktivitas cabai.

8. *Use Case Scenario* Melihat Data Konsumsi Cabai

Use case scenario melihat data konsumsi cabai menjelaskan urutan aksi petani saat melihat data konsumsi cabai yang terdapat dalam sistem. Urutan aksi dan reaksi sistem pada *use case scenario* melihat data konsumsi cabai diuraikan sebagai berikut:

- c. Admin memilih menu konsumsi dan reaksi sistem menampilkan halaman data produksi dengan tabel yang memiliki atribut provinsi, kabupaten, jenis cabai, tahun, dan total konsumsi yang tersedia tombol detail.
- d. Admin memilih tombol detail dan reaksi sistem menampilkan tabel pada halaman detail data kabupaten tertentu dengan atribut bulan dan jumlah konsumsi.

Penjelasan lengkap urutan aksi aktor dan reaksi sistem pada *use case scenario* normal dan *use case scenario* alternatif melihat data konsumsi cabai dapat dilihat di lampiran A.7 pada tabel A.7 *use case scenario* melihat data konsumsi cabai.

9. *Use Case Scenario* Melihat Data Jenis Cabai

Use case scenario melihat data jenis cabai menjelaskan urutan aksi petani saat melihat data jenis cabai yang terdapat dalam sistem. Urutan aksi dan reaksi sistem pada saat petani memilih menu data cabai dan reaksi sistem menampilkan halaman data cabai dengan tabel yang memiliki atribut nama dan ciri-cirinya.

Penjelasan lengkap urutan aksi aktor dan reaksi sistem pada *use case scenario* normal dan *use case scenario* alternatif melihat data jenis cabai dapat dilihat di lampiran A.8 pada tabel A.8 *use case scenario* melihat data jenis cabai.

10. *Use Case Scenario* Melihat Data Kab/Kota

Use case scenario melihat data jenis kab/kota menjelaskan urutan aksi petani saat melihat data jenis kab/kota yang terdapat dalam sistem. Urutan aksi dan reaksi sistem pada saat petani memilih menu distributor dan reaksi sistem menampilkan halaman data kab/kota dengan tabel yang memiliki atribut provinsi, kabupaten, dan *lead time*.

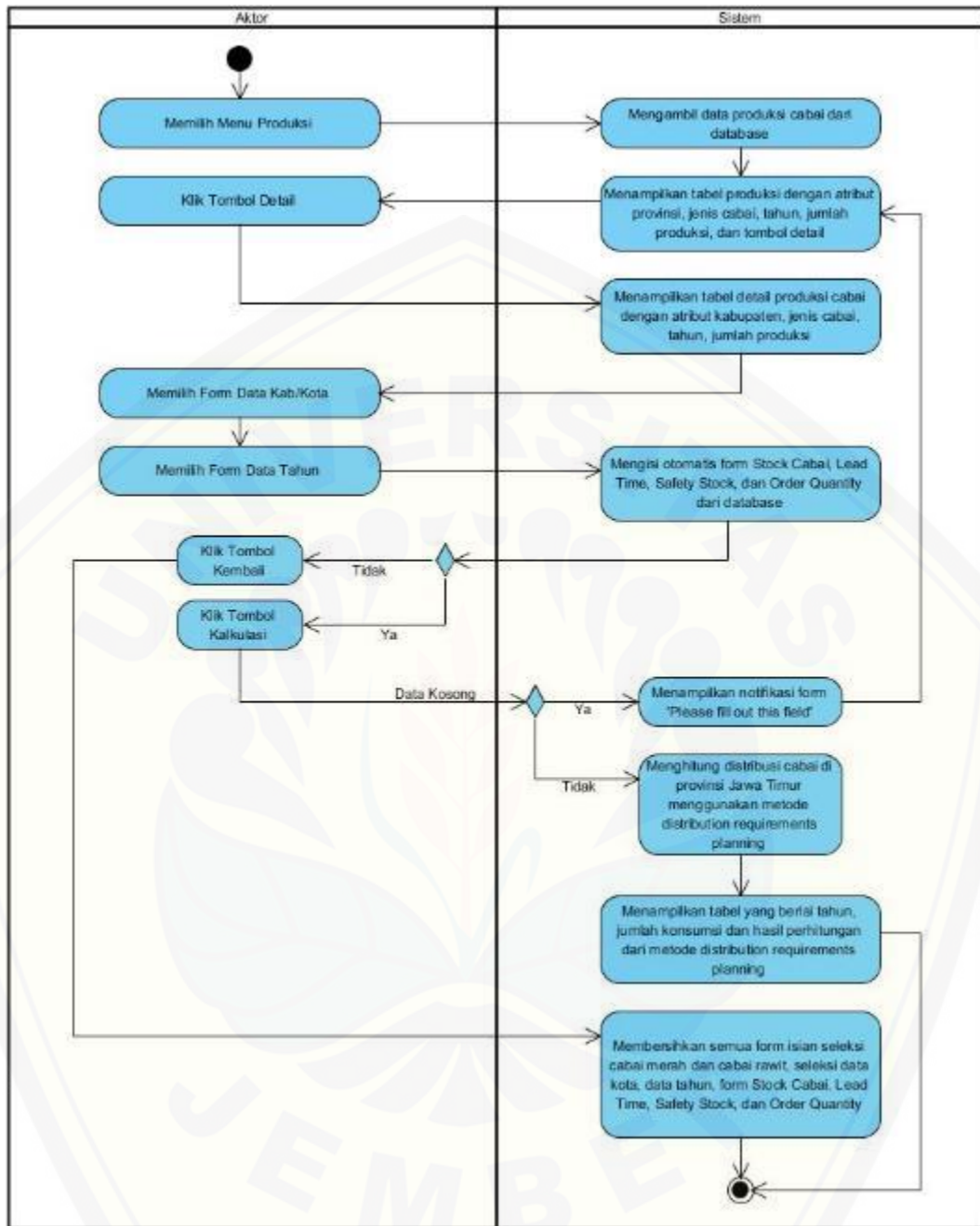
Penjelasan lengkap urutan aksi aktor dan reaksi sistem pada *use case scenario* normal dan *use case scenario* alternatif melihat data jenis kab/kota dapat dilihat di lampiran A.9 pada tabel A.9 *use case scenario* melihat data jenis kab/kota.

4.2.4. *Activity Diagram*

Activity diagram menggambarkan aliran aktivitas dalam sistem, mulai dari alir dimulai, *decision* yang terjadi dan akhir dari alir. Berikut ini *activity diagram* dari sistem informasi distribusi cabai:

1. *Activity Diagram* Kalkulasi Distribusi Cabai.

Activity ini diakses oleh pegawai digunakan untuk melakukan kalkulasi distribusi cabai besar dan rawit. *Activity Diagram* dapat dilihat pada Gambar 4.5.



Gambar 4. 5 Activity Diagram Kalkulasi Distribusi Cabai

2. Activity Diagram Mengelola Data Konsumsi Cabai

Activity Diagram mengelola data konsumsi cabai menggambarkan aliran aktivitas yang dilakukan oleh pegawai dan proses yang terjadi dalam sistem. Pengelolaan data konsumsi cabai yang dilakukan dalam sistem informasi distribusi

cabai meliputi proses melihat, menambah, mengubah, menghapus data konsumsi cabai. Aliran aktivitas admin saat mengelola data konsumsi cabai diuraikan sebagai berikut:

- a. Pegawai memilih menu data konsumsi dan reaksi sistem menampilkan halaman data cabai berisi tabel konsumsi, tombol tambah data, tombol edit data, dan tombol hapus data.
- b. Pegawai memilih tombol tambah data dan reaksi sistem menampilkan form pada halaman tambah data cabai dengan atribut provinsi, kabupaten, jenis cabai, tahun, konsumsi perkapita, serta tombol simpan dan tombol reset.
- c. Pegawai memilih tombol simpan dan reaksi sistem mengalihkan halaman tambah data ke halaman daftar konsumsi.

Gambaran lengkap aliran aktivitas pegawai dan proses yang terjadi dalam sistem dapat dilihat di lampiran B.1 pada gambar B.1 *activity diagram* mengelola data konsumsi cabai.

3. *Activity Diagram* Mengelola Data Produktivitas Cabai

Activity Diagram mengelola data produktivitas cabai menggambarkan aliran aktivitas yang dilakukan oleh pegawai dan proses yang terjadi dalam sistem. Pengelolaan data produktivitas cabai yang dilakukan dalam sistem informasi distribusi cabai meliputi proses melihat, menambah, mengubah, menghapus data produktivitas cabai. Aliran aktivitas admin saat mengelola data produktivitas cabai diuraikan sebagai berikut:

- d. Pegawai memilih menu data produksi dan reaksi sistem menampilkan halaman data cabai berisi tabel produksi, tombol tambah data, tombol edit data, dan tombol hapus data.
- e. Pegawai memilih tombol tambah data dan reaksi sistem menampilkan form pada halaman tambah data cabai dengan atribut provinsi, kabupaten, jenis cabai, tahun, dan jumlah produksi dengan tombol simpan dan tombol reset.
- f. Pegawai memilih tombol simpan dan reaksi sistem mengalihkan halaman tambah data ke halaman daftar produksi.

Gambaran lengkap aliran aktivitas pegawai dan proses yang terjadi dalam sistem dapat dilihat di lampiran B.2 pada gambar B.2 *activity diagram* mengelola data produktivitas cabai.

4. *Activity Diagram* Mengelola Data Pengguna

Activity Diagram mengelola data pengguna menggambarkan aliran aktivitas yang dilakukan oleh admin dan proses yang terjadi dalam sistem. Pengelolaan data pengguna yang dilakukan dalam sistem informasi distribusi cabai meliputi proses melihat, menambah, mengubah, menghapus data pengguna. Aliran aktivitas admin saat mengelola data pengguna diuraikan sebagai berikut:

- g. Admin memilih menu data cabai dan reaksi sistem menampilkan halaman data cabai berisi tabel jenis cabai, tombol tambah data, tombol edit data, dan tombol hapus data.
- h. Admin memilih tombol tambah data dan reaksi sistem menampilkan form pada halaman tambah data cabai dengan atribut jenis cabai dan ciri-ciri cabai dengan tombol simpan dan tombol reset.
- i. Admin memilih tombol simpan dan reaksi sistem mengalihkan halaman tambah data ke halaman daftar pengguna.

Gambaran lengkap aliran aktivitas admin dan proses yang terjadi dalam sistem dapat dilihat di lampiran B.3 pada gambar B.3 *activity diagram* mengelola data pengguna.

5. *Activity Diagram* Mengelola Data Cabai

Activity Diagram mengelola data cabai menggambarkan aliran aktivitas yang dilakukan oleh admin dan proses yang terjadi dalam sistem. Pengelolaan data cabai yang dilakukan dalam sistem informasi distribusi cabai meliputi proses melihat, menambah, mengubah, menghapus data pengguna. Aliran aktivitas admin saat mengelola data cabai diuraikan sebagai berikut:

- a. Admin memilih menu data cabai dan reaksi sistem menampilkan halaman data cabai berisi tabel jenis cabai, tombol tambah data, tombol edit data, dan tombol hapus data.
- b. Admin memilih tombol tambah data dan reaksi sistem menampilkan form pada halaman tambah data cabai dengan atribut jenis cabai dan ciri-ciri cabai dengan tombol simpan dan tombol reset.
- c. Admin memilih tombol simpan dan reaksi sistem mengalihkan halaman tambah data ke halaman daftar pengguna.

Gambaran lengkap aliran aktivitas admin dan proses yang terjadi dalam sistem dapat dilihat di lampiran B.4 pada gambar B.4 *activity diagram* mengelola data cabai.

6. *Activity Diagram* Mengelola Data Kab/Kota

Activity Diagram mengelola data kab/kota menggambarkan aliran aktivitas yang dilakukan oleh admin dan proses yang terjadi dalam sistem. Pengelolaan data cabai yang dilakukan dalam sistem informasi distribusi cabai meliputi proses melihat, menambah, mengubah, menghapus data pengguna. Aliran aktivitas admin saat mengelola data kab/kota diuraikan sebagai berikut:

- a. Admin memilih menu distributor dan reaksi sistem menampilkan halaman data kab/kota dengan tabel yang memiliki atribut provinsi, kabupaten, dan *lead time*, serta memiliki tombol tambah data, tombol edit data, dan tombol hapus data.
- b. Admin memilih tombol tambah data dan reaksi sistem menampilkan form halaman tambah data kab/kota dengan atribut provinsi, kabupaten, dan *lead time* yang memiliki tombol simpan dan tombol reset.
- c. Admin memilih tombol simpan dan reaksi sistem mengalihkan halaman tambah data ke halaman data kab/kota.

Gambaran lengkap aliran aktivitas admin dan proses yang terjadi dalam sistem dapat dilihat di lampiran B.5 pada gambar B.5 *activity diagram* mengelola data kab/kota.

7. *Activity Diagram* Melihat Data Produktivitas Cabai

Activity Diagram melihat data produktivitas cabai menggambarkan aliran aktivitas yang dilakukan oleh petani dan proses yang terjadi dalam sistem. Aliran aktivitas petani dan proses yang terjadi dalam sistem saat melihat data produktivitas cabai diuraikan sebagai berikut:

- a. Admin memilih menu produksi dan reaksi sistem menampilkan halaman data produksi dengan tabel yang memiliki atribut provinsi, kabupaten, jenis cabai, tahun, dan total produksi yang tersedia tombol detail.
- b. Admin memilih tombol detail dan reaksi sistem menampilkan tabel pada halaman detail data kabupaten tertentu dengan atribut bulan dan jumlah produksi.

Gambaran lengkap aliran aktivitas petani dan proses yang terjadi dalam sistem dapat dilihat dilampiran B.6 pada gambar B.6 *activity diagram* melihat data produktivitas cabai.

8. *Activity Diagram* Melihat Data Konsumsi Cabai

Activity Diagram melihat data konsumsi cabai menggambarkan aliran aktivitas yang dilakukan oleh petani dan proses yang terjadi dalam sistem. Aliran aktivitas petani dan proses yang terjadi dalam sistem saat melihat data konsumsi cabai diuraikan sebagai berikut:

- c. Admin memilih menu konsumsi dan reaksi sistem menampilkan halaman data konsumsi dengan tabel yang memiliki atribut provinsi, kabupaten, jenis cabai, tahun, dan total konsumsi yang tersedia tombol detail.
- d. Admin memilih tombol detail dan reaksi sistem menampilkan tabel pada halaman detail data kabupaten tertentu dengan atribut bulan dan jumlah konsumsi.

Gambaran lengkap aliran aktivitas petani dan proses yang terjadi dalam sistem dapat dilihat dilampiran B.7 pada gambar B.7 *activity diagram* melihat data konsumsi cabai.

9. *Activity Diagram* Melihat Data Jenis Cabai

Activity Diagram melihat data jenis cabai menggambarkan aliran aktivitas yang dilakukan petani saat melihat data jenis cabai yang terdapat dalam sistem. Aliran aktivitas petani dan proses yang terjadi dalam sistem saat memilih menu data cabai adalah menampilkan halaman data cabai dengan tabel yang memiliki atribut nama dan ciri-cirinya.

Gambaran lengkap aliran aktivitas petani dan proses yang terjadi dalam sistem dapat dilihat di lampiran B.8 pada gambar B.8 *activity diagram* melihat data data jenis cabai.

10. *Activity Diagram* Melihat Data Kab/Kota

Activity Diagram melihat data jenis kab/kota menggambarkan aliran aktivitas yang dilakukan petani saat melihat data jenis kab/kota yang terdapat dalam sistem. Aliran aktivitas petani dan proses yang terjadi dalam sistem saat memilih menu data cabai adalah menampilkan halaman data kab/kota dengan tabel yang memiliki atribut provinsi, kabupaten, dan *lead time*.

Gambaran lengkap aliran aktivitas petani dan proses yang terjadi dalam sistem dapat dilihat di lampiran B.9 pada gambar B.9 *activity diagram* melihat data data kab/kota.

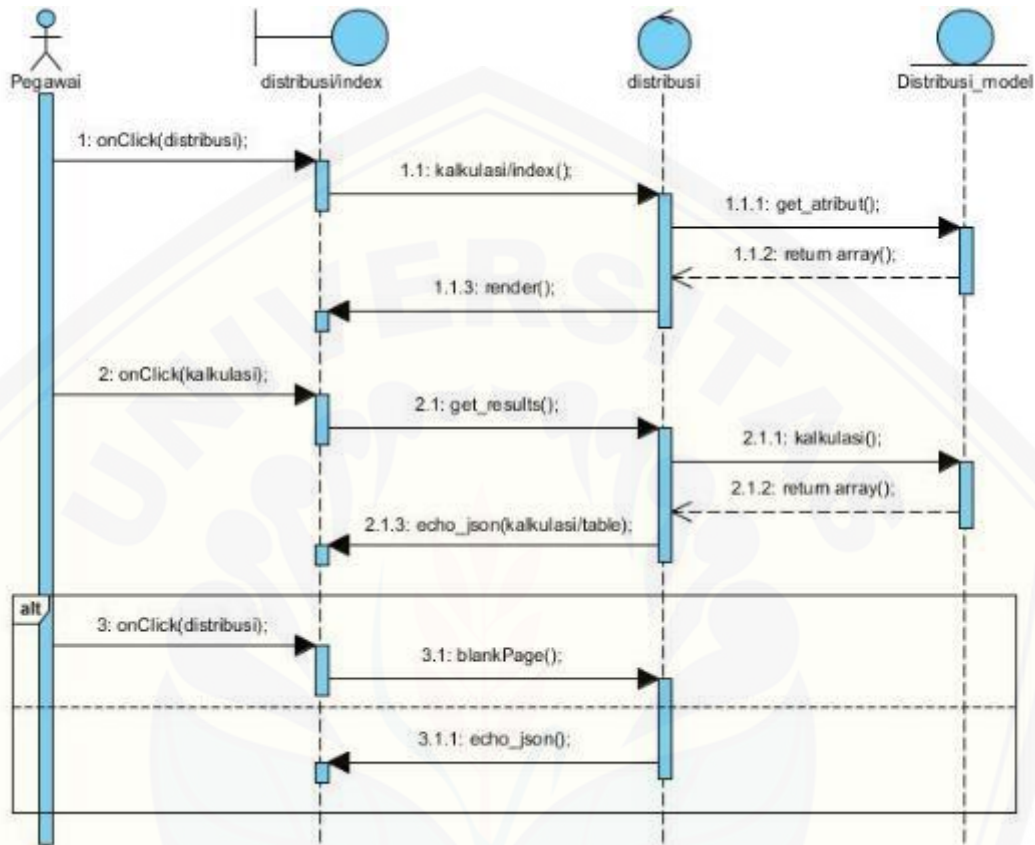
4.2.5. *Sequence Diagram*

Sequence Diagram digunakan untuk menunjukkan interaksi antar objek pada sebuah sistem berupa pesan yang digambarkan terhadap waktu. *Sequence* dari sistem informasi distribusi cabai adalah sebagai berikut:

1. *Sequence Diagram* Kalkulasi Distribusi Cabai.

Kalkulasi distribusi cabai besar dan cabai rawit diakses oleh pegawai. Menjelaskan alur kalkulasi distribusi cabai besar dan rawit. *Sequence* menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari suatu kejadian/event untuk melihat optimasi distribusi cabai besar dan cabai rawit.

Class yang berhubungan dengan *sequence diagram* ini adalah *Distribusi_controller* dan *Distribusi_model*. *Sequence diagram* untuk kalkulasi distribusi cabai besar dan cabai rawit dapat dilihat pada gambar 4.8



Gambar 4. 6 *Sequence Diagram* Kalkulasi Distribusi Cabai Jawa Timur

2. *Sequence Diagram* Mengelola Data Konsumsi Cabai

Mengelola data konsumsi cabai dilakukan oleh pegawai saat mengakses beberapa objek dan fungsi dalam sistem. *Sequence Diagram* untuk mengelola data konsumsi cabai menunjukkan interaksi antar objek pada sebuah sistem. Objek tersebut terletak dalam kode program mulai dari *class model*, *class view*, dan *class controller*. *Class* yang berhubungan dalam *sequence* mengelola data konsumsi cabai dapat dijelaskan sebagai berikut:

- a. *Class Konsumsi_model* adalah tempat penulisan kode *query* yang berfungsi untuk mengambil data konsumsi dari database.

- b. *Class Konsumsi_controller* adalah tempat membuat variabel yang akan digunakan untuk menampilkan halaman data konsumsi.
- c. *Class Konsumsi* adalah halaman tempat interaksi admin dalam mengelola data konsumsi cabai sesuai perintah yang berjalan pada *model* dan *controller*. *Sequence* untuk menggambarkan interaksi fungsi objek atau rangkaian langkah-langkah yang dilakukan sebagai respon dari suatu kejadian/event saat mengelola data konsumsi cabai dapat dilihat dilampiran C.1 pada gambar C.1 *sequence diagram* mengelola data konsumsi cabai.

3. *Sequence Diagram* Mengelola Data Produktivitas Cabai

Mengelola data produktivitas cabai dilakukan oleh pegawai saat mengakses beberapa objek dan fungsi dalam sistem. *Sequence Diagram* untuk mengelola data produktivitas cabai menunjukkan interaksi antar objek pada sebuah sistem. Objek tersebut terletak dalam kode program mulai dari *class model*, *class view*, dan *class controller*. *Class* yang berhubungan dalam *sequence* mengelola data produktivitas cabai dapat dijelaskan sebagai berikut:

- d. *Class Produksi_model* adalah tempat penulisan kode *query* yang berfungsi untuk mengambil data produksi dari database.
- e. *Class Produksi_controller* adalah tempat membuat variabel yang akan digunakan untuk menampilkan halaman data produksi.
- f. *Class Produksi* adalah halaman tempat interaksi admin dalam mengelola data produktivitas cabai sesuai perintah yang berjalan pada *model* dan *controller*. *Sequence* untuk menggambarkan interaksi fungsi objek atau rangkaian langkah-langkah yang dilakukan sebagai respon dari suatu kejadian/event saat mengelola data produktivitas cabai dapat dilihat dilampiran C.2 pada gambar C.2 *sequence diagram* mengelola data produktivitas cabai.

4. *Sequence Diagram* Mengelola Data Pengguna

Mengelola data pengguna dilakukan oleh admin saat mengakses beberapa objek dan fungsi dalam sistem. *Sequence Diagram* untuk mengelola data pengguna

menunjukkan interaksi antar objek pada sebuah sistem. Objek tersebut terletak dalam kode program mulai dari *class model*, *class view*, dan *class controller*. *Class* yang berhubungan dalam *sequence* mengelola data pengguna dapat dijelaskan sebagai berikut:

- g. *Class Users_model* adalah tempat penulisan kode *query* yang berfungsi untuk mengambil data pengguna dari database.
- h. *Class Users_controller* adalah tempat membuat variabel yang akan digunakan untuk menampilkan halaman data pengguna.
- i. *Class Users* adalah halaman tempat interaksi admin dalam mengelola data pengguna sesuai dengan perintah yang berjalan pada *model* dan *controller*.

Sequence untuk menggambarkan interaksi fungsi objek atau rangkaian langkah-langkah yang dilakukan sebagai respon dari suatu kejadian/event saat mengelola data pengguna dapat dilihat dilampiran C.3 pada gambar C.3 *sequence diagram* mengelola data pengguna.

5. *Sequence Diagram* Mengelola Data Cabai

Mengelola data cabai dilakukan oleh admin saat mengakses beberapa objek dan fungsi dalam sistem. *Sequence Diagram* untuk mengelola data cabai menunjukkan interaksi antar objek pada sebuah sistem. Objek tersebut terletak dalam kode program mulai dari *class model*, *class view*, dan *class controller*. *Class* yang berhubungan dalam *sequence* mengelola data cabai dapat dijelaskan sebagai berikut:

- a. *Class Users_model* adalah tempat penulisan kode *query* yang berfungsi untuk mengambil data cabai dari database.
- b. *Class Users_controller* adalah tempat membuat variabel yang akan digunakan untuk menampilkan halaman data cabai.
- c. *Class Users* adalah halaman tempat interaksi admin dalam mengelola data cabai sesuai dengan perintah yang berjalan pada *model* dan *controller*.

Sequence untuk menggambarkan interaksi fungsi objek atau rangkaian langkah-langkah yang dilakukan sebagai respon dari suatu kejadian/event saat

mengelola data cabai dapat dilihat dilampiran C.4 pada gambar C.4 *sequence diagram* mengelola data cabai.

6. *Sequence Diagram* Mengelola Data Kab/Kota

Mengelola data kab/kota dilakukan oleh admin saat mengakses beberapa objek dan fungsi dalam sistem. *Sequence Diagram* untuk mengelola data kab/kota menunjukkan interaksi antar objek pada sebuah sistem. Objek tersebut terletak dalam kode program mulai dari *class model*, *class view*, dan *class controller*. *Class* yang berhubungan dalam *sequence* mengelola data kab/kota dapat dijelaskan sebagai berikut:

- a. *Class Kabupaten_model* adalah tempat penulisan kode *query* yang berfungsi untuk mengambil data kabupaten dari database.
- b. *Class Kabupaten_controller* adalah tempat membuat variabel yang akan digunakan untuk menampilkan halaman data kabupaten.
- c. *Class Kabupaten* adalah halaman tempat interaksi admin dalam mengelola data kabupaten sesuai perintah yang berjalan pada *model* dan *controller*.

Sequence untuk menggambarkan interaksi fungsi objek atau rangkaian langkah-langkah yang dilakukan sebagai respon dari suatu kejadian/event saat mengelola data kab/kota dapat dilihat dilampiran C.5 pada gambar C.5 *sequence diagram* mengelola data kab/kota.

7. *Sequence Diagram* Melihat Data Produktivitas Cabai

Melihat data produktivitas cabai dilakukan oleh petani dengan mengakses beberapa objek dan fungsi dalam sistem. *Sequence Diagram* untuk melihat data produktivitas cabai menunjukkan interaksi antar objek pada sebuah sistem. Objek tersebut terletak dalam kode program mulai dari *class model*, *class view*, dan *class controller*. *Class* yang berhubungan dalam *sequence* melihat data produktivitas cabai dapat dijelaskan sebagai berikut:

- a. *Class Produksi_model* adalah tempat penulisan kode *query* yang berfungsi untuk mengambil data produksi cabai dari database.

- b. *Class Produksi_controller* adalah tempat membuat variabel yang akan digunakan untuk menampilkan halaman data produksi.
- c. *Class Produksi* adalah halaman tempat interaksi petani saat melihat data produktivitas cabai sesuai perintah yang berjalan pada *model* dan *controller*.
Sequence untuk menggambarkan interaksi fungsi objek atau rangkaian langkah-langkah yang dilakukan sebagai respon dari suatu kejadian/event saat melihat data produktivitas cabai dapat dilihat dilampiran C.6 pada gambar C.6 *sequence diagram* melihat data produktivitas cabai.

8. *Sequence Diagram* Melihat Data Konsumsi Cabai

Melihat data konsumsi cabai dilakukan oleh petani dengan mengakses beberapa objek dan fungsi dalam sistem. *Sequence Diagram* untuk melihat data konsumsi cabai menunjukkan interaksi antar objek pada sebuah sistem. Objek tersebut terletak dalam kode program mulai dari *class model*, *class view*, dan *class controller*. *Class* yang berhubungan dalam *sequence* melihat data konsumsi cabai dapat dijelaskan sebagai berikut:

- d. *Class Konsumsi_model* adalah tempat penulisan kode *query* yang berfungsi untuk mengambil data konsumsi cabai dari database.
- e. *Class Konsumsi_controller* adalah tempat membuat variabel yang akan digunakan untuk menampilkan halaman data konsumsi.
- f. *Class Konsumsi* adalah halaman tempat interaksi petani saat melihat data konsumsi cabai sesuai perintah yang berjalan pada *model* dan *controller*.
Sequence untuk menggambarkan interaksi fungsi objek atau rangkaian langkah-langkah yang dilakukan sebagai respon dari suatu kejadian/event saat melihat data konsumsi cabai dapat dilihat dilampiran C.7 pada gambar C.7 *sequence diagram* melihat data konsumsi cabai.

9. *Sequence Diagram* Melihat Data Jenis Cabai

Melihat data jenis cabai dilakukan oleh petani dengan mengakses beberapa objek dan fungsi dalam sistem. *Sequence Diagram* untuk melihat data jenis cabai

menunjukkan interaksi antar objek pada sebuah sistem. Objek tersebut terletak dalam kode program mulai dari *class model*, *class view*, dan *class controller*. *Class* yang berhubungan dalam *sequence* melihat data jenis cabai dapat dijelaskan sebagai berikut:

- a. *Class Cabai_model* adalah tempat penulisan kode *query* yang berfungsi untuk mengambil data jenis cabai dari database.
- b. *Class Cabai_controller* adalah tempat membuat variabel yang akan digunakan untuk menampilkan halaman data jenis cabai.
- c. *Class Cabai* adalah halaman tempat interaksi petani saat melihat data jenis cabai sesuai perintah yang berjalan pada *model* dan *controller*.

Sequence untuk menggambarkan interaksi fungsi objek atau rangkaian langkah-langkah yang dilakukan sebagai respon dari suatu kejadian/event saat melihat data jenis cabai dapat dilihat dilampiran C.8 pada gambar C.8 *sequence diagram* melihat data jenis cabai.

10. *Sequence Diagram* Melihat Data Kab/Kota

Melihat data kab/kota dilakukan oleh petani dengan mengakses beberapa objek dan fungsi dalam sistem. *Sequence Diagram* untuk melihat data kab/kota menunjukkan interaksi antar objek pada sebuah sistem. Objek tersebut terletak dalam kode program mulai dari *class model*, *class view*, dan *class controller*. *Class* yang berhubungan dalam *sequence* melihat data kab/kota dapat dijelaskan sebagai berikut:

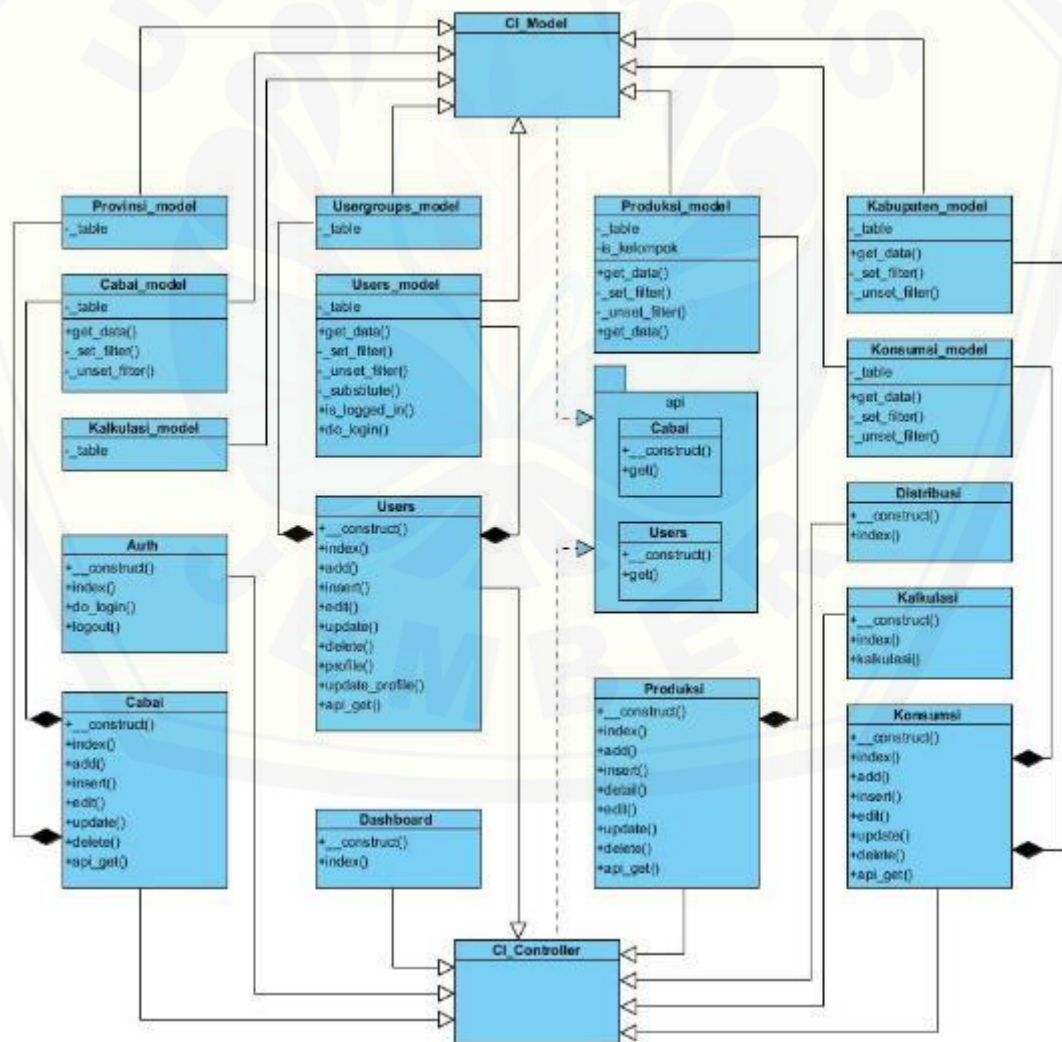
- a. *Class Kabupaten_model* adalah tempat penulisan kode *query* yang berfungsi untuk mengambil data kabupaten dari database.
- b. *Class Kabupaten_controller* adalah tempat membuat variabel yang akan digunakan untuk menampilkan halaman data kabupaten.
- c. *Class Kabupaten* adalah halaman tempat interaksi petani saat melihat data kab/kota sesuai perintah yang berjalan pada *model* dan *controller*.

Sequence untuk menggambarkan interaksi fungsi objek atau rangkaian langkah-langkah yang dilakukan sebagai respon dari suatu kejadian/event saat

melihat data kab/kota dapat dilihat dilampiran C.9 pada gambar C.9 *sequence diagram* melihat data kab/kota.

4.2.6. Class Diagram

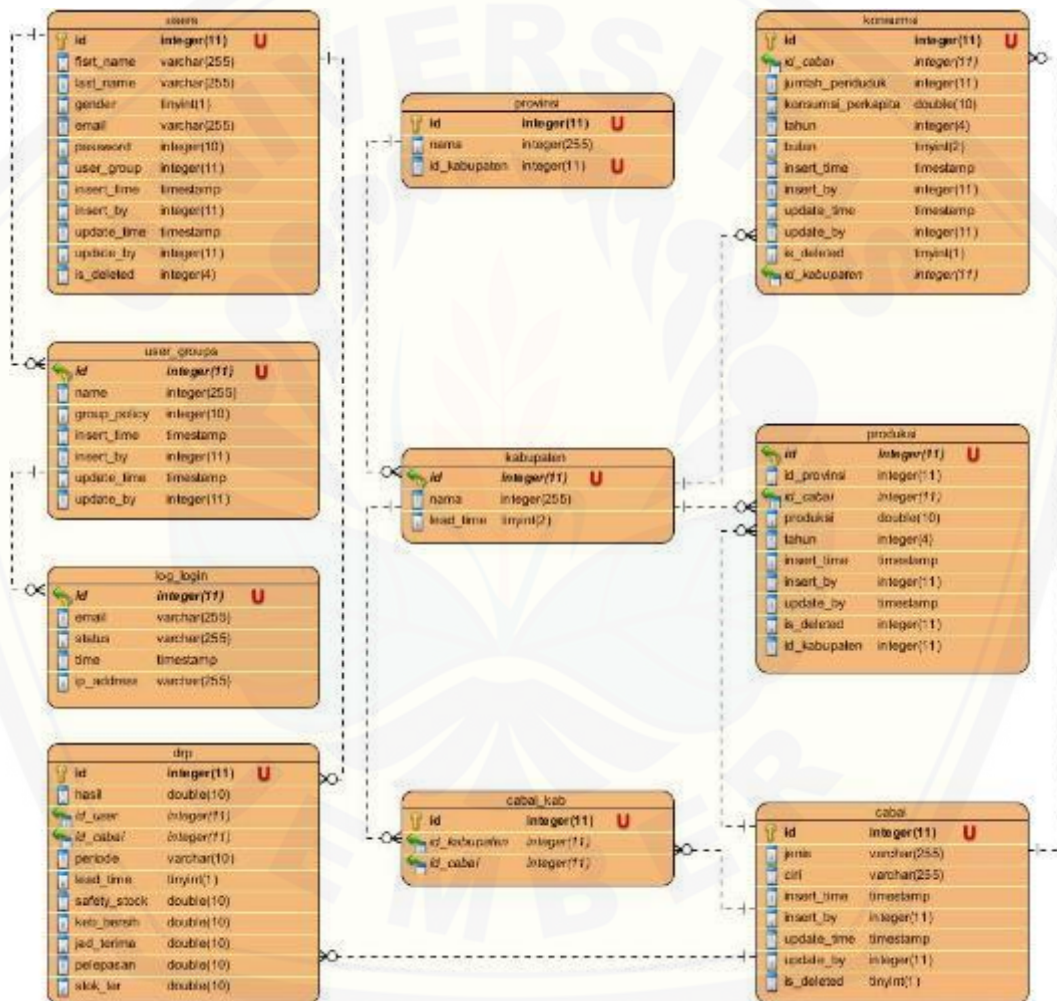
Class diagram pada sistem informasi ini terdiri dari dua paket umum yakni *model* dan *controller*. Pada *class model* terdapat beberapa class yang digunakan untuk mengakses database. Sedangkan pada *class controller* juga terdapat beberapa class yang digunakan untuk mendeklarasikan variabel untuk menampilkan sebuah halaman. *Class model* dan *controller* memiliki ketergantungan yang signifikan, dua *class* tersebut berperan menjalankan setiap interaksi dan proses yang terjadi dalam sistem. *Class Diagram* sistem informasi distribusi dapat dilihat pada gambar 4.11.



Gambar 4. 7 *Class Diagram* Sistem Informasi Distribusi Cabai

4.2.7. Entity Relationship Diagram

Entity Relationship Diagram (ERD) menggambarkan struktur database yang akan dibangun pada sistem. ERD pada pengembangan sistem informasi distribusi cabai memiliki beberapa tabel dalam database diantaranya tabel *users*, tabel *user_groups*, tabel *log_login*, tabel cabai, tabel kabupaten, tabel provinsi, tabel produksi, tabel konsumsi, tabel stok, dan tabel drp. Desain ERD pada sistem informasi distribusi cabai ini dapat dilihat pada gambar 4.12.



Gambar 4. 8 Entity Relationship Diagram Sistem Informasi Distribusi Cabai

4.3. Penulisan Kode Program

Tahap ini merupakan proses pembuatan aplikasi, yaitu merupakan proses implementasi dari hasil analisa kebutuhan yang dilakukan pada tahap perencanaan,

analisa dan desain sistem. Penulisan kode program dilakukan menggunakan bahasa pemrograman *Page Hyper Text-Processor* (PHP) dengan *framework CodeIgniter* (CI) dan menggunakan *MySQL Database*.

4.3.1. Kode Login

Penulisan kode *login* untuk mendapatkan akses dalam sistem informasi distribusi cabai berjalan pada *class Auth controller* dengan aksi menggunakan *Users_model* untuk mengambil data dari database. Penulisan kode *login* pada *model* tersebut tercantum dalam lampiran D.1 pada tabel D.1 *class model* kode *login*. Sedangkan penulisan kode *login* pada *controller* tercantum dalam lampiran D.1 pada tabel D.2 *class controller* kode *login*.

4.3.2. Kode Pengguna

Penulisan kode pengelolaan pengguna dalam sistem informasi distribusi cabai berjalan pada *class Users controller* dengan aksi menggunakan *Users_model* untuk mengambil data dari database. Penulisan kode pengelolaan pengguna pada *model* tersebut tercantum dalam lampiran D.2 pada tabel D.3 *class model* kode pengguna. Sedangkan penulisan kode pengguna pada *controller* tercantum dalam lampiran D.2 pada tabel D.4 *class controller* kode pengguna.

4.3.3. Kode Data Cabai

Penulisan kode pengelolaan data cabai dalam sistem informasi distribusi cabai berjalan pada *class Cabai controller* dengan aksi menggunakan *Cabai_model* untuk mengambil data dari database. Penulisan kode pengelolaan data cabai pada *model* tersebut tercantum dalam lampiran D.3 pada tabel D.5 *class model* kode cabai. Sedangkan penulisan kode cabai pada *controller* tercantum dalam lampiran D.3 pada tabel D.6 *class controller* kode cabai.

4.3.4. Kode Kab/Kota

Penulisan kode pengelolaan kab/kota dalam sistem informasi distribusi cabai berjalan pada *class Kabupaten controller* dengan aksi menggunakan *Kabupaten_model* untuk mengambil data dari database. Penulisan kode pengelolaan kab/kota pada *model* tersebut tercantum dalam lampiran D.2 pada tabel D.3 *class model* kode kab/kota. Sedangkan penulisan kode kab/kota pada *controller* tercantum dalam lampiran D.2 pada tabel D.4 *class controller* kode kab/kota.

4.3.5. Kode Produksi Cabai

Penulisan kode pengelolaan data produksi cabai dalam sistem informasi distribusi cabai berjalan pada *class Produksi controller* dengan aksi menggunakan *Produksi_model* untuk mengambil data dari database. Penulisan kode pengelolaan data produksi cabai pada *model* tersebut tercantum dalam lampiran D.4 pada tabel D.7 *class model* kode produksi cabai. Sedangkan penulisan kode produksi cabai pada *controller* tercantum dalam lampiran D.4 pada tabel D.8 *class controller* kode produksi cabai.

4.3.6. Kode Konsumsi Cabai

Penulisan kode pengelolaan data konsumsi cabai dalam sistem informasi distribusi cabai berjalan pada *class Konsumsi controller* dengan aksi menggunakan *Konsumsi_model* untuk mengambil data dari database. Penulisan kode pengelolaan data konsumsi cabai pada *model* tersebut tercantum dalam lampiran D.5 pada tabel D.9 *class model* kode konsumsi cabai. Sedangkan penulisan kode konsumsi cabai pada *controller* tercantum dalam lampiran D.5 pada tabel D.10 *class controller* kode konsumsi cabai.

4.3.7. Kode *Distribution Requirements Planning*

Penulisan kode kalkulasi DRP dalam sistem informasi distribusi cabai berjalan pada *class Kalkulasi controller* dengan aksi menggunakan *Kalkulasi_model* untuk mengambil data yang dibutuhkan dari database. Penulisan

kode kalkulasi DRP pada *model* tersebut tercantum dalam lampiran D.6 pada tabel D.11 *class model* kode *distribution requirements planning*. Sedangkan penulisan kode *distribution requirements planning* pada *controller* tercantum dalam lampiran D.6 pada tabel D.12 *class controller* kode *distribution requirements planning*.

4.4. Pengujian Sistem

Pengujian dan evaluasi dimaksudkan untuk mengetahui sejauh mana sistem yang dibuat ini dapat berfungsi sesuai dengan proses transaksi dalam optimasi distribusi ini nanti sesuai dengan yang diharapkan oleh pengguna. Pada tahap ini dilakukan uji coba terhadap sistem optimasi untuk menemukan kesalahan-kesalahan yang mungkin terjadi serta melakukan perbaikan untuk lebih menyempurnakan kinerja *website* tersebut.

Pengujian sistem dilakukan menggunakan metode *Black Box*. Pengujian *black box* merupakan pengujian untuk memeriksa apakah masukan dan keluaran dari sistem sesuai dengan kebutuhan fungsional atau tidak. Pengujian dilakukan dengan menjalankan aplikasi dan menganalisa input dan output yang dihasilkan aplikasi. Untuk pengujian yang berhasil tulis tanda centang ✓ dan untuk pengujian yang gagal tulis tanda silang ✘.

4.4.1. *Black Box Testing* Admin

Pengujian menggunakan *black box* untuk akses sistem yang dilakukan oleh admin mencakup ketika masuk dalam sistem, mengelola data cabai, mengelola data kabupaten, hingga mengelola data pengguna yang menjadi kebutuhan fungsional agar sistem bisa berjalan dengan baik. Pengujian *black box* untuk admin dapat dilihat pada tabel 4.11.

Tabel 4. 11 *Black Box Testing* Admin

No	Menu	Fungsi	Aksi	Hasil	Ket
1	Login (Admin)	Digunakan admin untuk masuk dalam sistem	Mengisi Username dan Password pada form login	Login berhasil dan menampilkan	[✓]

No	Menu	Fungsi	Aksi	Hasil	Ket
			kemudian menekan tombol login	halaman home admin	
			Username dan Password tidak diisi	Menampilkan notifikasi "Please fill out this field"	[✓]
2	Data Cabai (Admin)	Digunakan untuk melihat, & mengelola data cabai (menambah, mengubah, & menghapus)	Memilih menu data cabai	Menampilkan tabel data cabai dengan atribut jenis cabai dan ciri-cirinya	[✓]
			Memilih tombol tambah data cabai	Menampilkan form tambah data cabai dengan atribut jenis cabai, ciri- ciri cabai, tombol simpan dan tombol reset	[✓]
			Memilih tombol reset form data cabai	Menampilkan form tambah data cabai dengan atribut jenis cabai, ciri- ciri cabai, tombol simpan dan tombol reset	[✓]

No	Menu	Fungsi	Aksi	Hasil	Ket
			Memilih tombol simpan data cabai	Menampilkan tabel data cabai baru yang telah ditambahkan dengan atribut jenis cabai dan ciri-cirinya	[✓]
			Memilih tombol edit data cabai	Menampilkan form edit data cabai dengan rincian data yang terpilih sesuai atribut jenis cabai, ciri- ciri cabai, tombol simpan dan tombol reset	[✓]
			Memilih tombol simpan edit data cabai	Menampilkan tabel data cabai yang telah diubah dengan atribut jenis cabai dan ciri- cirinya	[✓]
			Memilih tombol hapus data cabai	Menampilkan tabel data cabai yang telah dihapus dengan atribut jenis	[✓]

No	Menu	Fungsi	Aksi	Hasil	Ket
				cabai dan ciri-cirinya	
3	Kalkulasi Distribusi (Admin)	Digunakan untuk menghitung distribusi cabai menggunakan metode <i>distribution requirement planning</i>	Mengisi form seleksi jenis cabai	Menampilkan data pilihan jenis cabai, yakni cabai besar dan cabai rawit	[✓]
			Mengisi form seleksi data kota	Menampilkan data pilihan kota yang tersedia dalam database.	[✓]
			Mengisi form data tahun produksi	Menampilkan data pilihan tahun produksi cabai yang tersedia dalam database.	[✓]
			Memilih tombol kalkulasi	Menampilkan tabel hasil perhitungan distribusi cabai menggunakan metode <i>distribution requirement planning</i>	[✓]
4		Digunakan untuk melihat,	Memilih menu data kab/kota	Menampilkan tabel data cabai	[✓]

No	Menu	Fungsi	Aksi	Hasil	Ket
	Data Kab/Kota (Admin)	& mengelola data kab/kota (menambah, mengubah, & menghapus)	Memilih tombol tambah data kab/kota	dengan atribut provinsi dan kabupaten Menampilkan form tambah data kab/kota dengan atribut provinsi, kabupaten, tombol simpan dan tombol reset	[✓]
			Memilih tombol reset form data kab/kota	Menampilkan form tambah data kab/kota dengan atribut provinsi, kabupaten, tombol simpan dan tombol reset	[✓]
			Memilih tombol simpan data kab/kota	Menampilkan tabel data kab/kota baru yang telah ditambahkan dengan atribut provinsi dan kabupaten.	[✓]

No	Menu	Fungsi	Aksi	Hasil	Ket
			Memilih tombol edit data kab/kota	Menampilkan form edit data kab/kota dengan rincian data yang terpilih sesuai atribut provinsi, kabupaten, tombol simpan dan tombol reset	[✓]
			Memilih tombol simpan edit data kab/kota	Menampilkan tabel data kab/kota yang telah diubah dengan atribut provinsi, kabupaten,	[✓]
			Memilih tombol hapus data kab/kota	Menampilkan tabel data cabai yang telah dihapus dengan atribut provinsi, kabupaten,	[✓]
5	Pengguna	Digunakan untuk melihat, & mengelola data pengguna (menambah,	Memilih menu data pengguna	Menampilkan tabel data cabai dengan atribut nama, email, hak akses, riwayat login,	[✓]

No	Menu	Fungsi	Aksi	Hasil	Ket
		mengubah, & menghapus)		dan tombol tambah data, tombol edit, tombol hapus	
			Memilih tombol tambah data pengguna	Menampilkan form tambah data pengguna dengan atribut nama, gender, email, password, hak akses, tombol simpan dan tombol reset	[✓]
			Memilih tombol reset form data pengguna	Menampilkan form kosong tambah data pengguna dengan atribut nama, gender, email, password, hak akses, tombol simpan dan tombol reset	[✓]
			Memilih tombol simpan data pengguna	Menampilkan tabel data cabai dengan atribut nama, email,	[✓]

No	Menu	Fungsi	Aksi	Hasil	Ket
				hak akses, riwayat login, dan tombol tambah data, tombol edit, tombol hapus	
			Memilih tombol edit data pengguna	Menampilkan form edit data pengguna dengan rincian data yang terpilih sesuai atribut nama, gender, email, password, hak akses, tombol simpan dan tombol reset	[✓]
			Memilih tombol simpan edit data pengguna	Menampilkan tabel data cabai dengan atribut nama, email, hak akses, riwayat login	[✓]
			Memilih tombol hapus data pengguna	Menampilkan tabel data cabai baru yang telah dihapus dengan atribut nama,	[✓]

No	Menu	Fungsi	Aksi	Hasil	Ket
				email, hak akses, riwayat login,	
6	Logout	Digunakan admin untuk keluar dari sistem	Memilih tombol logout	Logout berhasil dan menampilkan halaman login	[✓]

4.4.2. *Black Box Testing* Pegawai

Pengujian menggunakan *black box* untuk akses sistem yang dilakukan oleh admin mencakup ketika masuk dalam sistem, mengelola data cabai, mengelola data kabupaten, hingga mengelola data pengguna yang menjadi kebutuhan fungsional agar sistem bisa berjalan dengan baik. Pengujian *black box* untuk admin dapat dilihat dilampiran E.1 pada tabel E.1.

4.4.3. *Black Box Testing* Petani

Pengujian menggunakan *black box* untuk akses sistem yang dilakukan oleh petani mencakup ketika masuk dalam sistem, melihat data cabai, melihat data kabupaten, dan melihat data produktivitas cabai yang menjadi kebutuhan non-fungsional pada sistem informasi distribusi cabai. Pengujian *black box* untuk petani dapat dilihat dilampiran E.2 pada tabel E.2.

BAB 6. PENUTUP

Bab ini menjelaskan kesimpulan dan saran yang didapatkan dalam penelitian dan pengembangan sistem informasi distribusi cabai menggunakan metode *distribution requirements planning*.

6.1. Kesimpulan

Kesimpulan yang didapatkan pada penelitian ini adalah sebagai berikut:

1. Pola optimal distribusi cabai di Jawa Timur didapatkan dari hasil pengolahan data jumlah produksi dan jumlah permintaan cabai menggunakan metode perbandingan eksponensial. Data ini dijadikan acuan untuk merumuskan perhitungan menggunakan metode *distribution requirements planing*. Dari data permintaan cabai dapat diketahui rata-rata permintaan, tingkat persediaan, kebutuhan kotor, dan *safety stock* sehingga pendistribusian akan mampu berjalan dengan baik. Pola distribusi yang dihasilkan dari DRP adalah penentuan jumlah kebutuhan bersih serta jadwal pengiriman dan jumlah yang akan diterima pada tiap kota.
2. Metode *Distribution Requirements Planing* (DRP) yang diterapkan menghasilkan perhitungan rata-rata jumlah pelepasan rencana pemesanan. Hasil analisa didapatkan dari jumlah produksi sebagai olahan data persediaan cabai, sedangkan konsumsi dijadikan sebagai olahan data permintaan cabai di Jawa Timur. Salah satu kasus untuk kota Bojonegoro yang menjadi penghasil produksi cabai besar terbanyak berjumlah 1656,69 ton dari jumlah kebutuhan kotor 979,76. Kebutuhan bersih yang didapatkan dari perhitungan *safety stock* 170,36 dan jumlah persediaan sebanyak 1827,05 ton, maka hasil analisa rencana pelepasan berjumlah 548,97 ton. Sedangkan pelepasan rencana pemesanan untuk Kota Blitar sebagai penghasil produksi cabai rawit terbanyak berjumlah 6389,69 ton dengan jumlah kebutuhan kotor 3096,65 ton. Hasil nilai perhitungan dari semua kota yang tersedia akan dijadikan usulan untuk mengisi tabulasi DRP selama tahun 2015.

3. Implementasi sistem menggunakan model pengembangan *waterfall* dengan menerapkan metode *distribution requirement planning* yang digunakan untuk mendukung *supply chain management* dalam pengembangan sistem informasi untuk menghasilkan analisa distribusi cabai di Jawa Timur.

6.2. Saran

Beberapa saran berikut diharapkan dapat memberikan perbaikan dalam penelitian selanjutnya, yaitu:

1. Penggunaan objek penelitian untuk menghitung nilai *distribution requirement planning* harus memiliki data pasti terkait supplier dan distributor agar mampu mendukung *supply chain management*, sehingga perhitungan memiliki nilai akurasi yang tinggi.
2. Untuk memperoleh hasil pengoptimalan distribusi, maka proses distribusi harus dimaksimal mulai dari penjadwalan hingga barang diterima oleh konsumen, sehingga data optimal pada metode drp dapat diterapkan pada seluruh kota dengan produksi cabai besar atau cabai rawit.

DAFTAR PUSTAKA

- Abdillah, A. F. (2009). Perencanaan dan Penjadwalan Aktivitas Distribusi Hasil Perikanan Dengan Menggunakan Distribution Requirement Planning (DRP) (Studi Kasus di UD. Retro Gemilang Internasional - Sidoarjo). *Jurnal Skripsi*, 1-16.
- Bowersox, D., Closs, D., & Cooper, M. B. (2012). *Supply Chain Logistics Management*. New York: McGraw-Hill Education.
- BPS Jatim, B. (2015). *Perkembangan Indeks Harga Konsumen/Inflasi Jawa Timur*. Jawa Timur Indonesia: Badan Pusat Statistik.
- Cahyono, B. (2003). *Teknik dan Strategi Budidaya Sawi Hijau (Pai-Tsai)*. Hal 12-62. Yogyakarta: Yayasan Pustaka Nusatama.
- Departemen Pendidikan Indonesia. (2008). *Kamus Besar Bahasa Indonesia*. Jakarta: Balai Pustaka.
- Dermawan, R. (2010). Cabai Rawit Teknik Budi Daya Dan Analisis. *Usaha Tani*, 5-15.
- Gasper, V. (2004). *Production Planning and Inventory Control*. Jakarta: PT Gramedia Pustaka.
- Hasibuan, M. S. (2007). *Manajemen Sumber Daya Manusia Perusahaan*. Bandung: PT. Bumi Aksa.
- Hidayat, S., Nurhasanah, N., & Zulkifli, A. S. (2016). Perencanaan dan Penjadwalan Distribusi Pakaian Dengan Metode Distribution Resource Planning. *Jurnal*, 1-9.
- Juliani, W., & Ridwan, A. (2015). Penentuan Kebijakan Persediaan Critical Spare Part Di Dipo Bandung Pt. Kereta Api Indonesia Dengan Pendekatan Metode Continuous Review System (S, S) Untuk Menentukan Penghematan Total Biaya Persediaan. *Jurnal*, 1-10.

- Marimin. (2007). *Konsep dan Aplikasi Sistem Pendukung Keputusan*. Yogyakarta: Andi Publisher.
- Nadjib, B., & Dauzere, S. P. (2017). *Production Planning: New Lot-Sizing Models and Algorithm*. Editions Universitaires Europeennes.
- Pressman, R. S. (2001). *Software Engineering: A Practitioner's Approach, Fifth Edition*. New York: McGraw-Hill Higher Education ©2001.
- Rangkuti, F. (2011). *Riset Pemasaran*. Jakarta: PT. Gramedia Pustaka Utama.
- Setiadi. (2012). *Bertanam Cabai di Lahan dan Pot*. Jakarta: Penebar Swadaya.
- Simchi-Levi, D., & Kaminsky, P. (2003). Designing, and Managing the Supply Chain: Concepts, Strategies, and Case Studies. *Supply Chain Management*, 1-14.
- Sofyan, D. K. (2013). *Perencanaan dan Pengendalian Produksi, Edisi Pertama*. Yogyakarta: Graha Ilmu.
- Sugiyono. (2010). *Metode Penelitian Kuantitatif, Kualitatif dan R&D*. Bandung: Alfabeta.
- Taylor, & W, B. (2008). *Introduction to Manajement Science :Sains Manajemen, Edesi 8, Buku 2*. Jakarta: Salemba Empat.

LAMPIRAN

Lampiran A. Use Case Scenario

A.1. Use Case Scenario Mengelola Data Konsumsi Cabai

Pengelolaan data konsumsi cabai dilakukan oleh pegawai. Data yang digunakan berdasarkan data inputan yang telah tersedia dalam sistem informasi distribusi cabai. Penjelasan urutan aksi aktor dan reaksi sistem pada *use case scenario* normal dan *use case scenario* alternatif mengelola data konsumsi cabai dapat dilihat pada tabel A.1.

Tabel A. 1 *Use Case Scenario* Mengelola Data Konsumsi Cabai

Nomor Use case	UC-02
Nama	Mengelola Data Konsumsi Cabai
Aktor	Pegawai
<i>Precondition</i>	Pegawai memilih menu konsumsi cabai
<i>Postcondition</i>	Pegawai berhasil mengelola data konsumsi cabai
USE CASE SCENARIO NORMAL	
“Mengelola Data Konsumsi Cabai”	
Aktor	Sistem
1. Memilih Menu Konsumsi Cabai	
	2. Mengambil data konsumsi cabai dengan atribut sebagai berikut: <ul style="list-style-type: none"> - jenis cabai - jumlah penduduk - konsumsi perkapita - tahun konsumsi

Data diakses dari tabel konsumsi pada *database*, dengan *class* *konsumsi* dan *function* *get_all*.

3. Menampilkan tabel data cabai dengan atribut sebagai berikut:

- jenis cabai
 - jumlah penduduk
 - konsumsi perkapita
 - tahun konsumsi
 - tombol tambah data cabai
 - tombol edit data cabai
 - tombol hapus data cabai
-

4. Menekan Tombol Tambah Data Konsumsi Cabai

5. Menampilkan form data konsumsi dengan atribut sebagai berikut:

- pilihan jenis cabai
 - jumlah penduduk
 - konsumsi perkapita
 - tahun konsumsi
 - tombol simpan dan reset
-

6. Mengisi form data konsumsi cabai dengan kondisi sebagai berikut:

- pilihan jenis cabai tersedia dari database.
 - jumlah penduduk tidak boleh kosong.
 - konsumsi perkapita tidak boleh kosong.
-

-
- tahun konsumsi tidak boleh kosong.
-

7. Menyimpan form isian data konsumsi cabai.

8. Menyimpan data baru konsumsi cabai dengan atribut sebagai berikut:

- jenis cabai
- jumlah penduduk
- konsumsi perkapita
- tahun konsumsi

Data disimpan kedalam tabel konsumsi pada *database*.

9. Menekan Tombol Edit Data Konsumsi Cabai

10. Mengambil data konsumsi dengan atribut sebagai berikut:

- jenis cabai
- jumlah penduduk
- konsumsi perkapita
- tahun konsumsi

Data diakses dari tabel konsumsi pada *database*.

11. Menampilkan form detail data konsumsi dengan atribut sebagai berikut:

- pilihan jenis cabai
 - jumlah penduduk
 - konsumsi perkapita
 - tahun konsumsi
-

- tombol simpan dan reset

12. Mengubah form detail data konsumsi cabai

13. Menyimpan data konsumsi cabai yang telah diubah

14. Menyimpan data konsumsi cabai yang telah diubah dengan atribut sebagai berikut:

- jenis cabai
- jumlah penduduk
- konsumsi perkapita
- tahun konsumsi

Data disimpan kedalam tabel konsumsi pada *database*.

15. Menekan Tombol Delete Data Konsumsi Cabai

16. Menghapus data konsumsi cabai berdasarkan *id_konsumsi* dari tabel konsumsi pada *database*.

USE CASE SCENARIO ALTERNATIF

“Mengelola Data Konsumsi Cabai”

Jika aktor tidak mengisi form tambah data konsumsi cabai

Aktor

Sistem

7a. Menyimpan form isian kosong saat menambah data konsumsi cabai.

	8a. Menampilkan notifikasi pada form “Please fill out this form”
9a. Menyimpan form isian data baru konsumsi cabai.	
	10a. Menyimpan data baru konsumsi cabai dengan atribut sebagai berikut: <ul style="list-style-type: none"> - jenis cabai - jumlah penduduk - konsumsi perkapita - tahun konsumsi Data disimpan kedalam tabel konsumsi pada <i>database</i> .
	11a. Menampilkan tabel data cabai dengan atribut sebagai berikut: <ul style="list-style-type: none"> - nama jenis cabai - ciri-ciri cabai - tombol tambah data cabai - tombol edit data cabai - tombol hapus data cabai

A.2. *Use Case Scenario* Mengelola Data Produktivitas Cabai

Penjelasan urutan aksi aktor dan reaksi sistem pada *use case scenario* normal dan *use case scenario* alternatif mengelola data produktivitas cabai rawit dapat dilihat pada tabel A.2.

Tabel A. 2 *Use Case Scenario* Mengelola Data Produktivitas Cabai

Nomor <i>Use case</i>	UC-03
Nama	Mengelola Data Produktivitas Cabai
Aktor	Pegawai
<i>Precondition</i>	Pegawai memilih menu produksi cabai

Postcondition

Pegawai berhasil mengelola data produktivitas cabai

USE CASE SCENARIO NORMAL

“Mengelola Data Produksi Cabai”

Aktor

Sistem

1. Memilih Menu Produksi Cabai

2. Mengambil data produksi cabai dengan atribut sebagai berikut:

- provinsi
- jenis cabai
- tahun produksi
- jumlah produksi

Data diakses dari tabel produksi pada *database*, dengan *class produksi* dan *function get_data*.

3. Menampilkan tabel data cabai dengan atribut sebagai berikut:

- provinsi
- jenis cabai
- tahun produksi
- jumlah produksi
- tombol tambah data produksi
- tombol detail data produksi

4. Menekan Tombol Tambah Data Produksi Cabai

5. Menampilkan form data produksi dengan atribut sebagai berikut:

- pilihan provinsi
- pilihan kabupaten
- pilihan jenis cabai

-
- tahun produksi
 - jumlah produksi
 - tombol simpan dan reset
-

6. Mengisi form data produksi cabai dengan kondisi sebagai berikut:

- provinsi terpilih dan tersedia otomatis dalam sistem
 - pilihan kabupaten tersedia dari *database*
 - pilihan jenis cabai tersedia dari *database*
 - tahun produksi tersedia dari *database*
 - jumlah produksi tidak boleh kosong
-

7. Menyimpan form isian data produksi cabai.

8. Menyimpan data baru produksi cabai dengan atribut sebagai berikut:

- pilihan provinsi
- pilihan kabupaten
- pilihan jenis cabai
- tahun produksi
- jumlah produksi

Data disimpan kedalam tabel produksi pada *database*.

9. Menekan Tombol Detail Data
Produksi Cabai

10. Mengambil data produksi dengan atribut sebagai berikut:

- pilihan provinsi
- pilihan kabupaten
- pilihan jenis cabai
- tahun produksi
- jumlah produksi

Data diakses dari tabel produksi pada *database*.

11. Menampilkan tabel detail data produksi dengan atribut sebagai berikut:

- jenis cabai
 - tahun produksi
 - kabupaten
 - jumlah produksi
 - tombol edit dan hapus
-

12. Menekan Tombol Ubah Data
Produksi Cabai

13. Mengambil data produksi dengan atribut sebagai berikut:

- pilihan provinsi
- pilihan kabupaten
- pilihan jenis cabai
- tahun produksi
- jumlah produksi

Data diakses dari tabel produksi pada *database*.

-
14. Menampilkan form ubah data produksi dengan atribut sebagai berikut:
- pilihan provinsi
 - pilihan kabupaten
 - pilihan jenis cabai
 - tahun produksi
 - jumlah produksi
 - tombol simpan dan reset
-
15. Mengubah form detail data produksi cabai
-
16. Menyimpan data produksi cabai yang telah diubah
-
17. Menyimpan data produksi cabai yang telah diubah dengan atribut sebagai berikut:
- pilihan provinsi
 - pilihan kabupaten
 - pilihan jenis cabai
 - tahun produksi
 - jumlah produksi
- Data disimpan kedalam tabel produksi pada *database*.
-
18. Menekan Tombol Delete Data Produksi Cabai
-
19. Menghapus data produksi cabai berdasarkan *id_produk*si dari tabel produksi pada *database*.

USE CASE SCENARIO ALTERNATIF**“Mengelola Data Produksi Cabai”**

Jika aktor tidak mengisi form tambah data produksi cabai

Aktor	Sistem
7a. Menyimpan form isian kosong saat menambah data produksi cabai.	
	8a. Menampilkan notifikasi pada form “Please fill out this form”
9a. Menyimpan form isian data baru produksi cabai.	
	10a. Menyimpan data baru produksi cabai dengan atribut sebagai berikut: <ul style="list-style-type: none"> - pilihan provinsi - pilihan kabupaten - pilihan jenis cabai - tahun produksi - jumlah produksi Data disimpan kedalam tabel produksi pada <i>database</i> .
	11a. Menampilkan tabel detail data produksi cabai dengan atribut sebagai berikut: <ul style="list-style-type: none"> - pilihan provinsi - pilihan kabupaten - pilihan jenis cabai - tahun produksi - jumlah produksi - tombol simpan dan reset

A.3. *Use Case Scenario* Mengelola Data Pengguna

Penjelasan urutan aksi aktor dan reaksi sistem pada *use case scenario* normal dan *use case scenario* alternatif *use case scenario* mengelola data pengguna dapat dilihat pada Tabel A.3.

Tabel A. 3 *Use Case Scenario* Mengelola Data Pengguna

Nomor Use case	UC-09
Nama	Mengelola Data Pengguna
Aktor	Admin
<i>Precondition</i>	Admin memilih menu pengguna
<i>Postcondition</i>	Admin berhasil Mengelola Data Pengguna
USE CASE SCENARIO NORMAL	
“Mengelola Data Pengguna”	
Aktor	Sistem
1. Memilih Menu Pengguna	
	2. Mengambil data pengguna dengan atribut sebagai berikut: <ul style="list-style-type: none"> - nama - email - hak akses - riwayat login Data diakses dari tabel <i>Users</i> pada <i>database</i> .
	3. Menampilkan tabel data cabai dengan atribut sebagai berikut: <ul style="list-style-type: none"> - nama - email - hak akses - riwayat login - tombol tambah data

-
- tombol edit data
 - tombol hapus data
-
4. Menekan Tombol Tambah Pengguna
-
5. Menampilkan form tambah data pengguna dengan atribut sebagai berikut:
- nama
 - pilihan jenis kelamin
 - email
 - password
 - pilihan hak akses
 - tombol simpan dan reset
-
6. Mengisi form tambah data pengguna cabai dengan kondisi sebagai berikut:
- nama depan dan nama belakang tidak boleh kosong
 - pilihan jenis kelamin tersedia dari *database*
 - email diisi sesuai dengan aturan penulisan email dan tidak boleh kosong
 - password harus kuat dan tidak boleh kosong
 - pilihan hak akses dalam sistem disediakan dalam *database*
-

7. Menyimpan form isian tambah data pengguna

8. Menyimpan data pengguna dengan atribut sebagai berikut:

- nama
- pilihan jenis kelamin
- email
- password
- pilihan hak akses

Data disimpan kedalam tabel *Users* pada *database*.

9. Menekan Tombol Ubah Data Pengguna

10. Mengambil data pengguna dengan atribut sebagai berikut:

- nama
- email
- hak akses
- riwayat login

Data diakses dari tabel *Users* pada *database*.

11. Menampilkan form ubah data pengguna dengan atribut sebagai berikut:

- nama
 - email
 - hak akses
 - riwayat login
 - tombol tambah data
 - tombol edit data
-

	- tombol hapus data
12. Mengubah form detail data pengguna	
13. Menyimpan data pengguna yang telah diubah	
	14. Menyimpan data pengguna yang telah diubah dengan atribut sebagai berikut: <ul style="list-style-type: none"> - nama - pilihan jenis kelamin - email - password - pilihan hak akses Data disimpan kedalam tabel <i>Users</i> pada <i>database</i> .
15. Menekan Tombol Delete Data Pengguna	
	16. Menghapus data produksi cabai berdasarkan <i>id_users</i> dari tabel <i>Users</i> pada <i>database</i> .

USE CASE SCENARIO ALTERNATIF

“Mengelola Data Pengguna”

Jika aktor tidak mengisi form tambah data pengguna

Aktor	Sistem
12a. Menyimpan form isian kosong saat menambah data pengguna	
	13a. Menampilkan notifikasi pada form “Please fill out this form”
14a. Menyimpan form isian data pengguna baru.	

15a. Menyimpan data baru produksi cabai dengan atribut sebagai berikut:

- nama
- pilihan jenis kelamin
- email
- password
- pilihan hak akses

Data disimpan kedalam tabel *Users* pada *database*.

16a. Menampilkan tabel data pengguna dengan atribut sebagai berikut:

- nama
 - email
 - hak akses
 - riwayat login
 - tombol tambah data
 - tombol edit data
 - tombol hapus data
-

A.4. *Use Case Scenario* Mengelola Data Cabai

Penjelasan urutan aksi aktor dan reaksi sistem pada *use case scenario* normal dan *use case scenario* alternatif *use case scenario* mengelola data cabai dapat dilihat pada Tabel A.4.

Tabel A. 4 *Use Case Scenario* Mengelola Data Cabai

Nomor <i>Use case</i>	UC-10
Nama	Mengelola Data Cabai
Aktor	Admin
<i>Precondition</i>	Admin memilih menu cabai
<i>Postcondition</i>	Admin berhasil Mengelola Data Cabai

USE CASE SCENARIO NORMAL**“Mengelola Data Cabai”**

Aktor	Sistem
1. Memilih Menu Cabai	
	2. Mengambil data cabai dengan atribut sebagai berikut: <ul style="list-style-type: none"> - jenis cabai - ciri-ciri Data diakses dari tabel <i>Cabai</i> pada <i>database</i> .
	3. Menampilkan tabel data cabai dengan atribut sebagai berikut: <ul style="list-style-type: none"> - jenis cabai - ciri-ciri - tombol tambah data - tombol edit data - tombol hapus data
4. Menekan Tombol Tambah Data Cabai	
	5. Menampilkan form tambah data cabai dengan atribut sebagai berikut: <ul style="list-style-type: none"> - jenis cabai - ciri-ciri - tombol simpan dan reset
6. Mengisi form tambah data cabai dengan kondisi sebagai berikut:	<ul style="list-style-type: none"> - nama jenis cabai tidak boleh kosong

-
- ciri-ciri cabai hanya sebagai pelengkap, tapi tidak boleh kosong
-

7. Menyimpan form isian tambah data cabai

8. Menyimpan data cabai dengan atribut sebagai berikut:

- jenis cabai
- ciri-ciri

Data disimpan kedalam tabel *Cabai* pada *database*.

9. Menekan Tombol Ubah Data Cabai

10. Mengambil data cabai dengan atribut sebagai berikut:

- jenis cabai
- ciri-ciri

Data diakses dari tabel *Cabai* pada *database*.

11. Menampilkan form ubah data cabai dengan atribut sebagai berikut:

- jenis cabai
 - ciri-ciri
 - tombol simpan dan reset
-

12. Mengubah form detail data cabai

13. Menyimpan data cabai yang telah diubah

14. Menyimpan data cabai yang telah diubah dengan atribut sebagai berikut:

- jenis cabai
- ciri-ciri

Data disimpan kedalam tabel *Cabai* pada *database*.

15. Menekan Tombol Delete Data Cabai

16. Menghapus data produksi cabai berdasarkan *id_cabai* dari tabel *Cabai* pada *database*.

USE CASE SCENARIO ALTERNATIF

“Mengelola Data Cabai”

Jika aktor tidak mengisi form tambah data cabai

Aktor

Sistem

7a. Menyimpan form isian kosong saat menambah data cabai

8a. Menampilkan notifikasi pada form “Please fill out this form”

9a. Menyimpan form isian data cabai baru.

10a. Menyimpan data baru produksi cabai dengan atribut sebagai berikut:

- jenis cabai
- ciri-ciri

Data disimpan kedalam tabel *Cabai* pada *database*.

11a. Menampilkan tabel data cabai dengan atribut sebagai berikut:

- jenis cabai
- ciri-ciri
- tombol tambah data
- tombol edit data
- tombol hapus data

A.5. *Use Case Scenario* Mengelola Data Kab/Kota

Penjelasan urutan aksi aktor dan reaksi sistem pada *use case scenario* normal dan *use case scenario* alternatif *use case scenario* mengelola data kab/kota dapat dilihat pada Tabel A.5.

Tabel A. 5 *Use Case Scenario* Mengelola Data Kab/Kota

Nomor Use case	UC-11
Nama	Mengelola Data Kab/Kota
Aktor	Admin
<i>Precondition</i>	Admin memilih menu kab/kota
<i>Postcondition</i>	Admin berhasil Mengelola Data Kab/kota
USE CASE SCENARIO NORMAL	
“Mengelola Data Kab/kota”	
Aktor	Sistem
1. Memilih Menu Kab/kota	2. Mengambil data kab/kota dengan atribut sebagai berikut: <ul style="list-style-type: none"> - provinsi - kabupaten Data diakses dari tabel <i>kabupaten</i> pada <i>database</i> .

-
3. Menampilkan tabel data kab/kota dengan atribut sebagai berikut:
 - provinsi
 - kabupaten
 - tombol tambah data
 - tombol edit data
 - tombol hapus data
-
4. Menekan Tombol Tambah Data Kab/kota
-
5. Menampilkan form tambah data kab/kota dengan atribut sebagai berikut:
 - provinsi
 - kabupaten
 - tombol simpan dan reset
-
6. Mengisi form tambah data kab/kota dengan kondisi sebagai berikut:
 - provinsi tersedia otomatis dalam database
 - kabupaten diisi manual dan tidak boleh kosong
-
7. Menyimpan form isian tambah data kab/kota
-
8. Menyimpan data kab/kota dengan atribut sebagai berikut:
 - provinsi
 - kabupatenData disimpan kedalam tabel *kabupaten* pada *database*.
-

-
9. Menekan Tombol Ubah Data
Kab/kota
-
10. Mengambil data kab/kota dengan atribut sebagai berikut:
- provinsi
 - kabupaten
- Data diakses dari tabel *kabupaten* pada *database*.
-
11. Menampilkan form ubah data kab/kota dengan atribut sebagai berikut:
- provinsi
 - kabupaten
 - tombol simpan dan reset
-
12. Mengubah form detail data kab/kota
-
13. Menyimpan data kab/kota yang telah diubah
-
14. Menyimpan data kab/kota yang telah diubah dengan atribut sebagai berikut:
- provinsi
 - kabupaten
- Data disimpan kedalam tabel *kabupaten* pada *database*.
-
15. Menekan Tombol Delete Data
Kab/kota
-
16. Menghapus data kab/kota berdasarkan *id_kabupaten* dari tabel *Kabupaten* pada *database*.
-

USE CASE SCENARIO ALTERNATIF	
“Mengelola Data Kab/kota”	
Jika aktor tidak mengisi form tambah data kab/kota	
Aktor	Sistem
7a. Menyimpan form isian kosong saat menambah data kab/kota	8a. Menampilkan notifikasi pada form “Please fill out this form”
9a. Menyimpan form isian data kab/kota baru.	10a. Menyimpan data baru produksi kab/kota dengan atribut sebagai berikut: <ul style="list-style-type: none"> - provinsi - kabupaten Data disimpan kedalam tabel <i>kabupaten</i> pada <i>database</i> .
	11a. Menampilkan tabel data kab/kota dengan atribut sebagai berikut: <ul style="list-style-type: none"> - provinsi - kabupaten - tombol tambah data - tombol edit data - tombol hapus data

A.6. *Use Case Scenario* Melihat Data Produktivitas Cabai

Penjelasan urutan aksi aktor dan reaksi sistem pada *use case scenario* normal dan *use case scenario* alternatif *use case scenario* melihat data produktivitas cabai dapat dilihat pada Tabel A.6.

Tabel A. 6 *Use Case Scenario* Melihat Data Produktivitas Cabai

Nomor Use case	UC-04
Nama	Mengelola Data Produktivitas Cabai
Aktor	Petani
<i>Precondition</i>	Petani memilih menu produksi cabai
<i>Postcondition</i>	Petani berhasil mengelola data produktivitas cabai

USE CASE SCENARIO NORMAL
“Melihat Data Produksi Cabai”

Aktor	Sistem
1. Memilih Menu Produksi Cabai	2. Mengambil data produksi cabai dengan atribut sebagai berikut: <ul style="list-style-type: none"> - provinsi - jenis cabai - tahun produksi - jumlah produksi Data diakses dari tabel produksi pada <i>database</i> , dengan <i>class produksi</i> dan <i>function get_data</i> .
4. Menekan Tombol Detail Data Produksi Cabai	3. Menampilkan tabel data cabai dengan atribut sebagai berikut: <ul style="list-style-type: none"> - provinsi - jenis cabai - tahun produksi - jumlah produksi - tombol tambah data produksi - tombol detail data produksi

5. Mengambil data produksi dengan atribut sebagai berikut:

- pilihan provinsi
- pilihan kabupaten
- pilihan jenis cabai
- tahun produksi
- jumlah produksi

Data diakses dari tabel produksi pada *database*.

6. Menampilkan tabel detail data produksi dengan atribut sebagai berikut:

- jenis cabai
- tahun produksi
- kabupaten
- jumlah produksi

A.7. *Use Case Scenario* Melihat Data Konsumsi Cabai

Penjelasan urutan aksi aktor dan reaksi sistem pada *use case scenario* normal dan *use case scenario* alternatif *use case scenario* melihat data konsumsi cabai dapat dilihat pada Tabel A.7.

Tabel A. 7 *Use Case Scenario* Melihat Data Konsumsi Cabai

Nomor Use case	UC-05
Nama	Melihat Data Konsumsi Cabai
Aktor	Petani
<i>Precondition</i>	Petani memilih menu konsumsi cabai
<i>Postcondition</i>	Petani berhasil melihat data konsumsi cabai
USE CASE SCENARIO NORMAL	
“Melihat Data Konsumsi Cabai”	

Aktor	Sistem
1. Memilih Menu Konsumsi Cabai	2. Mengambil data produksi cabai dengan atribut sebagai berikut: <ul style="list-style-type: none"> - provinsi - jenis cabai - tahun konsumsi - jumlah konsumsi Data diakses dari tabel konsumsi pada <i>database</i> , dengan <i>class produksi</i> dan <i>function get_data</i> .
	3. Menampilkan tabel data cabai dengan atribut sebagai berikut: <ul style="list-style-type: none"> - provinsi - jenis cabai - tahun konsumsi - jumlah konsumsi
4. Menekan Tombol Detail Data Konsumsi Cabai	5. Mengambil data produksi dengan atribut sebagai berikut: <ul style="list-style-type: none"> - pilihan provinsi - pilihan kabupaten - pilihan jenis cabai - tahun konsumsi - jumlah konsumsi Data diakses dari tabel produksi pada <i>database</i> .

-
6. Menampilkan tabel detail data produksi dengan atribut sebagai berikut:
 - jenis cabai
 - tahun konsumsi
 - kabupaten
 - jumlah konsumsi
-

A.8. *Use Case Scenario* Melihat Data Jenis Cabai

Penjelasan urutan aksi aktor dan reaksi sistem pada *use case scenario* normal dan *use case scenario* alternatif *use case scenario* melihat data jenis cabai dapat dilihat pada Tabel A.8.

Tabel A. 8 *Use Case Scenario* Melihat Data Jenis Cabai

Nomor Use case	UC-06
Nama	Melihat Data Cabai
Aktor	Petani
<i>Precondition</i>	Petani memilih menu cabai
<i>Postcondition</i>	Petani berhasil Melihat Data Cabai
USE CASE SCENARIO NORMAL	
“Melihat Data Cabai”	
Aktor	Sistem
1. Memilih Menu Cabai	
	2. Mengambil data cabai dengan atribut sebagai berikut: <ul style="list-style-type: none"> - jenis cabai - ciri-ciri Data diakses dari tabel <i>Cabai</i> pada <i>database</i> .
	3. Menampilkan tabel data cabai dengan atribut sebagai berikut:

- jenis cabai
- ciri-ciri

A.9. *Use Case Scenario* Melihat Data Kab/Kota

Penjelasan urutan aksi aktor dan reaksi sistem pada *use case scenario* normal dan *use case scenario* alternatif *use case scenario* melihat data kab/kota dapat dilihat pada Tabel A.9.

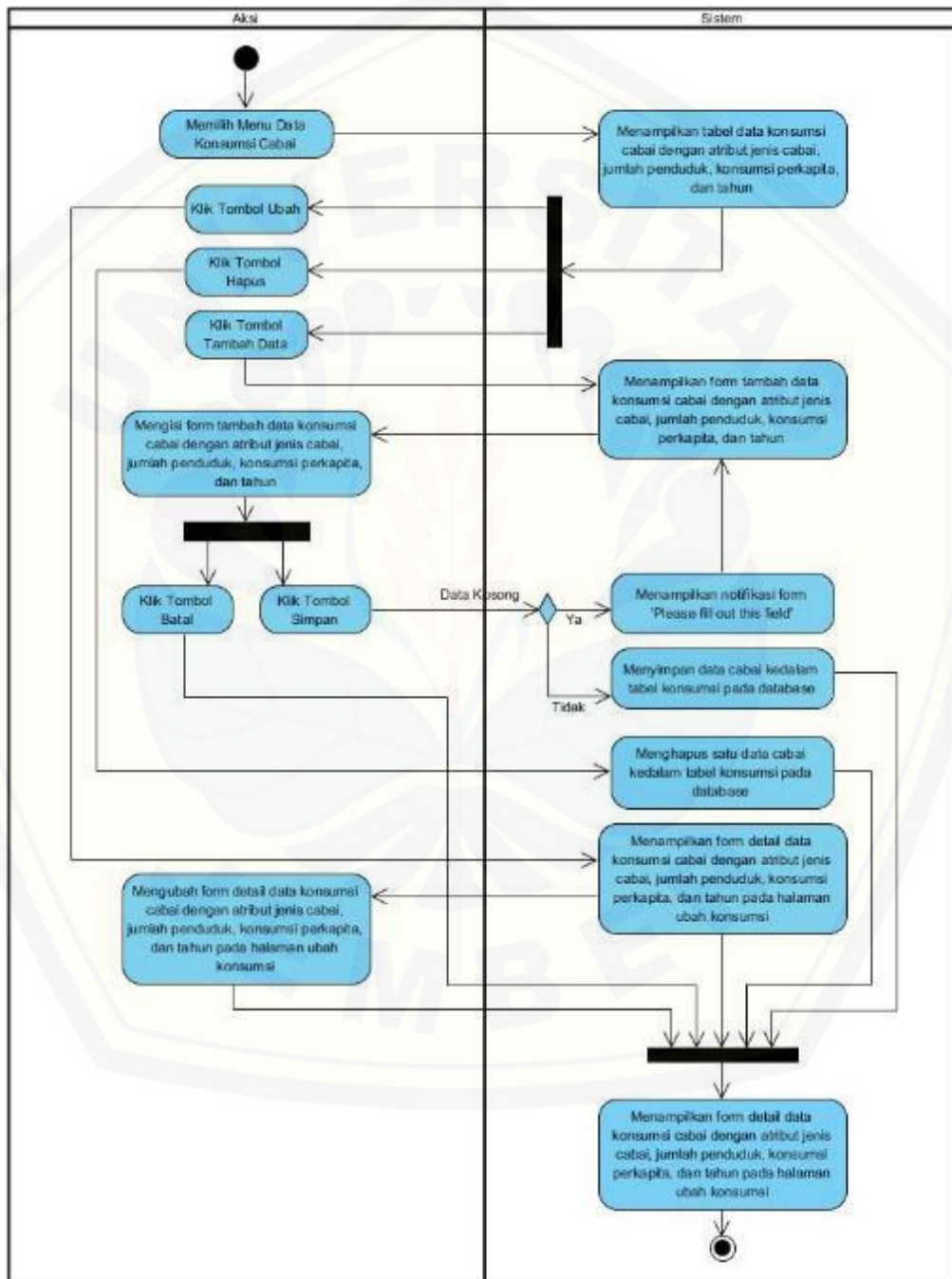
Tabel A. 9 *Use Case Scenario* Melihat Data Kab/Kota

Nomor Use case	UC-08
Nama	Melihat Data Kab/Kota
Aktor	Petani
<i>Precondition</i>	Petani memilih menu kab/kota
<i>Postcondition</i>	Petani berhasil Melihat Data Kab/kota
USE CASE SCENARIO NORMAL	
“Melihat Data Kab/kota”	
Aktor	Sistem
1. Memilih Menu Kab/kota	
	2. Mengambil data kab/kota dengan atribut sebagai berikut: <ul style="list-style-type: none"> - provinsi - kabupaten Data diakses dari tabel <i>kabupaten</i> pada <i>database</i> .
	3. Menampilkan tabel data kab/kota dengan atribut sebagai berikut: <ul style="list-style-type: none"> - provinsi - kabupaten

Lampiran B. Activity Diagram

B.1. Activity Diagram Mengelola Data Konsumsi Cabai

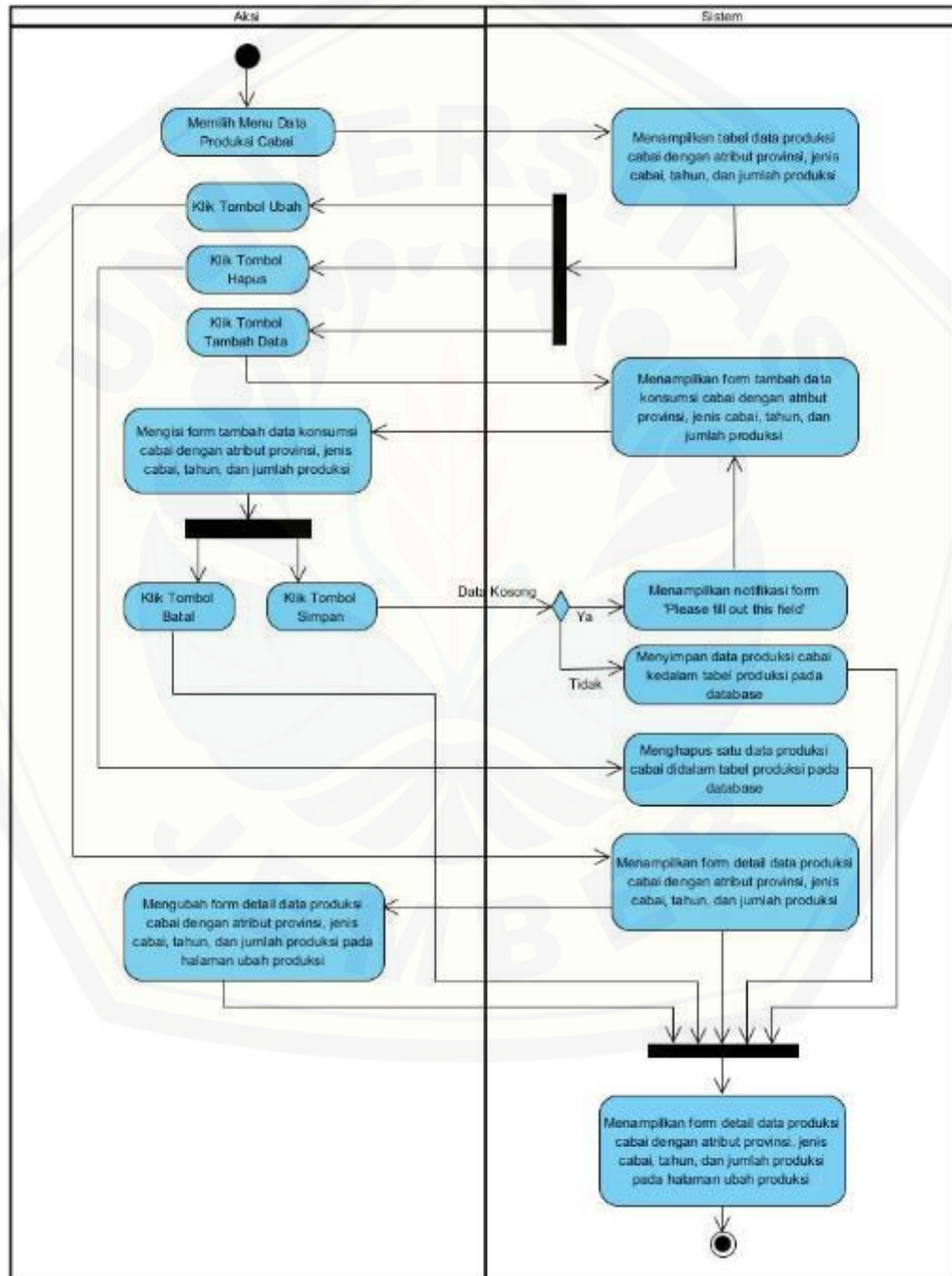
Activity ini diakses oleh pegawai digunakan untuk mengelola data konsumsi cabai di Provinsi Jawa Timur. Activity Diagram dapat dilihat pada gambar B.1.



Gambar B. 1 Activity Diagram Mengelola Data Konsumsi Cabai

B.2. Activity Diagram Mengelola Data Produktivitas Cabai

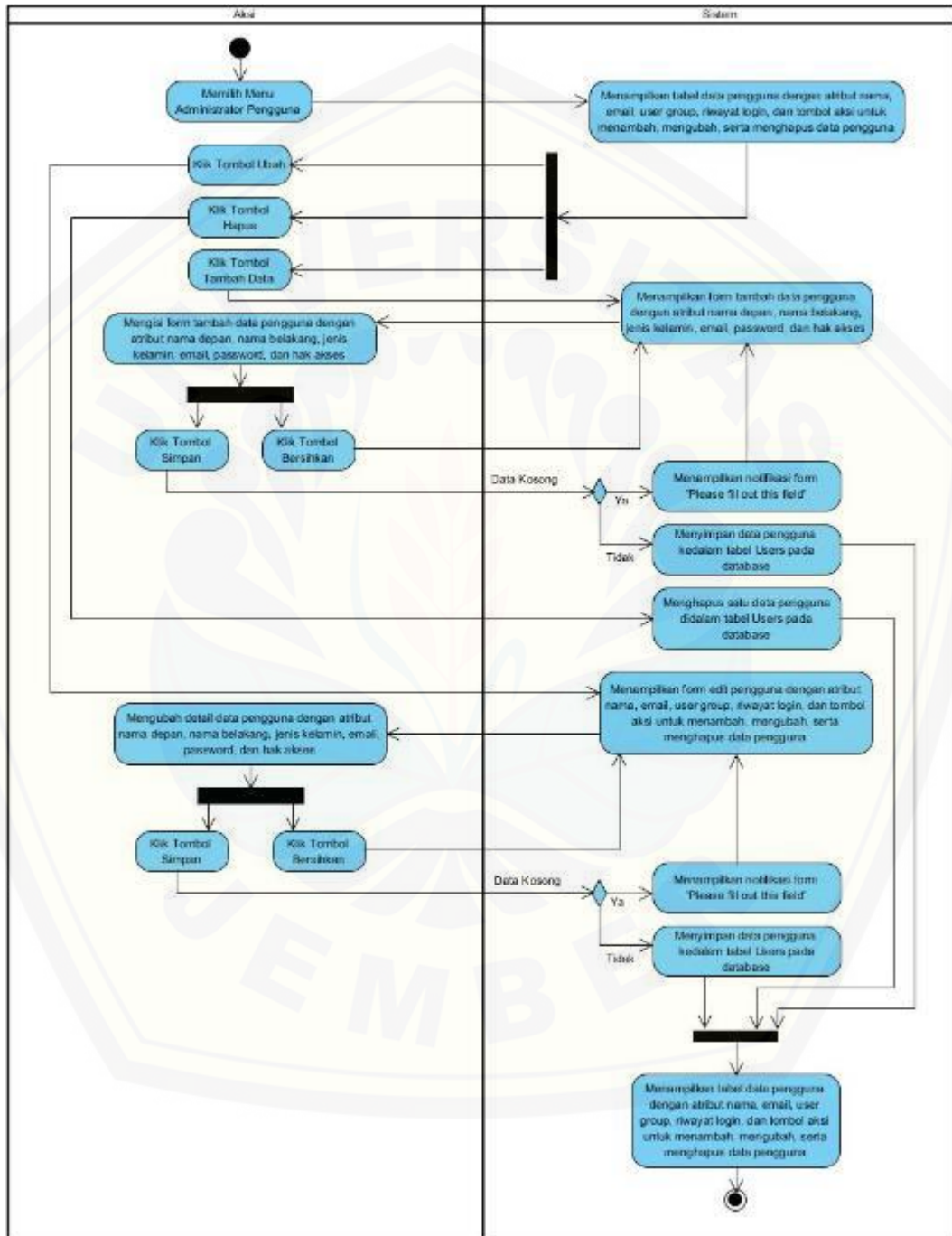
Activity ini diakses oleh pegawai digunakan untuk mengelola data produktivitas cabai rawit di Provinsi Jawa Timur. Activity Diagram dapat dilihat pada gambar B.2.



Gambar B. 2 Activity Diagram Mengelola Data Produktivitas Cabai

B.3. Activity Diagram Mengelola Data Pengguna

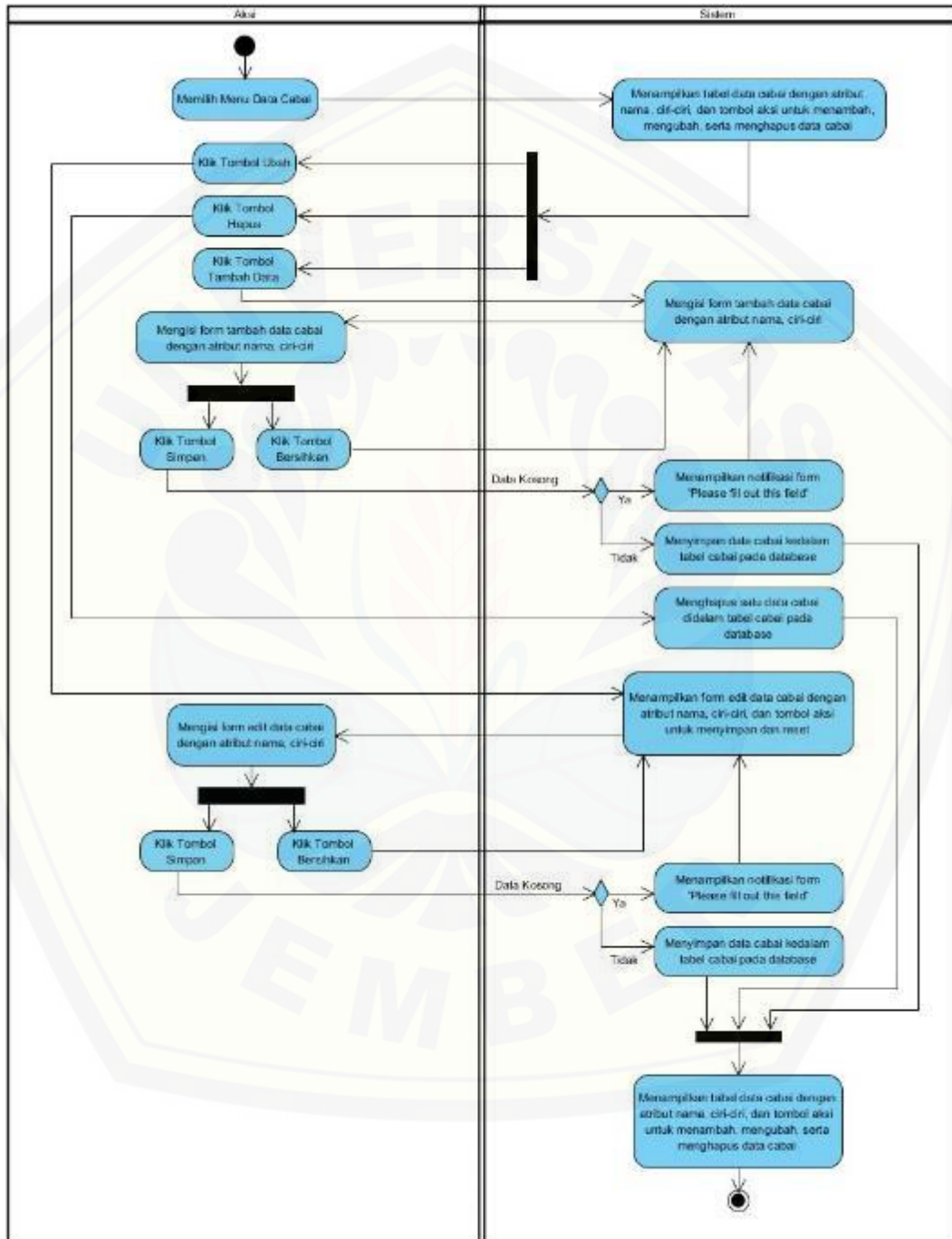
Activity ini diakses oleh admin digunakan untuk mengelola data pengguna dalam sistem distribusi cabai. Activity Diagram dapat dilihat pada gambar B.3.



Gambar B. 3 Activity Diagram Mengelola Data Pengguna

B.4. Activity Diagram Mengelola Data Cabai

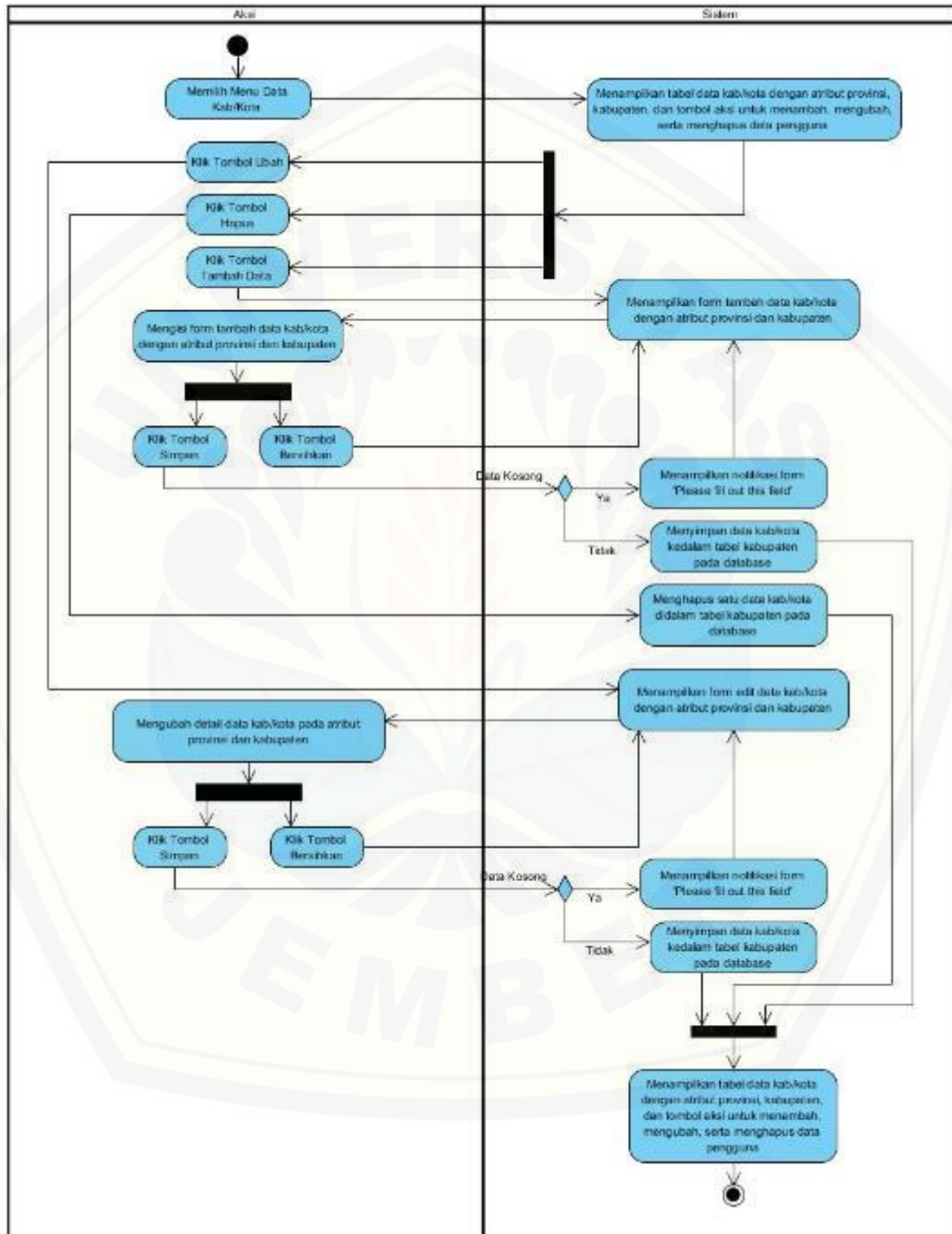
Activity ini diakses oleh admin digunakan untuk mengelola data cabai dalam sistem distribusi cabai. Activity Diagram dapat dilihat pada gambar B.4.



Gambar B. 4 Activity Diagram Mengelola Data Cabai

B.5. Activity Diagram Mengelola Data Kab/Kota

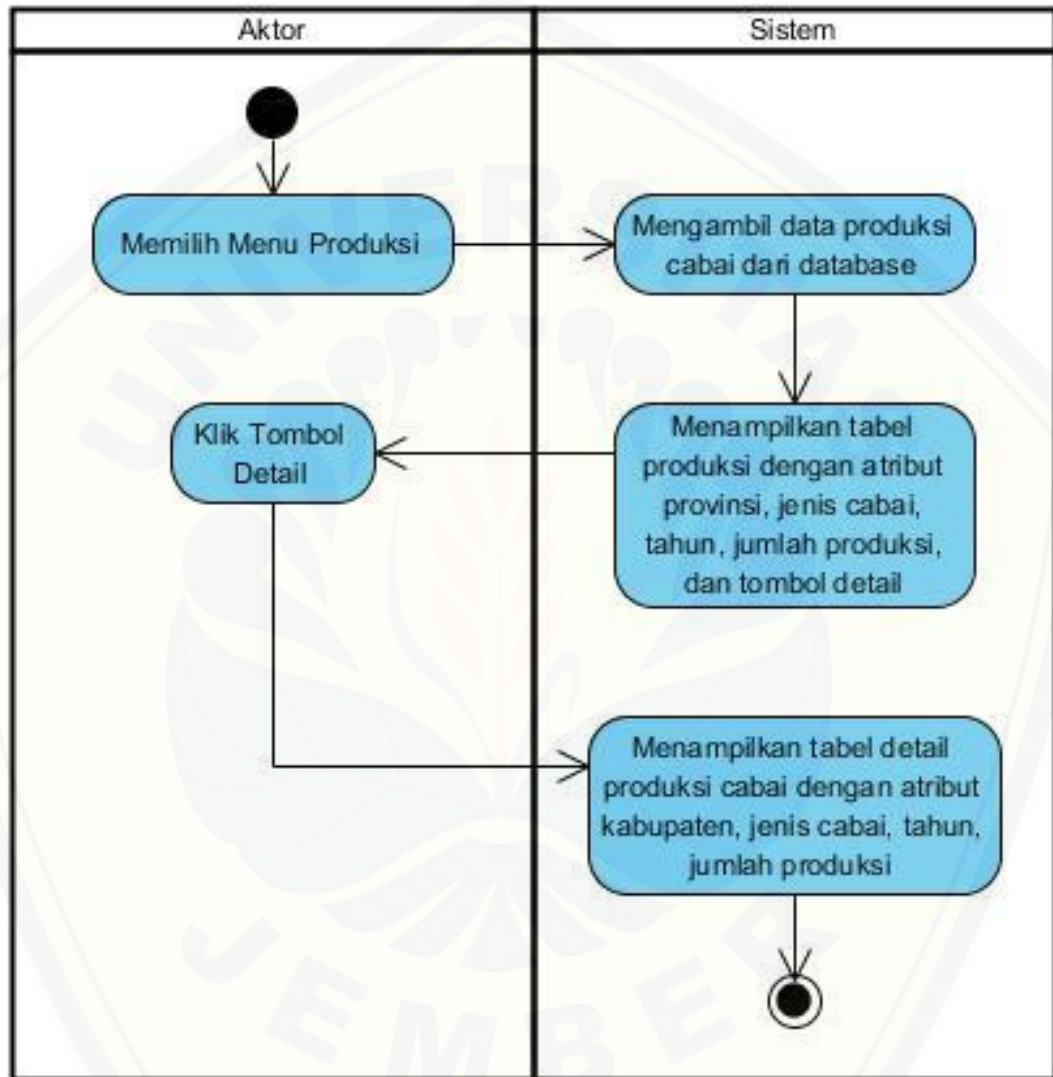
Activity ini diakses oleh admin digunakan untuk mengelola data kab/kota dalam sistem distribusi cabai. Activity Diagram dapat dilihat pada gambar B.5.



Gambar B. 5 Activity Diagram Mengelola Data Kab/Kota

B.6. Activity Diagram Melihat Data Produktivitas Cabai

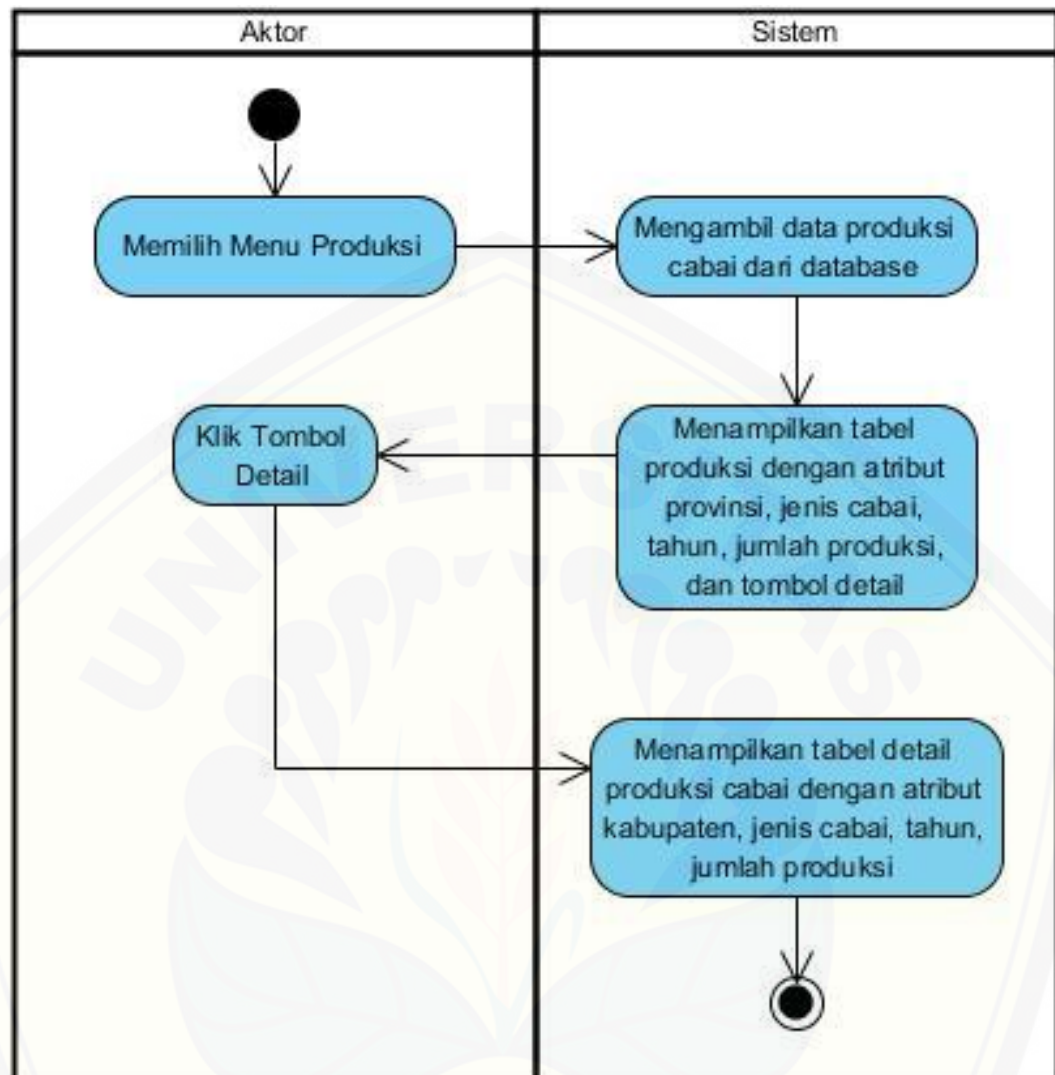
Activity ini diakses oleh petani digunakan untuk melihat data produktivitas cabai di Provinsi Jawa Timur sebagai data akurat jumlah produksi tiap tahun yang didapatkan pada tiap kabupaten. Activity Diagram dapat dilihat pada gambar B.6.



Gambar B. 6 Activity Diagram Melihat Data Produktivitas Cabai

B.7. Activity Diagram Melihat Data Konsumsi Cabai

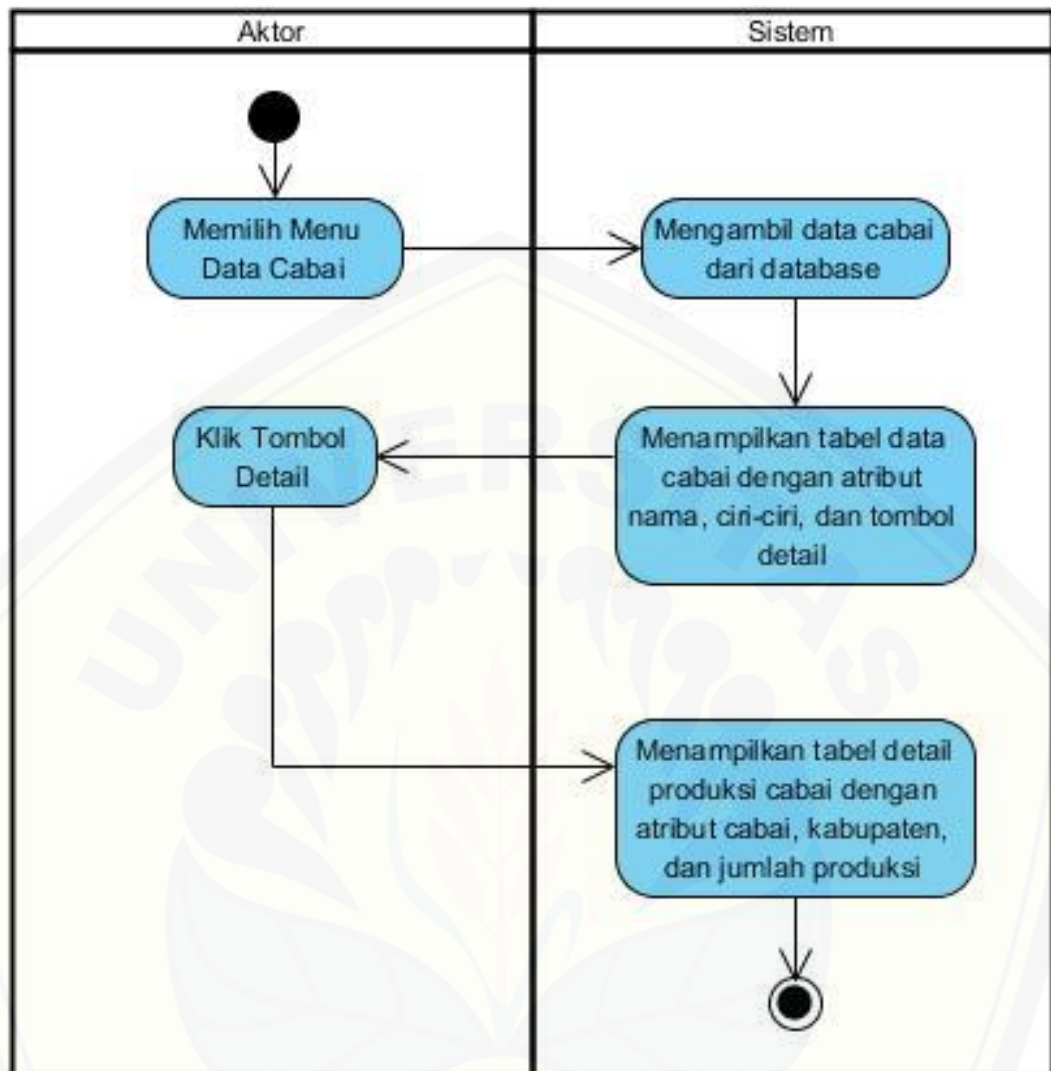
Activity ini diakses oleh petani digunakan untuk melihat data konsumsi cabai di Provinsi Jawa Timur sebagai data akurat jumlah konsumsi tiap tahun yang didapatkan pada tiap kabupaten. Activity Diagram dapat dilihat pada gambar B.7.



Gambar B. 7 *Activity Diagram* Melihat Data Konsumsi Cabai

B.8. *Activity Diagram* Melihat Data Jenis Cabai

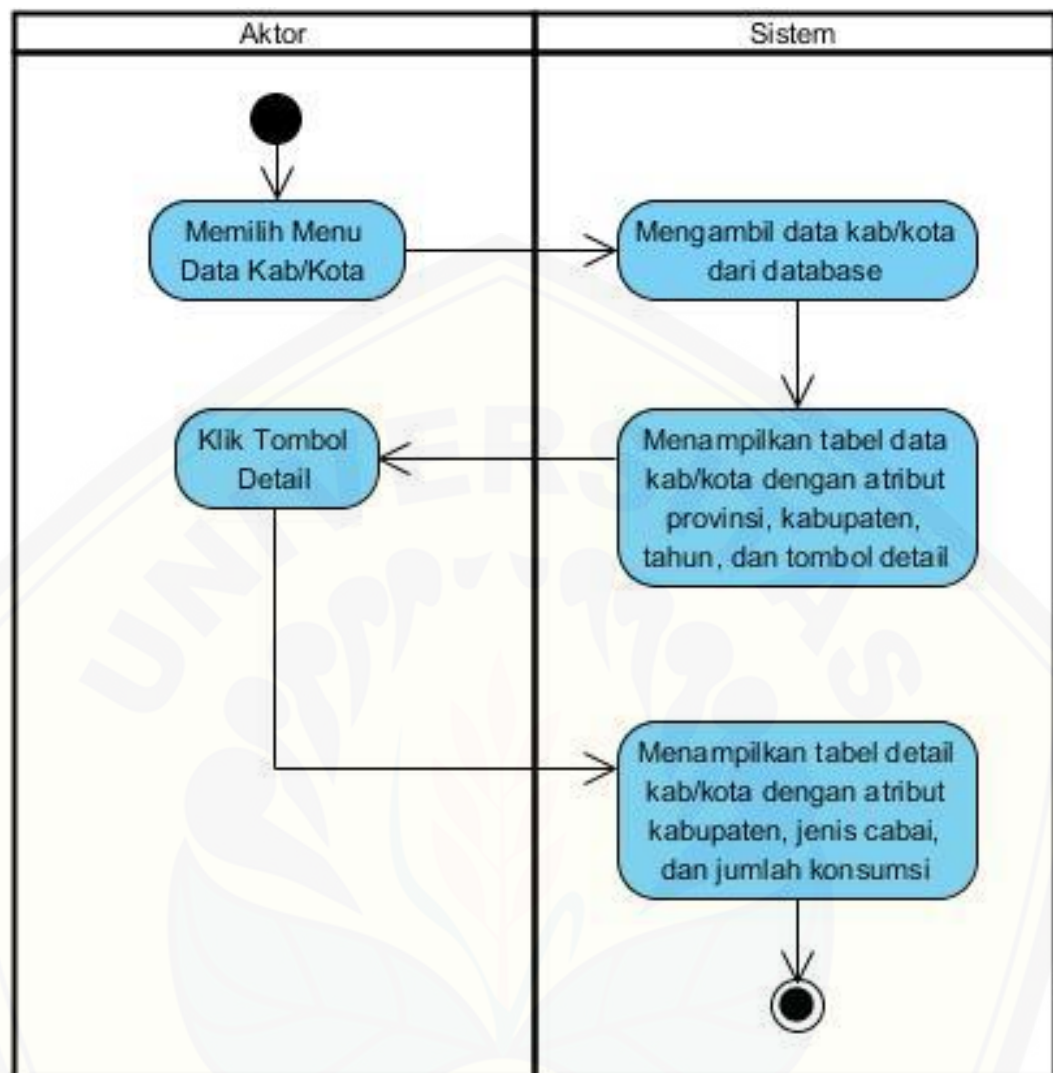
Activity ini diakses oleh petani digunakan untuk melihat data produktivitas cabai di Provinsi Jawa Timur sebagai data akurat jumlah produksi tiap tahun yang didapatkan pada tiap kabupaten. *Activity Diagram* dapat dilihat pada gambar B.8.



Gambar B. 8 *Activity Diagram* Melihat Data Jenis Cabai

B.9. *Activity Diagram* Melihat Data Kab/Kota

Activity ini diakses oleh petani digunakan untuk melihat data produktivitas cabai di Provinsi Jawa Timur sebagai data akurat jumlah produksi tiap tahun yang didapatkan pada tiap kabupaten. *Activity Diagram* dapat dilihat pada gambar B.9.

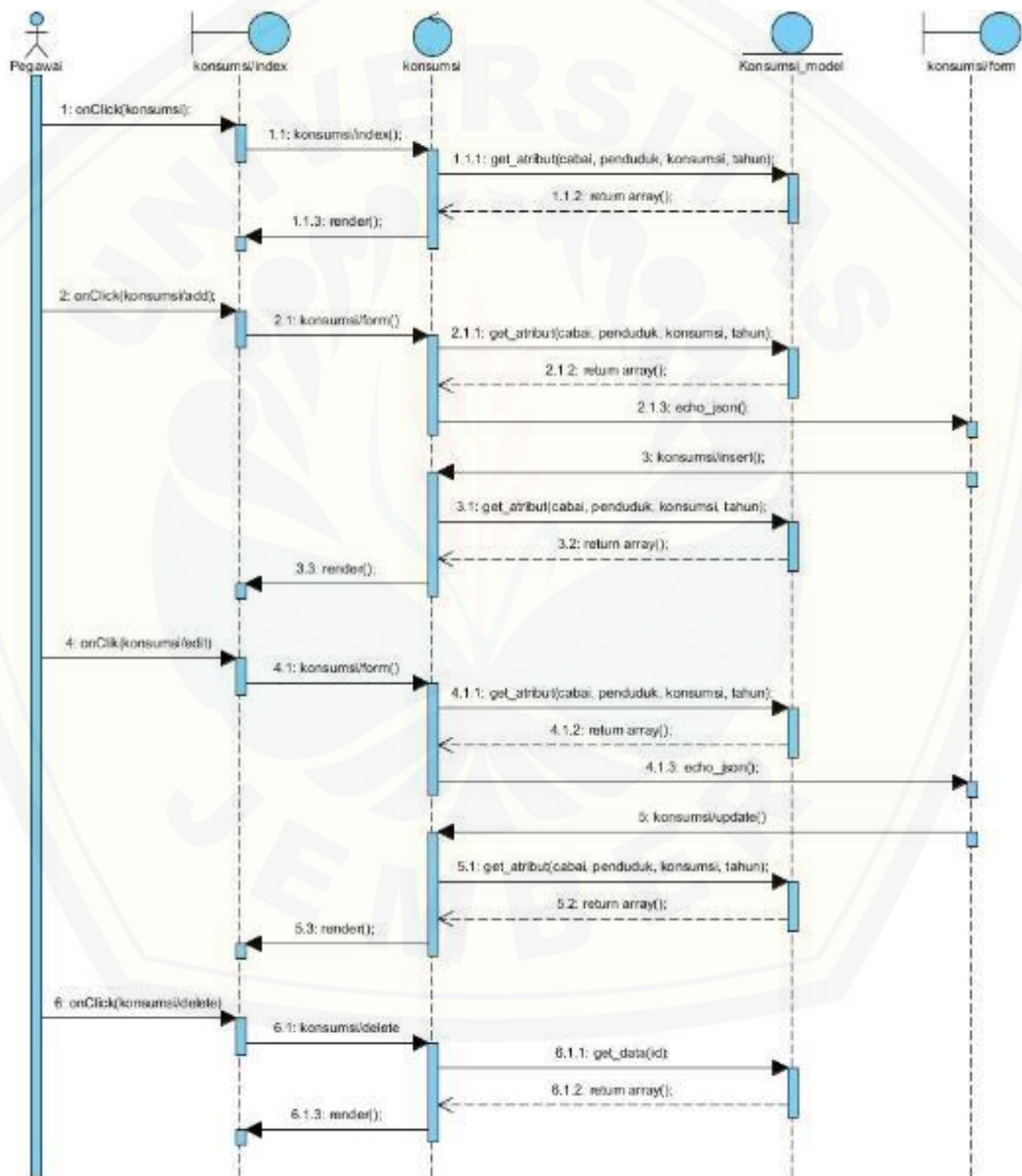


Gambar B. 9 Activity Diagram Melihat Data Kab/Kota

Lampiran C. Sequence Diagram

C.1. Sequence Diagram Mengelola Data Konsumsi Cabai

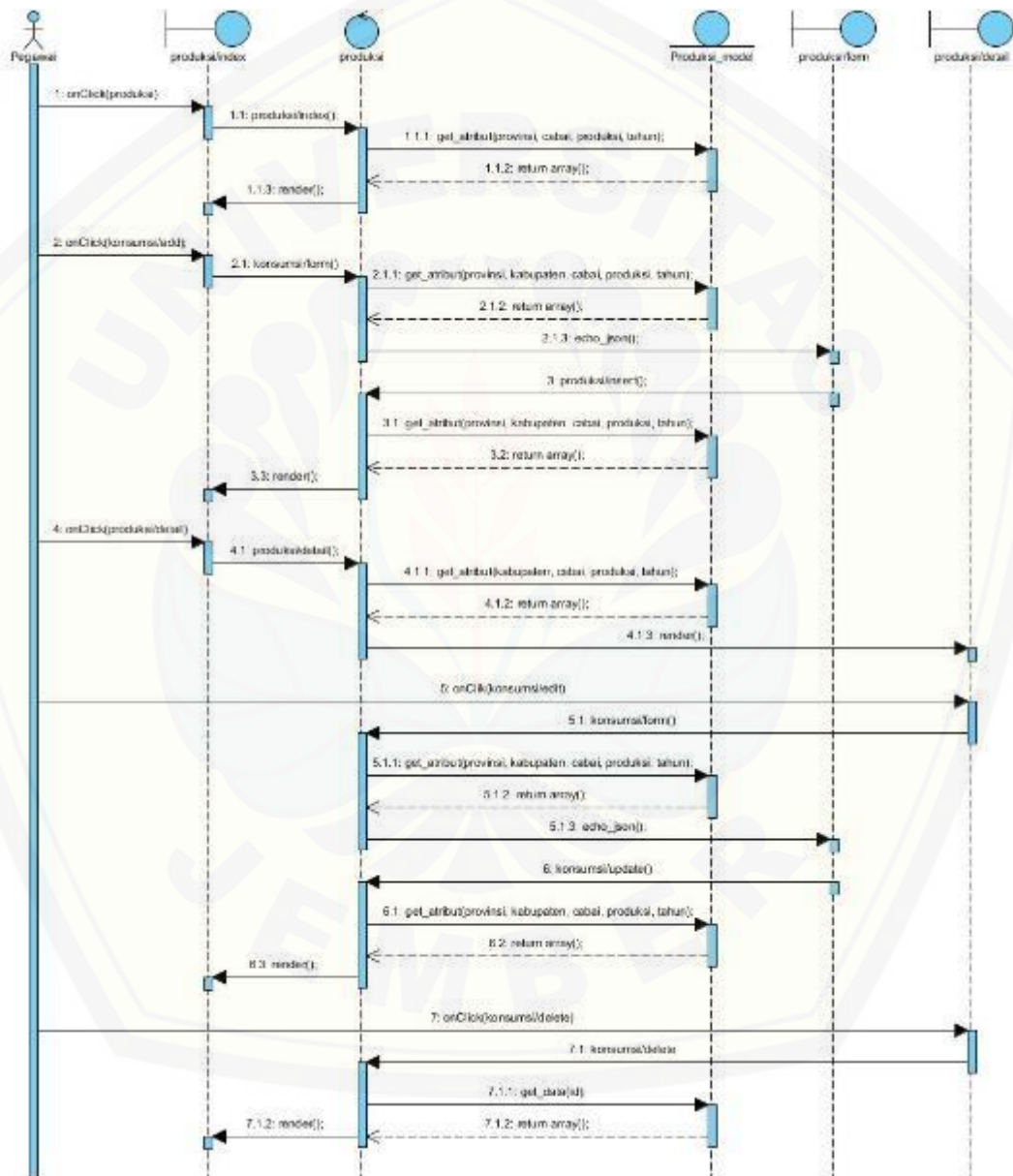
Mengelola data konsumsi cabai rawit diakses oleh pegawai. Class yang berhubungan dengan sequence ini adalah *Konsumsi_controller* dan *Konsumsi_model*. Sequence diagram untuk mengelola data konsumsi cabai dapat dilihat pada gambar C.1.



Gambar C. 1 Sequence Diagram Mengelola Data Konsumsi Cabai

C.2. *Sequence Diagram* Mengelola Data Produktivitas Cabai

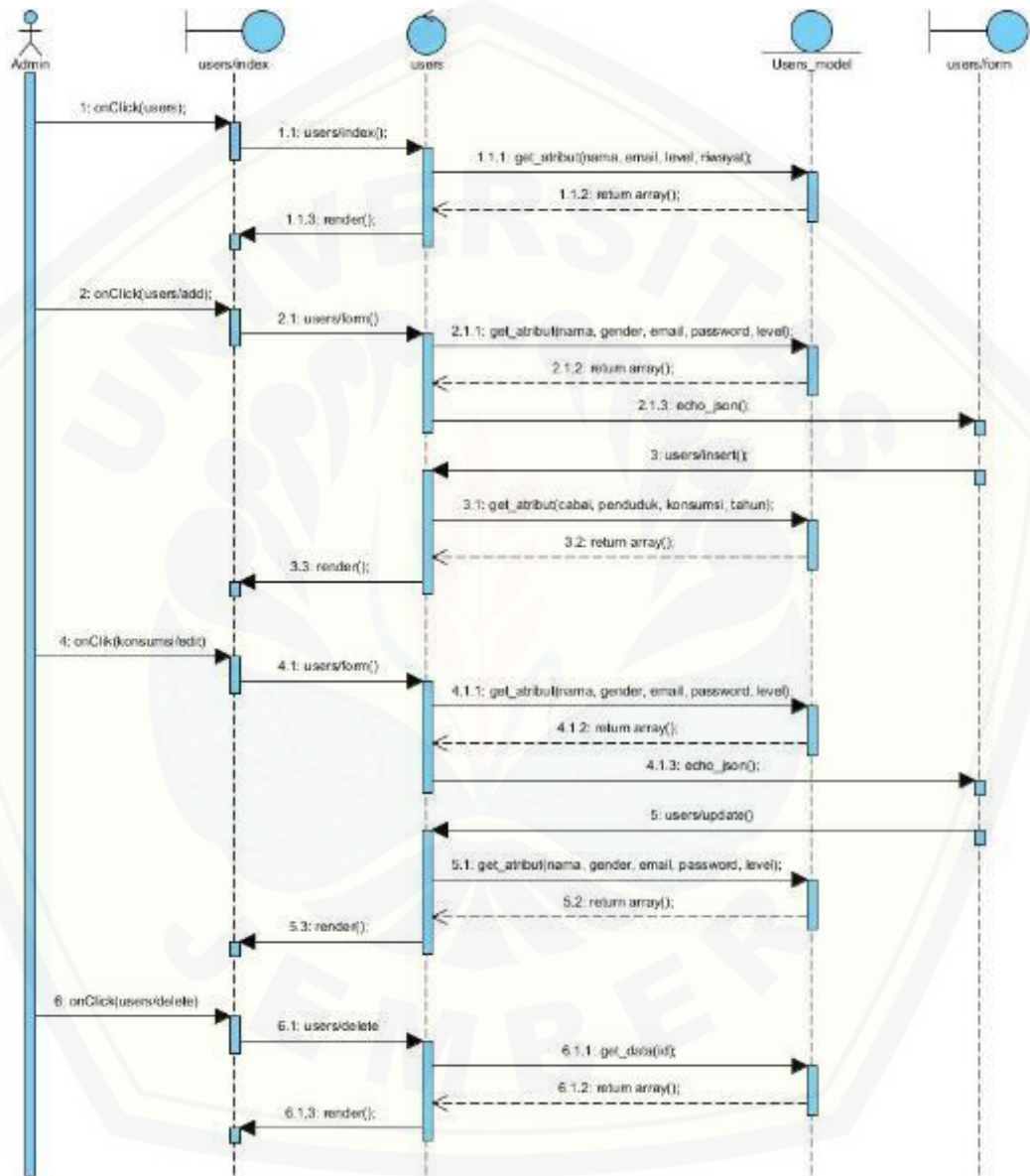
Mengelola data produktivitas cabai rawit diakses oleh pegawai. *Class* yang berhubungan dengan *sequence* ini adalah *Produksi_controller* dan *Produksi_model*. *Sequence diagram* untuk mengelola data produktivitas cabai dapat dilihat pada gambar C.2.



Gambar C. 2 *Sequence Diagram* Mengelola Data Produktivitas Cabai

C.3. *Sequence Diagram* Mengelola Data Pengguna

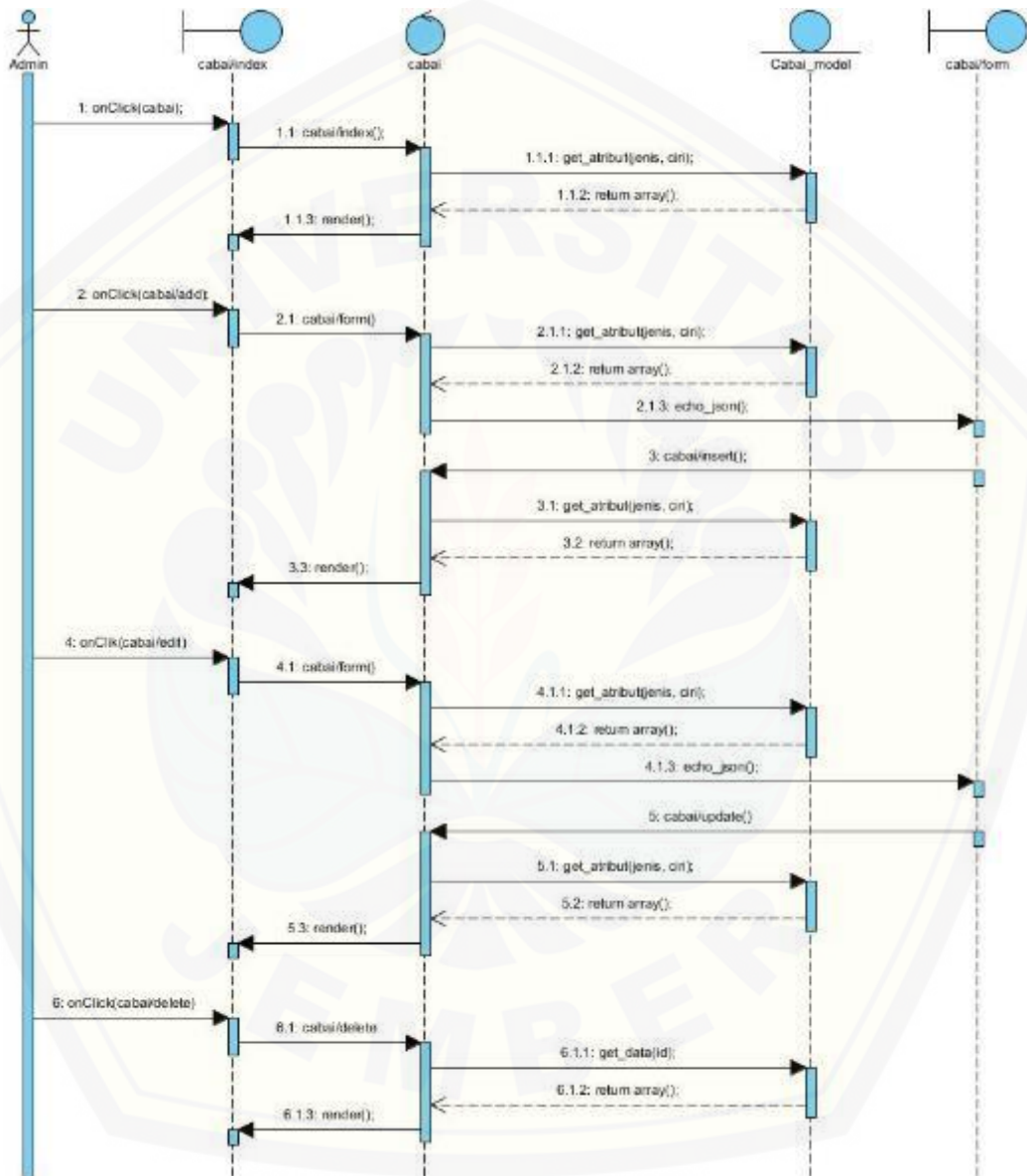
Mengelola data pengguna diakses oleh admin. *Class* yang berhubungan dengan *sequence* ini adalah *Users_controller* dan *Users_model*. *Sequence diagram* untuk mengelola data pengguna dapat dilihat pada gambar C.3.



Gambar C. 3 *Sequence Diagram* Mengelola Data Pengguna

C.4. *Sequence Diagram* Mengelola Data Cabai

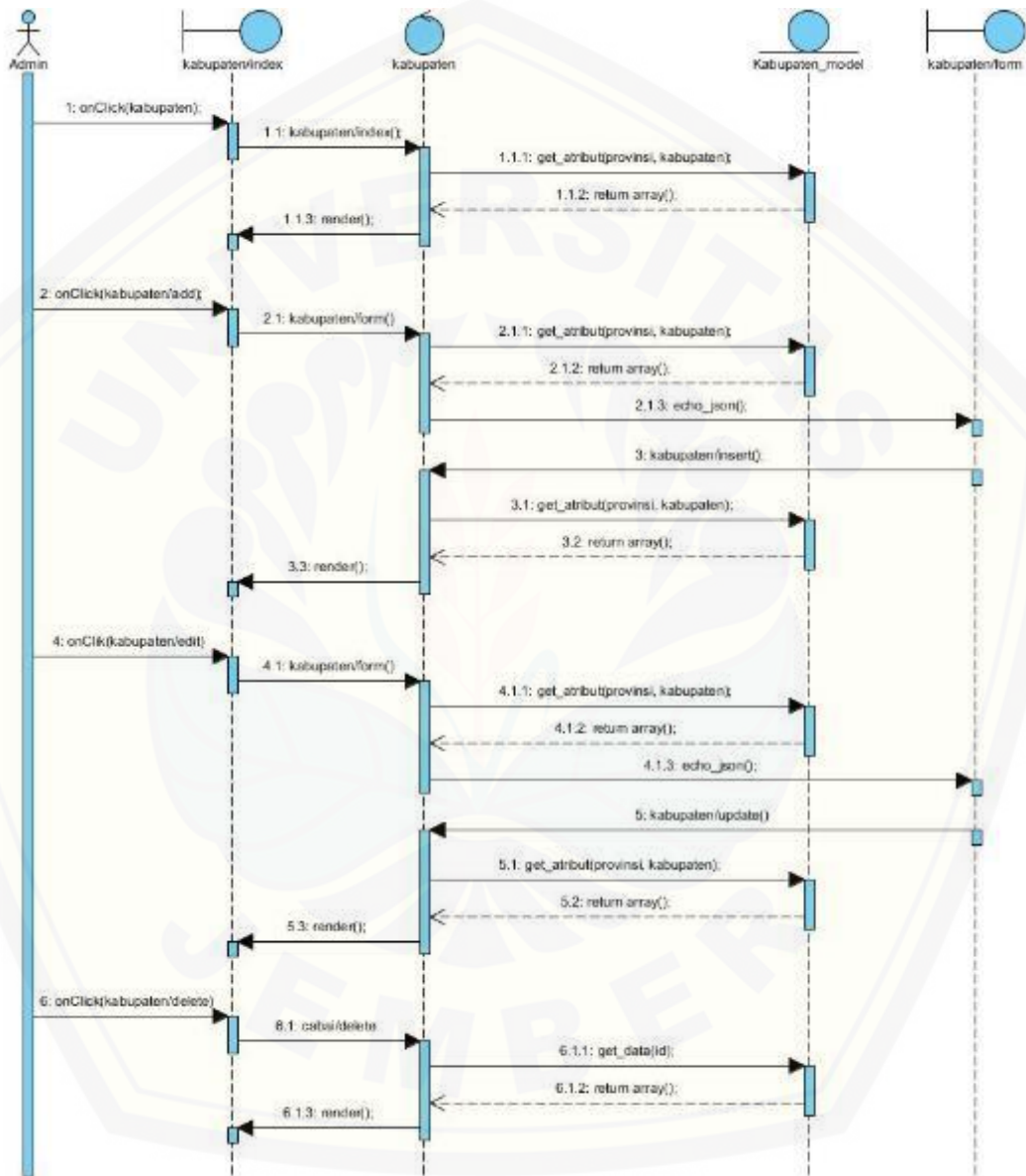
Mengelola data cabai diakses oleh admin. *Class* yang berhubungan dengan *sequence* ini adalah *Cabai_controller* dan *Cabai_model*. *Sequence diagram* untuk mengelola data cabai dapat dilihat pada gambar C.4.



Gambar C. 4 *Sequence Diagram* Mengelola Data Cabai

C.5. Sequence Diagram Mengelola Data Kab/Kota

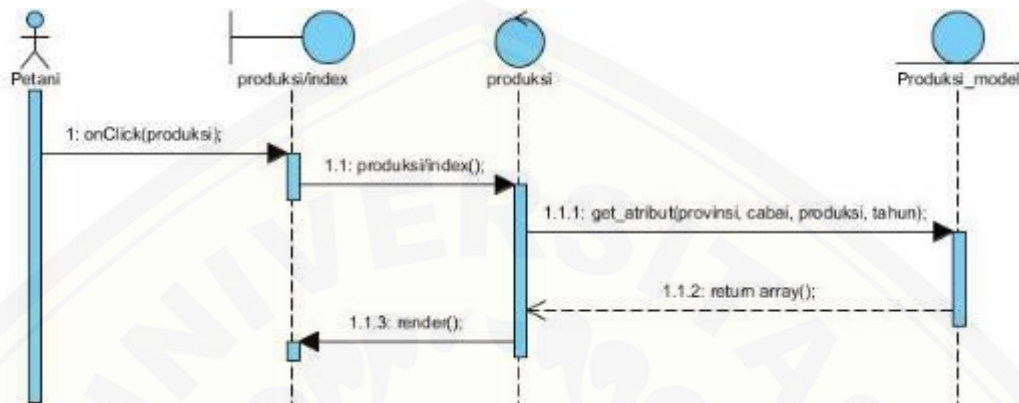
Mengelola data kab/kota diakses oleh admin. Class yang berhubungan dengan sequence ini adalah Kabupaten_controller dan Kabupaten_model. Sequence diagram untuk mengelola data kab/kota dapat dilihat pada gambar C.5.



Gambar C. 5 Sequence Diagram Mengelola Data Kab/Kota

C.6. *Sequence Diagram* Melihat Data Produktivitas Cabai

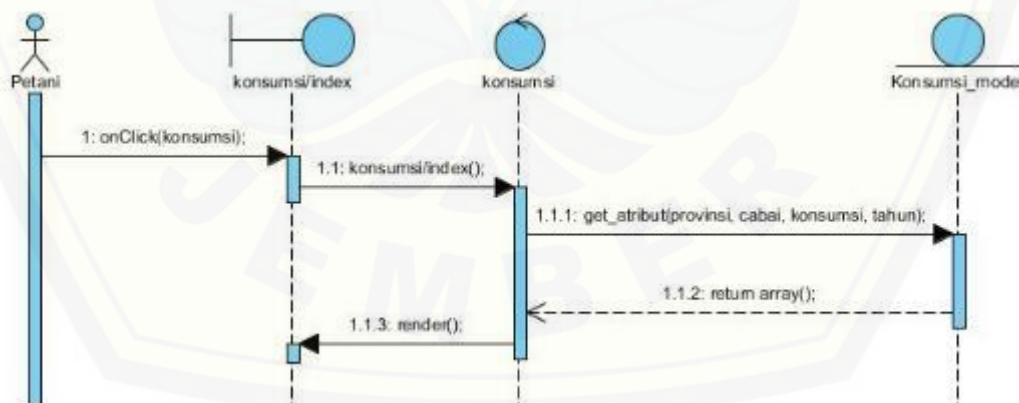
Melihat data produktivitas cabai diakses oleh petani. *Class* yang berhubungan dengan *sequence* ini adalah *Produksi_controller* dan *Produksi_model*. *Sequence diagram* untuk melihat data produktivitas cabai dapat dilihat pada gambar C.6.



Gambar C. 6 *Sequence Diagram* Melihat Data Produktivitas Cabai

C.7. *Sequence Diagram* Melihat Data Konsumsi Cabai

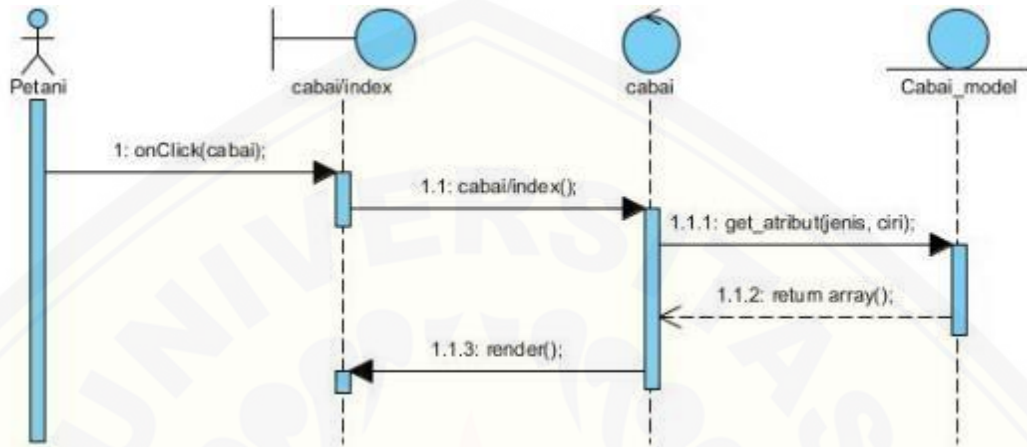
Melihat data konsumsi cabai diakses oleh petani. *Class* yang berhubungan dengan *sequence* ini adalah *Konsumsi_controller* dan *Konsumsi_model*. *Sequence diagram* untuk melihat data konsumsi cabai dapat dilihat pada gambar C.7.



Gambar C. 7 *Sequence Diagram* Melihat Data Konsumsi Cabai

C.8. *Sequence Diagram* Melihat Data Jenis Cabai

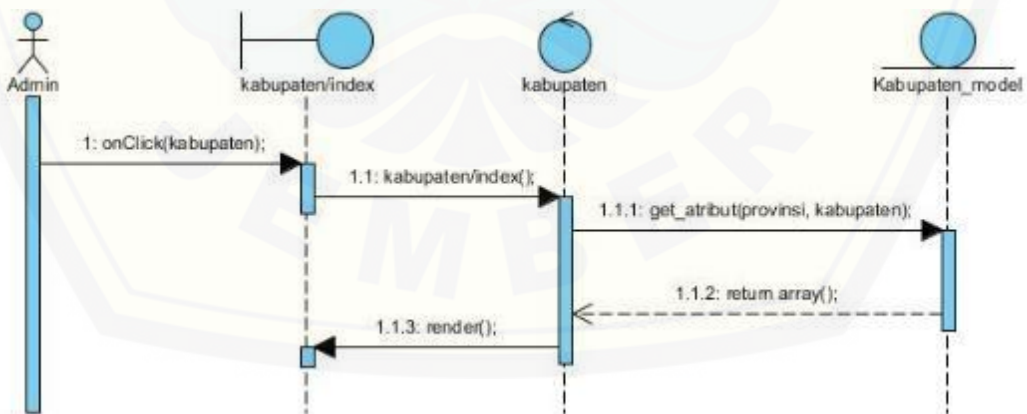
Melihat data jenis cabai diakses oleh petani. *Class* yang berhubungan dengan *sequence* ini adalah *Cabai_controller* dan *Cabai_model*. *Sequence diagram* untuk melihat data jenis cabai dapat dilihat pada gambar C.8.



Gambar C. 8 *Sequence Diagram* Melihat Data Jenis Cabai

C.9. *Sequence Diagram* Melihat Data Kab/Kota

Melihat data kab/kota diakses oleh petani. *Class* yang berhubungan dengan *sequence* ini adalah *Kabupaten_controller* dan *Kabupaten_model*. *Sequence diagram* untuk melihat data kab/kota dapat dilihat pada gambar C.9.



Gambar C. 9 *Sequence Diagram* Melihat Data Kab/Kota

Lampiran D. Kode Program

D.1. Kode Login

Penulisan kode login dalam sistem informasi distribusi cabai terbagi menjadi dua aktivitas yang terletak di *class User_model* pada *model* dan *class Auth* pada *controller*. Penulisan kode login dapat dilihat pada tabel D.1 dan tabel D.2.

Tabel D. 1 *Class Model* Kode Login

class User_model pada *model*

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Users_model extends MY_Model{
    public $_table = 'users';
    public function get_data($filter = array()){
    public function is_logged_in(){
        $data = $this->session->userdata('auth');
        Return !empty($data);
    }
    public function do_login($email, $password){
        $user = $this->select('users.id AS id, first_name,
last_name, password, group_policy')->join('user_groups','user_group =
user_groups.id')->get_by('email', $email);
        if (!empty($user)) {
            if (password_verify($password, $user->password)) {
                # write session;
                $auth = array(
                    'id'=>$user->id,
                    'first_name'=>$user->first_name,
                    'last_name'=>$user->last_name,
                    'policy'=>$user->group_policy
                );
                $this->session->set_userdata('auth',
$auth);
                $this->db->query("INSERT log_login(email,
status) VALUES('$email','success')");
                return true;
            }
        }
    }
}
```

```

        }
    }
    $this->db->query("INSERT log_login(email, status)
VALUES('$email','fail')");
    return false;
}
}

```

Tabel D. 2 *Class Controller* Kode Login

class Auth pada controller

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Auth extends MY_Controller {
    public function __construct(){
        parent::__construct();
        $this->public_class = TRUE;
        $this->load->model('Users_model', 'users');
    }
    public function index(){
        if ( $this->users->is_logged_in() )
            $this->go('dashboard');
        $this->render('auth/index');
    }
    public function do_login(){
        $email = $this->input->post('email');
        $password = $this->input->post('password');
        if ($this->users->do_login($email, $password)) {
            $this->go('dashboard');
        }else{
            $this->message('Email dan Password tidak cocok');
            $this->go('auth');
        }
    }
    public function logout(){
        $this->session->sess_destroy();
    }
}

```

```

        $this->go('auth');
    }
}

```

D.2. Kode Pengguna

Penulisan kode pengguna dalam sistem informasi distribusi cabai terletak di *class Users_model* pada *model* dan *class Users* pada *controller*. Penulisan kode pengguna dapat dilihat pada tabel D.3 dan tabel D.4:

Tabel D. 3 *Class Model* Kode Pengguna

class User_model pada *model*

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Users_model extends MY_Model{
    public $_table = 'users';
    public function get_data($filter = array()){
        # LIMIT, OFFSET, AND SORT
        $limit = isset($filter['limit'])?$filter['limit']:'';
        $offset = isset($filter['offset'])?$filter['offset']:'';
        $sort = isset($filter['sort'])?$filter['sort']:'';
        $order = isset($filter['order'])?$filter['order']:'';
        # WHERE
        $where['id'] = isset($filter['id'])?$filter['id']:'';
        # Exclude
        $excludes =
isset($filter['excludes'])?$filter['excludes']:array();
        # UNSET FILTER
        $filter = $this->_unset_filter($this-
>_set_filter($filter));
        $where = trim_array($where);
        # SET LIKE
        if (count($filter) > 0) {
            $this->group_start();
            $this->or_like($filter);
            $this->group_end();

```

```

    }
    # SET WHERE NOT IN
    if (count($excludes) > 0) {
        foreach ($excludes as $key => $value) {
            $this->where_not_in($key, $value);
        }
    }
    if (!empty($limit) OR !empty($offset)) {
        $clone = clone($this->db);
        $results['total'] = $clone->where($where)-
>from($this->_table)->count_all_results();
        $this->limit($limit, $offset);
    }
    # SET SORT
    if (!empty($sort)) {
        $this->order_by($sort, $order);
    }
    $results['rows'] = $this->_substitute($this-
>get_many_by($where));
    return
isset($results['total'])?$results:$results['rows'];
}
private function _set_filter($filter = array()){
    if (isset($filter['search'])) {
        $kolom = array('first_name', 'last_name', 'email');
        foreach ($kolom as $k) {
            $filter[$k] = $filter['search'];
        }
    }
    return $filter;
}
private function _unset_filter($filter){
    unset($filter['limit'], $filter['offset'],
$filter['order'], $filter['sort']);
    unset($filter['id'], $filter['search'],
$filter['excludes']);

```

```

        return trim_array($filter);
    }
    private function _substitute($data = array()){
        foreach ($data as $k => $v) {
            $data[$k]->user_group = $this->db->where('id',$v->
            >user_group)->get('user_groups')->row()->name;
            $last_login = $this->db->query("SELECT time FROM
            log_login WHERE email = '$v->email' AND status = 'success' ORDER BY id
            DESC LIMIT 1");
            $data[$k]->last_login = $last_login->num_rows() >
            0 ? datify($last_login->result()[0]->time, 'd-m-Y h:i') : '';
        }
        return $data;
    }
    public function is_logged_in(){
        $data = $this->session->userdata('auth');
        return !empty($data);
    }
    public function do_login($email, $password){
        $user = $this->select('users.id AS id, first_name,
        last_name, password, group_policy')->join('user_groups','user_group =
        user_groups.id')->get_by('email', $email);
        if (!empty($user)) {
            if (password_verify($password, $user->password)) {
                # write session;
                $auth = array(
                    'id'=>$user->id,
                    'first_name'=>$user->first_name,
                    'last_name'=>$user->last_name,
                    'policy'=>$user->group_policy
                );
                $this->session->set_userdata('auth',
                $auth);
                $this->db->query("INSERT log_login(email,
                status) VALUES('$email','success')");
                return true;
            }
        }
    }

```

```

        }
    }
    $this->db->query("INSERT log_login(email, status)
VALUES('$email','fail')");
    return false;
}
}

```

Tabel D. 4 *Class Controller* Kode Pengguna

class Users pada *controller*

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Users extends MY_Controller {
    public function __construct(){
        parent::__construct();
        $this->load->model('Usergroups_model', 'usergroups');
        $this->load->model('Users_model', 'users');
    }
    public function index(){
        $this->render('users/index');
    }
    public function add(){
        $data['usergroups'] = $this->usergroups->get_all();
        $this->render('users/form', $data);
    }
    public function insert(){
        $data = $this->input->post();
        unset($data['password_re']);
        $data['password'] = password_hash($data['password'],
PASSWORD_BCRYPT);
        $this->users->insert($data);
        $this->message('User berhasil ditambah');
        $this->go('users');
    }
    public function edit($id = NULL){

```

```
        if (empty($id)) {
            $this->go('users');
        }
        $data['user'] = $this->users->get($id);
        $data['usergroups'] = $this->usergroups->get_all();
        $this->render('users/form-edit', $data);
    }
    public function update(){
        $data = $this->input->post();
        $id = $data['id'];
        unset($data['id'], $data['password_re']);
        if (!empty($data['password'])) {
            $data['password'] =
password_hash($data['password'], PASSWORD_BCRYPT);
        } else {
            unset($data['password']);
        }
        $this->users->update($id, $data);
        $this->message('Data berhasil disunting');
        $this->go('users');
    }
    public function delete($id = null){
        if (empty($id)) {
            show_404();
        }
        $this->users->delete($id);
        $this->message('User berhasil dihapus');
        $this->go('users');
    }
    public function profile(){
        $id = $this->session->userdata('auth')['id'];
        $data['user'] = $this->users->get($id);
        $this->render('users/profile', $data);
    }
    public function update_profile(){
        $data = $this->input->post();
        $id = $data['id'];
        unset($data['id'], $data['password_re']);
```

```
        if (!empty($data['password'])) {
            $data['password'] =
password_hash($data['password'], PASSWORD_BCRYPT);
        } else {
            unset($data['password']);
        }
        $this->users->update($id, $data);
        # Update session
        $currentSession = $this->session->userdata('auth');
        $currentSession['first_name'] = $data['first_name'];
        $currentSession['last_name'] = $data['last_name'];
        $this->session->set_userdata('auth', $currentSession);
        $this->message('Profil berhasil disunting');
        $this->go('users/profile');
    }
    public function api_get(){
        $filter = $this->input->get();
        $data = $this->users->get_data($filter);
        $url = site_url('users');
        foreach($data['rows'] as $d){
            $fix_url_edit = $url.'/edit/'.$d->id;
            $fix_url_hapus = $url.'/delete/'.$d->id;
            $d->action = "<a href='$fix_url_edit' class='btn
btn-sm btn-warning'><i class='icon-pencil'></i></a>";
            $d->action .= "<a href='$fix_url_hapus'
onclick=\"return confirm('Apakah anda yakin?')\" class='btn btn-sm
btn-danger'><i class='icon-trash'></i></a>";
        }
        echo json_encode($data);
    }
}
```

D.3. Kode Data Cabai

Penulisan kode data cabai dalam sistem informasi distribusi cabai terletak di *class Cabai_model* pada *model* dan *class Cabai* pada *controller*. Penulisan kode data cabai dapat dilihat pada tabel berikut ini:

Tabel D. 5 *Class Model* Kode Data Cabai

class Cabai_model pada *model*

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Cabai_model extends MY_Model{
    public $_table = 'cabai';
    public function get_data($filter = array()){
        # LIMIT, OFFSET, AND SORT
        $limit = isset($filter['limit'])?$filter['limit']:'';
        $offset = isset($filter['offset'])?$filter['offset']:'';
        $sort = isset($filter['sort'])?$filter['sort']:'';
        $order = isset($filter['order'])?$filter['order']:'';
        $where['id'] = isset($filter['id'])?$filter['id']:'';
        $excludes =
        isset($filter['excludes'])?$filter['excludes']:array();
        $filter = $this->unset_filter($this-
>_set_filter($filter));
        $where = trim_array($where);
        # SET LIKE
        if (count($filter) > 0) {
            $this->group_start();
            $this->or_like($filter);
            $this->group_end();
        }
        # SET WHERE NOT IN
        if (count($excludes) > 0) {
            foreach ($excludes as $key => $value) {
                $this->where_not_in($key, $value);
            }
        }
        if (!empty($limit) OR !empty($offset)) {

```

```

        $clone = clone($this->db);
        $results['total'] = $clone->where($where)-
>from($this->_table)->count_all_results();
        $this->limit($limit, $offset);
    }
    if (!empty($sort)) {
        $this->order_by($sort, $order);
    }
    $results['rows'] = $this->get_many_by($where);
    return
isset($results['total'])?$results:$results['rows'];
}
private function _set_filter($filter = array()){
    if (isset($filter['search'])) {
        $kolom = array('jenis', 'ciri');
        foreach ($kolom as $k) {
            $filter[$k] = $filter['search'];
        }
    }
    return $filter;
}
private function _unset_filter($filter){
    unset($filter['limit'], $filter['offset'],
$filter['order'], $filter['sort']);
    unset($filter['id'], $filter['search'],
$filter['excludes']);
    return trim_array($filter);
}
}
}

```

Tabel D. 6 *Class Controller* Kode Data Cabai

class Cabai pada controller

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Cabai extends MY_Controller {

```

```
public function __construct(){
    parent::__construct();
    $this->load->model('Cabai_model', 'cabai');
}
public function index(){
    $this->render('cabai/index');
}
public function add(){
    $this->render('cabai/form');
}
public function insert(){
    $data = $this->input->post();
    $this->cabai->insert($data);
    $this->message('Data berhasil ditambah', 'success');
    $this->go('cabai');
}
public function edit($id = null){
    if (empty($id)) {
        show_404();
    }
    $data['cabai'] = $this->cabai->get($id);
    $this->render('cabai/form', $data);
}
public function update(){
    $data = $this->input->post();
    $id = $data['id'];
    unset($data['id']);
    $this->cabai->update($id, $data);
    $this->message('Data berhasil disunting', 'success');
    $this->go('cabai');
}
public function delete($id = null){
    if (empty($id)) {
        show_404();
    }
    $this->cabai->delete($id);
}
```

```

        $this->message('Data berhasil dihapus');
        $this->go('cabai');
    }
    public function api_get(){
        $filter = $this->input->get();
        $data = $this->cabai->get_data($filter);
        $url = site_url('cabai');
        foreach($data['rows'] as $d){
            $fix_url_edit = $url.'/edit/'.$d->id;
            $fix_url_hapus = $url.'/delete/'.$d->id;
            $d->action = "<a href='$fix_url_edit' class='btn
btn-sm btn-warning'><i class='icon-pencil'></i></a>";
            $d->action .= "<a href='$fix_url_hapus'
onclick=\"return confirm('Apakah anda yakin?')\" class='btn btn-sm
btn-danger'><i class='icon-trash'></i></a>";
        }
        echo json_encode($data);
    }
}

```

D.4. Kode Produksi Cabai

Penulisan kode produksi cabai dalam sistem informasi distribusi cabai terletak di *class Produksi_model* pada *model* dan *class Produksi* pada *controller*. Penulisan kode produksi cabai dapat dilihat pada tabel berikut ini:

Tabel D. 7 *Class Model* Kode Produksi Cabai

```

class Produksi_model pada model

```

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Produksi_model extends MY_Model{
    public $_table = 'produksi';
    public function get_data($filter = array()){
        # LIMIT, OFFSET, AND SORT
        $limit = isset($filter['limit'])?$filter['limit']:'';
        $offset = isset($filter['offset'])?$filter['offset']:'';

```

```

    $sort = isset($filter['sort'])? $filter['sort']: '';
    $order = isset($filter['order'])? $filter['order']: '';
    # SELECT
    $this->select('produksi.*, kabupaten.nama AS kabupaten,
cabai.jenis AS cabai');
    # WHERE
    $where['id'] = isset($filter['id'])? $filter['id']: '';
    # JOIN
    $this->join('cabai', 'id_cabai = cabai.id')
        ->join('kabupaten', 'id_kabupaten =
kabupaten.id');
    $excludes =
isset($filter['excludes'])? $filter['excludes']: array();
    $filter = $this->unset_filter($this-
>set_filter($filter));
    $where = trim_array($where);
    if (count($filter) > 0) {
        $this->group_start();
        $this->or_like($filter);
        $this->group_end();
    }
    if (count($excludes) > 0) {
        foreach ($excludes as $key => $value) {
            $this->where_not_in($key, $value);
        }
    }
    if (!empty($limit) OR !empty($offset)) {
        $clone = clone($this->db);
        $results['total'] = $clone->where($where)-
>from($this->_table)->count_all_results();
        $this->limit($limit, $offset);
    }
    if (!empty($sort)) {
        $this->order_by($sort, $order);
    }
    $results['rows'] = $this->get_many_by($where);

```

```

        return
    isset($results['total'])? $results: $results['rows'];
    }
    private function _set_filter($filter = array()){
        if (isset($filter['search'])) {
            $kolom = array('jumlah', 'tahun', 'cabai.jenis',
'kabupaten.nama');
            foreach ($kolom as $k) {
                $filter[$k] = $filter['search'];
            }
        }
        return $filter;
    }
    private function _unset_filter($filter){
        unset($filter['limit'], $filter['offset'],
$filter['order'], $filter['sort']);
        unset($filter['id'], $filter['search'],
$filter['excludes']);
        return trim_array($filter);
    }
}

```

Tabel D. 8 *Class Controller* Kode Produksi Cabai

class Produksi pada *controller*

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Produksi extends MY_Controller {
    public function __construct(){
        parent::__construct();
        $this->load->model('Produksi_model', 'produksi');
        $this->load->model('Cabai_model', 'cabai');
        $this->load->model('Kabupaten_model', 'kabupaten');
    }
    public function index(){
        $this->render('produksi/index');
    }
}

```

```
    }
    public function add(){
        $data['cabai'] = $this->cabai->order_by('jenis')-
>get_all();
        $data['kabupaten'] = $this->kabupaten->order_by('nama')-
>get_all();
        $this->render('produksi/form', $data);
    }
    public function insert(){
        $data = $this->input->post();
        $this->produksi->insert($data);
        $this->message('Data berhasil ditambah', 'success');
        $this->go('produksi');
    }
    public function edit($id = null){
        if (empty($id)) {
            show_404();
        }
        $data['cabai'] = $this->cabai->get_all();
        $data['kabupaten'] = $this->kabupaten->get_all();
        $data['produksi'] = $this->produksi->get($id);
        $this->render('produksi/form', $data);
    }
    public function update(){
        $data = $this->input->post();
        $id = $data['id'];
        unset($data['id']);
        $this->produksi->update($id, $data);
        $this->message('Data berhasil disunting', 'success');
        $this->go('produksi');
    }
    public function delete($id = null){
        if (empty($id)) {
            show_404();
        }
        $this->produksi->delete($id);
    }
}
```

```

        $this->message('Data berhasil dihapus');
        $this->go('produksi');
    }
    public function api_get(){
        $filter = $this->input->get();
        $data = $this->produksi->get_data($filter);
        $url = site_url('produksi');
        foreach($data['rows'] as $d){
            $fix_url_edit = $url.'/edit/'.$d->id;
            $fix_url_hapus = $url.'/delete/'.$d->id;
            $d->action = "<a href='$fix_url_edit' class='btn
btn-sm btn-warning'><i class='icon-pencil'></i></a>";
            $d->action .= "<a href='$fix_url_hapus'
onclick=\"return confirm('Apakah anda yakin?')\" class='btn btn-sm
btn-danger'><i class='icon-trash'></i></a>";
        }
        echo json_encode($data);
    }
}

```

D.5. Kode Konsumsi Cabai

Penulisan kode pengelolaan konsumsi cabai dalam sistem informasi distribusi cabai berjalan pada *class Konsumsi controller* dengan aksi menggunakan *Konsumsi_model* untuk mengambil data dari database. Penulisan kode konsumsi cabai dapat dilihat pada tabel berikut ini:

Tabel D. 9 *Class Model* Kode Konsumsi Cabai

class Konsumsi_model pada *model*

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Konsumsi_model extends MY_Model{
    public $_table = 'konsumsi';
    public function get_data($filter = array()){
        # LIMIT, OFFSET, AND SORT
        $limit = isset($filter['limit'])? $filter['limit']: '';
    }
}

```

```
$offset = isset($filter['offset'])? $filter['offset']: '';
$sort   = isset($filter['sort'])? $filter['sort']: '';
$order  = isset($filter['order'])? $filter['order']: '';
# SELECT
$this->select('konsumsi.*, cabai.jenis AS cabai');
# WHERE
$where['id'] = isset($filter['id'])? $filter['id']: '';
# JOIN
$this->join('cabai', 'id_cabai = cabai.id');
# Exclude
$excludes =
isset($filter['excludes'])? $filter['excludes']: array();
# UNSET FILTER
$filter = $this->_unset_filter($this->_set_filter($filter));
$where = trim_array($where);
# SET LIKE
if (count($filter) > 0) {
    $this->group_start();
    $this->or_like($filter);
    $this->group_end();
}
# SET WHERE NOT IN
if (count($excludes) > 0) {
    foreach ($excludes as $key => $value) {
        $this->where_not_in($key, $value);
    }
}
if (!empty($limit) OR !empty($offset)) {
    $clone = clone($this->db);
    $results['total'] = $clone->where($where)->from($this->_table)->count_all_results();
    $this->limit($limit, $offset);
}
# SET SORT
if (!empty($sort)) {
```

```

        $this->order_by($sort, $order);
    }
    $results['rows'] = $this->get_many_by($where);
    return
isset($results['total'])?$results:$results['rows'];
}
private function _set_filter($filter = array()){
    if (isset($filter['search'])) {
        $kolom = array('jumlah_penduduk',
'konsumsi_perkapita', 'tahun', 'cabai.jenis');
        foreach ($kolom as $k) {
            $filter[$k] = $filter['search'];
        }
    }
    return $filter;
}
private function _unset_filter($filter){
    unset($filter['limit'],$filter['offset'],
$filter['order'], $filter['sort']);
    unset($filter['id'], $filter['search'],
$filter['excludes']);
    return trim_array($filter);
}
}
}

```

Tabel D. 10 *Class Controller* Kode Konsumsi Cabai

class Konsumsi pada *controller*

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Konsumsi extends MY_Controller {
    public function __construct(){
        parent::__construct();
        $this->load->model('Konsumsi_model', 'konsumsi');
        $this->load->model('Cabai_model', 'cabai');
    }
}

```

```
public function index(){
    $this->render('konsumsi/index');
}
public function add(){
    $data['cabai'] = $this->cabai->order_by('jenis')->get_all();
    $this->render('konsumsi/form', $data);
}
public function insert(){
    $data = $this->input->post();
    $this->konsumsi->insert($data);
    $this->message('Data berhasil ditambah', 'success');
    $this->go('konsumsi');
}
public function edit($id = null){
    if (empty($id)) {
        show_404();
    }
    $data['cabai'] = $this->cabai->get_all();
    $data['konsumsi'] = $this->konsumsi->get($id);
    $this->render('konsumsi/form', $data);
}
public function update(){
    $data = $this->input->post();
    $id = $data['id'];
    unset($data['id']);
    $this->konsumsi->update($id, $data);
    $this->message('Data berhasil disunting', 'success');
    $this->go('konsumsi');
}
public function delete($id = null){
    if (empty($id)) {
        show_404();
    }
    $this->konsumsi->delete($id);
    $this->message('Data berhasil dihapus');
```

```

        $this->go('konsumsi');
    }
    public function api_get(){
        $filter = $this->input->get();
        $data = $this->konsumsi->get_data($filter);
        $url = site_url('konsumsi');
        foreach($data['rows'] as $d){
            $fix_url_edit = $url.'/edit/'.$d->id;
            $fix_url_hapus = $url.'/delete/'.$d->id;
            $d->action = "<a href='$fix_url_edit' class='btn btn-sm btn-warning'><i class='icon-pencil'></i></a>";
            $d->action .= "<a href='$fix_url_hapus' onclick=\"return confirm('Apakah anda yakin?')\" class='btn btn-sm btn-danger'><i class='icon-trash'></i></a>";
        }
        echo json_encode($data);
    }
}

```

D.6. Kode *Distribution Requirements Planning*

Penulisan kode *Distribution Requirements Planning* dalam sistem informasi distribusi cabai terletak di *class Distribusi_model* pada model dan *class Distribusi* pada *controller*. Penulisan kode konsumsi cabai dapat dilihat pada tabel berikut ini

Tabel D. 11 *Class Model Distribution Requirements Planning*

class Kalkulasi_model pada model

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Kalkulasi_model extends MY_Model{
    public $_table = 'distribusi';
    public function get_data($idKab){
        $cabai = $this->cabaikab->get_many_by(array('id_kabupaten' =>
        $idKab));
        $html = "<option selected>Pilih Jenis Cabai...</option>";
        foreach($cabai as $row){

```

```

        $cab = $this->cabai->get($row->id_cabai);
        $html .= "<option value='". $cab->id .'>".$cab-
>jenis."</option>";
    }
    echo $html;
}
}

```

Tabel D. 12 *Class Controller Distribution Requirements Planning*

class Kalkulasi pada controller

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Kalkulasi extends MY_Controller {
    public function __construct(){
        parent::__construct();
        $this->load->model('Produksi_model', 'produksi');
        $this->load->model('Konsumsi_model', 'konsumsi');
        $this->load->model('Cabai_model', 'cabai');
        $this->load->model('Cabaikab_model', 'cabaikab');
        $this->load->model('Stok_model', 'stok');
        $this->load->model('Provinsi_model', 'provinsi');
        $this->load->model('Kabupaten_model', 'kabupaten');
    }
    public function index(){
        $data['kabupaten'] = $this->kabupaten->get_all();
        $this->render('kalkulasi/index', $data);
    }
    public function kalkulasi($idkota, $idcabai){
        $bulan = date('n') + 1;
        if($bulan == 13)
            $bulan = 1;
        $produksiSebelumnya = $this->produksi
->order_by(array("tahun" => 'asc'))

```

```

-
>get_by(array('id_kabupaten'=>$idkota,'id_cabai'=>$idcabai, 'bulan' =>
$bulan))->jumlah;
    $konsumsiSebelumnya = $this->konsumsi
        ->order_by(array("tahun" => 'asc'))
-
>get_by(array('id_kabupaten'=>$idkota,'id_cabai'=>$idcabai, 'bulan' =>
$bulan))->konsumsi_perkapita;
    $leadtime = $this->kabupaten->get_by(array('id'=>$idkota))-
>lead_time;
    $persediaan = $this->produksi
        ->order_by(array("tahun" => 'asc'))
-
>get_by(array('id_kabupaten'=>$idkota,'id_cabai'=>$idcabai, 'bulan' =>
$bulan))->jumlah;
    $dataKonsumsi = $this->konsumsi
        ->order_by(array("tahun" => 'asc'))
        ->limit(12)
-
>get_many_by(array('id_kabupaten'=>$idkota,'id_cabai'=>$idcabai));
    $totalKonsumsi = 0;
    foreach($dataKonsumsi as $k) {
        $totalKonsumsi = $k->konsumsi_perkapita;
    }
    $keb_kotor = $konsumsiSebelumnya;
    $MA_keb_kotor = $totalKonsumsi/12;
    $tmp = 0;
    foreach($dataKonsumsi as $konsumsi){
        $tmp += pow(($konsumsi->konsumsi_perkapita -
$MA_keb_kotor)/12, 2);
    }
    $stdev = sqrt($tmp/12);
    $safety_stock = $stdev * 1.28 * sqrt($leadtime);
    $keb_bersih = ($keb_kotor + $safety_stock) - ($persediaan -
$safety_stock);
    $pelepasan = $keb_bersih + $safety_stock;

```

```

    $data['leadtime'] = $leadtime;
    $data['safety_stock'] = $safety_stock;
    $data['persediaan'] = $produksiSebelumnya;
    $data['keb_kotor'] = $keb_kotor;
    $data['keb_bersih'] = $keb_bersih;
    $data['pelepasan'] = $pelepasan;
    $dataProduksi = $this->produksi
        ->order_by(array("tahun" => 'asc', 'bulan' => 'asc'))
    -
>get_many_by(array('id_kabupaten'=>$idkota,'id_cabai'=>$idcabai));
    $kk = array();
    $kb = array();
    $pp = array();
    for($i = 0; $i < 12; $i++) {
        $kk[$i] = (int) $dataKonsumsi[$i]->konsumsi_perkapita;
        $ss = $stdev * 1.28 * sqrt($leadtime);
        $kb[$i] = ($kk[$i] + $safety_stock) - ($dataProduksi[$i]-
>jumlah - $safety_stock);
        $pp[$i] = $kb[$i] + $ss;
    }
    $data['kk'] = $kk;
    $data['kb'] = $kb;
    $data['pp'] = $pp;
    $data['stdev'] = $stdev;
    echo json_encode($data);
}
}

```

Lampiran E. *Black Box Testing*

E.1. *Black Box Testing* Pegawai

Pengujian menggunakan *black box* untuk akses sistem yang dilakukan oleh pegawai dapat dilihat pada tabel berikut:

Tabel E. 1 *Black Box Testing* Pegawai

No	Menu	Fungsi	Aksi	Hasil	Ket
1	Login (Admin)	Digunakan admin untuk masuk dalam sistem	Mengisi Username dan Password pada form login kemudian menekan tombol login	Login berhasil dan menampilkan halaman home admin	[✓]
			Username dan Password tidak diisi	Menampilkan notifikasi "Please fill out this field"	[✓]
2	Data Produksi Cabai (Pegawai)	Digunakan untuk melihat, & mengelola data cabai (menambah, mengubah, & menghapus)	Memilih menu data cabai	Menampilkan tabel data cabai dengan atribut jenis cabai dan ciri-cirinya	[✓]
			Memilih tombol tambah data cabai	Menampilkan form tambah data cabai dengan atribut jenis cabai, ciri- ciri cabai, tombol simpan dan tombol reset	[✓]
			Memilih tombol reset form data cabai	Menampilkan form tambah data cabai dengan atribut jenis cabai, ciri-	[✓]

No	Menu	Fungsi	Aksi	Hasil	Ket
				ciri cabai, tombol simpan dan tombol reset	
			Memilih tombol simpan data cabai	Menampilkan tabel data cabai baru yang telah ditambahkan dengan atribut jenis cabai dan ciri-cirinya	[✓]
			Memilih tombol edit data cabai	Menampilkan form edit data cabai dengan rincian data yang terpilih sesuai atribut jenis cabai, ciri- ciri cabai, tombol simpan dan tombol reset	[✓]
			Memilih tombol simpan edit data cabai	Menampilkan tabel data cabai yang telah diubah dengan atribut jenis cabai dan ciri- cirinya	[✓]
			Memilih tombol hapus data cabai	Menampilkan tabel data cabai	[✓]

No	Menu	Fungsi	Aksi	Hasil	Ket
				yang telah dihapus dengan atribut jenis cabai dan ciri-cirinya	
3	Kalkulasi Distribusi (Admin)	Digunakan untuk menghitung distribusi cabai menggunakan metode <i>distribution requirement planning</i>	Mengisi form seleksi jenis cabai	Menampilkan data pilihan jenis cabai, yakni cabai besar dan cabai rawit	[✓]
			Mengisi form seleksi data kota	Menampilkan data pilihan kota yang tersedia dalam database.	[✓]
			Mengisi form data tahun produksi	Menampilkan data pilihan tahun produksi cabai yang tersedia dalam database.	[✓]
			Memilih tombol kalkulasi	Menampilkan tabel hasil perhitungan distribusi cabai menggunakan metode <i>distribution</i>	[✓]

No	Menu	Fungsi	Aksi	Hasil	Ket
				<i>requirement</i> <i>planning</i>	
4	Data Konsumsi Cabai (Pegawai)	Digunakan untuk melihat, & mengelola data kab/kota (menambah, mengubah, & menghapus)	Memilih menu data kab/kota	Menampilkan tabel data cabai dengan atribut provinsi dan kabupaten	[✓]
			Memilih tombol tambah data kab/kota	Menampilkan form tambah data kab/kota dengan atribut provinsi, kabupaten, tombol simpan dan tombol reset	[✓]
			Memilih tombol reset form data kab/kota	Menampilkan form tambah data kab/kota dengan atribut provinsi, kabupaten, tombol simpan dan tombol reset	[✓]
			Memilih tombol simpan data kab/kota	Menampilkan tabel data kab/kota baru yang telah ditambahkan dengan atribut	[✓]

No	Menu	Fungsi	Aksi	Hasil	Ket
				provinsi dan kabupaten.	
			Memilih tombol edit data kab/kota	Menampilkan form edit data kab/kota dengan rincian data yang terpilih sesuai atribut provinsi, kabupaten, tombol simpan dan tombol reset	[✓]
			Memilih tombol simpan edit data kab/kota	Menampilkan tabel data kab/kota yang telah diubah dengan atribut provinsi, kabupaten,	[✓]
			Memilih tombol hapus data kab/kota	Menampilkan tabel data cabai yang telah dihapus dengan atribut provinsi, kabupaten,	[✓]
5	Pengguna	Digunakan untuk melihat, & mengelola data pengguna	Memilih menu data pengguna	Menampilkan tabel data cabai dengan atribut nama, email,	[✓]

No	Menu	Fungsi	Aksi	Hasil	Ket
		(menambah, mengubah, & menghapus)		hak akses, riwayat login, dan tombol tambah data, tombol edit, tombol hapus	
			Memilih tombol tambah data pengguna	Menampilkan form tambah data pengguna dengan atribut nama, gender, email, password, hak akses, tombol simpan dan tombol reset	[✓]
			Memilih tombol reset form data pengguna	Menampilkan form kosong tambah data pengguna dengan atribut nama, gender, email, password, hak akses, tombol simpan dan tombol reset	[✓]

No	Menu	Fungsi	Aksi	Hasil	Ket
			Memilih tombol simpan data pengguna	Menampilkan tabel data cabai dengan atribut nama, email, hak akses, riwayat login, dan tombol tambah data, tombol edit, tombol hapus	[✓]
			Memilih tombol edit data pengguna	Menampilkan form edit data pengguna dengan rincian data yang terpilih sesuai atribut nama, gender, email, password, hak akses, tombol simpan dan tombol reset	[✓]
			Memilih tombol simpan edit data pengguna	Menampilkan tabel data cabai dengan atribut nama, email, hak akses, riwayat login	[✓]

No	Menu	Fungsi	Aksi	Hasil	Ket
			Memilih tombol hapus data pengguna	Menampilkan tabel data cabai baru yang telah dihapus dengan atribut nama, email, hak akses, riwayat login,	[✓]
6	Logout	Digunakan admin untuk keluar dari sistem	Memilih tombol logout	Logout berhasil dan menampilkan halaman login	[✓]

E.2. *Black Box Testing* Petani

Pengujian menggunakan *black box* untuk akses sistem yang dilakukan oleh petani dapat dilihat pada tabel berikut:

Tabel E. 2 *Black Box Testing* Petani

No	Menu	Fungsi	Aksi	Hasil	Ket
1	Login (Petani)	Digunakan admin untuk masuk dalam sistem	Mengisi Username dan Password pada form login kemudian menekan tombol login	Login berhasil dan menampilkan halaman home admin	[✓]
			Username dan Password tidak diisi	Menampilkan notifikasi "Please fill out this field"	[✓]

No	Menu	Fungsi	Aksi	Hasil	Ket
2	Data Cabai (Admin)	Digunakan untuk melihat data cabai	Memilih menu data cabai	Menampilkan tabel data cabai dengan atribut jenis cabai dan ciri-cirinya	[✓]
3	Data Kab/Kota (Petani)	Digunakan untuk melihat data kab/kota	Memilih menu data kab/kota	Menampilkan tabel data cabai dengan atribut provinsi dan kabupaten	[✓]
4	Data Produksi Cabai (Petani)	Digunakan untuk melihat, data cabai	Memilih menu data cabai	Menampilkan tabel data cabai dengan atribut nama, dan jenis	[✓]
			Memilih tombol logout	Logout berhasil dan menampilkan halaman login	[✓]
6	Logout	Digunakan admin untuk keluar dari sistem	Memilih tombol logout	Logout berhasil dan menampilkan halaman login	[✓]