



**SISTEM PENDUKUNG PENGAMBILAN KEPUTUSAN OPTIMASI
JUMLAH PRODUKSI UNTUK MENINGKATKAN KEUNTUNGAN
MENGUNAKAN METODE *SIMPLEX LINEAR PROGRAMMING*
(STUDI KASUS : RUMAH YOGHURT MALANG)**

SKRIPSI

Oleh

Tiara Anggraini Prasetyo

122410101083

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2017



**SISTEM PENDUKUNG PENGAMBILAN KEPUTUSAN OPTIMASI
JUMLAH PRODUKSI UNTUK MENINGKATKAN KEUNTUNGAN
MENGUNAKAN METODE *SIMPLEX LINEAR PROGRAMMING*
(STUDI KASUS : RUMAH YOGHURT MALANG)**

SKRIPSI

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

Oleh

Tiara Anggraini Prasetyo

122410101083

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER**

2017

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT, yang telah memberikan kemudahan dan kelancaran dalam mengerjakan skripsi ini.
2. Ayahanda Eko Prasetyo dan Ibunda Winari.
3. Kakak Arini Prasetyo Ningtyas, Kakak Anisa Ingesti Prasetyo, dan Adik Shafira Yunita Prasetyo
4. Sahabatku dengan dukungan beserta doanya.
5. Saudara-saudaraku beserta keluarga besar.
6. Guru-guruku sejak taman kanak-kanak hingga perguruan tinggi.
7. Almamater Program Studi Sistem Informasi Universitas Jember.

MOTO

“Kehidupan tidak akan menghendaki sesuatu apapun kepada manusia tanpa ia mau untuk berusaha dan bekerja keras.”

“Yang tidak akan pernah meninggalkanmu selain Tuhanmu, adalah doa ibumu.”

“Hai orang-orang yang beriman, mintalah pertolongan (kepada Allah) dengan sabar dan (mengerjakan) shalat, sesungguhnya Allah beserta orang-orang yang sabar.”
(terjemahan Surat Al-Baqarah ayat 153)¹

¹ Kementerian Agama Republik Indonesia. 2004. Al Qur'an dan Terjemahannya. Semarang: CV. Al Waah

PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Tiara Anggraini Prasetyo

NIM : 122410101083

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Pendukung Pengambilan Keputusan Optimasi Jumlah Produksi Untuk Meningkatkan Keuntungan Menggunakan Metode *Simplex Linear Programming* (Studi Kasus Rumah Yoghurt Malang)”, 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, 17 Maret 2017

Yang menyatakan,

Tiara Anggraini Prasetyo

SKRIPSI

**SISTEM PENDUKUNG PENGAMBILAN KEPUTUSAN OPTIMASI
JUMLAH PRODUKSI UNTUK MENINGKATKAN KEUNTUNGAN
MENGUNAKAN METODE *SIMPLEX LINEAR PROGRAMMING*
(STUDI KASUS : RUMAH YOGHURT MALANG)**

Oleh:

Tiara Anggraini Prasetyo

122410101083

Pembimbing :

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

Dosen Pembimbing Pendamping : Nelly Oktavia Adiwijaya S.Si., MT.

PENGESAHAN PEMBIMBING

Skripsi ini berjudul “Sistem Pendukung Pengambilan Keputusan Optimasi Jumlah Produksi Untuk Meningkatkan Keuntungan Menggunakan Metode *Simplex Linear Programming* (Studi Kasus Rumah Yoghurt Malang)”, telah diuji dan disahkan pada :

Hari, tanggal : Jumat, 31 Maret 2017

Tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh :

Pembimbing I,

Pembimbing II,

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

Nelly Oktavia Adiwijaya S.Si., MT.

NIP. 19670420 1992011001

NIP. 198410242009122008

PENGESAHAN

Skripsi berjudul “Sistem Pendukung Pengambilan Keputusan Optimasi Jumlah Produksi Untuk Meningkatkan Keuntungan Menggunakan Metode *Simplex Linear Programming* (Studi Kasus Rumah Yoghurt Malang)”, telah diuji dan disahkan pada :

Hari, tanggal : Jumat, 31 Maret 2017

Tempat : Program Studi Sistem Informasi Universitas Jember

Penguji I,

Penguji II,

Anang Andrianto, ST., MT

NIP. 196906151997021002

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

NIP. 198101232010121003

Mengesahkan
Ketua Program Studi

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

NIP. 19670420 1992011001

RINGKASAN

Sistem Pendukung Pengambilan Keputusan Optimasi Jumlah Produksi Untuk Meningkatkan Keuntungan Menggunakan Metode *Simplex Linear Programming* (Studi Kasus Rumah Yoghurt Malang); Tiara Anggraini Prasetyo, 122410101083; 2017 : 264 halaman; Program Studi Sistem Informasi Universitas Jember.

Rumah Yoghurt adalah salah satu perusahaan yang menghasilkan produk olahan susu sapi, yaitu Yoghurt dan Keju Mozzarella. Dalam menjalankan usahanya saat ini Rumah Yoghurt sedang memiliki kendala dalam proses produksi, permasalahan yang terjadi antara lain seperti keterbatasan sumber daya bahan baku, keterbatasan waktu produksi dan keterbatasan jumlah mesin yang digunakan dalam proses produksi sehingga perusahaan belum dapat merealisasikan rencana jumlah produksi yang paling optimal untuk mendapatkan penjualan yang maksimal. Berdasarkan masalah tersebut Rumah Yoghurt memerlukan sistem yang dapat memecahkan permasalahan pengalokasian sumber daya yang terbatas untuk mendapatkan keuntungan yang maksimal. Pada penelitian ini dibangun sistem pendukung pengambilan keputusan optimasi jumlah produksi untuk meningkatkan keuntungan. Metode yang digunakan dalam sistem ini adalah metode *simplex linear programming*. Hasil keluaran dari sistem ini adalah banyaknya jumlah produksi keju mozzarella dan yoghurt yang akan diproduksi berdasarkan sumber daya bahan baku dan jam kerja mesin yang ada untuk meningkatkan keuntungan perusahaan.

PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Sistem Pendukung Pengambilan Keputusan Optimasi Jumlah Produksi Untuk Meningkatkan Keuntungan Menggunakan Metode *Simplex Linear Programming* (Studi Kasus Rumah Yoghurt Malang)”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

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

1. Prof. Drs. Slamini, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Prof. Drs. Slamini, M.Comp.Sc., Ph.D., selaku Dosen Pembimbing Utama dan Nelly Oktavia Adiwijaya, S.Si., M.T. selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Windi Eka Yulia Retnani S.Kom., M.T., sebagai dosen pembimbing akademik, yang telah mendampingi penulis sebagai mahasiswa.
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
5. Ayahanda Eko Prasetyo dan Ibunda Winari yang selalu mendukung dan mendoakan;
6. Kakak Arini Prasetyo Ningtyas, Kakak Anisa Ingesti Prasetyo, dan Adik Shafira Yunita Prasetyo;
7. Kakek Alm. Parjono dan kakek Nasir beserta keluarga besar;
8. Alfiyatul Himma beserta keluarga yang turut membantu dalam penyelesaian skripsi;
9. Sahabat seperjuangan yang selalu menemani dan memberikan semangat serta doa Dinda, Alfi, Inez, Dhevi, Rifta, Indira, Bintang, Ratih, Lintang, Hedrin, Haris dan teman-teman seperjuangan FORMATION dan semua mahasiswa Program Studi

Sistem Informasi yang telah menjadi keluarga baru bagi penulis selama menempuh pendidikan S1;

10. Keluarga besar asisten laboratorium Georapich Information Systems (GIS) tahun 2014-2015;
11. Keluarga besar Koperasi Mahasiswa Sistem Informasi (KOPMASI) periode 2013-2014 dan 2014-2015;
12. Rumah singgah Jalan Jawa 4 no.1B, beserta penghuni lainnya yang telah menjadi rumah singgah penulis selama di Jember;
13. Rumah Yoghurt Malang yang selalu memudahkan penulis dalam pengumpulan data;
14. 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, 17 Maret 2017

Penulis

DAFTAR ISI

HALAMAN JUDUL	
PERSEMBAHAN.....	ii
MOTO	iii
PERNYATAAN.....	iv
SKRIPSI.....	v
PENGESAHAN PEMBIMBING	vi
PENGESAHAN.....	vii
RINGKASAN	viii
PRAKATA	ix
DAFTAR ISI.....	xi
DAFTAR TABEL	xvii
DAFTAR GAMBAR.....	xx
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Tujuan.....	3
1.4 Batasan Masalah.....	4
1.5 Sistematika Penulisan.....	4
BAB 2 TINJAUAN PUSTAKA	6
2.1 Penelitian Terdahulu.....	6
2.2 Sistem Pendukung Pengambilan Keputusan	7
2.3 Model Optimasi.....	7
2.4 Linear Programming.....	8
2.5 Perumusan Model Linear Programming	8
2.6 Metode Simpleks.....	10
BAB 3 METODOLOGI PENELITIAN	12

3.1	Jenis Penelitian	12
3.2	Tempat dan Waktu Penelitian	12
3.3	Tahapan Penelitian	12
3.3.1	Tahapan Analisis Kebutuhan.....	13
3.3.2	Desain Sistem	16
3.3.3	Penulisan Kode Program (Implementation)	17
3.3.4	Pengujian Program (<i>Testing</i>).....	18
3.3.5	Pemeliharaan.....	19
BAB 4	PENGEMBANGAN SISTEM.....	20
4.1	Deskripsi Umum Sistem.....	20
4.1.1	SOP (Statement Of Pupose)	20
4.1.2	Fungsi Sistem	21
4.2	Pengumpulan Data.....	22
4.3	Analisis Kebutuhan	25
4.3.1	Kebutuhan Fungsional.....	25
4.3.2	Kebutuhan Non-Fungsional.....	26
4.4	Desain Sistem	26
4.4.1	Business Process.....	26
4.4.2	Usecase Diagram	27
4.4.3	Skenario	32
4.4.4	Sequence Diagram	35
4.4.5	Activity Diagram	38
4.4.6	Class Diagram.....	41
4.4.7	Entity Relationship Diagram (ERD).....	42
4.5	Implementasi	42
4.6	Pengujian Sistem	50
4.6.1	Pengujian white box	50
4.6.2	Pengujian Black Box	68
BAB 5	PEMBAHASAN	70

5.1	Perhitungan Manual Optimasi Produksi Menggunakan Metode Simplex Linear Programming.....	70
5.2	Pengujian Sistem Pendukung Pengambilan Keputusan Optimasi Jumlah Produksi.....	100
5.3	Perbandingan Data Aktual dengan Data Optimasi.....	100
5.4	Analisa Sensitivitas Perhitungan Jumlah Produksi	103
5.5	Implementasi <i>Coding</i> Metode <i>Simplex Linear Programming</i> Pada Sistem Pendukung Pengambilan Keputusan Optimasi Jumlah Produksi	104
5.6	Hasil Pembuatan Sistem Pendukung Pengambilan Keputusan Optimasi Jumlah Produksi Menggunakan Metode Simplex Linear Programming	108
5.6.1	Tampilan Halaman Login	109
5.6.2	Tampilan Update Data Produk	109
5.6.3	Tampilan Mengelola Bahan Baku	110
5.6.4	Tampilan Mengelola Mesin.....	112
5.6.5	Tampilan Mengelola Penggunaan Jam Kerja Mesin.....	113
5.6.6	Tampilan Mengelola Penggunaan Bahan Baku.....	115
5.6.7	Tampilan Halaman Pengoptimalan dengan Linear Programming	116
5.6.8	Tampilan Halaman Cetak Laporan Perhitungan Optimasi Produksi	118
5.6.9	Tampilan Halaman View Optimasi Produksi	118
BAB 6	PENUTUP.....	120
6.1	Kesimpulan.....	120
6.2	Saran	120
DAFTAR PUSTAKA		122
LAMPIRAN.....		123
LAMPIRAN A.	<i>USE CASE</i> SKENARIO.....	123

A.1 Use Case Skenario Login	123
A.2 Use Case Skenario Update Data Produk	124
A.3 Use Case Skenario Mengelola data satuan.....	125
A.4 Use Case Skenario Mengelola data user	127
A.5 Use Case Skenario View Data Produk	130
A. 6 Use Case Skenario Mengelola Bahan Baku	130
A. 7 Use Case Skenario View Bahan Baku	135
A. 8 Use Case Skenario Mengelola Data Mesin	135
A. 9 Use Case Skenario View Data Mesin.....	139
A. 10 Use Case Skenario View Penggunaan Jam Kerja Mesin	140
A. 11 Use Case Skenario Mengelola Penggunaan Jam Kerja Mesin.	141
A. 12 Use Case Skenario Mengelola Data Produksi	145
A. 13 Use Case Skenario View Data Produksi	150
A. 14 Use Case Skenario Mengelola Data Penggunaan Bahan Baku	150
A. 15 Use Case Skenario View Data Penggunaan Bahan Baku	155
A. 16 Use Case Skenario Pengoptimalan dengan <i>Linear Programming</i>	155
A. 17 Use Case Skenario Cetak Laporan Perhitungan optimasi produksi	156
A. 18 Use Case Skenario View Optimasi Produksi - pemilik.....	157
LAMPIRAN B. <i>SEQUENCE</i> DIAGRAM	159
B.1 Sequence Login.....	159
B.2 Sequence Update Produk	160
B. 3 Sequence Mengelola Data Satuan.....	161
B. 4 Sequence View Data Produk	162
B. 5 Sequence View Bahan Baku.....	162
B. 6 Sequence View Data Mesin.....	162
B. 7 Sequence View Penggunaan Jam Kerja Mesin.....	163
B. 8 Sequence View Data Produksi.....	163

B. 9 Sequence View Data Penggunaan Bahan Baku.....	164
B. 10 Sequence View Optimasi Produksi.....	164
B. 11 Sequence Mengelola Bahan Baku	165
B. 12 Sequence Mengelola Data Mesin	166
B. 13 Sequence Mengelola Data Produksi	167
B. 14 Sequence Pengoptimalan dengan <i>Linear Programming</i>	168
B. 15 Sequence Cetak Laporan Perhitungan optimasi produksi	168
LAMPIRAN C. <i>ACTIVITY</i> DIAGRAM	169
C.1 Activity Login.....	169
C.2 Activity Update Produk	169
A. 3 Activity Mengelola Data Satuan.....	170
C. 4 Activity Mengelola User.....	171
C. 5 Activity View Data Produk.....	172
C. 6 Activity View Bahan Baku	172
C. 7 Activity View Data Mesin	173
C. 8 Activity View Penggunaan Jam Kerja Mesin.....	173
C. 9 Activity View Data Produksi.....	174
C. 10 Activity View Data Penggunaan Bahan Baku	174
C. 11 Activity Mengelola Bahan Baku.....	175
C. 12 Activity Mengelola Data Mesin.....	176
C. 13 Activity Mengelola Penggunaan Jam Kerja Mesin	177
C. 14 Activity Mengelola Data Penggunaan Bahan Baku	178
C. 15 Activity Mengelola Data Produksi	179
C. 16 Activity Cetak Laporan Perhitungan optimasi produksi.....	180
C. 17 Activity View Optimasi Produksi - Pemilik.....	180
C. 18 Activity View Optimasi Produksi – Bagian Produksi	181
LAMPIRAN D. KODE PROGRAM.....	182
D.1 Kode program class <i>controller</i> /Bahanbaku	182
D.2 Kode program class <i>controller</i> /C_Login.....	187

D.3 Kode program class <i>controller</i> /C_UserSimpleks	188
D.4 Kode program class <i>controller</i> /Komposisi	194
D.5 Kode program class <i>controller</i> /Mesin	199
D.6 Kode program class <i>controller</i> /Optimasi	204
D.7 Kode program class <i>controller</i> /Pekerjaan	208
D.8 Kode program class <i>controller</i> / Produk.....	213
D.9 Kode program class <i>controller</i> / satuan.....	218
LAMPIRAN E. GAMBAR MESIN	222
LAMPIRAN F. PENGUJIAN BLACK BOX	225
LAMPIRAN G. PROSES PRODUKSI	242

DAFTAR TABEL

Tabel 3.1 Tabel Operasional Variabel Penelitian	14
Tabel 4.1 Data Jenis Bahan Baku	22
Tabel 4.2 Tabel Jumlah Mesin Yang dimiliki.....	23
Tabel 4.3 Elemen Pekerjaan Produk Keju Mozarella	23
Tabel 4.4 Elemen Pekerjaan Produk Yoghurt.....	24
Tabel 4.5 Data Harga Jual Produk	24
Tabel 4.6 Definisi Aktor	29
Tabel 4.7 Definisi Usecase.....	30
Tabel 4.8 Skenario Pengoptimalan dengan <i>Linear Programming</i>	34
Tabel 4.9 Kode Program <i>function</i> <i>index()</i> pada <i>controller</i> optimasi	43
Tabel 4.10 Kode Program <i>function</i> <i>hitungOp()</i> pada <i>controller</i> optimasi	43
Tabel 4.11 Kode Program kelas model optimasi_model	47
Tabel 4.12 Kode program kelas model produk_model.....	49
Tabel 4.13 Test case function <i>hitungOp()</i>	55
Tabel 4.14 Test case function <i>insertdata()</i>	64
Tabel 4.15 Test case function <i>get_nilai_optimasi()</i>	67
Tabel 4.16 Test case function <i>get_jumlahx()</i>	67
Tabel 4.17 Test case function <i>insertindex()</i>	67
Tabel 4.18 Test case function <i>get_min_index()</i>	68
Tabel 4.19 Hasil Pengujian black box.....	68
Tabel 5.1 Tabel Harga Jual Jenis Produk.....	70
Tabel 5.2 Persediaan Bahan Baku pada Bulan Agustus sampai Oktober 2016.....	71
Tabel 5.3 Kebutuhan Bahan Baku untuk Setiap Jenis Produk.....	72
Tabel 5.4 Jumlah Jam Kerja Mesin Bulan Agustus-Oktober 2016.....	72
Tabel 5.5 Pengiterasian Tabel Simplex Agustus 2016 Tahap Dasar	82
Tabel 5.6 Tabel Simplex Agustus 2016 Dasar Pemilihan kolom kunci	83

Tabel 5.7 Tabel Simplex Agustus 2016 Dasar Pemilihan Baris Kunci	84
Tabel 5.8 Tabel Simplex Agustus 2016 kedua iterasi pertama dengan nilai baru	89
Tabel 5.9 Tabel Simplex Agustus 2016 Kedua Iterasi Pertama Dengan Kolom dan Baris Kunci	90
Tabel 5.10 Pengiterasian Tabel Simplex Agustus 2016 Tahap Optimal	92
Tabel 5.11 Pengiterasian Tabel Simplex September 2016 Tahap Dasar	93
Tabel 5.12 Pengiterasian Tabel Simplex September 2016 Tahap Pertama	94
Tabel 5.13 Pengiterasian Tabel Simplex September 2016 Tahap Optimal	95
Tabel 5.14 Pengiterasian Tabel Simplex Oktober 2016 Tahap Dasar	97
Tabel 5.15 Pengiterasian Tabel Simplex Oktober 2016 Tahap Pertama	98
Tabel 5.16 Pengiterasian Tabel Simplex Oktober 2016 Tahap Optimal	99
Tabel 5.17 Tabel hasil perhitungan manual	100
Tabel 5.18 Tabel Perbandingan Perhitungan	100
Tabel 5.19 Data Produksi Aktual Bulan Agustus 2016	101
Tabel 5.20 Data Produksi Aktual Bulan September 2016	101
Tabel 5.21 Data Produksi Aktual Bulan Oktober 2016	101
Tabel 5.22 Perbandingan data aktual bulan Agustus 2016	102
Tabel 5.23 Perbandingan data aktual bulan September 2016	102
Tabel 5.24 Perbandingan data aktual bulan Oktober 2016	102
Tabel 5.25 Kelebihan Kapasitas Produksi Bulan Agustus – Oktober 2016.....	103
Tabel 1 Use Case Skenario Login.....	123
Tabel 2 Use Case Skenario Update Data Produk	124
Tabel 3 Use Case Skenario Mengelola data satuan	125
Tabel 4 Use Case Skenario Mengelola data user	127
Tabel 5 Use Case Skenario View Data Produk.....	130
Tabel 6 Use Case Skenario Mengelola Bahan Baku.....	130
Tabel 7 Use Case Skenario View Bahan Baku	135
Tabel 8 Use Case Skenario Mengelola Data Mesin.....	135
Tabel 9 Use Case Skenario View Data Mesin	139

Tabel 10 Use Case Skenario View Penggunaan Jam Kerja Mesin	140
Tabel 11 Use Case Skenario Mengelola Penggunaan Jam Kerja Mesin	141
Tabel 12 Use Case Skenario Mengelola Data Produksi.....	145
Tabel 13 Use Case Skenario View Data Produksi.....	150
Tabel 14 Case Skenario Mengelola Data Penggunaan Bahan Baku.....	150
Tabel 15 Use Case Skenario View Data Penggunaan Bahan Baku	155
Tabel 16 Use Case Skenario Pengoptimalan dengan Linear Programming	155
Tabel 17 Use Case Skenario Cetak Laporan Perhitungan optimasi produksi.....	156
Tabel 18 Use Case Skenario View Optimasi Produksi - pemilik	157
Tabel 19 Pengujian black box.....	225

DAFTAR GAMBAR

Gambar 2.1 Diagram alir *linear programming* 9

Gambar 3.1 Model *Waterfall* (Sommerville, 2011) 13

Gambar 3.2 Diagram Alir Penerapan Metode *Simplex Linear Programming* 16

Gambar 4.1 Business Process Business Process 27

Gambar 4.2 Usecase Diagram 28

Gambar 4.3 Sequence Pengoptimalan dengan Linear Programming 38

Gambar 4.4 Activity diagram Pengoptimalan dengan *Linear Programming* 39

Gambar 4.5 Class Diagram 41

Gambar 4.6 Entity Relationship Diagram (ERD) 42

Gambar 4.7 Listing Program Method hitungOp 53

Gambar 4.8 Diagram alir *method* hitungOp 54

Gambar 4.9 Diagram alir method insertData() 63

Gambar 4.10 Diagram alir method get_nilai_optimasi() 64

Gambar 4.11 Diagram alir method get_jumlahx() 64

Gambar 4.12 Diagram alir method insertindex() 64

Gambar 4.13 Diagram alir method get_min_index() 64

Gambar 5.1 Kode Program Membuat Tabel Dasar Simplex 105

Gambar 5.2 Kode Program Mencari Baris Kunci 105

Gambar 5.3 Kode Program Menghitung Nilai Index 106

Gambar 5.4 Kode Program Mencari Baris Kunci 106

Gambar 5.5 Kode Program Mengubah Baris Kunci 107

Gambar 5.6 Kode Program Mengubah Baris Selain Baris Kunci 107

Gambar 5.7 Kode program mengulang langkah perhitungan simplex 108

Gambar 5.8 Kode program mengambil nilai optimalisasi 108

Gambar 5.9 Halaman Login 109

Gambar 5.10 Halaman Data Produk 109

Gambar 5.11 Halaman Update data produk 110

Gambar 5.12 Halaman Detail Produk	110
Gambar 5.13 Halaman Bahan Baku.....	111
Gambar 5.14 Halaman Tambah Bahan Baku.....	111
Gambar 5.15 Halaman Update Bahan Baku	111
Gambar 5.16 Halaman Data Mesin.....	112
Gambar 5.17 Halaman Tambah Mesin	112
Gambar 5.18 Halaman Update Mesin.....	113
Gambar 5.19 Halaman Penggunaan Jam Kerja Mesin.....	113
Gambar 5.20 Halaman Tambah Data Penggunaan Jam Kerja Mesin.....	114
Gambar 5.21 Halaman Update Data Penggunaan Jam Kerja Mesin	114
Gambar 5.22 Halaman Detail Data Penggunaan Jam Kerja Mesin	114
Gambar 5.23 Mengelola Data Penggunaan Bahan Baku	115
Gambar 5.24 Halaman Tambah Data Penggunaan Bahan Baku.....	115
Gambar 5.25 Halaman Update Data Penggunaan Bahan Baku	116
Gambar 5.26 Halaman Tombol lihat Persediaan Sumber Daya	116
Gambar 5.27 Halaman Perhitungan Optimasi	117
Gambar 5.28 Halaman Perhitungan Optimasi – Tabel Iterasi	117
Gambar 5.29 Halaman Perhitungan Optimasi – Hasil Akhir.....	117
Gambar 5.30 Halaman Cetak Laporan Perhitungan	118
Gambar 5.31 Halaman view optimasi produksi.....	118
Gambar 5.32 Halaman detail optimasi produksi.....	119
Gambar 1 Sequence Login.....	159
Gambar 2 Sequence Update Produk	160
Gambar 3 Sequence Mengelola Data Satuan	161
Gambar 4 Sequence View Data Produk.....	162
Gambar 5 Sequence View Bahan Baku	162
Gambar 6 Sequence View Data Mesin	162
Gambar 7 Sequence View Penggunaan Jam Kerja Mesin	163
Gambar 8 Sequence View Data Produksi	163

Gambar 9 Sequence View Data Penggunaan Bahan Baku	164
Gambar 10 Sequence View Optimasi Produksi	164
Gambar 11 Sequence Mengelola Bahan Baku	165
Gambar 12 Sequence Mengelola Data Mesin	166
Gambar 13 Sequence Mengelola Data Produksi	167
Gambar 14 Sequence Pengoptimalan dengan Linear Programming	168
Gambar 15 Sequence Cetak Laporan Perhitungan optimasi produksi	168
Gambar 16 Activity Login	169
Gambar 17 Activity Update Produk	169
Gambar 18 Activity Mengelola Data Satuan	170
Gambar 19 Activity Mengelola User	171
Gambar 20 View Data Produk	172
Gambar 21 Activity View Bahan Baku	172
Gambar 22 Activity View Data Mesin	173
Gambar 23 Activity View Penggunaan Jam Kerja Mesin	173
Gambar 24 Activity View Data Produksi	174
Gambar 25 Activity View Data Penggunaan Bahan Baku	174
Gambar 26 Activity Mengelola Bahan Baku	175
Gambar 27 Activity Mengelola Data Mesin	176
Gambar 28 Activity Mengelola Penggunaan Jam Kerja Mesin	177
Gambar 29 Activity Mengelola Data Penggunaan Bahan Baku	178
Gambar 30 Activity Mengelola Data Produksi	179
Gambar 31 Activity Cetak Laporan Perhitungan optimasi produksi	180
Gambar 32 Activity View Optimasi Produksi - Pemilik	180
Gambar 33 Activity View Optimasi Produksi – Bagian Produksi	181
Gambar 34 cheese vat	222
Gambar 35 alat mixing (mixer)	222
Gambar 36 vacuum sealer	223
Gambar 37 Stretcher	223

Gambar 38 Kompur..... 224
Gambar 39 Cup Sealer 224



BAB 1 PENDAHULUAN

Bab ini akan menjelaskan beberapa hal yang melatar belakangi penelitian dan pembuatan aplikasi optimasi produksi untuk meningkatkan keuntungan. Hal-hal yang akan dijelaskan adalah latar belakang penelitian, rumusan masalah, tujuan dan manfaat penelitian, batasan masalah dan sistematika penelitian.

1.1 Latar Belakang

Pertumbuhan dunia usaha pada saat ini sangatlah pesat. Hal ini ditandai dengan banyaknya perusahaan yang muncul dengan berbagai macam usaha dan bidang. Pada hakekatnya setiap perusahaan baik perusahaan besar maupun perusahaan kecil bertujuan untuk mencari keuntungan yang sebesar-besarnya. Dan salah satu aspek penting yang harus dimiliki perusahaan dalam meningkatkan daya saing bisnis yaitu melalui kinerja produksi. Proses produksi merupakan salah satu penentu keberhasilan perusahaan di samping fungsi – fungsi lainnya seperti persediaan, keuangan dan pemasaran.

Sumber daya yang dimiliki perusahaan merupakan faktor penting yang akan menentukan keberhasilan suatu proses produksi. Maka dari itu, penggunaan sumber daya dalam pabrik harus dilakukan secara optimal. Tanpa adanya penggunaan sumber daya yang optimal dalam suatu pabrik, maka akan mempengaruhi pada jumlah hasil produksi serta keuntungan yang akan didapatkan oleh perusahaan tersebut.

Rumah Yoghurt merupakan sebuah perusahaan yang bergerak dalam industri pengolahan susu sapi. Adapun jenis produk hasil olahan susu sapi yang diproduksi yaitu Keju Mozzarella kemasan dan Yoghurt drink kemasan. Saat ini, Rumah Yoghurt sedang memiliki kendala dalam proses produksi, permasalahan yang terjadi antara lain seperti keterbatasan sumber daya bahan baku, keterbatasan waktu produksi dan keterbatasan jumlah mesin yang digunakan dalam proses produksi. Sehingga dalam proses aktifitas produksinya tentu Rumah Yoghurt membutuhkan

perencanaan dan pengalokasian sumber daya yang tersedia untuk mengoptimalkan hasil produksinya guna mendapatkan keuntungan yang maksimal.

Dalam kondisi keterbatasan sumber daya tersebut perusahaan dituntut untuk dapat tetap berupaya dalam mendapatkan keuntungan yang maksimal dengan mengoptimalkan jumlah produksi. Maka dari itu untuk mencapai tujuan produksi yang optimal diperlukannya sebuah metode yang tepat untuk mengatasinya. Metode simpleks Linear Programming, yaitu suatu model umum yang dapat digunakan dalam pemecahan masalah pengalokasian sumber daya yang terbatas untuk mendapatkan keuntungan yang maksimal.

Penelitian sebelumnya yang pernah dilakukan dengan mengimplementasikan metode Linear Programming yaitu dengan judul "*A Linear Programming Technique for the Optimization of the Activities in Maintenance Projects*" yang dilakukan oleh Omar M. Elmabrouk pada tahun 2011. Pada penelitian ini membahas mengenai bagaimana pemrograman linear digunakan untuk membantu seorang manajer proyek untuk menyelesaikan proyek dengan tepat waktu, tepat sasaran serta sesuai dengan anggaran. Karena itu dalam penelitian ini menggunakan metode linear programming yang merupakan sebuah cara yang biasanya digunakan untuk mengambil keputusan dalam situasi tertentu berdasarkan pada sumber daya yang terbatas seperti, waktu yang dibutuhkan oleh tiap aktivitas proyek dan biaya yang terkait. Sedangkan penelitian lainnya dengan judul "*An Integer Linear Programming Model and Heuristics Scheduling EDD for Hardboards Cutting Problem in a Modular Production System*" oleh Parwadi Moengin, Weny Ango Fransiska pada tahun 2011. Penelitian ini membahas mengenai bagaimana sebuah perusahaan manufaktur menggunakan model linear programming untuk menentukan jumlah papan yang akan dipotong oleh masing-masing pola pemotongan. Dengan tujuan meminimalkan jumlah papan yang digunakan dan meminimalkan jumlah limbah. Adapun kendala atau variable pembatas pada penelitian kedua ini yaitu, permintaan produk oleh pelanggan, kapasitas papan yang digunakan, ukuran papan, dan kombinasi pola pemotongan. Setelah menentukan variable pembatas, maka menentukan fungsi

tujuannya biasanya menggunakan symbol “Z”, apakah fungsi tujuannya di minimumkan atau di maksimumkan. Namun pada penelitian ini fungsi tujuannya yaitu meminimalkan jumlah limbah dan jumlah pemotongan papan.

Berdasarkan penelitian tersebut dapat disimpulkan bahwa pemrograman linear merupakan suatu cara untuk menyelesaikan persoalan pengalokasian sumberdaya yang terbatas diantara beberapa aktivitas yang berjalan untuk mencapai tujuan yang diinginkan. Metode pengambilan keputusan menggunakan *linear programming* juga dapat digunakan dalam penelitian ini. Dikarenakan dalam penelitian ini akan meneliti penggunaan sumber daya yang terbatas secara optimal dalam sebuah perusahaan dan juga diharapkan penggunaan metode simpleks Linear Programming ini dapat menghasilkan keuntungan yang maksimal dengan memperhatikan keterbatasan sumber daya yang ada.

1.2 Rumusan Masalah

Berdasarkan dari beberapa permasalahan yang telah diuraikan diatas, maka dapat diambil rumusan masalah sebagai berikut:

1. Bagaimana cara perencanaan jumlah produksi yang optimal untuk meningkatkan keuntungan?
2. Bagaimana merancang sistem pendukung pengambilan keputusan optimasi jumlah produksi untuk meningkatkan keuntungan menggunakan metode *simplex linear programming*?

1.3 Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah:

1. Menentukan perencanaan jumlah produksi yang bisa memberikan keuntungan secara maksimal berdasarkan sumber daya yang dimiliki.
2. Merancang dan membuat sistem penunjang keputusan optimalisasi produksi untuk meningkatkan keuntungan menggunakan metode simpleks linear programming.

1.4 Batasan Masalah

Agar tidak terjadi penyimpangan dalam proses penelitian dan pembuatan sistem pendukung keputusan optimalisasi produksi, maka ditetapkan beberapa batasan permasalahan. Adapun batasan masalah dalam penelitian ini adalah sebagai berikut :

1. Penelitian dilakukan pada bagian produksi Rumah Yoghurt.
2. Pengolahan data optimasi menggunakan model Pemrograman Linier dengan metode *simplex linear programming*.
3. System berbasis website.
4. Menggunakan framework CodeIgniter.
5. Jenis produk yang diamati adalah Keju Mozzarella dan Yoghurt.

1.5 Sistematika Penulisan

Sistematika penulisan dalam penyusunan tugas akhir ini adalah sebagai berikut :

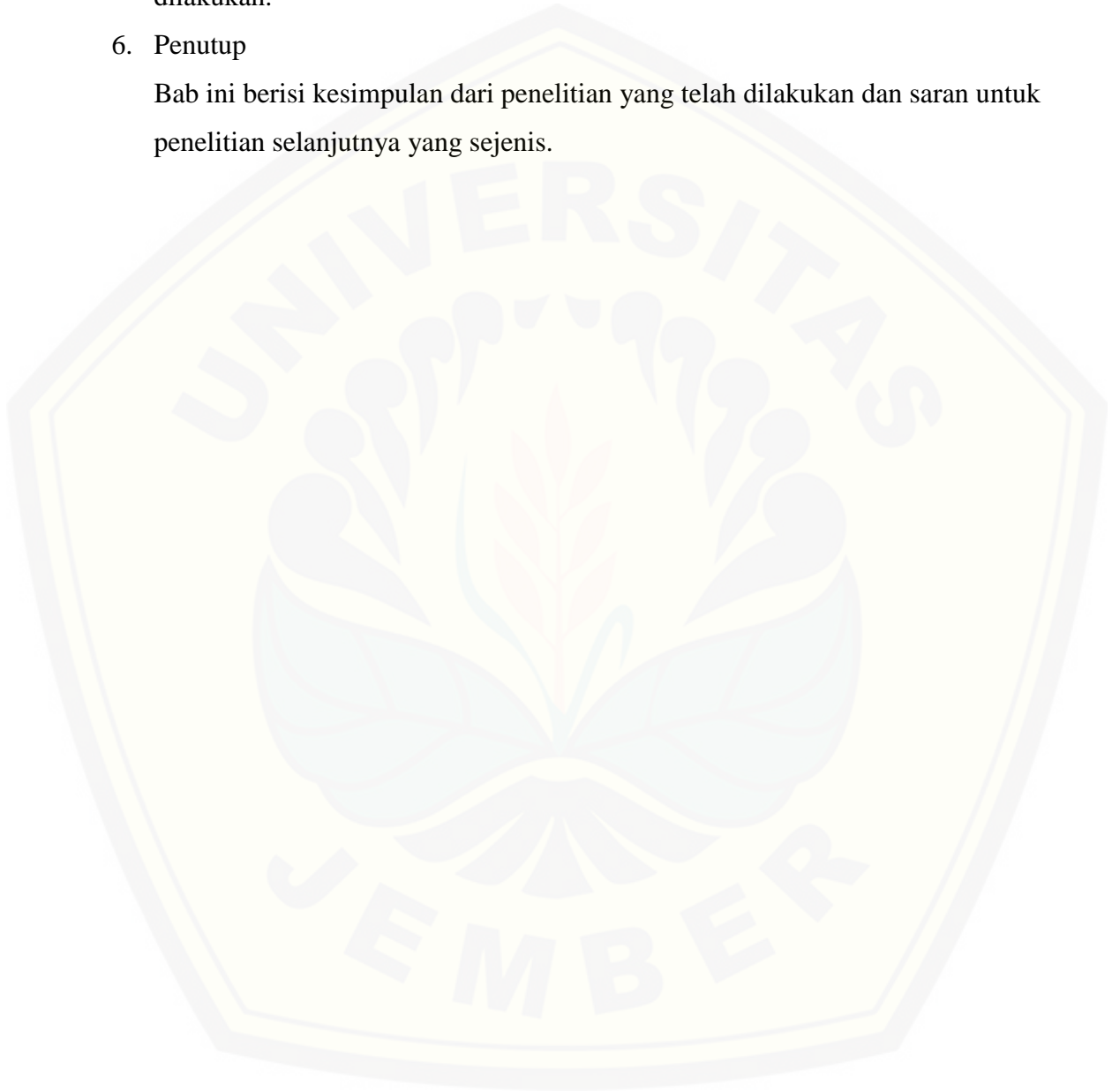
1. Pendahuluan
Bab ini terdiri dari latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah dan sistematika penulisan.
2. Tinjauan Pustaka
Bab ini berisi tentang teori-teori, materi dan informasi yang digunakan dalam penelitian yang dimulai dari kajian pustaka.
3. Metodologi Penelitian
Bab ini menguraikan tentang metode yang akan digunakan selama penelitian. Meliputi pendekatan penelitian, studi literature, sumber data penelitian dan tahapan analisis hingga model perancangan sistem.
4. Pengembangan Sistem
Bab ini menjelaskan tentang analisis dan perancangan sistem yang akan dibangun.

5. Hasil dan Pembahasan

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

6. Penutup

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



BAB 2 TINJAUAN PUSTAKA

Pada bagian ini akan dijelaskan teori-teori dan pustaka yang digunakan dalam penelitian. Teori yang akan dibahas adalah teori tentang sistem pendukung keputusan, model optimasi, *linear programming* dan metode *simplex*.

2.1 Penelitian Terdahulu

Penelitian sebelumnya yang pernah dilakukan dengan mengimplementasikan metode Linear Programming yaitu dengan judul “*A Linear Programming Technique for the Optimization of the Activities in Maintenance Projects*” yang dilakukan oleh (Elmabrouk, 2011). Pada penelitian ini membahas mengenai bagaimana pemrograman linear digunakan untuk membantu seorang manajer proyek untuk menyelesaikan proyek dengan tepat waktu, tepat sasaran serta sesuai dengan anggaran. Karena itu dalam penelitian ini menggunakan metode linear programming yang merupakan sebuah cara yang biasanya digunakan untuk mengambil keputusan dalam situasi tertentu berdasarkan pada sumber daya yang terbatas seperti, waktu yang dibutuhkan oleh tiap aktivitas proyek dan biaya yang terkait.

Penelitian lainnya dengan judul “*An Integer Linear Programming Model and Heuristics Scheduling EDD for Hardboards Cutting Problem in a Modular Production System*” oleh (Moengin, Fransiska, Member, & IAENG, 2011). Penelitian ini membahas mengenai bagaimana sebuah perusahaan manufaktur menggunakan model linear programming untuk menentukan jumlah papan yang akan dipotong oleh masing-masing pola pemotongan. Dengan tujuan meminimalkan jumlah papan yang digunakan dan meminimalkan jumlah limbah. Adapun kendala atau variable pembatas pada penelitian kedua ini yaitu, permintaan produk oleh pelanggan, kapasitas papan yang digunakan, ukuran papan, dan kombinasi pola pemotongan. Setelah menentukan variable pembatas, maka menentukan fungsi tujuannya biasanya menggunakan symbol “Z”, apakah fungsi tujuannya di minimumkan atau di

maksimumkan. Namun pada penelitian ini fungsi tujuannya yaitu meminimalkan jumlah limbah dan jumlah pemotongan papan.

Berdasarkan penelitian tersebut dapat disimpulkan bahwa pemrograman linear merupakan suatu cara untuk menyelesaikan persoalan pengalokasian sumberdaya yang terbatas diantara beberapa aktivitas yang berjalan untuk mencapai tujuan yang diinginkan. Metode pengambilan keputusan menggunakan *linear programming* juga dapat digunakan dalam penelitian ini. Dikarenakan dalam penelitian ini akan meneliti penggunaan sumber daya yang terbatas secara optimal dalam sebuah perusahaan untuk memaksimalkan keuntungan.

2.2 Sistem Pendukung Pengambilan Keputusan

Pengambilan keputusan adalah pemilihan beberapa tindakan alternative yang ada untuk mencapai sebuah tujuan yang telah di tetapkan. Untuk membantu sebuah pengambilan keputusan dibutuhkan sebuah sistem penunjang pengambilan keputusan. Sistem pendukung keputusan merupakan kumpulan dari aplikasi perangkat lunak dan perangkat keras yang saling terintegrasi yang merupakan tulang punggung dari sebuah proses pengambilan keputusan.

2.3 Model Optimasi

Optimasi adalah serangkaian proses untuk mendapatkan sebuah kondisi yang diperlukan untuk mendapatkan hasil terbaik dalam situasi tertentu. Optimalisasi dapat mengidentifikasi penyelesaian terbaik suatu masalah yang diarahkan pada tujuan maksimalisasi atau minimalisasi melalui fungsi tujuan (Rustiani, 2006).

Nilai keuntungan maksimum yang didapatkan dari proses dengan memperhatikan kendala-kendala yang berada di dalam kegiatan, merupakan tujuan dilakukannya optimalisasi. Oleh karena itu, dalam upaya melaksanakan tujuan tersebut, kegiatan produksi berusaha untuk mengalokasikan sumber daya yang terbatas di antara berbagai produk yang bersaing (Rustiani, 2006).

2.4 Linear Programming

Linear Programming (LP) ditemukan oleh George Dantzig. Teknik analisis ini berkembang secara menakjubkan dan mampu memecahkan berbagai masalah (problem solving) yang terdapat dalam kehidupan nyata (Prawirosentono, 2005). Salah satu teknik riset operasional yang relative paling banyak digunakan dalam dunia industri yaitu pemrograman linear atau linear programming (LP). LP biasanya digunakan untuk mencari solusi dari masalah pengalokasian sumber daya yang terbatas secara optimal menggunakan model matematis (Subagyo, Asri, & Handoko, 2011).

Masalah pengalokasian sumber daya akan muncul ketika seseorang diharuskan untuk memilih prioritas setiap kegiatan yang akan dilakukannya, dimana masing-masing kegiatan tersebut membutuhkan sumber daya yang sama dengan jumlah sumber daya yang terbatas. Dalam membangun model dari formulasi persoalan akan digunakan karakteristik-karakteristik yang biasa digunakan dalam persoalan program linier, yaitu :

- a. Variabel Keputusan harus bernilai positif
- b. Fungsi Tujuan
- c. Fungsi Pembatas

2.5 Perumusan Model Linear Programming

Hal utama yang perlu dilaksanakan dalam penyelesaian persoalan menggunakan program linier adalah mengidentifikasi masalah dalam bentuk matematis. Masalah linier programming adalah masalah optimasi bersyarat, yakni pencarian nilai maksimal atau nilai minimum suatu fungsi tujuan berdasarkan dengan kendala-kendala yang harus dipenuhi. Adapun tahapan perhitungan menggunakan model linear programming dapat dilihat pada gambar 2.1.



Gambar . Diagram alir *linear programming*

Dimana secara umum memformulasikan fungsi tujuan dapat dirumuskan sebagai berikut:

$$Z = C_1X_1 + C_2X_2 + C_3X_3 + \dots + C_nX_n \dots \dots \dots \text{Persamaan (2.3)}$$

Keterangan :

Z : Fungsi tujuan

C_n : nilai profit/keuntungan untuk setiap X_n

X_n : variabel keputusan ke-n

Memformulasikan fungsi kendala :

$$\left. \begin{aligned} a_{11}X_1 + a_{12}X_2 + a_{13}X_3 + \dots + a_{1n}X_n &\leq b_1 \\ &\vdots \\ a_{m1}X_1 + a_{m2}X_2 + a_{m3}X_3 + \dots + a_{mn}X_n &\leq b_m \end{aligned} \right\} \dots \text{Persamaan (2.4)}$$

Keterangan :

a_{mn} = Kebutuhan sumber daya m untuk setiap X_n

b_m = Jumlah sumber daya m yang tersedia

n = Banyaknya variabel keputusan mulai dari 1, 2 , ... n

m = Banyaknya jenis sumber daya yang digunakan mulai dari 1, 2 , ... m

2.6 Metode Simpleks

Metode simpleks berjalan melalui serangkaian operasi yang berulang dari satu titik solusi ke titik solusi lainnya yang lebih baik sampai solusi yang paling optimal didapatkan. Adapun langkah-langkah dalam penyelesaian optimasi menggunakan metode simpleks adalah :

1. Mengubah fungsi tujuan dan fungsi pembatas menjadi fungsi implisit. Fungsi implisit yaitu C_1X_1 digeser ke sebelah kiri (Subagyo, Asri, & Handoko, 2011).

Fungsi Tujuan :

$$Z = C_1X_1 + C_2X_2 + C_3X_3 + \dots + C_nX_n \dots\dots\dots \text{persamaan (2.3)}$$

Menjadi

$$Z - C_1X_1 - C_2X_2 - C_3X_3 - \dots - C_nX_n = 0 \dots\dots\dots \text{persamaan (2.5)}$$

Fungsi Pembatas

$$a_{11}X_1 + a_{12}X_2 + a_{13}X_3 + \dots + a_{1n}X_n \leq b_1 \dots\dots\dots \text{persamaan (2.4)}$$

Menjadi

$$a_{11}X_1 + a_{12}X_2 + a_{13}X_3 + \dots + a_{1n}X_n + S_1 = b_1 \dots\dots\dots \text{persamaan (2.6)}$$

2. Menyusun hasil formulasi fungsi ke dalam tabel,
3. Memilih kolom kunci dan baris kunci sebagai dasar pengubah tabel.
 - a. Menentukan kolom kunci dengan cara memilih pada baris Z lalu memilih kolom yang memiliki nilai negative dengan angka terbesar.
 - b. Menentukan baris kunci dengan cara memilih baris dengan nilai indeks terkecil.

$$\text{Indeks} = \frac{\text{Nilai Kanan (NK)}}{\text{Nilai Kolom Kunci}} \dots\dots\dots \text{persamaan (2.7)}$$

4. Menentukan angka kunci, dengan cara mengambil nilai yang merupakan perpotongan antara kolom kunci dan baris kunci.

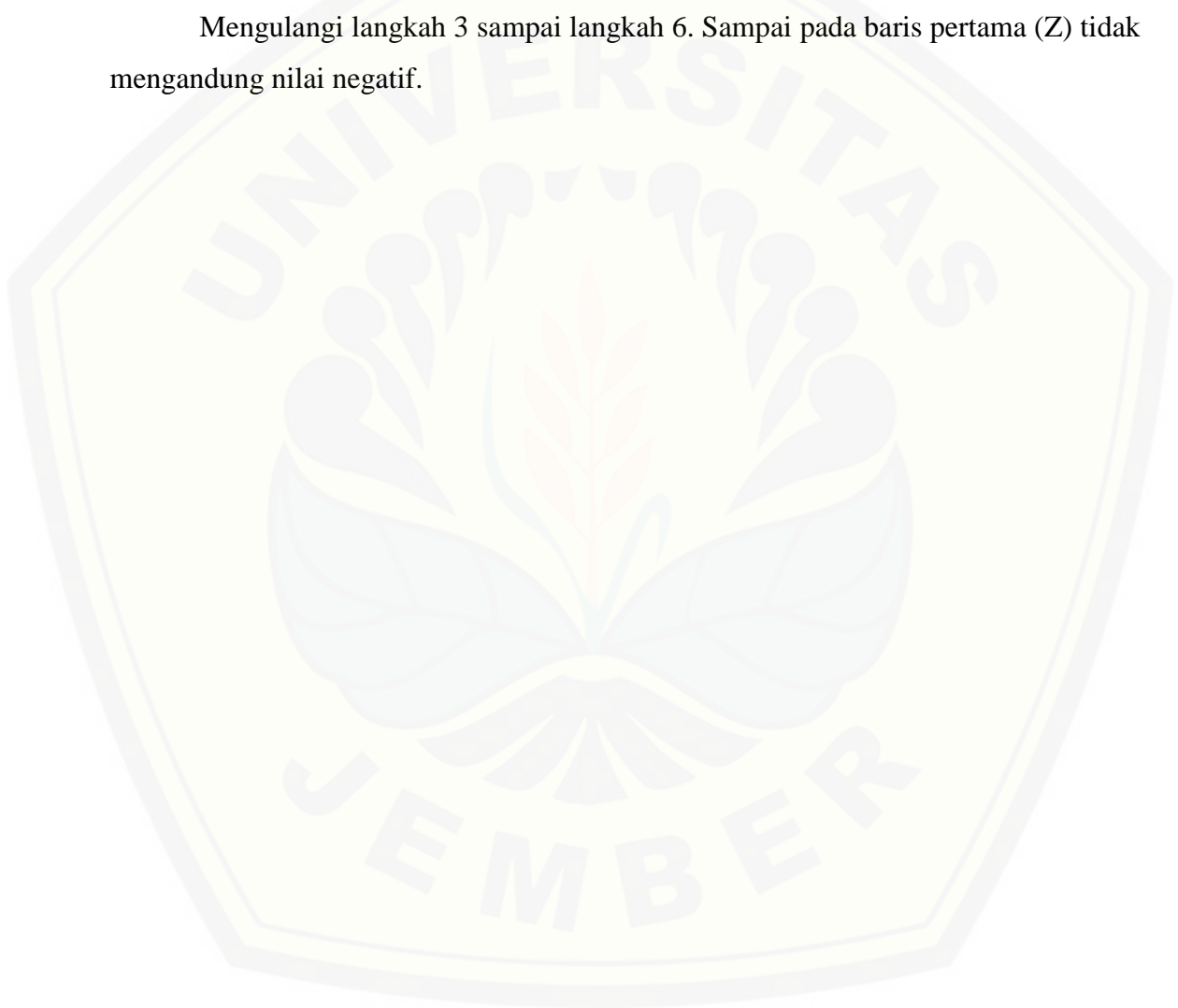
5. Mengubah nilai-nilai baris kunci dengan cara membaginya dengan angka kunci.

$$\text{Baris baru kunci} = \frac{\text{baris kunci}}{\text{angka kunci}} \dots\dots\dots \text{persamaan (2.8)}$$

6. Mengubah nilai-nilai selain baris kunci.

$$\text{Baris baru} = \text{baris lama} - (\text{koefisien angka kolom kunci} \times \text{nilai baris baru kunci}) \dots\dots\dots \text{persamaan (2.9)}$$

Mengulangi langkah 3 sampai langkah 6. Sampai pada baris pertama (Z) tidak mengandung nilai negatif.



BAB 3 METODOLOGI PENELITIAN

Metodologi penelitian merupakan langkah dan prosedur yang akan dilakukan dalam mengumpulkan data atau informasi guna memecahkan permasalahan, dan mengumpulkan informasi yang diperlukan untuk menganalisa data dan mengembangkan sistem pada penelitian ini.

3.1 Jenis Penelitian

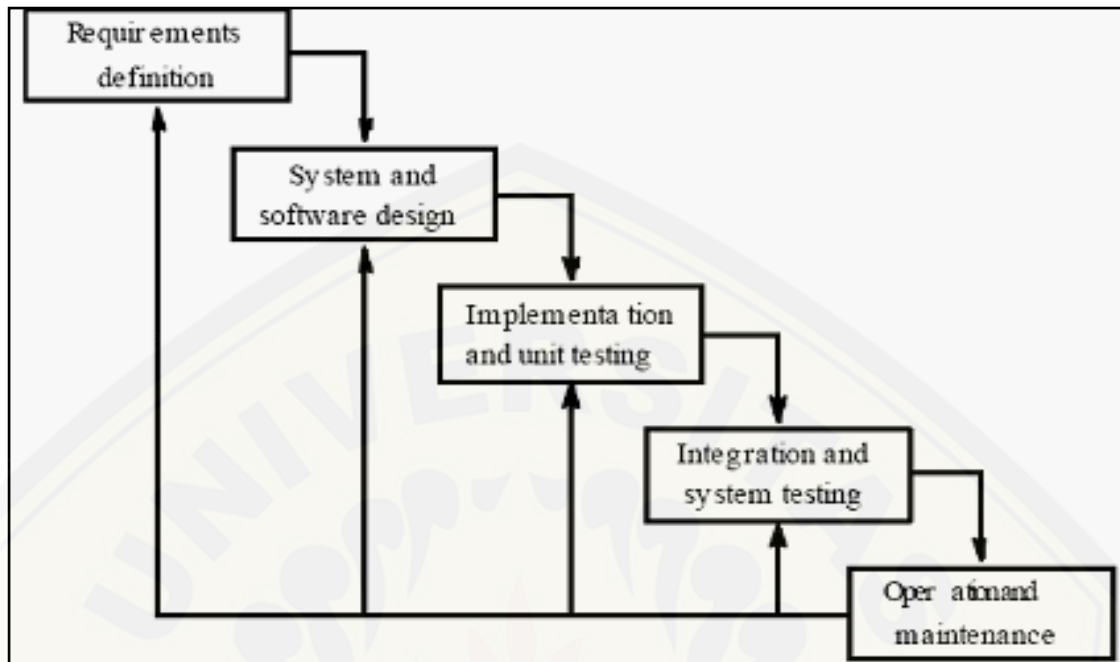
Pada penelitian ini, jenis penelitian yang akan digunakan adalah penelitian pengembangan. Jenis penelitian pengembangan digunakan karena penelitian ini mengembangkan sistem pendukung pengambilan keputusan optimasi jumlah produksi menggunakan metode *simplex linear programming* di Rumah Yoghurt Malang. Metode pengembangan yang digunakan adalah *System Development Life Cycle* (SDLC) dengan menggunakan model *Waterfall*.

3.2 Tempat dan Waktu Penelitian

Tempat penelitian ini dilakukan di Rumah Yoghurt yang berada di Kabupaten Malang. Waktu penelitian dilakukan selama 6 bulan yaitu dimulai pada bulan Mei 2016 sampai dengan bulan Oktober 2016

3.3 Tahapan Penelitian

Tahapan penelitian yang digunakan pada penelitian ini yaitu tahapan penelitian pengembangan sistem yang mengadopsi dari model *waterfall* dengan tahapan-tahapan seperti pada gambar berikut:



Gambar . Model *Waterfall* (Sommerville, 2011)

3.3.1 Tahapan Analisis Kebutuhan

Tahap analisis kebutuhan adalah tahap pengumpulan data kebutuhan fungsional dan non fungsional dari sistem yang akan dibangun, yakni dengan cara melakukan studi literatur yang terkait dengan penelitian, wawancara terhadap pihak terkait. Setelah itu menganalisa data menggunakan metode *simplex linear programming* dan menentukan fitur apa saja yang akan dibuat dalam sistem yang akan dibangun. Metode yang digunakan adalah sebagai berikut:

a. Metode Pengumpulan Data

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

1. Melakukan wawancara kepada bagian produksi Rumah Yoghurt untuk mendapatkan data dan informasi yang dibutuhkan.

2. Melakukan studi literatur, jurnal, media, maupun internet mengenai pembangunan sistem optimasi jumlah produksi.

Jenis data yang digunakan dalam penelitian ini adalah data primer dan data skunder. Data primer merupakan data yang dikumpulkan secara langsung dari objek yang diteliti. Dalam pengumpulan data primer dilakukan menggunakan metode wawancara. Wawancara merupakan salah satu cara yang digunakan dalam pengumpulan data dengan mengajukan pertanyaan-pertanyaan untuk mendapatkan informasi dari narasumber. Dan data yang didapatkan dari hasil wawancara meliputi data mesin, data elemen pekerjaan, data produk, data bahan baku, data komposisi, data hasil penjualan, serta alur proses produksi setiap produk. Data skunder merupakan data yang diperoleh dari studi literatur.

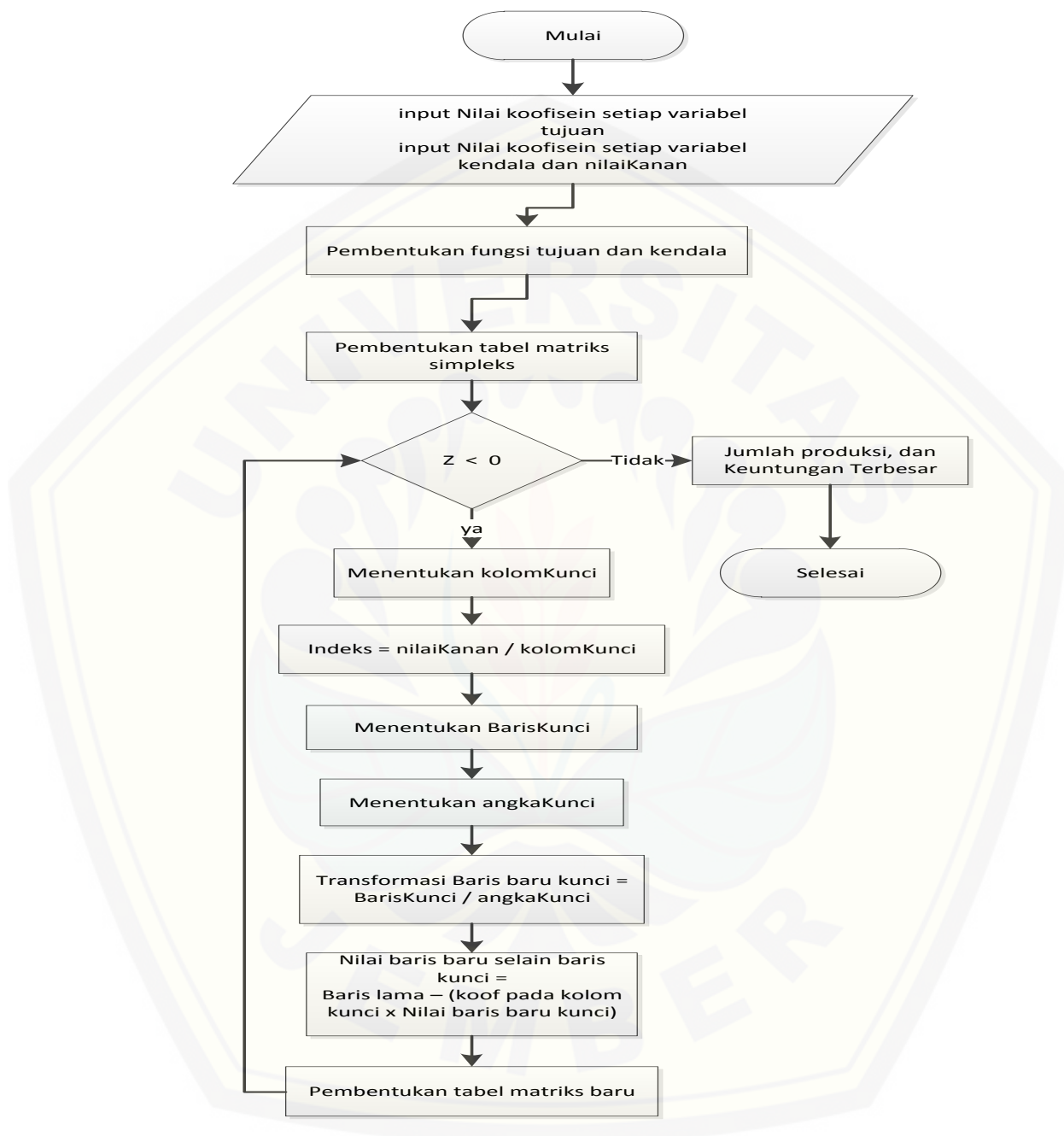
b. Metode Analisis Data

Tahap analisis data dilakukan setelah melakukan pengumpulan data mengenai data produksi. Dan data yang diperoleh akan dianalisis menggunakan metode *simplex linear programming* untuk menghitung jumlah optimal produksi dari setiap produk. Dimana data kapasitas tiap sumber daya, dan waktu proses pembuatan setiap produknya akan menjadi data acuan untuk menghitung serta menentukan jumlah produksi yang paling optimal dengan metode *simplex linear programming*. Pembuatan model optimasi, penentuan batasan-batasan dan perumusan fungsi tujuan akan dijelaskan pada Tabel 3.1 dan flowchart optimasi jumlah produksi dapat dilihat pada Gambar. 3.2

Tabel . Tabel Operasional Variabel Penelitian

Variabel	Subvariabel	Indikator
Menentukan Variabel Keputusan	1. Keju 2. Yoghurt	1. Jumlah Produksi Keju 2. Jumlah Produksi Yoghurt

Menentukan Fungsi Tujuan	$Z = C_1X_1 + C_2X_2$	<ol style="list-style-type: none"> 1. Memaksimalkan keuntungan total yang diperoleh dari produk Keju 2. Memaksimalkan keuntungan total yang diperoleh dari produk Yoghurt
Menentukan Fungsi Kendala	<p>Bahan Baku :</p> $a_{11}X_1 + a_{12}X_2 + a_{13}X_3 + \dots + a_{1n}X_n \leq b_1$ <p>Jam Kerja Mesin</p> $a_{21}X_1 + a_{22}X_2 + a_{23}X_3 + \dots + a_{2n}X_n \leq b_2$	<ol style="list-style-type: none"> 1. Kemampuan menyediakan bahan baku sebesar b_1 2. Kemampuan ketersediaan jam kerja mesin sebesar b_2



Gambar . Diagram Alir Penerapan Metode *Simplex Linear Programming*

3.3.2 Desain Sistem

Pembuatan desain system menggunakan metode *Unified Modeling Language* (UML). Beberapa tahapan yang akan dibuat dalam perancangan sistim ini antara lain:

1. Business Process

Business Process digunakan untuk menggambarkan data yang dibutuhkan sistem sebagai input, lalu diproses menjadi output serta tujuan dari pembuatan sistem.

2. Use Case Diagram

Usecase diagram menggambarkan fitur-fitur apa saja yang tersedia serta aktor yang mempunyai hak akses sistem.

3. Skenario

Skenario menjelaskan alur sistem bekerja dan keadaan yang terjadi ketika terjadi aksi tertentu.

4. *Sequence* Diagram

Sequence merupakan gambaran interaksi antar objek di dalam sistem. Interaksi antar objek yaitu pengguna, *display*, dan sebagian berupa pesan.

5. *Activity* Diagram

Activity Diagram menggambarkan aliran aktivitas dalam sistem, mulai dari alir dimulai, *decision* yang terjadi dan akhir dari alir.

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.

7. *Entity Relationship* Diagram (ERD)

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

3.3.3 Penulisan Kode Program (Implementation)

Penulisan kode program atau *coding* merupakan penerjemahan *design* dalam bahasa yang bisa dikenali oleh komputer serta manajemen basis datanya. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu system. Karena penggunaan komputer akan sangat dimaksimalkan. Beberapa hal yang dilakukan pada tahap ini antara lain :

1. Penulisan kode program(coding) menggunakan bahasa pemrograman *Page Hyper Text Pre-Processor (PHP)* dengan bantuan *framework CodeIgniter, CSS, highchart, jquery, javascript.*
2. Database menggunakan *MySQL.*

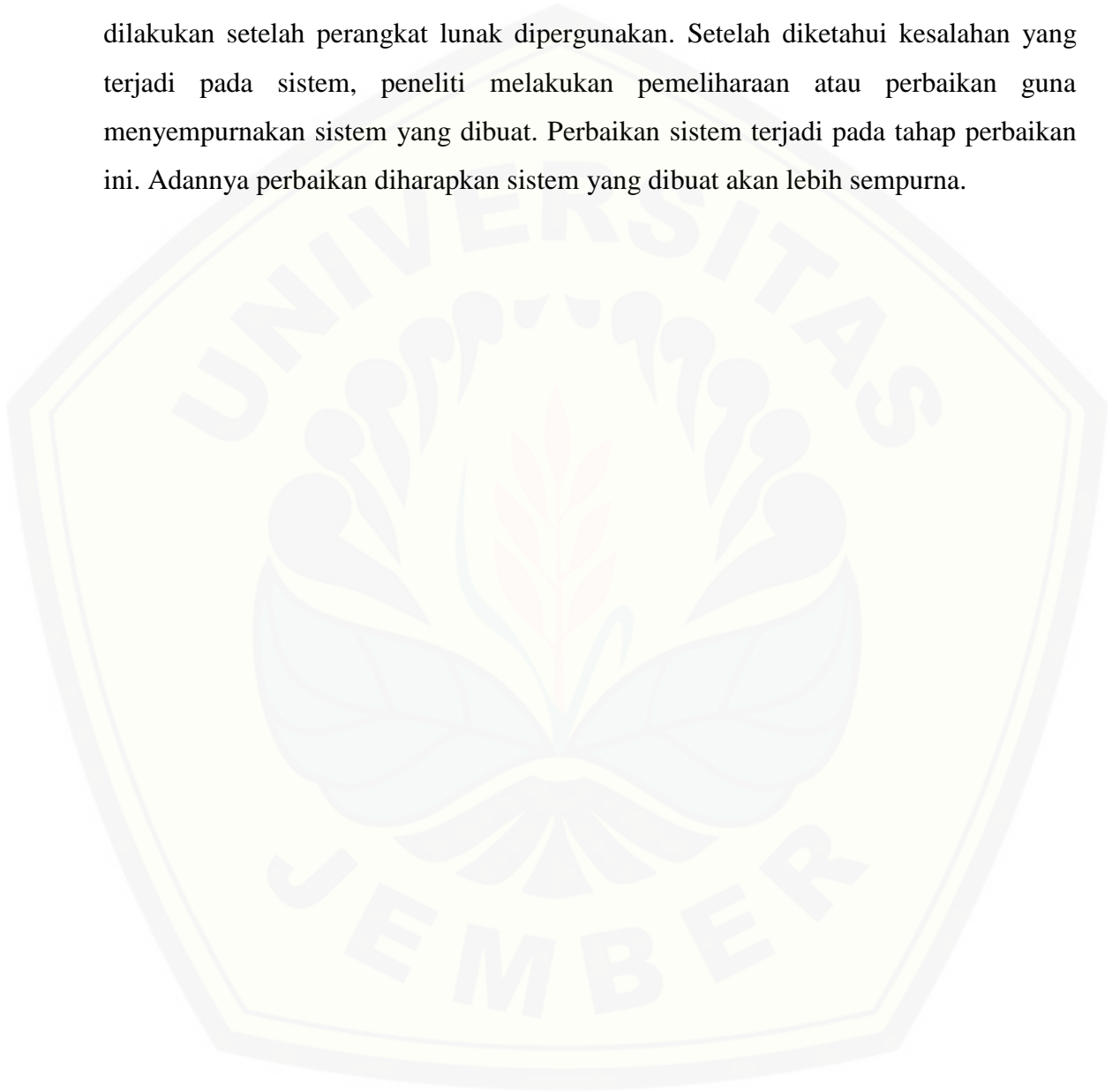
3.3.4 Pengujian Program (*Testing*)

Tahapan dimana system yang telah selesai tahap penulisan kodenya akan diuji kemampuan dan keefektifannya sehingga didapatkan kekurangan dan kelemahan system. Dan pada tahap ini dilakukan uji coba sistem yang telah dibuat dengan pengujian *White Box* dan *Black Box*.

1. *White Box Testing* merupakan cara pengujian dengan melihat ke dalam modul yang telah dibuat dengan kode-kode program yang ada untuk di analisis adanya kesalahan atau tidak. Jika modul yang menghasilkan output yang tidak sesuai dengan proses bisnis yang dilakukan, maka baris-baris program, variabel dan parameter yang terlibat pada unit tersebut akan dicek satu persatu dan diperbaiki, kemudian di-*compile* ulang (Fatta, 2007). Tahapan teknik pengujian *white box* meliputi :
 - a. *ListingProgram*
 - b. Grafik Alir
 - c. Kompleksitas Siklomatik
 - d. Jalur Program Independen
 - e. Pengujian Basis Set
2. *Black Box Testing* merupakan cara pengujian yang hanya dilakukan dengan menjalankan atau mengeksekusi unit atau modul, kemudian diamati apakah hasil dari unit itu sesuai dengan proses bisnis yang diinginkan (Fatta, 2007). Dan pengetahuan khusus dari kode aplikasi, struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Metode ini memfokuskan pada keperluan fungsionalitas dari *software* (Agissa, 2013).

3.3.5 Pemeliharaan

Pemeliharaan sistem diperlukan ketika sistem telah digunakan oleh *user*. Tahap pemeliharaan merupakan bagian paling akhir dari siklus pengembangan dan dilakukan setelah perangkat lunak dipergunakan. Setelah diketahui kesalahan yang terjadi pada sistem, peneliti melakukan pemeliharaan atau perbaikan guna menyempurnakan sistem yang dibuat. Perbaikan sistem terjadi pada tahap perbaikan ini. Adanya perbaikan diharapkan sistem yang dibuat akan lebih sempurna.



BAB 4 PENGEMBANGAN SISTEM

Bab ini akan menguraikan tentang proses perancangan sistem untuk mengimplementasikan metode *simplex linear programming* untuk menentukan optimasi jumlah produksi. Proses perancangan sistem dimulai dari analisis kebutuhan fungsional dan nonfungsional sistem, dilanjutkan dengan pembuatan *bussiness proses*, *usecase diagram*, skenario, *activity diagram*, *sequence diagram*, *class diagram* dan *entity relationship diagram* (ERD).

4.1 Deskripsi Umum Sistem

Deskripsi umum dari sistem pendukung keputusan optimasi jumlah produksi menggunakan metode *simplex linear programming* yang akan dibangun dalam penelitian ini akan dijelaskan lebih detail pada SOP (Statement Of Purpose) sistem dan fungsi sistem.

4.1.1 SOP (Statement Of Pupose)

Sistem pendukung pengambilan keputusan optimasi jumlah produksi menggunakan metode *simplex linear programming* yang dibangun dalam penelitian ini merupakan sistem yang mampu menghitung dan menampilkan hasil perhitungan optimasi jumlah produksi. Hasil perhitungan tersebut berdasarkan analisis data dengan metode *simplex linear programming*. Adapun variabel yang digunakan untuk dijadikan fungsi tujuan dan batasan, yaitu data produk, data mesin, data penggunaan jam kerja mesin, data bahan baku, data komposisi (penggunaan bahan baku). Proses pada fitur optimasi ini menghasilkan kombinasi jumlah produksi yang optimal sesuai dengan batasan-batasan sumber daya yang tersedia seperti dari persediaan bahan baku serta ketersediaan jam kerja mesin selama sebulannya. Sesuai dengan tujuan dibangunnya sistem optimasi jumlah produksi ini menampilkan hasil perhitungan produksi yang optimal dan mencetaknya sebagai laporan perencanaan jumlah produksi, sehingga dapat memudahkan pihak perusahaan dalam menentukan jumlah produksi yang optimal dengan persediaan sumber daya yang terbatas.

Pada aplikasi ini ada tiga aktor yaitu : admin, pemilik dan bagian produksi. Admin bertugas mengelola data user, data satuan dan mengupdate data produk. Aktor bagian produksi berhak mengelola data bahan baku, data mesin, data komposisi, data elemen pekerjaan, data produksi serta menghitung, menyimpan dan mencetak laporan hasil perhitungan optimasi peroduksi menggunakan metode *simplex linear programming*. Sedangkan aktor pemilik memiliki hak akses untuk melihat data bahan baku, mesin, penggunaan jam kerja mesin, penggunaan jam kerja mesin serta melihat data hasil perhitungan optimasi produksi.

Tahap awal sistem ini yaitu admin memasukkan user baru sebagai admin untuk mengelola (menambah, melihat, mengupdate, menghapus) data satuan serta mengupdate data produk. Lalu aktor bagian produksi mengelola (menambah, mengubah, melihat, menghapus) data bahan baku, data komposisi, data mesin dan data pekerjaan. Setelah data penggunaan bahan baku dan mesin berhasil ditambahkan, maka aktor bagian produksi dapat menggunakan fungsi sistem untuk melakukan proses perhitungan optimasi produksi menggunakan metode *simplex linear programming* untuk perencanaan jumlah produksi bulan mendatang dan mencetak laporan perencanaan jumlah produksi yang menggunakan metode *simplex linear programming*.

4.1.2 Fungsi Sistem

Setiap user dalam penelitian ini akan diatur pada fitur login dimana user menggunakan sistem sesuai hak aksesnya masing-masing. Ketika user melakukan login maka sistem akan melakukan autentifikasi username dan password. Selanjutnya sistem akan menampilkan tampilan user sesuai dengan hak akses dari pengguna yang meliputi :

- a. Admin
- b. Bagian Produksi
- c. Pemilik

4.2 Pengumpulan Data

Pengumpulan data pada penelitian ini dilakukan melalui wawancara langsung pada bagian produksi Rumah Yoghurt. Wawancara merupakan salah satu cara yang digunakan dalam pengumpulan data dengan mengajukan pertanyaan-pertanyaan untuk mendapatkan informasi dari narasumber. Data yang dibutuhkan adalah sebagai berikut:

a. Data Bahan Baku

Data bahan baku merupakan data bahan baku apa saja yang digunakan oleh Rumah Yoghurt dalam proses produksinya. Berikut ini adalah data bahan baku apa saja yang digunakan oleh perusahaan dalam memproduksi produknya:

Tabel . Data Jenis Bahan Baku

No.	Nama Bahan Baku
1	Susu sapi segar
2	Susu Skim bubuk
3	Starter Yoghurt
4	Air mineral
5	Tepung maizena
6	Gula pasir
7	Asam Sitrat
8	Enzim Rennet
9	Pengemulsi
10	Garam

b. Data Mesin

Data mesin merupakan data mesin yang mencakup jenis mesin serta jumlah mesin yang dimiliki oleh perusahaan. Berikut ini adalah beberapa jumlah jenis mesin dan penggunaan jam kerja mesin yang digunakan untuk membantu proses produksi:

Tabel . Tabel Jumlah Mesin Yang dimiliki

No.	Elemen Pekerjaan	Nama Mesin	Jumlah Mesin	Waktu penggunaan (detik)	
				Keju	Yoghurt
1.	Pengemasan Yoghurt	Sealer Cup	1	-	5
2.	Pengemasan Keju	Vacum	2	9	-
3.	Pasteurisasi Keju	<i>Cheese vat</i>	1	30	-
4.	Pencampuran Enzim Rennin				
5.	Pemotongan hasil				
6.	Mixing curd	Mixer	1	19	-
7.	Pemuluran	Stracher	1	3	
8.	Pasteurisasi Yoghurt	Kompore	4	-	7
9.	Pencampuran Starter				

c. Data Elemen Pekerjaan

Data Elemen Pekerjaan merupakan data proses tahapan pembuatan setiap jenis produk dari Rumah Yoghurt. Proses produksi dapat dibagi menjadi beberapa elemen pekerjaan. Elemen pekerjaan tersebut dikerjakan secara manual maupun menggunakan mesin. Berikut ini adalah elemen pekerjaan:

Tabel . Elemen Pekerjaan Produk Keju Mozzarella

No.	Elemen Pekerjaan	Pengerjaan
1.	Pasteurisasi	Mesin
2.	Pencampuran Enzim Rennin	Mesin
3.	Pemotongan hasil / pembuangan <i>whey</i>	Mesin
4.	<i>Mixing curd</i>	Mesin

5.	Pemuluran/Stracher	Mesin
6.	Pengemasan	Mesin

Tabel . Elemen Pekerjaan Produk Yoghurt

No.	Elemen Pekerjaan	Pengerjaan
1.	Pasteurisasi	Mesin
2.	Pencampuran Starter	Mesin
3.	Inkubasi	Manual
4.	Pengemasan	Mesin

d. Data Produk

Data produk merupakan data jenis produk yang di produksi oleh perusahaan Rumah Yoghurt. Rumah Yoghurt menjual produknya dengan harga :

Tabel . Data Harga Jual Produk

No.	Produk	Harga Jual Produk
1.	Keju Mozarella	Rp. 87.000 / Kg
2.	Yoghurt	Rp. 3500 / Cups

e. Data Komposisi

Data komposisi merupakan data komposisi yang digunakan oleh Rumah Yoghurt dalam proses produksi tiap jenis produknya. Berikut ini adalah data kebutuhan bahan baku untuk memproduksi jenis-jenis produk, yang didapatkan dari hasil wawancara:

1. Bahan Baku Untuk Keju Mozarella :

- a. Susu Sapi Segar : 8 liter
- b. Air Mineral : 0.025 liter
- c. Garam : 2.8 gram
- d. Asam Sitrat : 0.16 liter
- e. Enzim Renin : 0.002 liter

- f. Pengemulsi : 0.025 liter
2. Bahan Baku Untuk Yoghurt :
- a. Susu Sapi Segar : 0.14 liter
 - b. Susu Skim Bubuk : 0.42 gram
 - c. Starter Yoghurt : 0.007 liter
 - d. Air mineral : 0.14 liter
 - e. Tepung Maizena : 0.42 gram
 - f. Gula Pasir : 5 gram

4.3 Analisis Kebutuhan

Berdasarkan metode pengembangan sistem model *waterfall*, tahapan awal yang dilakukan adalah tahapan analisis kebutuhan. Tahapan ini dilakukan analisis terhadap kebutuhan-kebutuhan dari sistem yang dibangun, baik berupa kebutuhan fungsional maupun kebutuhan non-fungsional.

4.3.1 Kebutuhan Fungsional

Dari data yang terkumpul lalu dianalisa. Maka diperoleh kebutuhan fungsional dan non fungsional yang dibutuhkan untuk membangun aplikasi. 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 apa saja yang nantinya dilakukan oleh sistem antara lain :

1. Sistem dapat meng*update* data harga jual produk yang dihasilkan.
2. Sistem dapat mengelola data Mesin mesin meliputi *insert, update, delete*.
3. Sistem dapat mengelola data Penggunaan Jam Kerja Mesin mesin meliputi *insert, update, delete*.
4. Sistem dapat mengelola data Bahan Baku mesin meliputi *insert, update, delete*.
5. Sistem dapat mengelola data Penggunaan Bahan Baku mesin meliputi *insert, update, delete*.

6. Sistem dapat mengelola data jumlah produksi dari setiap produk perbulannya.
7. Sistem dapat menghitung jumlah optimasi produksi menggunakan metode Simplex Linear Programming.
8. Sistem dapat menampilkan hasil perhitungan jumlah produksi yang optimal dan mencetak laporan optimasi produksi.

4.3.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan hal yang dibutuhkan oleh sistem untuk mendukung aktifitas sistem sesuai dengan kebutuhan fungsional yang ada. Adapun kebutuhan non-fungsional dari sistem ini yaitu :

1. Sistem memiliki batasan hak akses pengguna dengan menggunakan username dan password
2. Sistem memiliki user interface yang user friendly.

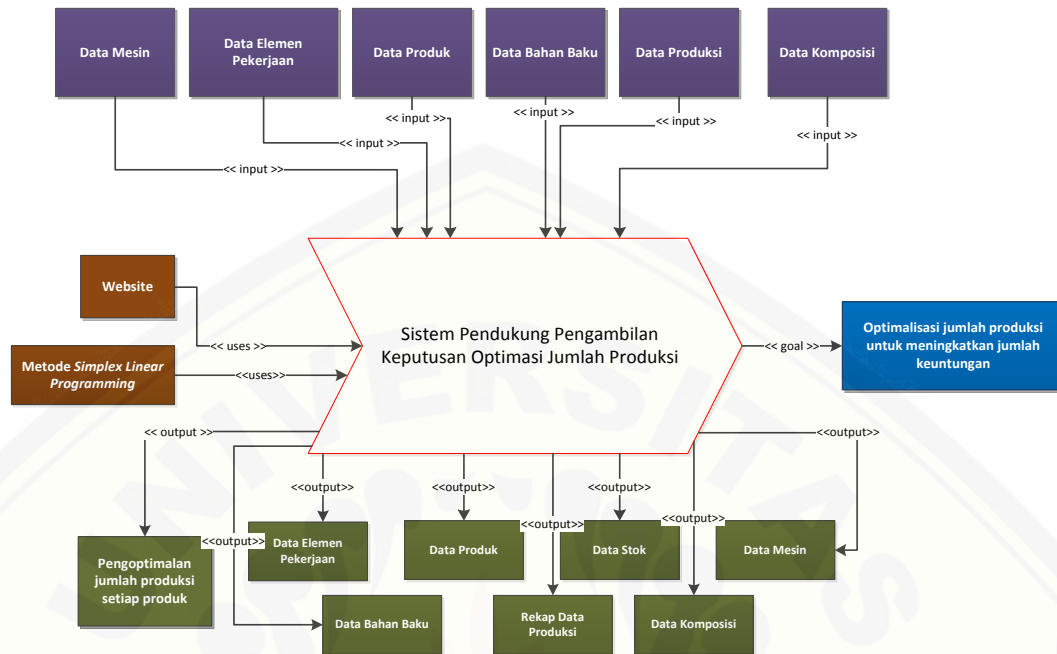
4.4 Desain Sistem

Desain sistem yang dibuat meliputi business process, use case diagram, use case scenario, sequence diagram, activity diagram, class diagram, dan ERD.

4.4.1 Business Process

Bussines process merupakan diagram yang menggambarkan kebutuhan data yang dibutuhkan oleh sistem. Bussinens process didalamnya terdiri atas beberapa poin diantaranya:

1. Input : Data yang dimasukkan kedalam sistem
2. Output : Data yang dihasilkan oleh sistem
3. Goal : Tujuan dibangun suatu sistem
4. Used : platform yang menjadi basis sistem
5. Process : Sistem yang bekerja

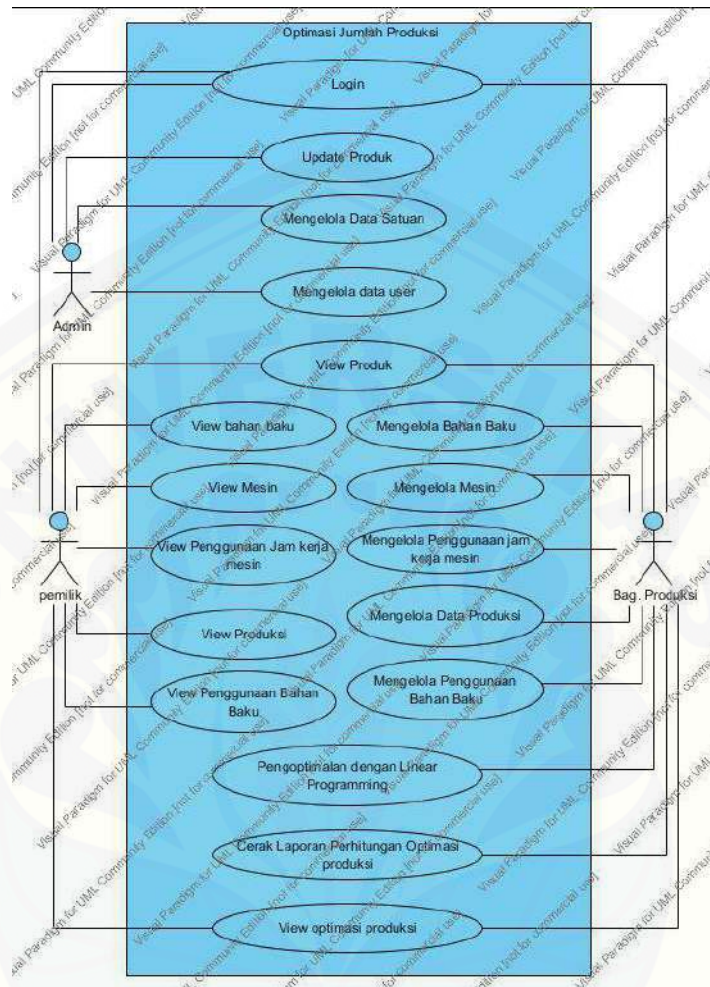


Gambar . Business Process Business Process

Gambar 4.1 merupakan gambar business proses. Business proses menjelaskan tentang data-data yang diperlukan untuk diinputkan lalu menghasilkan output dan hal yang akan dicapai oleh sebuah aplikasi pengoptimalisasi jumlah produksi.

4.4.2 Usecase Diagram

Usecase Diagram merupakan diagram yang digunakan untuk menjelaskan apa saja fitur yang ada dalam aplikasi pengoptimalisasi jumlah produksi. *Usecase diagram* dapat dilihat pada gambar 4.2



Gambar . Usecase Diagram

Usecase diagram aplikasi pengoptimalisasi jumlah produksi di Rumah Yoghurt Malang terdiri dari delapan belas usecase dengan dua aktor. Penjelasan tentang definisi actor dan definisi usecase dalam usecase diagram pada gambar 4.2 akan dijelaskan dibawah ini.

1. Definisi Aktor

Definisi actor merupakan penjelasan tentang aktor-aktor sebagai pengguna dari sistem yang akan dibangun. Terdapat 3 aktor seperti yang akan dijelaskan pada Tabel 4.6

Tabel . Definisi Aktor

No	Aktor	Deskripsi
1.	Admin	Aktor admin mempunyai hak akses untuk mengatur hak akses yang menggunakan sistem ini. Aktor admin dapat melakukan <i>login</i> dan <i>logout</i> , mengelola data user, mengelola data satuan dan mengupdate data jenis produk yang akan digunakan dalam perhitungan optimasi jumlah produksi.
2.	Pegawai Produksi	Aktor ini memiliki hak akses untuk mengontrol jumlah produksi dengan cara melakukan perhitungan perencanaan produksi yang baik menggunakan metode <i>simplex linear programming</i> dan mencetak laporan hasil perhitungan optimasi produksi. Selain itu juga memiliki hak akses untuk mengentry atau menginputkan data-data yang di perlukan dalam proses pengoptimalan produksi seperti data bahan baku, data komposisi, data mesin dan data jam kerja mesin.
3.	Pemilik	Aktor ini memiliki hak akses untuk melihat data-data yang diperlukan dalam proses pengoptimalan produksi serta melihat hasil perhitungan optimasi produksi.

2. Definisi Use Case

Definisi *use case* merupakan penjelasan dari masing-masing *use case* atau fitur-fitur dari sistem yang akan dibangun. Terdapat 19 usecase yang akan dijelaskan pada tabel 4.7.

Tabel . Definisi Usecase

No.	Usecase	Penjelasan
UC-01	Login	Merupakan <i>usecase</i> yang menggambarkan proses user masuk ke halaman utama sesuai dengan hak akses.
UC-02	Update Produk	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan dan mengubah data produk.
UC-03	Mengelola Data Satuan	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan, mengubah, menambah dan menghapus data satuan.
UC-04	Mengelola Data User	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan, mengubah, menambah, dan menghapus data user yang berhak mengakses sistem ini.
UC-05	View Data Produk	Merupakan <i>usecase</i> yang digunakan untuk melihat data produk yang ada.
UC-06	Mengelola Bahan Baku	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan, mengubah, menambah, dan menghapus data bahan baku.
UC-07	View Bahan Baku	Merupakan <i>usecase</i> yang digunakan hanya untuk melihat data bahan baku yang ada.
UC-08	Mengelola Mesin	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan, mengubah, menambah, dan menghapus data mesin.
UC-09	View Mesin	Merupakan <i>usecase</i> yang digunakan hanya untuk melihat data mesin.
UC-10	View	Merupakan <i>usecase</i> yang digunakan hanya

	Penggunaan Jam Kerja Mesin	untuk melihat penggunaan jam kerja mesin.
UC-11	Mengelola Penggunaan Jam Kerja Mesin	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan, mengubah, menambah, dan menghapus data penggunaan jam kerja mesin.
UC-12	Mengelola data produksi	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan, mengubah, menambah, dan menghapus data produksi.
UC-13	View Data Produksi	Merupakan <i>usecase</i> yang digunakan hanya untuk melihat data produksi.
UC-14	Mengelola Data Penggunaan Bahan Baku	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan, mengubah, menambah dan menghapus data penggunaan bahan baku.
UC-15	View Data Penggunaan Bahan Baku	Merupakan <i>usecase</i> yang digunakan untuk melihat data penggunaan bahan baku.
UC-16	Pengoptimalan dengan <i>Linear Programming</i>	Merupakan <i>usecase</i> yang digunakan untuk menghitung pengoptimalan jumlah produksi untuk mrningkatkan keuntungan yang menggunakan metode <i>simplex linear programming</i> .
UC-17	Cetak laporan perhitungan optimasi produksi	Merupakan <i>usecase</i> yang digunakan untuk mencetak laporan hasil perhitungan optimasi produksi
UC-18	View optimasi produksi	Merupakan <i>usecase</i> yang digunakan untuk melihat hasil perhitungan optimasi produksi

4.4.3 Skenario

Skenario menjelaskan alur sistem bekerja dan keadaan yang terjadi ketika terjadi aksi tertentu. Berikut ini skenario dari optimasi jumlah produksi menggunakan metode linear programming:

1) Skenario Login

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case login* dijelaskan pada lampiran A.

2) Skenario Update Produk

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case Update Produk* dijelaskan pada lampiran A.

3) Skenario Mengelola Data Satuan

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case Kelola Data Satuan* dijelaskan pada lampiran A.

4) Skenario Mengelola User

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case Kelola Data User* dijelaskan pada lampiran A.

5) Skenario View Data Produk

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case View Data Produk* dijelaskan pada lampiran A.

6) Skenario Mengelola Bahan Baku

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case Kelola Data Bahan Baku* dijelaskan pada lampiran A.

7) Skenario View Bahan Baku

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case View Data Bahan Baku* dijelaskan pada lampiran A.

8) Skenario Mengelola Data Mesin

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case Kelola Data Mesin* dijelaskan pada lampiran A.

9) Skenario View Data Mesin

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case* View Data Mesin dijelaskan pada lampiran A.

10) Skenario View Penggunaan Jam Kerja Mesin

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case* View Penggunaan Jam Kerja Mesin dijelaskan pada lampiran A.

11) Skenario Mengelola Penggunaan Jam Kerja Mesin

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case* Mengelola Penggunaan Jam Kerja Mesin dijelaskan pada lampiran A.

12) Skenario Mengelola Data Produksi

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case* Kelola Data Produksi dijelaskan pada lampiran A.

13) Skenario View Data Produksi

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case* View Data Produksi dijelaskan pada lampiran A.

14) Skenario Mengelola Data Penggunaan Bahan Baku

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case* Kelola Data Penggunaan Bahan Baku dijelaskan pada lampiran A.

15) Skenario View Data Penggunaan Bahan Baku

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case* View Data Penggunaan Bahan Baku dijelaskan pada lampiran A.

16) Skenario Pengoptimalan dengan *Linear Programming*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif usecase pengoptimalan dengan *linear programming* dapat dilihat pada tabel 4.8

Tabel . Skenario Pengoptimalan dengan *Linear Programming*

Nama Usecase	Pengoptimalan dengan <i>Linear Programming</i>
Aktor	Bagian Produksi
Deskripsi Singkat	Perhitungan Optimasi jumlah produksi dengan <i>Linear Programming</i>
Prekondisi	Pemilik memilih menu Pengoptimalan dengan <i>Linear Programming</i>
Prakondisi	Pemilik berhasil menghitung optimasi produksi dengan sistem dan melihat hasil perhitungan optimasi
Skenario Normal Melihat Data Produk	
Aktor	Sistem
1. Memilih menu Optimasi	
	1. Menampilkan halaman cek dengan tombol lihat persediaan bahan baku, tombol lihat persediaan jam kerja mesin dan tombol next perhitungan
2. Klik tombol next perhitungan	
	3. Mengambil data user dari database
	4. Melakukan perhitungan optimasi produksi dengan metode <i>simplex Linear Programming</i>
	5. Menampilkan halaman hasil perhitungan optimasi dan tombol simpan
Skenario Alternatif - klik lihat persediaan bahan baku	
3. Klik lihat persediaan bahan baku	
	4. Mengambil data user dari database
	5. Mengambil data bahanbaku dari database
	6. Menampilkan halaman bahan baku
Skenario Alternatif - klik lihat persediaan jam kerja mesin	
5. Klik lihat persediaan jam kerja mesin	
	6. Mengambil data user dari database
	7. Mengambil data jam kerja mesin dari database
	8. Menampilkan halaman jam kerja mesin

17) Skenario Cetak Laporan Perhitungan optimasi produksi

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case* Cetak Laporan Perhitungan optimasi produksi dijelaskan pada lampiran A.

18) Skenario View Optimasi Produksi

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case* View Optimasi Produksi dijelaskan pada lampiran A.

4.4.4 Sequence Diagram

Sequence diagram adalah yang digunakan untuk menggambarkan interaksi yang terjadi antar objek di dalam sistem yang disusun pada sebuah urutan dan rangkaian waktu.

1) Sequence Login

Penggambaran sequence diagram login digunakan untuk menjelaskan fungsi atau method yang akan dibuat seperti yang ditunjukkan pada lampiran B.

2) Sequence Update Produk

Penggambaran sequence diagram Update Produk digunakan untuk menjelaskan fungsi atau method yang akan dibuat seperti yang ditunjukkan pada lampiran B.

3) Sequence Mengelola Data Satuan

Penggambaran sequence diagram Mengelola Data Satuan digunakan untuk menjelaskan fungsi atau method yang akan dibuat seperti yang ditunjukkan pada lampiran B.

4) Sequence Mengelola User

Penggambaran sequence diagram Mengelola User digunakan untuk menjelaskan fungsi atau method yang akan dibuat seperti yang ditunjukkan pada lampiran B.

5) Sequence View Data Produk

Penggambaran sequence diagram login digunakan untuk menjelaskan fungsi atau method yang akan dibuat seperti yang ditunjukkan pada lampiran B.

6) Sequence Mengelola Bahan Baku

Penggambaran sequence diagram Mengelola Bahan Baku digunakan untuk menjelaskan fungsi atau method yang akan dibuat seperti yang ditunjukkan pada lampiran B.

7) Sequence View Bahan Baku

Penggambaran sequence diagram View Bahan Baku digunakan untuk menjelaskan fungsi atau method yang akan dibuat seperti yang ditunjukkan pada lampiran B.

8) Sequence Mengelola Data Mesin

Penggambaran sequence diagram Mengelola Data Mesin digunakan untuk menjelaskan fungsi atau method yang akan dibuat seperti yang ditunjukkan pada lampiran B.

9) Sequence View Data Mesin

Penggambaran sequence diagram View Data Mesin digunakan untuk menjelaskan fungsi atau method yang akan dibuat seperti yang ditunjukkan pada lampiran B.

10) Sequence View Penggunaan Jam Kerja Mesin

Penggambaran sequence diagram View Penggunaan Jam Kerja Mesin digunakan untuk menjelaskan fungsi atau method yang akan dibuat seperti yang ditunjukkan pada lampiran B.

11) Sequence Mengelola Penggunaan Jam Kerja Mesin

Penggambaran sequence diagram Penggunaan Jam Kerja Mesin digunakan untuk menjelaskan fungsi atau method yang akan dibuat seperti yang ditunjukkan pada lampiran B.

12) Sequence Mengelola Data Produksi

Penggambaran sequence diagram Mengelola Data Produksi digunakan untuk menjelaskan fungsi atau method yang akan dibuat seperti yang ditunjukkan pada lampiran B.

13) Sequence View Data Produksi

Penggambaran sequence diagram View Data Produksi digunakan untuk menjelaskan fungsi atau method yang akan dibuat seperti yang ditunjukkan pada lampiran B.

14) Sequence Mengelola Data Penggunaan Bahan Baku

Penggambaran sequence diagram Mengelola Data Penggunaan Bahan Baku digunakan untuk menjelaskan fungsi atau method yang akan dibuat seperti yang ditunjukkan pada lampiran B.

15) Sequence View Data Penggunaan Bahan Baku

Penggambaran sequence diagram View Data Penggunaan Bahan Baku digunakan untuk menjelaskan fungsi atau method yang akan dibuat seperti yang ditunjukkan pada lampiran B.

16) Sequence Cetak Laporan Perhitungan optimasi produksi

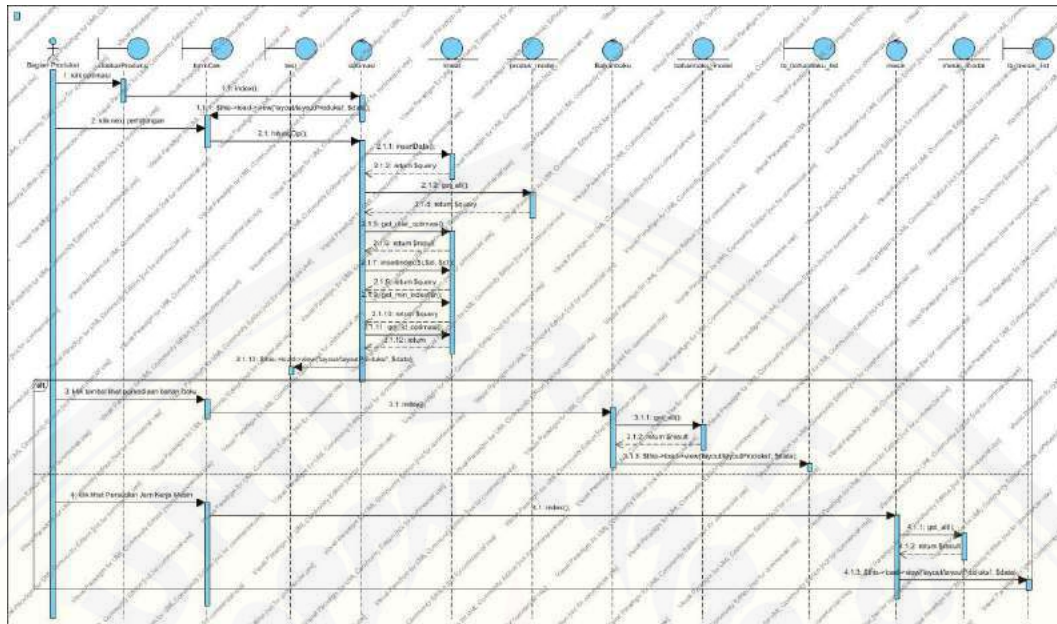
Penggambaran sequence diagram Cetak Laporan Perhitungan optimasi produksi digunakan untuk menjelaskan fungsi atau method yang akan dibuat seperti yang ditunjukkan pada lampiran B.

17) Sequence View Optimasi Produksi

Penggambaran sequence diagram View Optimasi Produksi digunakan untuk menjelaskan fungsi atau method yang akan dibuat seperti yang ditunjukkan pada lampiran B.

18) Sequence Pengoptimalan dengan *Linear Programming*

Penggambaran sequence diagram Pengoptimalan dengan *Linear Programming* digunakan untuk menjelaskan fungsi atau method yang akan dibuat seperti yang ditunjukkan pada Gambar 4.3



Gambar . Sequence Pengoptimalan dengan Linear Programming

4.4.5 Activity Diagram

Activity diagram merupakan diagram tentang aktivitas yang dapat dilakukan oleh masing-masing aktor dan aktivitas sistem setelah menerima perlakuan dari actor tersebut. Activity diagram menggambarkan bagaimana fitur dapat berfungsi dari awal hingga akhir fitur tersebut dijalankan.

1) Activity diagram Login

Penggambaran Activity diagram login ditunjukkan pada lampiran C.

2) Activity diagram Update Produk

Penggambaran Activity diagram Update Produk ditunjukkan pada lampiran C.

3) Activity diagram Mengelola Data Satuan

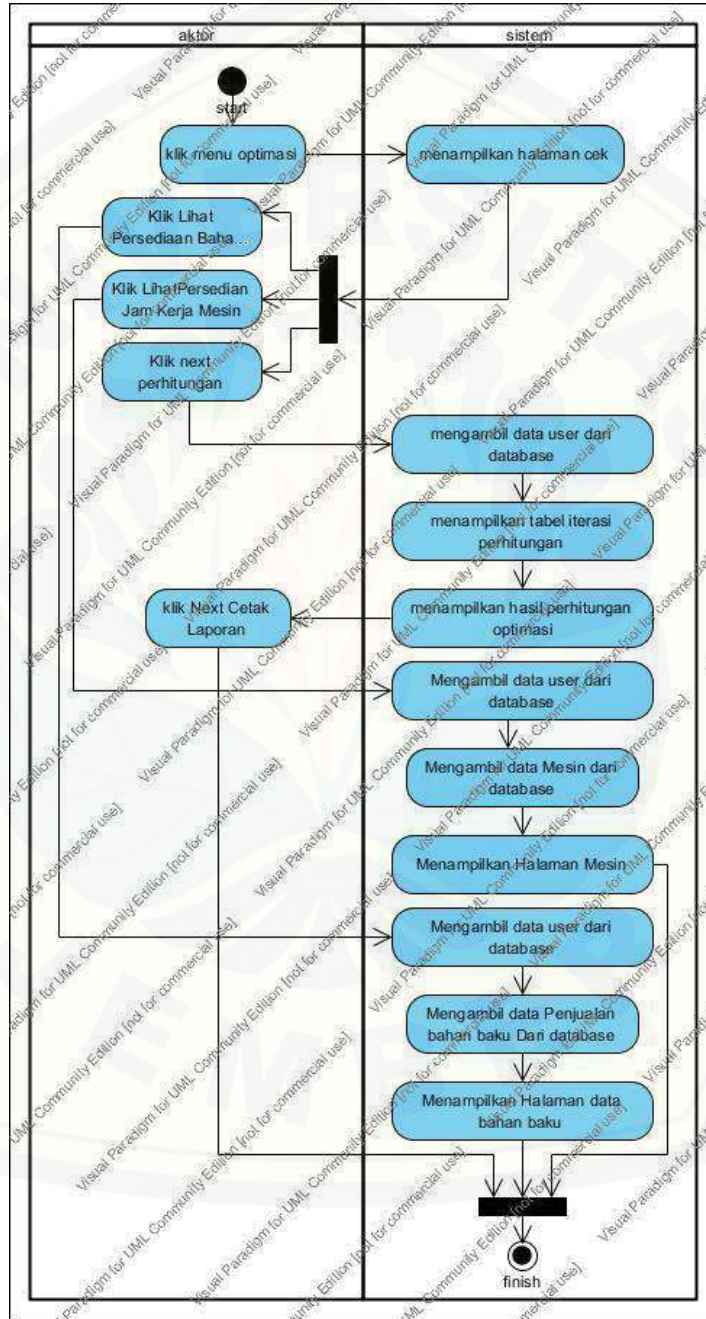
Penggambaran Activity diagram Mengelola Data Satuan yang ditunjukkan pada lampiran C.

4) Activity diagram Mengelola User

Penggambaran Activity diagram diagram Mengelola User yang ditunjukkan pada lampiran C.

5) Activity diagram Pengoptimalan dengan *Linear Programming*

Penggambaran Activity diagram diagram Pengoptimalan dengan *Linear Programming* yang ditunjukkan pada Gambar 4.4



Gambar . Activity diagram Pengoptimalan dengan *Linear Programming*

- 6) Activity diagram View Data Produk
Penggambaran Activity diagram diagram login ditunjukkan pada lampiran C.
- 7) Activity diagram Mengelola Bahan Baku
Penggambaran Activity diagram diagram Mengelola Bahan Baku yang ditunjukkan pada lampiran C.
- 8) Activity diagram View Bahan Baku
Penggambaran Activity diagram diagram View Bahan Baku yang ditunjukkan pada lampiran C.
- 9) Activity diagram Mengelola Data Mesin
Penggambaran Activity diagram diagram Mengelola Data Mesin yang ditunjukkan pada lampiran C.
- 10) Activity diagram View Data Mesin
Penggambaran Activity diagram diagram View Data Mesin yang ditunjukkan pada lampiran C.
- 11) Activity diagram View Penggunaan Jam Kerja Mesin
Penggambaran Activity diagram diagram View Penggunaan Jam Kerja Mesin yang ditunjukkan pada lampiran C.
- 12) Activity diagram Mengelola Penggunaan Jam Kerja Mesin
Penggambaran Activity diagram diagram Penggunaan Jam Kerja Mesin yang ditunjukkan pada lampiran C.
- 13) Activity diagram Mengelola Data Produksi
Penggambaran Activity diagram diagram Mengelola Data Produksi yang ditunjukkan pada lampiran C.
- 14) Activity diagram View Data Produksi
Penggambaran Activity diagram diagram View Data Produksi ditunjukkan pada lampiran C.
- 15) Activity diagram Mengelola Data Penggunaan Bahan Baku
Penggambaran Activity diagram diagram Mengelola Data Penggunaan Bahan Baku yang akan dibuat seperti yang ditunjukkan pada lampiran C.

16) Activity diagram View Data Penggunaan Bahan Baku

Penggambaran Activity diagram View Data Penggunaan Bahan Baku yang ditunjukkan pada lampiran C.

17) Activity diagram Cetak Laporan Perhitungan optimasi produksi

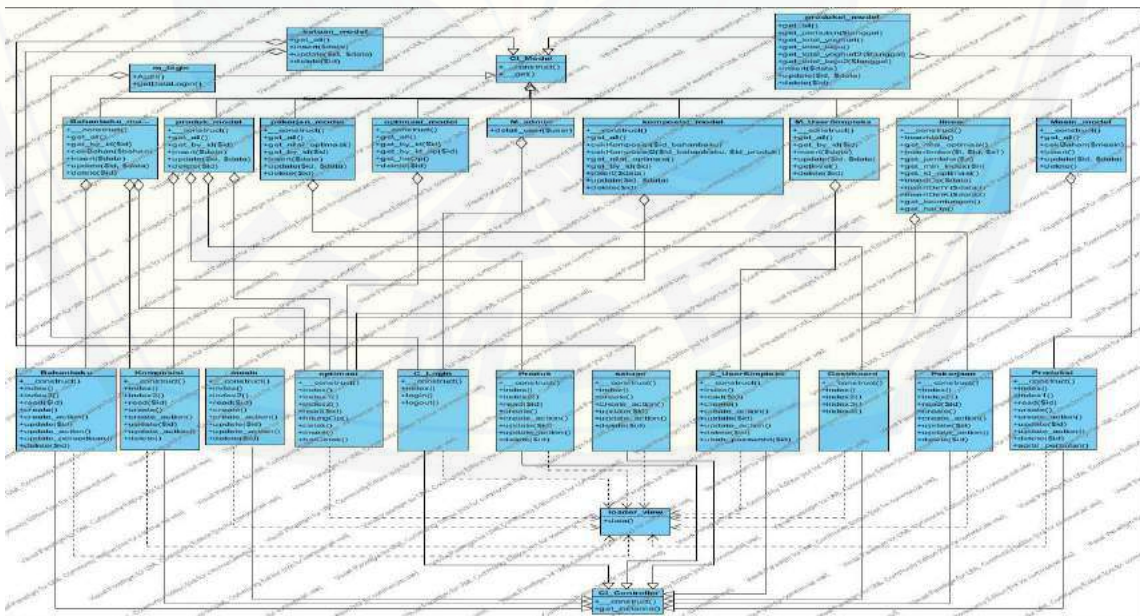
Penggambaran Activity diagram diagram Cetak Laporan Perhitungan optimasi produksi yang ditunjukkan pada lampiran C.

18) Activity diagram View Optimasi Produksi

Penggambaran Activity diagram diagram View Optimasi Produksi yang ditunjukkan pada lampiran C.

4.4.6 Class Diagram

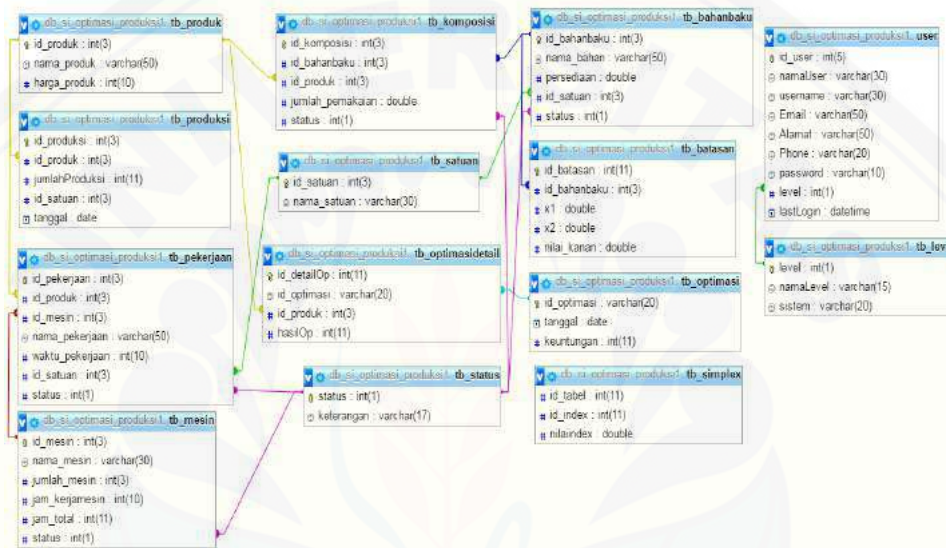
Class Diagram menggambarkan hubungan antarkelas yang digunakan untuk membangun suatu sistem. Class Diagram Sistem Pendukung Pengambilan Keputusan Optimasi Jumlah Produksi Untuk Meningkatkan Keuntungan Menggunakan Metode Simplex Linear Programming (Studi Kasus Rumah Yoghurt Malang) dapat dilihat pada Gambar 4.5



Gambar . Class Diagram

4.4.7 Entity Relationship Diagram (ERD)

ERD merupakan gambaran komponen dan struktur database yang digunakan dalam pembangunan sistem. ERD Sistem Pendukung Pengambilan Keputusan Optimasi Jumlah Produksi Untuk Meningkatkan Keuntungan Menggunakan Metode *Simplex Linear Programming* (Studi Kasus Rumah Yoghurt Malang) dapat dilihat pada Gambar 4.6



Gambar . Entity Relationship Diagram (ERD)

4.5 Implementasi

Setelah desain yang dibuat selesai langkah selanjutnya yaitu mengimplementasikan desain ke dalam kode program. Pada tahap ini penulisan kode program (coding) menggunakan bahasa pemrograman *Page Hyper text-Processor* (PHP) dengan bantuan *Code Igniter* (CI) dan manajemen database menggunakan *MySQL*.

Sistem Pendukung Pengambilan Keputusan Optimasi Jumlah Produksi Untuk Meningkatkan Keuntungan Menggunakan Metode *Simplex Linear Programming* ini dibangun menggunakan framework *Code Igniter* (CI). Baris kode program dari Sistem Pendukung Pengambilan Keputusan Optimasi Jumlah Produksi Untuk

Meningkatkan Keuntungan Menggunakan Metode *Simplex Linear Programming* terletak pada kelas *view* test, *controller* optimasi, serta kelas model linear dan *produk_model*. Penulisan kode program dapat dilihat pada tabel 4.9 sampai tabel 4.12

Tabel . Kode Program *function* *index()* pada *controller* optimasi

No.	Kode
26	<code>public function index(){</code>
27	
28	<code> \$user = \$this->session->userdata('username');</code>
29	<code> \$this->breadcrumbs->push('Bahanbaku', '/bahanbaku');</code>
30	<code> \$coba = \$this->madmin->detail_user(\$user)->result();</code>
31	
32	<code> \$data = array(</code>
33	<code> 'title' => 'Optimalisasi',</code>
34	<code> 'content' => 'formCek',</code>
35	<code> 'breadcrumbs' => \$this->breadcrumbs->show(),</code>
36	<code> 'user' => \$user ,</code>
37	<code> 'detail' => \$coba);</code>
38	
39	<code> \$this->load->view('layout/layoutProduksi', \$data);</code>
40	<code>}</code>

Tabel . Kode Program *function* *hitungOp()* pada *controller* optimasi

No	Kode
42.	<code>public function hitungOp(){</code>
43.	<code> \$this->linear->insertdata();</code>
44.	<code> \$produk = \$this->Produk_model->get_all();</code>
45.	<code> \$batasan = \$this->linear->get_nilai_optimasi();</code>
46.	<code> \$user = \$this->session->userdata('username');</code>
47.	<code> \$coba = \$this->madmin->detail_user(\$user)->result();</code>
48.	<code> \$this->breadcrumbs->push('Optimasi', '/optimasi/test');</code>
49.	
50.	<code> foreach (\$produk as \$pd) {</code>
51.	<code> \$nama_produk[\$pd->id_produk] = \$pd->nama_produk;</code>
52.	<code> \$harga_produk[\$pd->id_produk] = \$pd->harga_produk;</code>
53.	<code> \$id_produk[\$pd->id_produk] = \$pd->id_produk;</code>
54.	<code> }</code>

```
55.
56. //===== tabel fungsi batasan dan tabel dasar=====
57.     $optimasi = array();
58.     $tabel = array();
59.     $hasilz = array();
60.     $tabel1 = array();
61.     $tabel2 = array();
62.     $i = 3;
63.     foreach ($batasan as $km ) {
64.         $temp = array($km->nilai_kanan,$km->x1,$km->x2);
65.         for ($j=0; $j < count($batasan); $j++) {
66.             $angka = 0;
67.             array_push($temp, $angka);
68.         }
69.         $temp[$i] = 1;
70.         array_push($tabel1, $temp);
71.         $i++;
72.     }
73.     array_push($tabel, $tabel1);
74.     $max = count($batasan);
75.     $z = array(0,-$harga_produk[1],-$harga_produk[2]);
76.     for ($j=0; $j < count($batasan); $j++) {
77.         $angka = 0;
78.         array_push($z, $angka);
79.     }
80.     array_push($hasilz, $z);
81.
82. //===== isi matrix simplex ke i =====
83.     $ulang = false;
84.     $count = 1;
85.     for ($n=0; $n < $count; $n++) {
86.         $minz = min($hasilz[$n]);
87.         if ($minz < 0) {
88.             $count++;
89.             array_push($tabel, $tabel1);
90.             $index = array();
91.             $nilai2 = array();
92.             $bariskunci = 0;
93.             $id = 0;
94.             $kunci = array_search($minz, $hasilz[$n]);
95.             foreach ($tabel[$n] as $optima) {
96.                 $kolomkunci[$n][ ] = $optima[$kunci];
```

```

97.     $skanan[$n][ ] = $optima[0];
98.     }
99.     for ($i=0; $i < $max; $i++) {
100.        if ($kolomkunci[$n][$i] != 0) {
101.            $index[$i] = $skanan[$n][$i] / $kolomkunci[$n][$i];
102.        }else if ($kolomkunci[$n][$i] == 0) {
103.            $index[$i] = "null";
104.        }
105.        $this->linear->insertindex($n,$id,$index[$i]);
106.        $id++;
107.    }
108.    $qmin = $this->linear->get_min_index($n);
109.    foreach ($qmin->result_array() as $key) {
110.        $minin = $key['minindex'];
111.    }
112.    $minindex = array_search($minin, $index);
113.    $nilaibarisminindex = array();
114.    for ($i=0; $i < $max + 3; $i++) {
115.        for ($j=0; $j < $max; $j++) {
116.            $nilai2=$tabel[$n][$minindex][$i] / $kolomkunci[$n][$minindex];
117.            $tabel[$n+1][$minindex][$i] = $nilai2;
118.        }
119.        array_push($nilaibarisminindex, $nilai2);
120.    }
121.
122.    //===== tabel bayangan =====
123.    $tabelbayangan = array();
124.    for ($i=0; $i < $max; $i++) {
125.        for ($j=0; $j < $max + 3; $j++) {
126.            $nilaibayangan=$tabel[$n+1][$minindex][$j] * $kolomkunci[$n][$i];
127.            $tabelbayangan[$n][$i][$j] = $nilaibayangan;
128.        }
129.    }
130.    $zbayangan = array();
131.    for ($i=0; $i < $max + 3; $i++) {
132.        $nilaiz = $tabel[$n+1][$minindex][$i] * $hasilz[$n][$skunci];
133.        if ($nilaiz == -0) {
134.            $nilaiz = 0;
135.        }
136.        $zbayangan[$n][$i] = $nilaiz;
137.    }
138.    //===== mengisi tabel berikutnya =====

```

```
139.
140.         for ($i=0; $i < $max; $i++) {
141.             if ($i == $minindex) {
142.                 $tabel[$n+1][$minindex] = $nilaibarisminindex;
143.             }else{
144.                 for ($j=0; $j < $max + 3; $j++) {
145.                     $nilai2 = $tabel[$n][$i][$j] - $tabelbayangan[$n][$i][$j];
146.                     $tabel[$n+1][$i][$j] = $nilai2;
147.                 }
148.             }
149.         }
150.         $zi = array();
151.         for ($i=0; $i < $max + 3; $i++) {
152.             $nilaiz = $hasilz[$n][$i] - $zbayangan[$n][$i];
153.             if ($nilaiz == -0) {
154.                 $nilaiz = 0;
155.             }
156.             array_push($zi, $nilaiz);
157.         }
158.         array_push($hasilz, $zi);
159.     }
160. }
161.
162. //===== mencari hasil akhir =====
163.
164. $tabelmax = count($tabel) - 1;
165. foreach ($tabel[$tabelmax] as $optima) {
166.     $kolomx1[] = $optima[1];
167.     $kolomx2[] = $optima[2];
168. }
169. $x1 = array_search(1, $kolomx1);
170. $x2 = array_search(1, $kolomx2);
171.
172. //===== load data =====
173. $data = array(
174.     'title' => 'Optimasi' ,
175.     'content' => 'test',
176.     'breadcrumbs' => $this->breadcrumbs->show(),
177.     'user' => $user ,
178.     'produk' => $produk,
179.     'detail' => $coba,
180.     'nama_produk' => $nama_produk,
```

181.	'harga_produk' => \$harga_produk,
182.	'optimasi' => \$optimasi,
183.	'tabel' => \$tabel,
184.	'z' => \$hasilz,
185.	'x1' => \$x1,
186.	'x2' => \$x2,
187.	'max' => \$max
188.);
189.	
190.	\$this->load->view('layout/layoutProduksi', \$data);
191.	}
192.	

Tabel . Kode Program kelas model optimasi_model

No	Kode
1.	public function insertdata() {
2.	\$this->db->query("truncate tb_batasan");
3.	\$this->db->query("truncate tb_simplex");
4.	
5.	//===== mencari penggunaan bahan baku =====
6.	\$query1=\$this->db->query(
7.	"SELECT * FROM tb_bahanbaku");
8.	foreach (\$query1->result_array() as \$key) {
9.	\$хid_bahanbaku[] = \$key['id_bahanbaku'];
10.	\$хpersediaan[] = \$key['persediaan'];
11.	}
12.	\$хcount1 = count(\$хid_bahanbaku);
13.	\$query2 = \$this->db->query("SELECT * FROM tb_produk");
14.	foreach (\$query2->result_array() as \$key) {
15.	\$хid_produk[] = \$key['id_produk'];
16.	}
17.	for (\$хi=0; \$хi < \$хcount1 ; \$хi++) {
18.	\$query3=\$this->db->query(
19.	"SELECT sum(jumlah_pemakaian) as jumlah
20.	FROM tb_komposisi
21.	WHERE id_bahanbaku = \$хid_bahanbaku[\$хi] and
22.	id_produk = \$хid_produk[0]");
23.	
24.	foreach (\$query3->result_array() as \$key) {

```
25.         $x1 = $key['jumlah'];
26.     }
27.     $query4 = $this->db->query(
28. "SELECT sum(jumlah_pemakaian) as jumlah
29. FROM tb_komposisi
30. WHERE id_bahanbaku = $id_bahanbaku[$i] and
31. id_produk = $id_produk[1]");
32.     foreach ($query4->result_array() as $key) {
33.         $x2 = $key['jumlah'];
34.     }
35.     $this->db->query( "insert into tb_batasan values (
36. ", $id_bahanbaku[$i],
37. $x1,
38. $x2,
39. $persediaan[$i]);
40.     }
41.
42. //===== mencari penggunaan mesin =====
43.
44.     $query5 = $this->db->query("SELECT * FROM tb_mesin");
45.     foreach ($query5->result_array() as $key) {
46.         $id_mesin[] = $key['id_mesin'];
47.         $jamtotal[] = $key['jam_total'];
48.     }
49.     $count3 = count($id_mesin);
50.     $x1 = 0; $x2 = 0;
51.     for ($i=0; $i < $count3 ; $i++) {
52.         $query6=$this->db->query(
53. "SELECT sum(waktu_pekerjaan) as jumlah
54. FROM tb_pekerjaan
55. WHERE id_mesin = $id_mesin[$i]
56. and id_produk = $id_produk[0]");
57.         foreach ($query6->result_array() as $key) {
58.             $x1 = $key['jumlah'];
59.             if ($key['jumlah'] == null) {
60.                 $x1 = 0;
61.             }
62.         }
63.         $query7 = $this->db->query(
64. "SELECT sum(waktu_pekerjaan) as jumlah
65. FROM tb_pekerjaan WHERE id_mesin = $id_mesin[$i] and
66. id_produk = $id_produk[1]");
```



```

67.         foreach ($query7->result_array() as $key) {
68.             $x2 = $key['jumlah'];
69.             if ($key['jumlah'] == null) {
70.                 $x2 = 0;
71.             }
72.         }
73.         $this->db->query("insert into tb_batasan values
74.         (', $id_mesin[$i], $x1, $x2, $jamtotal[$i]");
75.     }
76. }
77.
78.
79.     function get_nilai_optimasi()
80.     {
81.         return $this->db->get('tb_batasan')->result();
82.     }
83.
84.     function get_jumlahx($x)
85.     {
86.         return $this->db->query("SELECT sum($x) as jumlah from
87.         tb_batasan")->result_array();
88.     }
89.     public function insertindex($i,$id, $x1) {
90.         $this->db->query("insert into tb_simplex values ($i, $id, $x1)");
91.     }
92.
93.     function get_min_index($n)
94.     {
95.         return $this->db->query("SELECT min(nilaiindex) as minindex
96.         from tb_simplex where id_tabel = '$n' and nilaiindex <> 'null' and
97.         nilaiindex >=0");
98.     }
99. }
100.

```

Tabel . Kode program kelas model produk_model

No.	Kode
19	function get_all(){
20	return \$this->db->get(\$this->table)->result();
21	}

4.6 Pengujian Sistem

Pengujian dilakukan untuk mengevaluasi aplikasi yang telah dibuat. Proses pengujian dilakukan dengan pengujian *whitebox* terlebih dahulu kemudian akan dilanjutkan dengan pengujian *blackbox*.

4.6.1 Pengujian white box

Pengujian *white box* pada “Sistem Pendukung Pengambilan Keputusan Optimasi Jumlah Produksi Untuk Meningkatkan Keuntungan Menggunakan Metode *Simplex Linear Programming*” (Studi Kasus Rumah Yoghurt Malang) ini dengan cara membuat listing program, membuat diagram alir dari listing program, menghitung *cyclomatic complexity*, menentukan jalur independen, dan *test case*. Hal ini bertujuan untuk mengevaluasi apakah sistem mampu menghasilkan fungsi-fungsi, inputan dan keluaran yang sesuai dengan spesifikasi dari kebutuhan sistem. Pengujian White Box Testing optimasi produksi adalah sebagai berikut:

a. Listing Program

Listing program merupakan baris-baris kode yang nantinya akan diuji.

b. Diagram Alir

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

c. Kompleksitas siklomatik (*cyclomatic complexity*)

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

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

Keterangan :

V(G) : Kompleksitas siklomatik

E : Jumlah *edge*

N : Jumlah *node*

d. Pengujian Basis Set (*Test Case*)

Pada bagian ini diberikan contoh data yang akan memaksa pelaksanaan jalur di basis set. Data yang dieksekusi dimasukkan ke dalam grafik alir apakah sudah melewati basis set yang tersedia. Sistem telah memenuhi syarat kelayakan perangkat lunak jika salah satu jalur yang dieksekusi setidaknya satu kali.

Pengujian pada Sistem Pendukung Pengambilan Keputusan Optimasi Jumlah Produksi Untuk Meningkatkan Keuntungan Menggunakan Metode *Simplex Linear Programming* menggunakan metode *white box* akan diterapkan pada fitur yang dinilai dapat mewakili sistem sebagai berikut.

1) *White box testing* pada *method* `hitungOp`

Pengujian *white box* menggunakan tahapan listing program, diagram alir, dan pengujian test case.

```

37     public function hitungOp(){
38         $this->linear->insertdata();
39         $produk = $this->Produk_model->get_all();
40         $batasan = $this->linear->get_nilai_optimasi();
41         $user = $this->session->userdata('username');
42         $coba = $this->madmin->detail_user($user)->result();
43         $this->breadcrumbs->push('Optimasi', '/optimasi/test');
44
45         foreach ($produk as $pd ) {
46             $nama_produk[$pd->id_produk] = $pd->nama_produk;
47             $harga_produk[$pd->id_produk] = $pd->harga_produk;
48             $id_produk[$pd->id_produk] = $pd->id_produk;
49         }
50         //-----|tabel fungsi batasan dan tabel simplex 1-----
51         $optimasi = array();
52         $tabel = array();
53         $hasilz = array();
54         $tabel1 = array();
55         $tabel2 = array();
56         $i = 3;
57         foreach ($batasan as $km ) {
58             $temp = array($km->nilai_kanan,$km->x1/1000,$km->x2/1000);
59             for ($j=0; $j < count($batasan); $j++) {
60                 $angka = 0;
61                 array_push($temp, $angka);
62             }
63             $temp[$i] = 1;
64             array_push($tabel1, $temp);
65             $i++;
66         }
67
68         array_push($tabel, $tabel1);
69         $max = count($batasan);
70         $z = array(0, -$harga_produk[1], -$harga_produk[2]);
71         for ($j=0; $j < count($batasan); $j++) {
72             $angka = 0;
73             array_push($z, $angka);
74         }
75
76         array_push($hasilz, $z);

```

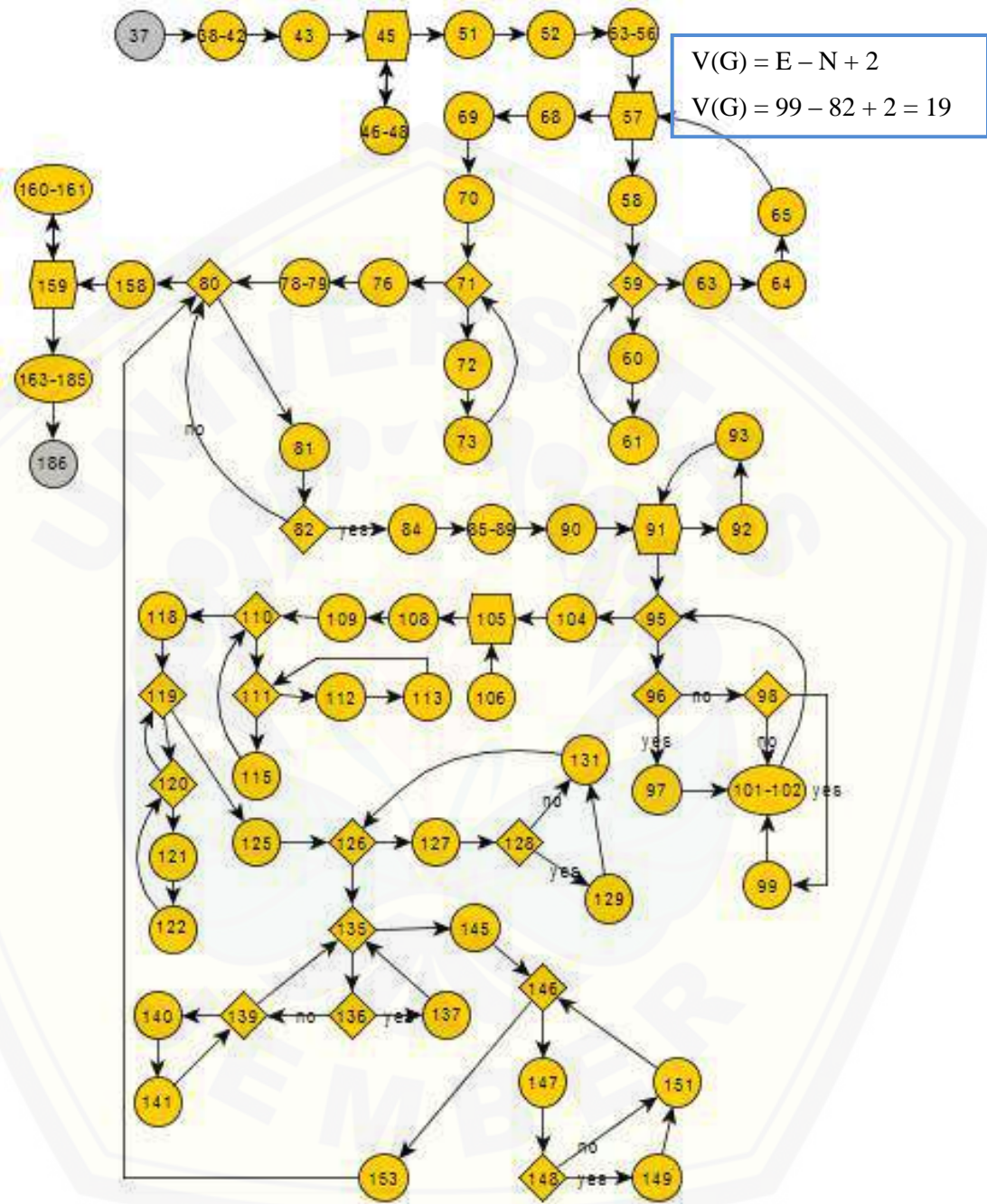
```

77 //----- isi matrix simplex ke i -----
78 $ulang = false;
79 $count = 1;
80 for ($n=0; $n < $count; $n++) {
81     $minz = min($hasilz[$n]);
82     if ($minz < 0) {
83
84         $count++;
85         array_push($tabel, $tabel1);
86         $index = array();
87         $nilai2 = array();
88         $bariskunci = 0;
89         $id = 0;
90         $kunci = array_search($minz, $hasilz[$n]);
91         foreach ($tabel[$n] as $optima) {
92             $kolomkunci[$n][] = $optima[$kunci];
93             $kanan[$n][] = $optima[0];
94         }
95         for ($i=0; $i < $max; $i++) {
96             if ($kolomkunci[$n][$i] != 0) {
97                 $index[$i] = $kanan[$n][$i] / $kolomkunci[$n][$i];
98             }else if ($kolomkunci[$n][$i] == 0) {
99                 $index[$i] = "null";
100             }
101             $this->linear->insertindex($n,$id,$index[$i]);
102             $id++;
103         }
104         $qmin = $this->linear->get_min_index($n);
105         foreach ($qmin->result_array() as $key) {
106             $minin = $key['minindex'];
107         }
108         $minindex = array_search($minin, $index);
109         $nilaibarisminindex = array();
110         for ($i=0; $i < $max + 3; $i++) {
111             for ($j=0; $j < $max; $j++) {
112                 $nilai2 = $tabel[$n][$minindex][$i] / $kolomkunci[$n][$minindex];
113                 $tabel[$n+1][$minindex][$i] = $nilai2;
114             }
115             array_push($nilaibarisminindex, $nilai2);
116         }
117 //----- tabel bayangan-----
118 $tabelbayangan = array();
119 for ($i=0; $i < $max; $i++) {
120     for ($j=0; $j < $max + 3; $j++) {
121         $nilaibayangan = $tabel[$n+1][$minindex][$j] * $kolomkunci[$n][$i];
122         $tabelbayangan[$n][$i][$j] = $nilaibayangan;
123     }
124 }
125 $zbayangan = array();
126 for ($i=0; $i < $max + 3; $i++) {
127     $nilaiz = $tabel[$n+1][$minindex][$i] * $hasilz[$n][$kunci];
128     if ($nilaiz == -0) {
129         $nilaiz = 0;
130     }
131     $zbayangan[$n][$i] = $nilaiz;
132 }
133 //----- mengisi tabel berikutnya-----
134 for ($i=0; $i < $max; $i++) {
135     if ($i == $minindex) {
136         $tabel[$n+1][$minindex] = $nilaibarisminindex;
137     }else{
138         for ($j=0; $j < $max + 3; $j++) {
139             $nilai2 = $tabel[$n][$i][$j] - $tabelbayangan[$n][$i][$j];
140             $tabel[$n+1][$i][$j] = $nilai2;
141         }
142     }
143 }
144 $zi = array();
145 for ($i=0; $i < $max + 3; $i++) {
146     $nilaiz = $hasilz[$n][$i] - $zbayangan[$n][$i];
147     if ($nilaiz == -0) {
148         $nilaiz = 0;
149     }
150     array_push($zi, $nilaiz);
151 }
152 array_push($hasilz, $zi);
153 }
154 }
155 }

```

```
156
157 //-----mencari hasil akhir-----
158     $tabelmax = count($tabel) - 1;
159     foreach ($tabel[$tabelmax] as $optima) {
160         $kolomx1[] = $optima[1];
161         $kolomx2[] = $optima[2];
162     }
163     $x1 = array_search(1, $kolomx1);
164     $x2 = array_search(1, $kolomx2);
165
166 //-----Load data-----
167     $data = array(
168         'title'     => 'Optimasi' ,
169         'content'   => 'test',
170         'breadcrumbs' => $this->breadcrumbs->show(),
171         'user'      => $user ,
172         'produk'    => $produk,
173         'detail'    => $coba,
174         'nama_produk' => $nama_produk,
175         'harga_produk' => $harga_produk,
176         'optimasi' => $optimasi,
177         'tabel' => $tabel,
178         'z' => $hasilz,
179         'x1' => $x1,
180         'x2' => $x2,
181         'max' => $max
182     );
183
184     $this->load->view('layout/layoutProduksi', $data);
185 }
186 }
187 }
```

Gambar . Listing Program Method hitungOp



Gambar . Diagram alir *method* hitungOp

Tabel . Test case function hitungOp()

<i>Test case function hitungOp()</i>	
Jalur 1	
Test Case	Jika ada data produk pada tb_produk
Target yang Diharapkan	Mengambil semua data produk didatabase
Hasil Pengujian	Benar
Path / Jalur	37-38-39-40-41-42-43-45-56-57-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62-63-64-65-66-67-68-69-70-71-72-73-74-75-76-77-78-79-80-81-82-83-84-85-86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-101-102-103-104-105-106-107-108-109-110-111-112-113-114-115-116-117-118-119-120-121-122-123-124-125-126-127-128-129-130-131-132-133-134-135-136-137-138-139-140-141-142-143-144-145-146-147-148-149-150-151-152-153-154-155-156-157-158-159-160-161-162-163-164-165-166-167-167-168-169-170-171-172-173-174-175-176-178-179-180-181-182-183-184-185-186
Jalur 2	
Test Case	Jika ada data pada tb_batasan
Target yang Diharapkan	Mengambil semua nilai batasan pada tb_batasan dan menampungnya pada array \$temp
Hasil Pengujian	benar
Path / Jalur	37-38-39-40-41-42-43-45-56-57-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62-63-64-65-66-67-68-69-70-71-72-73-74-75-76-77-78-79-80-81-82-83-84-85-86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-101-102-103-104-105-106-107-108-109-110-111-112-113-114-115-116-117-118-119-120-121-122-123-124-125-126-127-128-129-130-131-132-133-134-135-136-137-138-139-140-141-142-143-144-145-146-147-148-149-150-151-152-153-154-155-156-157-158-159-160-161-162-163-164-165-166-167-167-168-169-170-171-172-173-174-175-176-178-179-180-181-182-183-184-185-186
Jalur 3	
Test Case	Jika ada data pada array \$temp
Target yang Diharapkan	Memberikan nilai variabel slack/tambahan pada array \$temp untu membentuk tabel dasar simplex
Hasil Pengujian	Benar

Path / Jalur	37-38-39-40-41-42-43-45-56-57-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62-63-64-65-66-67-68-69-70-71-72-73-74-75-76-77-78-79-80-81-82-83-84-85-86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-101-102-103-104-105-106-107-108-109-110-111-112-113-114-115-116-117-118-119-120-121-122-123-124-125-126-127-128-129-130-131-132-133-134-135-136-137-138-139-140-141-142-143-144-145-146-147-148-149-150-151-152-153-154-155-156-157-158-159-160-161-162-163-164-165-166-167-167-168-169-170-171-172-173-174-175-176-178-179-180-181-182-183-184-185-186
Jalur 4	
Test Case	Jika ada data harga jual produk
Target yang Diharapkan	Mengambil data harga jual produk untuk membentuk array Z dan memasukkan ke dalam tabel simplex dengan menggunakan fungsi <i>array_push()</i>
Hasil Pengujian	Benar
Path / Jalur	37-38-39-40-41-42-43-45-56-57-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62-63-64-65-66-67-68-69-70-71-72-73-74-75-76-77-78-79-80-81-82-83-84-85-86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-101-102-103-104-105-106-107-108-109-110-111-112-113-114-115-116-117-118-119-120-121-122-123-124-125-126-127-128-129-130-131-132-133-134-135-136-137-138-139-140-141-142-143-144-145-146-147-148-149-150-151-152-153-154-155-156-157-158-159-160-161-162-163-164-165-166-167-167-168-169-170-171-172-173-174-175-176-178-179-180-181-182-183-184-185-186
Jalur 5	
Test Case	Jika ada nilai negatif pada baris Z
Target yang Diharapkan	Mengulang perhitungan sebanyak beberapa kali putaran sampai tidak ada nilai negatif pada baris Z pada tabel simplex yang telah terbentuk
Hasil Pengujian	Benar
Path / Jalur	37-38-39-40-41-42-43-45-56-57-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62-63-64-65-66-67-68-69-70-71-72-73-74-75-76-77-78-79-80-81-82-83-84-85-86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-101-102-103-104-105-106-107-108-109-110-111-112-113-114-115-116-117-118-119-120-121-122-123-124-125-

126-127-128-129-130-131-132-133-134-135-136-137-138-139-140-141-142-143-144-145-146-147-148-149-150-151-152-153-154-155-156-157-158-159-160-161-162-163-164-165-166-167-167-168-169-170-171-172-173-174-175-176-178-179-180-181-182-183-184-185-186

Jalur 6

Test Case	Jika ada nilai negatif pada baris Z
Target yang Diharapkan	Mengecek apakah masih ada nilai negatif pada baris Z pada tabel simplex
Hasil Pengujian	Benar
Path / Jalur	37-38-39-40-41-42-43-45-56-57-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62-63-64-65-66-67-68-69-70-71-72-73-74-75-76-77-78-79-80-81-82-83-84-85-86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-101-102-103-104-105-106-107-108-109-110-111-112-113-114-115-116-117-118-119-120-121-122-123-124-125-126-127-128-129-130-131-132-133-134-135-136-137-138-139-140-141-142-143-144-145-146-147-148-149-150-151-152-153-154-155-156-157-158-159-160-161-162-163-164-165-166-167-167-168-169-170-171-172-173-174-175-176-178-179-180-181-182-183-184-185-186

Jalur 7

Test Case	Jika ada nilai negatif yang ditemukan pada baris Z pada tabel simplex
Target yang Diharapkan	Mencari nilai negatif terbesar dari z untuk menentukan kolom kunci
Hasil Pengujian	Benar
Path / Jalur	37-38-39-40-41-42-43-45-56-57-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62-63-64-65-66-67-68-69-70-71-72-73-74-75-76-77-78-79-80-81-82-83-84-85-86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-101-102-103-104-105-106-107-108-109-110-111-112-113-114-115-116-117-118-119-120-121-122-123-124-125-126-127-128-129-130-131-132-133-134-135-136-137-138-139-140-141-142-143-144-145-146-147-148-149-150-151-152-153-154-155-156-157-158-159-160-161-162-163-164-165-166-167-167-168-169-170-171-172-173-174-175-176-178-179-180-181-182-183-184-185-186

Jalur 8	
Test Case	Jika nilai negatif terbesar pada baris Z pada tabel simplex ada
Target yang Diharapkan	Menentukan kolom kunci dan nilai kanannya yang digunakan untuk menghitung nilai index
Hasil Pengujian	Benar
Path / Jalur	37-38-39-40-41-42-43-45-56-57-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62-63-64-65-66-67-68-69-70-71-72-73-74-75-76-77-78-79-80-81-82-83-84-85-86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-101-102-103-104-105-106-107-108-109-110-111-112-113-114-115-116-117-118-119-120-121-122-123-124-125-126-127-128-129-130-131-132-133-134-135-136-137-138-139-140-141-142-143-144-145-146-147-148-149-150-151-152-153-154-155-156-157-158-159-160-161-162-163-164-165-166-167-167-168-169-170-171-172-173-174-175-176-178-179-180-181-182-183-184-185-186
Jalur 9	
Test Case	Jika kolom kunci dan nilai kanan telah ditentukan
Target yang Diharapkan	Menghitung nilai index dengan cara membagi nilai kanan dengan setiap koefisien pada kolom kunci
Hasil Pengujian	Benar
Path / Jalur	37-38-39-40-41-42-43-45-56-57-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62-63-64-65-66-67-68-69-70-71-72-73-74-75-76-77-78-79-80-81-82-83-84-85-86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-101-102-103-104-105-106-107-108-109-110-111-112-113-114-115-116-117-118-119-120-121-122-123-124-125-126-127-128-129-130-131-132-133-134-135-136-137-138-139-140-141-142-143-144-145-146-147-148-149-150-151-152-153-154-155-156-157-158-159-160-161-162-163-164-165-166-167-167-168-169-170-171-172-173-174-175-176-178-179-180-181-182-183-184-185-186
Jalur 10	
Test Case	Jika semua nilai index sudah terbentuk
Target yang Diharapkan	Mencari nilai index positif terkecil berada dibaris keberapa untuk menentukan baris kunci
Hasil Pengujian	Benar
Path / Jalur	37-38-39-40-41-42-43-45-56-57-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62-63-64-65-66-67-68-69-

70-71-72-73-74-75-76-77-78-79-80-81-82-83-84-85-
86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-101-
102-103-104-105-106-107-108-109-110-111-112-113-
114-115-116-117-118-119-120-121-122-123-124-125-
126-127-128-129-130-131-132-133-134-135-136-137-
138-139-140-141-142-143-144-145-146-147-148-149-
150-151-152-153-154-155-156-157-158-159-160-161-
162-163-164-165-166-167-167-168-169-170-171-172-
173-174-175-176-178-179-180-181-182-183-184-185-
186

Jalur 11

Test Case	Jika sudah menemukan baris kunci
Target yang Diharapkan	Mencari letak nilai kunci dengan melihat perpotongan nilai antara baris dan kolom kunci
Hasil Pengujian	Benar
Path / Jalur	37-38-39-40-41-42-43-45-56-57-48-49-50-51-52-53- 54-55-56-57-58-59-60-61-62-63-64-65-66-67-68-69- 70-71-72-73-74-75-76-77-78-79-80-81-82-83-84-85- 86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-101- 102-103-104-105-106-107-108-109-110-111-112-113- 114-115-116-117-118-119-120-121-122-123-124-125- 126-127-128-129-130-131-132-133-134-135-136-137- 138-139-140-141-142-143-144-145-146-147-148-149- 150-151-152-153-154-155-156-157-158-159-160-161- 162-163-164-165-166-167-167-168-169-170-171-172- 173-174-175-176-178-179-180-181-182-183-184-185- 186

Jalur 12

Test Case	Jika nilai kunci telah ditentukan
Target yang Diharapkan	Mengubah baris kunci dengan cara membagi setiap koefien pada baris kunci dengan nilai kunci untuk membentuk baris kunci baru
Hasil Pengujian	Benar
Path / Jalur	37-38-39-40-41-42-43-45-56-57-48-49-50-51-52-53- 54-55-56-57-58-59-60-61-62-63-64-65-66-67-68-69- 70-71-72-73-74-75-76-77-78-79-80-81-82-83-84-85- 86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-101- 102-103-104-105-106-107-108-109-110-111-112-113- 114-115-116-117-118-119-120-121-122-123-124-125- 126-127-128-129-130-131-132-133-134-135-136-137- 138-139-140-141-142-143-144-145-146-147-148-149-

	150-151-152-153-154-155-156-157-158-159-160-161-162-163-164-165-166-167-167-168-169-170-171-172-173-174-175-176-178-179-180-181-182-183-184-185-186
Jalur 13	
Test Case	Jika baris kunci baru telah terbentuk
Target yang Diharapkan	Memasukkan hasil perhitungan baris kunci baru ke dalam tabel simplex yang baru
Hasil Pengujian	Benar
Path / Jalur	37-38-39-40-41-42-43-45-56-57-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62-63-64-65-66-67-68-69-70-71-72-73-74-75-76-77-78-79-80-81-82-83-84-85-86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-101-102-103-104-105-106-107-108-109-110-111-112-113-114-115-116-117-118-119-120-121-122-123-124-125-126-127-128-129-130-131-132-133-134-135-136-137-138-139-140-141-142-143-144-145-146-147-148-149-150-151-152-153-154-155-156-157-158-159-160-161-162-163-164-165-166-167-167-168-169-170-171-172-173-174-175-176-178-179-180-181-182-183-184-185-186
Jalur 14	
Test Case	Jika baris baru kunci telah terbentuk
Target yang Diharapkan	Mengubah nilai baris selain baris kunci untuk membentuk baris baru dengan cara mengalikan setiap koefisien pada baris dengan koefisien pada kolom kunci
Hasil Pengujian	Benar
Path / Jalur	37-38-39-40-41-42-43-45-56-57-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62-63-64-65-66-67-68-69-70-71-72-73-74-75-76-77-78-79-80-81-82-83-84-85-86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-101-102-103-104-105-106-107-108-109-110-111-112-113-114-115-116-117-118-119-120-121-122-123-124-125-126-127-128-129-130-131-132-133-134-135-136-137-138-139-140-141-142-143-144-145-146-147-148-149-150-151-152-153-154-155-156-157-158-159-160-161-162-163-164-165-166-167-167-168-169-170-171-172-173-174-175-176-178-179-180-181-182-183-184-185-186
Jalur 15	
Test Case	Jika baris baru kunci telah terbentuk
Target yang Diharapkan	Mengubah nilai baris z untuk membentuk baris baru

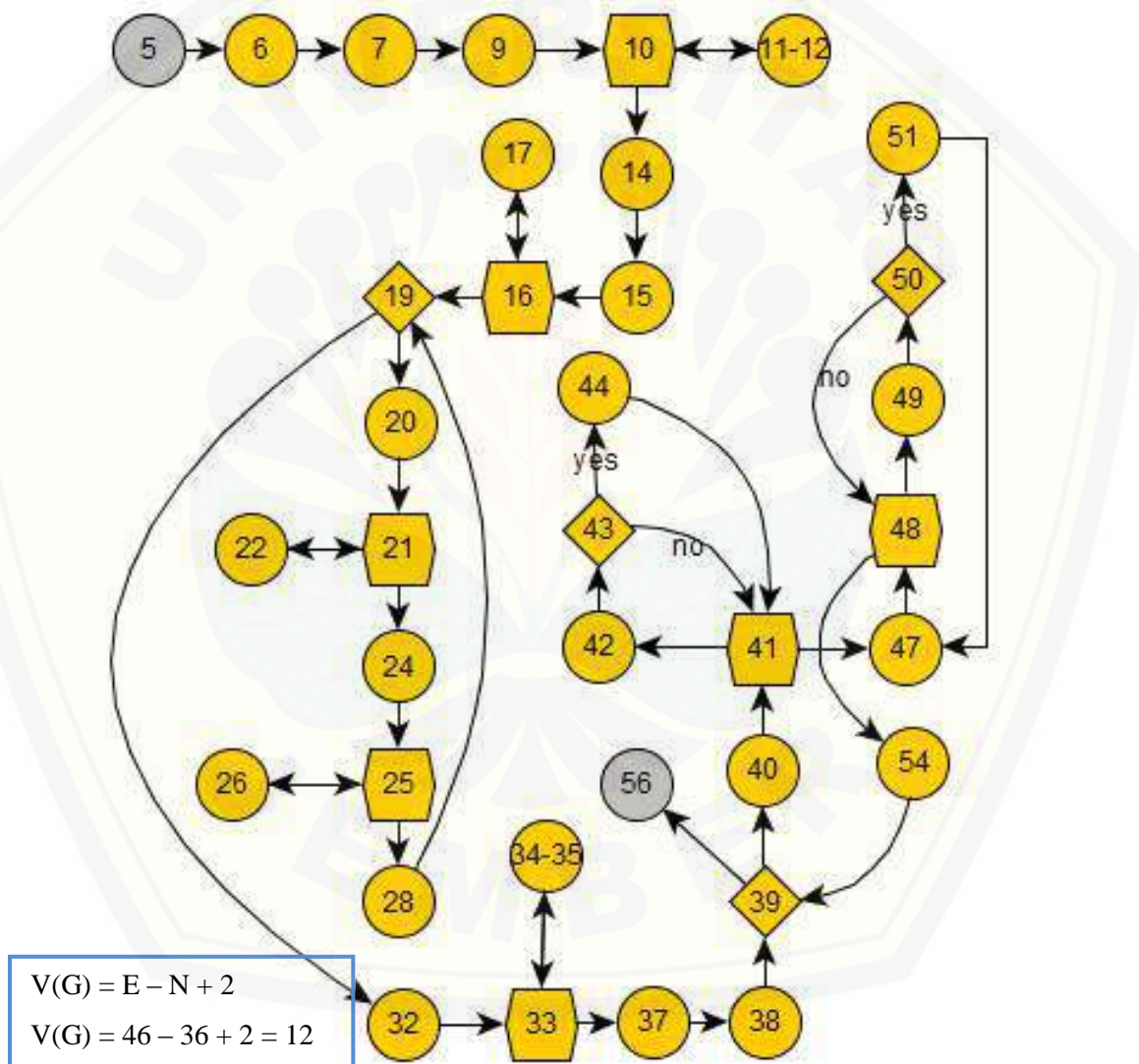
Diharapkan	dengan cara mengalikan setiap koefisien pada baris z dengan koefisien pada kolom kunci
Hasil Pengujian	Benar
Path / Jalur	37-38-39-40-41-42-43-45-56-57-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62-63-64-65-66-67-68-69-70-71-72-73-74-75-76-77-78-79-80-81-82-83-84-85-86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-101-102-103-104-105-106-107-108-109-110-111-112-113-114-115-116-117-118-119-120-121-122-123-124-125-126-127-128-129-130-131-132-133-134-135-136-137-138-139-140-141-142-143-144-145-146-147-148-149-150-151-152-153-154-155-156-157-158-159-160-161-162-163-164-165-166-167-167-168-169-170-171-172-173-174-175-176-178-179-180-181-182-183-184-185-186
Jalur 16	
Test Case	Jika baris yang akan diisi sama dengan baris kunci yang telah diisi sebelumnya maka tidak diubah
Target yang Diharapkan	Tidak mengisi baris baru kunci yang ada pada tabel simplex baru
Hasil Pengujian	Benar
Path / Jalur	37-38-39-40-41-42-43-45-56-57-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62-63-64-65-66-67-68-69-70-71-72-73-74-75-76-77-78-79-80-81-82-83-84-85-86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-101-102-103-104-105-106-107-108-109-110-111-112-113-114-115-116-117-118-119-120-121-122-123-124-125-126-127-128-129-130-131-132-133-134-135-136-137-138-139-140-141-142-143-144-145-146-147-148-149-150-151-152-153-154-155-156-157-158-159-160-161-162-163-164-165-166-167-167-168-169-170-171-172-173-174-175-176-178-179-180-181-182-183-184-185-186
Jalur 17	
Test Case	Jika baris yang akan diisi sama dengan baris kunci yang telah diisi sebelumnya maka tidak diubah
Target yang Diharapkan	Menghitung dan mengisi baris selain baris kunci baru pada tabel simplex baru
Hasil Pengujian	Benar
Path / Jalur	37-38-39-40-41-42-43-45-56-57-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62-63-64-65-66-67-68-69-70-71-72-73-74-75-76-77-78-79-80-81-82-83-84-85-

	86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-101-102-103-104-105-106-107-108-109-110-111-112-113-114-115-116-117-118-119-120-121-122-123-124-125-126-127-128-129-130-131-132-133-134-135-136-137-138-139-140-141-142-143-144-145-146-147-148-149-150-151-152-153-154-155-156-157-158-159-160-161-162-163-164-165-166-167-167-168-169-170-171-172-173-174-175-176-178-179-180-181-182-183-184-185-186
Jalur 18	
Test Case	Jika baris yang akan diisi sama dengan baris kunci yang telah diisi sebelumnya maka tidak diubah
Target yang Diharapkan	Menghitung dan mengisi baris nilai z tabel simplex baru.
Hasil Pengujian	Benar
Path / Jalur	37-38-39-40-41-42-43-45-56-57-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62-63-64-65-66-67-68-69-70-71-72-73-74-75-76-77-78-79-80-81-82-83-84-85-86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-101-102-103-104-105-106-107-108-109-110-111-112-113-114-115-116-117-118-119-120-121-122-123-124-125-126-127-128-129-130-131-132-133-134-135-136-137-138-139-140-141-142-143-144-145-146-147-148-149-150-151-152-153-154-155-156-157-158-159-160-161-162-163-164-165-166-167-167-168-169-170-171-172-173-174-175-176-178-179-180-181-182-183-184-185-186
Jalur 19	
Test Case	Jika tabel simplex baru telah terbentuk
Target yang Diharapkan	Mencari nilai optimasi produksi dan menampilkannya pada sistem
Hasil Pengujian	Benar
Path / Jalur	37-38-39-40-41-42-43-45-56-57-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62-63-64-65-66-67-68-69-70-71-72-73-74-75-76-77-78-79-80-81-82-83-84-85-86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-101-102-103-104-105-106-107-108-109-110-111-112-113-114-115-116-117-118-119-120-121-122-123-124-125-126-127-128-129-130-131-132-133-134-135-136-137-138-139-140-141-142-143-144-145-146-147-148-149-150-151-152-153-154-155-156-157-158-159-160-161-162-163-164-165-166-167-167-168-169-170-171-172-

173-174-175-176-178-179-180-181-182-183-184-185-186

2) *White box Testing class model linear*

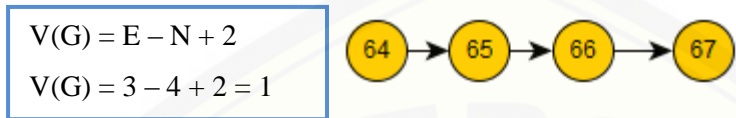
Pengujian white box menggunakan tahapan listing program, diagram alir, dan pengujian test case.



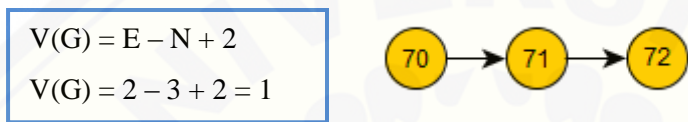
Gambar . Diagram alir method insertData()



Gambar . Diagram alir method get_nilai_optimasi()



Gambar . Diagram alir method get_jumlahx()



Gambar . Diagram alir method insertindex()



Gambar . Diagram alir method get_min_index()

Tabel . Test case function insertdata()

<i>Test case function insertdata()</i>	
Jalur 1	
Test Case	Jika ada data pada tabel tb_batasan dan tb_simplex
Target yang Diharapkan	Menghapus data yang ada pada tabel tb_batasan dan data yang ada pada tabel tb_simplex
Hasil Pengujian	Benar
Path / Jalur	5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56
Jalur 2	
Test Case	Jika ada data bahan baku
Target yang Diharapkan	Mencari data bahan baku dan menyimpan sementara data bahan baku di dalam array
Hasil Pengujian	Benar
Path / Jalur	5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56

Jalur 3	
Test Case	Jika ada data produk
Target yang Diharapkan	Mencari data produk dan menyimpan sementara ke dalam array
Hasil Pengujian	Benar
Path / Jalur	5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56
Jalur 4	
Test Case	Jika ada data pemakaian bahan baku untuk produk yoghurt
Target yang Diharapkan	Mencari data jumlah pemakaian setiap bahan baku untuk produk yoghurt dan menyimpan data jumlah pemakaian setiap bahan baku untuk produk yoghurt kedalam array.
Hasil Pengujian	Benar
Path / Jalur	5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56
Jalur 5	
Test Case	Jika ada data pemakaian bahan baku untuk produk keju
Target yang Diharapkan	Mencari data jumlah pemakaian setiap bahan baku untuk produk keju mozzarella dan menyimpan data jumlah pemakaian setiap bahan baku untuk produk yoghurt kedalam array.
Hasil Pengujian	Benar
Path / Jalur	5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56
Jalur 6	
Test Case	Jika ada data pemakaian bahan baku untuk produk yoghurt dan keju
Target yang Diharapkan	Menyimpan data pemakaian bahan baku yang telah di tampung di dalam array ke dalam tabel di database
Hasil Pengujian	Benar
Path / Jalur	5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56
Jalur 7	
Test Case	Jika ada data bahan baku

Target yang Diharapkan	Mengambil jumlah total jenis bahan baku
Hasil Pengujian	Benar
Path / Jalur	5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56
Jalur 8	
Test Case	Jika ada data mesin
Target yang Diharapkan	Mencari data pada tabel mesin di database dan menyimpannya sementara di dalam array
Hasil Pengujian	Benar
Path / Jalur	5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56
Jalur 9	
Test Case	Jika ada data pemakaian jam kerja mesin untuk produk yoghurt
Target yang Diharapkan	Mencari data jumlah pemakaian jam kerja mesin untuk produk yoghurt dan menyimpan data jumlah pemakaian jam kerja mesin kedalam array.
Hasil Pengujian	Benar
Path / Jalur	5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56
Jalur 10	
Test Case	Jika ada data pemakaian jam kerja mesin untuk produk keju
Target yang Diharapkan	Mencari data jumlah pemakaian jam kerja mesin untuk produk keju dan menyimpan data jumlah pemakaian jam kerja mesin kedalam array.
Hasil Pengujian	Benar
Path / Jalur	5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56
Jalur 11	
Test Case	Jika ada data pemakaian jam kerja mesin untuk produk yoghurt
Target yang Diharapkan	Mencari data jumlah pemakaian setiap bahan baku untuk produk yoghurt dan menyimpan data jumlah pemakaian setiap bahan baku untuk produk yoghurt kedalam array.
Hasil Pengujian	Benar
Path / Jalur	5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-

	24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56
Jalur 12	
Test Case	Jika ada data mesin
Target yang Diharapkan	Mengambil jumlah total jenis mesin
Hasil Pengujian	Benar
Path / Jalur	5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56

Tabel . Test case function get_nilai_optimasi()

<i>Test case function get_nilai_optimasi()</i>	
Jalur 1	
Test Case	Jika ada nilai optimasi
Target yang Diharapkan	Mengambil nilai optimasi di database
Hasil Pengujian	Benar
Path / Jalur	59-60-61-62

Tabel . Test case function get_jumlahx()

<i>Test case function get_jumlahx()</i>	
Jalur 1	
Test Case	Jika ada data pada tb_batasan di database
Target yang Diharapkan	Mengambil jumlah total data pada tb_batasan
Hasil Pengujian	Benar
Path / Jalur	64-65-66-67

Tabel . Test case function insertindex()

<i>Test case function insertindex()</i>	
Jalur 1	
Test Case	Jika ada inputan nilai index
Target yang Diharapkan	Insert nilai index ke tabel dalam database
Hasil Pengujian	Benar
Path / Jalur	70-71-72

Tabel . Test case function get_min_index()

<i>Test case function get_min_index()</i>	
Jalur 1	
Test Case	Jika ada data index
Target yang Diharapkan	Mengambil data nilai index dari database
Hasil Pengujian	Benar
Path / Jalur	74 – 75 – 76 – 77

4.6.2 Pengujian Black Box

Pengujian *black box* merupakan pengujian yang dilakukan oleh *user* dengan menjalankan program secara langsung dan menganalisis *input* dan *output* yang dihasilkan aplikasi. Pengujian *black box* berfungsi untuk menguji sistem dari segi spesifikasi fungsional sistem dengan tujuan mengetahui apakah fungsi-fungsi, inputan dan keluaran sistem sesuai dengan kebutuhan pengguna. Hasil dari pengujian dengan metode *black box* dapat dilihat pada tabel

Tabel . Hasil Pengujian black box

No	Menu	Fungsi	Kasus	Hasil	Ket
1.	Pngoptimal an produksi dengan Linear Programmi ng	Menu ini digunakan bagian produksi untuk menghitung optimasi jumlah produksi menggunakan metode simplex linear programming	Ketika aktor memilih menu optimasi	Sistem akan menampilkan halaman yang berisi tombol lihat persediaan bahan baku, lihat persediaan jam kerja mesin dan next perhitungan	Ok

			<p>Ketika aktor mengklik tombol lihat persediaan bahan baku</p>	<p>Sistem akan mengarahkan ke halaman bahan baku untuk melihat persediaan bahan baku apabila user ingin mengupdate persediaan</p>	Ok
			<p>Ketika aktor mengklik tombol next perhitungan</p>	<p>Sistem akan menampilkan halaman optimasi yang berisi formulasi fungsi batasan, fungsi tujuan, tabel iterasi (putaran), dan hasil perhitungan optimasi jumlah produksi yang optimal.</p>	Ok

BAB 6 PENUTUP

Bab ini membahas tentang kesimpulan dan saran dari peneliti mengenai penelitian yang telah dilakukan. Kesimpulan dan saran tersebut diharapkan dapat digunakan sebagai acuan dalam penelitian selanjutnya

6.1 Kesimpulan

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

1. Hasil penelitian menunjukkan bahwa dengan jumlah persediaan sumberdaya yang sama jumlah produksi aktual yang dilakukan oleh perusahaan berbeda dengan jumlah produksi optimal yang menggunakan metode *simplex linear programming*. Perhitungan jumlah produksi dengan metode *simplex linear programming* dapat mengoptimalkan jumlah produksi, sehingga meningkatkan keuntungan perusahaan dibandingkan dengan perhitungan jumlah produksi aktual.
2. Sistem ini dibangun berbasis *website* dengan 3 hak akses yaitu Pemilik, Bagian Produksi dan Admin. Sistem ini dikembangkan dengan menggunakan model pengembangan *waterfall* karena prosesnya yang dilakukan secara bertahap dan terstruktur namun pada penerapannya memiliki kelemahan yaitu terjadi proses pengulangan sehingga menyebabkan proses pengembangan kurang mengikuti urutan seperti pada teori.

6.2 Saran

Beberapa saran dan masukan yang diharapkan dapat memberikan perbaikan sistem dalam penelitian selanjutnya ialah sebagai berikut:

1. Pengembangan sistem pada penelitian ini selanjutnya diharapkan dapat menambahkan jenis produk yang akan diproduksi yang berfungsi sebagai fungsi tujuannya.

2. Pengembangan sistem pada penelitian ini selanjutnya diharapkan dapat menambahkan fungsi kendala lainnya seperti fungsi kendala sumberdaya manusia (tenaga kerja).
3. Pengembangan sistem pada penelitian selanjutnya disarankan untuk menambahkan fitur gambar grafik persamaan linearnya.
4. Pengembangan sistem pada penelitian berikutnya diharapkan sistem dapat mengkonversikan fungsi kendala kedalam satuan uang atau rupiah agar dapat lebih mengetahui biaya produksi yang digunakan atau diperlukan oleh perusahaan setiap akan melakukan proses produksi.

DAFTAR PUSTAKA

- Agissa, W. (2013). *White box and Black Box Testing*. Retrieved April 03, 2016, from <http://bangwildan.web.id/berita-176-white-box-testing--black-box-testing.html>
- Elmabrouk, O. M. (2011). A Linear Programming Technique for the Optimazation of the Activities in Maintenance Project. *11* (1).
- Fatta, H. A. (2007). *Analisis dan Perancangan Sistem Informasi untuk Keunggulan Bersaing Perusahaan dan Organisasi Modern*. Yogyakarta: ANDI OFFSET.
- Moengin, P., Fransiska, W. A., Member, & IAENG. (2011). An Integer Linear Programming Model and Heuristics Scheduling EDD for Hardboards Cutting Problem in a Modular Production System. *11* (1).
- Mulyono, S. (1999). *Dasar-Dasar Operations Research Edisi 2*. Jakarta: Penerbit Fakultas Ekonomi Universitas Indonesia.
- Prawirosentono, S. (2005). *Riset Operasi dan Ekonofisika*. Jakarta: PT. Bumi Aksara.
- Pressman, R. S. (2002). *Rekayasa Perangkat Lunak Pendekatan Praktisi Buku 1*. Yogyakarta: Andi.
- Rustiani. (2006). Optimalisasi Distribusi Sarimi Pada PT Sari Indo Prakarsa Di Wilayah Bogor Dan Depok.
- Sommerville, I. (2011). *Software Engineering (Rekayasa Perangkat Lunak)*. Jakarta: Erlangga.
- Subagyo, P., Asri, M., & Handoko, T. H. (2011). *Dasar-Dasar Operations Research Edisi 2*. Yogyakarta: BPFE.

LAMPIRAN

LAMPIRAN A. USE CASE SKENARIO

A.1 Use Case Skenario Login

Tabel Use Case Skenario Login

Nama Usecase	Login
Aktor	Admin, Pemilik dan Pegawai Produksi
Deskripsi Singkat	Melakukan Login
Prekondisi	Admin, Pemilik dan Pegawai Produksi membuka sistem.
Prakondisi	Admin, Pemilik dan Pegawai Produksi berhasil login
Skenario Normal Login	
Aktor	Sistem
1. Menjalankan sistem	2. Menampilkan halaman Login
3. Mengisi username dan password pada form login	5. Memeriksa username dan password ke database
4. Klik tombol Login	6. Menampilkan halaman dashboard sesuai hak akses masing-masing user.
Skenario Alternatif Login	
Jika form tidak lengkap	
Aktor	Sistem
	3a. Menampilkan <i>message</i> “please fill out this field” pada kolom login yang belum diisi
Jika username atau password salah	
Aktor	Sistem
	6b. Menampilkan <i>message</i> “Username dan Password tidak sesuai” dengan tombol OK
7b. Klik Ok	

8b. Menampilkan halaman Login

A.2 Use Case Skenario Update Data Produk

Tabel Use Case Skenario Update Data Produk

Nama Usecase	Update Data Produk
Aktor	Admin
Deskripsi Singkat	Update data Jenis Produk
Prekondisi	Admin memilih data Jenis Produk
Prakondisi	Admin berhasil mengubah data jenis produk
Skenario Normal	
Aktor	Sistem
1. Memilih menu Jenis Produk	2. Mengambil data jenis produk dari <i>database</i>
	3. Menampilkan data jenis produk dengan tombol <i>detail</i> dan tombol <i>update</i>
4. Klik <i>update</i> pada data jenis produk yang akan diubah	5. Mengambil data jenis produk yang dipilih dari <i>database</i>
	6. Menampilkan halaman <i>update</i> jenis produk dengan tombol <i>update</i> dan <i>cancel</i>
7. Mengubah nilai kolom edit jenis produk	
8. Klik tombol <i>update</i> Jika <i>cancel</i> lanjut ke nomor 7a.	9. Menyimpan data yang diubah ke <i>database</i>
	10. Menampilkan data Jenis produk dalam bentuk tabel di halaman jenis produk dan menampilkan pemberitahuan "Success data berhasil diupdate"
Skenario Alternatif Mengubah Data Jenis produk	
Jika aktor menekan tombol <i>cancel</i>	
7a. Klik <i>cancel</i>	8a. Menampilkan halaman <i>update</i> jenis produk

A.3 Use Case Skenario Mengelola data satuan

Tabel Use Case Skenario Mengelola data satuan

Nama Usecase	Mengelola data satuan
Aktor	Admin
Deskripsi Singkat	Mengelola data satuan
Prekondisi	Admin memilih data satuan
Prakondisi	Admin berhasil melihat, menambah, menghapus dan mengubah data satuan.
Skenario Normal View Data Satuan	
Aktor	Sistem
1. Memilih menu Satuan	2. Mengambil data User
	3. Mengambil data satuan dari database, tombol tambah, tombol <i>update</i> dan tombol <i>delete</i>
Skenario Normal - Create Data Satuan	
Aktor	Sistem
4. Klik tombol Tambah	5. Menampilkan halaman form tambah satuan
6. Mengisi seluruh form data satuan (nama satuan)	
7. Klik tombol Tambah	8. Menyimpan ke databse
	9. Menampilkan data satuan dalam bentuk tabel di halaman satuan
Skenario Alternatif Menambah Satuan	
Jika aktor tidak mengisi form isian secara lengkap	
Aktor	Sistem
6a. Tidak Mengisi isian form tambah data satuan	
7a. Klik tombol Tambah	8a. Web browser memeriksa masukan form data satuan
	9a. Menampilkan <i>message</i> dengan menunjuk bagian form yang belum diisi. <i>Message</i> peringatan akan muncul disamping masing-masing <i>field</i> masukan.
Jika aktor menekan tombol cancel	

7b. Klik tombol Kembali	
	8b. Mengambil data user
	9b. Menampilkan data Satuan dalam bentuk tabel di halaman satuan
Jika inputan pada form sama dengan data yang sudah ada di database	
7a. Klik tombol Submit	
	8a. Web browser memeriksa masukan form data dengan mencocokkan data yang ada di database
	9a. Menampilkan <i>message</i> “Satuan Sudah Ada”
10a. Klik Ok	
	11a. Kembali ke halaman satuan
Skenario Normal Update Data Satuan	
4. Klik ikon <i>update</i> data pada data satuan yang akan diubah	
	5. Mengambil data satuan yang dipilih dari database
	6. Menampilkan form <i>update</i> satuan
7. Mengubah form edit satuan (nama satuan)	
8. Klik tombol <i>Update</i>	
	9. Menyimpan data yang diubah ke database
	10. Menampilkan data Satuan dalam bentuk tabel di halaman satuan
Skenario Alternatif Update Data Satuan	
Jika aktor tidak mengisi form secara lengkap	
7a. Tidak mengisi form edit satuan	
8a. Klik tombol <i>Update</i>	
	9a. Menampilkan <i>message</i> dengan menunjuk bagian form yang belum diisi. <i>Message</i> peringatan akan muncul disamping masing-masing <i>field</i> masukan.
Jika aktor menekan tombol kembali	
8b. Klik tombol <i>Cancel</i>	
	9a. Menampilkan data Satuan dalam bentuk tabel di halaman satuan
Skenario Normal Menghapus Data Satuan	
Aktor	Sistem

- | | |
|---|--|
| 4. Klik ikon <i>delete</i> data pada data satuan yang akan diubah | 5. Menampilkan <i>alert</i> “Apakah anda yakin akan menghapus data ini?” disertai tombol Ok dan Cancel |
| 6. Klik tombol Ok pada <i>alert</i> | 7. Menghapus data satuan |
| | 8. Mengambil data user dari database |
| | 9. Menampilkan data satuan dalam bentuk tabel pada halaman satuan |

Skenario Alternatif Menghapus Data Satuan

Jika Aktor meng-klik tombol cancel pada *alert*

- | | |
|--|--|
| 6a. Klik tombol Cancel pada <i>alert</i> | 7a. Menampilkan data satuan dalam bentuk tabel pada halaman satuan |
|--|--|

A.4 Use Case Skenario Mengelola data user

Tabel Use Case Skenario Mengelola data user

Nama Usecase	Mengelola data user
Aktor	Admin
Deskripsi Singkat	Mengelola data user
Prekondisi	Admin memilih menu Manajemen User
Prakondisi	Admin berhasil melihat, menambah, menghapus dan mengubah data tahun.

Skenario Normal Melihat Data User

Aktor	Sistem
1. Memilih menu Manajemen User	2. Mengambil data User
	3. Mengambil data user dari database
	4. Menampilkan data user

Skenario Normal Menambah Data User

Aktor	Sistem
1. Klik tombol Tambah Data	2. Mengambil data user dari database
	3. Menampilkan form tambah user
4. Mengisi seluruh form data user	
5. Klik tombol Submit	

	6. Menyimpan ke databse
	7. Menampilkan data user dalam bentuk tabel di halaman user
Skenario Alternatif Menambah User	
Jika aktor tidak mengisi form isian secara lengkap	
Aktor	Sistem
10. Mengisi seluruh isian form tambah data user	
11. Klik tombol Submit	
	12. Web browser memeriksa masukan form data user
	13. Menampilkan <i>message</i> dengan menunjuk bagian form yang belum diisi. <i>Message</i> peringatan akan muncul disamping masing-masing <i>field</i> masukan.
Jika aktor menekan tombol kembali	
7. Klik tombol Tambah Data	
	8. Mengambil data user dari database
	9. Menampilkan form tambah user
10. Klik tombol Kembali	
	11. Mengambil data user
	12. Menampilkan data User dalam bentuk tabel di halaman user
Jika inputan pada form sama dengan data yang sudah ada di database	
10. Mengisi seluruh isian form tambah data user	
11. Klik tombol Submit	
	12. Web browser memeriksa masukan form data dengan mencocokkan data yang ada di databse
	13. Menampilkan <i>message</i> "User Sudah Ada"
14. Klik Ok	
	15. Kembali ke halaman user
Skenario Normal Mengubah Data User	
7. Klik ikon edit data pada data user yang akan diubah	
	8. Mengambil data user dari database
	9. Mengambil data user yang dipilih dari database
	10. Menampilkan form ubah user

11. Mengisi isian form edit user	
12. Klik tombol Submit	
	13. Menyimpan data yang diubah ke database
	14. Menampilkan data User dalam bentuk tabel di halaman user
Skenario Alternatif Mengubah Data User	
Jika aktor tidak mengisi form secara lengkap	
11. Mengisi isian form edit user	
12. Klik tombol Submit	
	13. Web browser memeriksa masukan <i>form</i> ubah data user
	14. Menampilkan <i>message</i> dengan menunjuk bagian form yang belum diisi. <i>Message</i> peringatan akan muncul disamping masing-masing <i>field</i> masukan.
Jika aktor menekan tombol kembali	
7. Klik ikon edit data pada data user yang akan diubah	
	8. Mengambil data user dari database
	9. Mengambil data user yang dipilih dari database
	10. Menampilkan form ubah user
11. Klik tombol Kembali	
	12. Mengambil data user
Skenario Normal Menghapus Data User	
Aktor	Sistem
7. Klik ikon delete data pada data user yang akan diubah	
	8. Menampilkan <i>alert</i> “Apakah anda yakin akan menghapus data ini?” disertai tombol Ok dan Cancel
9. Klik tombol Ok pada <i>alert</i>	
	10. Menghapus data user
	11. Mengambil data user dari database
	12. Menampilkan data user dalam bentuk tabel pada halaman user
Skenario Alternatif Menghapus Data User	
Jika Aktor meng-klik tombol cancel pada <i>alert</i>	
7. Klik ikon delete data pada data user yang akan diubah	

	8. Menampilkan <i>alert</i> “Apakah anda yakin akan menghapus data ini?” disertai tombol Ok dan Cancel
9. Klik tombol Cancel pada <i>alert</i>	
	10. Menampilkan data user dalam bentuk tabel pada halaman user

A.5 Use Case Skenario View Data Produk

Tabel Use Case Skenario View Data Produk

Nama Usecase	View Data Produk
Aktor	Pemilik
Deskripsi Singkat	View data Jenis Produk
Prekondisi	Pemilik memilih data Jenis Produk
Prakondisi	Pemilik berhasil melihat data jenis produk
Skenario Normal Melihat Data Produk	
Aktor	Sistem
1. Memilih menu Jenis Produk	2. Mengambil data User
	3. Mengambil data jenis produk dari database
	4. Menampilkan halaman data jenis produk

A. 6 Use Case Skenario Mengelola Bahan Baku

Tabel Use Case Skenario Mengelola Bahan Baku

Nama Usecase	Mengelola data bahan baku
Aktor	Bagian Produksi
Deskripsi Singkat	Mengelola data bahan baku
Prekondisi	Admin memilih data bahan baku
Prakondisi	Admin berhasil melihat, menambah, menghapus dan mengubah data bahan baku.
Skenario Normal View Data Bahan baku	
Aktor	Sistem
1. Memilih menu Data Master	1. Menampilkan sub menu data master (menu bahan baku, menu Mesin,

	menu Komposisi dan menu penggunaan jam kerja mesin)
2. Memilih menu bahan baku	
	3. Mengambil data user dari database
	4. mengambil data bahan baku berupa data nama bahan dan data persediaan dari tabel tb_bahanbaku dan data satuan dari tb_satuan pada database
	5. menampilkan data nama bahan baku, persediaan bahan baku dan tombol tambah update dan delete pada halaman data bahan baku

Skenario Normal - Create Data Bahan baku

Aktor	Sistem
7. Klik tombol Tambah	
	8. Mengambil data satuan dari database
	9. Menampilkan form input bahan baku dengan tombol tambah pada halaman form input data bahan baku
10. Mengisi seluruh form data bahan baku (nama bahan, jumlah persediaan dan satuannya)	
11. Klik tombol Tambah	
	12. Menyimpan data inputan berupa data nama bahan dan persediaan kedalam tabel tb_bahanbaku pada database
	13. mengambil data bahan baku berupa data nama bahan dan data persediaan dari tabel tb_bahanbaku dan data satuan dari tb_satuan pada database dan menampilkan pemberitahuan "Success data berhasil ditambahkan"

Skenario Alternatif Create Bahan baku

Jika aktor tidak mengisi form isian secara lengkap

Aktor	Sistem
10a. Tidak Mengisi isian form tambah data satuan	
11a. Klik tombol Tambah	
	8a. sistem memeriksa masukan form data satuan
	9a. Menampilkan <i>message</i> dengan menunjuk bagian form yang belum

	diisi. <i>Message</i> peringatan akan muncul disamping masing-masing <i>field</i> masukan yang masih kosong.
Jika aktor tidak mengisi form isian sesuai dengan tipe data	
10b. Tidak Mengisi isian form sesuai dengan tipe datanya	
11b. Klik tombol Tambah	
	12b. sistem memeriksa masukan form data satuan
	13b. Menampilkan pop Up “Gagal tambah data”
14b. klik Oke	
	15b. Menampilkan halaman form tambah bahan baku dan menampilkan <i>message</i> dengan menunjuk bagian form yang tidak sesuai dengan tipe data. <i>Message</i> peringatan “ <i>field must contain only numbers</i> ” akan muncul disamping masing-masing <i>field</i> yang tidak sesuai dengan tipe data.
Jika aktor menekan tombol cancel	
11c. Klik tombol Kembali	
	8c. Mengambil data user
	9c. Menampilkan data bahan baku dalam bentuk tabel di halaman bahan baku
Jika inputan pada form sama dengan data yang sudah ada di database	
11d. Klik tombol tambah	
	12d. Sistem memeriksa masukan form data dengan mencocokkan data yang ada di databse
	13d. Menampilkan <i>pop Up</i> “Data Bahan Sudah ada!”
14d. Klik Ok	
	15d. Kembali menampilkan halaman data bahan baku
Skenario Normal Update Data Bahan Baku	
6. Klik <i>update</i> data pada data satuan yang akan diubah	
	8. mengambil data bahan baku berupa data nama bahan dan data persediaan

	dari tabel <i>tb_bahanbaku</i> dan data satuan dari <i>tb_satuan</i> pada database berdasarkan id bahanbaku yang pilih.
	9. menampilkan data nama bahan baku dan persediaan serta tombol submit pada form edit bahan baku
10. Mengubah form edit bahan baku (nama bahan, persediaan dan satuan)	
11. Klik tombol <i>Update</i>	
	12. menyimpan isian form edit bahanbaku kedalam tabel <i>tb_bahanbaku</i> berdasarkan id bahan baku yang di edit.
	13. mengambil data bahan baku berupa data nama bahan dan data persediaan dari tabel <i>tb_bahanbaku</i> dan data satuan dari <i>tb_satuan</i> pada database.
	14. menampilkan data nama bahan baku, persedian bahan baku dan tombol tambah update dan delete pada halaman data bahan baku dan menampilkan pesan “Success! Data berhasil di update”
Skenario Alternatif Update Data Bahan baku	
Jika aktor tidak mengisi form secara lengkap	
10a. Tidak mengisi form edit bahan baku secara lengkap	
11a. Klik tombol <i>Update</i>	
	12a. Menampilkan <i>message</i> dengan menunjuk bagian form yang belum diisi. <i>Message</i> peringatan “field is required.” akan muncul disamping masing-masing <i>field</i> masukan.
Jika aktor tidak mengisi form isian sesuai dengan tipe data	
10b. Tidak Mengisi isian form sesuai dengan tipe datanya	
11b. Klik tombol <i>Update</i>	
	12b. sistem memeriksa masukan form data satuan

	13b. Menampilkan pop Up “Gagal tambah data”
14b. klik Oke	
	15b. Menampilkan halaman form tambah bahan baku dan menampilkan <i>message</i> dengan menunjuk bagian form yang tidak sesuai dengan tipe data. <i>Message</i> peringatan “ <i>field must contain only numbers</i> ” akan muncul disamping masing-masing <i>field</i> yang tidak sesuai dengan tipe data.
Jika aktor menekan tombol kembali	
11c. Klik tombol <i>Cancel</i>	
	12c. Menampilkan data bahan baku dalam bentuk tabel di halaman bahan baku
Skenario Normal Menghapus Data Bahan baku	
Aktor	Sistem
7. Klik ikon <i>delete</i> data pada data bahan baku yang akan diubah	
	8. Menampilkan <i>alert</i> “Apakah anda yakin akan menghapus data ini?” disertai tombol Ok dan Cancel
9. Klik tombol Ok	
	10. Menghapus data bahan baku berdasarkan id bahan baku dari tabel <i>tb_bahanbaku</i> di database
	11. mengambil data bahan baku berupa data nama bahan dan data persedian dari tabel <i>tb_bahanbaku</i> dan data satuan dari <i>tb_satuan</i> pada database menggunakan function <i>index()</i>
	12. menampilkan data nama bahan baku, persedian bahan baku dan tombol tambah update dan delete pada halaman data bahan baku dan menampilkan pesan pemberitahuan “Success! Data berhasil di hapus”
Skenario Alternatif Menghapus Data Bahan Baku	
Jika Aktor meng-klik tombol cancel pada <i>alert</i>	
9a. Klik tombol Cancel pada <i>alert</i>	

10a.	mengambil data bahan baku berupa data nama bahan dan data persediaan dari tabel tb_bahanbaku dan data satuan dari tb_satuan pada database menggunakan function index()
11a.	menampilkan data nama bahan baku, persediaan bahan baku dan tombol tambah update dan delete pada halaman data bahan baku

A. 7 Use Case Skenario View Bahan Baku

Tabel Use Case Skenario View Bahan Baku

Nama Usecase	View Bahan Baku
Aktor	Pemilik
Deskripsi Singkat	View Bahan Baku
Prekondisi	Pemilik memilih menu bahanbaku
Prakondisi	Pemilik berhasil melihat data bahan baku
Skenario Normal Melihat Data Produk	
Aktor	Sistem
1. Memilih menu Data Master	2. Menampilkan sub menu data master (menu bahan baku, menu Mesin, menu Komposisi dan menu penggunaan jam kerja mesin)
3. Memilih menu bahan baku	4. Mengambil data user dari database
	5. Mengambil data bahanbaku dari database
	6. Menampilkan halaman bahan baku

A. 8 Use Case Skenario Mengelola Data Mesin

Tabel Use Case Skenario Mengelola Data Mesin

Nama Usecase	Mengelola data mesin
Aktor	Bagian Produksi
Deskripsi Singkat	Mengelola data mesin
Prekondisi	Admin memilih data mesin

Prakondisi	Admin berhasil melihat, menambah, menghapus dan mengubah data mesin.
Skenario Normal View Data Mesin	
Aktor	Sistem
8. Memilih menu Data Master	1. Menampilkan sub menu data master (menu Bahan Baku, menu Mesin, menu Komposisi dan menu penggunaan jam kerja mesin)
2. Memilih menu mesin	3. Mengambil data user dari database
	4. Mengambil data mesin dari database
	5. Menampilkan halaman mesin dengan tombol tambah, <i>detail</i> , <i>update</i> , <i>delete</i>
Skenario Normal - Create Data Mesin	
Aktor	Sistem
7. Klik tombol Tambah	8. Menampilkan halaman form tambah mesin dengan tombol tambah dan tombol <i>cancel</i>
9. Mengisi seluruh form data mesin (nama mesin, jumlah mesin, jam kerja mesin dan hari kerja aktif)	
10. Klik tombol Tambah	11. Menyimpan ke databse
	12. Menampilkan data mesin dalam bentuk tabel di halaman mesin dan menampilkan pemberitahuan "Success data berhasil ditambahkan"
Skenario Alternatif Create Mesin	
Jika aktor tidak mengisi form isian secara lengkap	
Aktor	Sistem
9a. Tidak Mengisi isian form tambah data mesin	
10a. Klik tombol Tambah	11a. sistem memeriksa masukan form data mesin
	12a. Menampilkan <i>message</i> dengan menunjuk bagian form yang belum diisi. <i>Message</i> peringatan akan

	muncul disamping masing-masing <i>field</i> masukan yang masih kosong.
Jika aktor tidak mengisi form isian sesuai dengan tipe data	
9b. Tidak Mengisi isian form sesuai dengan tipe datanya	
10b. Klik tombol Tambah	
	11b. sistem memeriksa masukan form data mesin
	12b. Menampilkan pop Up “Gagal tambah data”
13b. klik Oke	
	14b. Menampilkan halaman form tambah mesin dan menampilkan <i>message</i> dengan menunjuk bagian form yang tidak sesuai dengan tipe data. <i>Message</i> peringatan “ <i>field must contain only numbers</i> ” akan muncul disamping masing-masing <i>field</i> yang tidak sesuai dengan tipe data.
Jika aktor menekan tombol cancel	
10c. Klik tombol <i>cancel</i>	
	11c. Mengambil data user
	12c. Menampilkan data mesin dalam bentuk tabel di halaman mesin
Jika inputan pada form sama dengan data yang sudah ada di database	
10d. Klik tombol tambah	
	11d. Sistem memeriksa masukan form data dengan mencocokkan data yang ada di databse
	12d. Menampilkan <i>pop Up</i> “Data Bahan Sudah ada!”
13d. Klik Ok	
	14d. Kembali menampilkan halaman data mesin
Skenario Normal Update Data Mesin	
6. Klik <i>update</i> data pada data mesin yang akan diubah	
	8. Mengambil data mesin yang dipilih dari database
	9. Menampilkan form <i>update</i> mesin
10. Mengubah form edit mesin (nama	

mesin, jumlah mesin, jam kerja mesin dan hari kerja aktif)	
11. Klik tombol <i>Update</i>	12. Menyimpan data yang diubah ke database
	13. Menampilkan data mesin dalam bentuk tabel di halaman mesin dan menampilkan pesan “Success! Data berhasil di update”
Skenario Alternatif Update Data Mesin	
Jika aktor tidak mengisi form secara lengkap	
10a. Tidak mengisi form edit mesin secara lengkap	
11a. Klik tombol <i>Update</i>	12a. Menampilkan <i>message</i> dengan menunjuk bagian form yang belum diisi. <i>Message</i> peringatan “field is required.” akan muncul disamping masing-masing <i>field</i> masukan.
Jika aktor tidak mengisi form isian sesuai dengan tipe data	
10b. Tidak Mengisi isian form sesuai dengan tipe datanya	
11b. Klik tombol <i>Update</i>	12b. sistem memeriksa masukan form data mesin
	13b. Menampilkan pop Up “Gagal tambah data”
14b. klik Oke	15b. Menampilkan halaman form tambah mesin dan menampilkan <i>message</i> dengan menunjuk bagian form yang tidak sesuai dengan tipe data. <i>Message</i> peringatan “ <i>field must contain only numbers</i> ” akan muncul disamping masing-masing <i>field</i> yang tidak sesuai dengan tipe data.
Jika aktor menekan tombol kembali	
11c. Klik tombol <i>Cancel</i>	12c. Menampilkan data mesin dalam bentuk tabel di halaman mesin

Skenario Normal Menghapus Data Mesin	
Aktor	Sistem
9. Klik <i>delete</i> data pada data mesin yang akan diubah	13. Menampilkan <i>alert</i> “Apakah anda yakin akan menghapus data ini?” disertai tombol Ok dan Cancel
14. Klik tombol Ok	15. Menonaktifkan data mesin
	16. Mengambil data user dari database
	17. Menampilkan data mesin dalam bentuk tabel pada halaman mesin dan menampilkan pesan pemberitahuan “Success! Data berhasil di hapus”

Skenario Alternatif Menghapus Data Satuan Jika Aktor meng-klik tombol cancel pada <i>alert</i>	
9a. Klik tombol Cancel pada <i>alert</i>	10a. Menampilkan data mesin dalam bentuk tabel pada halaman data mesin

Skenario Normal detail Data Mesin	
Aktor	Sistem
7. Klik <i>detail</i> data pada data mesin yang ingin dilihat.	7. Mengambil data user dari database
	8. Mengambil data mesin yang dipilih dari database
	9. Menampilkan halaman detail data mesin dan tombol Back
10. Klik tombol back	11. Menampilkan data mesin dalam bentuk tabel pada halaman data mesin

A. 9 Use Case Skenario View Data Mesin

Tabel Use Case Skenario View Data Mesin

Nama Usecase	View Mesin
Aktor	Pemilik
Deskripsi Singkat	View Meisn
Prekondisi	Pemilik memilih menu Mesin

Prakondisi	Pemilik berhasil melihat data mesin
Skenario Normal Melihat Data Produk	
Aktor	Sistem
8. Memilih menu Data Master	1. Menampilkan sub menu data master (menu bahan baku, menu Mesin, menu Komposisi dan menu penggunaan jam kerja mesin)
2. Memilih menu mesin	3. Mengambil data user dari database
	4. Mengambil data mesin dari database
	5. Menampilkan halaman mesin

A. 10 Use Case Skenario View Penggunaan Jam Kerja Mesin

Tabel Use Case Skenario View Penggunaan Jam Kerja Mesin

Nama Usecase	View Penggunaan Jam Kerja Mesin
Aktor	Pemilik
Deskripsi Singkat	View Penggunaan Jam Kerja Mesin
Prekondisi	Pemilik memilih menu Penggunaan Jam Kerja Mesin
Prakondisi	Pemilik berhasil melihat data Penggunaan Jam Kerja Mesin
Skenario Normal Melihat Data Produk	
Aktor	Sistem
1. Memilih menu Data Master	2. Menampilkan sub menu data master (menu bahan baku, menu Mesin, menu Komposisi dan menu penggunaan jam kerja mesin)
3. Memilih menu penggunaan jam kerja mesin	4. Mengambil data user dari database
	5. Mengambil data penggunaan jam kerja mesin dari database
	6. Menampilkan halaman penggunaan jam kerja mesin

A. 11 Use Case Skenario Mengelola Penggunaan Jam Kerja Mesin

Tabel Use Case Skenario Mengelola Penggunaan Jam Kerja Mesin

Nama Usecase	Mengelola Penggunaan Jam Kerja Mesin
Aktor	Bagian Produksi
Deskripsi Singkat	Mengelola Penggunaan Jam Kerja Mesin
Prekondisi	Admin memilih data Penggunaan Jam Kerja Mesin
Prakondisi	Admin berhasil melihat, menambah, menghapus dan mengubah data Penggunaan Jam Kerja Mesin
Skenario Normal View Data Penggunaan Jam Kerja Mesin	
Aktor	Sistem
1. Memilih menu Data Master	2. Menampilkan sub menu data master (menu Bahan Baku, menu Mesin, menu Komposisi dan menu penggunaan jam kerja mesin)
3. Memilih menu Data Penggunaan Jam Kerja Mesin	4. Mengambil data user dari database
	5. Mengambil data Penggunaan Jam Kerja Mesin dari database
	6. Menampilkan halaman Penggunaan Jam Kerja Mesin dengan tombol tambah, <i>detail</i> , <i>update</i> , <i>delete</i>
Skenario Normal - Create Data Penggunaan Jam Kerja Mesin	
Aktor	Sistem
7. Klik tombol Tambah	8. Menampilkan halaman form tambah Penggunaan Jam Kerja Mesin dengan tombol tambah dan tombol <i>cancel</i>
9. Mengisi seluruh form data Penggunaan Jam Kerja Mesin	
10. Klik tombol Tambah	11. Menyimpan ke databse
	12. Menampilkan data Penggunaan Jam

	Kerja Mesin dalam bentuk tabel di halaman mesin dan menampilkan pemberitahuan “Success data berhasil ditambahkan”
Skenario Alternatif Create Penggunaan Jam Kerja Mesin	
Jika aktor tidak mengisi form isian secara lengkap	
Aktor	Sistem
9a. Tidak Mengisi isian form tambah data Penggunaan Jam Kerja Mesin	
10a. Klik tombol Tambah	11a. sistem memeriksa masukan form data Penggunaan Jam Kerja Mesin
	12a. Menampilkan <i>message</i> dengan menunjuk bagian form yang belum diisi. <i>Message</i> peringatan akan muncul disamping masing-masing <i>field</i> masukan yang masih kosong.
Jika aktor tidak mengisi form isian sesuai dengan tipe data	
9b. Tidak Mengisi isian form sesuai dengan tipe datanya	
10b. Klik tombol Tambah	11b. sistem memeriksa masukan form data Penggunaan Jam Kerja Mesin
	12b. Menampilkan pop Up “Gagal tambah data”
13b. klik Oke	14b. Menampilkan halaman form tambah Penggunaan Jam Kerja Mesin dan menampilkan <i>message</i> dengan menunjuk bagian form yang tidak sesuai dengan tipe data. <i>Message</i> peringatan “ <i>field must contain only numbers</i> ” akan muncul disamping masing-masing <i>field</i> yang tidak sesuai dengan tipe data.
Jika aktor menekan tombol cancel	
10c. Klik tombol <i>cancel</i>	11c. Mengambil data user
	12c. Menampilkan data Penggunaan Jam Kerja Mesin dalam bentuk tabel

	di halaman Penggunaan Jam Kerja Mesin
Jika inputan pada form sama dengan data yang sudah ada di database	
10d. Klik tombol tambah	
	11d. Sistem memeriksa masukan form data dengan mencocokkan data yang ada di databse
	12d. Menampilkan <i>pop Up</i> "Data Bahan Sudah ada!"
13d. Klik Ok	
	14d. Kembali menampilkan halaman data Penggunaan Jam Kerja Mesin
Skenario Normal Update Data Mesin	
7. Klik <i>update</i> data pada data Penggunaan Jam Kerja Mesin yang akan diubah	
	14. Mengambil data Penggunaan Jam Kerja Mesin yang dipilih dari database
	15. Menampilkan form <i>update</i> Penggunaan Jam Kerja Mesin
16. Mengubah form edit Penggunaan Jam Kerja Mesin	
17. Klik tombol <i>Update</i>	
	18. Menyimpan data yang diubah ke database
	19. Menampilkan data Penggunaan Jam Kerja Mesin dalam bentuk tabel di halaman Penggunaan Jam Kerja Mesin dan menampilkan pesan "Success! Data berhasil di update"
Skenario Alternatif Update Data Penggunaan Jam Kerja Mesin	
Jika aktor tidak mengisi form secara lengkap	
10a. Tidak mengisi form edit Penggunaan Jam Kerja Mesin secara lengkap	
11a. Klik tombol <i>Update</i>	
	12a. Menampilkan <i>message</i> dengan menunjuk bagian form yang belum diisi. <i>Message</i> peringatan "field is required." akan muncul disamping masing-masing <i>field</i> masukan.

Jika aktor tidak mengisi form isian sesuai dengan tipe data

10b. Tidak Mengisi isian form sesuai dengan tipe datanya	
11b. Klik tombol <i>Update</i>	
	12b. sistem memeriksa masukan form data Penggunaan Jam Kerja Mesin
	13b. Menampilkan pop Up “Gagal tambah data”
14b. klik Oke	
	15b. Menampilkan halaman form tambah Penggunaan Jam Kerja Mesin dan menampilkan <i>message</i> dengan menunjuk bagian form yang tidak sesuai dengan tipe data. <i>Message</i> peringatan “ <i>field must contain only numbers</i> ” akan muncul disamping masing-masing <i>field</i> yang tidak sesuai dengan tipe data.

Jika aktor menekan tombol kembali

11c. Klik tombol <i>Cancel</i>	
	12c. Menampilkan data Penggunaan Jam Kerja Mesin dalam bentuk tabel di halaman Penggunaan Jam Kerja Mesin.

Skenario Normal Menghapus Data Mesin

Aktor	Sistem
8. Klik <i>delete</i> data pada data Penggunaan Jam Kerja Mesin yang akan diubah	
	9. Menampilkan <i>alert</i> “Apakah anda yakin akan menghapus data ini?” disertai tombol Ok dan Cancel
10. Klik tombol Ok	
	11. Menonaktifkan data Penggunaan Jam Kerja Mesin
	12. Mengambil data user dari database
	13. Menampilkan data Penggunaan Jam Kerja Mesin dalam bentuk tabel pada halaman mesin dan menampilkan pesan pemberitahuan “ Success! Data berhasil di hapus”

Skenario Alternatif Menghapus Data Penggunaan Jam Kerja Mesin	
Jika Aktor meng-klik tombol cancel pada <i>alert</i>	
9a. Klik tombol Cancel pada <i>alert</i>	10a. Menampilkan data Penggunaan Jam Kerja Mesin dalam bentuk tabel pada halaman data mesin
Skenario Normal detail Data Penggunaan Jam Kerja Mesin	
Aktor	Sistem
9. Klik <i>detail</i> data pada data Penggunaan Jam Kerja Mesin yang ingin dilihat.	10. Mengambil data user dari database
	11. Mengambil data Penggunaan Jam Kerja Mesin yang dipilih dari database
	12. Menampilkan halaman detail data Penggunaan Jam Kerja Mesin dan tombol Back
13. Klik tombol back	14. Menampilkan data Penggunaan Jam Kerja Mesin dalam bentuk tabel pada halaman data Penggunaan Jam Kerja Mesin

A. 12 Use Case Skenario Mengelola Data Produksi

Tabel Use Case Skenario Mengelola Data Produksi

Nama Usecase	Mengelola Penggunaan Jam Kerja Mesin
Aktor	Bagian Produksi
Deskripsi Singkat	Mengelola Penggunaan Jam Kerja Mesin
Prekondisi	Admin memilih data Penggunaan Jam Kerja Mesin
Prakondisi	Admin berhasil melihat, menambah, menghapus dan mengubah data Penggunaan Jam Kerja Mesin
Skenario Normal View Data Penggunaan Jam Kerja Mesin	
Aktor	Sistem
1. Memilih menu Data Master	2. Menampilkan sub menu data master

	(menu Bahan Baku, menu Mesin, menu Komposisi dan menu penggunaan jam kerja mesin)
3. Memilih menu Data Penggunaan Jam Kerja Mesin	
	4. Mengambil data user dari database
	5. Mengambil data Penggunaan Jam Kerja Mesin dari database
	6. Menampilkan halaman Penggunaan Jam Kerja Mesin dengan tombol tambah, <i>detail</i> , <i>update</i> , <i>delete</i>
Skenario Normal - Create Data Penggunaan Jam Kerja Mesin	
Aktor	Sistem
7. Klik tombol Tambah	
	8. Menampilkan halaman form tambah Penggunaan Jam Kerja Mesin dengan tombol tambah dan tombol <i>cancel</i>
9. Mengisi seluruh form data Penggunaan Jam Kerja Mesin	
10. Klik tombol Tambah	
	11. Menyimpan ke databse
	12. Menampilkan data Penggunaan Jam Kerja Mesin dalam bentuk tabel di halaman mesin dan menampilkan pemberitahuan "Success data berhasil ditambahkan"
Skenario Alternatif Create Penggunaan Jam Kerja Mesin	
Aktor	Sistem
9a. Tidak Mengisi isian form tambah data Penggunaan Jam Kerja Mesin	
10a. Klik tombol Tambah	
	11a. sistem memeriksa masukan form data Penggunaan Jam Kerja Mesin
	12a. Menampilkan <i>message</i> dengan menunjuk bagian form yang belum diisi. <i>Message</i> peringatan akan muncul disamping masing-masing <i>field</i> masukan yang masih kosong.

9b. Tidak Mengisi isian form sesuai dengan tipe datanya	
10b. Klik tombol Tambah	
	11b. sistem memeriksa masukan form data Penggunaan Jam Kerja Mesin
	12b. Menampilkan pop Up “Gagal tambah data”
13b. klik Oke	
	14b. Menampilkan halaman form tambah Penggunaan Jam Kerja Mesin dan menampilkan <i>message</i> dengan menunjuk bagian form yang tidak sesuai dengan tipe data. <i>Message</i> peringatan “ <i>field must contain only numbers</i> ” akan muncul disamping masing-masing <i>field</i> yang tidak sesuai dengan tipe data.
Jika aktor menekan tombol cancel	
10c. Klik tombol <i>cancel</i>	
	11c. Mengambil data user
	12c. Menampilkan data Penggunaan Jam Kerja Mesin dalam bentuk tabel di halaman Penggunaan Jam Kerja Mesin
Jika inputan pada form sama dengan data yang sudah ada di database	
10d. Klik tombol tambah	
	11d. Sistem memeriksa masukan form data dengan mencocokkan data yang ada di databse
	12d. Menampilkan <i>pop Up</i> “Data Bahan Sudah ada!”
13d. Klik Ok	
	14d. Kembali menampilkan halaman data Penggunaan Jam Kerja Mesin
Skenario Normal Update Data Mesin	
7. Klik <i>update</i> data pada data Penggunaan Jam Kerja Mesin yang akan diubah	
	8. Mengambil data Penggunaan Jam Kerja Mesin yang dipilih dari database

	9. Menampilkan form <i>update</i> Penggunaan Jam Kerja Mesin
10. Mengubah form edit Penggunaan Jam Kerja Mesin	
11. Klik tombol <i>Update</i>	
	12. Menyimpan data yang diubah ke database
	13. Menampilkan data Penggunaan Jam Kerja Mesin dalam bentuk tabel di halaman Penggunaan Jam Kerja Mesin dan menampilkan pesan “Success! Data berhasil di update”
Skenario Alternatif Update Data Penggunaan Jam Kerja Mesin	
Jika aktor tidak mengisi form secara lengkap	
10a. Tidak mengisi form edit Penggunaan Jam Kerja Mesin secara lengkap	
11a. Klik tombol <i>Update</i>	
	12a. Menampilkan <i>message</i> dengan menunjuk bagian form yang belum diisi. <i>Message</i> peringatan “field is required.” akan muncul disamping masing-masing <i>field</i> masukan.
Jika aktor tidak mengisi form isian sesuai dengan tipe data	
10b. Tidak Mengisi isian form sesuai dengan tipe datanya	
11b. Klik tombol <i>Update</i>	
	12b. sistem memeriksa masukan form data Penggunaan Jam Kerja Mesin
	13b. Menampilkan pop Up “Gagal tambah data”
14b. klik Oke	
	15b. Menampilkan halaman form tambah Penggunaan Jam Kerja Mesin dan menampilkan <i>message</i> dengan menunjuk bagian form yang tidak sesuai dengan tipe data. <i>Message</i> peringatan “ <i>field must contain only numbers</i> ” akan muncul disamping masing-masing <i>field</i> yang tidak sesuai dengan tipe data.

Jika aktor menekan tombol kembali

11c. Klik tombol *Cancel*

12c. Menampilkan data Penggunaan Jam Kerja Mesin dalam bentuk tabel di halaman Penggunaan Jam Kerja Mesin.

Skenario Normal Menghapus Data Mesin

Aktor	Sistem
7. Klik <i>delete</i> data pada data Penggunaan Jam Kerja Mesin yang akan diubah	8. Menampilkan <i>alert</i> “Apakah anda yakin akan menghapus data ini?” disertai tombol Ok dan Cancel
9. Klik tombol Ok	10. Menonaktifkan data Penggunaan Jam Kerja Mesin
	11. Mengambil data user dari database
	12. Menampilkan data Penggunaan Jam Kerja Mesin dalam bentuk tabel pada halaman mesin dan menampilkan pesan pemberitahuan “ Success! Data berhasil di hapus”

Skenario Alternatif Menghapus Data Penggunaan Jam Kerja Mesin

9a. Klik tombol Cancel pada *alert*

10a. Menampilkan data Penggunaan Jam Kerja Mesin dalam bentuk tabel pada halaman data mesin

Skenario Normal detail Data Penggunaan Jam Kerja Mesin

Aktor	Sistem
7. Klik <i>detail</i> data pada data Penggunaan Jam Kerja Mesin yang ingin dilihat.	8. Mengambil data user dari database
	9. Mengambil data Penggunaan Jam Kerja Mesin yang dipilih dari database
	10. Menampilkan halaman detail data Penggunaan Jam Kerja Mesin dan tombol Back
11. Klik tombol back	

-
12. Menampilkan data Penggunaan Jam Kerja Mesin dalam bentuk tabel pada halaman data Penggunaan Jam Kerja Mesin
-

A. 13 *Use Case Skenario View Data Produksi*

Tabel Use Case Skenario View Data Produksi

Nama Usecase	View Data Produksi
Aktor	Pemilik
Deskripsi Singkat	View Data Produksi
Prekondisi	Pemilik memilih menu Data Produksi
Prakondisi	Pemilik berhasil melihat data Data Produksi
Skenario Normal Melihat Data Produk	
Aktor	Sistem
1. Memilih menu Laporan	2. Menampilkan sub menu Laporan (menu produksi dan menu optimasi)
3. Memilih menu produksi	4. Mengambil data user dari database
	5. Mengambil data produksi
	6. Menampilkan halaman produksi
7. Menginputkan tanggal lihat Produksi Berdasarkan Bulan	
8. Klik cek	9. Menampilkan halaman sortir produksi berdasarkan tanggal yang diinputkan

A. 14 *Use Case Skenario Mengelola Data Penggunaan Bahan Baku*

Tabel Case Skenario Mengelola Data Penggunaan Bahan Baku

Nama Usecase	Mengelola Data Penggunaan Bahan Baku
Aktor	Bagian Produksi
Deskripsi Singkat	Mengelola Data Penggunaan Bahan Baku

Prekondisi	Admin memilih data komposisi
Prakondisi	Admin berhasil melihat, menambah, menghapus dan mengubah data komposisi.
Skenario Normal View Data Penggunaan Data Bahan Baku	
Aktor	Sistem
1. Memilih menu Data Master	2. Menampilkan sub menu data master (menu Bahan Baku, menu Mesin, menu Komposisi dan menu penggunaan jam kerja mesin)
3. Memilih menu komposisi	4. Mengambil data user dari database
	5. Mengambil data komposisi dari database
	6. Menampilkan halaman komposisi dengan tombol tambah, <i>detail</i> , <i>update</i> , <i>delete</i>
Skenario Normal - Create Data Mesin	
Aktor	Sistem
7. Klik tombol Tambah	8. Mengambil data bahan baku dan data produk dari database
	9. Menampilkan halaman form tambah komposisi dengan tombol tambah dan tombol <i>cancel</i>
10. Mengisi seluruh form data komposisi (nama bahan baku, nama produk dan jumlah pemakaian)	
11. Klik tombol Tambah	12. Menyimpan ke database
	13. Menampilkan data komposisi dalam bentuk tabel di halaman komposisi dan menampilkan pemberitahuan "Success data berhasil ditambahkan"
Skenario Alternatif Create Komposisi	
Jika aktor tidak mengisi form isian secara lengkap	
Aktor	Sistem
10a. Tidak Mengisi isian form tambah data komposisi	
11a. Klik tombol Tambah	

	12a. sistem memeriksa masukan form data komposisi
	13a. Menampilkan <i>message</i> dengan menunjuk bagian form yang belum diisi. <i>Message</i> peringatan akan muncul disamping masing-masing <i>field</i> masukan yang masih kosong.
Jika aktor tidak mengisi form isian sesuai dengan tipe data	
10b. Tidak Mengisi isian form sesuai dengan tipe datanya	
11b. Klik tombol Tambah	
	12b. sistem memeriksa masukan form data komposisi
	13b. Menampilkan pop Up “Gagal tambah data”
14b. klik Oke	
	15b. Menampilkan halaman form tambah komposisi dan menampilkan <i>message</i> dengan menunjuk bagian form yang tidak sesuai dengan tipe data. <i>Message</i> peringatan “ <i>field must contain only numbers</i> ” akan muncul disamping masing-masing <i>field</i> yang tidak sesuai dengan tipe data.
Jika aktor menekan tombol cancel	
11c. Klik tombol <i>cancel</i>	
	12c. Mengambil data user
	13c. Menampilkan data komposisi dalam bentuk tabel di halaman komposisi
Jika inputan pada form sama dengan data yang sudah ada di database	
11d. Klik tombol tambah	
	12d. Sistem memeriksa masukan form data dengan mencocokkan data yang ada di databse
	13d. Menampilkan <i>pop Up</i> “Data Bahan Sudah ada!”
14d. Klik Ok	
	15d. Kembali menampilkan halaman data komposisi
Skenario Normal Update Data Komposisi	

7. Klik <i>update</i> data pada data komposisi yang akan diubah	
	8. Mengambil data komposisi yang dipilih dari database
	9. Menampilkan form <i>update</i> komposisi
10. Mengubah form edit komposisi	
11. Klik tombol <i>Update</i>	
	12. Menyimpan data yang diubah ke database
	13. Menampilkan data komposisi dalam bentuk tabel di halaman komposisi dan menampilkan pesan “Success! Data berhasil di update”
Skenario Alternatif Update Data Komposisi	
Jika aktor tidak mengisi form secara lengkap	
10a. Tidak mengisi form edit komposisi secara lengkap	
11a. Klik tombol <i>Update</i>	
	12a. Menampilkan <i>message</i> dengan menunjuk bagian form yang belum diisi. <i>Message</i> peringatan “field is required.” akan muncul disamping masing-masing <i>field</i> masukan.
Jika aktor tidak mengisi form isian sesuai dengan tipe data	
10b. Tidak Mengisi isian form sesuai dengan tipe datanya	
11b. Klik tombol <i>Update</i>	
	12b. sistem memeriksa masukan form data komposisi
	13b. Menampilkan pop Up “Gagal tambah data”
14b. klik Oke	
	15b. Menampilkan halaman form tambah komposisi dan menampilkan <i>message</i> dengan menunjuk bagian form yang tidak sesuai dengan tipe data. <i>Message</i> peringatan “ <i>field must contain only numbers</i> ” akan muncul disamping masing-masing <i>field</i> yang tidak

	sesuai dengan tipe data.
Jika aktor menekan tombol kembali	
11c. Klik tombol <i>Cancel</i>	
	12c. Menampilkan data komposisi dalam bentuk tabel di halaman komposisi
Skenario Normal Menghapus Data Komposisi	
Aktor	Sistem
7. Klik <i>delete</i> data pada data komposisi yang akan diubah	
	8. Menampilkan <i>alert</i> “Apakah anda yakin akan menghapus data ini?” disertai tombol Ok dan Cancel
9. Klik tombol Ok	
	10. Menonaktifkan data komposisi
	11. Mengambil data user dari database
	12. Menampilkan data komposisi dalam bentuk tabel pada halaman komposisi dan menampilkan pesan pemberitahuan “ Success! Data berhasil di hapus”
Skenario Alternatif Menghapus Data Komposisi	
Jika Aktor meng-klik tombol cancel pada <i>alert</i>	
9a. Klik tombol Cancel pada <i>alert</i>	
	10a. Menampilkan data komposisi dalam bentuk tabel pada halaman data komposisi
Skenario Normal detail Data Komposisi	
Aktor	Sistem
7. Klik <i>detail</i> data pada data komposisi yang ingin dilihat.	
	8. Mengambil data user dari database
	9. Mengambil data komposisi yang dipilih dari database
	10. Menampilkan halaman detail data komposisi dan tombol Back
11. Klik tombol back	
	12. Menampilkan data komposisi dalam bentuk tabel pada halaman data mesin

A. 15 Use Case Skenario View Data Penggunaan Bahan Baku

Tabel Use Case Skenario View Data Penggunaan Bahan Baku

Nama Usecase	View Data Penggunaan Bahan Baku
Aktor	Pemilik
Deskripsi Singkat	View Data Penggunaan Bahan Baku
Prekondisi	Pemilik memilih menu Komposisi
Prakondisi	Pemilik berhasil melihat data komposisi
Skenario Normal Melihat Data Produk	
Aktor	Sistem
9. Memilih menu Data Master	10. Menampilkan sub menu data master (menu bahan baku, menu Mesin, menu Komposisi dan menu penggunaan jam kerja mesin)
11. Memilih menu komposisi	12. Mengambil data user dari database
	13. Mengambil data komposisi dari database
	14. Menampilkan halaman komposisi

A. 16 Use Case Skenario Pengoptimalan dengan *Linear Programming*Tabel Use Case Skenario Pengoptimalan dengan *Linear Programming*

Nama Usecase	Pengoptimalan dengan <i>Linear Programming</i>
Aktor	Bagian Produksi
Deskripsi Singkat	Perhitungan Optimasi jumlah produksi dengan <i>Linear Programming</i>
Prekondisi	Pemilik memilih menu Pengoptimalan dengan <i>Linear Programming</i>
Prakondisi	Pemilik berhasil menghitung optimasi produksi dengan sistem dan melihat hasil perhitungan optimasi
Skenario Normal Melihat Data Produk	
Aktor	Sistem

15. Memilih menu Optimasi	16. Menampilkan halaman cek dengan tombol lihat persediaan bahan baku, tombol lihat persediaan jam kerja mesin dan tombol next perhitungan
17. Klik tombol next perhitungan	18. Mengambil data user dari database
	19. Melakukan perhitungan optimasi produksi dengan metode <i>simplex Linear Programming</i>
	20. Menampilkan halaman hasil perhitungan optimasi dan tombol simpan
Skenario Alternatif - klik lihat persediaan bahan baku	
7. Klik lihat persediaan bahan baku	8. Mengambil data user dari database
	21. Mengambil data bahanbaku dari database
	22. Menampilkan halaman bahan baku
Skenario Alternatif - klik lihat persediaan jam kerja mesin	
9. Klik lihat persediaan jam kerja mesin	10. Mengambil data user dari database
	23. Mengambil data jam kerja mesin dari database
	24. Menampilkan halaman jam kerja mesin

A. 17 Use Case Skenario Cetak Laporan Perhitungan optimasi produksi

Tabel Use Case Skenario Cetak Laporan Perhitungan optimasi produksi

Nama Usecase	Cetak Laporan Perhitungan optimasi produksi
Aktor	Bagian Produksi
Deskripsi Singkat	Mencetak Laporan Perhitungan optimasi produksi
Prekondisi	Pemilik memilih menu Cetak Laporan Perhitungan optimasi produksi
Prakondisi	Pemilik berhasil mencetak Laporan

Perhitungan optimasi produksi	
Skenario Normal Melihat Data Produk	
Aktor	Sistem
	1. Menampilkan halaman hasil perhitungan optimasi dan tombol simpan
2. Menginputkan tanggal optimasi	
3. Klik tombol simpan	
	4. Menyimpan data optimasi kedalam database
	5. Menampilkan halaman cetak laporan
6. Klik tombol print laporan	
	7. Mengunduh dan menyimpan laporan dalam bentuk word kedalam komputer

A. 18 Use Case Skenario View Optimasi Produksi - pemilik

Tabel Use Case Skenario View Optimasi Produksi - pemilik

Nama Usecase	View Optimasi Produksi
Aktor	Pemilik
Deskripsi Singkat	View Optimasi Produksi
Prekondisi	Pemilik memilih menu Optimasi
Prakondisi	Pemilik berhasil melihat data optimasi

Skenario Normal Melihat Data Produk	
Aktor	Sistem
1. Memilih menu Data Laporan	
	2. Menampilkan sub menu Laporan (menu produksi dan menu optimasi)
3. Memilih menu optimasi	
	4. Mengambil data user dari database
	5. Mengambil data optimasi dari database
	6. Menampilkan halaman optimasi dan tombol <i>detail</i>
7. Klik <i>detail</i> pada data optimasi yang ingin dilihat lebih rinci.	
	8. Mengambil data detail optimasi

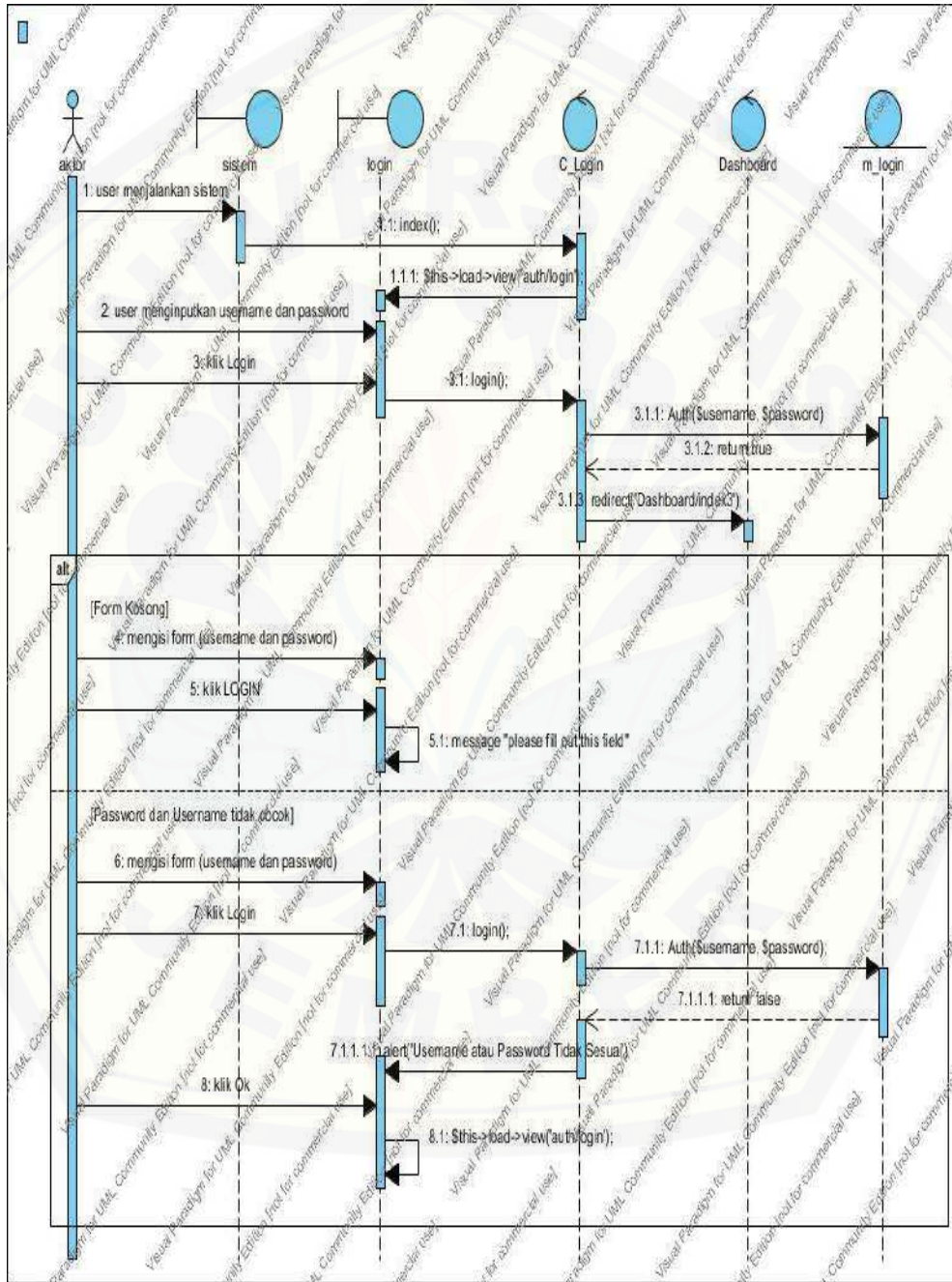
sesuai dengan id optimasi yang
dipilih dari database

9. Menampilkan halaman detail
optimasi.



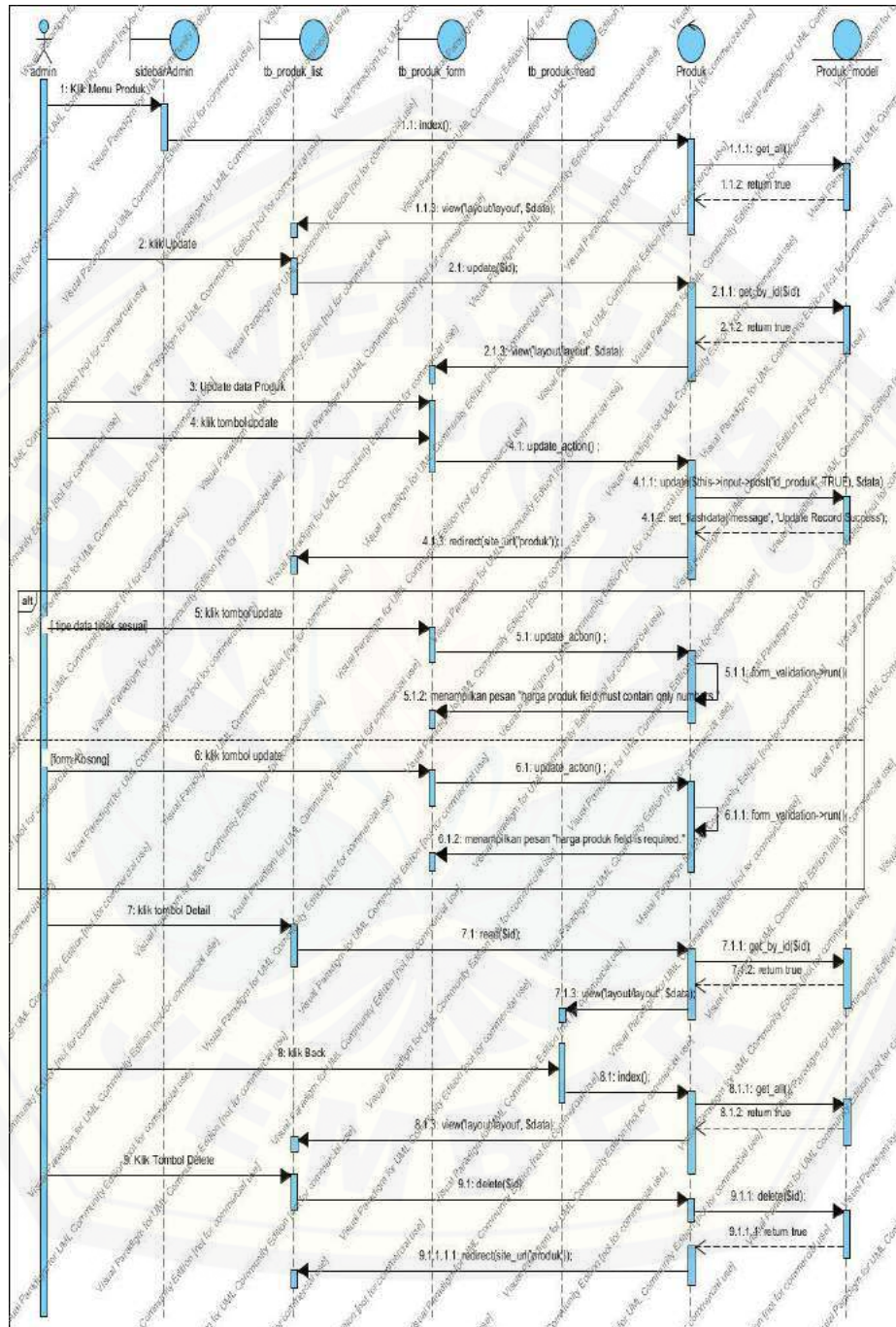
LAMPIRAN B. SEQUENCE DIAGRAM

B.1 Sequence Login



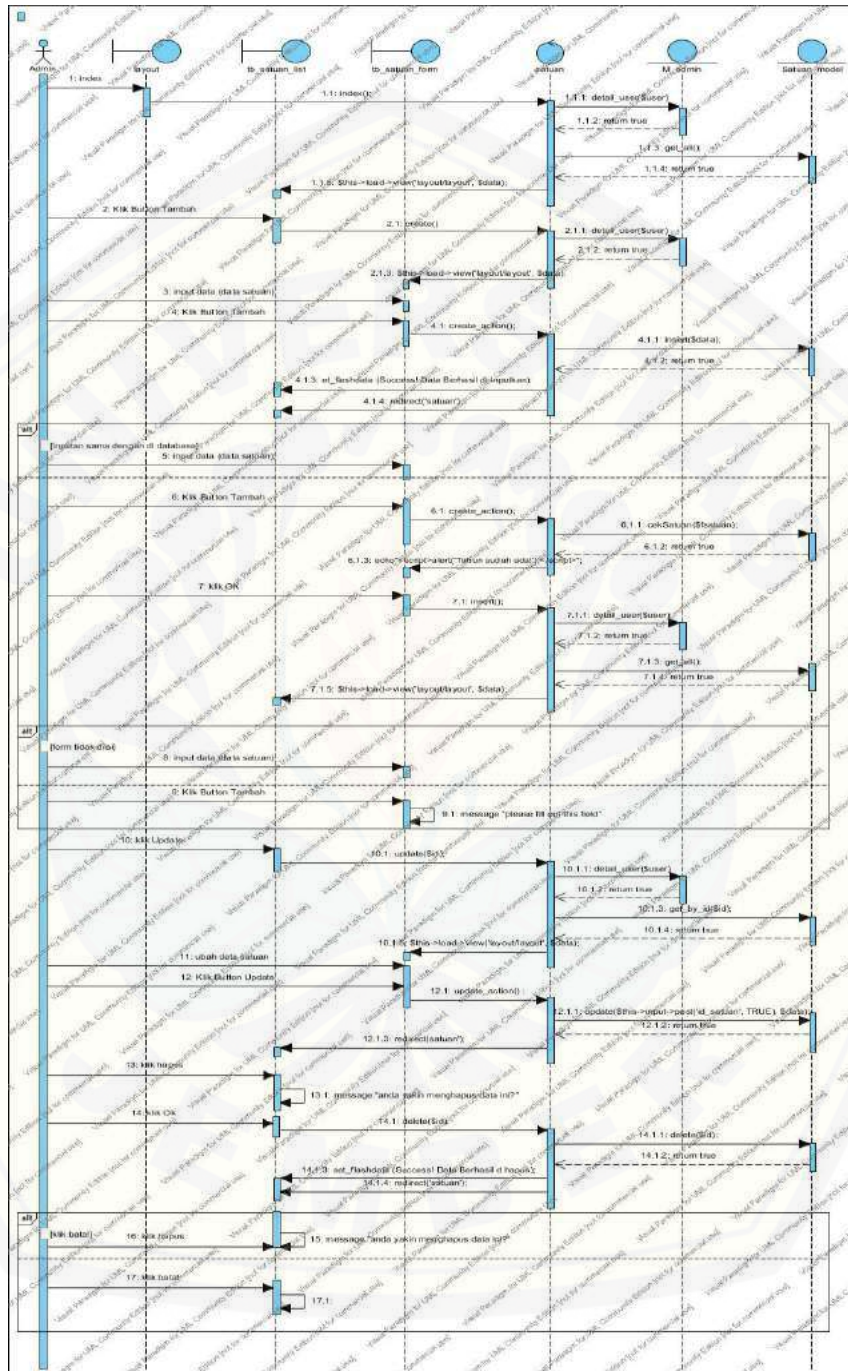
Gambar Sequence Login

B.2 Sequence Update Produk



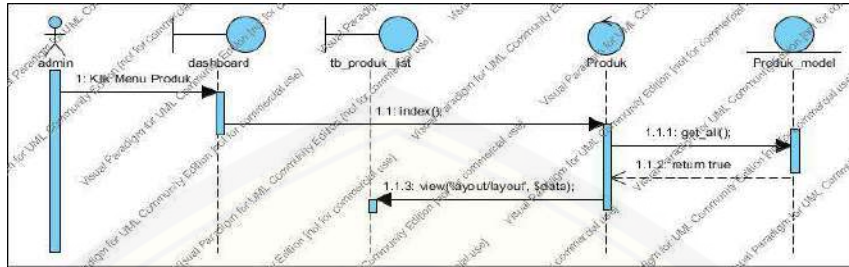
Gambar Sequence Update Produk

B. 3 Sequence Mengelola Data Satuan



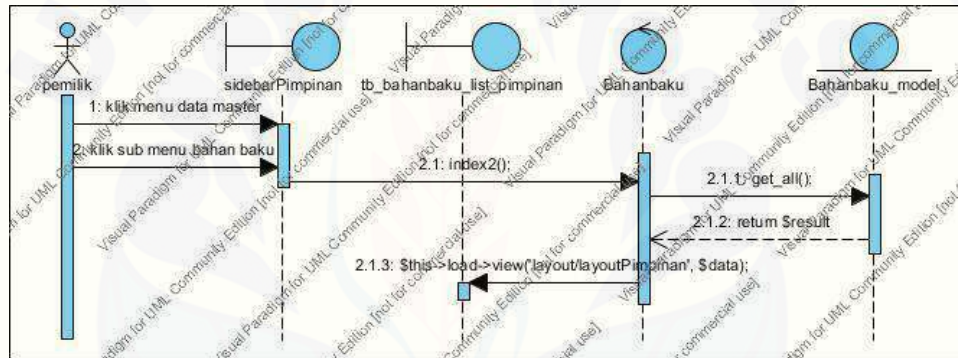
Gambar Sequence Mengelola Data Satuan

B. 4 Sequence View Data Produk



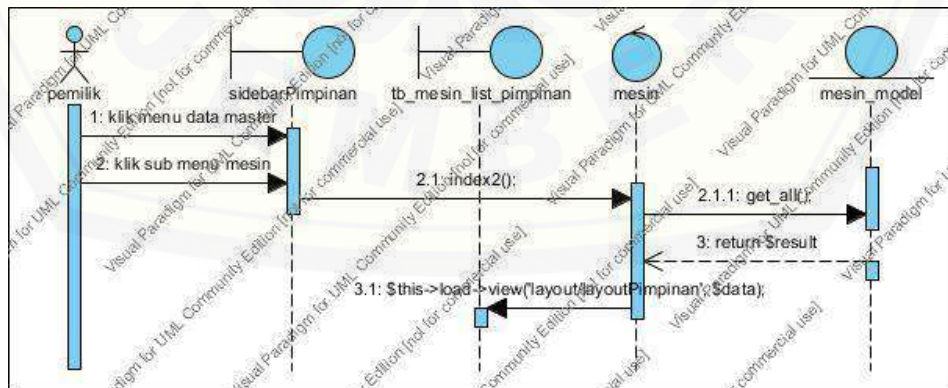
Gambar Sequence View Data Produk

B. 5 Sequence View Bahan Baku



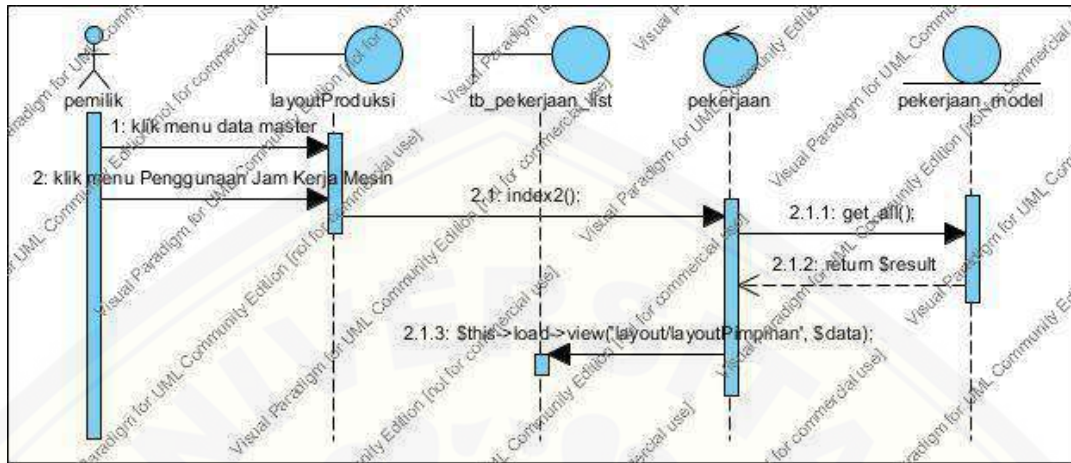
Gambar Sequence View Bahan Baku

B. 6 Sequence View Data Mesin



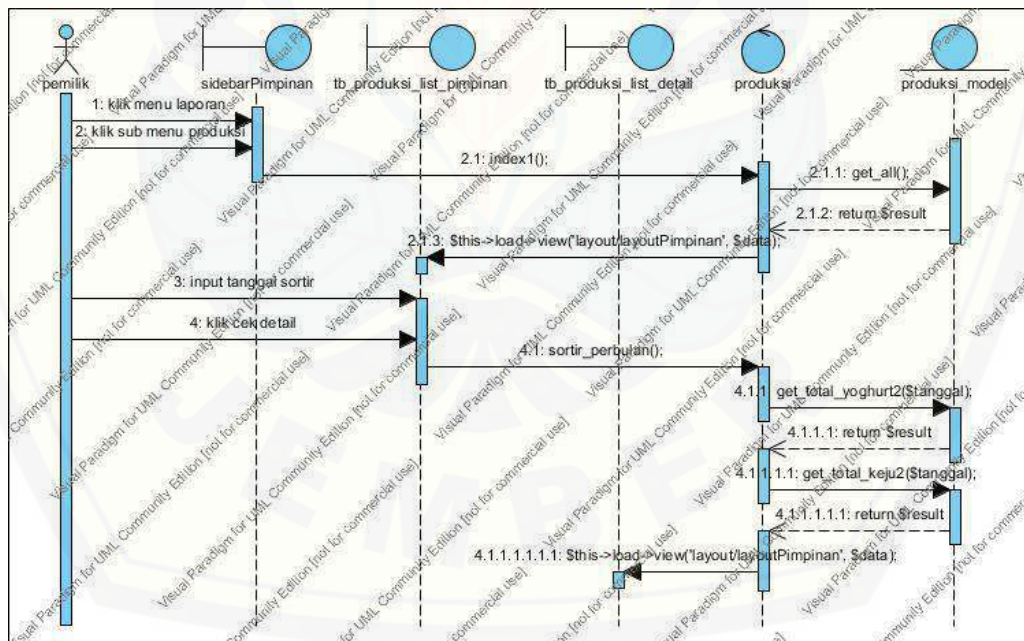
Gambar Sequence View Data Mesin

B. 7 Sequence View Penggunaan Jam Kerja Mesin



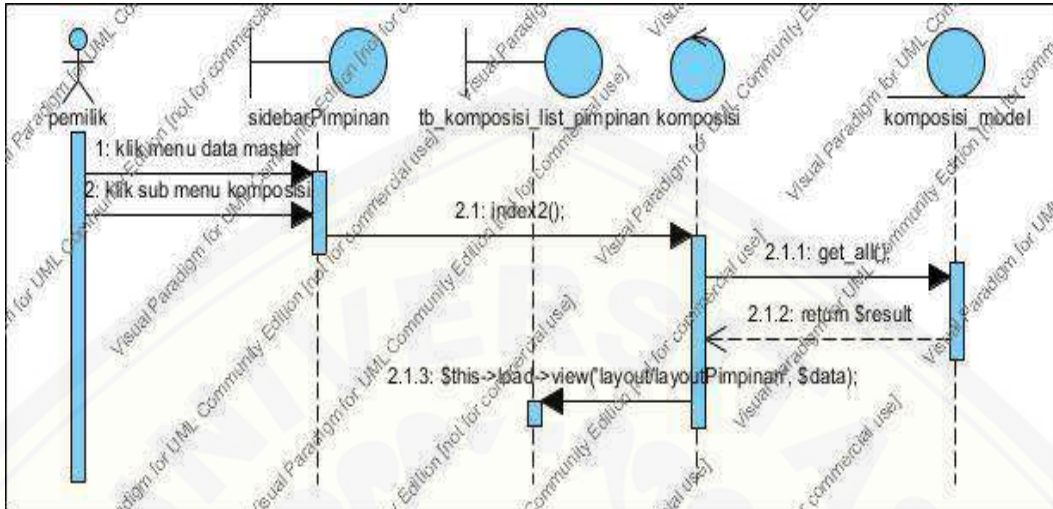
Gambar Sequence View Penggunaan Jam Kerja Mesin

B. 8 Sequence View Data Produksi



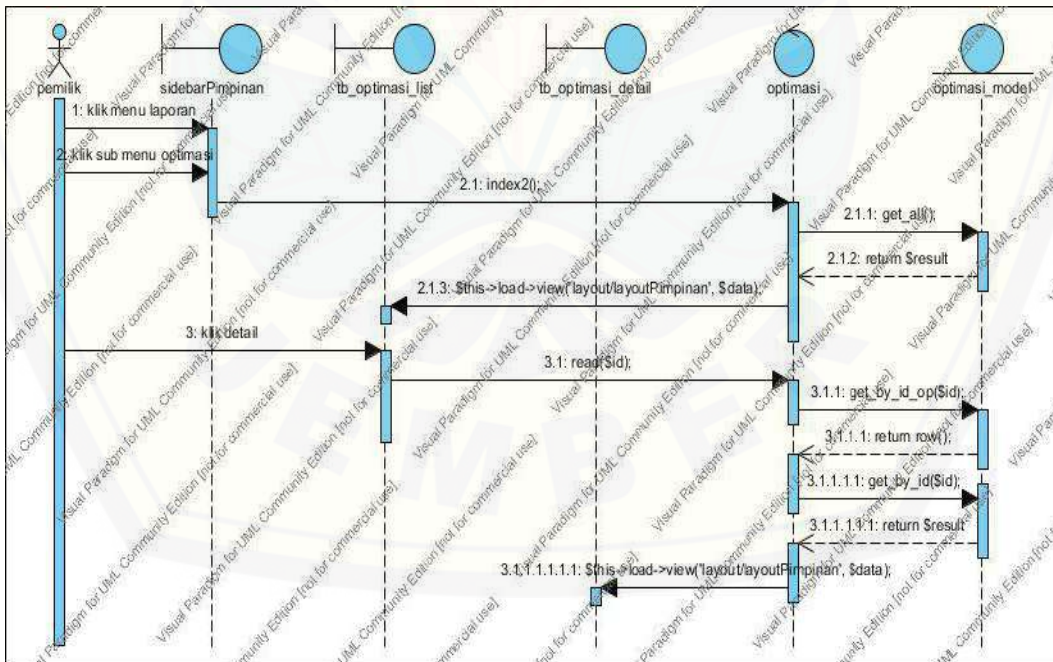
Gambar Sequence View Data Produksi

B. 9 Sequence View Data Penggunaan Bahan Baku



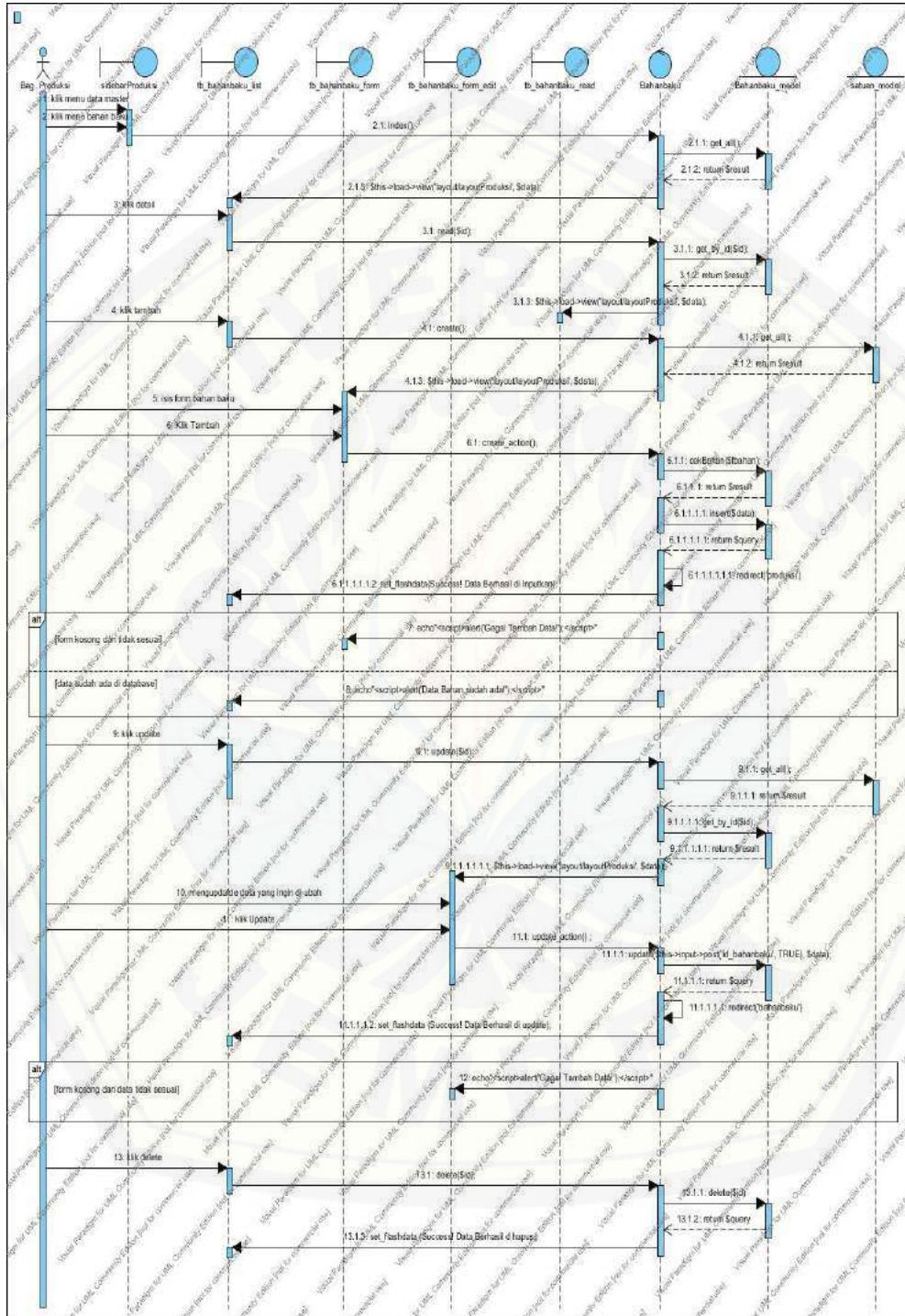
Gambar Sequence View Data Penggunaan Bahan Baku

B. 10 Sequence View Optimasi Produksi



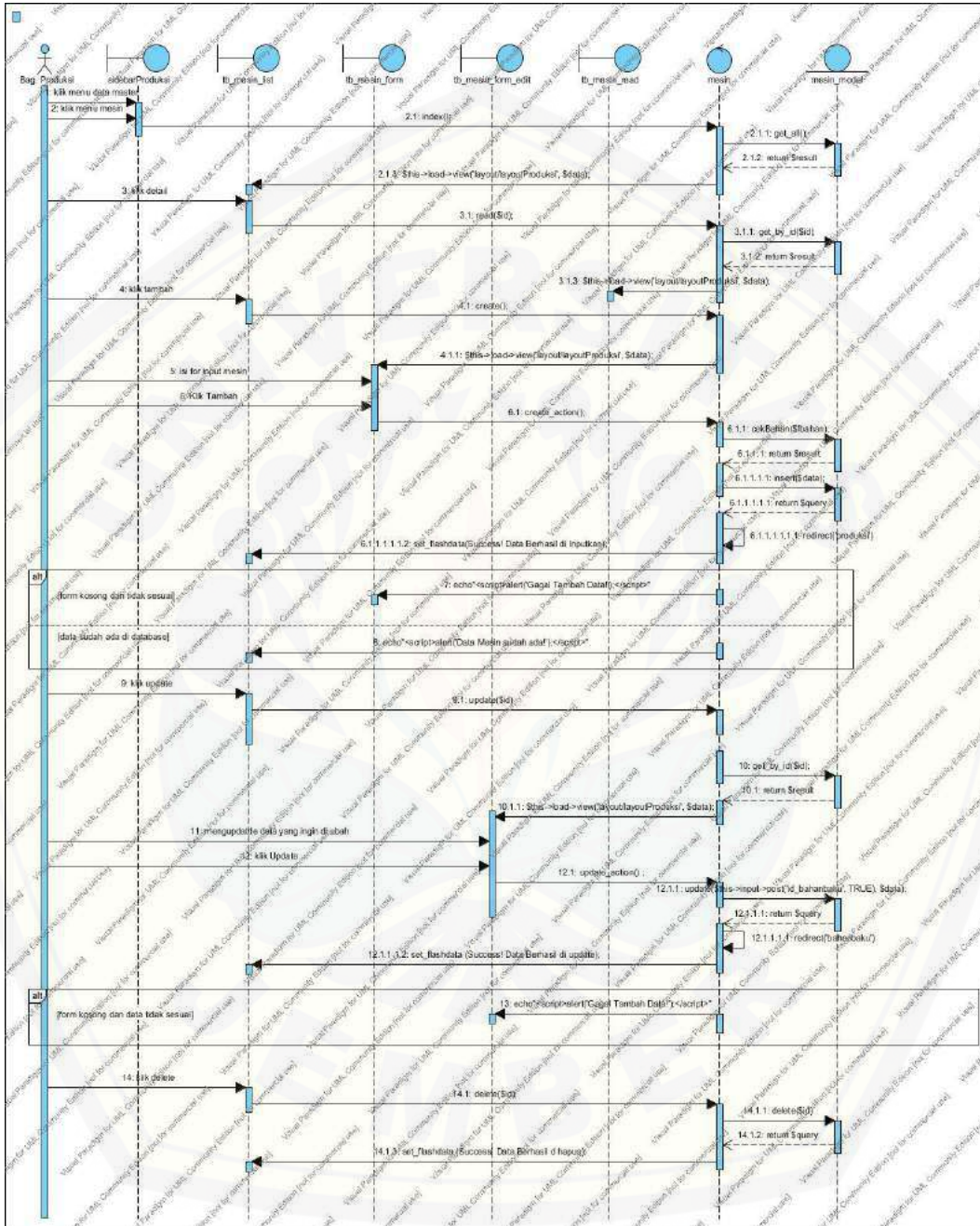
Gambar Sequence View Optimasi Produksi

B. 11 Sequence Mengelola Bahan Baku



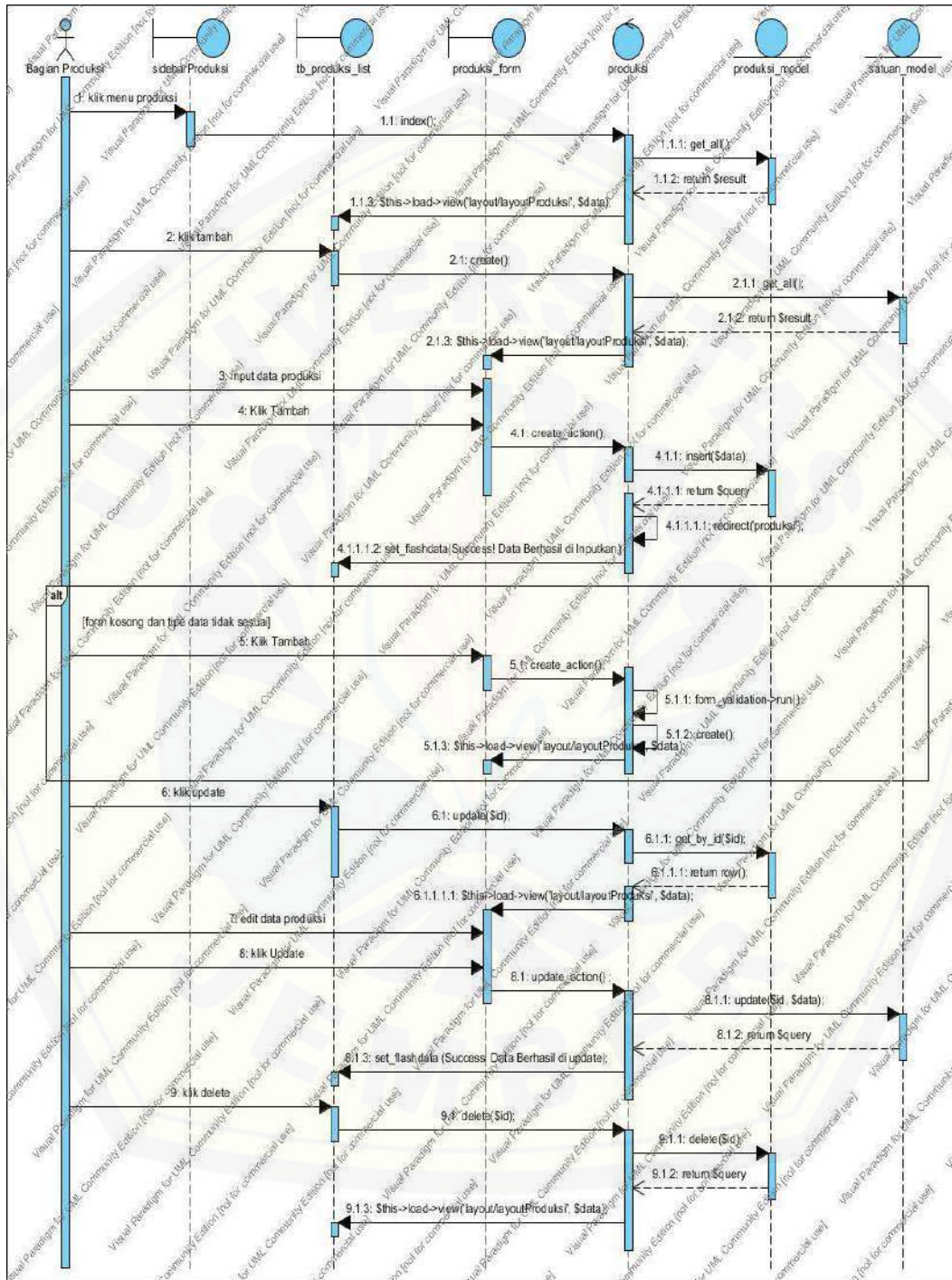
Gambar Sequence Mengelola Bahan Baku

B. 12 Sequence Mengelola Data Mesin



Gambar Sequence Mengelola Data Mesin

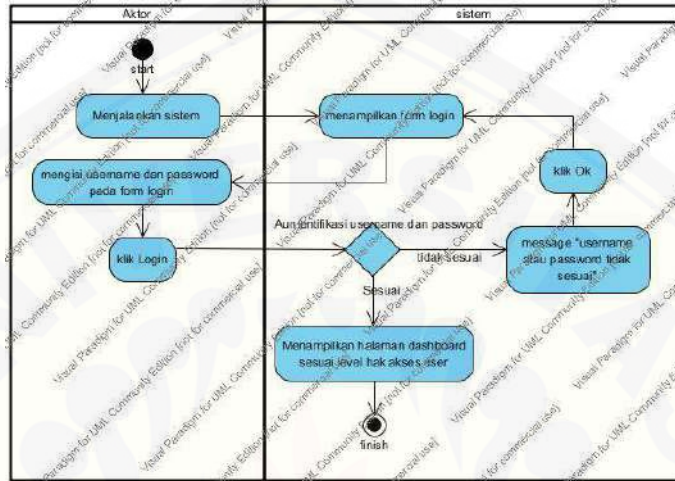
B. 13 Sequence Mengelola Data Produksi



Gambar Sequence Mengelola Data Produksi

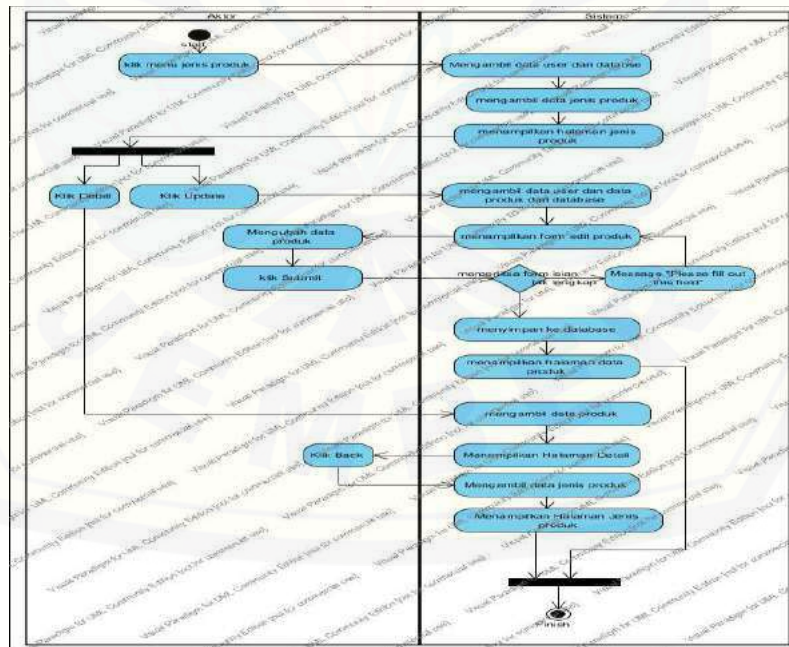
LAMPIRAN C. ACTIVITY DIAGRAM

C.1 Activity Login



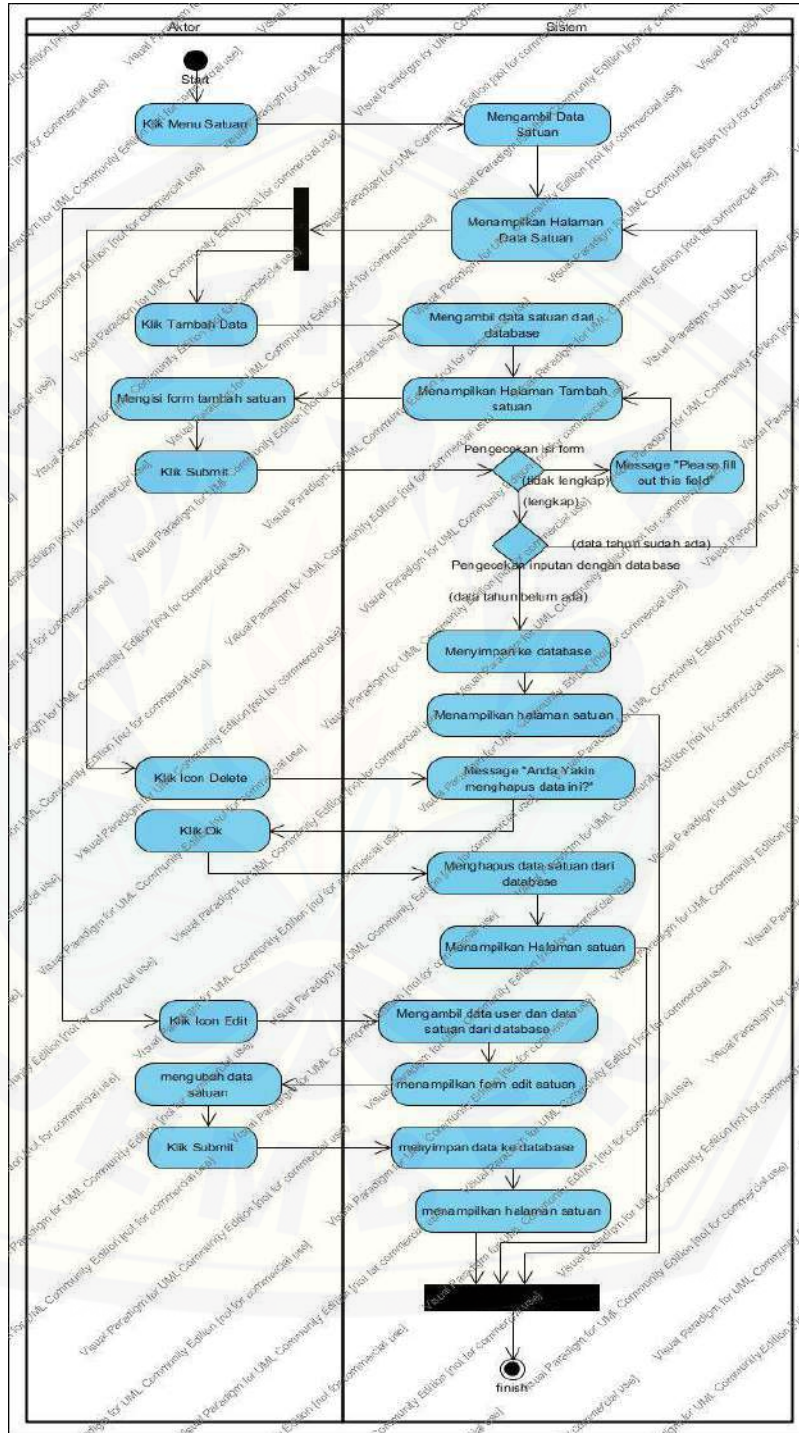
Gambar Activity Login

C.2 Activity Update Produk



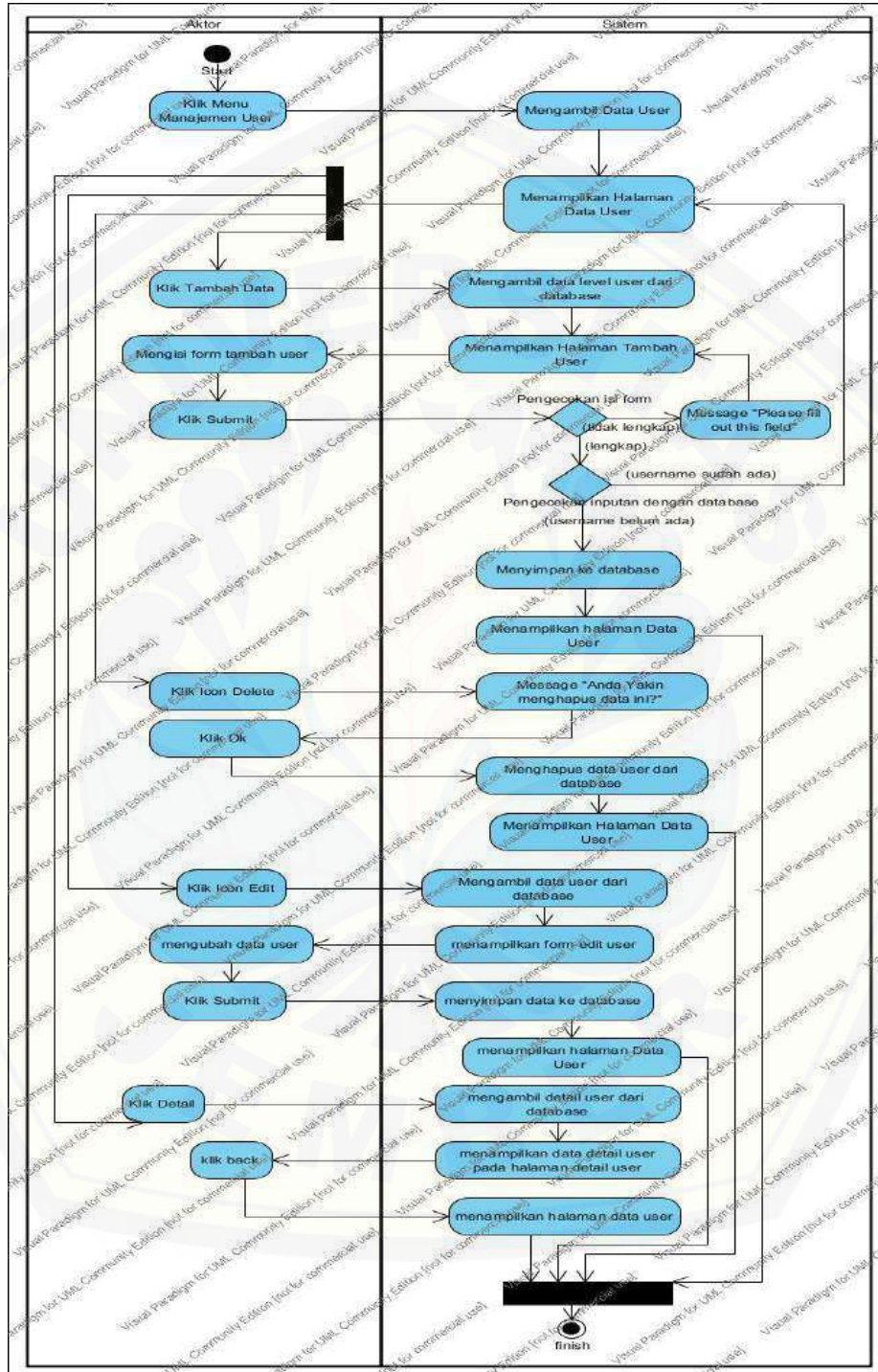
Gambar Activity Update Produk

A. 3 Activity Mengelola Data Satuan



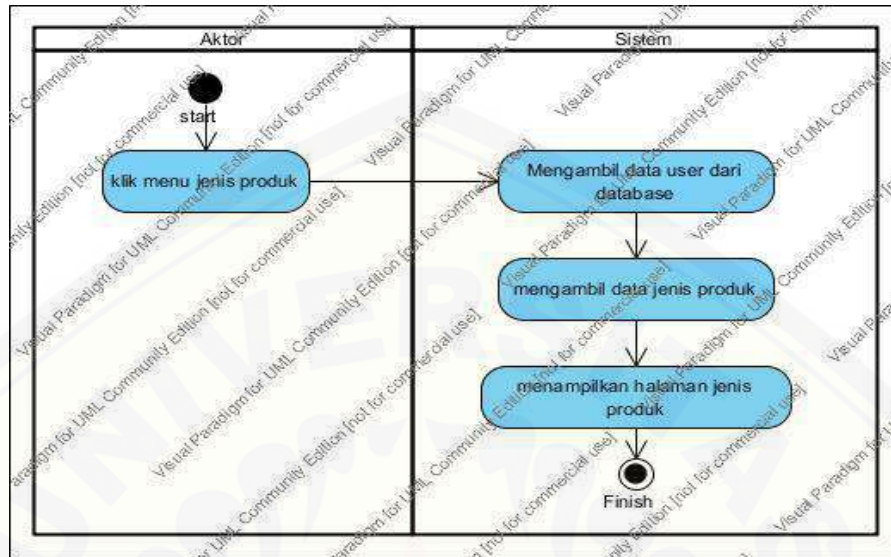
Gambar Activity Mengelola Data Satuan

C. 4 Activity Mengelola User



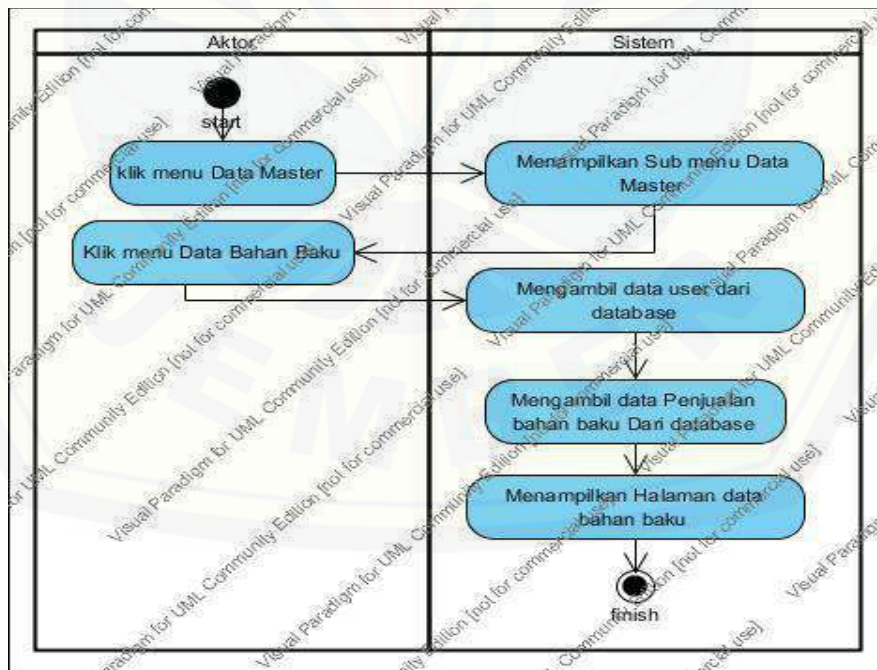
Gambar Activity Mengelola User

C. 5 Activity View Data Produk



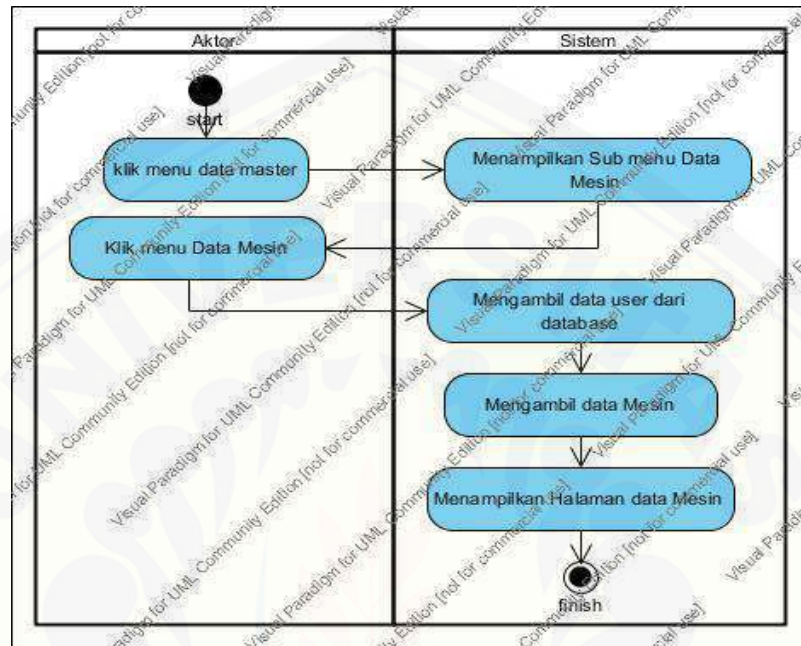
Gambar View Data Produk

C. 6 Activity View Bahan Baku



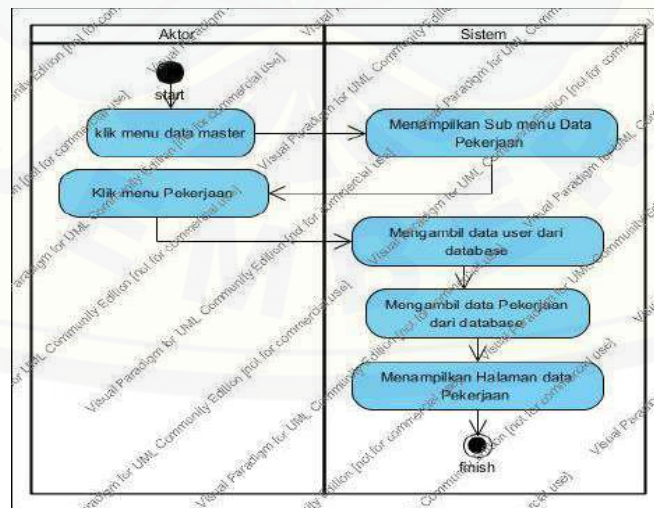
Gambar Activity View Bahan Baku

C. 7 Activity View Data Mesin



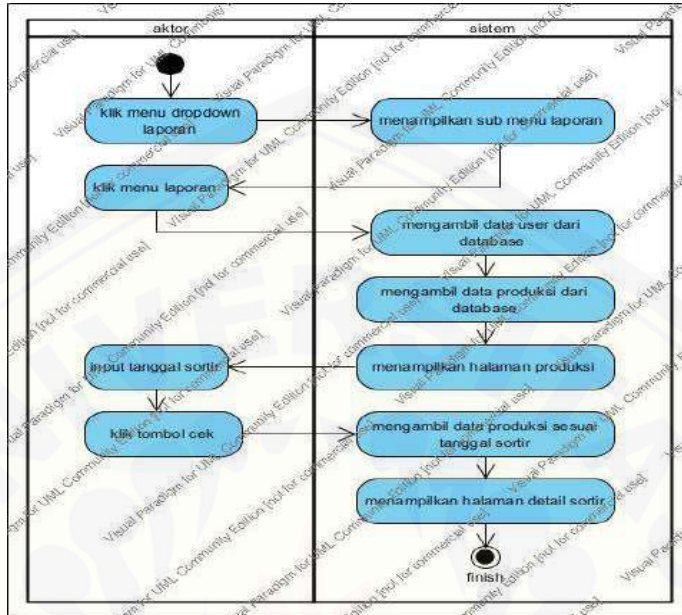
Gambar Activity View Data Mesin

C. 8 Activity View Penggunaan Jam Kerja Mesin



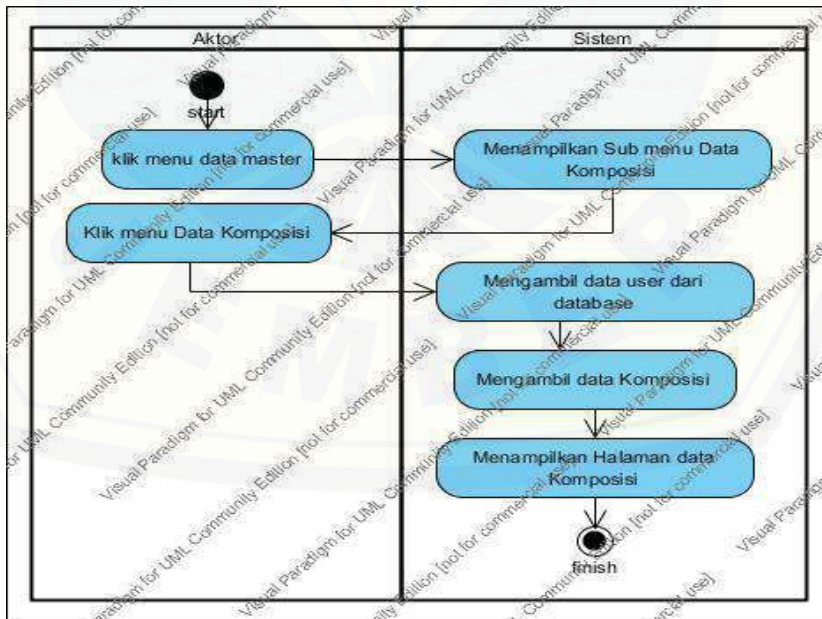
Gambar Activity View Penggunaan Jam Kerja Mesin

C. 9 Activity View Data Produksi



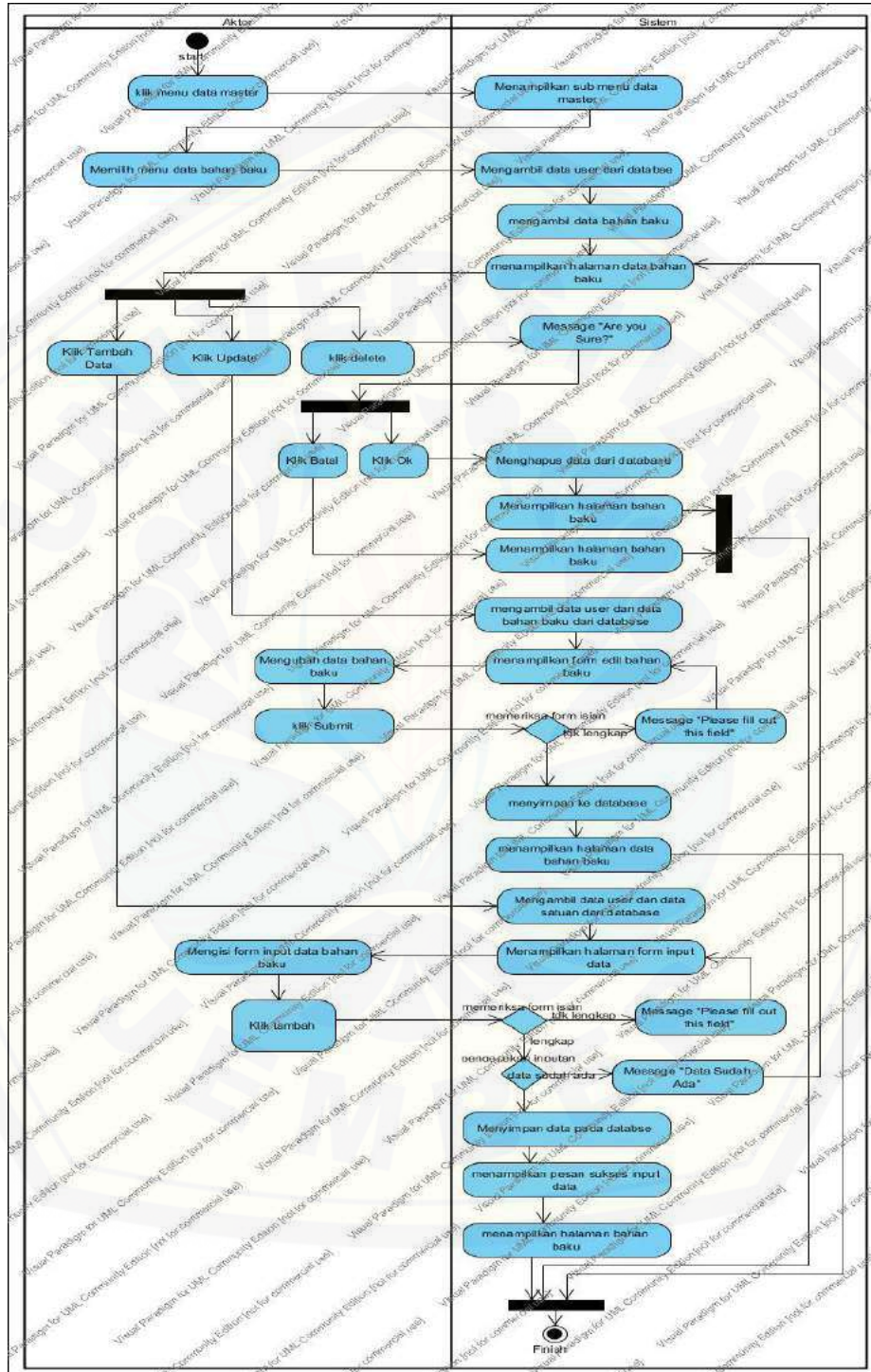
Gambar Activity View Data Produksi

C. 10 Activity View Data Penggunaan Bahan Baku



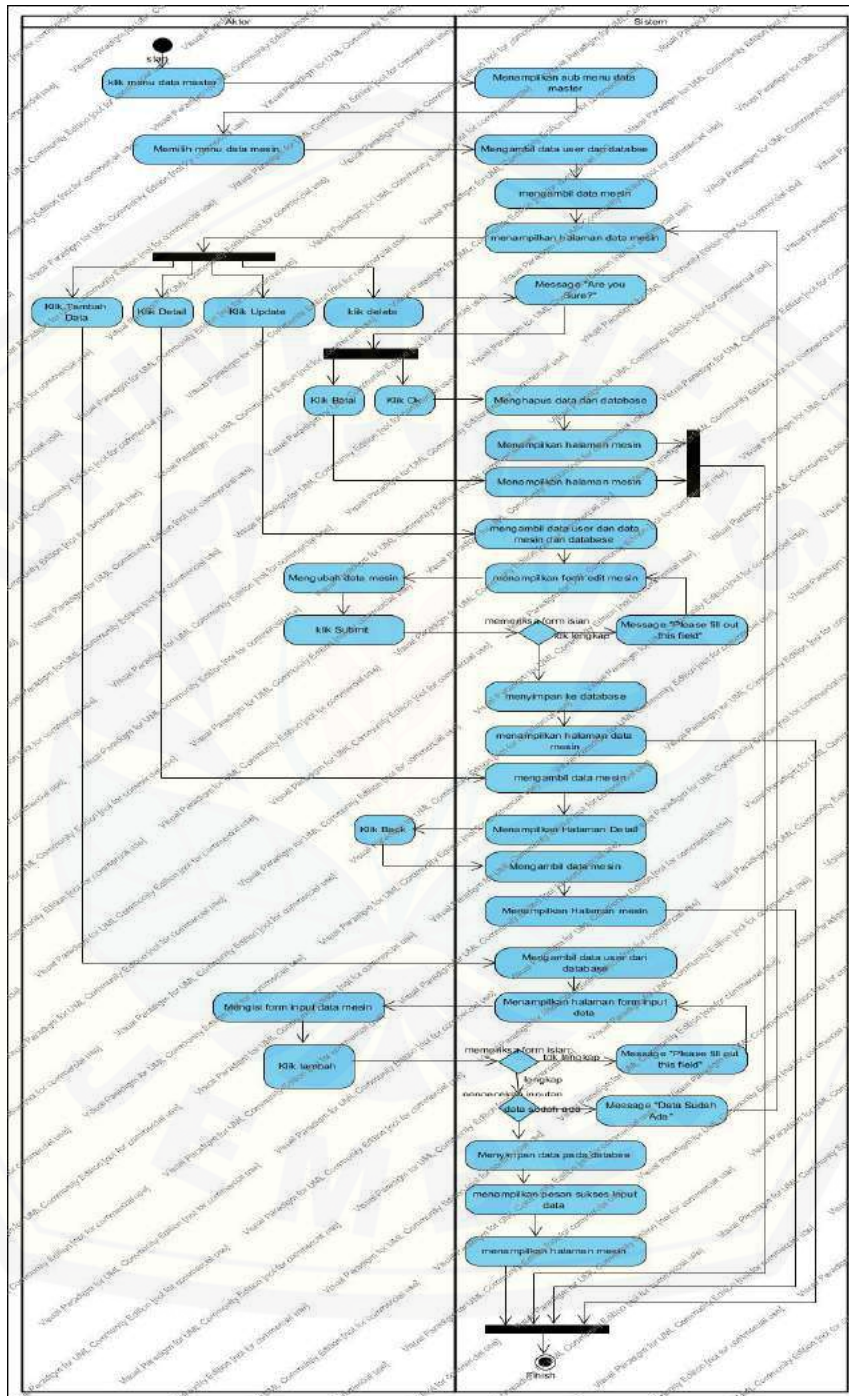
Gambar Activity View Data Penggunaan Bahan Baku

C. 11 Activity Mengelola Bahan Baku



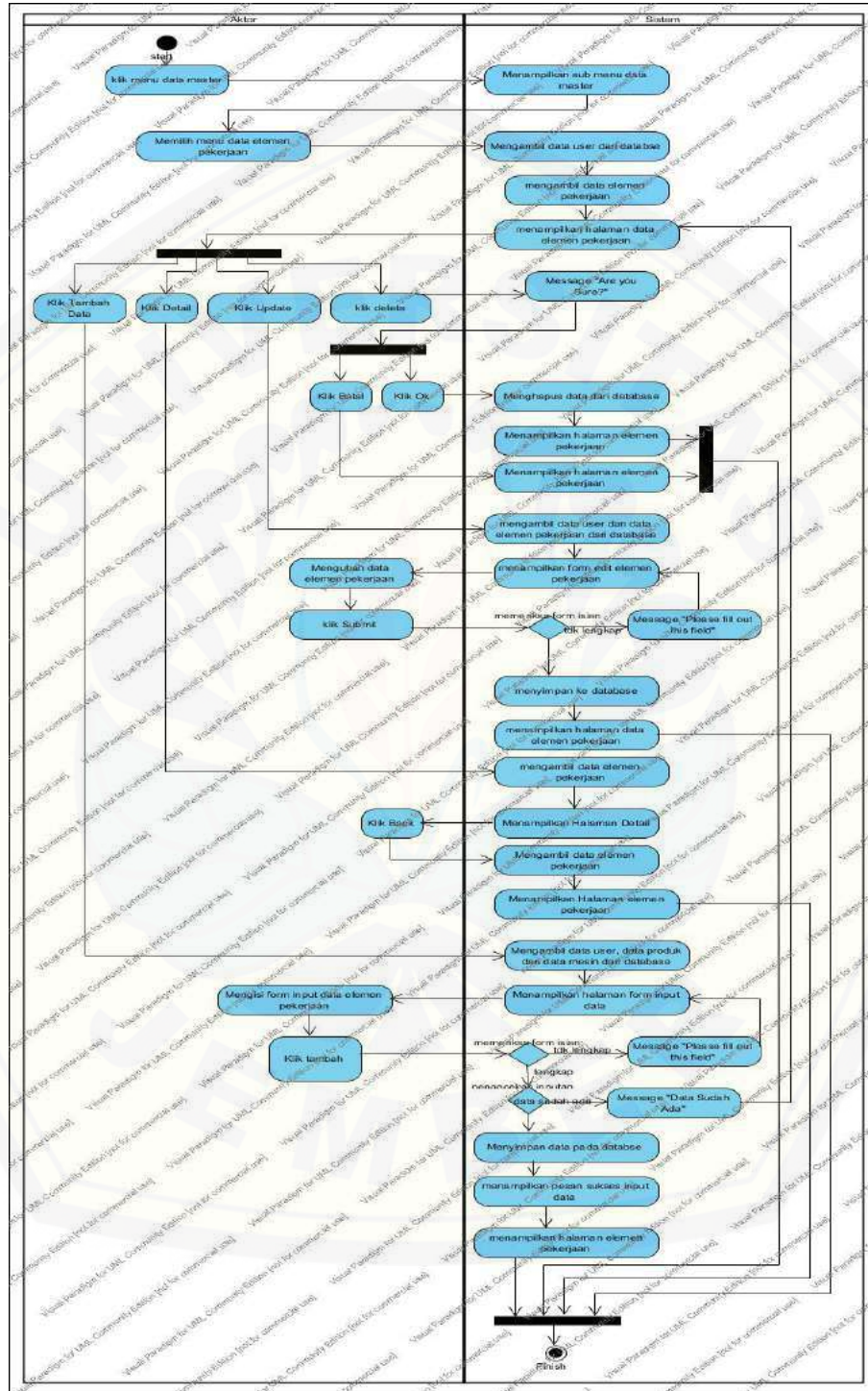
Gambar Activity Mengelola Bahan Baku

C. 12 Activity Mengelola Data Mesin



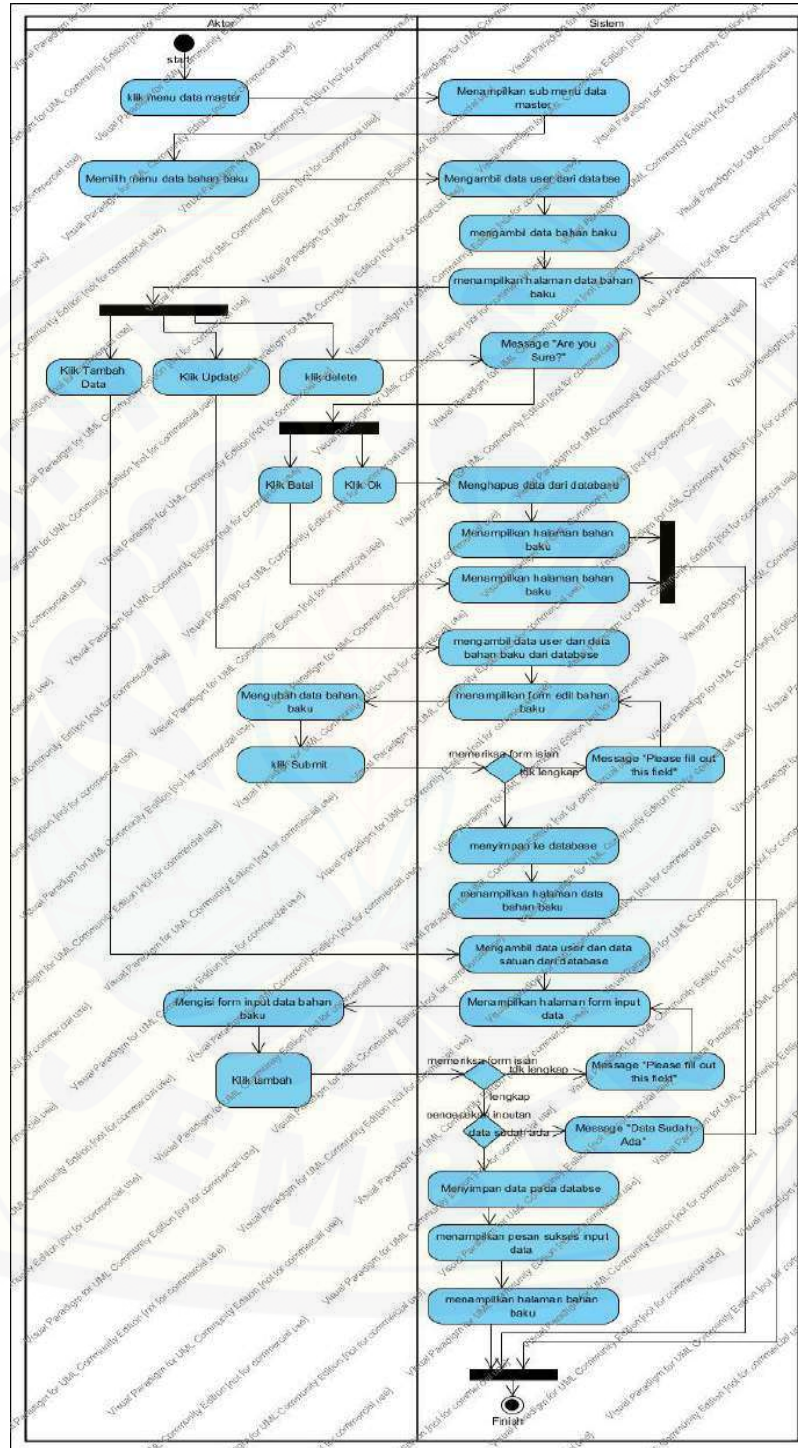
Gambar Activity Mengelola Data Mesin

C. 13 Activity Mengelola Penggunaan Jam Kerja Mesin



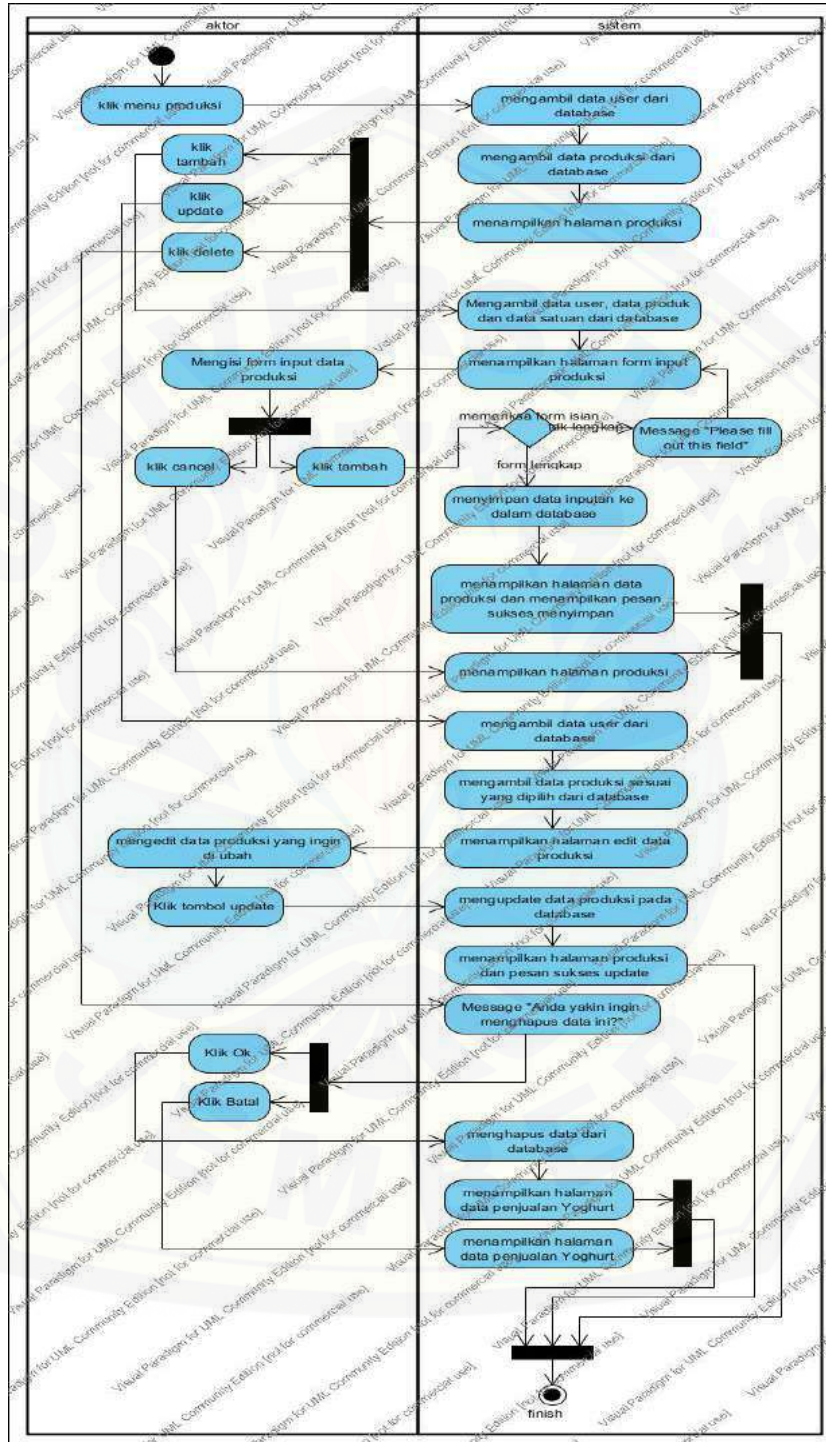
Gambar Activity Mengelola Penggunaan Jam Kerja Mesin

C. 14 Activity Mengelola Data Penggunaan Bahan Baku



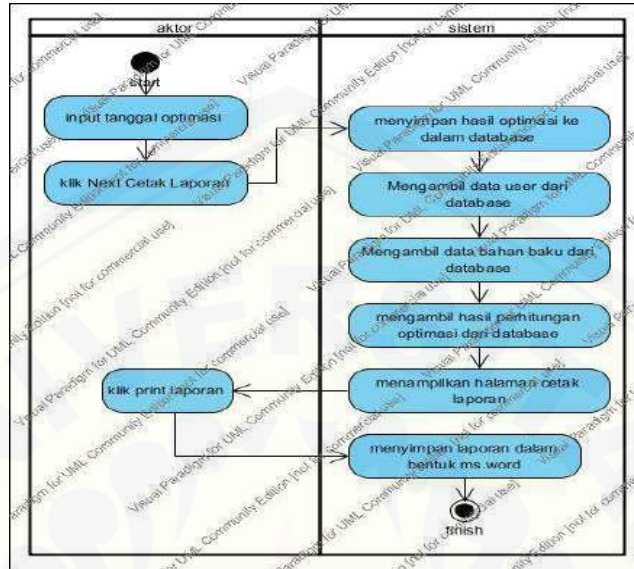
Gambar Activity Mengelola Data Penggunaan Bahan Baku

C. 15 Activity Mengelola Data Produksi



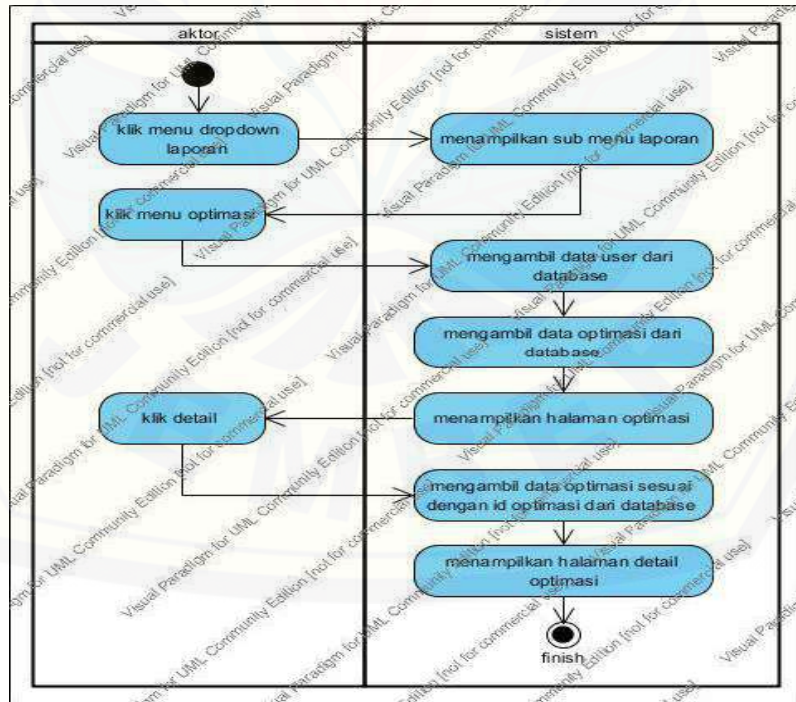
Gambar Activity Mengelola Data Produksi

C. 16 Activity Cetak Laporan Perhitungan optimasi produksi

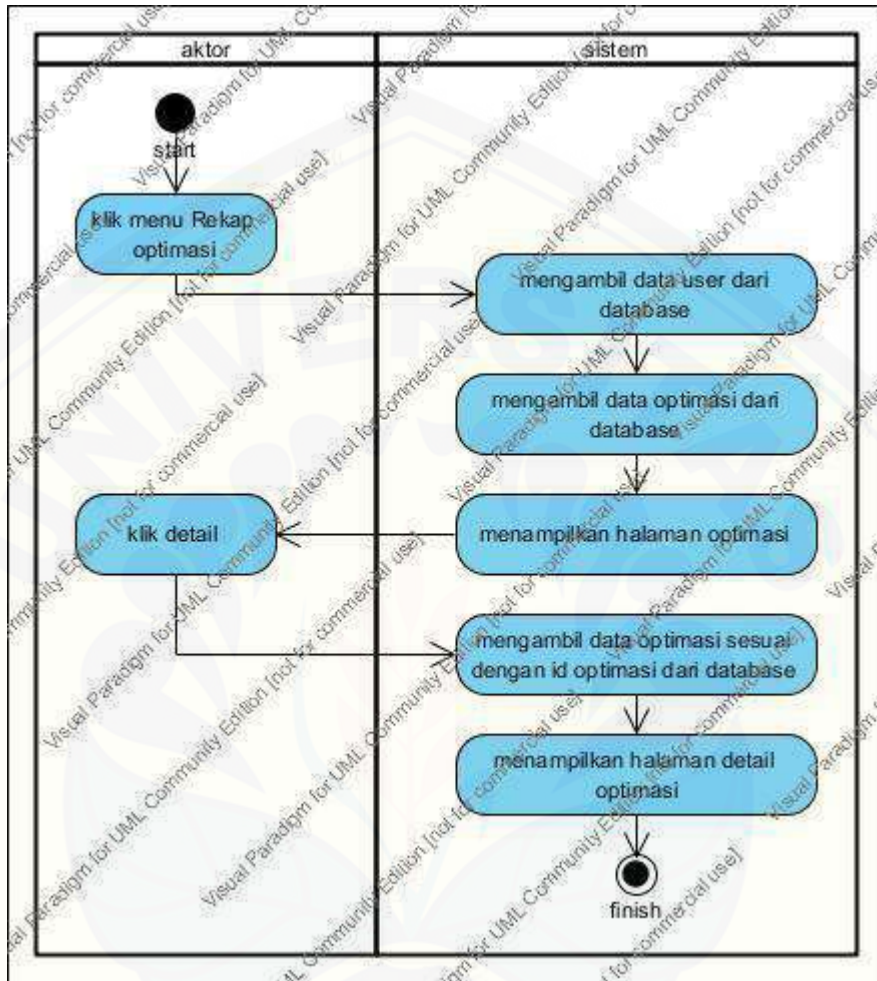


Gambar Activity Cetak Laporan Perhitungan optimasi produksi

C. 17 Activity View Optimasi Produksi - Pemilik



Gambar Activity View Optimasi Produksi - Pemilik

C. 18 Activity View Optimasi Produksi – Bagian Produksi

Gambar Activity View Optimasi Produksi – Bagian Produksi

LAMPIRAN D. KODE PROGRAMD.1 Kode program class *controller/Bahanbaku*

```
<?php

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

class Bahanbaku extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
        $this->load->model('Bahanbaku_model');
        $this->load->model('Satuan_model');
        $this->load->model('m_admin', 'madmin');
        if (!$this->session->userdata('username')) {
            redirect('C_Login');
        } else {

        }
    }
}

public function index()
{
    $bahanbaku = $this->Bahanbaku_model->get_all();
    $user = $this->session->userdata('username');
    $this->breadcrumbs->push('Bahanbaku', '/bahanbaku');
    $scoba = $this->madmin->detail_user($user)->result();

    $data = array(
        'title' => 'Bahanbaku',
        'content' => 'bahanbaku/tb_bahanbaku_list',
        'breadcrumbs' => $this->breadcrumbs->show(),
        'user' => $user,
        'detail' => $scoba,

        'bahanbaku_data' => $bahanbaku
    );

    $this->load->view('layout/layoutProduksi', $data);
}

public function index2()
{
    $bahanbaku = $this->Bahanbaku_model->get_all();
    $user = $this->session->userdata('username');
```

```
$this->breadcrumbs->push('Bahanbaku', '/bahanbaku');
$scoba = $this->madmin->detail_user($user)->result();

$data = array(
    'title' => 'Bahanbaku',
    'content' => 'bahanbaku/tb_bahanbaku_list_pimpinan',
    'breadcrumbs' => $this->breadcrumbs->show(),
    'user' => $user,
    'detail' => $scoba,

    'bahanbaku_data' => $bahanbaku
);

$this->load->view('layout/layoutPimpinan', $data);
}

public function read($id)
{
    $user = $this->session->userdata('username');
    $scoba = $this->madmin->detail_user($user)->result();
    $this->breadcrumbs->push('Bahanbaku', '/bahanbaku');
    $this->breadcrumbs->push('detail', '/bahanbaku/read');
    $row = $this->Bahanbaku_model->get_by_id($id);
    if ($row) {
        $data = array(
            'title' => 'Bahanbaku',
            'content' => 'bahanbaku/tb_bahanbaku_read',
            'breadcrumbs' => $this->breadcrumbs->show(),
            'user' => $user,
            'detail' => $scoba,

            'id_bahanbaku' => $row->id_bahanbaku,
            'nama_bahan' => $row->nama_bahan,
            'persediaan' => $row->persediaan,
            'nama_satuan' => $row->nama_satuan,
        );
        $this->load->view('layout/layoutProduksi', $data);
    } else {
        $this->session->set_flashdata('message', 'Record Not Found');
        redirect(site_url('bahanbaku'));
    }
}

public function create()
{
    $user = $this->session->userdata('username');
```

```
$scoba = $this->madmin->detail_user($user)->result();
$satuan = $this->Satuan_model->get_all();
$this->breadcrumbs->push('Bahanbaku', '/bahanbaku');
$this->breadcrumbs->push('tambah', '/bahanbaku/create');
$data = array(
    'title' => 'Bahanbaku',
    'content' => 'bahanbaku/tb_bahanbaku_form',
    'breadcrumbs' => $this->breadcrumbs->show(),
    'user' => $user,
    'detail' => $scoba,
    'satuan' => $satuan,

    'button' => 'Tambah',
    'action' => site_url('bahanbaku/create_action'),
    'id_bahanbaku' => set_value('id_bahanbaku'),
    'nama_bahan' => set_value('nama_bahan'),
    'persediaan' => set_value('persediaan'),
);
$this->load->view('layout/layoutProduksi', $data);
}

public function create_action()
{
    $this->_rules();
    $fbahan=$this->input->post('nama_bahan');
    $hasil=$this->Bahanbaku_model->cekBahan($fbahan);

    if ($hasil) {
        $this->create();
        echo"<script>alert('Data Bahan sudah ada!');</script>";
    } else if($this->form_validation->run() == FALSE) {
        $this->create();
        echo"<script>alert('Gagal Tambah Data!');</script>";
    }
    else {
        $data = array(
            'id_satuan' => $this->input->post('id_satuan', TRUE),
            'nama_bahan' => $this->input->post('nama_bahan',TRUE),
            'persediaan' => $this->input->post('persediaan',TRUE),
        );

        $this->Bahanbaku_model->insert($data);
        $this->session->set_flashdata('message', '<div class="alert alert-success
no-border">
        <button type="button" class="close" data-
dismiss="alert"><span>x</span><span class="sr-
```

```
only">Close</span></button>
    <span class="text-semibold">Success!</span> Data Berhasil di
Inputkan .
    </div>');
    redirect(site_url('bahanbaku'));
}
}

public function update($id)
{
    $user = $this->session->userdata('username');
    $this->breadcrumbs->push('Bahanbaku', '/bahanbaku');
    $satuan = $this->Satuan_model->get_all();
    $this->breadcrumbs->push('update', '/bahanbaku/update');
    $scoba = $this->madmin->detail_user($user)->result();

    $row = $this->Bahanbaku_model->get_by_id($id);
    if ($row) {
        $data = array(
            'title' => 'Bahanbaku',
            'content' => 'bahanbaku/tb_bahanbaku_form_edit',
            'breadcrumbs' => $this->breadcrumbs->show(),
            'user' => $user,
            'detail' => $scoba,
            'satuan' => $satuan,

            'button' => 'Update',
            'action' => site_url('bahanbaku/update_action'),
            'id_bahanbaku' => set_value('id_bahanbaku', $row->id_bahanbaku),
            'nama_bahan' => set_value('nama_bahan', $row->nama_bahan),
            'persediaan' => set_value('persediaan', $row->persediaan),
        );
        $this->load->view('layout/layoutProduksi', $data);
    } else {
        $this->session->set_flashdata('message', 'Record Not Found');
        redirect(site_url('bahanbaku'));
    }
}

public function update_action()
{
    $this->_rules();

    if ($this->form_validation->run() == FALSE) {
        $this->update($this->input->post('id_bahanbaku', TRUE));
    } else {
```

```
$data = array(
    'nama_bahan' => $this->input->post('nama_bahan',TRUE),
    'persediaan' => $this->input->post('persediaan',TRUE),
    'id_satuan' => $this->input->post('id_satuan', TRUE),
);

$this->Bahanbaku_model->update($this->input->post('id_bahanbaku',
TRUE), $data);
$this->session->set_flashdata('message', ' <div class="alert alert-info
no-border">
    <button type="button" class="close" data-
dismiss="alert"><span>x</span></button><span class="sr-
only">Close</span></div>
    <span class="text-semibold">Success!</span> Data Berhasil di
Update .
    </div>');
    redirect(site_url('bahanbaku'));
}
}

public function delete($id)
{
    $row = $this->Bahanbaku_model->get_by_id($id);

    if ($row) {
        $this->Bahanbaku_model->delete($id);
        $this->session->set_flashdata('message', '<div class="alert bg-danger">
            <button type="button" class="close" data-
dismiss="alert"><span>x</span></button><span class="sr-
only">Close</span></div>
            <span class="text-semibold">Success!</span> Data Berhasil dihapus
            .
            </div>');
        redirect(site_url('bahanbaku'));
    } else {
        $this->session->set_flashdata('message', 'Record Not Found');
        redirect(site_url('bahanbaku'));
    }
}

public function _rules()
{
    $this->form_validation->set_rules('nama_bahan', 'nama bahan',
'trim|required');
    $this->form_validation->set_rules('persediaan', 'persediaan',
'trim|required|numeric');
```



```

        $this->form_validation->set_rules('id_satuan', 'satuan', 'trim|required');

        $this->form_validation->set_rules('id_bahanbaku', 'id_bahanbaku', 'trim');
        $this->form_validation->set_error_delimiters('<span class="text-
danger">', '</span>');
    }
}

```

D.2 Kode program class *controller/C_Login*

```

<?php

class C_Login extends CI_Controller {

    function __construct() {
        parent::__construct();
        $this->load->helper('url');
        $this->load->model('m_admin', 'madmin');
        $this->load->model('m_login');
    }

    public function index() {

        $this->load->view('auth/login');
    }

    public function login() {
        $user = $this->session->userdata('username');
        $data['detail'] = $this->madmin->detail_user($user)->result();
        $username = $this->input->post('username');
        $password = $this->input->post('password');
        $status = $this->m_login->Auth($username, $password);
        if ($status) {
            //jika username & password match
            $data = $this->m_login->getDataLogin($username)->result();
            $sesi = array(
                'username' => $username,
                'level' => $data[0]->level,
                'lastLogin' => $data[0]->lastLogin
            );
            $this->session->set_userdata($sesi);
            $this->db->query("UPDATE user set lastLogin = NOW() where
username='".$username."'");

            if ($this->session->userdata('level') == 1) {

```

```

        //load view pegawai
        echo "<script>alert('Selamat Datang, " . $data[0]->namaUser .
        "');</script>";
        redirect('Dashboard/index4');
    } else if ($this->session->userdata('level') == 2) {
        //load view admin
        echo "<script>alert('Selamat Datang, " . $data[0]->namaUser .
        "');</script>";
        redirect('Dashboard/index2');
    } else if ($this->session->userdata('level') == 3){
        //load view petani
        echo"<script>alert('Selamat Datang, " . $data[0]-
        >namaUser . "');</script>";
        redirect('Dashboard/index3');
    }else if ($this->session->userdata('level') == 4){
        //load view petani
        echo"<script>alert('Selamat Datang, " . $data[0]->namaUser
        . "');</script>";
        redirect('Dashboard');
    } else {
    }
    }else{
        echo "<script>alert('Username atau Password Tidak
        Sesuai');</script>";
        $this->load->view('auth/login');
    }
    }

    public function logout() {
        $this->session->unset_userdata('username');
        $this->session->unset_userdata('level');
        $this->session->sess_destroy();
        redirect('C_Login');
    }
}
?>

```

D.3 Kode program class *controller/C_UserSimpleks*

```

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

class C_UserSimpleks extends CI_Controller
{

```

```
function __construct()
{
    parent::__construct();
    $this->load->model('M_Admin', 'madmin');
    if (!$this->session->userdata('username')) {
        redirect('C_Login');
    } else {
        $this->load->model('M_UserSimpleks');
    }
}

public function index()
{
    $user = $this->session->userdata('username');
    $scoba = $this->madmin->detail_user($user)->result();
    $ambil = $this->M_UserSimpleks->get_all();
    $this->breadcrumbs->push('Users', '/users');

    $data = array(
        'title' => 'Users',
        'content' => 'users/users_list',
        'breadcrumbs' => $this->breadcrumbs->show(),
        'user' => $user,
        'detail' => $scoba,
        'uName' => $ambil,

        // 'users_data' => $users
    );

    $this->load->view('layout/layout', $data);
}

public function read($id)
{
    $user = $this->session->userdata('username');
    $scoba = $this->madmin->detail_user($user)->result();
    $this->breadcrumbs->push('Users', '/C_UserSimpleks');
    $this->breadcrumbs->push('detail', '/C_UserSimpleks/read');
    $row = $this->M_UserSimpleks->get_by_id($id);
    if ($row) {
        $data = array(
            'title' => 'Users',
            'content' => 'users/users_read',
            'breadcrumbs' => $this->breadcrumbs->show(),
            'user' => $user,
            'detail' => $scoba,
```

```
        'id_user' => $row->id_user,
        'namaUser' => $row->namaUser,
        'Email' => $row->Email,
        'username' => $row->username,
        'password' => $row->password,
        'Phone' => $row->Phone,
        'Alamat' => $row->Alamat,
        'namaLevel' => $row->namaLevel,
        'lastLogin' => $row->lastLogin,
    );
    $this->load->view('layout/layout', $data);
} else {
    $this->session->set_flashdata('message', 'Record Not Found');
    redirect(site_url('users'));
}
}

public function create()
{
    $level = $this->M_UserSimpleks->getlevel();
    $user = $this->session->userdata('username');
    $scoba = $this->madmin->detail_user($user)->result();
    $this->breadcrumbs->push('Users', '/C_UserSimpleks');
    $this->breadcrumbs->push('tambah', '/C_UserSimpleks/create');
    $data = array(
        'title' => 'Users',
        'content' => 'users/users_form',
        'breadcrumbs' => $this->breadcrumbs->show(),
        'user' => $user,
        'id_user' => set_value('id_user'),
        'namaUser' => set_value('namaUser'),
        'Email' => set_value('Email'),
        'username' => set_value('username'),
        'password' => set_value('password'),
        'Phone' => set_value('Phone'),
        'Alamat' => set_value('Alamat'),
        'level' => $level,
        'levell' => set_value('levell'),
        'detail' => $scoba,

        'button' => 'Tambah',
        'action' => site_url('C_UserSimpleks/create_action'),
    );
    $this->load->view('layout/layout', $data);
}
```

```

public function create_action()
{
    $this->form_validation->set_rules('namaUser', 'namaUser', 'trim|required');
    $this->form_validation->set_rules('username', 'username', 'trim|required');
    $this->form_validation->set_rules('Email', 'Email', 'trim|required');
    $this->form_validation->set_rules('Alamat', 'Alamat', 'trim|required');
    $this->form_validation->set_rules('Phone', 'Phone', 'trim|required|numeric');
    $this->form_validation->set_rules('level', 'level', 'trim|required');
    $this->form_validation->set_rules('password', 'password', 'trim|required');
    $this->form_validation->set_rules('password2', 'Konfirmasi Password',
'trim|matches[password]');
    $this->form_validation->set_rules('id_user', 'id_user', 'trim');
    $this->form_validation->set_error_delimiters('<span class="text-danger">',
'</span>');

    if ($this->form_validation->run() == FALSE) {
        $this->create();
    } else {
        $data = array(
            'namaUser' => $this->input->post('namaUser',TRUE),
            'username' => $this->input->post('username',TRUE),
            'Email' => $this->input->post('Email',TRUE),
            'Alamat' => $this->input->post('Alamat',TRUE),
            'Phone' => $this->input->post('Phone',TRUE),
            'password' => $this->input->post('password',TRUE),
            'level' => $this->input->post('level',TRUE),
        );

        $this->M_UserSimpleks->insert($data);
        $this->session->set_flashdata('message', '<div class="alert alert-success
no-border">
        <button type="button" class="close" data-
dismiss="alert"><span>x</span></button><span class="sr-only">Close</span></div>
        <span class="text-semibold">Success!</span> Data Berhasil di
Inputkan .
        </div>');
        redirect(site_url('C_UserSimpleks'));
    }
}

public function update($id)
{
    $level = $this->M_UserSimpleks->getlevel();
    $user = $this->session->userdata('username');
    $coba = $this->madmin->detail_user($user)->result();
}

```

```
$this->breadcrumbs->push('Users', '/C_UserSimpleks');
$this->breadcrumbs->push('update', '/users/update');

$row = $this->M_UserSimpleks->get_by_id($id);
if ($row) {
    $data = array(
        'title' => 'Users',
        'content' => 'users/users_form',
        'breadcrumbs' => $this->breadcrumbs->show(),
        'user' => $user,
        'detail' => $coba,
        'level' => $level,

        'button' => 'Update',
        'action' => site_url('C_UserSimpleks/update_action'),
        'id_user' => set_value('id_user', $row->id_user),
        'namaUser' => set_value('namaUser', $row->namaUser),
        'Email' => set_value('Email', $row->Email),
        'username' => set_value('username', $row->username),
        'password' => set_value('password', $row->password),
        'Phone' => set_value('Phone', $row->Phone),
        'Alamat' => set_value('Alamat', $row->Alamat),
        'lastLogin' => set_value('lastLogin', $row->lastLogin)
    );
    $this->load->view('layout/layout', $data);
} else {
    $this->session->set_flashdata('message', 'Record Not Found');
    redirect(site_url('C_UserSimpleks'));
}
}

public function update_action() {
    $this->form_validation->set_rules('namaUser', 'namaUser', 'trim|required');
    $this->form_validation->set_rules('username', 'username', 'trim|required');
    $this->form_validation->set_rules('Email', 'Email', 'trim|required');
    $this->form_validation->set_rules('Alamat', 'Alamat', 'trim|required');
    $this->form_validation->set_rules('Phone', 'Phone', 'trim|required|numeric');
    $this->form_validation->set_rules('level', 'level', 'trim|required');
    $this->form_validation->set_rules('password', 'password', 'trim|required');
    $this->form_validation->set_rules('lastLogin', 'lastLogin', 'trim|required');
    $this->form_validation->set_rules('id_user', 'id_user', 'trim');
    $this->form_validation->set_error_delimiters('<span class="text-danger">',
'</span>');

    if ($this->form_validation->run() == FALSE) {
        $this->update($this->input->post('id_user', TRUE));
    }
}
```

```
    } else {
        $data = array(
            'namaUser' => $this->input->post('namaUser',TRUE),
            'username' => $this->input->post('username',TRUE),
            'Email' => $this->input->post('Email',TRUE),
            'Alamat' => $this->input->post('Alamat',TRUE),
            'Phone' => $this->input->post('Phone',TRUE),
            'password' => $this->input->post('password',TRUE),
            'level' => $this->input->post('level',TRUE),
            'lastLogin' => $this->input->post('lastLogin',TRUE),
        );
        print_r($data);
        // $this->M_UserSimpleks->update($this->input->post('id_user', TRUE),
        $data);
        // $this->session->set_flashdata('message',
        //     '<div class="alert alert-info no-border">
        //     <button type="button" class="close" data-
        dismiss="alert"><span>x</span><span class="sr-only">Close</span></button>
        //     <span class="text-semibold">Success!</span> Data Berhasil di
        Update .
        //     </div>');
        // redirect(site_url('C_UserSimpleks'));
    }
}

public function delete($id)
{
    $row = $this->M_UserSimpleks->get_by_id($id);

    if ($row) {
        $this->M_UserSimpleks->delete($id);
        $this->session->set_flashdata('message', 'Delete Record Success');
        redirect(site_url('C_UserSimpleks'));
    } else {
        $this->session->set_flashdata('message', 'Record Not Found');
        redirect(site_url('C_UserSimpleks'));
    }
}

public function ubah_password($id)
{
    $password = $this->input->post('password',TRUE);
    $this->form_validation->set_rules('password', 'password', 'trim|required');
    if ($this->form_validation->run() == FALSE) {
        $this->update($id);
    } else {
```

```
$data = array(
    'password' => $password,
);
$this->M_UserSimpleks->update($id,$data);
$this->session->set_flashdata('message', 'Password Berhasil dirubah');
    redirect(site_url('C_UserSimpleks/update/'. $id));
}
}
}
```

D.4 Kode program class *controller*/Komposisi

```
<?php

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

class Komposisi extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
        $this->load->model('Komposisi_model');
        $this->load->model('Bahanbaku_model');
        $this->load->model('Produk_model');
        $this->load->model('m_admin', 'madmin');
        if (!$this->session->userdata('username')) {
            redirect('C_Login');
        } else {

        }
    }

    public function index()
    {
        $komposisi = $this->Komposisi_model->get_all();
        $user = $this->session->userdata('username');
        $this->breadcrumbs->push('Komposisi', '/komposisi');
        $scoba = $this->madmin->detail_user($user)->result();

        $data = array(
            'title' => 'Komposisi',
            'content' => 'komposisi/tb_komposisi_list',
            'breadcrumbs' => $this->breadcrumbs->show(),
            'user' => $user,
            'detail' => $scoba,
```



```
        'komposisi_data' => $komposisi
    );

    $this->load->view('layout/layoutProduksi', $data);
}

public function index2()
{
    $komposisi = $this->Komposisi_model->get_all();
    $coba = $this->madmin->detail_user($user)->result();
    $user = $this->session->userdata('username');
    $this->breadcrumbs->push('Komposisi', '/komposisi');

    $data = array(
        'title' => 'Komposisi',
        'content' => 'komposisi/tb_komposisi_list_pimpinan',
        'breadcrumbs' => $this->breadcrumbs->show(),
        'user' => $user,
        'detail' => $coba,

        'komposisi_data' => $komposisi
    );

    $this->load->view('layout/layoutPimpinan', $data);
}

public function read($id)
{
    $user = $this->session->userdata('username');
    $coba = $this->madmin->detail_user($user)->result();
    $this->breadcrumbs->push('Komposisi', '/komposisi');
    $this->breadcrumbs->push('detail', '/komposisi/read');
    $row = $this->Komposisi_model->get_by_id($id);
    if ($row) {
        $data = array(
            'title' => 'Komposisi',
            'content' => 'komposisi/tb_komposisi_read',
            'breadcrumbs' => $this->breadcrumbs->show(),
            'user' => $user,
            'detail' => $coba,

            'id_komposisi' => $row->id_komposisi,
            'nama_bahan' => $row->nama_bahan,
            'nama_produk' => $row->nama_produk,
            'jumlah_pemakaian' => $row->jumlah_pemakaian,
```

```
);
$this->load->view('layout/layoutProduksi', $data);
} else {
    $this->session->set_flashdata('message', 'Record Not Found');
    redirect(site_url('komposisi'));
}
}

public function create()
{
    $user = $this->session->userdata('username');
    $coba = $this->admin->detail_user($user->result());
    $this->breadcrumbs->push('Komposisi', '/komposisi');
    $this->breadcrumbs->push('tambah', '/komposisi/create');
    $data = array(
        'title' => 'Komposisi',
        'content' => 'komposisi/tb_komposisi_form',
        'breadcrumbs' => $this->breadcrumbs->show(),
        'user' => $user,
        'detail' => $coba,

        'button' => 'Tambah',
        'action' => site_url('komposisi/create_action'),
        'id_komposisi' => set_value('id_komposisi'),
        'id_bahanbaku' => set_value('id_bahanbaku'),
        'id_produk' => set_value('id_produk'),
        'jumlah_pemakaian' => set_value('jumlah_pemakaian'),
    );
    $this->load->view('layout/layoutProduksi', $data);
}

public function create_action()
{
    $this->_rules();

    if ($this->form_validation->run() == FALSE) {
        $this->update($this->input->post('id_komposisi', TRUE));
    } else {
        $id_bahanbaku=$this->input->post('id_bahanbaku');
        $id_produk=$this->input->post('id_produk');
        $cek=$this->Komposisi_model->cekKomposisi2($id_bahanbaku,
        $id_produk);
        if ($cek==true) {
            $this->session->set_flashdata('message', '<div class="alert alert-danger
no-border">
            <button type="button" class="close" data-
```

```

dismiss="alert"><span>x</span><span class="sr-only">Close</span></button>
  <span class="text-semibold">Failed!</span> Data Komposisi sudah ada
.
  </div>');
  redirect(site_url('komposisi'));
}else {
$data = array(
  'id_bahanbaku' => $this->input->post('id_bahanbaku',TRUE),
  'id_produk' => $this->input->post('id_produk',TRUE),
  'jumlah_pemakaian' => $this->input->post('jumlah_pemakaian',TRUE),
);

$this->Komposisi_model->insert($data);
$this->session->set_flashdata('message', '<div class="alert alert-success
no-border">
  <button type="button" class="close" data-
dismiss="alert"><span>x</span><span class="sr-only">Close</span></button>
  <span class="text-semibold">Success!</span> Data Berhasil di
Inputkan .
  </div>');
  redirect(site_url('komposisi'));
}
}
}

public function update($id)
{
  $user = $this->session->userdata('username');
  $this->breadcrumbs->push('Komposisi', '/komposisi');
  $this->breadcrumbs->push('update', '/komposisi/update');
  $coba = $this->madmin->detail_user($user)->result();
  $bhn_bku = $this->Bahanbaku_model->get_all();
  $produk = $this->Produk_model->get_all();

  $row = $this->Komposisi_model->get_by_id($id);
  if ($row) {
    $data = array(
      'title' => 'Komposisi',
      'content' => 'komposisi/tb_komposisi_form_edit',
      'breadcrumbs' => $this->breadcrumbs->show(),
      'user' => $user,
      'detail' => $coba,
      'bhn_bku' => $bhn_bku,
      'produk' => $produk,

```

```

        'button' => 'Update',
        'action' => site_url('komposisi/update_action'),
        'id_komposisi' => set_value('id_komposisi', $row->id_komposisi),
        'id_bahanbaku' => set_value('id_bahanbaku', $row->id_bahanbaku),
        'nama_bahan' => set_value('nama_bahan', $row->nama_bahan),
        'nama_produk' => set_value('nama_produk', $row->nama_produk),
        'id_produk' => set_value('id_produk', $row->id_produk),
        'jumlah_pemakaian' => set_value('jumlah_pemakaian', $row-
>jumlah_pemakaian),
    );
    $this->load->view('layout/layoutProduksi', $data);
} else {
    $this->session->set_flashdata('message', 'Record Not Found');
    redirect(site_url('komposisi'));
}
}

public function update_action() {
    $this->_rules();

    if ($this->form_validation->run() == FALSE) {
        $this->update($this->input->post('id_komposisi', TRUE));
    }else{
        $data = array(
            'id_bahanbaku' => $this->input->post('id_bahanbaku',TRUE),
            'id_produk' => $this->input->post('id_produk',TRUE),
            'jumlah_pemakaian' => $this->input-
>post('jumlah_pemakaian',TRUE),
        );

        $this->Komposisi_model->update($this->input->post('id_komposisi',
TRUE), $data);
        $this->session->set_flashdata('message', ' <div class="alert alert-info
no-border">
            <button type="button" class="close" data-
dismiss="alert"><span>x</span></button><span class="sr-only">Close</span></div>
            <span class="text-semibold">Success!</span> Data Berhasil di
Update .
            </div>');
        redirect(site_url('komposisi'));
    }
}

public function delete($id)
{
    $row = $this->Komposisi_model->get_by_id($id);
}

```

```

        if ($row) {
            $this->Komposisi_model->delete($id);
            $this->session->set_flashdata('message', '<div class="alert bg-danger">
                <button type="button" class="close" data-
dismiss="alert"><span>x</span></span><span class="sr-only">Close</span></button>
                <span class="text-semibold">Success!</span> Data Berhasil dihapus
            </div>');
            redirect(site_url('komposisi'));
        } else {
            $this->session->set_flashdata('message', 'Record Not Found');
            redirect(site_url('komposisi'));
        }
    }

    public function _rules()
    {
        $this->form_validation->set_rules('id_bahanbaku', 'id bahanbaku',
'trim|required');
        $this->form_validation->set_rules('id_produk', 'id produk', 'trim|required');
        $this->form_validation->set_rules('jumlah_pemakaian', 'jumlah
pemakaian', 'trim|required|numeric');

        $this->form_validation->set_rules('id_komposisi', 'id_komposisi', 'trim');
        $this->form_validation->set_error_delimiters('<span class="text-
danger">', '</span>');
    }
}

```

D.5 Kode program class *controller*/Mesin

```

<?php

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

class Mesin extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
        $this->load->model('Mesin_model');
        $this->load->model('m_admin', 'madmin');
        if (!$this->session->userdata('username')) {
            redirect('C_Login');
        }
    }
}

```

```
    } else {  
  
    }  
}  
  
public function index()  
{  
    $mesin = $this->Mesin_model->get_all();  
    $user = $this->session->userdata('username');  
    $this->breadcrumbs->push('Mesin', '/mesin');  
    $scoba = $this->madmin->detail_user($user)->result();  
  
    $data = array(  
        'title' => 'Mesin',  
        'content' => 'mesin/tb_mesin_list',  
        'breadcrumbs' => $this->breadcrumbs->show(),  
        'user' => $user ,  
        'detail' => $scoba,  
  
        'mesin_data' => $mesin  
    );  
  
    $this->load->view('layout/layoutProduksi', $data);  
}  
  
public function index2()  
{  
    $mesin = $this->Mesin_model->get_all();  
    $user = $this->session->userdata('username');  
    $this->breadcrumbs->push('Mesin', '/mesin');  
    $scoba = $this->madmin->detail_user($user)->result();  
  
    $data = array(  
        'title' => 'Mesin',  
        'content' => 'mesin/tb_mesin_list_pimpinan',  
        'breadcrumbs' => $this->breadcrumbs->show(),  
        'user' => $user ,  
        'detail' => $scoba,  
  
        'mesin_data' => $mesin  
    );  
  
    $this->load->view('layout/layoutPimpinan', $data);  
}  
  
public function read($id)
```

```
{
    $user = $this->session->userdata('username');
    $this->breadcrumbs->push('Mesin', '/mesin');
    $this->breadcrumbs->push('detail', '/mesin/read');
    $row = $this->Mesin_model->get_by_id($id);
    $coba = $this->admin->detail_user($user)->result();
    if ($row) {
        $data = array(
            'title' => 'Mesin',
            'content' => 'mesin/tb_mesin_read',
            'breadcrumbs' => $this->breadcrumbs->show(),
            'user' => $user,
            'detail' => $coba,

            'id_mesin' => $row->id_mesin,
            'nama_mesin' => $row->nama_mesin,
            'jumlah_mesin' => $row->jumlah_mesin,
            'jam_kerjamesin' => $row->jam_kerjamesin,
            'jam_total' => $row->jam_total,
        );
        $this->load->view('layout/layoutProduksi', $data);
    } else {
        $this->session->set_flashdata('message', 'Record Not Found');
        redirect(site_url('mesin'));
    }
}

public function create()
{
    $user = $this->session->userdata('username');
    $this->breadcrumbs->push('Mesin', '/mesin');
    $this->breadcrumbs->push('tambah', '/mesin/create');
    $coba = $this->admin->detail_user($user)->result();
    $data = array(
        'title' => 'Mesin',
        'content' => 'mesin/tb_mesin_form',
        'breadcrumbs' => $this->breadcrumbs->show(),
        'user' => $user,
        'detail' => $coba,

        'button' => 'Tambah',
        'action' => site_url('mesin/create_action'),
        'id_mesin' => set_value('id_mesin'),
        'nama_mesin' => set_value('nama_mesin'),
        'jumlah_mesin' => set_value('jumlah_mesin'),
        'jam_kerjamesin' => set_value('jam_kerjamesin'),
```

```
'jam_total' => set_value('jam_total'),
);
$this->load->view('layout/layoutProduksi', $data);
}

public function create_action()
{
    $this->_rules();

    if ($this->form_validation->run() == FALSE) {
        $this->create();
    } else {
        $jam_kerjamesin=$this->input->post('jam_kerjamesin', TRUE)*3600;
        $jam_total=$jam_kerjamesin*$this->input->post('jam_aktif', TRUE);
        $data = array(
            'nama_mesin' => $this->input->post('nama_mesin',TRUE),
            'jumlah_mesin' => $this->input->post('jumlah_mesin',TRUE),
            'jam_kerjamesin' => $jam_kerjamesin,
            'jam_total' => $jam_total
        );

        $this->Mesin_model->insert($data);
        $this->session->set_flashdata('message', 'Create Record Success');
        redirect(site_url('mesin'));
    }
}

public function update($id)
{
    $user = $this->session->userdata('username');
    $this->breadcrumbs->push('Mesin', '/mesin');
    $this->breadcrumbs->push('update', '/mesin/update');
    $coba = $this->madmin->detail_user($user)->result();

    $row = $this->Mesin_model->get_by_id($id);
    if ($row) {
        $data = array(
            'title' => 'Mesin',
            'content' => 'mesin/tb_mesin_form',
            'breadcrumbs' => $this->breadcrumbs->show(),
            'user' => $user ,
            'detail' => $coba,

            'button' => 'Update',
            'action' => site_url('mesin/update_action'),
            'id_mesin' => set_value('id_mesin', $row->id_mesin),
```



```
'nama_mesin' => set_value('nama_mesin', $row->nama_mesin),
'jumlah_mesin' => set_value('jumlah_mesin', $row->jumlah_mesin),
'jam_kerjamesin' => set_value('jam_kerjamesin', $row-
>jam_kerjamesin),
'jam_total' => set_value('jam_total', $row->jam_total),
);
$this->load->view('layout/layoutProduksi', $data);
} else {
$this->session->set_flashdata('message', 'Record Not Found');
redirect(site_url('mesin'));
}
}

public function update_action()
{
$this->_rules();

if ($this->form_validation->run() == FALSE) {
$this->update($this->input->post('id_mesin', TRUE));
} else {
$data = array(
'nama_mesin' => $this->input->post('nama_mesin',TRUE),
'jumlah_mesin' => $this->input->post('jumlah_mesin',TRUE),
'jam_kerjamesin' => $this->input->post('jam_kerjamesin',TRUE),
'jam_total' => $this->input->post('jam_total',TRUE),
);

$this->Mesin_model->update($this->input->post('id_mesin', TRUE),
$data);
$this->session->set_flashdata('message', 'Update Record Success');
redirect(site_url('mesin'));
}
}

public function delete($id)
{
$row = $this->Mesin_model->get_by_id($id);

if ($row) {
$this->Mesin_model->delete($id);
$this->session->set_flashdata('message', 'Delete Record Success');
redirect(site_url('mesin'));
} else {
$this->session->set_flashdata('message', 'Record Not Found');
redirect(site_url('mesin'));
}
}
```

```
}

public function _rules()
{
    $this->form_validation->set_rules('nama_mesin', 'nama mesin',
'trim|required');
    $this->form_validation->set_rules('jumlah_mesin', 'jumlah mesin',
'trim|required');
    $this->form_validation->set_rules('jam_pemakaian', 'jam pemakaian',
'trim|required');
    $this->form_validation->set_rules('jam_aktif', 'jam aktif', 'trim|required');

    $this->form_validation->set_rules('id_mesin', 'id mesin', 'trim');
    $this->form_validation->set_error_delimiters('<span class="text-danger">',
'</span>');
}
}
```

D.6 Kode program class *controller/Optimasi*

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Optimasi extends CI_Controller {

    function __construct()
    {
        parent::__construct();
        $this->load->model('Produk_model');
        $this->load->model('linear');
        $this->load->model('m_admin', 'madmin');
        $this->load->model('m_login');

        if (!$this->session->userdata('username')) {
            redirect('C_Login');
        } else {
        }
    }
}

public function index(){

    $user = $this->session->userdata('username');
    $this->breadcrumbs->push('Bahanbaku', '/bahanbaku');
    $coba = $this->madmin->detail_user($user)->result();
```

```
$data = array(
    'title' => 'Optimalisasi',
    'content' => 'formCek',
    'breadcrumbs' => $this->breadcrumbs->show(),
    'user' => $user,
    'detail' => $coba );

$this->load->view('layout/layoutProduksi', $data);
}

public function hitungOp(){
    $this->linear->insertdata();
    $produk = $this->Produk_model->get_all();
    $batasan = $this->linear->get_nilai_optimasi();
    $user = $this->session->userdata('username');
    $coba = $this->madmin->detail_user($user)->result();
    $this->breadcrumbs->push('Optimasi', '/optimasi/test');

    foreach ($produk as $pd ) {
        $nama_produk[$pd->id_produk] = $pd->nama_produk;
        $harga_produk[$pd->id_produk] = $pd->harga_produk;
        $id_produk[$pd->id_produk] = $pd->id_produk;
    }
}

//-----tabel fungsi batasan dan tabel simplex 1-----
$optimasi = array();
$tabel = array();
$hasilz = array();
$tabel1 = array();
$tabel2 = array();
$i = 3;
foreach ($batasan as $km ) {
    $stemp = array($km->nilai_kanan,$km->x1/1000,$km->x2/1000);
    for ($j=0; $j < count($batasan); $j++) {
        $angka = 0;
        array_push($stemp, $angka);
    }
    $stemp[$i] = 1;
    array_push($tabel1, $stemp);
    $i++;
}

array_push($tabel, $tabel1);
$max = count($batasan);
$z = array(0,-$harga_produk[1],-$harga_produk[2]);
for ($j=0; $j < count($batasan); $j++) {
```

```

    $angka = 0;
    array_push($z, $angka);
}

array_push($hasilz, $z);
//----- isi matrix simplex ke i -----
$ulang = false;
$count = 1;
for ($n=0; $n < $count; $n++) {
    $minz = min($hasilz[$n]);
    if ($minz < 0) {

        $count++;
        array_push($tabel, $tabel1);
        $index = array();
        $nilai2 = array();
        $bariskunci = 0;
        $sid = 0;
        $kunci = array_search($minz, $hasilz[$n]);
        foreach ($tabel[$n] as $optima) {
            $kolomkunci[$n][] = $optima[$kunci];
            $skanan[$n][] = $optima[0];
        }
        for ($i=0; $i < $max; $i++) {
            if ($kolomkunci[$n][$i] != 0) {
                $index[$i] = $skanan[$n][$i] / $kolomkunci[$n][$i];
            } else if ($kolomkunci[$n][$i] == 0) {
                $index[$i] = "null";
            }
        }
        $this->linear->insertindex($n,$sid,$index[$i]);
        $sid++;
    }
    $sqmin = $this->linear->get_min_index($n);
    foreach ($sqmin->result_array() as $key) {
        $minin = $key['minindex'];
    }
    $minindex = array_search($minin, $index);
    $nilaibarisminindex = array();
    for ($i=0; $i < $max + 3; $i++) {
        for ($j=0; $j < $max; $j++) {
            $nilai2 = $tabel[$n][$minindex][$i] /
$kolomkunci[$n][$minindex];
            $tabel[$n+1][$minindex][$i] = $nilai2;
        }
        array_push($nilaibarisminindex, $nilai2);
    }
}

```

```

//----- tabel bayangan-----
$tabelbayangan = array();
for ($i=0; $i < $max; $i++) {
    for ($j=0; $j < $max + 3; $j++) {
        $nilaibayangan = $tabel[$n+1][$minindex][$j] *
$kolomkunci[$n][$i];
        $tabelbayangan[$n][$i][$j] = $nilaibayangan;
    }
}
$zbayangan = array();
for ($i=0; $i < $max + 3; $i++) {
    $nilaiz = $tabel[$n+1][$minindex][$i] * $hasilz[$n][$kunci];
    if ($nilaiz == -0) {
        $nilaiz = 0;
    }
    $zbayangan[$n][$i] = $nilaiz;
}

//----- mengisi tabel berikutnya-----
for ($i=0; $i < $max; $i++) {
    if ($i == $minindex) {
        $tabel[$n+1][$minindex] = $nilaibarisminindex;
    }else{
        for ($j=0; $j < $max + 3; $j++) {
            $nilai2 = $tabel[$n][$i][$j] - $tabelbayangan[$n][$i][$j];
            $tabel[$n+1][$i][$j] = $nilai2;
        }
    }
}
$zi = array();
for ($i=0; $i < $max + 3; $i++) {
    $nilaiz = $hasilz[$n][$i] - $zbayangan[$n][$i];
    if ($nilaiz == -0) {
        $nilaiz = 0;
    }
    array_push($zi, $nilaiz);
}
array_push($hasilz, $zi);
}

//-----mencari hasil akhir-----
-----
$stabelmax = count($tabel) - 1;
foreach ($tabel[$stabelmax] as $optima) {
    $kolomx1[] = $optima[1];
}

```

```
$kolomx2[] = $optima[2];
}
$x1 = array_search(1, $kolomx1);
$x2 = array_search(1, $kolomx2);

//-----load data-----
$data = array(
    'title' => 'Optimasi' ,
    'content' => 'test',
    'breadcrumbs' => $this->breadcrumbs->show(),
    'user' => $user ,
    'produk' => $produk,
    'detail' => $coba,
    'nama_produk' => $nama_produk,
    'harga_produk' => $harga_produk,
    'optimasi' => $optimasi,
    'tabel' => $tabel,
    'z' => $hasilz,
    'x1' => $x1,
    'x2' => $x2,
    'max' => $max
);

$this->load->view('layout/layoutProduksi', $data);
}
}
```

D.7 Kode program class *controller*/Pekerjaan

```
<?php

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

class Pekerjaan extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
        $this->load->model('Pekerjaan_model');
        $this->load->model('m_admin', 'madmin');
        if (!$this->session->userdata('username')) {
            redirect('C_Login');
        } else {
        }
    }
}
```

```
}

public function index()
{
    $pekerjaan = $this->Pekerjaan_model->get_all();
    $user = $this->session->userdata('username');
    $this->breadcrumbs->push('Pekerjaan', '/pekerjaan');
    $coba = $this->madmin->detail_user($user->result());

    $data = array(
        'title' => 'Pekerjaan',
        'content' => 'pekerjaan/tb_pekerjaan_list',
        'breadcrumbs' => $this->breadcrumbs->show(),
        'user' => $user ,
        'detail' => $coba,

        'pekerjaan_data' => $pekerjaan
    );

    $this->load->view('layout/layoutProduksi', $data);
}

public function index2()
{
    $pekerjaan = $this->Pekerjaan_model->get_all();
    $user = $this->session->userdata('username');
    $this->breadcrumbs->push('Pekerjaan', '/pekerjaan');
    $coba = $this->madmin->detail_user($user->result());

    $data = array(
        'title' => 'Pekerjaan',
        'content' => 'pekerjaan/tb_pekerjaan_list_pimpinan',
        'breadcrumbs' => $this->breadcrumbs->show(),
        'user' => $user ,
        'detail' => $coba,

        'pekerjaan_data' => $pekerjaan
    );

    $this->load->view('layout/layoutPimpinan', $data);
}

public function read($id)
{
    $user = $this->session->userdata('username');
    $this->breadcrumbs->push('Pekerjaan', '/pekerjaan');
```

```
$this->breadcrumbs->push('detail', '/pekerjaan/read');
$coba = $this->admin->detail_user($user->result());
$row = $this->Pekerjaan_model->get_by_id($id);
if ($row) {
    $data = array(
        'title' => 'Pekerjaan',
        'content' => 'pekerjaan/tb_pekerjaan_read',
        'breadcrumbs' => $this->breadcrumbs->show(),
        'user' => $user,
        'detail' => $coba,

        'id_pekerjaan' => $row->id_pekerjaan,
        'id_produk' => $row->nama_produk,
        'id_mesin' => $row->nama_mesin,
        'nama_pekerjaan' => $row->nama_pekerjaan,
        'waktu_pekerjaan' => $row->waktu_pekerjaan,
    );
    $this->load->view('layout/layoutProduksi', $data);
} else {
    $this->session->set_flashdata('message', 'Record Not Found');
    redirect(site_url('pekerjaan'));
}
}

public function create()
{
    $user = $this->session->userdata('username');
    $this->breadcrumbs->push('Pekerjaan', '/pekerjaan');
    $this->breadcrumbs->push('tambah', '/pekerjaan/create');
    $coba = $this->admin->detail_user($user->result());
    $data = array(
        'title' => 'Pekerjaan',
        'content' => 'pekerjaan/tb_pekerjaan_form',
        'breadcrumbs' => $this->breadcrumbs->show(),
        'user' => $user,
        'detail' => $coba,

        'button' => 'Tambah',
        'action' => site_url('pekerjaan/create_action'),
        'id_pekerjaan' => set_value('id_pekerjaan'),
        'id_produk' => set_value('id_produk'),
        'id_mesin' => set_value('id_mesin'),
        'nama_pekerjaan' => set_value('nama_pekerjaan'),
        'waktu_pekerjaan' => set_value('waktu_pekerjaan'),
    );
    $this->load->view('layout/layoutProduksi', $data);
```



```
}

public function create_action()
{
    $this->_rules();

    if ($this->form_validation->run() == FALSE) {
        $this->create();
    } else {
        $data = array(
            'id_produk' => $this->input->post('id_produk',TRUE),
            'id_mesin' => $this->input->post('id_mesin',TRUE),
            'nama_pekerjaan' => $this->input->post('nama_pekerjaan',TRUE),
            'waktu_pekerjaan' => $this->input->post('waktu_pekerjaan',TRUE),
        );

        $this->Pekerjaan_model->insert($data);
        $this->session->set_flashdata('message', 'Create Record Success');
        redirect(site_url('pekerjaan'));
    }
}

public function update($id)
{
    $user = $this->session->userdata('username');
    $this->breadcrumbs->push('Pekerjaan', '/pekerjaan');
    $this->breadcrumbs->push('update', '/pekerjaan/update');
    $coba = $this->madmin->detail_user($user)->result();

    $row = $this->Pekerjaan_model->get_by_id($id);
    if ($row) {
        $data = array(
            'title' => 'Pekerjaan',
            'content' => 'pekerjaan/tb_pekerjaan_form',
            'breadcrumbs' => $this->breadcrumbs->show(),
            'user' => $user,
            'detail' => $coba,

            'button' => 'Update',
            'action' => site_url('pekerjaan/update_action'),
            'id_pekerjaan' => set_value('id_pekerjaan', $row->id_pekerjaan),
            'id_produk' => set_value('id_produk', $row->id_produk),
            'id_mesin' => set_value('id_mesin', $row->id_mesin),
            'nama_pekerjaan' => set_value('nama_pekerjaan', $row-
>nama_pekerjaan),
            'waktu_pekerjaan' => set_value('waktu_pekerjaan', $row-
```

```
>waktu_pekerjaan),
    );
    $this->load->view('layout/layoutProduksi', $data);
} else {
    $this->session->set_flashdata('message', 'Record Not Found');
    redirect(site_url('pekerjaan'));
}
}

public function update_action()
{
    $this->_rules();

    if ($this->form_validation->run() == FALSE) {
        $this->update($this->input->post('id_pekerjaan', TRUE));
    } else {
        $data = array(
            'id_produk' => $this->input->post('id_produk', TRUE),
            'id_mesin' => $this->input->post('id_mesin', TRUE),
            'nama_pekerjaan' => $this->input->post('nama_pekerjaan', TRUE),
            'waktu_pekerjaan' => $this->input->post('waktu_pekerjaan', TRUE),
        );

        $this->Pekerjaan_model->update($this->input->post('id_pekerjaan', TRUE),
        $data);
        $this->session->set_flashdata('message', 'Update Record Success');
        redirect(site_url('pekerjaan'));
    }
}

public function delete($id)
{
    $row = $this->Pekerjaan_model->get_by_id($id);

    if ($row) {
        $this->Pekerjaan_model->delete($id);
        $this->session->set_flashdata('message', 'Delete Record Success');
        redirect(site_url('pekerjaan'));
    } else {
        $this->session->set_flashdata('message', 'Record Not Found');
        redirect(site_url('pekerjaan'));
    }
}

public function _rules()
{
```

```

    $this->form_validation->set_rules('id_produk', 'id produk', 'trim|required');
    $this->form_validation->set_rules('id_mesin', 'id mesin', 'trim|required');
    $this->form_validation->set_rules('nama_pekerjaan', 'nama pekerjaan',
'trim|required');
    $this->form_validation->set_rules('waktu_pekerjaan', 'waktu pekerjaan',
'trim|required');

    $this->form_validation->set_rules('id_pekerjaan', 'id pekerjaan', 'trim');
    $this->form_validation->set_error_delimiters('<span class="text-danger">',
'</span>');
    }
}

```

D.8 Kode program class *controller*/ Produk

```

<?php

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

class Produk extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
        $this->load->model('Produk_model');
        $this->load->model('m_admin', 'madmin');
        if (!$this->session->userdata('username')) {
            redirect('C_Login');
        } else {

        }
    }
}

//=====View Produk
Admin=====
public function index()
{
    $produk = $this->Produk_model->get_all();
    $user = $this->session->userdata('username');
    $this->breadcrumbs->push('Produk', '/produk');
    $cobra = $this->madmin->detail_user($user)->result();

    $data = array(
        'title' => 'Produk',
        'content' => 'produk/tb_produk_list',
        'breadcrumbs' => $this->breadcrumbs->show(),

```

```
'user' => $user ,
'detail' => $coba,

'produk_data' => $produk
);

$this->load->view('layout/layout', $data);
}
//=====View Produk
Pimpinan=====
public function index2()
{
    $produk = $this->Produk_model->get_all();
    $user = $this->session->userdata('username');
    $this->breadcrumbs->push('Produk', '/produk');
    $coba = $this->madmin->detail_user($user)->result();

    $data = array(
        'title' => 'Produk' ,
        'content' => 'produk/tb_produk_list_pimpinan',
        'breadcrumbs' => $this->breadcrumbs->show(),
        'user' => $user ,
        'detail' => $coba,

        'produk_data' => $produk
    );

    $this->load->view('layout/layoutPimpinan', $data);
}
public function read($id)
{
    $user = $this->session->userdata('username');
    $this->breadcrumbs->push('Produk', '/produk');
    $this->breadcrumbs->push('detail', '/produk/read');
    $coba = $this->madmin->detail_user($user)->result();
    $row = $this->Produk_model->get_by_id($id);
    if ($row) {
        $data = array(
            'title' => 'Produk' ,
            'content' => 'produk/tb_produk_read',
            'breadcrumbs' => $this->breadcrumbs->show(),
            'user' => $user ,
            'detail' => $coba,

            'id_produk' => $row->id_produk,
            'nama_produk' => $row->nama_produk,
```

```
        'harga_produk' => $row->harga_produk,
    );
    $this->load->view('layout/layout', $data);
} else {
    $this->session->set_flashdata('message', 'Record Not Found');
    redirect(site_url('produk'));
}
}

public function create()
{
    $user = $this->session->userdata('username');
    $this->breadcrumbs->push('Produk', '/produk');
    $this->breadcrumbs->push('tambah', '/produk/create');
    $coba = $this->madmin->detail_user($user)->result();
    $data = array(
        'title'      => 'Produk',
        'content'    => 'produk/tb_produk_form',
        'breadcrumbs' => $this->breadcrumbs->show(),
        'user'       => $user,
        'detail'     => $coba,

        'button' => 'Tambah',
        'action' => site_url('produk/create_action'),
        'id_produk' => set_value('id_produk'),
        'nama_produk' => set_value('nama_produk'),
        'harga_produk' => set_value('harga_produk'),
    );
    $this->load->view('layout/layout', $data);
}

public function create_action()
{
    $this->_rules();

    if ($this->form_validation->run() == FALSE) {
        $this->create();
    } else {
        $data = array(
            'nama_produk' => $this->input->post('nama_produk',TRUE),
            'harga_produk' => $this->input->post('harga_produk',TRUE),
        );

        $this->Produk_model->insert($data);
        $this->session->set_flashdata('message', '<div class="alert alert-success no-
border">
```

```
<button type="button" class="close" data-  
dismiss="alert"><span>×</span><span class="sr-only">Close</span></button>  
  <span class="text-semibold">Success!</span> Data Berhasil di Inputkan  
.  
  </div>');  
  redirect(site_url('produk'));  
}  
}  
  
public function update($id)  
{  
  $user = $this->session->userdata('username');  
  $this->breadcrumbs->push('Produk', '/produk');  
  $this->breadcrumbs->push('update', '/produk/update');  
  $coba = $this->madmin->detail_user($user)->result();  
  
  $row = $this->Produk_model->get_by_id($id);  
  if ($row) {  
    $data = array(  
      'title' => 'Produk',  
      'content' => 'produk/tb_produk_form',  
      'breadcrumbs' => $this->breadcrumbs->show(),  
      'user' => $user ,  
      'detail' => $coba,  
  
      'button' => 'Update',  
      'action' => site_url('produk/update_action'),  
      'id_produk' => set_value('id_produk', $row->id_produk),  
      'nama_produk' => set_value('nama_produk', $row->nama_produk),  
      'harga_produk' => set_value('harga_produk', $row->harga_produk),  
    );  
    $this->load->view('layout/layout', $data);  
  } else {  
    $this->session->set_flashdata('message', 'Record Not Found');  
    redirect(site_url('produk'));  
  }  
}  
  
public function update_action()  
{  
  $this->_rules();  
  
  if ($this->form_validation->run() == FALSE) {  
    $this->update($this->input->post('id_produk', TRUE));  
  } else {  
    $data = array(  

```

```
'nama_produk' => $this->input->post('nama_produk',TRUE),
'harga_produk' => $this->input->post('harga_produk',TRUE),
);

$this->Produk_model->update($this->input->post('id_produk', TRUE),
$data);
$this->session->set_flashdata('message',
' <div class="alert alert-info no-border">
<button type="button" class="close" data-
dismiss="alert"><span>×</span><span class="sr-only">Close</span></button>
<span class="text-semibold">Success!</span> Data Berhasil di Update .
</div>');
redirect(site_url('produk'));
}
}

public function delete($id)
{
$row = $this->Produk_model->get_by_id($id);

if ($row) {
$this->Produk_model->delete($id);
$this->session->set_flashdata('message', '<div class="alert bg-danger">
<button type="button" class="close" data-
dismiss="alert"><span>×</span><span class="sr-only">Close</span></button>
<span class="text-semibold">Success!</span> Data Berhasil dihapus .
</div>');
redirect(site_url('produk'));
} else {
$this->session->set_flashdata('message', 'Record Not Found');
redirect(site_url('produk'));
}
}

public function _rules()
{
$this->form_validation->set_rules('nama_produk', 'nama produk',
'trim|required');
$this->form_validation->set_rules('harga_produk', 'harga produk',
'trim|required|numeric');

$this->form_validation->set_rules('id_produk', 'id_produk', 'trim');
$this->form_validation->set_error_delimiters('<span class="text-danger">',
'</span>');
}
}
```

```
}
```

D.9 Kode program class *controller*/ satuan

```
<?php  
  
if (!defined('BASEPATH')) exit('No direct script access allowed');  
  
class satuan extends CI_Controller  
{  
    function __construct()  
    {  
        parent::__construct();  
        $this->load->model('Satuan_model');  
        $this->load->model('m_admin', 'madmin');  
        if (!$this->session->userdata('username')) {  
            redirect('C_Login');  
        } else {  
  
        }  
    }  
  
    public function index(){  
  
        $user = $this->session->userdata('username');  
        $coba = $this->madmin->detail_user($user)->result();  
        $satuan = $this->Satuan_model->get_all();  
        $this->breadcrumbs->push('Satuan', '/satuan');  
  
        $data = array(  
            'title' => 'Satuan',  
            'content' => 'satuan/tb_satuan_list',  
            'breadcrumbs' => $this->breadcrumbs->show(),  
            'user' => $user ,  
            'detail' => $coba,  
  
            'satuan_data' => $satuan  
        );  
  
        $this->load->view('layout/layout', $data);  
    }  
  
    public function create()  
    {  
        $user = $this->session->userdata('username');
```



```
$this->breadcrumbs->push('satuan', '/satuan');
$this->breadcrumbs->push('tambah', '/satuan/create');
$coaba = $this->madmin->detail_user($user->result());
$data = array(
    'title' => 'satuan',
    'content' => 'satuan/tb_satuan_form',
    'breadcrumbs' => $this->breadcrumbs->show(),
    'user' => $user,
    'detail' => $coaba,

    'button' => 'Tambah',
    'action' => site_url('satuan/create_action'),
    'id_satuan' => set_value('id_satuan'),
    'nama_satuan' => set_value('nama_satuan'),
);
$this->load->view('layout/layout', $data);
}

public function create_action()
{
    $this->_rules();

    if ($this->form_validation->run() == FALSE) {
        $this->create();
    } else {
        $data = array(
            'nama_satuan' => $this->input->post('nama_satuan', TRUE),
        );

        $this->Satuan_model->insert($data);
        $this->session->set_flashdata('message', '<div class="alert alert-success no-
border">
    <button type="button" class="close" data-
dismiss="alert"><span>×</span><span class="sr-only">Close</span></button>
    <span class="text-semibold">Success!</span> Data Berhasil di Inputkan
    </div>');
        redirect(site_url('satuan'));
    }
}

public function update($id)
{
    $user = $this->session->userdata('username');
    $this->breadcrumbs->push('satuan', '/satuan');
    $this->breadcrumbs->push('update', '/satuan/update');
```

```
$coba = $this->madmin->detail_user($user)->result();

$row = $this->Satuan_model->get_by_id($id);
if ($row) {
    $data = array(
        'title' => 'satuan',
        'content' => 'satuan/tb_satuan_form',
        'breadcrumbs' => $this->breadcrumbs->show(),
        'user' => $user,
        'detail' => $coba,

        'button' => 'Update',
        'action' => site_url('satuan/update_action'),
        'id_satuan' => set_value('id_satuan', $row->id_satuan),
        'nama_satuan' => set_value('nama_satuan', $row->nama_satuan)
    );
    $this->load->view('layout/layout', $data);
} else {
    $this->session->set_flashdata('message', 'Record Not Found');
    redirect(site_url('satuan'));
}
}

public function update_action()
{
    $this->_rules();

    if ($this->form_validation->run() == FALSE) {
        $this->update($this->input->post('id_satuan', TRUE));
    } else {
        $data = array(
            'nama_satuan' => $this->input->post('nama_satuan', TRUE),
        );

        $this->Satuan_model->update($this->input->post('id_satuan', TRUE),
        $data);
        $this->session->set_flashdata('message',
        ' <div class="alert alert-info no-border">
        <button type="button" class="close" data-
        dismiss="alert"><span></span><span class="sr-only">Close</span></button>
        <span class="text-semibold">Success!</span> Data Berhasil di Update .
        </div>');
        redirect(site_url('satuan'));
    }
}
```

```
public function delete($id)
{
    $row = $this->Satuan_model->get_by_id($id);

    if ($row) {
        $this->Satuan_model->delete($id);
        $this->session->set_flashdata('message', '<div class="alert bg-danger">
            <button type="button" class="close" data-
dismiss="alert"><span>×</span><span class="sr-only">Close</span></button>
            <span class="text-semibold">Success!</span> Data Berhasil dihapus .
            </div>');
        redirect(site_url('satuan'));
    } else {
        $this->session->set_flashdata('message', 'Record Not Found');
        redirect(site_url('satuan'));
    }
}

public function _rules()
{
    $this->form_validation->set_rules('nama_satuan', 'nama_satuan',
'trim|required');

    $this->form_validation->set_rules('id_satuan', 'id_satuan', 'trim');
    $this->form_validation->set_error_delimiters('<span class="text-danger">',
'</span>');
}
}
```

LAMPIRAN E. GAMBAR MESIN



Gambar cheese vat



Gambar alat mixing (mixer)



Gambar vacuum sealer



Gambar Stretcher



Gambar Kompor



Gambar Cup Sealer

LAMPIRAN F. PENGUJIAN BLACK BOX

Tabel Pengujian black box

No.	Menu	Fungsi	Kasus	Hasil	Ket
1.	Kelola data Produk	Menu ini digunakan oleh admin untuk menambah dan mengubah data produk	Ketika aktor memilih menu data produk dan tidak ada data produk yang tersimpan	Menampilkan tabel kosong	OK
			Ketika aktor memilih menu data produk dan ada data produk yang tersimpan	Menampilkan tabel yang terdapat data produk di dalamnya	OK
			Ketika aktor memilih button update	Menampilkan halaman form edit dan data produk yang dipilih	OK
			Ketika aktor klik button simpan pada halaman edit produk namun data yang diisi tidak lengkap	Menampilkan pesan error pada bagian form yang belum di isi	OK

			Ketika aktor klik button update pada halaman edit produk dan data yang diisi sudah lengkap	Menampilkan halaman data produk dan menampilkan pesan “ Data Berhasil Diubah”	OK
			Ketika aktor klik button cancel pada halaman edit data	Kembali menampilkan halaman data produk	OK
2.	Kelola data bahan baku	Menu ini digunakan bagian produksi untuk menambah dan mengubah data bahan baku	Ketika aktor memilih menu data bahan baku dan tidak ada data bahan baku yang tersimpan	Menampilkan tabel kosong	OK
			Ketika aktor memilih menu data bahan baku dan ada data bahan baku yang tersimpan	Menampilkan tabel yang terdapat data bahan baku di dalamnya	OK
			Ketika aktor klik button tambah bahan baku	Menampilkan halaman form tambah bahan baku	OK

			Ketika aktor klik button simpan pada halaman form tambah bahan baku namun data yang diisi tidak lengkap	Menampilkan pesan error pada bagian form yang belum di isi	OK
			Ketika aktor klik button simpan pada halaman form tambah bahan baku dan data yang diisi sudah lengkap	Menampilkan halaman data bahan baku dan pesan “Data Berhasil Disimpan”	OK
			Ketika aktor klik button kembalipada halaman form tambah bahan baku	Menampilkan halaman data bahan baku	OK
			Ketika aktor memilih button update	Menampilkan form edit dan data bahan baku yang dipilih	OK
			Ketika aktor	Menampilkan	OK

			klik button update pada halaman edit bahan baku namun data yang diisi tidak lengkap	pesan error pada bagian form yang belum di isi	
			Ketika aktor klik button update pada halaman edit bahan baku dan data yang diisi sudah lengkap	Menampilkan halaman data bahan baku dan menampilkan pesan “ Data Berhasil Diubah”	OK
			Ketika aktor klik button teks bahan baku pada edit bahan baku	Menampilkan halaman data bahan baku	OK
3.	Kelola data mesin	Menu ini digunakan bagian produksi untuk menambah dan mengubah data mesin	Ketika aktor memilih menu data mesin dan tidak ada data mesin yang tersimpan	Menampilkan tabel kosong	OK
			Ketika aktor memilih menu data mesin dan	Menampilkan tabel yang terdapat data	OK

			ada data mesin yang tersimpan	mesin di dalamnya	
			Ketika aktor klik button tambah mesin	Menampilkan halaman form tambah mesin	OK
			Ketika aktor klik button simpan pada halaman form tambah mesin namun data yang diisi tidak lengkap	Menampilkan pesan error pada bagian form yang belum di isi	OK
			Ketika aktor klik button simpan pada halaman form tambah mesin dan data yang diisi sudah lengkap	Menampilkan halaman data mesin dan pesan “Data Berhasil Disimpan”	OK
			Ketika aktor klik button kembalipada halaman form tambah mesin	Menampilkan halaman data mesin	OK
			Ketika aktor memilih button	Menampilkan form edit dan	OK

			update	data mesin yang dipilih	
			Ketika aktor klik button update pada halaman edit mesin namun data yang diisi tidak lengkap	Menampilkan pesan error pada bagian form yang belum di isi	OK
			Ketika aktor klik button update pada halaman edit mesin dan data yang diisi sudah lengkap	Menampilkan halaman data mesin dan menampilkan pesan “ Data Berhasil Diubah”	OK
			Ketika aktor klik button teks mesin pada edit mesin	Menampilkan halaman data mesin	OK
4.	Kelola data elemen pekerjaan	Menu ini digunakan bagian produksi untuk menambah dan mengubah data elemen pekerjaan	Ketika aktor memilih menu data elemen pekerjaan dan tidak ada data elemen pekerjaan yang tersimpan	Menampilkan tabel kosong	OK

			Ketika aktor memilih menu data elemen pekerjaan dan ada data elemen pekerjaan yang tersimpan	Menampilkan tabel yang terdapat data elemen pekerjaan di dalamnya	OK
			Ketika aktor klik button tambah elemen pekerjaan	Menampilkan halaman form tambah elemen pekerjaan	OK
			Ketika aktor klik button simpan pada halaman form tambah elemen pekerjaan namun data yang diisi tidak lengkap	Menampilkan pesan error pada bagian form yang belum di isi	OK
			Ketika aktor klik button simpan pada halaman form tambah elemen pekerjaan dan data yang diisi	Menampilkan halaman data elemen pekerjaan dan pesan “Data Berhasil Disimpan”	OK

			sudah lengkap		
			Ketika aktor klik button kembalipada halaman form tambah elemen pekerjaan	Menampilkan halaman data elemen pekerjaan	OK
			Ketika aktor memilih button update	Menampilkan form edit dan data elemen pekerjaan yang dipilih	OK
			Ketika aktor klik button update pada halaman edit elemen pekerjaan namun data yang diisi tidak lengkap	Menampilkan pesan error pada bagian form yang belum di isi	OK
			Ketika aktor klik button update pada halaman edit elemen pekerjaan dan data yang diisi	Menampilkan halaman data elemen pekerjaan dan menampilkan pesan “ Data Berhasil	OK

			sudah lengkap	Diubah”	
			Ketika aktor klik button teks elemen pekerjaan pada edit elemen pekerjaan	Menampilkan halaman data elemen pekerjaan	OK
5.	Kelola data penggunaan bahan baku / komposisi	Menu ini digunakan bagian produksi untuk menambah dan mengubah data penggunaan bahan baku / komposisi	Ketika aktor memilih menu data penggunaan bahan baku / komposisi dan tidak ada data penggunaan bahan baku / komposisi yang tersimpan	Menampilkan tabel kosong	OK
			Ketika aktor memilih menu data penggunaan bahan baku / komposisi dan ada data penggunaan bahan baku / komposisi yang tersimpan	Menampilkan tabel yang terdapat data penggunaan bahan baku / komposisi di dalamnya	OK
			Ketika aktor	Menampilkan	OK

			klik button tambah penggunaan bahan baku / komposisi	halaman form tambah penggunaan bahan baku / komposisi	
			Ketika aktor klik button simpan pada halaman form tambah penggunaan bahan baku / komposisi namun data yang diisi tidak lengkap	Menampilkan pesan error pada bagian form yang belum di isi	OK
			Ketika aktor klik button simpan pada halaman form tambah penggunaan bahan baku / komposisi dan data yang diisi sudah lengkap	Menampilkan halaman data penggunaan bahan baku / komposisi dan pesan “Data Berhasil Disimpan”	OK
			Ketika aktor klik button	Menampilkan halaman data	OK

			kembalipada halaman form tambah penggunaan bahan baku / komposisi	penggunaan bahan baku / komposisi	
			Ketika aktor memilih button update	Menampilkan form edit dan data penggunaan bahan baku / komposisi yang dipilih	OK
			Ketika aktor klik button update pada halaman edit penggunaan bahan baku / komposisi namun data yang diisi tidak lengkap	Menampilkan pesan error pada bagian form yang belum di isi	OK
			Ketika aktor klik button update pada halaman edit penggunaan	Menampilkan halaman data penggunaan bahan baku / komposisi	OK

			bahan baku / komposisi dan data yang diisi sudah lengkap	dan menampilkan pesan “ Data Berhasil Diubah”	
			Ketika aktor klik button teks penggunaan bahan baku / komposisi pada edit penggunaan bahan baku / komposisi	Menampilkan halaman data penggunaan bahan baku / komposisi	OK
6.	Kelola data user	Menu ini digunakan bagian produksi untuk menambah dan mengubah data user	Ketika aktor memilih menu data user dan tidak ada data user yang tersimpan	Menampilkan tabel kosong	OK
			Ketika aktor memilih menu data user dan ada data user yang tersimpan	Menampilkan tabel yang terdapat data user di dalamnya	OK
			Ketika aktor klik button tambah user	Menampilkan halaman form tambah user	OK
			Ketika aktor	Menampilkan	OK

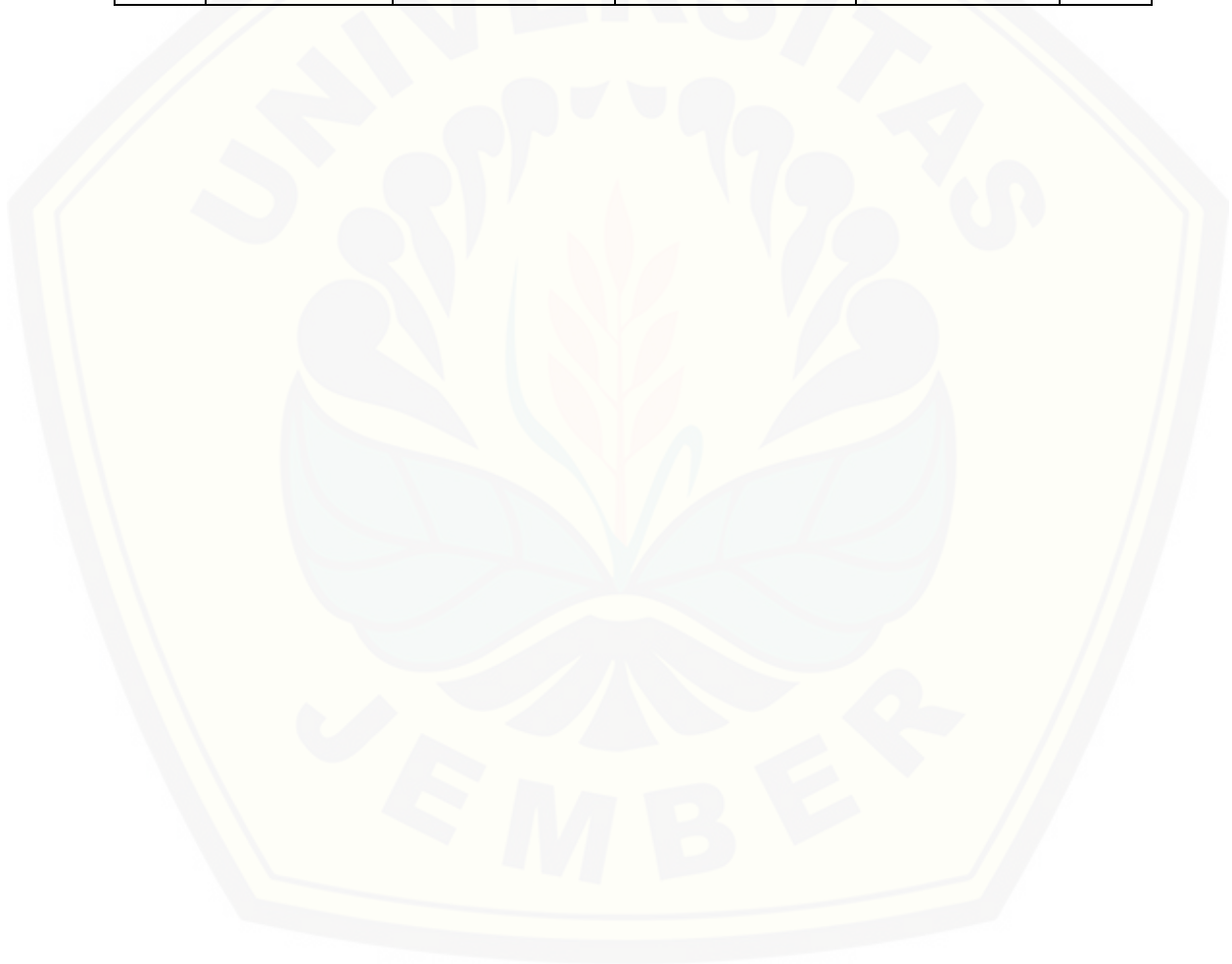
			klik button simpan pada halaman form tambah user namun data yang diisi tidak lengkap	pesan error pada bagian form yang belum di isi	
			Ketika aktor klik button simpan pada halaman form tambah user dan data yang diisi sudah lengkap	Menampilkan halaman data user dan pesan “Data Berhasil Disimpan”	OK
			Ketika aktor klik button kembali pada halaman form tambah user	Menampilkan halaman data user	OK
			Ketika aktor memilih button update	Menampilkan form edit dan data user yang dipilih	OK
			Ketika aktor klik button update pada halaman edit user namun data	Menampilkan pesan error pada bagian form yang belum di isi	

			yang diisi tidak lengkap		
			Ketika aktor klik button update pada halaman edit user dan data yang diisi sudah lengkap	Menampilkan halaman data user dan menampilkan pesan “ Data Berhasil Diubah”	OK
			Ketika aktor klik button teks user pada edit user	Menampilkan halaman data user	OK
7.	Lihat data produk	Menu ini digunakan bagian produksi dan pemimpin untuk melihat data produk	Ketika aktor memilih menu produk dan tidak ada data produk yang tersimpan	Menampilkan tabel kosong	OK
			Ketika aktor memilih menu data produk dan ada data produk yang tersimpan	Menampilkan tabel yang terdapat data produk di dalamnya	OK
8.	Lihat data bahan baku	Menu ini digunakan bagian produksi dan	Ketika aktor memilih menu bahan baku dan tidak ada data	Menampilkan tabel kosong	OK

		pemimpin untuk melihat data bahan baku	bahan baku yang tersimpan		
			Ketika aktor memilih menu data bahan baku dan ada data bahan baku yang tersimpan	Menampilkan tabel yang terdapat data bahan baku di dalamnya	OK
9.	Lihat data mesin	Menu ini digunakan bagian produksi dan pemimpin untuk melihat data mesin	Ketika aktor memilih menu mesin dan tidak ada data mesin yang tersimpan	Menampilkan tabel kosong	OK
			Ketika aktor memilih menu data mesin dan ada data mesin yang tersimpan	Menampilkan tabel yang terdapat data mesin di dalamnya	OK
10.	Lihat data elemen pekerjaan	Menu ini digunakan bagian produksi dan pemimpin untuk melihat data elemen pekerjaan	Ketika aktor memilih menu elemen pekerjaan dan tidak ada data elemen pekerjaan yang tersimpan	Menampilkan tabel kosong	OK
			Ketika aktor memilih menu	Menampilkan tabel yang	OK

			data elemen pekerjaan dan ada data elemen pekerjaan yang tersimpan	terdapat data elemen pekerjaan di dalamnya	
11.	Lihat data penggunaan bahan baku / komposisi	Menu ini digunakan bagian produksi dan pemimpin untuk melihat data penggunaan bahan baku / komposisi	Ketika aktor memilih menu penggunaan bahan baku / komposisi dan tidak ada data penggunaan bahan baku / komposisi yang tersimpan	Menampilkan tabel kosong	OK
			Ketika aktor memilih menu data penggunaan bahan baku / komposisi dan ada data penggunaan bahan baku / komposisi yang tersimpan	Menampilkan tabel yang terdapat data penggunaan bahan baku / komposisi di dalamnya	OK
12.	Lihat data user	Menu ini digunakan bagian	Ketika aktor memilih menu user dan tidak	Menampilkan tabel kosong	OK

		produksi dan pemimpin untuk melihat data user	ada data user yang tersimpan		
			Ketika aktor memilih menu data user dan ada data user yang tersimpan	Menampilkan tabel yang terdapat data user di dalamnya	OK



LAMPIRAN G. PROSES PRODUKSI

Kegiatan proses produksi keju dan yoghurt dilakukan oleh pegawai bagian produksi. Rata-rata bahan baku susu yang dibutuhkan oleh Rumah Yoghurt dalam proses produksi keju mozzarella adalah 816 kg dimana 1 liter susu setara dengan 1,02 kg susu dan menghasilkan 100 kg keju mozzarella. Sedangkan rata-rata bahan baku susu yang dibutuhkan oleh Rumah Yoghurt dalam proses produksi yoghurt adalah 40 liter dan menghasilkan 40 liter yoghurt. Proses produksi keju dan yoghurt dilakukan di ruang utama produksi Rumah Yoghurt. Proses pembuatan keju di Rumah Yoghurt secara ringkas tersaji pada gambar dibawah ini



Gambar 40 Proses Produksi Keju Mozzarella

Pada gambar diatas dapat dilihat proses pembuatan keju mozzarella pada Rumah Yoghurt. Berikut ini penjelasan dari proses produksi keju dan kebutuhan bahan baku yang digunakan sebagai berikut:

1. Pemilihan bahan baku

Pemilihan bahan baku dilakukan secara manual. Susu yang dipilih harus benar-benar dalam kualitas yang terbaik. Rata-rata bahan baku susu yang dibutuhkan

oleh Rumah Yoghurt dalam proses produksi keju mozzarella adalah 816 kg susu dimana 1 liter susu setara dengan 1,02 kg susu dan menghasilkan 100 kg keju mozzarella dari 800 liter susu. Sehingga didapatkan untuk setiap 1 kg keju membutuhkan bahan baku susu sapi segar sebanyak 8 liter. Susu diangkut menggunakan pick up yang masuk ke dalam ruang produksi. Susu dialirkan dari tangki menuju cheese vat dengan menggunakan selang panjang yang dialiri dengan air panas sebelumnya. Selang dihubungkan dengan pompa, supaya dapat mengalirkan susu dengan mudah.

2. Pasteurisasi

Pasteurisasi susu pada suhu $71 - 72^{\circ}\text{C}$ selama 15 detik dan menurunkan suhu hingga menjadi 36°C .



Gambar 1 Pemanasan Susu

3. Penambahan asam sitrat, enzim rennin dan garam

Kemudian ditambahkan asam sitrat sebanyak 2% dari jumlah susu yang digunakan. Kemudian ditambahkan enzim rennin (0,025%) dari jumlah susu yang digunakan. Misalkan susu yang digunakan 8 liter, maka enzim rennin yang digunakan adalah 2 ml. Namun sebelum penggunaannya enzim rennin harus diencerkan terlebih dahulu dengan menggunakan air matang minimum sebanyak 10 kali dari jumlah rennet yang digunakan (misal 25 ml air untuk mengencerkan 2 ml enzim rennin). Sedangkan penggunaan bahan baku garam sebanyak 3,5 % dari jumlah susu yang digunakan (0,28 gram).

4. Pemotongan hasil dan Mixing

Pemotongan curd berbentuk kubus dan dibiarkan selama 15 menit. Dan melakukan pembuangan hasil sisa keju.



Gambar 42 pemotongan curd

Proses pencampuran (mixing) dilakukan dengan menggunakan *mixer stainless* yang telah dimodifikasi oleh perusahaan. Proses pencampuran dihentikan jika sudah tercampur dengan rata. Proses mixing ditunjukkan pada gambar 3.6



Gambar 43 Proses Mixing

Proses selanjutnya adalah pemuluran pada mesin stretching. Mesin stretching dipanaskan terlebih dahulu selama kurang lebih setengah jam dengan suhu sekitar 85°C . Dalam proses ini *curd* dimasukkan ke dalam mesin stretching selama 30 menit melalui mulut mesin bagian atas. Pada saat *curd* dimasukkan ke dalam stretcher terdapat proses pemasakan dan penekanan, sehingga nanti akan

didapatkan tekstur keju mozzarella yang diinginkan, yaitu mulur, kenyal dan lunak.

5. Pengemasan

Setelah proses pemuluran selesai, tahapan selanjutnya adalah penakaran. Proses ini dilakukan dimeja dengan membagi penakaran menjadi menjadi, yaitu kemasan 1 Kg. Dalam proses ini penakaran menggunakan sarung tangan plastic dan masker wajah, supaya keju yang akan dikemas supaya tidak terkontaminasi oleh kuman. Keju diletakan wadah didalam kotak yang sebelumnya dilapisi plastik. Selanjutnya keju tersebut dikemas lagi menggunakan plastic yang tebal dan segera di *vacuum* agar keju bertahan lebih lama.



Gambar 44 Proses Produksi Yoghurt

Pada gambar diatas dapat dilihat proses pembuatan yoghurt pada Rumah Yoghurt. Berikut ini penjelasan dari masing-masing proses produksi sebagai berikut:

1. Pemilihan bahan baku

Susu merupakan hasil pemerahan sapi atau hewan menyusui lainnya yang susunya dapat dikonsumsi sebagai makanan yang sehat, serta tidak dikurangi komponen-komponennya atau ditambah bahan-bahan lain. Susu yang digunakan di Rumah Yoghurt berasal dari peternak sapi disekitar kecamatan junrejo yang dikumpul di KUD Mitra Bhakti Makmur. Susu dari KUD merupakan pasokan susu dari peternak sapi perah yang berada dikawasan Kecamatan Junrejo ketika susu dari peternak datang dan masuk ke ruang penampungan susu di KUD terlebih dahulu dilakukan penyaringan dengan kain kasa dengan tujuan untuk memisahkan susu dengan kotoran yang tercampur.

2. Pasteurisasi

Proses pasteurisasi susu dirumah yoghurt menggunakan sistem *High Temperature Short Time* (HTST) yaitu dengan cara memanaskan susu dalam panci *double wall* terbuat dari *stainless steel*, sehingga susu tidak rusak karena langsung terkena api pada suhu 72°C selama 15 detik dengan cara ini dapat membunuh bakteri yang dapat memacu kerusakan susu. Proses pasteurisasi sesekali susu diaduk untuk menghindari penggumpalan pada susu.

3. Memasukkan starter

Starter yang digunakan oleh Rumah Yoghurt terdiri dari bakteri *Lactobacillus bulgaricus*, *Lactobacillus acidophilus* dan *streptococcus thermophiles*. Susu pasteurisasi yang akan dibuat yoghurt dimasukkan dalam toples dan botol steril untuk ditambahkan *sarter*. Penambahan *starter* sebanyak 5% dari berat susu yang digunakan.

4. Inkubasi

Susu yang telah ditambahkan starter ditutup rapat, kemudian diletakkan di tempat khusus untuk di inkubasi selama 24 jam.

5. Penambahan air dan bahan tambahan lainnya

Penambahan air dilakukan dengan perbandingan 1 : 1. Air yang digunakan merupakan air mineral kemasan isi ulang. Penambahan pada yoghurt drink agar

yoghurt tidak kental dan lebih mudah dikonsumsi. Setelah itu yoghurt dimasukkan kedalam panci double wall, untuk dilakukan pemanasan dan penambahan gula, tepung maizena dan susu skim bubuk sedikit demi sedikit sambil diaduk hingga rata.

6. Pengemasan

Yoghurt *drink* dikemas dalam dua bentuk kemasan yaitu kemasan cup dan kemasan botol. Proses pengemasan yoghurt *drink* dalam *cup* dilakukan dengan cara memasukkan yoghurt *drink* kedalam cup yang berukuran 140 ml kemudian *cup* ditutup dengan *cup sealer* dan diberi label.

