



**PENERAPAN METODE *PERIOD ORDER QUANTITY* (POQ) PADA
APLIKASI PENDUKUNG OPTIMALISASI PERSEDIAAN
BAHAN BAKU KAIN DI UD.DWIDAKU JAYA**

SKRIPSI

Oleh:

Arif Putra Ardi Hansa

NIM 112410101072

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2015





**PENERAPAN METODE *PERIOD ORDER QUANTITY* (POQ) PADA
APLIKASI PENDUKUNG OPTIMALISASI PERSEDIAAN
BAHAN BAKU KAIN DI UD.DWIDAKU JAYA**

SKRIPSI

diajukan guna melengkapai dan memenuhi salah satu syarat
untuk menyelesaikan pendidikan Program Studi Sistem Informasi (SI)
dan mencapai gelar Sarjana Komputer

Oleh:

Arif Putra Ardi Hansa

NIM 112410101072

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER**

2015

PERSEMBAHAN

1. Allah SWT yang selalu memberikan hal-hal terbaik dan terindah bagiku.
2. Bapak Samsul Arifin, ibu Titik Budi Hananingsih yang selalu mendukung dan memanjatkan doa bagi setiap langkah yang saya jalani.
3. Adek Putri Ayu Budi Arifinda, Ajeng Larasati dan Arum Dwi Safitri yang selalu mendoakan saya
4. Santi Ermania, S.P yang membantu saya menyelesaikan penelitian
5. Keluarga cafe 86 yang selalu memberi inspirasi dan motivasi
6. Teman-teman angkatan 2011 yang tidak bisa saya sebutkan satu per satu

MOTTO

∞∞∞∞

“Orang – orang hebat di bidang apapun bukan baru bekerja karena mereka terinspirasi, namun mereka menjadi terinspirasi karena mereka lebih suka bekerja. Mereka tidak menyia-nyiakan waktu untuk menunggu inspirasi.” (Ernest Newman)

∞∞∞∞

“Orang – Orang yang sukses telah belajar membuat mereka melakukan hal yang harus dikerjakan ketika mereka melakukan hal yang harus dikerjakan ketika hal itu memang harus dikerjakan, entah mereka menyukainya atau tidak.” (Schopenhauer)

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Arif Putra Ardi Hansa

NIM : 112410101072

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul **“Penerapan Metode *Period Order Quantity* (POQ) pada Aplikasi Pendukung Optimalisasi Persediaan Bahan Baku Kain di UD. Dwidaku Jaya”** adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

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

Jember, Agustus 2015

Yang menyatakan,

Arif Putra Ardi Hansa

NIM 112410101072

SKRIPSI

diajukan guna melengkapi dan memenuhi salah satu syarat
untuk menyelesaikan pendidikan Program Studi Sistem Informasi (SI)
dan mencapai gelar Sarjana Komputer

**PENERAPAN METODE *PERIOD ORDER QUANTITY* (POQ) PADA
APLIKASI PENDUKUNG OPTIMALISASI PERSEDIAAN BAHAN BAKU
KAIN DI UD.DWIDAKU JAYA**

Oleh

Arif Putra Ardi Hansa

NIM 112410101072

Pembimbing

Dosen Pembimbing Utama : Anang Andrianto, S.T.,M.T.

Dosen Pembimbing Anggota : Yanuar Nurdiansyah, ST., M.Cs

PENGESAHAN PEMBIMBING

Skripsi berjudul “Penerapan Metode *Period Order Quantity* (POQ) pada Aplikasi Pendukung Optimalisasi Persediaan Bahan Baku Kain di UD.Dwidaku Jaya”, telah diuji dan disahkan pada:

Hari, tanggal : Senin, 24 Agustus 2015

Tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Anang Andrianto, ST., M.T.

NIP 196906151997021002

Yanuar Nurdiansyah, ST., M.Cs

NIP 198201012010121004

PENGESAHAN

Skripsi yang berjudul “Penerapan Metode *Period Order Quantity* (POQ) pada Aplikasi Pendukung Optimalisasi Persediaan Bahan Baku Kain di UD.Dwidaku Jaya” telah diuji dan disahkan pada :

Hari, tanggal : Senin, 24 Agustus 2015

Tempat : Program Studi Sistem Informasi Universitas Jember

Tim Penguji :

Penguji I,

Penguji II,

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

NIP 196811131994121001

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

NIP 196909281993021001

Mengesahkan,
Ketua Program Studi,

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

NIP 196704201992011001

RINGKASAN

Penerapan Metode *Period Order Quantity* ((POQ) pada Aplikasi Optimalisasi Persediaan Bahan Baku Kain di UD.Dwidaku Jaya; Arif Putra Ardi Hansa, 112410101072; halaman; Program Studi Sistem Informasi Universitas Jember.

Strategi pemasaran yang baik merupakan hal yang mendasar dalam mengikuti persaingan pasar, dengan didukung pengendalian persediaan bahan baku. Strategi pemasaran dan pengendalian persediaan bahan baku yang baik akan mempermudah menembus persaingan pasar yang begitu ketatnya pada saat sekarang ini. Kurangnya persediaan bahan baku yang ada dapat menyebabkan kekecewaan pelanggan yang akan memesan sehingga pelanggan akan berpindah ke perusahaan lain. Namun perusahaan harus dapat menjalankan strategi pemasaran dan pengendalian persediaan bahan baku dengan baik dan seimbang. Sebab strategi pemasaran yang kurang baik juga memberikan pengaruh terhadap perkembangan perusahaan yang berakibat pada jumlah pesanan sehingga mempengaruhi proses produksi yang dilakukan oleh perusahaan. Usaha untuk mempertahankan pelanggan inilah yang harus selalu diperhatikan oleh manajemen perusahaan.

UD. Dwidaku Jaya merupakan perusahaan yang mengolah kain menjadi produk pakaian. Agar dapat memenuhi permintaan konsumen yang tinggi maka kegiatan produksi harus dilakukan secara optimal. Salah satu hal yang harus diperhatikan yaitu pelaksanaan pengendalian persediaan kain. Pengendalian persediaan kain pada perusahaan tersebut masih dilakukan secara manual dengan melakukan pencatatan pada sebuah buku besar yang di dalamnya dituliskan data-data mengenai pemesanan, penjualan, dan pembelian persediaan kain. Oleh sebab itu, seringkali perusahaan mengalami kesulitan dalam mengendalikan persediaan kain. Pemesanan kain yang dilakukan oleh perusahaan UD.Dwidaku Jaya selama ini mengikuti jumlah pesanan yang ada. Perusahaan hanya akan memesan atau membeli kain untuk produksi jika ada pesanan yang masuk, sehingga perusahaan akan melakukan pembelian kain lebih dari 4 (empat) kali dalam satu bulan. Jumlah

pembelian yang dilakukan perusahaan UD.Dwidaku Jaya selama ini sangat tidak optimal, karena dapat membebani keuangan perusahaan dalam segi pengeluaran biaya pemesanan atau pembelian kain.

Metode *period order quantity* (POQ) merupakan metode yang dapat membantu proses pengendalian persediaan bahan baku kain. Metode ini menggunakan pendekatan pada konsep jumlah pemesanan ekonomis yang dipakai pada setiap periode yang bersifat permintaan diskrit atau beragam. Perhitungan metode *periode order quantity* (POQ) menggunakan dasar pemesanan ekonomis yang nantinya akan digunakan sebagai data pendukung untuk menghitung interval pemesanan yang optimal. Sehingga perusahaan dapat mengurangi beban biaya yang dikeluarkan dalam melakukan pemesanan sebab dengan menggunakan metode tersebut pemesanan yang dilakukan oleh perusahaan tidak akan lebih dari 4 (empat) kali dalam satu bulan.

Aplikasi optimalisasi persediaan bahan baku kain ini merupakan solusi yang tepat untuk menerapkan metode *Period Order Quantity* sehingga dapat memudahkan manajemen UD.Dwidaku Jaya dalam mengendalikan persediaan bahan baku kain. Aplikasi tersebut akan diimplementasikan dalam bentuk penggunaan *web* dan dikembangkan menggunakan bahasa pemrograman PHP dengan *Framework Code Igniter*.

PRAKATA

Alhamdulillah, puji syukur penulis haturkan kehadiran Allah SWT. yang telah memberikan rahmat, taufik beserta hidayahnya, sehingga penulis mampu menyelesaikan skripsi yang berjudul “**Penerapan Metode *Period Order Quantity* (POQ) Pada Aplikasi Pendukung Optimalisasi Persediaan Bahan Baku Kain di UD.Dwidaku Jaya**”. Sholawat serta salam, semoga senantiasa tetap tercurahkan atas junjungan Nabi Muhammad SAW.

Tanpa adanya bimbingan atau arahan dari berbagai pihak, skripsi yang telah tersusun ini tidak dapat terselesaikan dengan baik. Oleh karena itu, penulis menyampaikan terimakasih kepada :

1. Bapak, ibu dan keluarga besar yang telah mendidik, membimbing, mendukung, dan senantiasa mendoakan.
2. Ketua Program Studi Sistem Informasi, Bapak Prof. Drs. Slamin, M.Comp. Sc., Ph.D.
3. Bapak Anang Andrianto, S.T.,M.T. dan Bapak Yanuar Nurdiansyah, ST., M.Cs selaku dosen pembimbing yang senantiasa sabar dalam memberikan arahan dan motivasi untuk dapat menyelesaikan skripsi ini.
4. Dr. Saiful Bukhori, ST., M.Kom., selaku dosen penguji I dan Drs. Antonius Cahya P, M.App., Sc., Ph.D selaku dosen penguji II yang telah memberikan masukan dalam penulisan skripsi ini
5. Manajemen serta seluruh pegawai UD.Dwidaku Jaya yang telah mendukung.
6. Segenap dosen Program Studi Sistem Informasi Universitas Jember.
7. Almamater tercinta Program Studi Sistem Informasi Universitas Jember.

Semoga Allah SWT selalu memberikan hidayah dan rahmat kepada semua pihak yang telah membantu dengan ikhlas sehingga skripsi ini dapat terselesaikan.

Jember, Agustus 2015

Penulis

DAFTAR ISI

PERSEMBAHAN.....	ii
MOTTO	iii
PERNYATAAN.....	iv
SKRIPSI.....	v
PENGESAHAN PEMBIMBING.....	vi
PENGESAHAN	vii
RINGKASAN	viii
PRAKATA.....	x
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xviii
DAFTAR GAMBAR	xix
DAFTAR LAMPIRAN.....	xx
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan dan Manfaat.....	3
1.3.1 Tujuan	3
1.3.2 Manfaat	4
1.4 Ruang Lingkup.....	4
1.5 Sistematika Penulisan.....	5
BAB 2. TINJAUAN PUSTAKA	6
2.1 Persediaan.....	6

2.2	Jenis Persediaan.....	6
2.3	Biaya Pengelolaan Persediaan.....	8
2.4	Pengendalian Persediaan	9
2.5	Bahan Baku	10
2.7	Metode POQ.....	11
2.8	Metode Pengembangan Sistem	11
a)	Pengujian <i>White Box</i>	14
b)	Pengujian Kotak Hitam (Black Box Testing).....	15
BAB 3.	METODOLOGI PENELITIAN	16
3.1	Jenis Penelitian.....	16
3.2	Metode Penelitian.....	16
3.3	Tempat dan Waktu Penelitian	16
3.4	Tahapan Penelitian	16
3.4.1	Tahap Pengumpulan Data	17
3.4.2	Tahap Analisis.....	18
3.5	Tahap pengembangan aplikasi	20
3.5.1	Analisis Kebutuhan	20
3.5.2	Desain Aplikasi	20
3.5.3	Pengkodean	21
3.5.4	Pengujian.....	21
BAB 4.	PENGEMBANGAN APLIKASI.....	22
4.1	Statement of Purpose.....	22
4.2	Analisis Kebutuhan Aplikasi.....	22

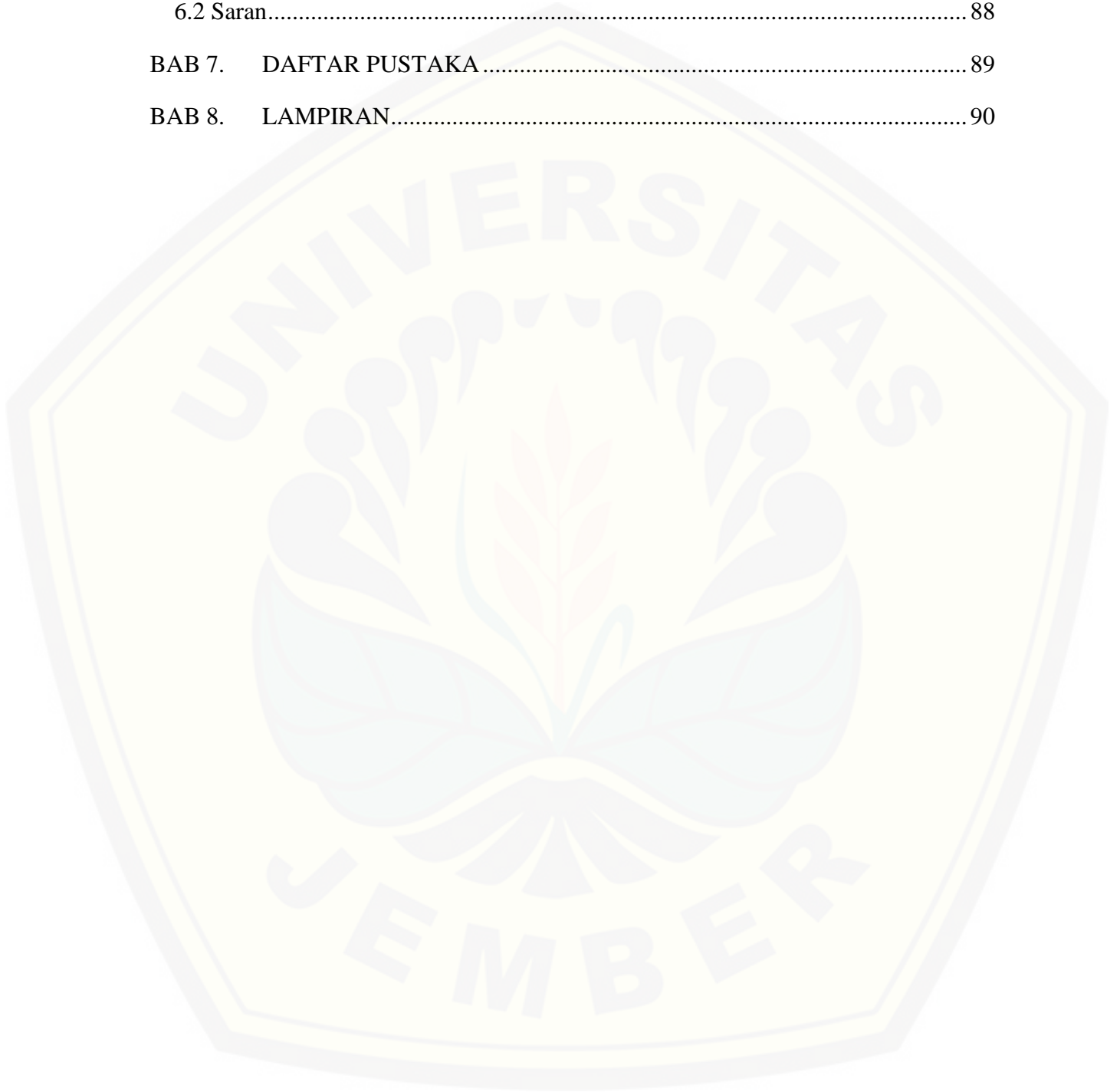
4.2.1	Kebutuhan Fungsional.....	22
4.2.2	Kebutuhan Non-Fungsional	23
4.3	Desain Aplikasi	23
4.3.1	<i>Business Process</i>	23
4.3.2	<i>Usecase Diagram</i>	24
4.3.3	<i>Usecase Skenario</i>	28
1.	<i>Usecase Skenario Manajemen Desain</i>	29
2.	<i>Usecase Skenario Manajemen Data Pemesanan</i>	31
4.	<i>Usecase Skenario Manajemen Data Kain</i>	33
4.	<i>Usecase Skenario Manajemen Data Stock</i>	36
5.	<i>Usecase Skenario Manajemen Data User</i>	39
6.	<i>Usecase Skenario Manajemen Data Prediksi</i>	42
7.	<i>Usecase Skenario Insert User</i>	44
4.3.4	<i>Activity Diagram</i>	45
1.	<i>Activity Diagram insert user konsumen</i>	45
2.	<i>Activity Diagram insert data desain</i>	46
3.	<i>Activity Diagram insert data kain</i>	47
4.	<i>Activity Diagram insert data pemesanan</i>	48
5.	<i>Activity Diagram insert data stock</i>	48
6.	<i>Activity Diagram insert data user</i>	48
7.	<i>Activity Diagram insert data prediksi</i>	48
8.	<i>Activity Diagram edit data user</i>	48
9.	<i>Activity Diagram Delete data desain</i>	50

10.	<i>Activity Diagram Delete data kain</i>	51
11.	<i>Activity Diagram Delete data pemesanan</i>	51
12.	<i>Activity Diagram Delete data prediksi</i>	51
13.	<i>Activity Diagram Delete data stock</i>	51
14.	<i>Activity Diagram Delete data user</i>	51
15.	<i>Activity Diagram View data desain</i>	51
16.	<i>Activity Diagram View data kain</i>	52
17.	<i>Activity Diagram View data pemesanan</i>	52
18.	<i>Activity Diagram View data stock</i>	52
19.	<i>Activity Diagram View data prediksi</i>	52
20.	<i>Activity Diagram View data user</i>	52
4.3.5	<i>Sequence Diagram</i>	52
1.	<i>Sequence Diagram insert data desain</i>	52
2.	<i>Sequence Diagram insert data kain</i>	54
3.	<i>Sequence Diagram insert data pemesanan</i>	54
4.	<i>Sequence Diagram insert data stock</i>	54
5.	<i>Sequence Diagram insert data prediksi</i>	55
6.	<i>Sequence Diagram insert data pemesanan user konsumen</i>	55
7.	<i>Sequence Diagram insert data user konsumen</i>	55
8.	<i>Sequence Diagram insert data user</i>	56
9.	<i>Sequence Diagram edit user</i>	56
10.	<i>Sequence Diagram Delete data desain</i>	57
11.	<i>Sequence Diagram Delete data kain</i>	58

12.	<i>Sequence Diagram Delete data prediksi</i>	59
13.	<i>Sequence Diagram Delete data user</i>	59
14.	<i>Sequence Diagram Delete data stock</i>	59
15.	<i>Sequence Diagram Delete data pemesanan</i>	60
16.	<i>Sequence Diagram view data desain</i>	60
17.	<i>Sequence Diagram view data kain</i>	61
18.	<i>Sequence Diagram view data stock</i>	61
19.	<i>Sequence Diagram view data user</i>	62
20.	<i>Sequence Diagram view data prediksi</i>	62
21.	<i>Sequence Diagram view data pemesanan</i>	62
4.3.6	<i>Class Diagram</i>	63
4.3.7	<i>Entity Relationship Diagram (ERD)</i>	63
4.4	Pengkodean Aplikasi	64
4.4.1	Menghitung persediaan optimal	64
4.5	Pengujian Aplikasi	71
4.5.1	<i>White Box Testing</i>	71
4.5.2	Pengujian <i>Black Box</i>	74
BAB 5.	HASIL DAN PEMBAHASAN	75
5.1	Hasil Penelitian.....	75
5.2	Pemesanan kain Sebelum Menggunakan Aplikasi.....	75
5.3	Penggunaan Metode <i>POQ</i> Pada Aplikasi.....	78
5.4	Implementasi Aplikasi.....	79
5.4.1	Fitur <i>login</i>	80

5.4.2	Fitur Manajemen Data Kain.....	80
5.4.2.1	Fitur menambah Data Kain.....	80
5.4.2.2	Fitur Menghapus Data Kain.....	80
5.4.3	Fitur Manajemen Data <i>user</i>	80
5.4.3.1	Fitur Menambah Data <i>user</i>	80
5.4.3.2	Fitur Mengubah Data <i>User</i>	81
5.4.3.3	Fitur Menghapus Data <i>user</i>	81
5.4.4	Fitur Manajemen Data Pemesanan.....	81
5.4.4.1	Fitur Menambah Data Pemesanan	81
4.4.4.2	Fitur Menghapus Data Pemesanan	81
5.4.5	Fitur Manajemen Data <i>Stock</i>	81
5.4.5.1	Fitur Menambah Data <i>Stock</i>	82
5.4.5.2	Fitur Menghapus Data <i>Stock</i>	82
5.4.6	Fitur Manajemen Data Desain	82
5.4.6.1	Fitur Menambah Data Desain	82
5.4.6.2	Fitur Menghapus Data desain	82
5.4.7	Fitur Manajemen Data Prediksi	82
5.4.7.1	Fitur Menambah Data Prediksi	83
5.4.7.2	Fitur Menghitung Data Prediksi	83
5.4.7.3	Fitur Menghapus Data Prediksi	84
5.4.7.4	Fitur Melihat data prediksi.....	85
5.5	Pembahasan	86
BAB 6.	PENUTUP	87

6.1 Kesimpulan	87
6.2 Saran.....	88
BAB 7. DAFTAR PUSTAKA.....	89
BAB 8. LAMPIRAN.....	90



DAFTAR TABEL

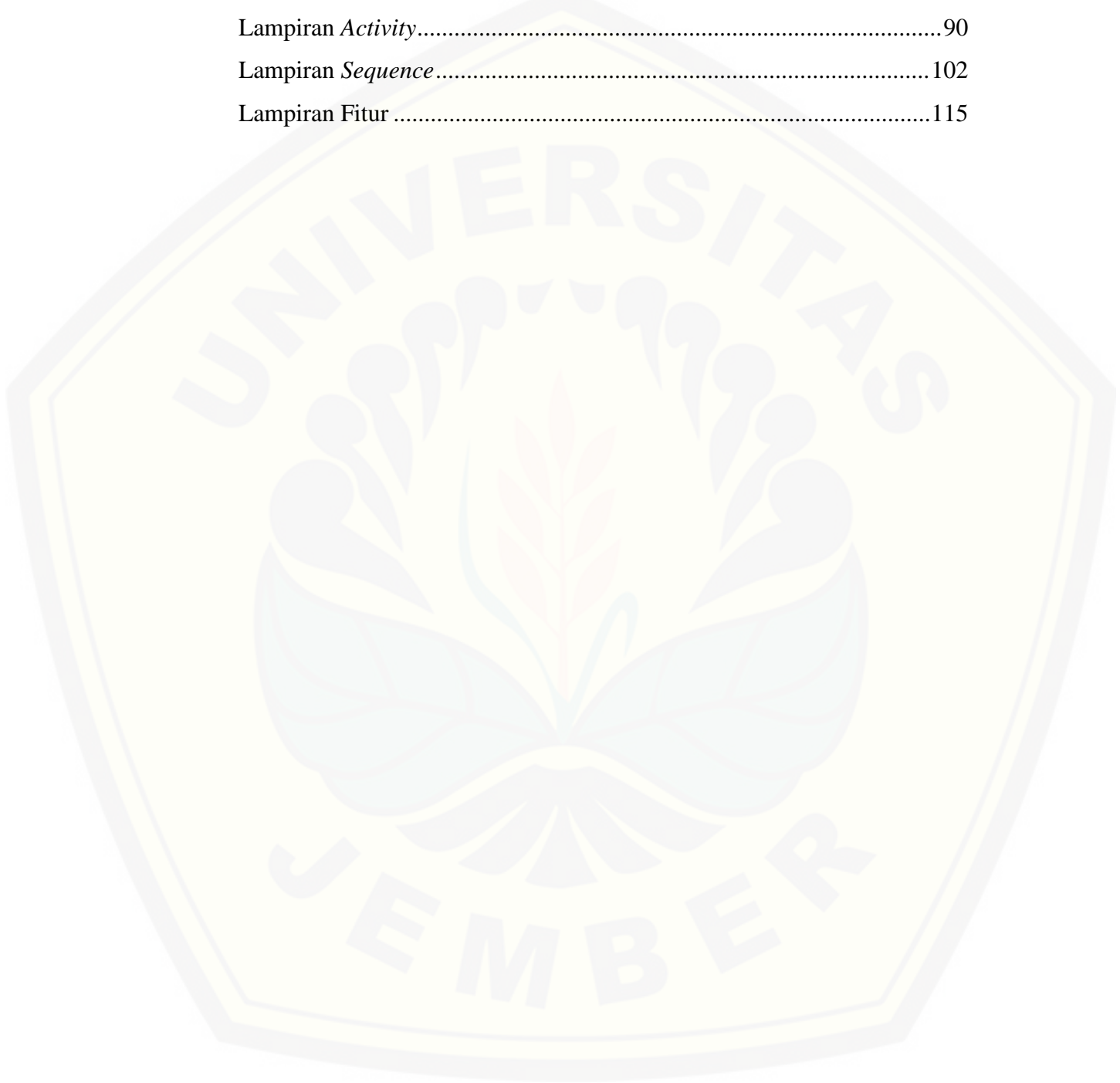
Tabel 4.1 Definisi Aktor	26
Tabel 4.2 Definisi <i>Usecase</i>	27
Tabel 4.3 <i>Usecase</i> Skenario Manajemen Data Desain.....	29
Tabel 4.4 <i>Usecase</i> Skenario Manajemen Data Pemesanan.....	31
Tabel 4.5 <i>Usecase</i> Skenario Manajemen Data Kain.....	33
Tabel 4.6 <i>Usecase</i> Skenario Manajemen Data <i>Stock</i>	37
Tabel 4.7 <i>Usecase</i> Skenario Manajemen Data <i>User</i>	39
Tabel 4.8 <i>Usecase</i> Skenario Manajemen Data Prediksi.....	42
Tabel 4.9 <i>Usecase</i> Skenario Manajemen <i>Insert User</i>	45
Tabel 4.10 <i>Test Case</i> Menghitung POQ	73
Tabel 4.11 Pengujian <i>Black Box</i> Perhitungan POQ	74
Tabel 5.1 Daftar Pemesanan Bulan Maret – Agustus 2015	76
Tabel 5.2 Perbandingan Sebelum dan Sesudah Menggunakan POQ.....	79

DAFTAR GAMBAR

Gambar 2.1 Model <i>Waterfall</i>	12
Gambar 3.1 Diagram Alir Tahapan Penelitian.....	17
Gambar 3.2 Analisis Data Menggunakan Metode POQ	19
Gambar 4.1 <i>Business Process</i> Aplikasi Optimalisasi Persediaan	24
Gambar 4.2 <i>Use Case</i> Diagram Aplikasi Optimalisasi Persediaan.....	25
Gambar 4.3 <i>Activity</i> Diagram <i>Insert User</i> Konsumen	46
Gambar 4.4 <i>Activity</i> Diagram <i>Insert</i> Data Desain.....	47
Gambar 4.5 <i>Activity</i> Diagram <i>Edit</i> Data <i>User</i>	49
Gambar 4.6 <i>Activity</i> Diagram <i>Delete</i> Data Desain.....	50
Gambar 4.7 <i>Activity</i> Diagram <i>View</i> Data Desain	51
Gambar 4.8 <i>Sequence</i> Diagram <i>Insert</i> Data Desain.....	53
Gambar 4.9 <i>Sequence</i> Diagram <i>Edit</i> Data Desain	57
Gambar 4.10 <i>Sequence</i> Diagram <i>Delete</i> Data Desain.....	58
Gambar 4.11 <i>Sequence</i> Diagram <i>View</i> Data Desain	61
Gambar 4.12 <i>Class</i> Diagram Aplikasi Optimalisasi Persediaan	63
Gambar 4.13 ERD.....	64
Gambar 4.14 Kode Program <i>Controler</i> Prediksi	65
Gambar 4.15 Kode Program <i>Model</i> Prediksi	66
Gambar 4.16 Kode Program <i>View</i> Prediksi	71
Gambar 4.17 Kode Program Menghitung POQ.....	72
Gambar 4.18 Grafik Alir Perhitungan.....	72
Gambar 5.1 Form untuk Menambah Data Prediksi	83
Gambar 5.2 Menghitung Data Prediksi.....	84
Gambar 5.3 Menghapus Data Prediksi	85
Gambar 5.4 Melihat Data Prediksi.....	85

DAFTAR LAMPIRAN

Lampiran	90
Lampiran <i>Activity</i>	90
Lampiran <i>Sequence</i>	102
Lampiran <i>Fitur</i>	115



BAB 1. PENDAHULUAN

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

1.1 Latar Belakang

Strategi pemasaran yang baik merupakan hal yang mendasar dalam mengikuti persaingan pasar, dengan didukung pengendalian persediaan bahan baku. Strategi pemasaran dan pengendalian persediaan bahan baku yang baik akan mempermudah menembus persaingan pasar yang begitu ketatnya pada saat sekarang ini. Kurangnya persediaan bahan baku yang ada dapat menyebabkan kekecewaan pelanggan yang akan memesan sehingga pelanggan akan berpindah ke perusahaan lain. Namun perusahaan harus dapat menjalankan strategi pemasaran dan pengendalian persediaan bahan baku dengan baik dan seimbang. Sebab strategi pemasaran yang kurang baik juga memberikan pengaruh terhadap perkembangan perusahaan yang berakibat pada jumlah pesanan sehingga mempengaruhi proses produksi yang dilakukan oleh perusahaan. Usaha untuk mempertahankan pelanggan inilah yang harus selalu diperhatikan oleh manajemen perusahaan.

UD.Dwidaku Jaya merupakan perusahaan yang mengolah kain menjadi produk pakaian. Agar dapat memenuhi permintaan konsumen yang tinggi maka kegiatan produksi harus dilakukan secara optimal. Salah satu hal yang harus diperhatikan yaitu pelaksanaan pengendalian persediaan kain. Pengendalian persediaan kain pada perusahaan tersebut masih dilakukan secara manual dengan melakukan pencatatan pada sebuah buku besar yang di dalamnya dituliskan data-data mengenai pemesanan, penjualan, dan pembelian persediaan kain. Oleh sebab itu, seringkali perusahaan mengalami kesulitan dalam mengendalikan persediaan kain. Pemesanan kain yang dilakukan oleh perusahaan UD.Dwidaku Jaya selama ini mengikuti jumlah pesanan yang ada. Perusahaan hanya akan memesan atau membeli kain untuk produksi jika ada pesanan yang masuk, sehingga perusahaan akan melakukan pembelian kain lebih dari 4 (empat) kali dalam satu bulan.

Jumlah pembelian yang dilakukan perusahaan UD.Dwidaku Jaya selama ini sangat tidak optimal, karena dapat membebani keuangan perusahaan dalam segi pengeluaran biaya pemesanan atau pembelian kain.

Metode *period order quantity* (POQ) merupakan metode yang dapat membantu proses pengendalian persediaan bahan baku kain. Metode ini menggunakan pendekatan pada konsep jumlah pemesanan ekonomis yang dipakai pada setiap periode yang bersifat permintaan diskrit atau beragam. Perhitungan metode *periode order quantity* (POQ) menggunakan dasar pemesanan ekonomis yang nantinya akan digunakan sebagai data pendukung untuk menghitung interval pemesanan yang optimal. Sehingga perusahaan dapat mengurangi beban biaya yang dikeluarkan dalam melakukan pemesanan sebab dengan menggunakan metode tersebut pemesanan yang dilakukan oleh perusahaan tidak akan lebih dari 4 (empat) kali dalam satu bulan.

Penelitian-penelitian tentang pengendalian persediaan telah banyak dilakukan, seperti yang dilakukan (Rahardianti & Azhari, 2013) dengan melakukan penelitian pada PT.ABAISIAT RAYA untuk mengkaji optimalisasi Aplikasi persediaan bahan baku karet mentah (*lateks*) untuk memenuhi permintaan konsumen yang pada saat itu perusahaan PT.Abaisiat Raya tidak ada Aplikasi persediaan *lateks*. Besarnya tingkat permintan konsumen dan konsumsi karet dunia tentunya harus diimbangi dengan ketersediaan bahan baku. Perhitungan persediaan dihitung dengan metode POQ dan hasilnya total biaya yang harus dikeluarkan oleh perusahaan lebih kecil dibanding dengan perhitungan manual yang dilakukan oleh manajemen perusahaan.

Penelitian yang dilakukan (Lestrai & Kusno, 2014) yang melakukan penelitian pada super market asia plaza untuk mengkaji manajemen persediaan buah naga merah dalam memenuhi permintaan konsumen yang pada saat itu pihak manajemen persediaan belum memiliki suatu model yang tetap untuk mengendalikan persediaan yang mereka miliki. Pemesanan yang dilakukan bersifat spekulatif atau hanya berdasarkan kebiasaan dan pengalaman sehingga interval waktu pemesanan pun sangat tidak menentu sehingga menyebabkan biaya penyimpanan yang tinggi karena kelebihan persediaan. Perhitungan persediaan

dihitung dengan metode POQ dan hasilnya terbukti dapat memperkecil biaya total persediaan sebesar 80%.

Berdasarkan latar belakang di atas, penggunaan metode POQ dapat membantu manajemen untuk mengoptimalkan pembelian bahan baku kain yang diperlukan, sehingga persediaan bahan baku kain pada perusahaan tidak mengalami kekurangan atau kelebihan yang dapat mengganggu kegiatan produksi. Metode POQ akan diterapkan pada aplikasi agar semua data yang masuk dan keluar dari perusahaan dapat terkomputerisasi serta dapat mengetahui interval pemesanan yang optimal.

1.2 Rumusan Masalah

Dari uraian latar belakang di atas dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana mengendalikan persediaan bahan baku kain di UD.Dwidaku Jaya agar dapat memenuhi permintaan konsumen, dengan mengoptimalkan periode pemesanan bahan baku kain menggunakan metode *Period Order Quantity* (POQ) ?
2. Bagaimana merancang dan membangun aplikasi untuk membantu mengoptimalkan periode pemesanan bahan baku kain agar tidak terjadi kelebihan stok dengan menggunakan metode *Period Order Quantity* (POQ) ?

1.3 Tujuan dan Manfaat

Tujuan dan manfaat ini merupakan jawaban dari rumusan masalah yang telah disebutkan

1.3.1 Tujuan

Tujuan penelitian ini adalah untuk :

1. Mengetahui persediaan yang optimal serta periode pemesanan sehingga semua permintaan dapat dipenuhi dan tidak terjadi penumpukan persediaan di gudang.
2. Merancang dan membangun aplikasi agar dapat membantu manajemen untuk menyimpan informasi yang digunakan untuk mengoptimalkan persediaan dengan metode *Period Order Quantity* (POQ).

1.3.2 Manfaat

Manfaat yang dari penelitian ini adalah :

1. Manfaat Akademis

Hasil dari penelitian ini diharapkan dapat membantu siapa saja yang membutuhkan informasi yang berhubungan dengan judul penelitian ini. Selain itu, untuk menambah varian judul penelitian yang ada di Program Studi Sistem Informasi Universitas Jember.

2. Manfaat bagi peneliti

a. Mengetahui bagaimana proses penerapan metode *POQ* pada perhitungan prediksi persediaan bahan baku kain yang optimal pada UD.DWIDAKU JAYA

b. Membantu manajemen UD.DWIDAKU JAYA dalam hal menyediakan persediaan bahan baku kain untuk periode berikutnya.

3. Manfaat bagi objek penelitian

a. Memberikan cara baru kepada perusahaan tempat penelitian mengenai penggunaan *Period Order Quantity* (POQ) dalam memprediksi persediaan untuk periode berikutnya.

b. Membantu perusahaan untuk melakukan perhitungan prediksi periode pemesanan persediaan yang optimal

1.4 Ruang Lingkup

Ruang lingkup ini merupakan batasan – batasan masalah dalam penulisan penelitian ini. Penulis memberi batasan masalah untuk objek dan tema yang dibahas sehingga tidak terjadi penyimpangan dalam penulisan dan pembuatan Aplikasi. Berikut adalah batasan yang dicantumkan :

1. Aplikasi yang dibangun hanya membantu proses pengendalian persediaan bahan baku kain dengan mengoptimalkan periode pemesanan.

2. Data yang diolah antara lain adalah data barang, data pemesanan, data *stock*, data kain, data desain, data *user* dan data prediksi

1.5 Sistematika Penulisan

Sistematika penulisan skripsi ini disusun sebagai berikut:

1. Pendahuluan

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

2. Tinjauan Pustaka

Bab ini menjelaskan tentang materi, informasi dan tinjauan pustaka yang menjadi kerangka pemikiran dalam penelitian.

3. Metodologi Penelitian

Bab ini menjelaskan tentang metode penelitian yang digunakan dalam penelitian.

4. Pengembangan Aplikasi

Bab ini menjelaskan tentang pengembangan Aplikasi yang dikembangkan.

5. Hasil dan Pembahasan

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

6. Penutup

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

BAB 2. TINJAUAN PUSTAKA

Pada bagian ini dipaparkan teori-teori serta pustaka yang akan dipakai dalam penelitian. Teori-teori ini diambil dari buku literatur, jurnal dan internet. Berikut merupakan teori-teori yang digunakan dan dibahas dalam penelitian:

2.1 Persediaan

Menurut Prawirosentono, persediaan adalah aktiva lancar yang terdapat dalam perusahaan dalam bentuk persediaan bahan mentah (bahan baku / raw material, bahan setengah jadi / work in process dan barang jadi / finished goods). Persediaan adalah bagian utama dari modal kerja, merupakan aktiva yang pada setiap saat mengalami perubahan. Sehingga dapat disimpulkan bahwa yang dimaksud dengan persediaan adalah suatu aktiva yang harus tersedia dalam perusahaan pada saat diperlukan untuk menjamin kelancaran dalam menjalankan perusahaan.

2.2 Jenis Persediaan

Karakteristik dari barang yang diklasifikasikan sebagai persediaan sangat bervariasi terhadap jenis usaha. Secara umum perusahaan dapat dikelompokkan menjadi tiga jenis, yaitu perusahaan jasa, perusahaan dagang dan perusahaan manufaktur. Oleh sebab itu, jenis-jenis persediaan pada ketiga perusahaan tersebut berbeda.

a. Jenis-jenis Persediaan menurut Fungsinya seperti yang dinyatakan oleh (rangkuti, 2004)

1) *Batch Stock / Lot Size Inventory*

Persediaan yang diadakan karena kita membeli atau membuat bahan-bahan atau barang-barang dalam jumlah yang lebih besar daripada jumlah yang dibutuhkan pada saat itu.

Keuntungannya :

- a) Potongan harga pada harga pembelian.
- b) Efisiensi produksi.
- c) Penghematan biaya angkutan.

2) *Fluctuation Stock*

Persediaan yang diadakan untuk menghadapi fluktuasi permintaan konsumen yang tidak dapat diramalkan.

3) *Anticipation Stock*

Persediaan yang diadakan untuk menghadapi fluktuasi permintaan yang dapat diramalkan, berdasarkan pola musiman yang terdapat dalam satu tahun dan untuk menghadapi penggunaan, penjualan, atau permintaan yang meningkat.

b. Jenis-jenis Persediaan menurut Jenis dan Posisi Barang

1) Persediaan barang dagangan (*merchandise inventory*)

Barang yang ada di gudang (*goods on hand*) dibeli oleh pengecer atau perusahaan perdagangan seperti importir atau eksportir untuk dijual kembali. Biasanya, barang yang diperoleh untuk dijual kembali secara fisik tidak diubah oleh perusahaan pembeli; barang-barang tersebut tetap dalam bentuk yang telah jadi ketika meninggalkan pabrik pembuatnya. Dalam beberapa hal, dapat terjadi beberapa komponen dibeli untuk kemudian dirakit menjadi barang jadi. Misalnya, sepeda yang dirakit dari kerangka, roda, gir, dan sebagainya serta dijual oleh pengecer sepeda adalah salah satu contoh.

2) Persediaan manufaktur (*manufacturing inventory*) Persediaan gabungan dari entitas manufaktur, yang terdiri dari :

a) Persediaan bahan baku

Barang berwujud yang dibeli atau diperoleh dengan cara lain (misalnya, dengan menambang) dan disimpan untuk penggunaan langsung dalam membuat barang untuk dijual kembali. Bagian atau suku cadang yang diproduksi sebelum digunakan kadang-kadang diklasifikasikan sebagai persediaan komponen suku cadang.

b) Persediaan barang dalam proses

Barang-barang yang membutuhkan pemrosesan lebih lanjut sebelum penyelesaian dan penjualan. Barang dalam proses, juga disebut persediaan barang dalam proses, meliputi biaya bahan langsung,

tenaga kerja langsung, dan alokasi biaya overhead pabrik yang terjadi sampai tanggal tersebut.

c) Persediaan barang jadi

Barang-barang manufaktur yang telah diselesaikan dan disimpan untuk dijual. Biaya persediaan barang jadi meliputi biaya bahan langsung, tenaga kerja langsung, dan alokasi biaya overhead pabrik yang berkaitan dengan manufaktur.

d) Persediaan perlengkapan manufaktur

Barang-barang seperti minyak pelumas untuk mesinmesin, bahan pembersih, dan barang lainnya yang merupakan bagian yang kurang penting dari produk jadi.

3) Persediaan rupa-rupa

Barang-barang seperti perlengkapan kantor, kebersihan, dan pengiriman. Persediaan jenis ini biasanya digunakan segera dan biasanya dicatat sebagai beban penjualan atau umum (*selling or general expenses*) ketika dibeli

2.3 Biaya Pengelolaan Persediaan

Kegiatan pengelolaan persediaan memerlukan beberapa hal yang perlu dipertimbangkan dalam pengambilan keputusan. Salah satu pertimbangan itu adalah biaya dalam pengelolaan persediaan. Terdapat tiga jenis biaya yang berkaitan dengan persediaan yang harus dipertimbangkan dalam pengelolaan persediaan (Sartono, 2001), yaitu :

a. Biaya Pesan (*Ordering Cost*)

Biaya pesan merupakan semua biaya yang timbul sebagai akibat pemesanan. Biaya itu meliputi biaya sejak dilakukan pemesanan hingga pesanan itu sampai di gudang, biaya tersebut seperti biaya persiapan, penerimaan, pengecekan, penimbangan dan biaya lainnya hingga persediaan siap untuk diproses.

b. Biaya Simpan (*Carrying Cost*)

Biaya simpan mencakup semua biaya yang dikeluarkan untuk menyimpan persediaan selama periode tertentu. Komponen biaya

simpan adalah *storage costs* yang termasuk sewa gudang, biaya keusangan yakni penurunan nilai persediaan termasuk keusangan teknologi, juga penurunan karena perubahan bentuk fisik persediaan itu sendiri asuransi baik asuransi kebakaran maupun asuransi kehilangan, pajak, biaya dana yang diinvestasikan pada persediaan.

c. Biaya Kehabisan Bahan (*Stockout Costs*)

Biaya Kehabisan Bahan, timbul pada saat perusahaan tidak dapat memenuhi permintaan karena persediaan yang tidak cukup. Biaya kehabisan bahan ini meliputi biaya pesan secara cepat atau khusus dan biaya produksi karena adanya operasi ekstra.

2.4 Pengendalian Persediaan

Masalah pengendalian persediaan merupakan salah satu masalah penting yang dihadapi oleh perusahaan. Kekurangan bahan baku akan mengakibatkan adanya hambatan-hambatan pada proses produksi. Kekurangan persediaan barang jadi di pasaran akan menimbulkan kekecewaan pada pelanggan dan akan mengakibatkan perusahaan kehilangan pelanggan, sedangkan kelebihan persediaan akan menimbulkan biaya ekstra (biaya penyimpanan dan lain-lain), di samping resiko kerusakan karena penyimpanan barang yang terlalu lama. Sehingga dapat dikatakan bahwa pengendalian persediaan yang efektif sangat diperlukan oleh suatu perusahaan. Oleh karena itu pengendalian persediaan pada hakikatnya mencakup dua fungsi yang berhubungan sangat erat yaitu:

a. Perencanaan persediaan

Aspek perencanaan harus dapat menjawab pertanyaan tentang apa yang akan disediakan atau diproduksi dan sumber terbaik pengadaan barang-barang.

b. Pengawasan persediaan

Aspek pengawasan yaitu:

1. Bilamana dan berapa kali pesanan atau produksi dilaksanakan.
2. Berapa banyak pesanan atau produksi tersebut.

2.5 Bahan Baku

Secara umum dari istilah bahan mentah dapat mempunyai arti sebagai sebuah bahan dasar yang bisa berasal dari berbagai tempat, yang mana bahan tersebut dapat digunakan untuk diolah dengan suatu proses tertentu ke dalam bentuk lain yang berbeda wujud dari bentuk aslinya. Sedangkan pengertian secara umum mengenai bahan baku merupakan bahan mentah yang menjadi dasar pembuatan suatu produk yang mana bahan tersebut dapat diolah melalui proses tertentu untuk dijadikan wujud yang lain. Berdasarkan dari pengertian antara bahan mentah dan bahan baku di atas terdapat beberapa contoh wujud dari istilah bahan mentah beberapa di antaranya adalah bijih perak, yang mempunyai arti penting didalam industri pembuatan perak, contoh yang lainnya adalah gandum yang mana biji dari tumbuhan tersebut bila dikeringkan dan di olah dapat menghasilkan tepung yang mana biji gandum ini sangat berguna bagi industri penghasil tepung. Sedangkan berdasarkan pengertian umum contoh wujud dari istilah bahan baku diantaranya adalah, perak dan kompor yang berguna sebagai bahan dasar bagi industri penghasil kerajinan dari perak, seperti anting – anting, kalung dan bingkai foto yang bisa di gunakan untuk hiasan dinding, contoh yang lainnya adalah tepung, margarin, telur, dan gula, yang sangat berguna sebagai bahan baku pembuatan roti, karena sifat dari tepung yang bisa mengikat emulsi dari udara yang terkandung di dalam adonan roti setelah adonan dasar di mixer, apabila adonan tersebut tidak di beri tepung, maka rasanya akan lebih berat dan eneg serta tidak bisa tahan lama, tetapi berbeda bila di kasih dengan campuran tepung, roti yang dihasilkan akan lebih terasa kenyal dan mempunyai bentuk yang lebih bagus dan bisa lebih lama disimpan. Setelah mengetahui pengertian bahan baku dan bahan mentah secara umum, maka terdapat pula pengertian bahan baku maupun bahan mentah menurut pendapat para ahli beserta pembagiannya. Pengertian dari bahan baku menurut Mulyadi, bahan baku adalah bahan yang membentuk bagian integral produk jadi. Sedangkan bahan baku yang di peroleh dapat berasal dari pembelian lokal, pembelian import, atau bisa juga berasal dari pengolahan sendiri

2.7 Metode POQ

Period Order Quantity (POQ) merupakan pendekatan menggunakan konsep jumlah pemesanan ekonomis agar dapat dipakai pada periode bersifat permintaan diskrit atau beragam, teknik ini dilandasi oleh metode EOQ. Dengan mengambil dasar perhitungan pada metode pesanan ekonomis maka akan diperoleh besarnya jumlah pesanan yang harus dilakukan dan interval periode pemesanannya dalam satu periode. Model ini dapat diterapkan ketika persediaan secara terus menerus mengalir atau terbentuk sepanjang suatu periode waktu setelah dilakukan pemesanan. POQ menghitung interval pemesanan yang optimal dengan menggunakan data bulan sebelumnya, serta dalam satu bulan diasumsikan menjadi 4 minggu. Dalam perhitungan, dapat diketahui kuantitas pemesanan yang ekonomis dengan satuan Kg serta interval pemesanan tetap atau jumlah interval pemesanan tetap dengan bilangan bulat (*integer*).

$$POQ = \frac{EOQ}{R} \dots\dots\dots (1)$$

dimana :

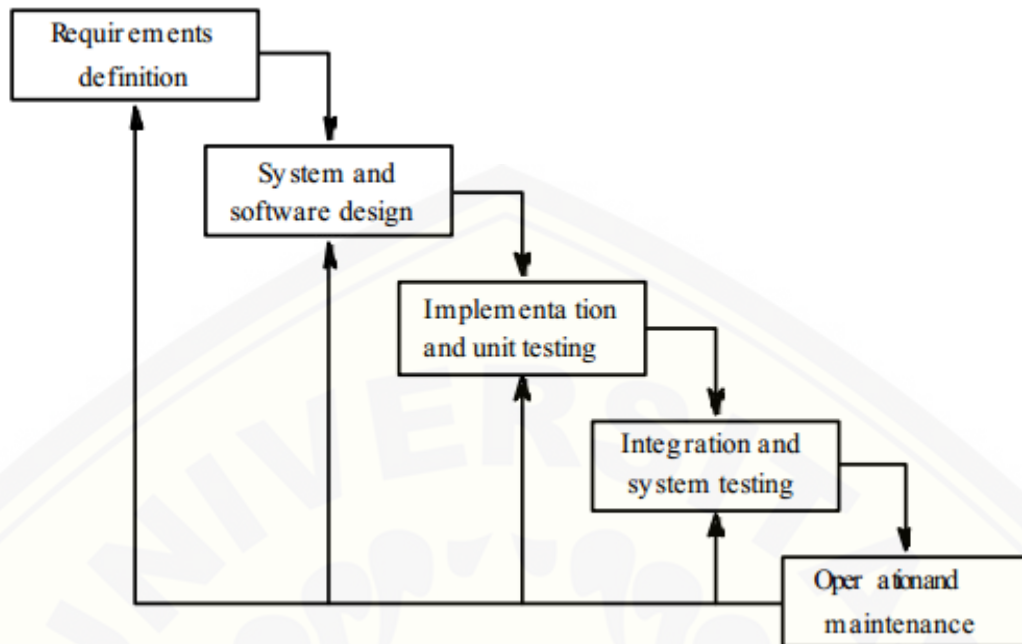
POQ = interval pemesanan ekonomis dalam satu periode

EOQ = kuantitas persediaan optimal

R = rata-rata pemakaian per minggu

2.8 Metode Pengembangan Sistem

Model *waterfall* merupakan metode yang sistematis dan sekuensial yang dimulai pada tingkat dan kemajuan sistem. Dengan mengikuti tahapan *software development life cycle* (SDLC) *waterfall*. Tahapan SDLC dengan metode *waterfall* meliputi tahapan perencanaan, analisis, desain, implementasi, pengujian, dan pemeliharaan. Rincian tahapan SDLC dengan metode *waterfall* dilakukan seperti pada Gambar 2.1



Gambar 2.1 Model *Waterfall* (Sommerville, 2001)

Keterangan dari skema di atas adalah :

1. Analisis Kebutuhan

Menganalisis kebutuhan yang akan digunakan dalam pembuatan aplikasi. Meliputi pengumpulan data kebutuhan fungsional dan non-fungsional dari aplikasi yang akan kita bangun. Setelah itu, menentukan fungsi dan fasilitas apa saja yang akan dibuat dalam aplikasi

2. Desain Aplikasi

Proses pendesainan Aplikasi dari aplikasi yang akan kita bangun yaitu dengan menggunakan *Unified Modeling Language* (UML). Penggunaan UML karena sudah menggunakan konsep *Object Oriented Design* yang tentunya akan sangat memudahkan peneliti untuk membangun Aplikasi optimalisasi bahan baku kain ini. Dalam UML ada beberapa diagram yang akan dibuat antara lain:

a. *Business Process*

Business Proses untuk menggambarkan inputan data yang dibutuhkan Aplikasi, output dari Aplikasi serta tujuan dari pembuatan Aplikasi.

b. *Use Case Diagram*

Use case diagram adalah rangkaian/uraian sekelompok yang saling terkait dan membentuk Aplikasi secara teratur yang dilakukan atau diawasi oleh sebuah aktor.

c. *Activity Diagram*

Activity diagram adalah alir aktivitas dalam Aplikasi yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.

d. *Sequence Diagram*

Sequence diagram memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam Aplikasi yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, display, dan sebagainya berupa pesan/message.

e. *Class Diagram*

Class Diagram adalah sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek.

f. *Entity Relationship Diagram*

ERD adalah suatu model untuk menjelaskan hubungan antara data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.

3. *Coding* (Pengkodean)

Setelah proses desain Aplikasi dikerjakan, proses selanjutnya adalah *coding* atau penulisan kode program. Bahasa pemrograman yang dipakai adalah php dengan *tool* yang digunakan *Sublime text*. Proses pengkodean menggunakan *framework CodeIgniter* (CI), CI dipilih karena mendukung penggunaan konsep OOP (*Object Oriented Programming*). Peneliti akan lebih mudah dalam melakukan *coding*, karena konsep OOP mengikuti model yang telah ada dalam kehidupan nyata. Dalam OOP, semua bagian (*entity*) dari suatu permasalahan adalah objek. Objek-objek ini kemudian juga dapat berupa gabungan dari beberapa objek yang lebih kecil. Objek besar dapat dibentuk dengan menggabungkan beberapa objek-objek dalam

bahasa pemrograman. Objek-objek tersebut berkomunikasi dengan saling interaksi kepada objek lain.

4. Pengujian/ Testing

Pengujian wajib dilakukan untuk menguji apakah Aplikasi ini sudah sesuai dengan kebutuhan dari user atau belum. Dan apakah masih ada kesalahan maupun kelemahan terhadap Aplikasi yang kami bangun tersebut. Diharapkan proses pengujian / testing dapat menyempurnakan Aplikasi yang kami buat. Pengujian yang dilakukan melibatkan semua aspek Aplikasi meliputi *hardware*, *software* aplikasi, *environment software*, penempatan aplikasi, dan *user* yang menggunakan aplikasi ini.

a) Pengujian *White Box*

White box testing adalah cara pengujian dengan melihat ke dalam modul untuk meneliti kode-kode program yang ada, dan menganalisis apakah ada kesalahan atau tidak. Teknik pengujian menggunakan:

1. Kompleksitas Siklomatik

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

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

Keterangan :

$V(G)$: Kompleksitas siklomatik

E : Jumlah *edge*

N : Jumlah *node*

Berdasarkan grafik alir yang ada pada tahapan kedua diketahui jumlah *edge* adalah 5 dan jumlah *node* adalah 5, sehingga dapat dihitung kompleksitas siklomatiknya $V(G) = E -$

$N + 2 = 5 - 5 + 2 = 2$. Jadi jumlah jalur independen adalah 2 jalur.

2. Jalur Program Independen

Jalur independen adalah setiap jalur yang melalui program yang memperkenalkan setidaknya satu kumpulan pernyataan-pernyataan pemrosesan atau kondisi baru. Bila dinyatakan dalam grafik alir, jalur independen harus bergerak setidaknya sepanjang satu *edge* yang belum dilintasi sebelum jalur tersebut didefinisi. Dari perhitungan kompleksitas siklomatik, basis set yang dihasilkan dari jalur independen secara linier adalah 2 jalur, yaitu:

Jalur 1 : 1-2-3-5

Jalur 2 : 1-2-4-5

b) Pengujian Kotak Hitam (Black Box Testing)

Black Box Testing adalah metode pengujian perangkat lunak yang memeriksa fungsionalitas dari Aplikasi yang berkaitan dengan struktur internal atau kerja. Pada pengujian *black box* ini, Aplikasi yang dibangun akan diuji dengan mengujikan langsung *running* Aplikasi dan melakukan kegiatan pengujian dengan menganalisis proses masukan dan keluaran yang dihasilkan oleh Aplikasi. Dalam pengujian *black box* juga dilakukan pengujian dengan cara memasukkan data normal dan data salah, dari masukan data inilah yang nantinya akan dilakukan analisis terhadap reaksi yang muncul pada Aplikasi.

BAB 3. METODOLOGI PENELITIAN

Penelitian yang dilakukan dalam pembuatan aplikasi ini terdiri dari beberapa tahapan. Metodologi penelitian menggambarkan bagaimana langkah-langkah yang akan dilakukan untuk membangun sebuah aplikasi optimalisasi persediaan bahan baku kain sehingga dapat mencapai tujuan sebenarnya dari penelitian.

3.1 Jenis Penelitian

Peneliti menggunakan metode penelitian kualitatif dan kuantitatif. Untuk metode kualitatif dalam penelitian ini meliputi tahapan penemuan masalah yang akan diteliti kemudian mengkaji studi literatur yang berkaitan dengan cara untuk menyelesaikan masalah yang ada dan wawancara. Untuk metode kuantitatif dalam penelitian ini yaitu pada tahapan mengolah data yang telah didapatkan dalam tahapan wawancara.

3.2 Metode Penelitian

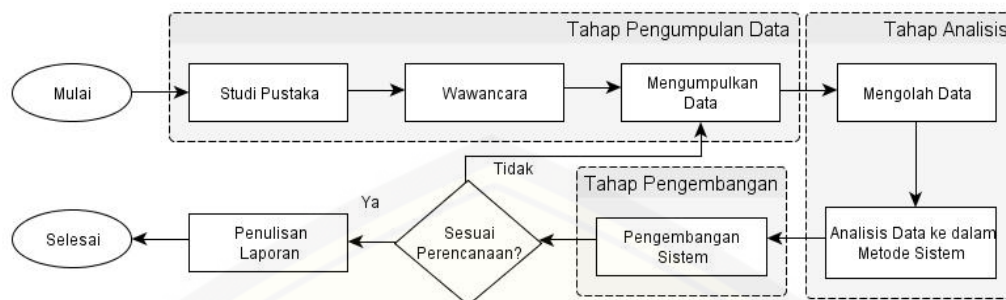
Metode penelitian yang digunakan pada penelitian ini adalah penelitian survei. Penelitian ini merupakan penelitian survei karena informasi dikumpulkan melalui pengajuan pertanyaan

3.3 Tempat dan Waktu Penelitian

Penelitian dilakukan di UD. Dwidaku Jaya yang bertempat di Jember. Waktu penelitian dilakukan selama 4 bulan, dimulai bulan maret 2015 sampai dengan bulan juni 2015. Pada penelitian ini penulis mengambil data mengenai pengendalian persediaan perusahaan.

3.4 Tahapan Penelitian

Penelitian ini dilakukan dalam beberapa tahapan diantaranya tahap pengumpulan data, tahap analisis, dan tahap pengembangan Aplikasi. Seperti yang digambarkan pada diagram alir pada Gambar 3.1



Gambar 3.1 Diagram Alir Tahapan Penelitian

Tahapan awal dari penelitian ini adalah tahap pengumpulan data. Tahap pengumpulan data, peneliti melakukan pengumpulan data dengan melakukan studi pustaka yang terkait dengan permasalahan yang ada. Peneliti melakukan wawancara kepada pihak-pihak yang terkait dalam hal ini adalah manajemen perusahaan untuk mendapatkan data-data yang diperlukan oleh peneliti dalam pembuatan Aplikasi informasi. Tahap selanjutnya adalah tahap analisis. Tahap analisis dimulai dari mengolah data yang telah diperoleh dalam tahap pengumpulan data secara keseluruhan lalu peneliti menganalisis data ke dalam metode Aplikasi. Tahap ketiga adalah tahap pengembangan, dimulai dari menganalisis kebutuhan Aplikasi sampai melakukan pengujian pada Aplikasi. Tahap terakhir adalah memeriksa apakah pengujian pada Aplikasi sesuai dengan perencanaan atau tidak. Penelitian akan kembali ke tahap pengumpulan data apabila dalam pengujian Aplikasi hasilnya tidak sesuai dengan yang direncanakan. Penelitian akan berlanjut ke penulisan laporan apabila hasil dari pengujian sesuai rencana dan penelitian telah selesai. Berikut penjelasan lebih detail dari tahapan penelitian.

3.4.1 Tahap Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh data atau informasi yang dibutuhkan oleh peneliti untuk mencapai tujuan penelitian. Tahap pengumpulan data dilakukan dengan beberapa cara yaitu:

1. *Studi Literatur*

Pada tahap ini dilakukan pengumpulan data dan informasi yang diperlukan untuk proses perancangan Aplikasi. Data dan informasi dapat diperoleh dari lokasi penelitian yaitu di UD.Dwidaku Jaya. Selain itu, studi literatur

juga dapat diperoleh dari *paper*, jurnal ilmiah, serta buku-buku referensi yang berkaitan dengan penelitian.

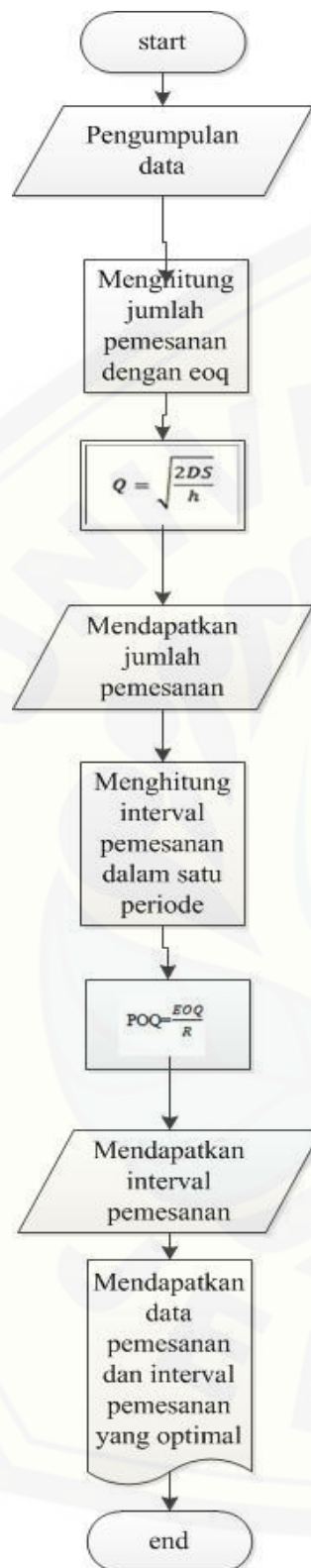
2. Wawancara

Wawancara merupakan salah satu cara untuk mendapatkan data dan informasi yang tepat. Dalam penelitian ini penulis melakukan wawancara secara langsung pada perusahaan UD.Dwidaku Jaya yang ada di kabupaten jember

3.4.2 Tahap Analisis

Tahap analisis dimulai dengan menelaah data secara keseluruhan dari tahap pengumpulan data, dari studi literatur dan wawancara. Pada tahap ini penulis melakukan pengumpulan data-data yang diperlukan kemudian peneliti menghitung pesanan optimal sehingga didapat kuantitas pemesanan optimal. Pemesanan optimal digunakan untuk menghitung interval pemesanan yang optimal dengan metode *Periode Order Quantity*(POQ) setelah itu di dapat interval pemesanan yang optimal dalam satu periode. Seperti digambarkan pada Gambar

3.2



Gambar 3.2 Analisis Data menggunakan Metode POQ

3.5 Tahap pengembangan aplikasi

Pengembangan aplikasi dilakukan setelah analisis data telah selesai dan digunakan untuk membangun aplikasi. Tahapan pengembangan aplikasi optimalisasi bahan baku kain ini sesuai dengan model *waterfall* yang dimulai dari analisis, desain, kode, *test* dan pemeliharaan.

3.5.1 Analisis Kebutuhan

Pada tahap ini, peneliti mencari permasalahan yang ada untuk dapat dianalisis kebutuhan yang diperlukan, sebagai solusi dari permasalahan yang muncul. Data-data yang telah didapat dari hasil wawancara kepada pemilik UD.Dwidaku Jaya maupun studi literatur kemudian dikelompokkan menjadi kebutuhan fungsional dan non-fungsional aplikasi.

3.5.2 Desain Aplikasi

Setelah menganalisis kebutuhan aplikasi maka proses selanjutnya yaitu tahap pembuatan desain aplikasi optimalisasi bahan baku kain, yang akan menggunakan *Unified Modeling Language (UML)* serta dirancang menggunakan konsep *Object-Oriented Programming (OOP)*. Desain aplikasi digunakan untuk membantu penulis dalam pengkodean. Pemodelan UML yang digunakan antara lain :

1. *Business Process*

Digunakan untuk menggambarkan masukan data, keluaran dari aplikasi dan tujuan dari pembuatan aplikasi. Dirancang sesuai dengan analisis kebutuhan aplikasi

2. *Usecase Diagram*

Digunakan untuk mendeskripsikan hak akses antara aktor dengan aplikasi. Dirancang sesuai dengan hasil wawancara dengan pemilik UD.Dwidaku Jaya

3. *Activity Diagram*

Digunakan untuk menggambarkan *workflow* atau aktivitas dari aplikasi untuk dapat mengetahui alur yang dilakukan aktor serta respon aplikasi sesuai dengan desain interface yang di butuhkan oleh UD.Dwidaku Jaya

4. *Sequence Diagram*

Digunakan untuk menggambarkan interaksi-interaksi antar objek di dalam aplikasi. Dirancang sesuai dengan *activity diagram* aplikasi optimalisasi yang telah dibuat, agar dapat mengetahui *method* yang berjalan ketika terjadi aksi

5. *Class Diagram*

Digunakan untuk menggambarkan struktur dari segi pendefinisian kelas – kelas yang ada pada aplikasi sesuai dengan *sequence* yang telah dibuat.

6. *Entity Relationship Diagram*

Digunakan untuk menggambarkan *database* aplikasi yang dibangun, agar dapat mengetahui tabel *database* yang diperlukan aplikasi sesuai *Bussiness Process* yang telah dibuat.

3.5.3 Pengkodean

Setelah proses desain aplikasi dikerjakan, proses selanjutnya adalah *coding* atau penulisan kode program. Bahasa pemrograman yang dipakai adalah php dengan *codeigniter framework* karena sesuai dengan desain aplikasi yang telah dibuat yaitu dengan menggunakan konsep *Object Oriented Program* (OOP), sedangkan *tools* yang digunakan adalah *sublime text*. Database yang digunakan yaitu *Mysql*

3.5.4 Pengujian

Setelah penulisan program, tahap selanjutnya yaitu pengujian. Pada tahap ini dilakukan ujicoba terhadap aplikasi yang telah dibangun untuk menemukan kesalahan yang mungkin terjadi agar dapat melakukan perbaikan dan menyempurnakan aplikasi. Pengujian aplikasi menggunakan metode *whitebox* dan *blackbox*, dimana metode *whitebox* untuk menguji aplikasi dari segi desain dan kode program. Pengujian ini dilakukan dengan memeriksa logika serta kesalahan yang mungkin terjadi. Metode *Blackbox* untuk menguji aplikasi dari segi spesifikasi fungsional yaitu untuk mengetahui apakah fungsi masukan , keluaran dari aplikasi sesuai dengan kebutuhan. Pengujian *Whitebox* dilakukan oleh penulis dengan menggunakan tehnik *Cyclomatic Complexity* (CC). Sedangkan untuk pengujian *Blackbox* dilakukan oleh manajemen UD.Dwidaku Jaya

BAB 4. PENGEMBANGAN APLIKASI

Bab ini akan membahas tentang pengembangan aplikasi optimalisasi persediaan bahan baku kain pada UD.DWIDAKUJAYA. Tahap pengembangan aplikasi dilaksanakan berdasarkan model *waterfall*, dimulai dari analisis kebutuhan fungsional dan non-fungsional aplikasi, pembuatan desain aplikasi, penulisan kode program dan pengujian aplikasi.

4.1 Statement of Purpose

Aplikasi optimalisasi persediaan bahan baku kain merupakan aplikasi peramalan untuk memberikan informasi tentang jumlah persediaan yang optimal dan interval pemesanan sesuai dengan jumlah pemesanan atau pemakaian bahan baku kain pada UD.DWIDAKUJAYA dalam 1 (satu) periode. Tujuan dari dikembangkannya aplikasi ini adalah untuk membantu manajemen dalam menyediakan persediaan bahan baku kain untuk disiapkan dalam periode berikutnya. Aplikasi ini dapat mengelola data kain, data stock, data desain, dan data pemesanan yang nantinya data – data pendukung tersebut akan digunakan untuk menghitung persediaan optimal dan interval pemesanan yang dilakukan oleh perusahaan.

4.2 Analisis Kebutuhan Aplikasi

Tahap analisis kebutuhan aplikasi merupakan tahapan yang sangat penting dalam pengembangan sebuah aplikasi. Seluruh kebutuhan penggunaan didefinisikan dan diformulasikan pada tahap kebutuhan fungsional dan kebutuhan non-fungsional

4.2.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan hal pokok yang harus dapat dilakukan oleh aplikasi dalam menerima masukan untuk diproses sehingga menghasilkan keluaran. Kebutuhan fungsional berisi proses-proses apa saja yang nantinya dilakukan oleh aplikasi. Kebutuhan fungsional dari aplikasi optimalisasi persediaan bahan baku kain adalah :

1. Aplikasi dapat menyimpan dan mengelola (*create, edit, delete*) data user
2. Aplikasi dapat menyimpan dan mengelola data kain
3. Aplikasi dapat menyimpan dan mengelola data stock
4. Aplikasi dapat menyimpan dan mengelola data pemesanan
5. Aplikasi dapat menyimpan dan mengelola data desain
6. Aplikasi dapat menyimpan dan mengelola data prediksi
7. Aplikasi dapat menganalisis persediaan dan interval pemesanan yang optimal untuk periode berikutnya
8. Aplikasi dapat menampilkan hasil dari analisis persediaan dan interval pemesanan yang optimal untuk periode berikutnya

4.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan hal yang dibutuhkan oleh aplikasi untuk mendukung aktivitas aplikasi sesuai dengan kebutuhan fungsional yang telah disusun. Kebutuhan nonfungsional menitikberatkan pada properti perilaku yang dimiliki oleh aplikasi. Kebutuhan Non-Fungsional dari aplikasi optimalisasi persediaan bahan baku kain adalah :

1. Aplikasi dapat digunakan oleh banyak *user* atau pengguna pada saat bersamaan dan menggunakan komputer yang berbeda
2. Tampilan dan bahasa yang digunakan oleh aplikasi mudah dimengerti dan mudah dioperasikan
3. Aplikasi menggunakan *username* dan *password* untuk mengautentikasi hak akses pengguna dalam aplikasi
4. Aplikasi mudah dioperasikan pada beberapa sistem operasi

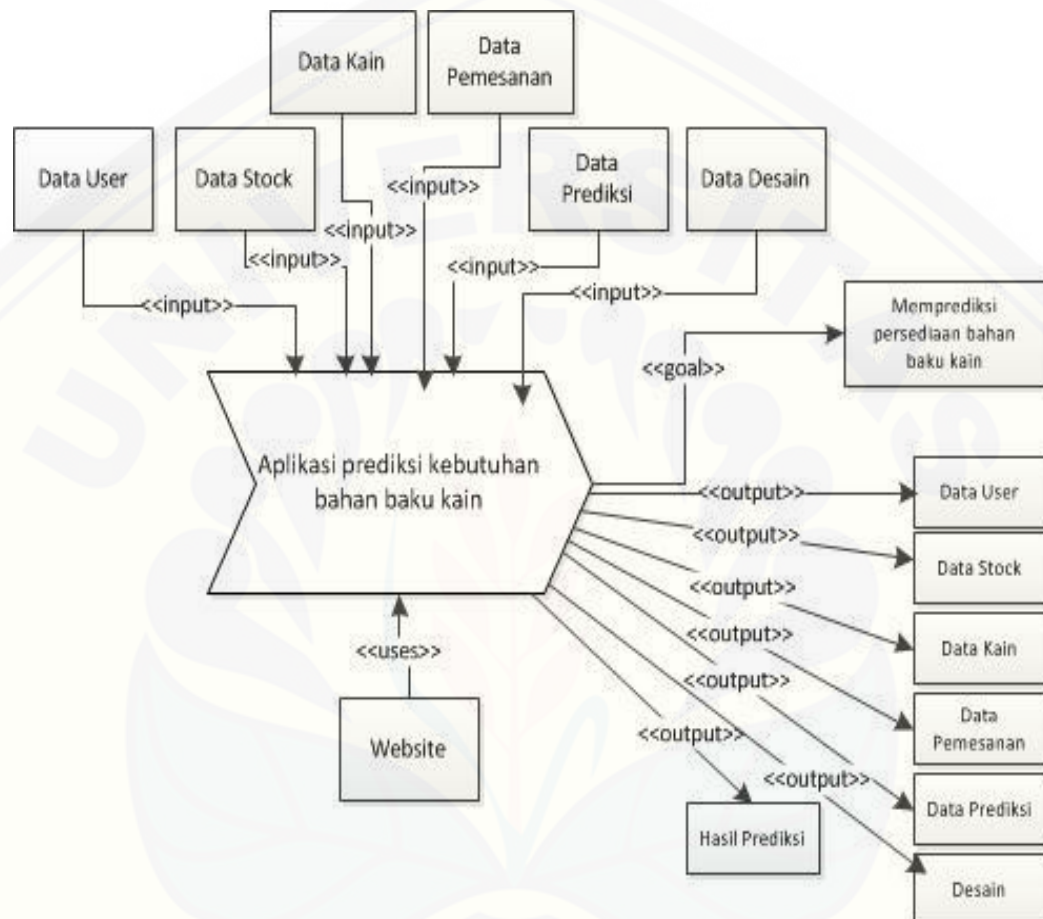
4.3 Desain Aplikasi

Desain aplikasi adalah tahapan untuk memodelkan Aplikasi . Dokumen desain aplikasi yang akan dibuat meliputi *Business Process*, *Usecase Diagram*, *Usecase Skenario*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram* dan *Entity Relationship Diagram* (ERD)

4.3.1 Business Process

Business Process menggambarkan input dan output aplikasi serta mencapai hasil yang diinginkan. Terdapat beberapa komponen untuk mencapai

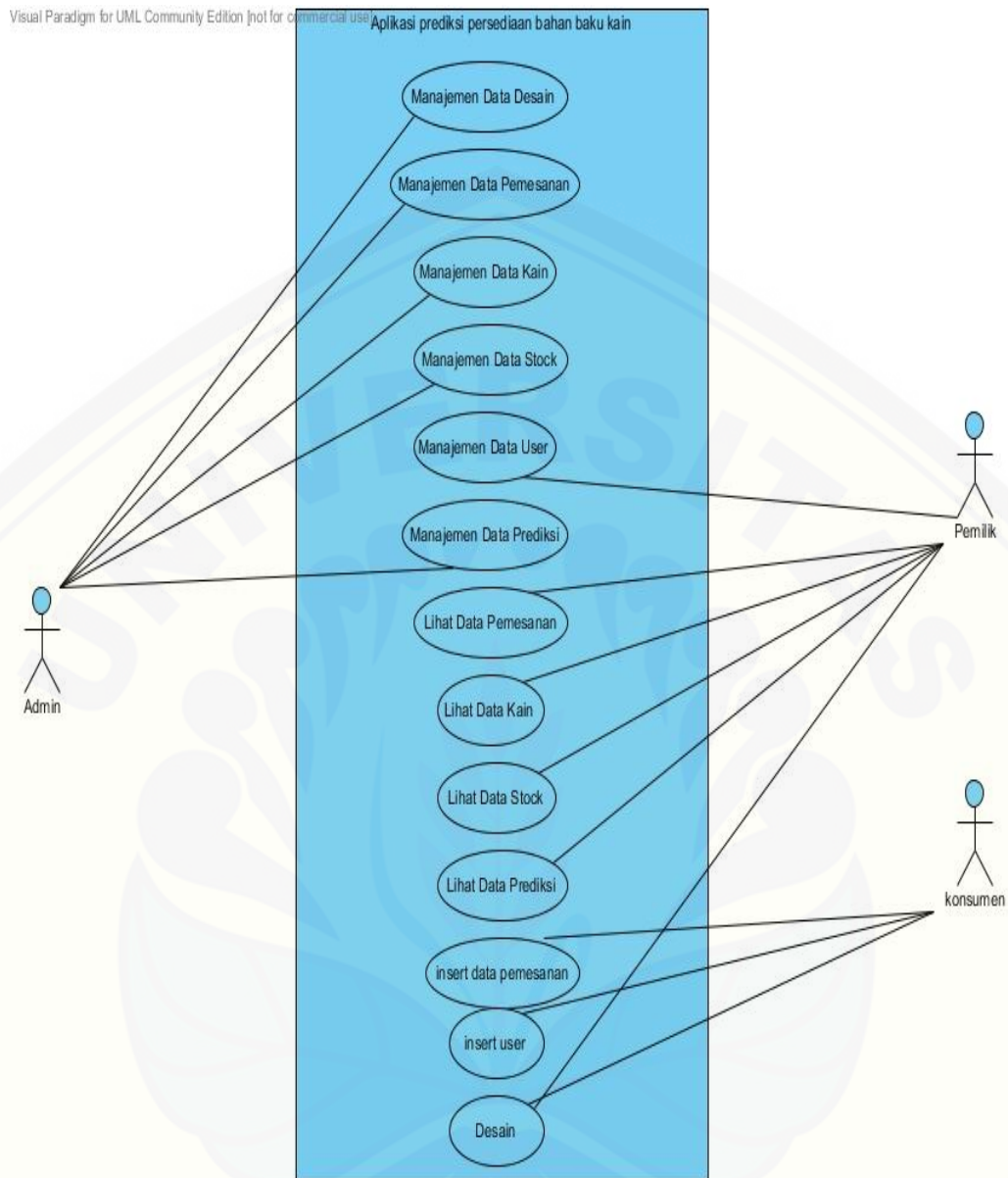
tujuan tersebut, yaitu data yang menjadi masukan (*input*) dan media yang digunakan sehingga akan dihasilkan data keluraan (*output*) untuk melakukan tujuan (*goal*) yang ingin dicapai. *Business Process* Aplikasi dapat dilihat pada gambar 4.1



Gambar 4.1 *Business Process* Aplikasi Optimalisasi persediaan

4.3.2 Usecase Diagram

Usecase Diagram untuk mendokumentasi atau menggambarkan fitur dan aktor yang terdapat pada Aplikasi. *Usecase* Diagram dari Aplikasi optimalisasi persediaan dapat dilihat pada Gambar 4.2



Gambar 4.2 *Usecase* Diagram Aplikasi Optimalisasi Persediaan

Penjelasan tentang definisi aktor dan definisi *usecase* dalam *Usecase* diagram sebagai berikut:

1. Definisi aktor merupakan penjelasan tentang aktor – aktor sebagai pengguna. Terdapat 3 (tiga) aktor dari hasil analisis seperti yang dijelaskan pada tabel 4.1

Tabel 4.1 Definisi aktor

No	Aktor	Deskripsi
1.	Admin	Aktor Admin merupakan aktor yang memiliki hak akses untuk mengatur Aplikasi. Aktor ini dapat melakukan mengelola data kain, mengelola data stock, mengelola data desain, mengelola data pemesanan, dan mengelola data prediksi
2.	User/Pemilik	Aktor <i>User/Pemilik</i> merupakan aktor yang hanya bisa melihat data. Data kain, data stock, data pemesanan, data prediksi, data desain. Aktor <i>User/Pemilik</i> dapat juga mengelola data user
3.	Konsumen	Aktor Konsumen merupakan aktor yang memiliki hak akses input data pemesanan jika sudah terdaftar sebagai pelanggan. Aktor konsumen dapat mendaftar sebagai konsumen agar dapat menginputkan

		pemesanan. Aktor ini juga dapat melihat data desain yang sudah di sediakan oleh perusahaan
--	--	--

2. Definisi *Usecase*

Definisi *Usecase* merupakan penjelasan dari masing – masing *Usecase* atau fitur – fitur pada Aplikasi. Penjelasan *usecase* dapat dilihat pada tabel 4.2

Tabel 4.2 Definisi *Usecase*

No	<i>Usecase</i>	Penjelasan
UCS-01	Manajemen Data Desain	Merupakan <i>usecase</i> yang menggambarkan proses menambah, menampilkan dan menghapus data desain
UCS-02	Manajemen Data Pemesanan	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan, menambah, dan membatalkan data pemesanan
UCS-03	Manajemen Data kain	Merupakan <i>usecase</i> yang menggambarkan proses menambah, menghapus dan menampilkan data kain
UCS-04	Manajemen Data Stock	Merupakan <i>usecase</i> yang menggambarkan proses menambah, menghapus dan menampilkan data stock
UCS-05	Manajemen Data User	Merupakan <i>usecase</i> yang menggambarkan proses menambah, mengedit, menampilkan dan menghapus data user

UCS-06	Manajemen Data Prediksi	Merupakan <i>usecase</i> yang menggambarkan proses menambah, menampilkan, dan menghapus data prediksi
UCS-07	Lihat Data Pemesanan	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan data pemesanan
UCS-08	Lihat Data Kain	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan data kain
UCS-09	Lihat Data Stock	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan data stock
UCS-10	Lihat Data Prediksi	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan data prediksi
UCS-11	Insert Data Pemesanan	Merupakan <i>usecase</i> yang menggambarkan proses menambah data pemesanan yang dilakukan oleh konsumen
UCS-12	Insert Data User	Merupakan <i>usecase</i> yang menggambarkan proses menambah data user yang dilakukan oleh konsumen
UCS-13	Lihat Data Desain	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan data desain

4.3.3 *Usecase* Skenario

Usecase skenario untuk mendokumentasi kebutuhan fungsional Aplikasi. *Usecase* skenario dari Aplikasi dapat dilihat sebagai berikut :

1. *Usecase* Skenario Manajemen Desain

Penjelasan mengenai urutan aktivitas aktor dan reaksi Aplikasi dapat dilihat pada Tabel 4.3

ID : UCS-01
 Name : Manajemen Desain(View)
 Participating Actor : Admin,pemilik,konsumen
 Entry Condition : admin/pemilik/konsumen ingin melihat Desain
 Exit Condition : Admin/pemilik/konsumen dapat melihat Desain

Tabel 4.3 Usecase Skenario manajemen Desain

Skenario Utama Manajemen Desain (View)	
Actor	Aplikasi
1. Klik menu “Data Desain” pada menu utama (home)	
	2. Menampilkan halaman Desain

Name : Manajemen Desain (Insert)
 Participating Actor : Admin
 Entry Condition : admin akan menambah Desain
 Exit Condition: Admin sukses menambah Desain

Skenario Utama Manajemen Desain (Insert)	
Actor	Aplikasi
1. Klik menu “Desain” pada halaman utama (home)	
	2. Menampilkan halaman Desain
3. Klik button Tambah	
	4. Menampilkan form Desain

5. Mengisi form Desain	
6. Klik button submit	
	7. Menyimpan data dalam database
	8. Menampilkan kembali halaman Desain
Skenario Alternatif	
5. Inputan tidak valid	
	6. Menampilkan warning kesalahan

Name : Manajemen Desain (Delete)

Participating Actor : Admin

Entry Condition : admin ingin menghapus Desain

Exit Condition : Admin sukses menghapus Desain

Skenario Utama Manajemen Desain (Delete)	
Actor	Aplikasi
1. Klik menu “Desain” pada halaman utama (home)	
	2. Menampilkan halaman Desain
3. Klik button Delete pada salah satu kolom yang ingin dihapus	
	4. Menampilkan kotak dialog “Anda Yakin Ingin Hapus Data ?”
5. Klik OK	
	6. Menghapus data dalam database

	7. Menampilkan kembali halaman Desain
Skenario Alternatif	
5 Klik Cancel	
	6 Menampilkan kembali halaman Desain

2. Usecase Skenario Manajemen Data Pemesanan

Penjelasan mengenai urutan aktivitas aktor dan reaksi Aplikasi dapat dilihat pada Tabel 4.4

ID : UCS-02
 Name : Manajemen Data Pemesanan(View)
 Participating Actor : Admin,pemilik
 Entry Condition : admin/pemilik ingin melihat data pemesanan
 Exit Condition : Admin/pemilik dapat melihat data pemesanan

Tabel 4.4 Usecase Skenario Manajemen Data Pemesanan

Skenario Utama Manajemen Data Pemesanan (View)	
Actor	Aplikasi
1. Klik menu “Datapemesanan” pada menu utama (home)	
	2. Menampilkan halaman data pemesanan

Name : Manajemen Data Pemesanan (Insert)
 Participating Actor : Admin,konsumen
 Entry Condition : Ketika admin akan menambah data pemesanan
 Exit Condition : Admin sukses menambah data pemesanan

Skenario Utama Manajemen Data Pemesanan (Insert)	
Actor	Aplikasi
1. Klik menu “Data Pemesanan” pada halaman utama (home)	
	2. Menampilkan halaman data pemesanan
3. Klik button Tambah	
	4. Menampilkan form data pemesanan
5. Mengisi form data pemesanan	
6. Klik button submit	
	7. Menyimpan data dalam database
	8. Menampilkan kembali halaman data pemesanan
Skenario Alternatif	
5. Inputan tidak valid	
	6. Menampilkan warning kesalahan

Name : Manajemen Data Pemesanan (Delete/pembatalan)

Participating Actor : Admin

Entry Condition : admin ingin menghapus data pemesanan

Exit Condition : Admin sukses menghapus data pemesanan

Skenario Utama Manajemen Data Pemesanan (Delete)	
Actor	Aplikasi
1. Klik menu “Data Pemesanan” pada halaman utama (home)	
	2. Menampilkan halaman data

	pemesanan
3. Klik button Batal pada salah satu kolom yang ingin dibatalkan	
	4. Menampilkan kotak dialog “Anda Yakin Ingin membatalkan pemesanan ?”
5. Klik iya	
	6. Menghapus data dalam database
	7. Menampilkan kembali halaman data pemesanan
Skenario Alternatif	
5 Klik Cancel	
	6 Menampilkan kembali halaman data pemesanan

3.

4. *Usecase* Skenario Manajemen Data Kain

Penjelasan mengenai urutan aktivitas aktor dan reaksi Aplikasi dapat dilihat pada Tabel 4.5

ID : UCS-03

Name : Manajemen Data Kain(View)

Participating Actor : Admin,pemilik

Entry Condition : admin/pemilik ingin melihat data kain

Exit Condition : Admin/pemilik dapat melihat data kain

Tabel 4.5 Usecase Skenario Manajemen Data kain

Skenario Utama Manajemen Data Kain (View)	
Actor	Aplikasi
1. Klik menu “Datakaint” pada menu utama (home)	

	2. Menampilkan halaman data kain
--	----------------------------------

Name : Manajemen Data Kain (Insert)

Participating Actor : Admin

Entry Condition : admin akan menambah data kain

Exit Condition : Admin sukses menambah data kain

Skenario Utama Manajemen Data Kain (Insert)	
Actor	Aplikasi
1. Klik menu “Data Kain” pada halaman utama (home)	
	2. Menampilkan halaman data kain
3. Klik button Tambah	
	4. Menampilkan form data kain
5. Mengisi form data kain	
6. Klik button submit	
	7. Menyimpan data dalam database
	8. Menampilkan kembali halaman data kain
Skenario Alternatif	
5. Inputan tidak valid	
	6. Menampilkan warning kesalahan

Name : Manajemen Data Kain (edit)

Participating Actor : Admin

Entry Condition : admin ingin mengupdate data kain

Exit Condition : Admin sukses mengupdate data kain

Skenario Utama Manajemen Data Kain (Update)	
Actor	Aplikasi
1. Klik menu “Data Kain” pada halaman utama (home)	
	2. Menampilkan halaman data kain
3. Klik button Edit pada salah satu kolom yang ingin di edit	
	4. Menampilkan form data kain
5. Mengedit form data kain	
6. Klik button submit	
	7. Menyimpan data dalam database
	8. Menampilkan kembali halaman data kain
Skenario Alternatif	
5. Inputan tidak valid	
	6. Menampilkan warning kesalahan
7. Klik button cancel	
	8. Menampilkan halaman data kain

Name : Manajemen Data Kain (Delete)

Participating Actor : Admin

Entry Condition : admin ingin menghapus data kain

Exit Condition : Admin sukses menghapus data kain

Skenario Utama Manajemen Data Kain (Delete)

Actor	Aplikasi
1. Klik menu “Data Kain” pada halaman utama (home)	
	2. Menampilkan halaman data kain
3. Klik button Delete pada salah satu kolom yang ingin dihapus	
	4. Menampilkan kotak dialog “Anda Yakin Ingin Hapus Data ?”
5. Klik OK	
	6. Menghapus data dalam database
	7. Menampilkan kembali halaman data kain
Skenario Alternatif	
5 Klik Cancel	
	6 Menampilkan kembali halaman data kain

4. *Usecase* Skenario Manajemen Data Stock

Penjelasan mengenai urutan aktivitas aktor dan reaksi Aplikasi dapat dilihat pada Tabel 4.6

ID : UCS-04

Name : Manajemen Data Stock(View)

Participating Actor : Admin,pemilik

Entry Condition : admin/pemilik ingin melihat data stock

Exit Condition : Admin/pemilik dapat melihat data stock

Tabel 4.6 Usecase Skenario Manajemen Data Stock

Skenario Utama Manajemen Data Stock (View)	
Actor	Aplikasi
1. Klik menu "Datastock" pada menu utama (home)	
	2. Menampilkan halaman data stock

Name : Manajemen Data Stock (Insert)

Participating Actor : Admin

Entry Condition : Ketika admin akan menambah data stock

Exit Condition : Admin sukses menambah data stock

Skenario Utama Manajemen Data Stock (Insert)	
Actor	Aplikasi
1. Klik menu "Data Stock" pada halaman utama (home)	
	2. Menampilkan halaman data stock
3. Klik button Tambah	
	4. Menampilkan form data stock
5. Mengisi form data stock	
6. Klik button submit	
	7. Menyimpan data dalam database
	8. Menampilkan kembali halaman data stock
Skenario Alternatif	
5. Inputan tidak valid	

	6. Menampilkan warning kesalahan
--	----------------------------------

Name : Manajemen Data Stock (Delete)
 Participating Actor : Admin
 Entry Condition : admin ingin menghapus data stock
 Exit Condition : Admin sukses menghapus data stock

Skenario Utama Manajemen Data Stock (Delete)	
Actor	Aplikasi
1. Klik menu "Data Stock" pada halaman utama (home)	
	2. Menampilkan halaman data stock
3. Klik button Delete pada salah satu kolom yang ingin dihapus	
	4. Menampilkan kotak dialog "Anda Yakin Ingin Hapus Data ?"
5. Klik OK	
	6. Menghapus data dalam database
	7. Menampilkan kembali halaman data stock
Skenario Alternatif	
5. Klik Cancel	
	6. Menampilkan kembali halaman data stock

5. *Usecase* Skenario Manajemen Data User

Penjelasan mengenai urutan aktivitas aktor dan reaksi Aplikasi dapat dilihat pada Tabel 4.7

ID : UCS-05
 Name : Manajemen Data User (View)
 Participating Actor : pemilik
 Entry Condition : pemilik akan melihat data user
 Exit Condition : pemilik dapat melihat data user

Tabel 4.7 Usecase Skenario Manajemen Data User

Skenario Utama Manajemen Data User (View)	
Actor	Aplikasi
1. Klik menu “Tambah User” pada menu utama (home)	
	2. Menampilkan halaman data user

Name : Manajemen Data User (Insert)
 Participating Actor : pemilik
 Entry Condition : pemilik akan menambah data user
 Exit Condition: pemilik sukses menambah data user

Skenario Utama Manajemen Data User (Insert)	
Actor	Aplikasi
1. Klik menu “Tambah User” pada halaman utama (home)	
	2. Menampilkan form data user
3. Mengisi form data user	

4. Klik button submit	
	5. Menyimpan data dalam database
	6. Menampilkan kembali halaman data user
Skenario Alternatif	
5. Inputan tidak valid	
	6. Menampilkan warning kesalahan

Name : Manajemen Data User (Update)

Participating Actor : pemilik

Entry Condition : pemilik ingin mengupdate data user

Exit Condition : pemilik sukses mengupdate data user

Skenario Utama Manajemen Data User (Update)	
Actor	Aplikasi
1. Klik menu “Tambah User” pada halaman utama (home)	
	2. Menampilkan halaman data user
3. Klik button edit pada salah satu kolom yang ingin diupdate	
	4. Menampilkan form data user
5. Mengedit form data user	
6. Klik button submit	
	7. Menyimpan data dalam database
	8. Menampilkan kembali halaman data user

Skenario Alternatif	
5. Inputan tidak valid	
	6. Menampilkan warning kesalahan
7. Klik button cancel	
	8. Menampilkan halaman data user

Name : Manajemen Data User (Delete)

Participating Actor : pemilik

Entry Condition : pemilik ingin menghapus data user

Exit Condition : pemilik sukses menghapus data user

Skenario Utama Manajemen Data User (Delete)	
Actor	Aplikasi
1. Klik menu "Tambah User" pada halaman utama (home)	
	2. Menampilkan halaman data user
3. Klik button Delete pada salah satu kolom yang ingin dihapus	
	4. Menampilkan kotak dialog "Anda Yakin Ingin Hapus Data ?"
5. Klik OK	
	6. Menghapus data dalam database
	7. Menampilkan kembali halaman data user
Skenario Alternatif	

5 Klik Cancel	
	6 Menampilkan kembali halaman data user

6. Usecase Skenario Manajemen Data Prediksi

Penjelasan mengenai urutan aktivitas aktor dan reaksi Aplikasi dapat dilihat pada Tabel 4.8

ID	: UCS-06
Name	: Manajemen Prediksi (View)
Participating Actor	: Admin, pemilik
Entry Condition	: admin/pemilik akan melihat data prediksi
Exit Condition	: Admin/pemilik dapat melihat data prediksi

Tabel 4.8 Usecase Skenario Manajemen Data Prediksi

Skenario Utama Manajemen Data prediksi (View)	
Actor	Aplikasi
1. Klik menu “prediksi” pada menu utama (home)	
	2. Menampilkan halaman data prediksi

Name	: Manajemen Data Prediksi (Insert)
Participating Actor	: Admin
Entry Condition	: admin akan menambah data prediksi
Exit Condition	: Admin sukses menambah data prediksi

Skenario Utama Manajemen Data prediksi (Insert)	
Actor	Aplikasi
1. Klik menu “Prediksi” pada halaman utama (home)	
	2. Menampilkan halaman prediksi
3. Klik button Tambah	
	4. Menampilkan form data prediksi
5. Mengisi form prediksi	
6. Klik button submit	
	7. Menyimpan data dalam database
	8. Menampilkan kembali halaman data prediksi
Skenario Alternatif	
5. Inputan tidak valid	
	6. Menampilkan warning kesalahan

Name : Manajemen Data Prediksi (Delete)

Participating Actor : Admin

Entry Condition : admin ingin menghapus data prediksi

Exit Condition : Admin sukses menghapus data prediksi

Skenario Utama Manajemen Data User (Delete)	
Actor	Aplikasi
1. Klik menu “Prediksi” pada halaman utama (home)	

	2. Menampilkan halaman prediksi
3. Klik button Delete pada salah satu kolom yang ingin dihapus	
	4. Menampilkan kotak dialog “Anda Yakin Ingin Hapus Data ?”
5. Klik OK	
	6. Menghapus data dalam database
	7. Menampilkan kembali halaman data user
Skenario Alternatif	
5 Klik Cancel	
	6 Menampilkan kembali halaman data user

7. Usecase Skenario Insert User

Penjelasan mengenai urutan aktivitas aktor dan reaksi Aplikasi dapat dilihat pada Tabel 4.9

ID : UCS-12
 Name : Insert User
 Participating Actor : konsumen
 Entry Condition : admin akan menambah data user
 Exit Condition : Admin sukses menambah data user

Tabel 4.9 Usecase Skenario Insert User

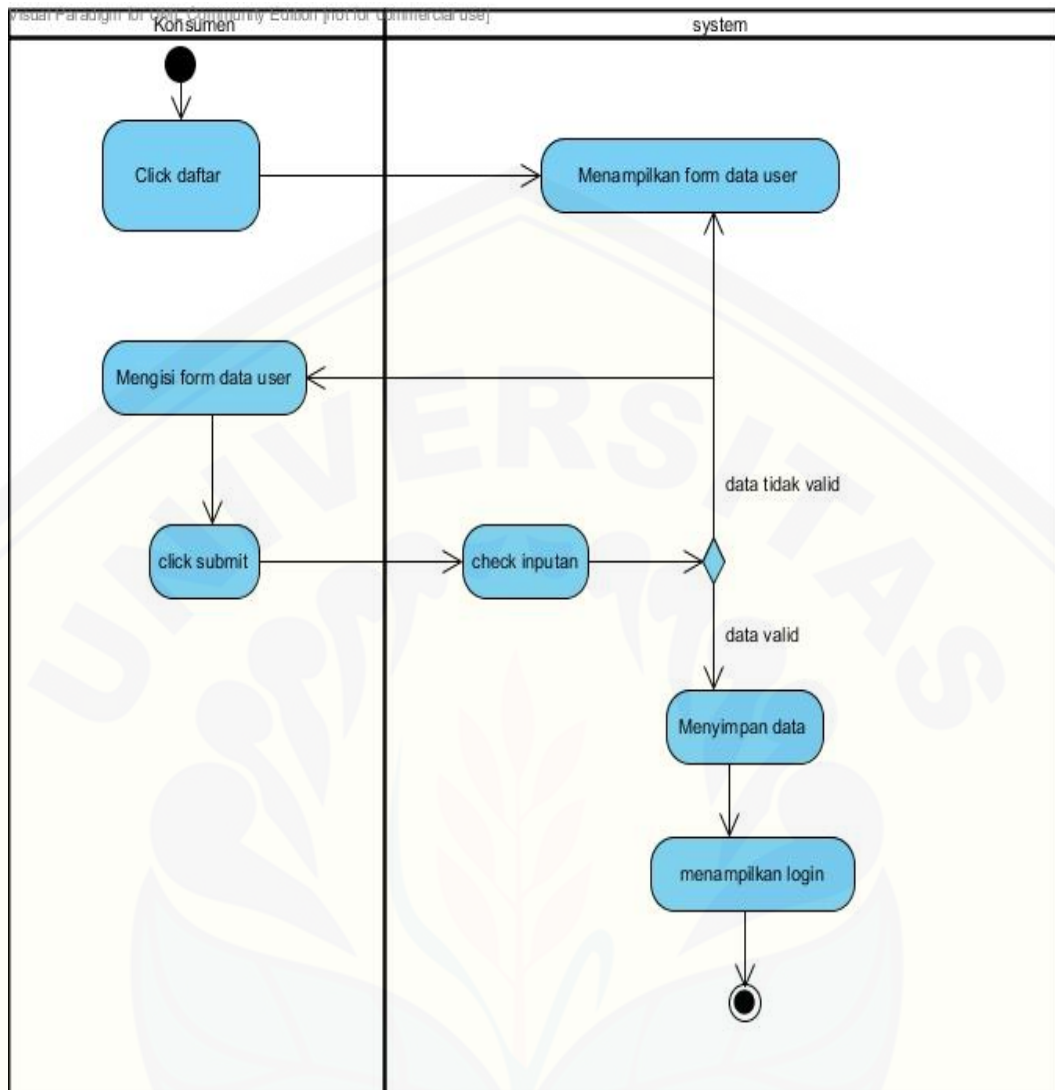
Skenario Utama Manajemen Data User (Insert)	
Actor	Aplikasi
1. Membuka web dwidakujaya	
	2. Menampilkan form login
3. Klik daftar	
	4. menampilkan form user
4. Klik button submit	
	5. Menyimpan data dalam database
	6. Menampilkan kembali halaman login
Skenario Alternatif	
5. Inputan tidak valid	
	6. Menampilkan warning kesalahan

4.3.4 Activity Diagram

Activity Diagram menggambarkan aliran aktivitas dalam Aplikasi optimalisasi persediaan yang akan dibangun

1. *Activity Diagram insert user* konsumen

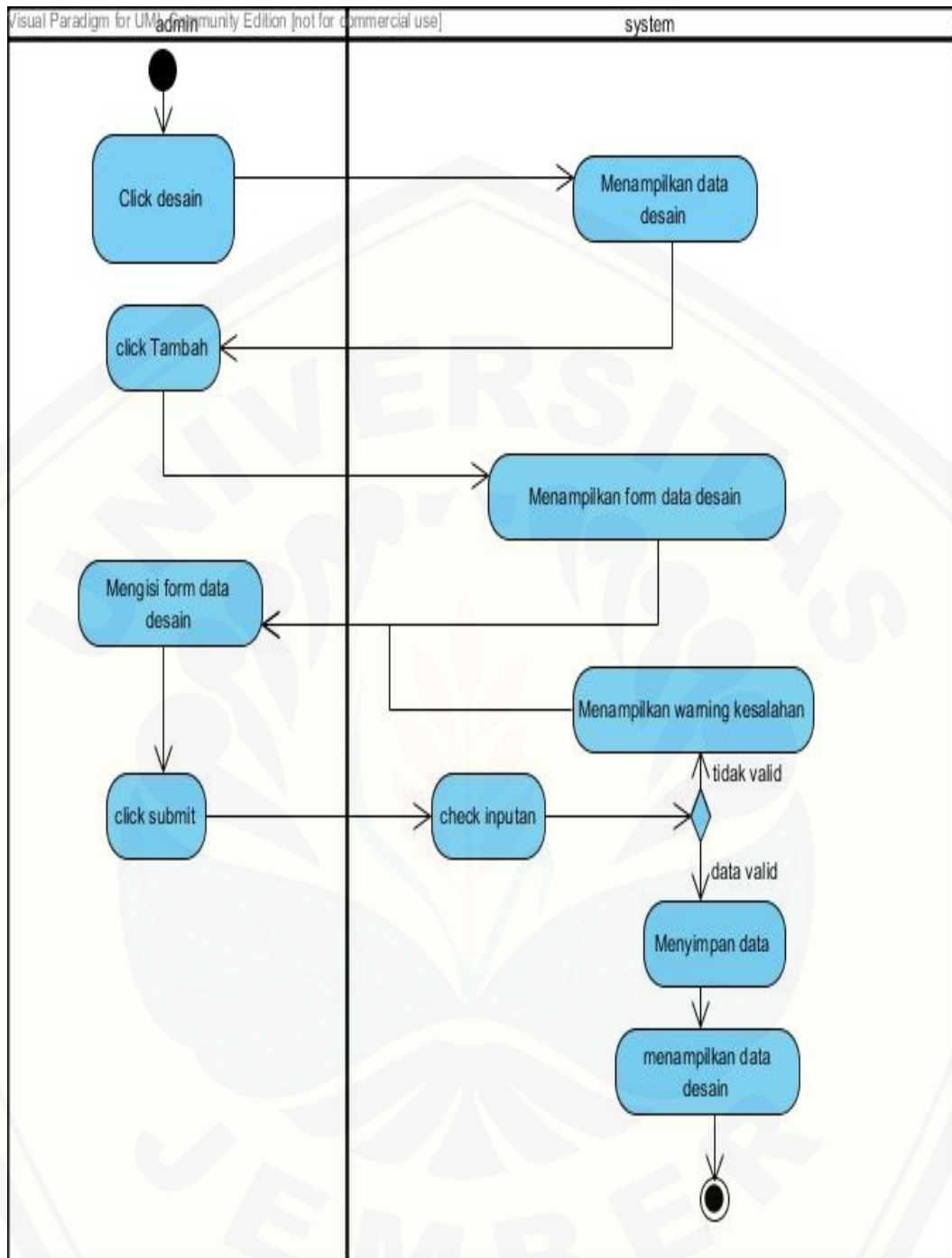
Activity Diagram insert user konsumen menjelaskan aktivitas menambah data *user* yang dilakukan oleh konsumen yang belum terdaftar sebagai pelanggan. Berikut terlihat pada Gambar 4.3



Gambar 4.3 Activity Diagram *insert user* konsumen

2. Activity Diagram *insert data* desain

Activity Diagram *insert data* desain menjelaskan aktivitas menambah data desain yang dilakukan oleh admin. Berikut terlihat pada Gambar 4.4



Gambar 4.4 Activity Diagram *insert* data desain

3. Activity Diagram *insert* data kain

Activity Diagram *insert* data kain menjelaskan aktivitas menambah data kain yang dilakukan oleh admin. Activity Diagram *insert* data kain dapat dilihat pada Lampiran (Lampiran Activity)

4. *Activity Diagram insert data pemesanan*

Activity Diagram insert data pemesanan menjelaskan aktivitas menambah data pemesanan yang dilakukan oleh admin. *Activity Diagram insert data pemesanan* dapat dilihat pada Lampiran (Lampiran *Activity*)

5. *Activity Diagram insert data stock*

Activity Diagram insert data stock menjelaskan aktivitas menambah data *stock* yang dilakukan oleh admin. *Activity Diagram insert Data pemesanan* dapat dilihat pada Lampiran (Lampiran *Activity*)

6. *Activity Diagram insert data user*

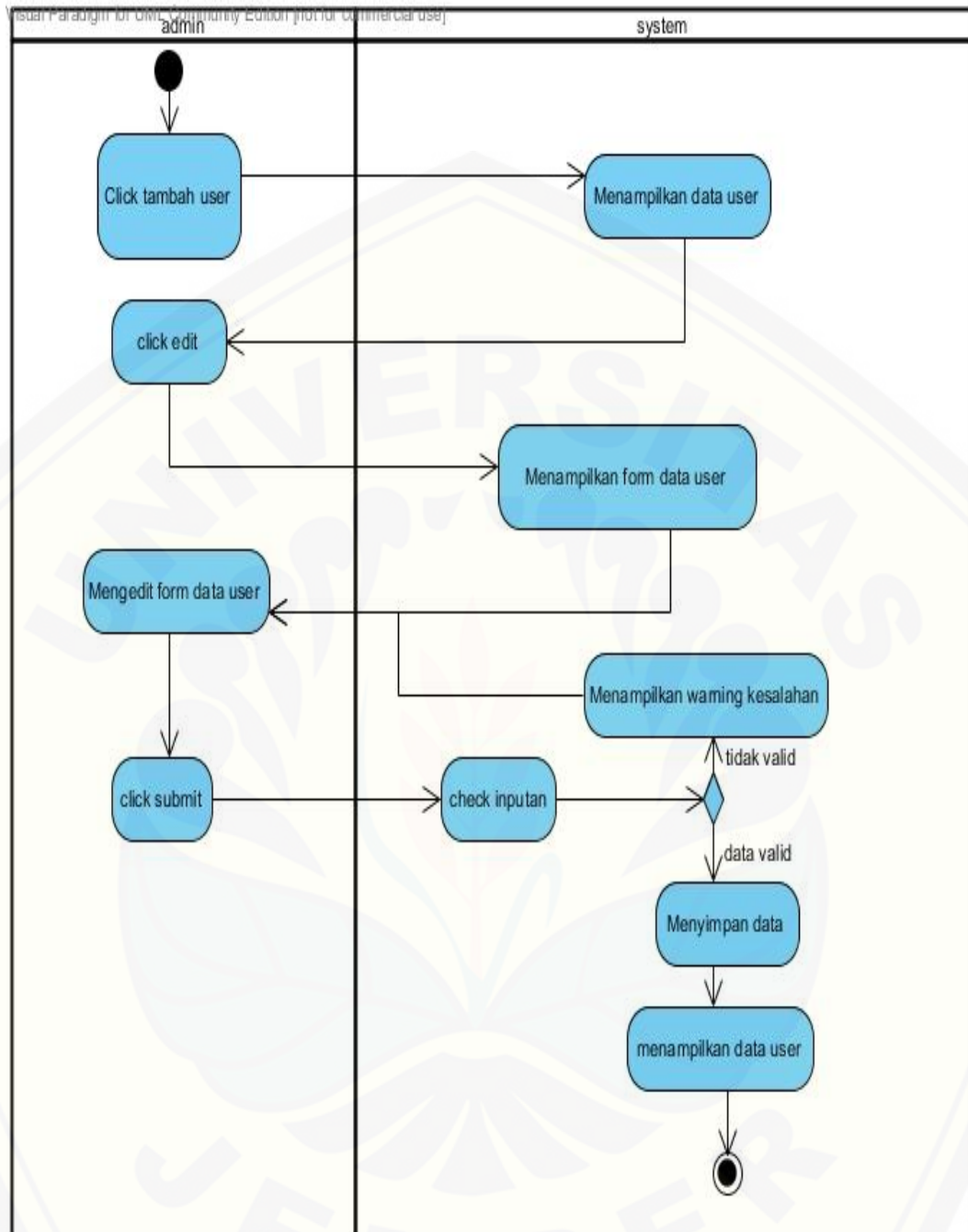
Activity Diagram insert data user menjelaskan aktivitas menambah data *user* yang dilakukan oleh *user/pemilik*. *Activity Diagram insert data user* dapat dilihat pada Lampiran (Lampiran *Activity*)

7. *Activity Diagram insert data prediksi*

Activity Diagram insert data prediksi menjelaskan aktivitas menambah data prediksi yang dilakukan oleh admin. *Activity Diagram insert data prediksi* dapat dilihat pada Lampiran (Lampiran *Activity*)

8. *Activity Diagram edit data user*

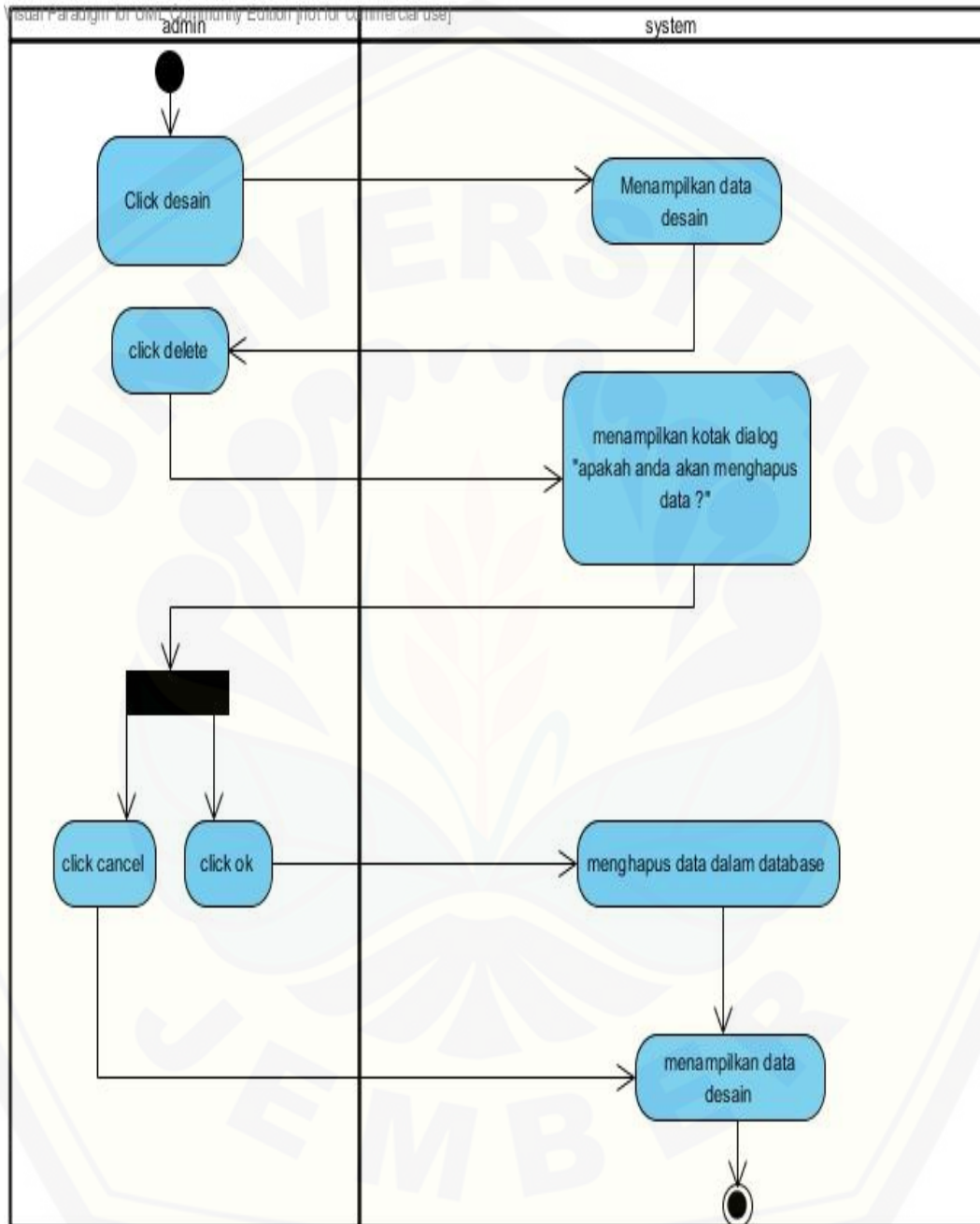
Activity Diagram edit data prediksi menjelaskan aktivitas mengubah data *user* yang dilakukan oleh pemilik. Berikut terlihat pada Gambar 4.5



Gambar 4.5 Activity Diagram *Edit data user*

9. Activity Diagram *Delete* data desain

Activity Diagram *delete* desain menjelaskan aktivitas menghapus data desain yang dilakukan oleh admin. Berikut terlihat pada Gambar 4.6



Gambar 4.6 Activity Diagram *delete* data desain

10. *Activity Diagram Delete data kain*

Activity Diagram delete data kain menjelaskan aktivitas menghapus data kain yang dilakukan oleh admin. *Activity Diagram delete data kain* dapat dilihat pada Lampiran (Lampiran *Activity*)

11. *Activity Diagram Delete data pemesanan*

Activity Diagram delete data kain menjelaskan aktivitas membatalkan data pemesanan yang dilakukan oleh admin. *Activity Diagram delete data pemesanan* dapat dilihat pada Lampiran (Lampiran *Activity*)

12. *Activity Diagram Delete data prediksi*

Activity Diagram delete data prediksi menjelaskan aktivitas menghapus data prediksi yang dilakukan oleh admin. *Activity Diagram delete data prediksi* dapat dilihat pada Lampiran (Lampiran *Activity*)

13. *Activity Diagram Delete data stock*

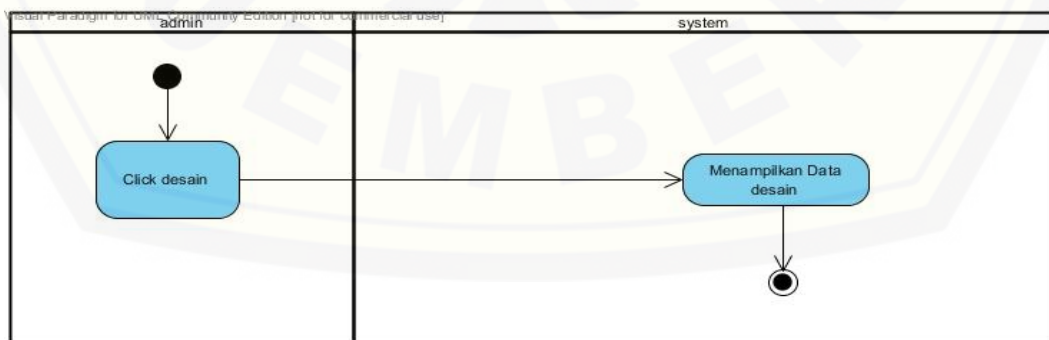
Activity Diagram delete data stock menjelaskan aktivitas menghapus data *stock* yang dilakukan oleh admin. *Activity Diagram delete data stock* dapat dilihat pada Lampiran (Lampiran *Activity*)

14. *Activity Diagram Delete data user*

Activity Diagram delete data user menjelaskan aktivitas menghapus data *user* yang dilakukan oleh admin. *Activity Diagram delete data user* dapat dilihat pada Lampiran (Lampiran *Activity*)

15. *Activity Diagram View data desain*

Activity Diagram view data desain menjelaskan aktivitas menampilkan data desain yang dilakukan oleh admin atau pemilik. Berikut terlihat pada Gambar 4.7



Gambar 4.7 *Activity Diagram view data desain*

16. *Activity Diagram View* data kain

Activity Diagram view data kain menjelaskan aktivitas menampilkan data kain yang dilakukan oleh admin atau pemilik. *Activity Diagram view* data kain dapat dilihat pada Lampiran (Lampiran *Activity*)

17. *Activity Diagram View* data pemesanan

Activity Diagram view data pemesanan menjelaskan aktivitas menampilkan data pemesanan yang dilakukan oleh admin atau pemilik. *Activity Diagram view* data pemesanan dapat dilihat pada Lampiran (Lampiran *Activity*)

18. *Activity Diagram View* data stock

Activity Diagram view data stock menjelaskan aktivitas menampilkan data stock yang dilakukan oleh admin atau pemilik. *Activity Diagram view* data stock dapat dilihat pada Lampiran (Lampiran *Activity*)

19. *Activity Diagram View* data prediksi

Activity Diagram view data prediksi menjelaskan aktivitas menampilkan data prediksi yang dilakukan oleh admin atau pemilik. *Activity Diagram view* data prediksi dapat dilihat pada Lampiran (Lampiran *Activity*)

20. *Activity Diagram View* data user

Activity Diagram view data user menjelaskan aktivitas menampilkan data user yang dilakukan oleh admin atau pemilik. *Activity Diagram view* data user dapat dilihat pada Lampiran (Lampiran *Activity*)

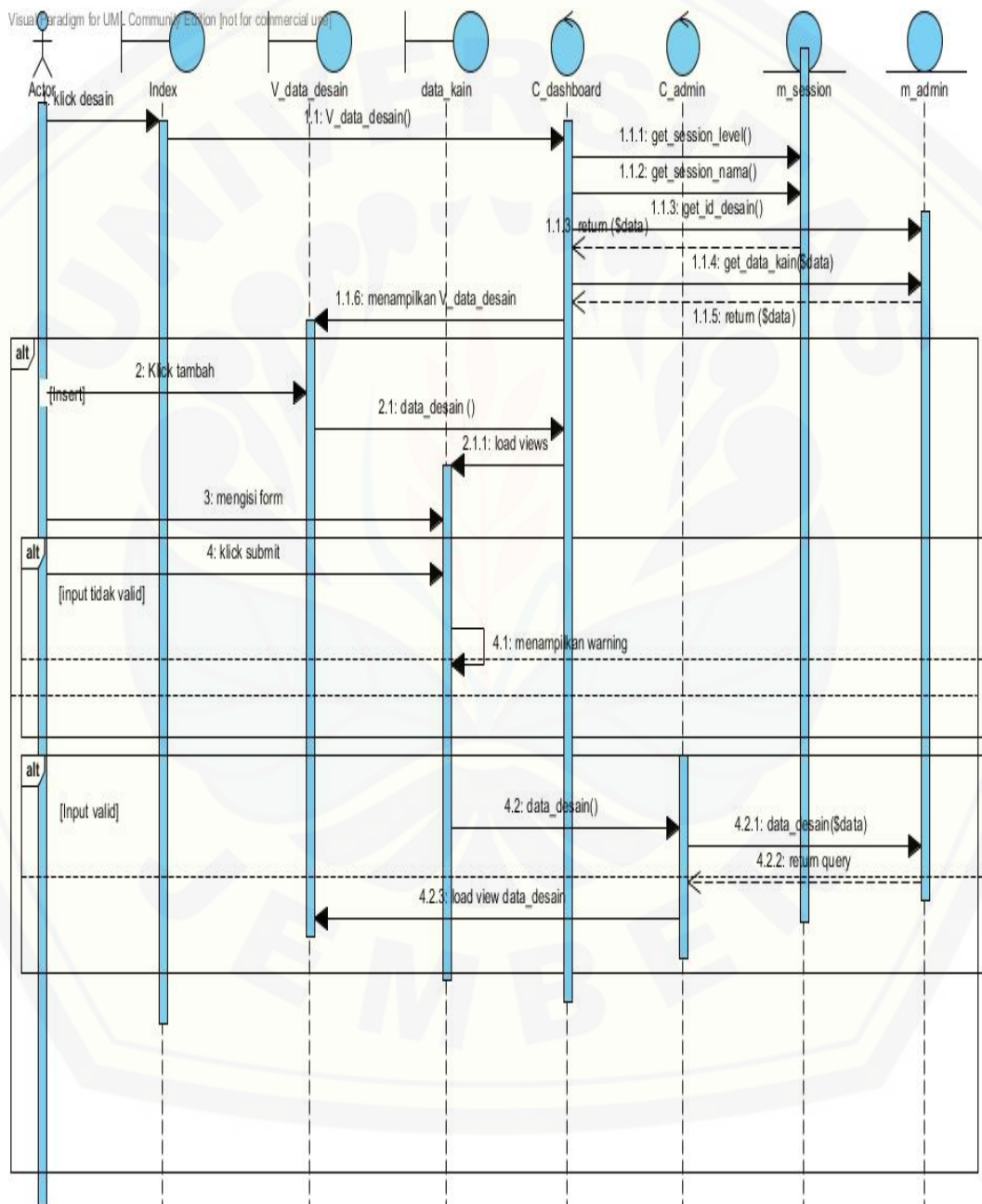
4.3.5 *Sequence Diagram*

Sequence Diagram untuk mendokumentasi suatu diagram terurut yang menampilkan interaksi - interaksi antar objek di dalam Aplikasi. *Sequence diagram* digunakan untuk menggambarkan skenario dan memodelkan aliran logika dalam Aplikasi dengan cara *visual*. *Sequence* dari Aplikasi optimalisasi persediaan adalah sebagai berikut :

1. *Sequence Diagram insert* data desain

Sequence Diagram insert data desain menjelaskan alur perintah atau *method* yang dijalankan pada saat aktor yaitu admin menambah data desain. *Method* atau perintah akan menuju ke *class* yang dituju. *Class* pada *Sequence Diagram insert* data desain ada index, V_data_desain, data_kain,

C_dashboard, C_admin, m_session, dan m_admin. Alur *method* jika aktor menambah data, aktor mengklik menu desain pada index, *method* yang berjalan dari index menuju c_dashboard adalah V_data_desain. *Method* tersebut untuk memanggil data dari database sehingga nantinya akan ditampilkan pada *class* v_data_desain kemudian aktor bisa menambahkan data desain. *Sequence* Diagram *insert* data desain oleh admin dijelaskan pada Gambar 4.8



Gambar 4.8 *Sequence* Diagram *insert* data desain

2. *Sequence Diagram insert data kain*

Sequence Diagram insert data kain menjelaskan alur perintah atau *method* yang dijalankan pada saat aktor yaitu admin menambah data kain. *Method* atau perintah akan menuju ke *class* yang dituju. *Class* pada *Sequence Diagram insert data kain* ada index, V_data_kain, data_kain, C_dashboard, C_admin, m_session, dan m_admin. Alur *method* jika aktor menambah data, aktor mengklik menu data kain pada index, *method* yang berjalan dari index menuju c_dashboard adalah V_data_kain. *Method* tersebut untuk memanggil data dari database sehingga nantinya akan ditampilkan pada *class* v_data_desain kemudian aktor bisa menambahkan data desain. *Sequence Diagram insert data kain* oleh admin dapat dilihat pada Lampiran (Lampiran Sequence)

3. *Sequence Diagram insert data pemesanan*

Sequence Diagram insert data pemesanan menjelaskan alur perintah atau *method* yang dijalankan pada saat aktor yaitu admin menambah data pemesanan. *Method* atau perintah akan menuju ke *class* yang dituju. *Class* pada *Sequence Diagram insert data pemesanan* ada index, V_data_pemesanan, data_pemesanan, C_dashboard, C_admin, m_session, dan m_admin. Alur *method* jika aktor menambah data, aktor mengklik menu data_pemesanan pada index, *method* yang berjalan dari index menuju c_dashboard adalah V_data_pemesanan. *Method* tersebut untuk memanggil data dari database sehingga nantinya akan ditampilkan pada *class* v_data_pemesanan kemudian aktor bisa menambahkan data pemesanan. *Sequence Diagram insert data pemesanan* oleh admin dapat dilihat pada Lampiran (Lampiran Sequence)

4. *Sequence Diagram insert data stock*

Sequence Diagram insert data stock menjelaskan alur perintah atau *method* yang dijalankan pada saat aktor yaitu admin menambah data stock. *Method* atau perintah akan menuju ke *class* yang dituju. *Class* pada *Sequence Diagram insert data stock* ada index, V_data_stock, data_stock, C_dashboard, C_admin, m_session, dan m_admin. Alur *method* jika aktor menambah data, aktor mengklik menu data_stock pada index, *method* yang berjalan dari index menuju c_dashboard adalah V_data_stock. *Method* tersebut untuk memanggil data dari

database sehingga nantinya akan ditampilkan pada *class v_data_stock* kemudian aktor bisa menambahkan data *stock*. *Sequence Diagram insert data stock* oleh admin dapat dilihat pada Lampiran (Lampiran Sequence)

5. *Sequence Diagram insert data prediksi*

Sequence Diagram insert data prediksi menjelaskan alur perintah atau *method* yang dijalankan pada saat aktor yaitu admin menambah data prediksi. *Method* atau perintah akan menuju ke *class* yang dituju. *Class* pada *Sequence Diagram insert data prediksi* ada index, *V_data_prediksi*, *data_prediksi*, *C_dashboard*, *C_admin*, *m_session*, dan *m_admin*. Alur *method* jika aktor menambah data, aktor mengklik menu *data_prediksi* pada index, *method* yang berjalan dari index menuju *c_dashboard* adalah *V_data_prediksi*. *Method* tersebut untuk memanggil data dari database sehingga nantinya akan ditampilkan pada *class v_data_prediksi* kemudian aktor bisa menambahkan data prediksi. *Sequence Diagram insert data prediksi* oleh admin dapat dilihat pada Lampiran (Lampiran Sequence)

6. *Sequence Diagram insert data pemesanan user konsumen*

Sequence Diagram insert data pemesanan user konsumen menjelaskan alur perintah atau *method* yang dijalankan pada saat aktor yaitu konsumen menambah data pemesanan *user* konsumen. *Method* atau perintah akan menuju ke *class* yang dituju. *Class* pada *Sequence Diagram insert data pemesanan user konsumen* ada index *user* konsumen, *V_data_pemesanan_user_konsumen*, *data_pemesanan_user_konsumen*, *C_dashboard*, *C_admin*, *m_session*, dan *m_admin*. Alur *method* jika aktor menambah data, aktor mengklik menu *data_pemesanan* pada *index_user_konsumen*, *method* yang berjalan dari index menuju *c_dashboard* adalah *data_pemesanan_user_konsumen*. *Method* tersebut untuk memanggil form sehingga nantinya aktor bisa menambahkan data pemesanan. *Sequence Diagram insert data pemesanan* oleh admin dapat dilihat pada Lampiran (Lampiran Sequence)

7. *Sequence Diagram insert data user konsumen*

Sequence Diagram insert data user konsumen menjelaskan alur perintah atau *method* yang dijalankan pada saat aktor yaitu konsumen menambah data *user*

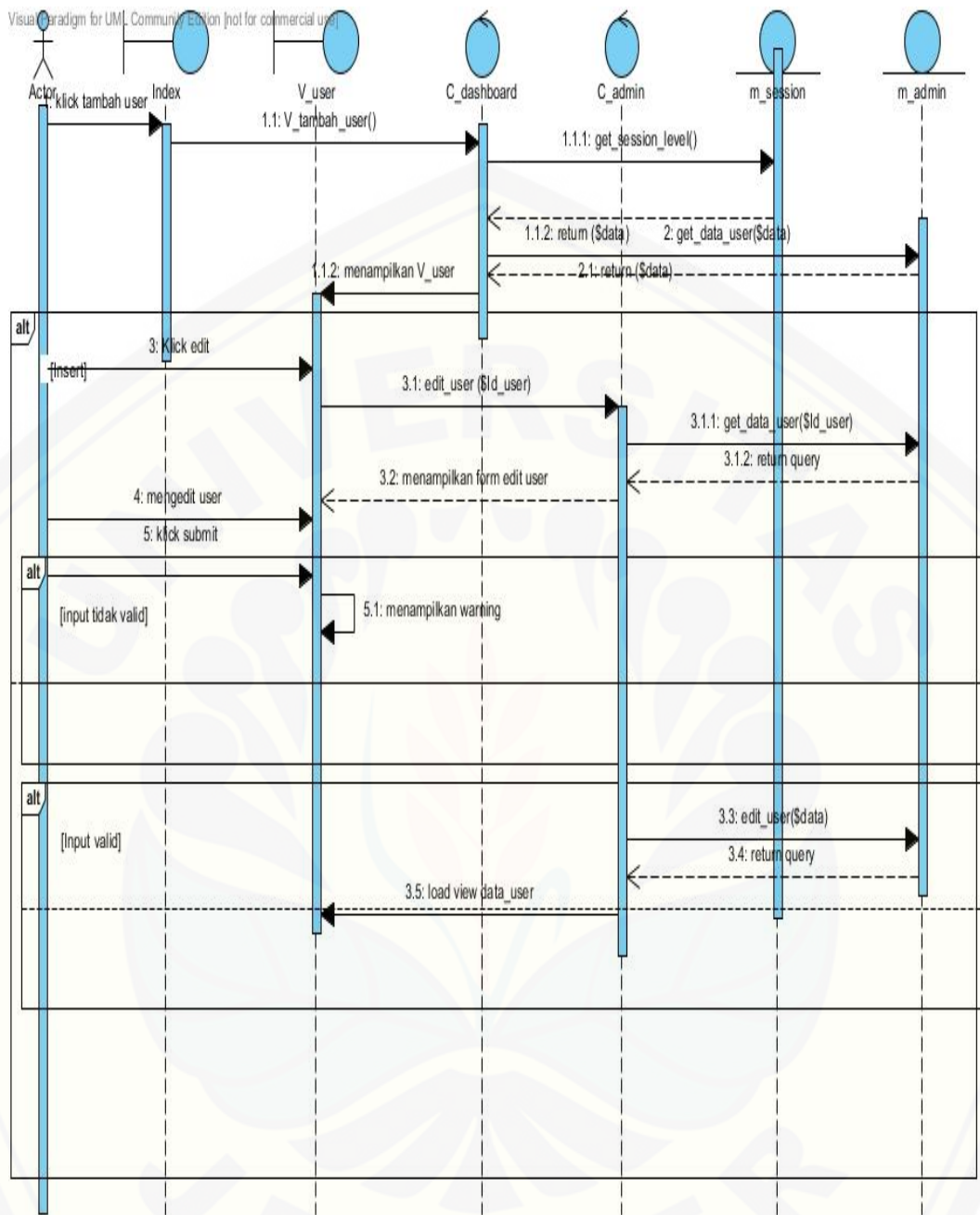
konsumen. *Method* atau perintah akan menuju ke *class* yang dituju. *Class* pada *Sequence Diagram insert data user* konsumen ada login, tambah_user_konsumen, C_dashboard, C_admin, dan m_admin. Alur *method* jika aktor menambah data, aktor mengklik menu daftar pada halaman login, *method* yang berjalan dari index menuju c_dashboard adalah tambah_user_konsumen. *Method* tersebut untuk memanggil form sehingga nantinya aktor bisa menambahkan data user. *Sequence Diagram insert data user oleh konsumen* dapat dilihat pada Lampiran (Lampiran Sequence)

8. *Sequence Diagram insert data user*

Sequence Diagram insert data user menjelaskan alur perintah atau *method* yang dijalankan pada saat aktor yaitu pemilik menambah data user. *Method* atau perintah akan menuju ke *class* yang dituju. *Class* pada *Sequence Diagram insert data user* ada index, V_user, tambah_user, C_dashboard, C_admin, m_session, dan m_admin. Alur *method* jika aktor menambah data, aktor mengklik menu data_user pada index, *method* yang berjalan dari index menuju c_dashboard adalah v_tambah_user. *Method* tersebut untuk memanggil data dari database sehingga nantinya aktor bisa menambahkan data pemesanan. *Sequence Diagram insert data user oleh pemilik* dapat dilihat pada Lampiran (Lampiran Sequence)

9. *Sequence Diagram edit user*

Sequence Diagram insert data pemesanan user konsumen menjelaskan alur perintah atau *method* yang dijalankan pada saat aktor yaitu pemilik mengedit data user. *Method* atau perintah akan menuju ke *class* yang dituju. *Class* pada *Sequence Diagram edit user* ada index, V_user, C_dashboard, C_admin, m_session, dan m_admin. Alur *method* jika aktor menambah data, aktor mengklik menu data_user pada index, *method* yang berjalan dari index menuju c_dashboard adalah v_tambah_user. *Method* tersebut untuk memanggil data sehingga nantinya aktor bisa mengedit data user. *Sequence Diagram edit user oleh pemilik* dijelaskan pada Gambar 4.9

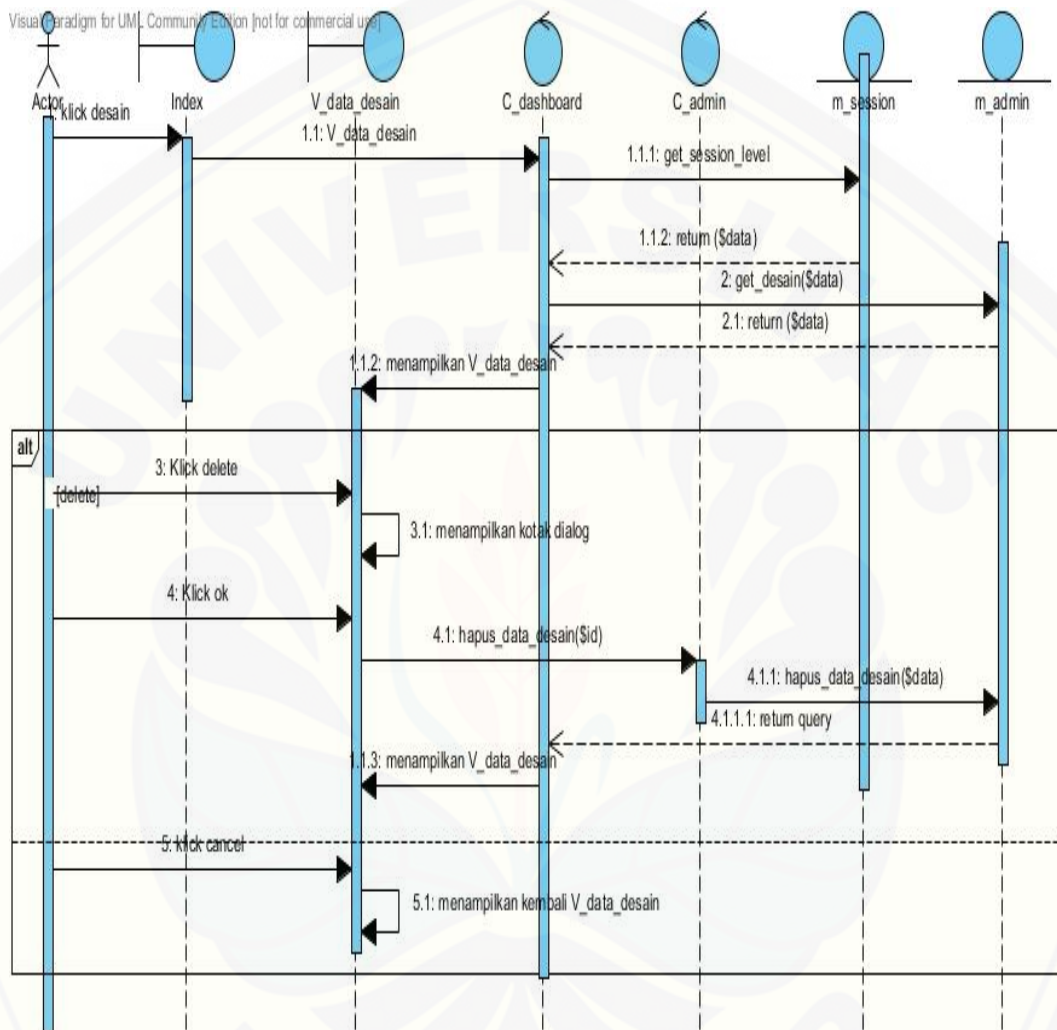


Gambar 4.9 edit user

10. Sequence Diagram Delete data desain

Sequence Diagram delete data desain menjelaskan alur perintah atau method yang dijalankan pada saat aktor yaitu admin menghapus data desain. Method atau perintah akan menuju ke class yang dituju. Class pada Sequence Diagram delete data desain ada index, V_data_desain, C_dashboard, C_admin, m_session, dan m_admin. Alur method jika aktor menghapus data, aktor

mengklik menu data desainr pada index, *method* yang berjalan dari index menuju *c_dashboard* adalah *v_data_desain*. *Method* tersebut untuk memanggil data dari database sehingga nantinya aktor bisa menghapus data desain. *Sequence Diagram delete* data desain oleh pemilik dapat dilihat pada Gambar 4.10



Gambar 4.10 *Sequence Diagram delete* data desain

11. *Sequence Diagram Delete* data kain

Sequence Diagram delete data kain menjelaskan alur perintah atau *method* yang dijalankan pada saat aktor yaitu admin menghapus data kain. *Method* atau perintah akan menuju ke *class* yang dituju. *Class* pada *Sequence Diagram delete* data kain ada index, *V_data_kain*, *C_dashboard*, *C_admin*, *m_session*, dan *m_admin*. Alur *method* jika aktor menghapus data, aktor mengklik menu data kain pada index, *method* yang berjalan dari index menuju *c_dashboard* adalah

v_data_kain. *Method* tersebut untuk memanggil data dari database sehingga nantinya aktor bisa menghapus data desain. *Sequence Diagram delete* data kain oleh pemilik dapat dilihat pada Lampiran (Lampiran Sequence)

12. *Sequence Diagram Delete* data prediksi

Sequence Diagram delete data prediksi menjelaskan alur perintah atau *method* yang dijalankan pada saat aktor yaitu admin menghapus data kain. *Method* atau perintah akan menuju ke *class* yang dituju. *Class* pada *Sequence Diagram delete* data prediksi ada index, V_data_prediksi, C_dashboard, C_admin, m_session, dan m_admin. Alur *method* jika aktor menghapus data, aktor mengklik menu data prediksi pada index, *method* yang berjalan dari index menuju c_dashboard adalah v_data_prediksi. *Method* tersebut untuk memanggil data dari database sehingga nantinya aktor bisa menghapus data prediksi. *Sequence Diagram delete* data prediksi oleh pemilik dapat dilihat pada Lampiran (Lampiran Sequence)

13. *Sequence Diagram Delete* data user

Sequence Diagram delete data user menjelaskan alur perintah atau *method* yang dijalankan pada saat aktor yaitu pemilik menghapus data user. *Method* atau perintah akan menuju ke *class* yang dituju. *Class* pada *Sequence Diagram delete* data user ada index, V_data_user, C_dashboard, C_admin, m_session, dan m_admin. Alur *method* jika aktor menghapus data, aktor mengklik menu data kain pada index, *method* yang berjalan dari index menuju c_dashboard adalah v_data_user. *Method* tersebut untuk memanggil data dari database sehingga nantinya aktor bisa menghapus data user. *Sequence Diagram delete* data user oleh pemilik dapat dilihat pada Lampiran (Lampiran Sequence)

14. *Sequence Diagram Delete* data stock

Sequence Diagram delete data stock menjelaskan alur perintah atau *method* yang dijalankan pada saat aktor yaitu admin menghapus data stock. *Method* atau perintah akan menuju ke *class* yang dituju. *Class* pada *Sequence Diagram delete* data stock ada index, V_data_stock, C_dashboard, C_admin, m_session, dan m_admin. Alur *method* jika aktor menghapus data, aktor mengklik menu data stock pada index, *method* yang berjalan dari index menuju

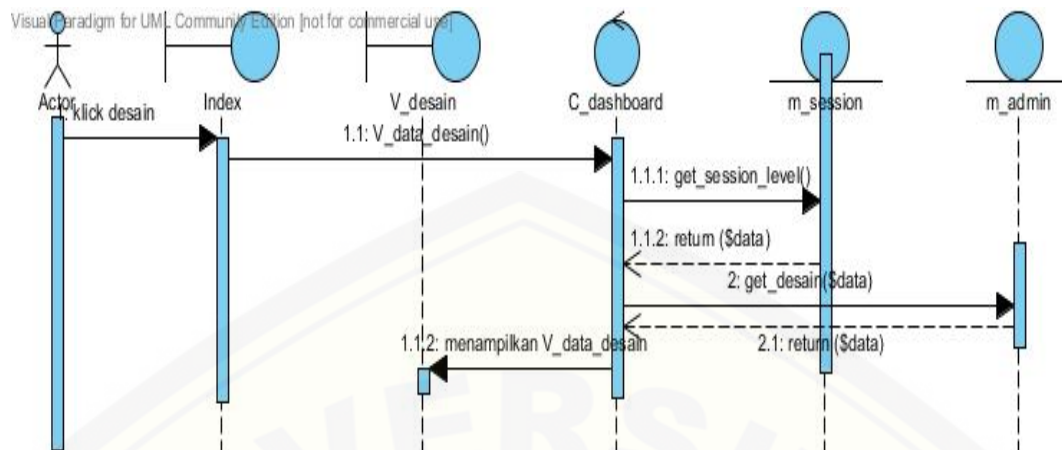
c_dashboard adalah v_data_stock. *Method* tersebut untuk memanggil data dari database sehingga nantinya aktor bisa menghapus data *stock*. *Sequence Diagram delete* data *stock* oleh admin dapat dilihat pada Lampiran (Lampiran *Sequence*)

15. *Sequence Diagram Delete* data pemesanan

Sequence Diagram delete data pemesanan menjelaskan alur perintah atau *method* yang dijalankan pada saat aktor yaitu admin menghapus data pemesanan. *Method* atau perintah akan menuju ke *class* yang dituju. *Class* pada *Sequence Diagram delete* data pemesanan ada index, V_data_pemesanan, C_dashboard, C_admin, m_session, dan m_admin. Alur *method* jika aktor menghapus data, aktor mengklik menu data pemesanan pada index, *method* yang berjalan dari index menuju c_dashboard adalah v_data_pemesanan. *Method* tersebut untuk memanggil data dari database sehingga nantinya aktor bisa menghapus data pemesanan. *Sequence Diagram delete* data pemesanan oleh pemilik dapat dilihat pada Lampiran (Lampiran *Sequence*)

16. *Sequence Diagram view* data desain

Sequence Diagram view data desain menjelaskan alur perintah atau *method* yang dijalankan pada saat aktor yaitu admin melihat data desain. *Method* atau perintah akan menuju ke *class* yang dituju. *Class* pada *Sequence Diagram view* data desain ada index, V_data_desain, C_dashboard, C_admin, m_session, dan m_admin. Alur *method* jika aktor melihat data, aktor mengklik menu data desain pada index, *method* yang berjalan dari index menuju c_dashboard adalah v_data_desain. *Method* tersebut untuk memanggil data dari database sehingga nantinya aktor bisa melihat data desain. *Sequence Diagram view* data desain oleh admin dapat dilihat pada Gambar 4.11



Gambar 4.11 Sequence Diagram view data desain

17. Sequence Diagram view data kain

Sequence Diagram view data kain menjelaskan alur perintah atau *method* yang dijalankan pada saat aktor yaitu admin melihat data desain. *Method* atau perintah akan menuju ke *class* yang dituju. *Class* pada *Sequence Diagram view data kain* ada index, V_data_kain, C_dashboard, C_admin, m_session, dan m_admin. Alur *method* jika aktor melihat data, aktor mengklik menu data kain pada index, *method* yang berjalan dari index menuju c_dashboard adalah v_data_kain. *Method* tersebut untuk memanggil data dari database sehingga nantinya aktor bisa melihat data kain. *Sequence Diagram view data kain* oleh admin dapat dilihat pada Lampiran (Lampiran *Sequence*)

18. Sequence Diagram view data stock

Sequence Diagram view data stock menjelaskan alur perintah atau *method* yang dijalankan pada saat aktor yaitu admin melihat data stock. *Method* atau perintah akan menuju ke *class* yang dituju. *Class* pada *Sequence Diagram view data desain* ada index, V_data_stock, C_dashboard, C_admin, m_session, dan m_admin. Alur *method* jika aktor melihat data, aktor mengklik menu data stock pada index, *method* yang berjalan dari index menuju c_dashboard adalah v_data_stock. *Method* tersebut untuk memanggil data dari database sehingga nantinya aktor bisa melihat data stock. *Sequence Diagram view data stock* oleh admin dapat dilihat pada Lampiran (Lampiran *Sequence*)

19. *Sequence Diagram view data user*

Sequence Diagram view data user menjelaskan alur perintah atau *method* yang dijalankan pada saat aktor yaitu pemilik melihat data *user*. *Method* atau perintah akan menuju ke *class* yang dituju. *Class* pada *Sequence Diagram view data user* ada index, *V_user*, *C_dashboard*, *C_admin*, *m_session*, dan *m_admin*. Alur *method* jika aktor melihat data, aktor mengklik menu data *user* pada index, *method* yang berjalan dari index menuju *c_dashboard* adalah *v_user*. *Method* tersebut untuk memanggil data dari database sehingga nantinya aktor bisa melihat data *user*. *Sequence Diagram view data user* oleh pemilik dapat dilihat pada Lampiran (Lampiran *Sequence*)

20. *Sequence Diagram view data prediksi*

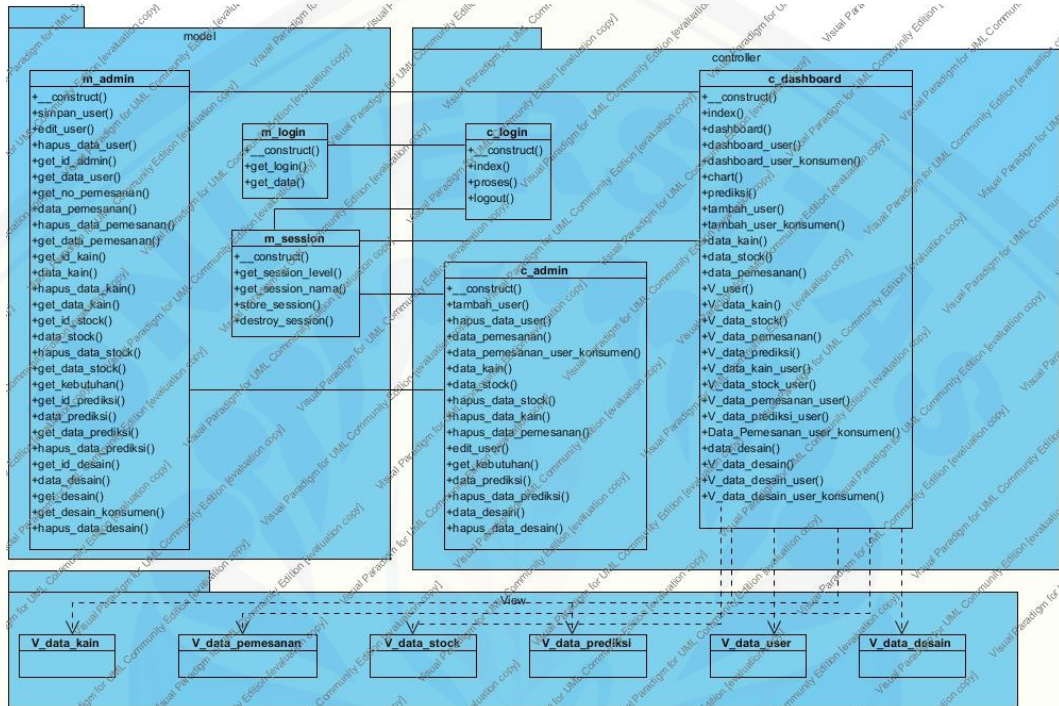
Sequence Diagram view data prediksi menjelaskan alur perintah atau *method* yang dijalankan pada saat aktor yaitu admin melihat data prediksi. *Method* atau perintah akan menuju ke *class* yang dituju. *Class* pada *Sequence Diagram view data prediksi* ada index, *V_data_prediksi*, *C_dashboard*, *C_admin*, *m_session*, dan *m_admin*. Alur *method* jika aktor melihat data, aktor mengklik menu data prediksi pada index, *method* yang berjalan dari index menuju *c_dashboard* adalah *v_data_prediksi*. *Method* tersebut untuk memanggil data dari database sehingga nantinya aktor bisa melihat data prediksi. *Sequence Diagram view data prediksi* oleh admin dapat dilihat pada Lampiran (Lampiran *Sequence*)

21. *Sequence Diagram view data pemesanan*

Sequence Diagram view data pemesanan menjelaskan alur perintah atau *method* yang dijalankan pada saat aktor yaitu admin melihat data pemesanan. *Method* atau perintah akan menuju ke *class* yang dituju. *Class* pada *Sequence Diagram view data pemesanan* ada index, *V_data_pemesanan*, *C_dashboard*, *C_admin*, *m_session*, dan *m_admin*. Alur *method* jika aktor melihat data, aktor mengklik menu data pemesanan pada index, *method* yang berjalan dari index menuju *c_dashboard* adalah *v_data_pemesanan*. *Method* tersebut untuk memanggil data dari database sehingga nantinya aktor bisa melihat data pemesanan. *Sequence Diagram view data pemesanan* oleh admin dapat dilihat pada Lampiran (Lampiran *Sequence*)

4.3.6 Class Diagram

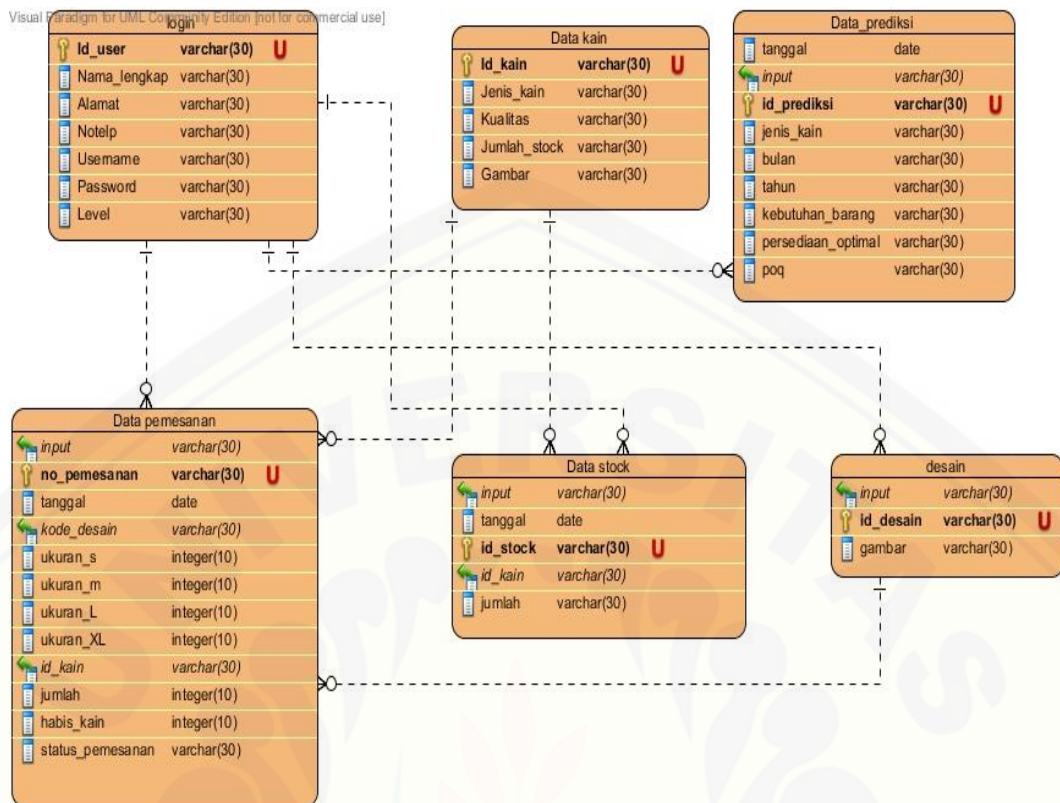
Class diagram menggambarkan hubungan antara kelas yang digunakan untuk membangun suatu Aplikasi. Dalam paradigma OOP terdapat 3 jenis kelas yaitu model, view dan controller. Berdasarkan *sequence* diagram yang telah dibangun, Gambar 4.12



Gambar 4.12 Class Diagram Aplikasi optimalisasi persediaan

4.3.7 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) menggambarkan komponen dan struktur *database* yang digunakan dalam pembuatan Aplikasi. ERD yang diimplementasikan pada Aplikasi ini terdiri dari 6 entitas sebagaimana dapat dilihat pada Gambar 4.13



Gambar 4.13 entity relationship diagram

4.4 Pengkodean Aplikasi

Pengkodean dalam pembuatan Aplikasi ini dilakukan dengan penulisan kode program menggunakan Bahasa pemrograman PHP (*Page Hypertext Pre-Processor*) dengan *Framework Code Igniter* yang terintegrasi dengan database menggunakan MySQL dan *tool XAMPP*.

4.4.1 Menghitung persediaan optimal

Kode program dari perhitungan persediaan optimal menggunakan metode *Period Order Quantity* (POQ) terdiri dari model, *controller*, dan *view*. Berikut kode program dari perhitungan persediaan optimal :

Controller prediksi seperti Gambar 4.14, model prediksi seperti Gambar 4.15, dan *view* prediksi seperti Gambar 4.16

```
function get_kebutuhan(){
    $status = $this->input->post('status');
    $kain = $this->input->post('jenis_kain');
    $bulan = $this->input->post('bulan');
    $tahun = $this->input->post('tahun');

    $result= $this->m_admin->get_kebutuhan($status,$kain,$bulan,$tahun);
    $kebutuhan="";
    foreach ($result->result_array() as $key) {
        $kebutuhan = $key['habis'] ;
    }
    echo $kebutuhan;
}

function data_prediksi(){
    $tanggal = $this->input->post('tanggal');
    $input = $this->input->post('input');
    $id_prediksi = $this->input->post('id_prediksi');
    $jenis_kain = $this->input->post('jenis_kain');
    $bulan = $this->input->post('bulan');
    $tahun = $this->input->post('tahun');
    $kebutuhan_barang = $this->input->post('kebutuhan_barang');
    $persediaan_optimal = $this->input->post('persediaan_optimal');
    $poq = $this->input->post('poq');

    $data = array(
        'tanggal' => $tanggal,
        'input' => $input,
        'id_prediksi' => $id_prediksi,
        'jenis_kain' => $jenis_kain,
        'bulan' => $bulan,
        'tahun' => $tahun,
        'kebutuhan_barang' => $kebutuhan_barang,
        'persediaan_optimal' => $persediaan_optimal,
        'poq' => $poq
    );

    $logic = $this->m_admin->data_prediksi($data);
    redirect('c_dashboard/v_data_prediksi');
}

function hapus_data_prediksi(){

    $data['id']= $this->input->post('id_stock');
    $data['jenis_kain']= $this->input->post('jenis_kain');
    $logic = $this->m_admin->hapus_data_prediksi($data);
    redirect('c_dashboard/v_data_stock');
}
```

Gambar 4.14 kode program *controller* prediksi

```
function get_kebutuhan($status,$id,$bulan,$tahun) {
    $q = $this->db->query("SELECT sum(habis_kain) as habis
    from data_pemesanan
    where id_kain='$id' and
    MONTH(tanggal)='$bulan' and
    YEAR(tanggal)='$tahun' and
    status_pemesanan='$status'
    group by id_kain ,
    MONTH(tanggal),
    YEAR(tanggal),
    status_pemesanan");
    return $q;
}
function get_id_prediksi() {
    $q = $this->db->query("select MAX(RIGHT(id_prediksi,3)) as id_max from data_prediksi");
    $kd = '';
    if ($q->num_rows() > 0) {
        foreach ($q->result_array() as $k) {
            $tmp = ((int) $k['id_max']) + 1;
            $kd = sprintf("%03s", $tmp);
        }
    } else {
        $kd = '001';
    }
    return 'P-' . $kd;
}
function data_prediksi($data) {
    return $this->db->insert('data_prediksi', $data);
}
function get_data_prediksi() {
    $q = $this->db->query("select * from data_prediksi");
    return $q;
}
function hapus_data_prediksi($data) {
    $this->db->query("delete from data_prediksi where id_prediksi='$data[id_prediksi]'");
}
}
```

Gambar 4.15 kode program model prediksi


```

<!DOCTYPE html>
<html lang="en">
<head>
  <!--
  ===
  This comment should NOT be removed.

  Charisma v2.0.0

  Copyright 2012-2014 Muhammad Usman
  Licensed under the Apache License v2.0
  http://www.apache.org/licenses/LICENSE-2.0

  http://usman.it
  http://twitter.com/halalit_usman
  ===
  -->
  <meta charset="utf-8">
  <title>Free HTML5 Bootstrap Admin Template</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="description" content="Charisma, a fully featured, responsive, HTML5, Bootstrap admin template.">
  <meta name="author" content="Muhammad Usman">

  <!-- The styles -->
  <link id="bs-css" href="{%= base_url(); %}assets/css/bootstrap-slate.min.css" rel="stylesheet">

  <link href="{%= base_url(); %}assets/css/charisma-app.css" rel="stylesheet">
  <link href="{%= base_url(); %}assets/bower_components/fullcalendar/dist/fullcalendar.css" rel="stylesheet">
  <link href="{%= base_url(); %}assets/bower_components/fullcalendar/dist/fullcalendar.print.css" rel="stylesheet" media="print">
  <link href="{%= base_url(); %}assets/bower_components/chosen/chosen.min.css" rel="stylesheet">
  <link href="{%= base_url(); %}assets/bower_components/colorbox/example3/colorbox.css" rel="stylesheet">
  <link href="{%= base_url(); %}assets/bower_components/responsive-tables/responsive-tables.css" rel="stylesheet">
  <link href="{%= base_url(); %}assets/bower_components/bootstrap-tour/build/css/bootstrap-tour.min.css" rel="stylesheet">
  <link href="{%= base_url(); %}assets/css/jquery.noty.css" rel="stylesheet">
  <link href="{%= base_url(); %}assets/css/noty_theme_default.css" rel="stylesheet">
  <link href="{%= base_url(); %}assets/css/elfinder.min.css" rel="stylesheet">
  <link href="{%= base_url(); %}assets/css/elfinder.theme.css" rel="stylesheet">
  <link href="{%= base_url(); %}assets/css/jquery.iphone.toggle.css" rel="stylesheet">
  <link href="{%= base_url(); %}assets/css/uploadify.css" rel="stylesheet">
  <link href="{%= base_url(); %}assets/css/animate.min.css" rel="stylesheet">

  <!-- jQuery -->
  <script src="{%= base_url(); %}assets/bower_components/jquery/jquery.min.js"></script>

  <link rel="shortcut icon" href="img/favicon.ico">
</head>
<body>
  <!-- topbar starts -->
  <div class="navbar navbar-default" role="navigation">
    <div class="navbar-inner">
      <button type="button" class="navbar-toggle pull-left animated flip">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="index.html"> 
      <span>DwidakuJaya</span></a>
    <!-- user dropdown starts -->
    <div class="btn-group pull-right">
      <button class="btn btn-default dropdown-toggle" data-toggle="dropdown">
        <i class="glyphicon glyphicon-user"></i><span class="hidden-sm hidden-xs"> {%= echo $level; %}</span>
        <span class="caret"></span>
      </button>
      <ul class="dropdown-menu">
        <li class="divider"></li>
        <li><a href="{%= base_url(); %}index.php/c_login/logout">Logout</a></li>
      </ul>
    </div>
    <!-- user dropdown ends -->
  </div>
  <!-- topbar ends -->
  <div class="ch-container">
    <div class="row">
      <!-- left menu starts -->
      <div class="col-sm-2 col-lg-2">
        <div class="sidebar-nav">
          <div class="nav-canvas">
            <div class="nav-sm nav nav-stacked">
              </div>
            <ul class="nav nav-pills nav-stacked main-menu">
              <li class="nav-header">Main</li>

```

```

<li><a class="ajax-link" href="<?= base_url(); ?>index.php/c_dashboard/dashboard">
<i class="glyphicon glyphicon-home"></i><span> Dashboard</span></a>
</li>
<li><a class="ajax-link" href="<?= base_url(); ?>index.php/c_dashboard/V_data_pemesanan"><i
class="glyphicon glyphicon-edit"></i><span> Data Pemesanan</span></a></li>
<li><a class="ajax-link" href="<?= base_url(); ?>index.php/c_dashboard/V_data_kain"><i
class="glyphicon glyphicon-edit"></i><span> Data Kain</span></a></li>
<li><a class="ajax-link" href="<?= base_url(); ?>index.php/c_dashboard/V_data_stock"><i
class="glyphicon glyphicon-edit"></i><span> Data Stock</span></a></li>
<li><a class="ajax-link" href="<?= base_url(); ?>index.php/c_dashboard/V_data_desain"><i
class="glyphicon glyphicon-edit"></i><span> Desain</span></a></li>
<li><a class="ajax-link" href="<?= base_url(); ?>index.php/c_dashboard/V_data_prediksi"><i
class="glyphicon glyphicon-edit"></i><span> Prediksi</span></a></li>
</ul>
</div>
</div>
</div>
<!-- /span -->
<!-- left menu ends -->
<noscript>
<div class="alert alert-block col-md-12">
<h4 class="alert-heading">Warning!</h4>
<p>You need to have <a href="http://en.wikipedia.org/wiki/JavaScript" target="_blank">JavaScript</a>
enabled to use this site.</p>
</div>
</noscript>
<div id="content" class="col-lg-10 col-sm-10">
<!-- content starts -->
<div>
<ul class="breadcrumb">
<li>
<a href="#">Home</a>
</li>
<li>
<a href="#">Prediksi</a>
</li>
</ul>
</div>
<div class="row">
<div class="box col-md-12">
<div class="box-inner">
<div class="box-header well" data-original-title="">
<h2><i class="glyphicon glyphicon-edit"></i> Prediksi</h2>
<div class="box-icon">
<a href="#" class="btn btn-setting btn-round btn-default"><i
class="glyphicon glyphicon-cog"></i></a>
<a href="#" class="btn btn-minimize btn-round btn-default"><i
class="glyphicon glyphicon-chevron-up"></i></a>
<a href="#" class="btn btn-close btn-round btn-default"><i
class="glyphicon glyphicon-remove"></i></a>
</div>
</div>
<div class="box-content">
<form role="form" action="<?= base_url(); ?>index.php/c_admin/data_prediksi" enctype="multipart/form-data" method="post">
<div class="form-group col-md-4">
<label for="exampleInputPassword1">Tanggal</label>
<input type="text" name="tanggal" class="form-control" value="<? echo date('Y-m-d'); ?>" id="tanggal" readonly>
</div>
<br/><br/><br/>
<div class="form-group col-md-4">
<label for="exampleInputPassword1">Di input</label>
<input type="text" name="input" value="<? echo $nama; ?>" class="form-control" id="" placeholder="" readonly>
</div>
<br/><br/><br/>
<div class="form-group col-md-4">
<label for="exampleInputPassword1">ID Prediksi</label>
<input type="text" name="id_prediksi" value="<? echo $id; ?>" class="form-control" id="id_prediksi" placeholder="" readonly>
</div>
<br/><br/><br/>
<div class="form-group col-md-4">
<input class="input hide" value="Pesan" type="text" class="form-control" name="status" id="status" placeholder="">
</div>
<br/>
<div class="form-group col-md-4">
<label for="exampleInputPassword1">Id kain</label>
<select class="form-control" id="jenis_kain" name="jenis_kain" >
<?php
foreach ($kain->result_array() as $row) {
echo '
<option value="" $row['id_kain'].'' $row['jenis_kain'].'</option>
';
}
}
>
</select>
</div>
<br/><br/><br/>

```



```

<footer class="row">
  <p class="col-md-9 col-sm-9 col-xs-12 copyright">&copy; <a href="#" target="#">Arif putra ardi hansa 2015 </a></p>
  <p class=""><a href="#" target="#">Sistem Informasi || Universitas Jember</a></p>
</footer>
</div><!--/.fluid-container-->

<!-- external javascript -->

<script src="<?>= base_url(); ?>assets/bower_components/bootstrap/dist/js/bootstrap.min.js"></script>

<!-- library for cookie management -->
<script src="<?>= base_url(); ?>assets/js/jquery.cookie.js"></script>
<!-- calendar plugin -->
<script src="<?>= base_url(); ?>assets/bower_components/moment/min/moment.min.js"></script>
<script src="<?>= base_url(); ?>assets/bower_components/fullcalendar/dist/fullcalendar.min.js"></script>
<!-- data table plugin -->
<script src="<?>= base_url(); ?>assets/js/jquery.dataTables.min.js"></script>

<!-- select or dropdown enhancer -->
<script src="<?>= base_url(); ?>assets/bower_components/chosen/chosen.jquery.min.js"></script>
<!-- plugin for gallery image view -->
<script src="<?>= base_url(); ?>assets/bower_components/colorbox/jquery.colorbox-min.js"></script>
<!-- notification plugin -->
<script src="<?>= base_url(); ?>assets/js/jquery.noty.js"></script>
<!-- library for making tables responsive -->
<script src="<?>= base_url(); ?>assets/bower_components/responsive-tables/responsive-tables.js"></script>
<!-- tour plugin -->
<script src="<?>= base_url(); ?>assets/bower_components/bootstrap-tour/build/js/bootstrap-tour.min.js"></script>
<!-- star rating plugin -->
<script src="<?>= base_url(); ?>assets/js/jquery.raty.min.js"></script>
<!-- for ios style toggle switch -->
<script src="<?>= base_url(); ?>assets/js/jquery.iphone.toggle.js"></script>
<!-- autogrowing textarea plugin -->
<script src="<?>= base_url(); ?>assets/js/jquery.autogrow-textarea.js"></script>
<!-- multiple file upload plugin -->
<script src="<?>= base_url(); ?>assets/js/jquery.uploadify-3.1.min.js"></script>
<!-- history.js for cross-browser state change on ajax -->
<script src="<?>= base_url(); ?>assets/js/jquery.history.js"></script>
</script>

jQuery(document).ready(function(){

  $('#tahun').change(function(){
    kebutuhan();
  });

  $('#bulan').change(function(){
    kebutuhan();
  });

  $('#jenis_kain').change(function(){
    kebutuhan();
  });

  function kebutuhan(){
    var status = $('#status').val();
    var kain = $('#jenis_kain').val();
    var bulan = $('#bulan').val();
    var tahun = $('#tahun').val();

    $.ajax({
      type:"POST",
      url: "<?>php echo base_url() ?>index.php/c_admin/get_kebutuhan/",
      data: 'status='+status+'&jenis_kain='+kain+'&bulan='+bulan+'&tahun='+tahun,
      success:function(data){
        $('#kebutuhan_barang').val(data);
      }
    });
  }

  $('#hitung').click(function(){
    hitung();
  });

});

```

```
function hitung(){
    var kebutuhan =parseInt( $('#kebutuhan_barang').val());
    var pemesanan = parseInt($('#pemesanan').val());
    var penyimpanan = parseInt($('#penyimpanan').val());
    var persediaan =Math.sqrt(2*kebutuhan*pemesanan/penyimpanan);
    $('#persediaan_optimal').val(Math.round(persediaan));
    var poq =persediaan/kebutuhan;
    $('#poq').val(Math.round(poq))
};
for( i = new Date().getFullYear(); i>2012; i++){
$.each(function(index,value){
    $('#tahun').append($('
```

Gambar 4.16 kode program *view* prediksi

4.5 Pengujian Aplikasi

Pada penelitian ini peneliti menggunakan dua metode pengujian Aplikasi yaitu *Black Box Testing* dan *White Box Testing*. Berikut adalah hasil dari pengujian Aplikasi:

4.5.1 *White Box Testing*

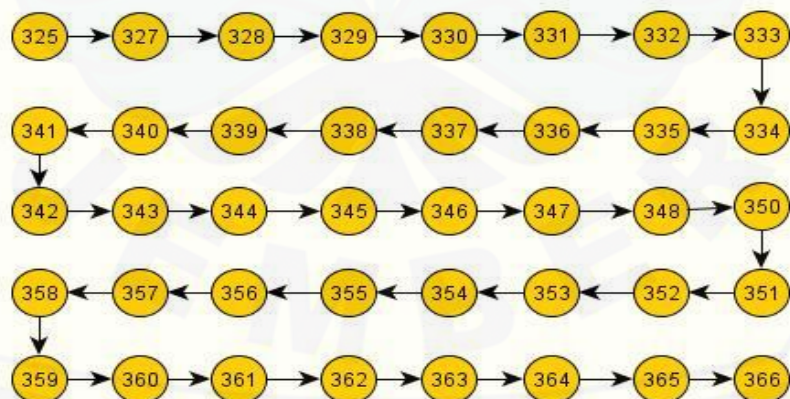
Pengujian *white box* dilakukan untuk mengevaluasi Aplikasi yang telah dibuat. Pengujian *white box* yang dilakukan pada penelitian ini meliputi grafik alir, kompleksitas siklomatis, pengujian jalur program dan *test case*. Pengujian yang dilakukan adalah pada fitur prediksi yang menggunakan metode *POQ*. Berikut merupakan hasil dari pengujian *white box*:

```

325 <script >
326
327 jQuery(document).ready(function(){
328     $('#tahun').change(function(){
329         kebutuhan();
330     });
331     $('#bulan').change(function(){
332         kebutuhan();
333     });
334     $('#jenis_kain').change(function(){
335
336         kebutuhan();
337     });
338     function kebutuhan(){
339         var status = $('#status').val();
340         var kain = $('#jenis_kain').val();
341         var bulan = $('#bulan').val();
342         var tahun = $('#tahun').val();
343         $.ajax({
344             type:"POST",
345             url: "<?php echo base_url() ?>index.php/c_admin/get_kebutuhan/",
346             data: 'status='+status+'&jenis_kain='+kain+'&bulan='+bulan+'&tahun='+tahun,
347             success:function(data){
348                 $('#kebutuhan_barang').val(data);
349             }
350         });
351     }
352     $('#hitung').click(function(){
353         hitung();
354     });
355     function hitung(){
356         var kebutuhan =parseInt( $('#kebutuhan_barang').val());
357         var pemesanan = parseInt($('#pemesanan').val());
358         var penyimpanan = parseInt($('#penyimpanan').val());
359         var persediaan =Math.sqrt(2*kebutuhan*pemesanan/penyimpanan);
360         $('#persediaan_optimal').val(Math.round(persediaan));
361         var poq =persediaan/kebutuhan;
362         $('#poq').val(Math.round(poq))
363     };
364 });
365 </script>

```

Gambar 4.17 Kode program menghitung pemesanan dan interval pemesanan



Gambar 4.18 Grafik alir perhitungan pemesanan dan interval pemesanan

Cyclomatic complexity dari fitur prediksi diperoleh dengan perhitungan sebagai berikut :

$$V = E - G + 2$$

$$V = 39 - 40 + 2$$

$$V = 1$$

Basis set pesanan efisien memiliki satu jalur *independent* dari perhitungan *cyclomatic complexity*, yaitu :

Jalur 1 : 325-327-328-329-330-331-332-333-334-335-336-337-338-339-340-341-342-343-344-345-346-347-348-349-350-351-352-353-354-355-356-357-358-359-360-361-362-363-364-365

Tabel 4.10 Test Case Menghitung POQ

Test Case : Optimalisasi persediaan bahan baku				
No.	Pengujian	Jalur	Kesimpulan	Target yang Diharapkan
1.	Data prediksi untuk menghitung pemesanan dan interval pemesanan yang optimal dengan cara <i>click</i> “hitung” pada form prediksi	Jalur 1	[√]berhasil []gagal	Berhasil menghitung dan menampilkan data jumlah pesanan dan interval pemesanan yang optimal dalam bentuk tabel

4.5.2 Pengujian *Black Box*

Pengujian dengan metode *black box* dilakukan dengan cara memberikan alur penggunaan Aplikasi kepada koresponden. Selanjutnya koresponden melakukan analisa fungsi yang terdapat pada Aplikasi sesuai atau tidak dengan kebutuhan Aplikasi. Pengujian ini berguna untuk menguji *interface* dari Aplikasi untuk memutuskan nilai masukan berhasil diproses oleh Aplikasi dan menghasilkan keluaran yang benar.

Pengujian *black box* yang dilakukan terhadap perhitungan pemesanan dan interval pemesanan selanjutnya seperti disajikan pada Tabel 4.11 dibawah ini

Tabel 4.11 Pengujian black box perhitungan pemesanan dan interval pemesanan

No.	Fungsi	Aksi	Hasil	Kesimpulan
1.	Menghitung pemesanan dan interval pemesanan	Memilih jenis kain, bulan, dan tahun serta menekan tombol “Hitung”	Data prediksi berhasil ditampilkan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
2.	Menampilkan peringatan bahwa data yang diminta tidak ada	Memilih jenis kain, bulan, dan tahun serta menekan tombol “Hitung”	Menampilkan peringatan bahwa data yang diminta kosong dan menampilkan form prediksi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

BAB 5. HASIL DAN PEMBAHASAN

Bab ini menjelaskan mengenai hasil penelitian yang telah dilakukan serta pembahasan Aplikasi yang telah dibuat.

5.1 Hasil Penelitian

Hasil penelitian yang diperoleh dari penelitian ini adalah Aplikasi informasi berbasis web yang dapat digunakan sebagai acuan atau dasar pengoptimalan pemesanan bahan baku kain dengan mengoptimalkan interval pemesanan dalam satu periode tertentu. Bagian manajemen UD.DwidakuJaya dapat menggunakan acuan tersebut untuk melakukan pemesanan bahan baku kain dan juga dapat mengetahui interval pemesanan yang disarankan dalam satu periode waktu..Aplikasi ini dapat menganalisa jumlah pemesanan yang optimal yang dilakukan dalam periode waktu tertentu berdasarkan jumlah kain yang terpakai pada periode waktu sebelumnya.

Jumlah pemakaian kain pada periode sebelumnya digunakan untuk menghitung jumlah pemesanan yang optimal pada periode berikutnya, karena jumlah pemakaian kain dalam setiap periode waktu berbeda. Sehingga jumlah dan interval pemesanan yang didapatkan adalah jumlah pemesanan dan interval pemesanan yang optimal untuk melakukan pemesanan.

5.2 Pemesanan kain Sebelum Menggunakan Aplikasi

UD.Dwidaku Jaya merupakan perusahaan yang bergerak pada bidang industri tekstil. Proses pengadaan bahan baku kain yang dilakukan oleh perusahaan dengan manual yaitu memesan bahan baku kain sesuai permintaan pelanggannya. Jumlah pemesanan dalam satu periode waktu tertentu tergantung oleh permintaan pelanggan. Interval pemesanan bahan baku kain yang dilakukan oleh perusahaan sangat tidak optimal, karena mengikuti berapa jumlah pelanggan yang memesan dalam satu periode. Daftar pemesanan sebelum menggunakan metode pada bulan April dapat dilihat pada Tabel 5.1

Tabel 5.1 Daftar pemesanan bulan maret - agustus 2015

No	Bulan	Nama Pemesanan	Jumlah Pesanan (potong)	Jumlah Kebutuhan (Kg)	Jumlah pesanan ekonomis (Kg)	POQ	error (%)
1	Agustus 2015	Adlan Haris	15	34	22	4	8,8
2		Ali bakri	17				
3		Firman	11				
4		Rivan Tri	15				
5		Supardi	11				
6		Sofyan Hadi	15				
7		Lutfi	7				
8		Bambang Hariyanto	11				
1	Juli 2015	Abdul rahman	20	25	31	3	6
2		M Tauhid	10				
3		Dwi Putra	14				
4		Deni Saaputra	25				
5		M. Syarif	11				
1	Juni 2015	Irwan	21	49	32	2	8,6
2		Rudy	28				
3		Pamungkas candrono	22				
4		M Zainudin	7				
5		Ferdi	14				
6		Sugeng	35				
7		Imam sujono	21				

Dilanjutkan

Lanjutan

No	Bulan	Nama Pemesanan	Jumlah Pesanan (potong)	Jumlah Kebutuhan (Kg)	Jumlah pesanan ekonomis (Kg)	POQ	error (%)
1	Mei 2015	Nando	30	51	35	2	7,8
2		Raka	15				
3		Erik	25				
4		Hendri santoso	35				
5		Riski	50				
6		Fredy gunawan	25				
7		Deny	12				
1	April 2015	Adi Nugroho	30	61	55	1	2,5
2		Nur Hadi	15				
3		Riyanto	25				
4		Edi Santoso	35				
5		Purwito	50				
6		alex	25				
1	Maret 2015	Ibnu	42	150	-	-	-
2		Mualim	46				
3		Budi setiawan	48				
4		Arif	55				
5		Dita	60				
6		Dafid	48				
7		Yoko	48				
8		Rama	45				
9		Mada	55				

5.3 Penggunaan Metode *POQ* Pada Aplikasi

Metode *Period Order Quantity* atau *POQ* digunakan untuk menghitung jumlah pemesanan dan menghitung interval pemesanan yang optimal pada periode selanjutnya berdasarkan jumlah pemesanan pada periode sebelumnya. Penggunaan metode *POQ* pada Aplikasi optimalisasi persediaan bahan baku kain dimaksudkan agar setiap kali melakukan pemesanan, perusahaan dapat menghemat biaya pemesanan dan biaya perawatan yang harus dikeluarkan.

Periode yang dimaksud adalah waktu yang digunakan sebagai batasan penggunaan bahan baku kain pada UD.Dwidaku Jaya. Satu periode waktu adalah satu bulan. Penetapan periode tersebut berdasarkan pada jumlah pemesanan yang mengalami kenaikan dan penurunan dalam setiap bulan. Kenaikan dan penurunan jumlah pemesanan disebabkan beberapa hal tertentu. Hal tersebut yaitu karena sebagian besar pelanggan perusahaan ini adalah sekolah-sekolah dan club olahraga.

Hasil dari akumulasi pemesanan dalam satu periode dihitung dengan rumus

$$POQ = \frac{EOQ}{R} \dots\dots\dots(1)$$

dimana :

POQ = interval pemesanan optimal dalam satu periode

EOQ = kuantitas pemesanan optimal

R = rata-rata permintaan per minggu

Contoh perhitungan menggunakan metode *POQ* adalah sebagai berikut :

Pada bulan Maret tahun 2015 jumlah pemakaian bahan kain mencapai 150 Kg dengan biaya pemesanan Rp. 10.000,- untuk sekali pesan. Dan biaya penyimpanan Rp. 1000,- untuk satu jenis kain. Berdasarkan data tersebut jumlah pemesanan dan interval pemesanan yang optimal untuk periode berikutnya berdasarkan metode *POQ* dapat dihitung :

$$EOQ = \sqrt{\frac{2 \times 150 \text{ kg} \times \text{Rp } 10.000}{\text{Rp } 1.000}}$$

$$EOQ = \sqrt{\frac{Rp\ 3.000.000\ kg}{Rp\ 1.000}}$$

$$EOQ = \sqrt{Rp\ 3.000}\ kg$$

$$EOQ = 55\ Kg$$

$$R = \frac{150}{4}$$

$$R = 38$$

$$POQ = \frac{eoq}{R}$$

$$POQ = \frac{55}{38}$$

$$POQ = 1,4\ (\text{dibulatkan menjadi 1 minggu})$$

Presentase error menggunakan

Mean Absolut Percentage Error

(Mape)

$$Mape = \frac{(y-y')/y}{n} \times 100$$

$$Mape = \frac{(61-55)/61}{4} \times 100$$

$$Mape = 2,5\ \%$$

Jadi pada periode bulan April 2015 jumlah pemesanan yang optimal berdasarkan metode POQ adalah sebesar 55 Kg kain dan interval pemesanan yang optimal adalah 1 minggu dalam satu periode serta dengan persentase *error* yaitu sebesar 15,833 %. Pemesanan dilakukan sebanyak 4 kali dalam satu periode bulan berikutnya. Hasil perhitungan ini dapat di bandingkan dengan sebelum menggunakan metode POQ seperti pada Tabel 5.2

Tabel 5.2 Perbandingan sebelum dan sesudah menggunakan metode POQ

	Sebelum	Sesudah
Kebutuhan	66 Kg	55 Kg
Frekuensi pemesanan	6 kali	4 kali

*pada bulan april dengan error 2,5%

5.4 Implementasi Aplikasi

Hasil penelitian ini diperoleh Aplikasi optimalisasi persediaan bahan baku kain di UD.DwidakuJaya yang dirancang menggunakan model perancangan *waterfall*. Aplikasi optimalisasi persediaan bahan baku kain di UDDwidakuJaya ini memiliki beberapa fitur. Fitur – fitur tersebut antara lain *login*, manajemen data

kain, manajemen data *user*, manajemen data pemesanan, manajemen data *stock*, manajemen data desain, dan manajemen data prediksi. Fitur – fitur tersebut akan dijelaskan pada sub bab dibawah ini :

5.4.1 Fitur *login*

Fitur *login* merupakan fitur yang berfungsi untuk mengautentikasi hak akses aktor yang akan masuk ke Aplikasi optimalisasi persediaan bahan baku kain. Aktor masuk ke fitur *login* dan akan memasukkan *username* dan *password* yang dimiliki. Jika *username* dan *password* yang dimasukkan oleh aktor sesuai dengan *database* maka Aplikasi akan menampilkan halaman dashboard yang disesuaikan dengan level atau hak akses dari aktor. Halaman *login* dapat dilihat pada Lampiran (Lampiran fitur)

5.4.2 Fitur Manajemen Data Kain

Fitur manajemen data kain pada aplikasi optimalisasi persediaan bahan baku kain meliputi fitur menambah dan menghapus data kain

5.4.2.1 Fitur menambah Data Kain

Fitur menambah data kain pada Aplikasi optimalisasi persediaan bahan baku kain merupakan fitur yang dapat diakses oleh aktor sebagai admin untuk menambah varian kain yang disediakan oleh perusahaan UD.DwidakuJaya. Halaman untuk menambah data kain dapat dilihat di Lampiran (Lampiran fitur).

5.4.2.2 Fitur Menghapus Data Kain

Fitur menghapus data kain pada Aplikasi optimalisasi persediaan bahan baku kain merupakan fitur yang dapat diakses oleh aktor sebagai admin untuk menghapus varian kain yang disediakan oleh perusahaan UD.DwidakuJaya. Halaman untuk menghapus data kain dapat dilihat di Lampiran (Lampiran fitur).

5.4.3 Fitur Manajemen Data *user*

Fitur manajemen data *user* pada aplikasi optimalisasi persediaan bahan baku kain meliputi fitur menambah, mengubah dan menghapus data *user*

5.4.3.1 Fitur Menambah Data *user*

Fitur menambah data *user* pada Aplikasi optimalisasi persediaan bahan baku kain merupakan fitur yang dapat diakses oleh aktor sebagai pemilik atau konsumen untuk menambah atau mendaftar data *user* untuk dapat masuk pada

Aplikasi web perusahaan UD.DwidakuJaya. Halaman untuk menambah data *user* dapat dilihat di Lampiran (Lampiran fitur).

5.4.3.2 Fitur Mengubah Data *User*

Fitur mengubah data *user* pada Aplikasi optimalisasi persediaan bahan baku kain merupakan fitur yang dapat diakses oleh aktor sebagai pemilik untuk mengubah data *user* yang telah tersimpan di dalam *database* Aplikasi optimalisasi persediaan bahan baku kain perusahaan UD.DwidakuJaya. Halaman untuk mengubah data *user* dapat dilihat di Lampiran (Lampiran fitur).

5.4.3.3 Fitur Menghapus Data *user*

Fitur menghapus data *user* pada Aplikasi optimalisasi persediaan bahan baku kain merupakan fitur yang dapat diakses oleh aktor sebagai pemilik untuk menghapus data *user* yang tersimpan di dalam *database* Aplikasi optimalisasi persediaan bahan baku perusahaan UD.DwidakuJaya. Halaman untuk menghapus data *user* dapat dilihat di Lampiran (Lampiran fitur).

5.4.4 Fitur Manajemen Data Pemesanan

Fitur manajemen data pemesanan pada aplikasi optimalisasi persediaan bahan baku kain meliputi fitur menambah dan menghapus data pemesanan

5.4.4.1 Fitur Menambah Data Pemesanan

Fitur menambah data pemesanan pada Aplikasi optimalisasi persediaan bahan baku kain merupakan fitur yang dapat diakses oleh aktor sebagai admin dan konsumen untuk melakukan pemesanan . Halaman untuk menambah data pemesanan dapat dilihat di Lampiran (Lampiran fitur).

5.4.4.2 Fitur Menghapus Data Pemesanan

Fitur menghapus data pemesanan pada Aplikasi optimalisasi persediaan bahan baku kain merupakan fitur yang dapat diakses oleh aktor sebagai admin untuk membatalkan pemesanan yang telah terdaftar di Aplikasi web perusahaan UD.DwidakuJaya. Halaman untuk menghapus data pemesanan dapat dilihat di Lampiran (Lampiran fitur).

5.4.5 Fitur Manajemen Data *Stock*

Fitur manajemen data stock pada aplikasi optimalisasi persediaan bahan baku kain meliputi fitur menambah dan menghapus data stock

5.4.5.1 Fitur Menambah Data *Stock*

Fitur menambah data *stock* pada Aplikasi optimalisasi persediaan bahan baku kain merupakan fitur yang dapat diakses oleh aktor sebagai admin untuk menambah menambah *stock* bahan baku kain yang disediakan oleh perusahaan UD.DwidakuJaya. Halaman untuk menambah data *stock* dapat dilihat di Lampiran (Lampiran fitur).

5.4.5.2 Fitur Menghapus Data *Stock*

Fitur menghapus data *Stock* pada Aplikasi optimalisasi persediaan bahan baku kain merupakan fitur yang dapat diakses oleh aktor sebagai admin untuk menghapus data *stock* jika terjadi kesalahan saat menambah data *stock* bahan baku kain yang disediakan oleh perusahaan UD.DwidakuJaya. Halaman untuk menghapus data kain dapat dilihat di Lampiran (Lampiran fitur).

5.4.6 Fitur Manajemen Data Desain

Fitur manajemen data desain pada aplikasi optimalisasi persediaan bahan baku kain meliputi fitur menambah dan menghapus data desain

5.4.6.1 Fitur Menambah Data Desain

Fitur menambah data desain pada Aplikasi optimalisasi persediaan bahan baku kain merupakan fitur yang dapat diakses oleh aktor sebagai admin untuk menambah varian desain yang disediakan oleh perusahaan UD.DwidakuJaya. Halaman untuk menambah data desain dapat dilihat di Lampiran (Lampiran fitur).

5.4.6.2 Fitur Menghapus Data desain

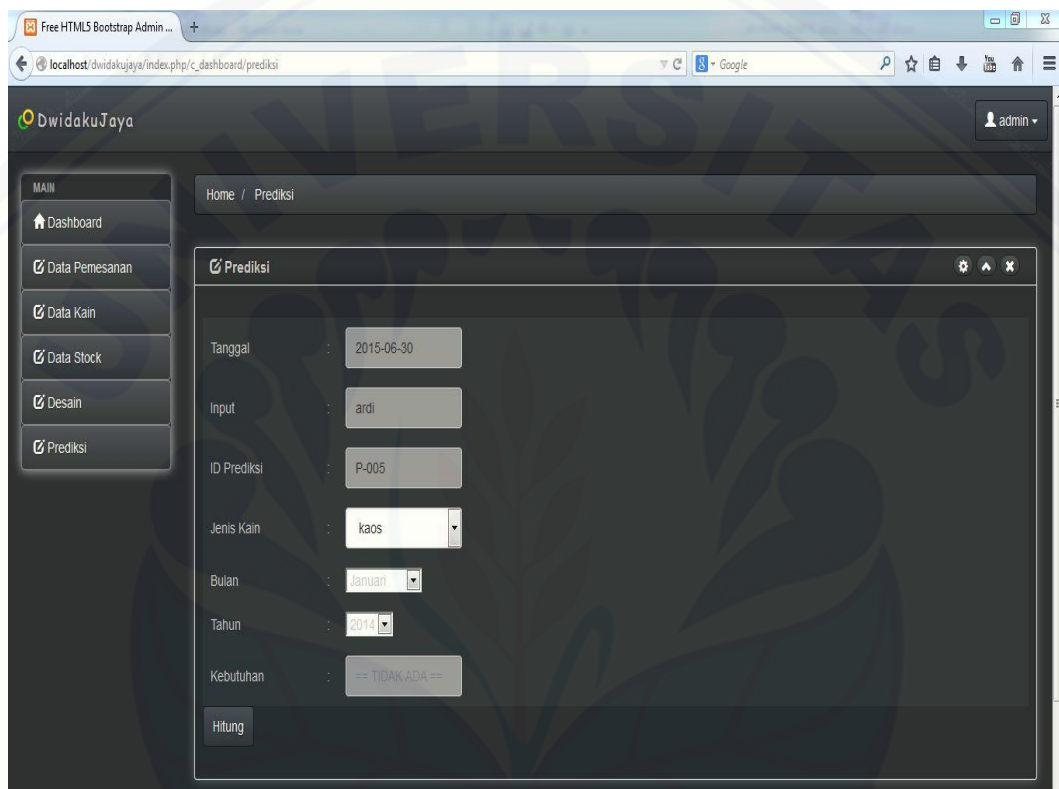
Fitur menghapus data desain pada Aplikasi optimalisasi persediaan bahan baku kain merupakan fitur yang dapat diakses oleh aktor sebagai admin untuk menghapus varian desain yang disediakan oleh perusahaan UD.DwidakuJaya. Halaman untuk menghapus data desain dapat dilihat di Lampiran (Lampiran fitur).

5.4.7 Fitur Manajemen Data Prediksi

Fitur manajemen data prediksi pada aplikasi optimalisasi persediaan bahan baku kain meliputi fitur menambah, menghitung dan menghapus data prediksi

5.4.7.1 Fitur Menambah Data Prediksi

Fitur menambah data prediksi pada Aplikasi optimalisasi persediaan bahan baku kain merupakan fitur yang dapat diakses oleh aktor sebagai admin untuk menghitung prediksi persediaan dan interval pemesanan yang optimal pada periode berikutnya. Halaman untuk menambah data prediksi dapat dilihat di Gambar 5.1



The screenshot shows a web browser window displaying the 'Prediksi' form in the DwidakuJaya application. The browser address bar shows the URL 'localhost/dwidakujaya/index.php/c_dashboard/prediksi'. The application header includes the logo 'DwidakuJaya' and a user profile 'admin'. The left sidebar contains a menu with options: Dashboard, Data Pemesanan, Data Kain, Data Stock, Desain, and Prediksi. The main content area shows the 'Prediksi' form with the following fields and values:

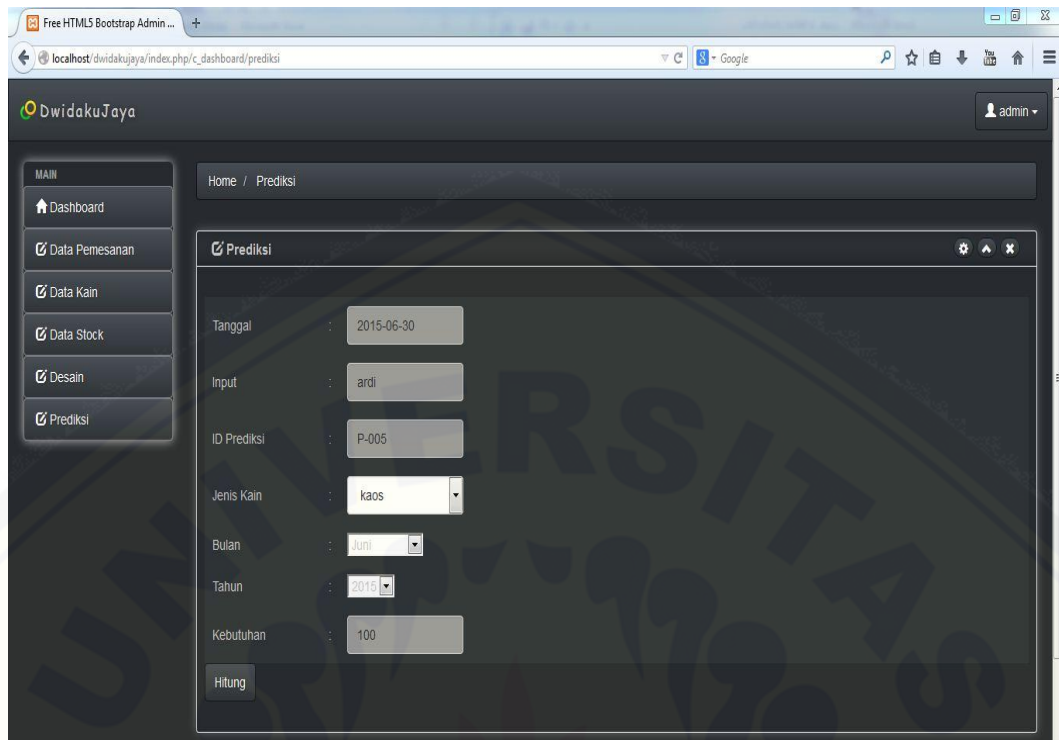
Tanggal	2015-06-30
Input	ardi
ID Prediksi	P-005
Jenis Kain	kaos
Bulan	Juni
Tahun	2014
Kebutuhan	TIDAK ADA

A 'Hitung' button is located at the bottom left of the form area.

Gambar 5.1 Form untuk menambah data prediksi

5.4.7.2 Fitur Menghitung Data Prediksi

Fitur menghapus data prediksi pada Aplikasi optimalisasi persediaan bahan baku kain merupakan fitur yang dapat diakses oleh aktor sebagai admin untuk menghitung persediaan dan interval pemesanan yang optimal pada periode berikutnya. Halaman menghitung prediksi persediaan dan interval yang optimal dapat dilihat pada Gambar 5.2



The screenshot displays a web browser window with the URL `localhost/dwidakujaya/index.php/c_dashboard/prediksi`. The application header shows the logo 'DwidakuJaya' and a user profile 'admin'. A sidebar menu on the left lists navigation options: Dashboard, Data Pemesanan, Data Kain, Data Stock, Desain, and Prediksi. The main content area is titled 'Prediksi' and contains a form with the following fields:

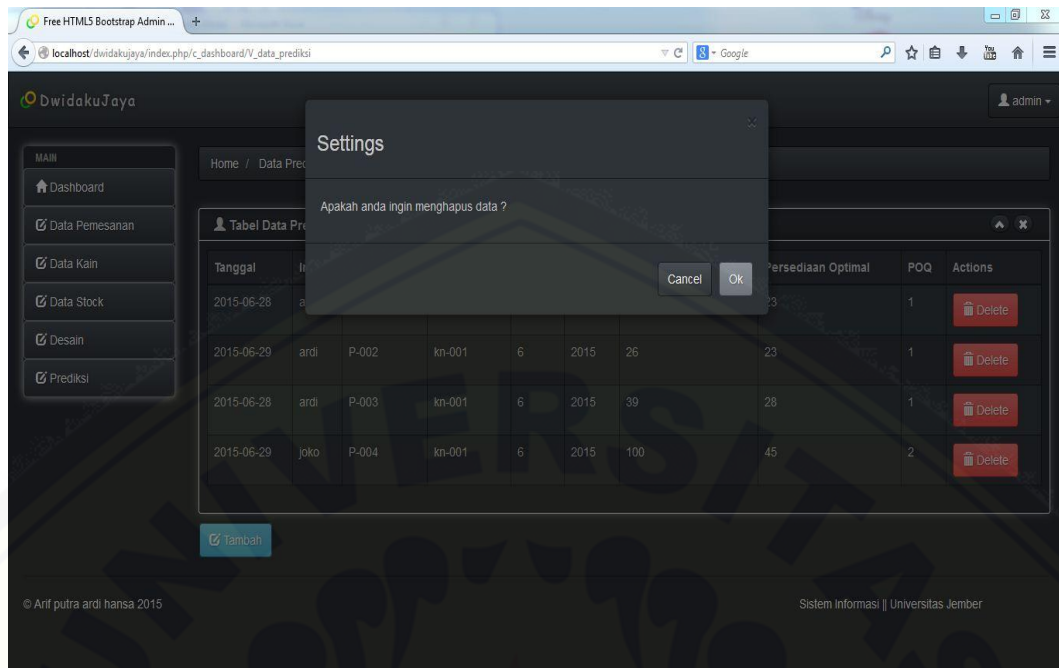
Tanggal	2015-06-30
Input	ardi
ID Prediksi	P-005
Jenis Kain	kaos
Bulan	Juni
Tahun	2015
Kebutuhan	100

A 'Hitung' button is located at the bottom of the form.

Gambar 5.2 Menghitung data prediksi

5.4.7.3 Fitur Menghapus Data Prediksi

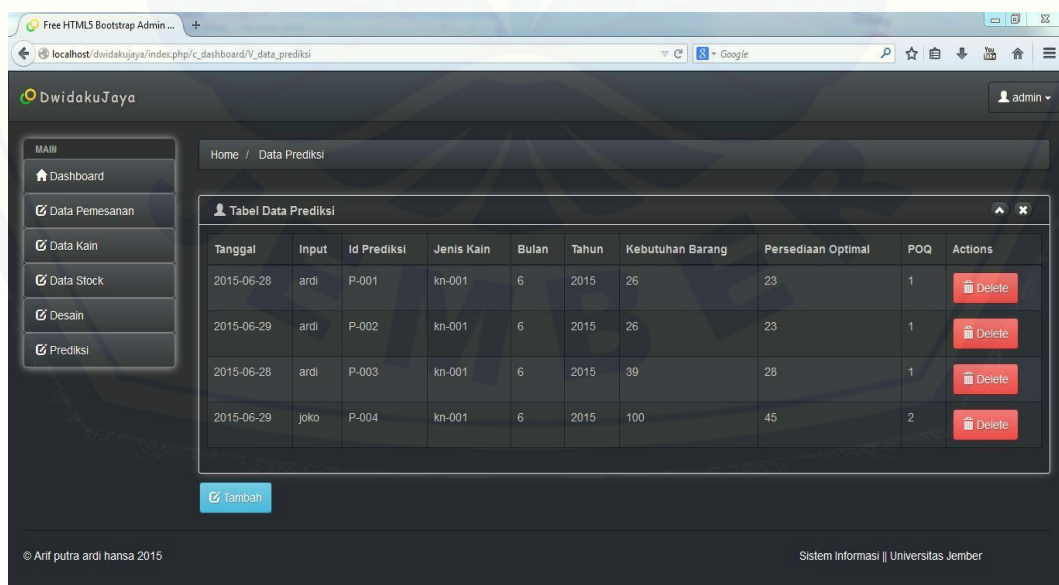
Fitur menghapus data prediksi pada Aplikasi optimalisasi persediaan bahan baku kain merupakan fitur yang dapat diakses oleh aktor sebagai admin untuk menghapus data prediksi yang telah tersimpan di Aplikasi optimalisasi persediaan bahan baku kain . Halaman untuk menghapus data prediksi dapat dilihat di Gambar 5.3



Gambar 5.3 Menghapus data prediksi

5.4.7.4 Fitur Melihat data prediksi

Fitur melihat data prediksi pada Aplikasi optimalisasi persediaan bahan baku kain merupakan fitur yang dapat diakses oleh aktor sebagai admin dan pemilik untuk dapat melihat data prediksi yang telah tersimpan di Aplikasi optimalisasi persediaan bahan baku kain . Halaman untuk melihat data prediksi dapat dilihat di Gambar 5.4



Gambar 5.4 Melihat data prediksi

5.5 Pembahasan

Aplikasi optimalisasi persediaan bahan baku kain dengan menggunakan metode *Period Order Quantity* (POQ) ini bertujuan untuk membantu dan mempermudah mengendalikan persediaan bahan baku untuk produksi agar persediaan yang tersedia optimal. Untuk menerapkan metode POQ hal yang harus diperhatikan yaitu, jumlah pemakaian atau kebutuhan bahan baku kain pada periode sebelumnya. Hal yang perlu diperhatikan juga adalah biaya pemesanan dalam sekali pesan serta biaya penyimpanan yang dikeluarkan untuk menyimpan bahan baku tersebut. Data tersebut digunakan untuk menghitung interval pemesanan yang optimal sehingga dapat mengurangi biaya persediaan yang dikeluarkan oleh perusahaan. Aplikasi optimalisasi persediaan bahan baku kain ini terdapat beberapa kelebihan dan kekurangan yang masih harus disempurnakan kembali, untuk mencapai tujuan tersebut aplikasi ini memiliki fitur utama yaitu fitur prediksi, dimana dalam fitur tersebut metode POQ diterapkan sehingga perhitungan lebih cepat diketahui.

Fungsi control dilakukan oleh manajemen UD.Dwidaku Jaya, untuk menampilkan prediksi jumlah kebutuhan bahan baku dan interval pemesanan yang optimal. Aplikasi dapat mencatat user sesuai data login yang terdaftar. Kemampuan aplikasi dalam mencapai tujuan yang telah ditentukan di awal masih terdapat banyak bagian yang harus diperbaiki. Dalam proses pengujian, aplikasi ini termasuk aplikasi yang sederhana sehingga masih dapat dikembangkan. Untuk lebih memanfaatkan aplikasi optimalisasi persediaan bahan baku ini, maka manajemen memberi saran untuk menambah fitur untuk menghitung re-order point serta keuntungan yang didapat perusahaan.

BAB 6. PENUTUP

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

6.1 Kesimpulan

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

1. Pembuatan aplikasi optimalisasi persediaan bahan baku kain dengan menggunakan metode *PERIOD ORDER QUANTITY* (POQ) dapat membantu manajemen perusahaan untuk mengendalikan persediaan bahan baku kain dengan cara mengolah data pemesanan yang nantinya akan di jumlahkan selama satu periode. Sehingga didapatkan jumlah kebutuhan bahan baku kain dalam satu periode. Setelah itu untuk memperoleh interval pemesanan yang optimal menggunakan metode POQ maka data kebutuhan kain dihitung dengan biaya pemesanan dalam sekali pesan serta biaya penyimpanan.
2. Aplikasi optimalisasi persediaan bahan baku kain dibangun dan dirancang menggunakan Bahasa pemrograman PHP (*Page Hypertext Pre-Processor*) dengan *Framework Code Igniter* yang terintegrasi dengan *database* menggunakan MySQL dan *tool* XAMPP. Aplikasi ini memiliki fitur utama yaitu fitur prediksi yang dapat menampilkan jumlah pesanan dan interval pemesanan yang optimal pada periode berikutnya, sehingga jumlah kain yang dipesan tidak berlebihan dan sesuai perhitungan pada satu periode.

6.2 Saran

Fitur yang terdapat pada optimalisasi persediaan bahan baku kain ini hanyalah fitur dasar yang mengelola data pelanggan, data pemesanan, data desain, data kain, data prediksi dan data stock. Pengembangan Aplikasi yang disarankan oleh peneliti, diantaranya adalah :

1. Penghematan Biaya

Semua proses pemesanan dan penanganan pelanggan dapat berorientasi pada biaya yang dimunculkan dan diikuti sertakan dalam data keuangan perusahaan. Presentase penghematan biaya yang dikeluarkan perusahaan untuk mengetahui besar penghematan biaya jika menggunakan metode POQ

2. Penjadwalan Pemesanan Ulang

Pemesanan ulang dengan metode *Periode Order Quantity* dapat menerapkan dasar perhitungan POQ yaitu dengan metode EOQ sehingga penjadwalan pemesanan ulang dapat diketahui.

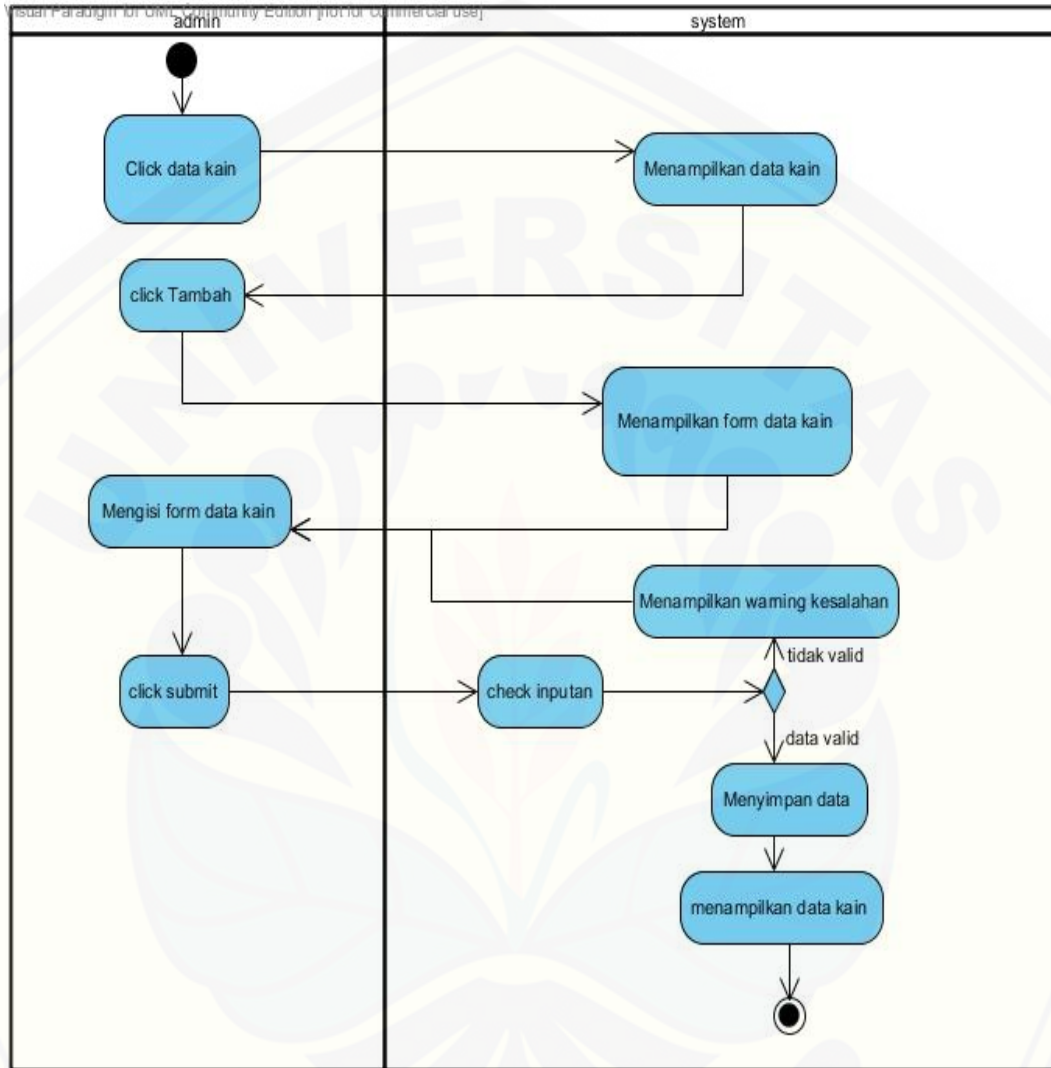
DAFTAR PUSTAKA

- Assauri, S. (2000). *Manajemen Produksi dan Operasi*. Jakarta: FEUI.
- Baroto. (2002). *Perencanaan dan Pengendalian Produksi*. Jakarta: BPFE UGM.
- Handoko, T. (1999). *Dasar-dasar manajemen produksi dan operasi*. Yogyakarta: BPFE-UGM.
- Lestrai, M., & Kusno, K. (2014). Kajian manajemen persediaan buah naga merah (*Hylocereus polyrhizus* sp.) dalam memenuhi permintaan konsumen. *Agric Sci.J.*, 225-234.
- Marimin. (2005). *Teori dan Aplikasi Aplikasi Pakar dalam Teknologi Manajerial*. Bogor: IPB Press.
- presman, r. (2005). *Software Engineering: a Practitioner's Approach-6th ed.* new york: McGraw-Hill.
- Pressman, r. (2010). *Software Engineering: a Practitioner's Approach-6th ed.* New York: McGraw-Hill.
- Rahmayanti, D., & Fauzan, A. (2013). Optimalisasi Aplikasi persediaan bahan baku karet mentah (lateks) dengan metode lot sizing. *issn 2088-4842*, 317-325.
- rangkuti, f. (2004). *Manajemen Persediaan: Aplikasi di Bidang Bisnis*. jakarta: Grafindo Persada.
- Rangkuti, f. (2007). *manajemen persediaan*. jakarta: PT. RajaGrafindo Persada.
- Sartono, R. A. (2001). *Manajemen Keuangan (Teori dan Aplikasi), Edisi Keempat, Cetakan Ketujuh*. Yogyakarta: BPFE Yogyakarta.
- Sommerville, I. (2001). *Software Engineering*. Addison Wesley.

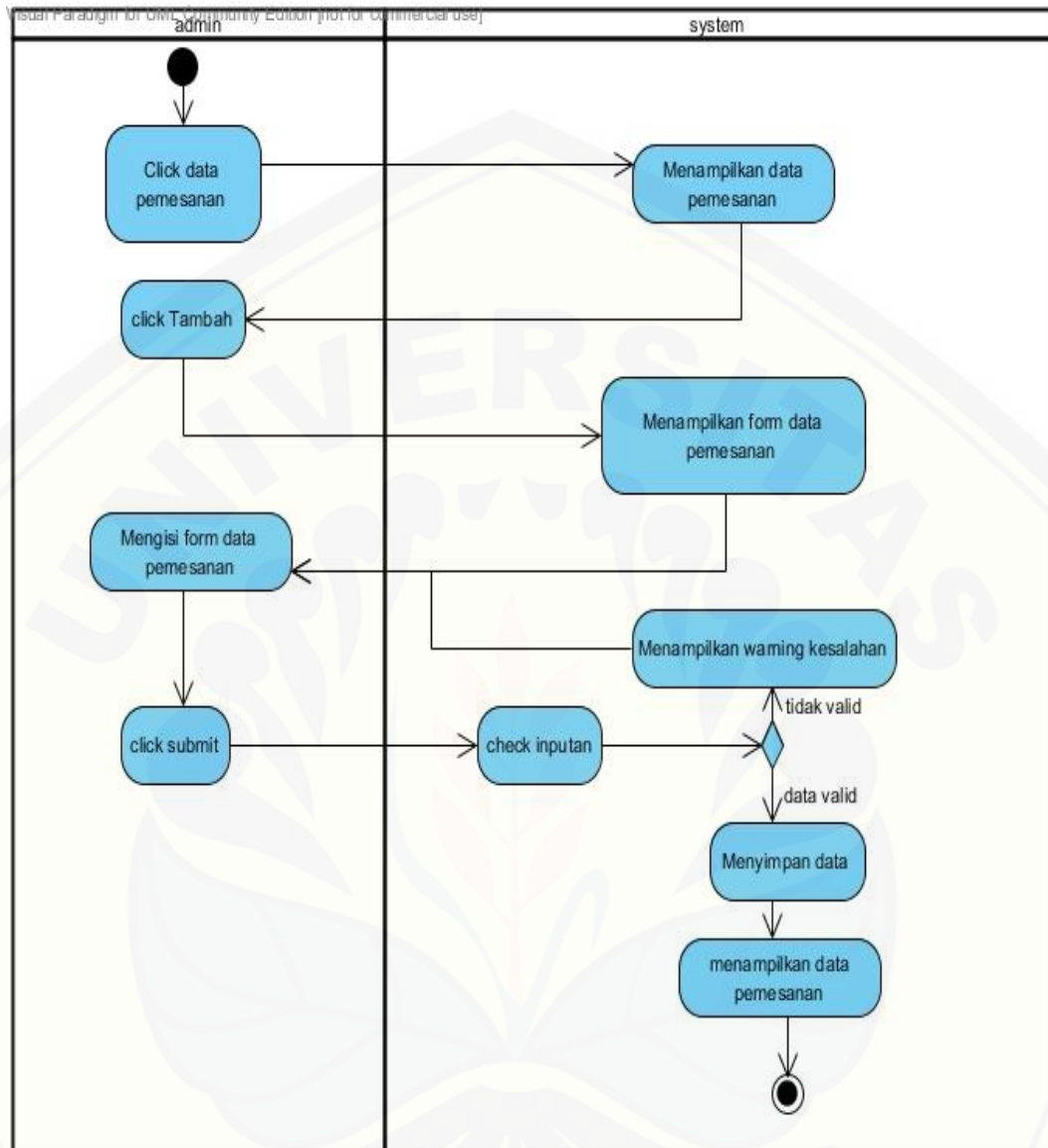
LAMPIRAN

1. Lampiran *activity*

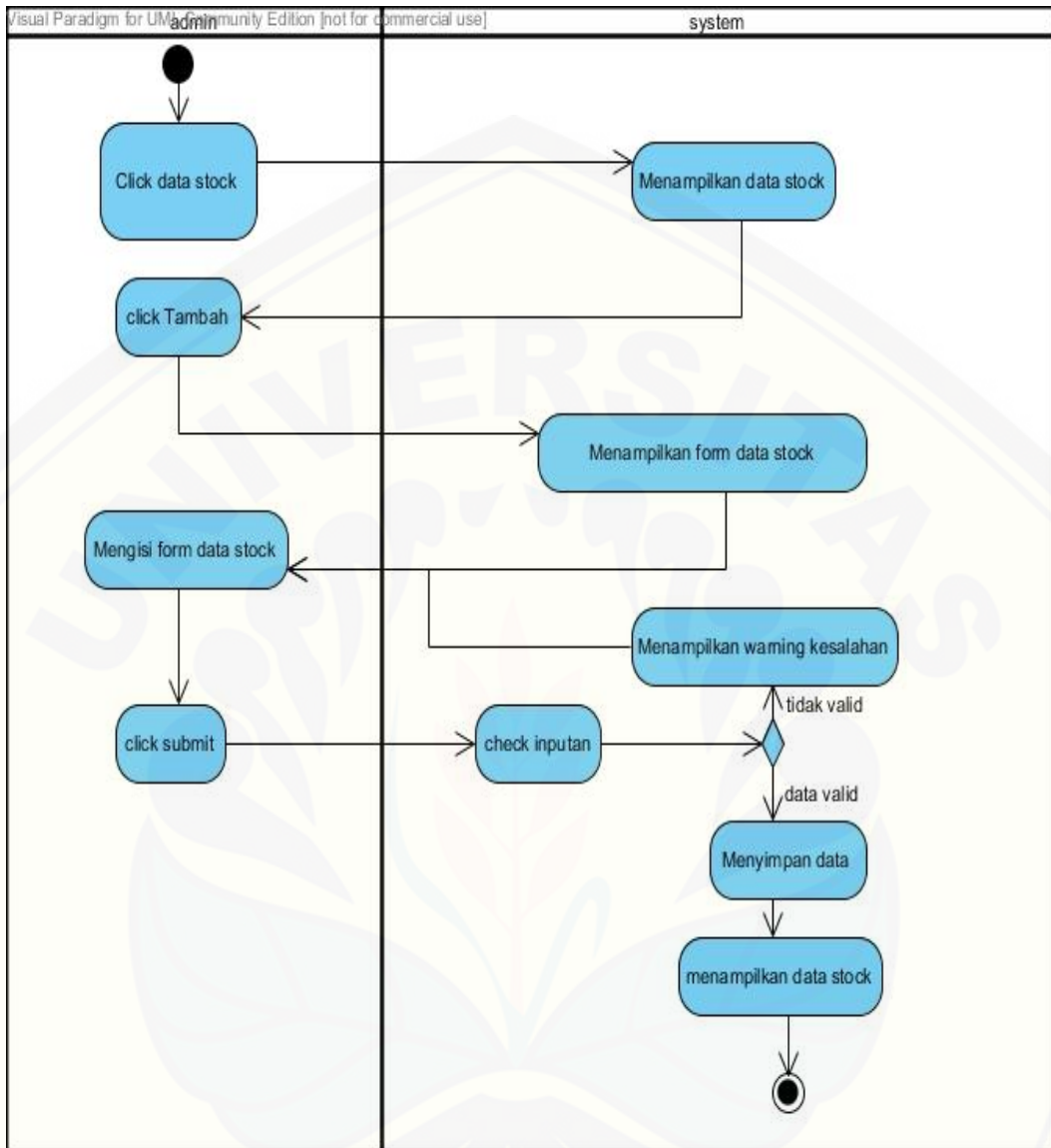
1.1 Activity Diagram *insert* Data Kain



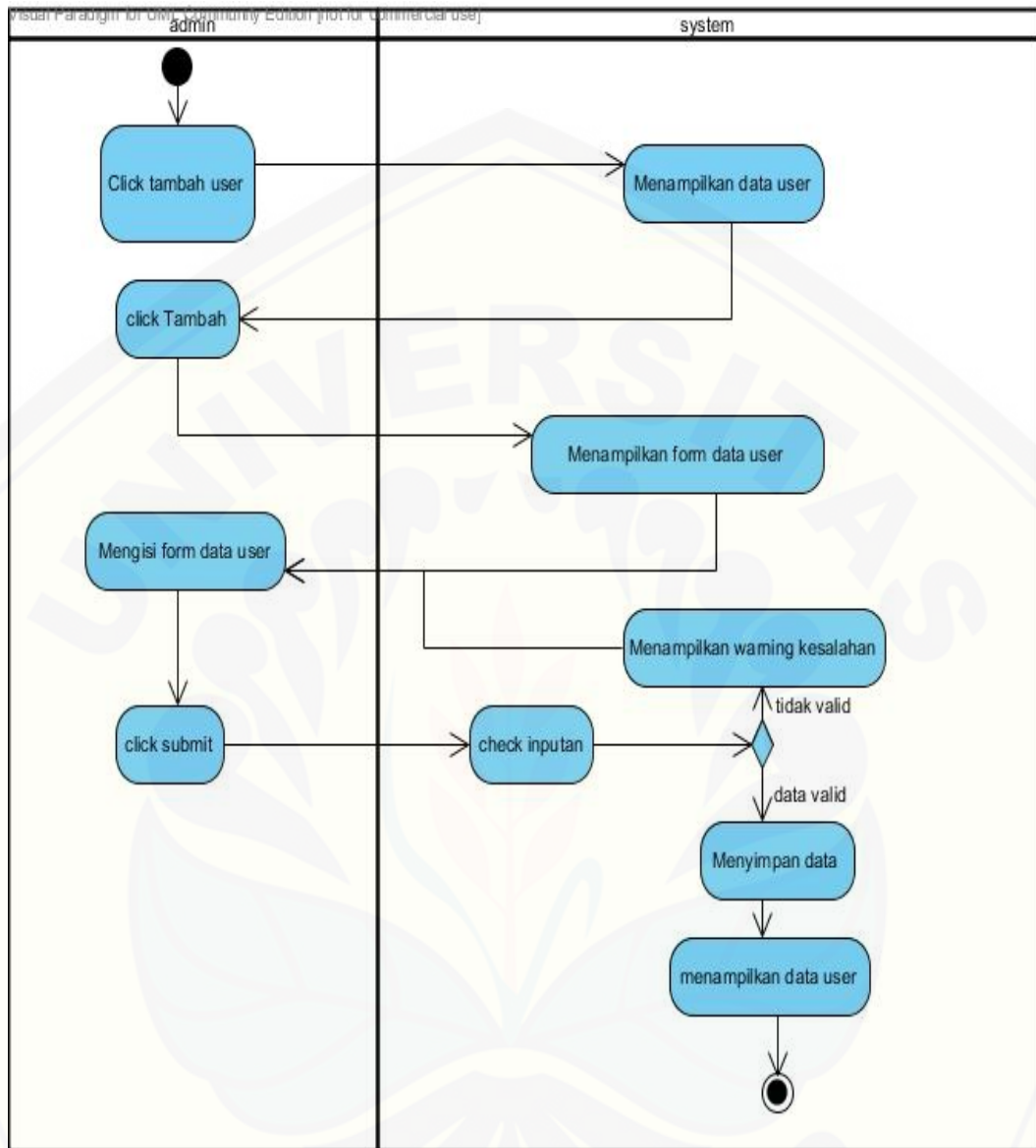
1.2 Activity Diagram *insert* Data Pemesanan



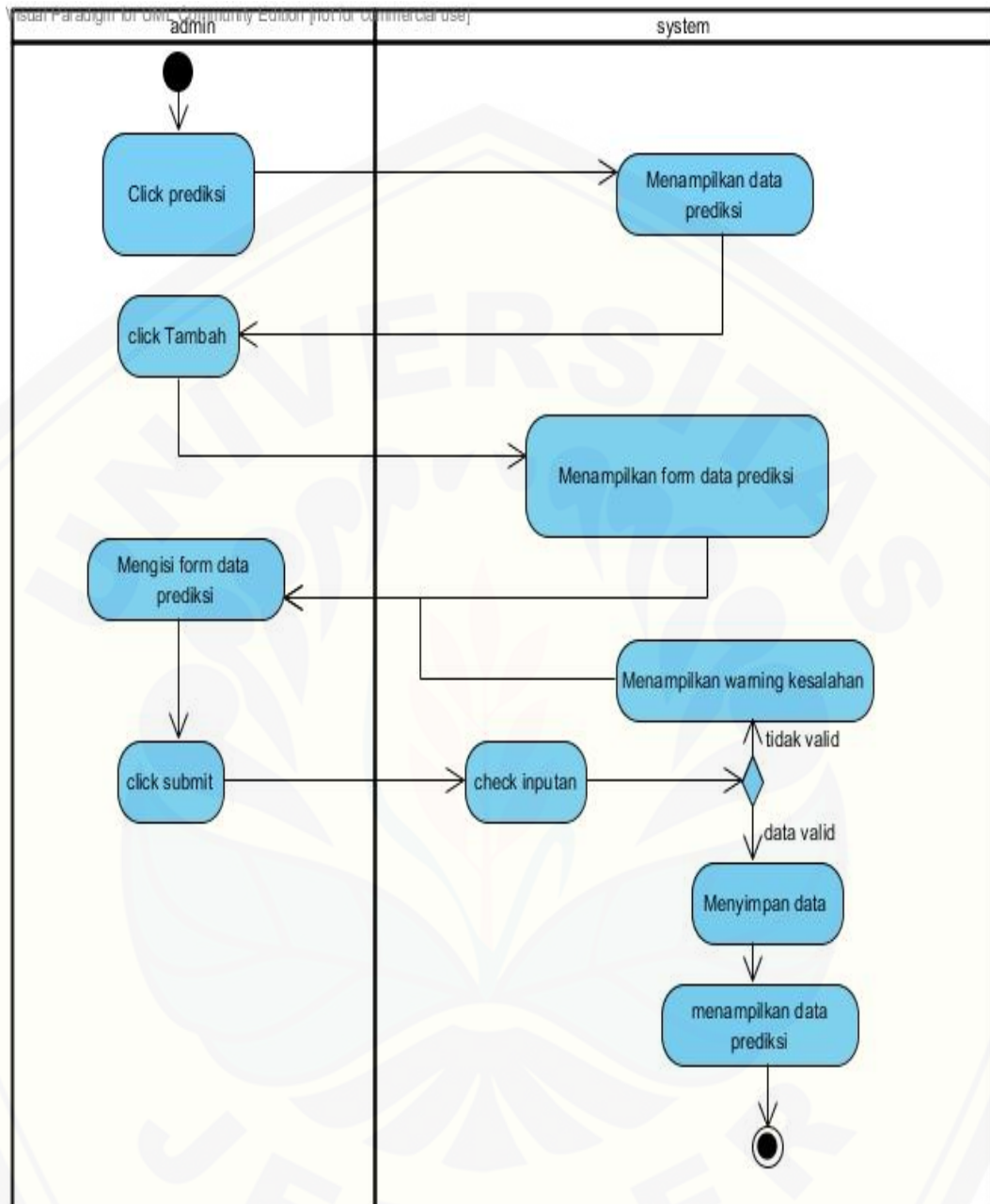
1.3 Activity Diagram insert Data Stock



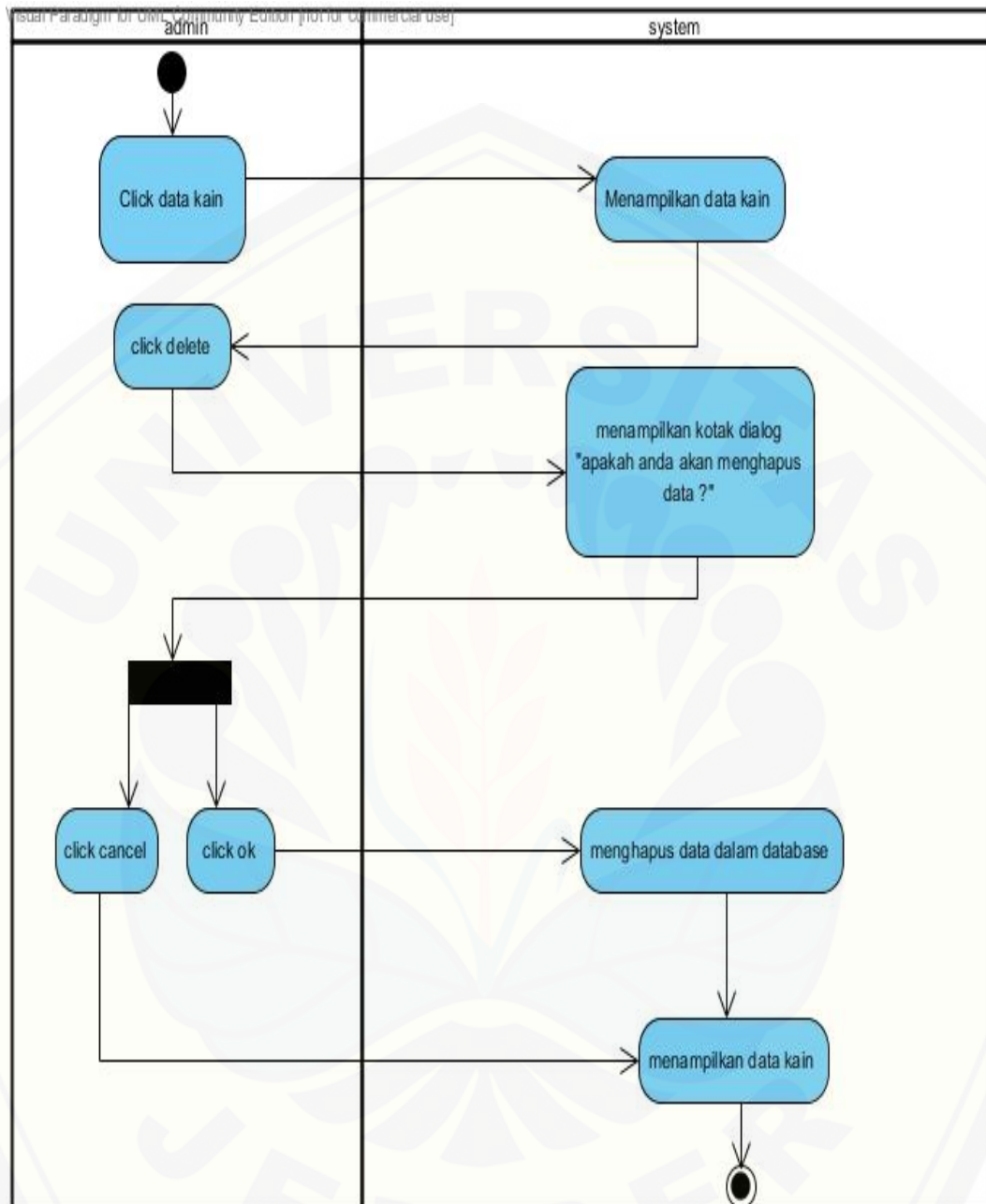
1.4 Activity Diagram insert Data user



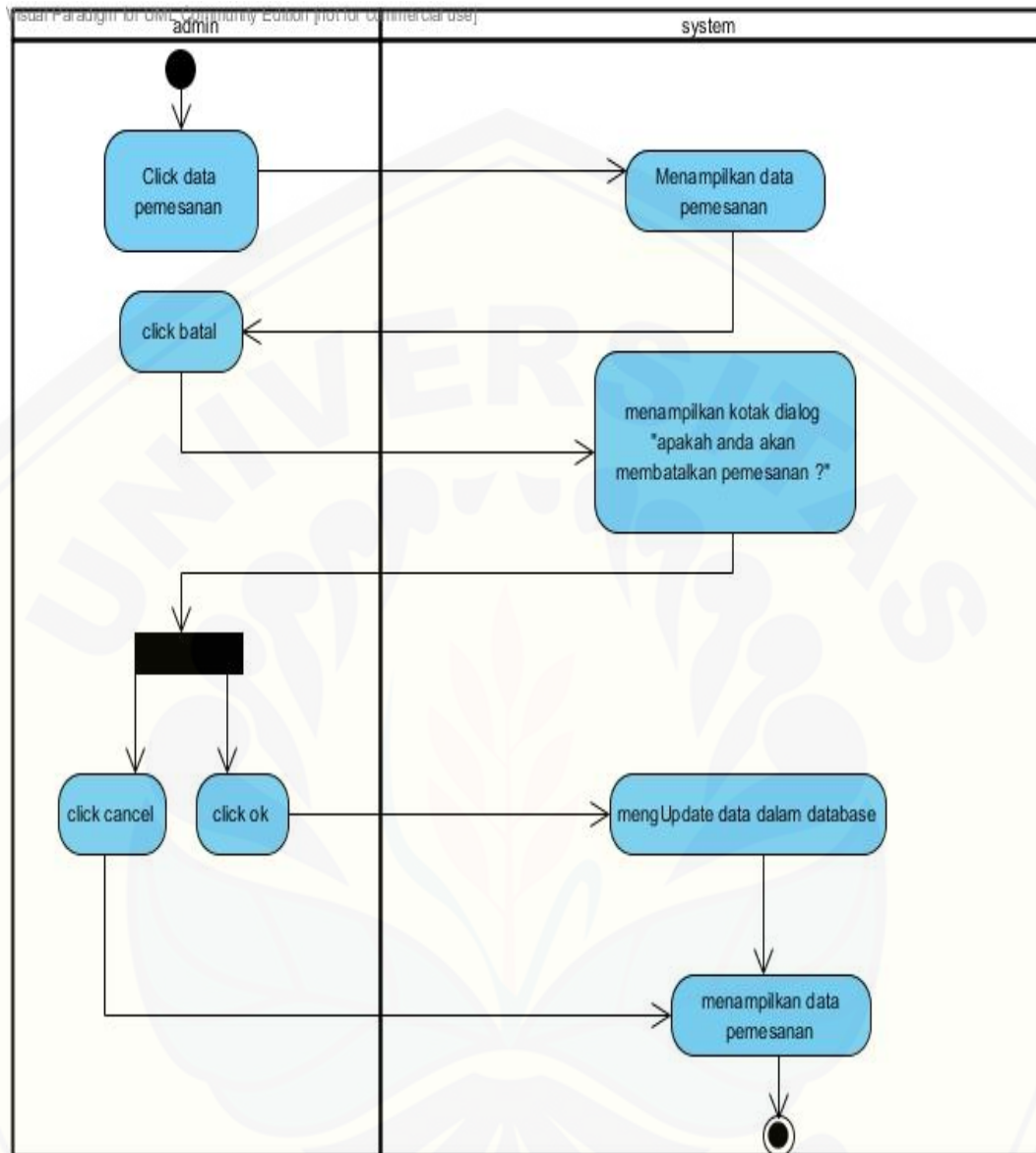
1.5 Activity Diagram *insert* Data Prediksi



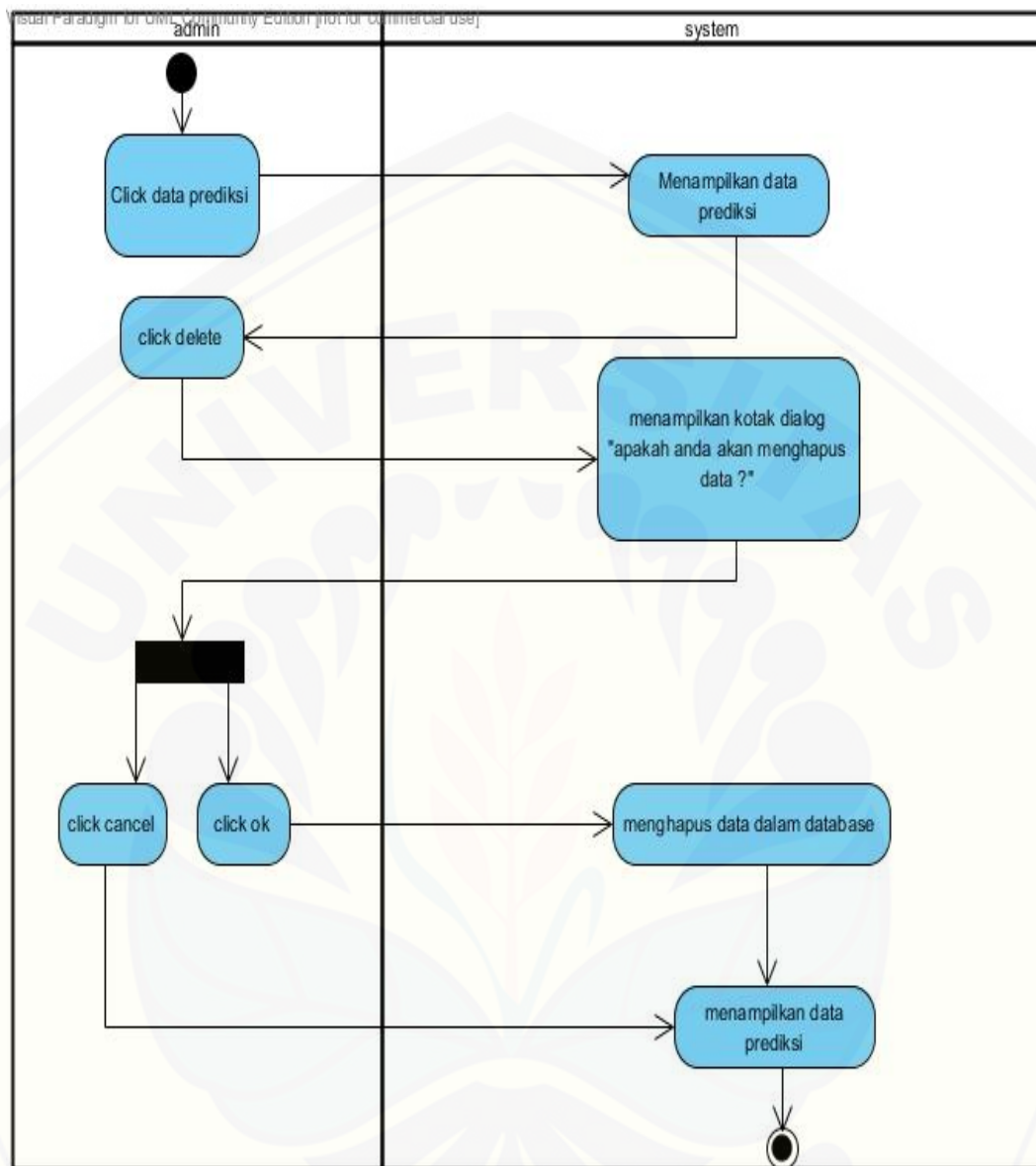
1.6 Activity Diagram Delete Data Kain



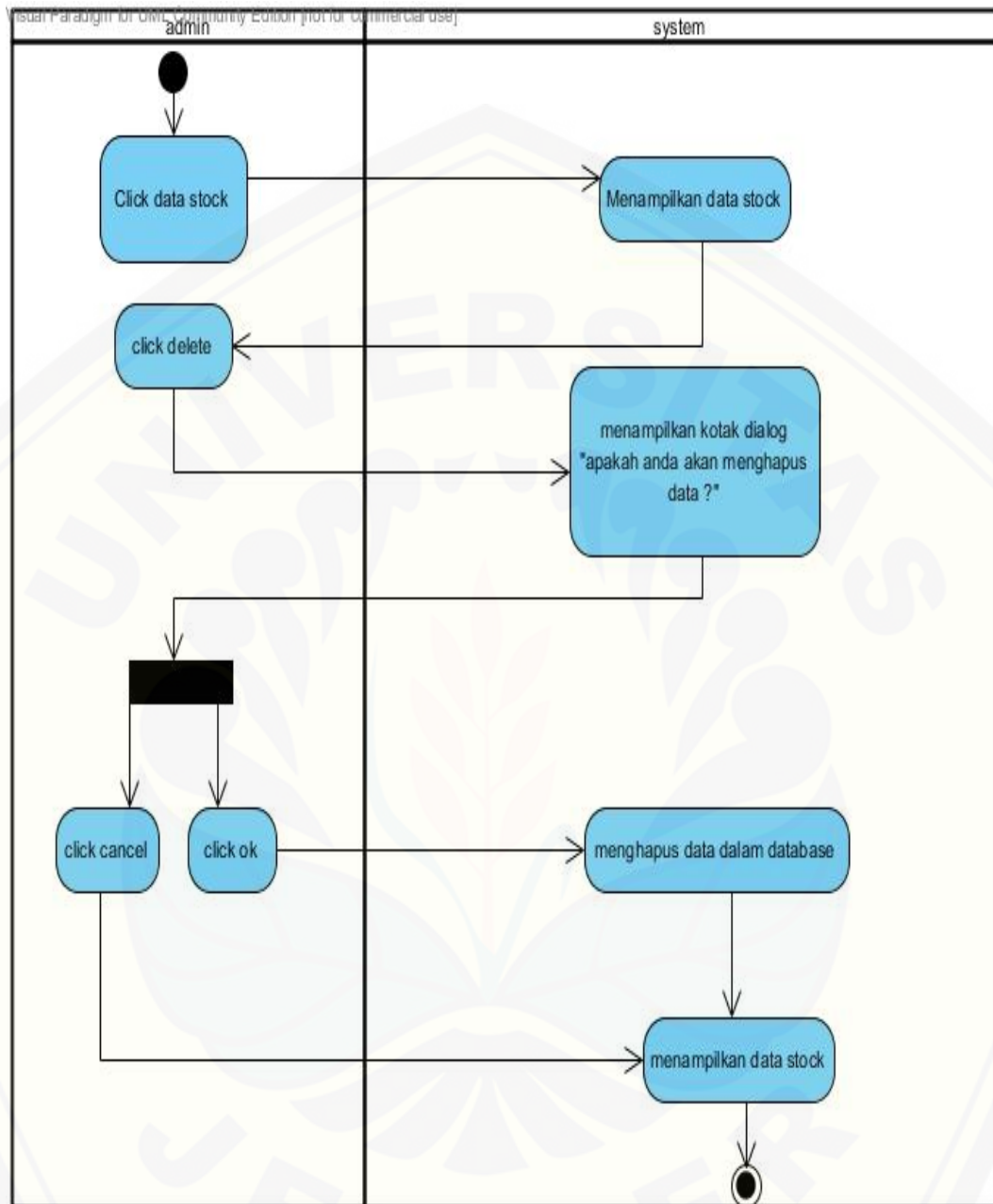
1.7 Activity Diagram Delete Data Pemesanan



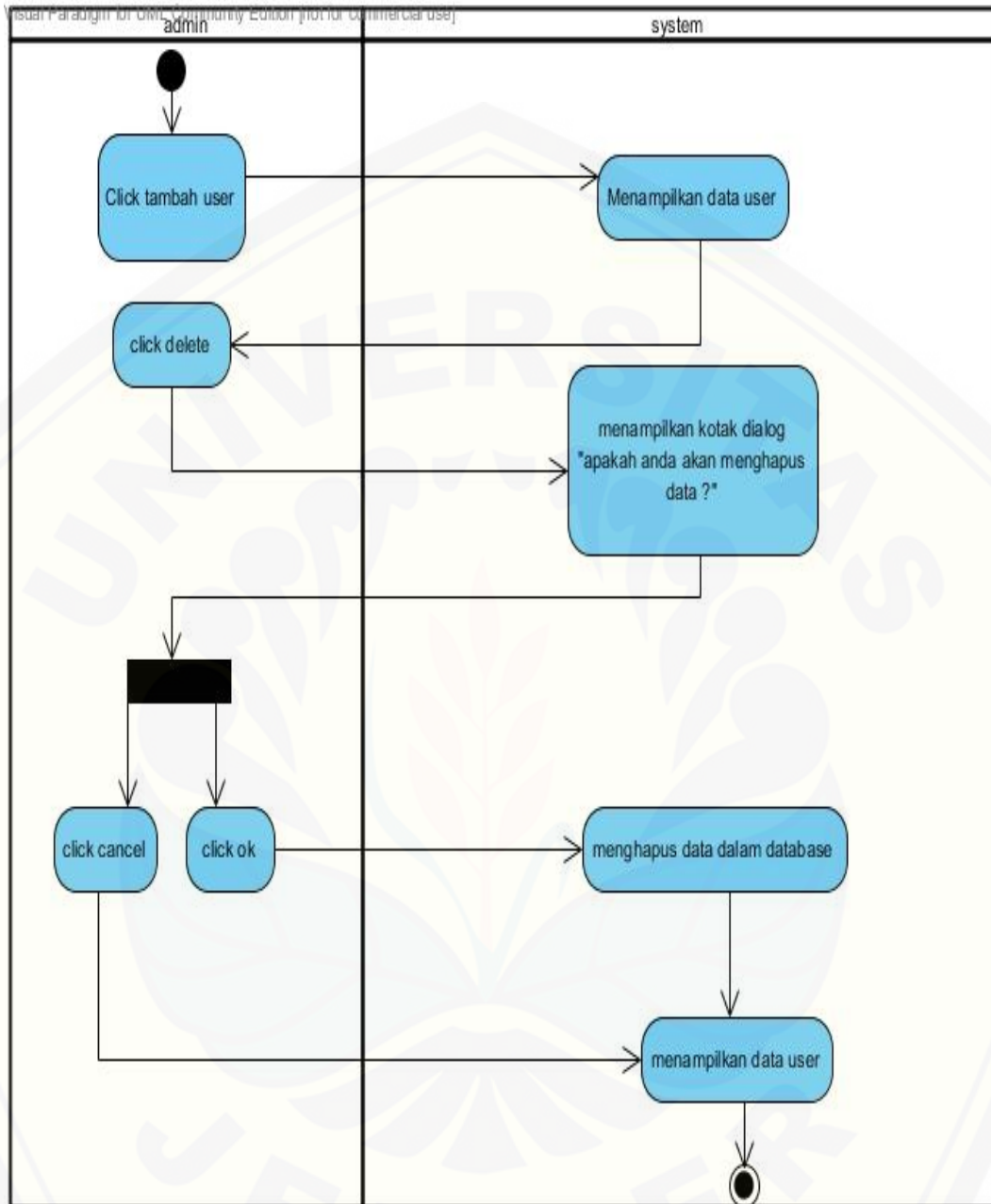
1.8 Activity Diagram Delete Data Prediksi



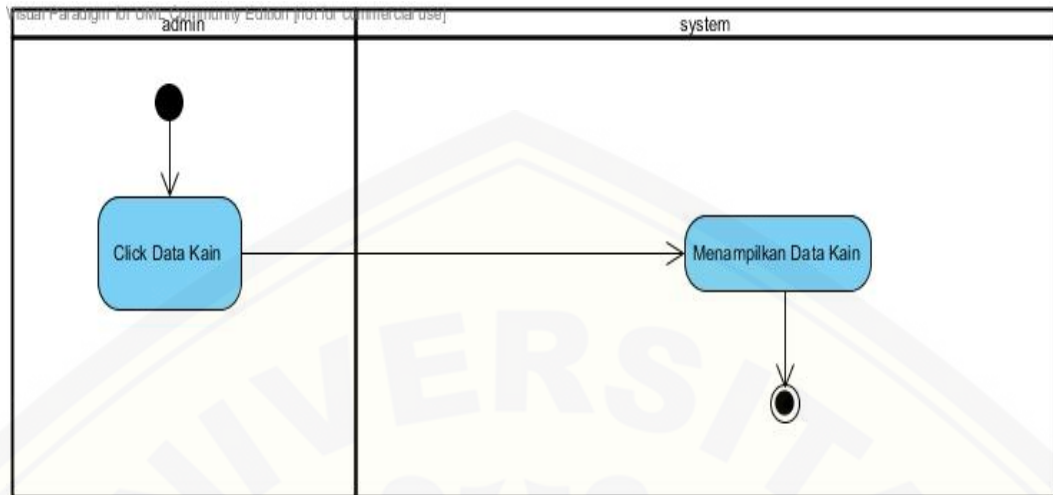
1.9 Activity Diagram Delete Data Stock



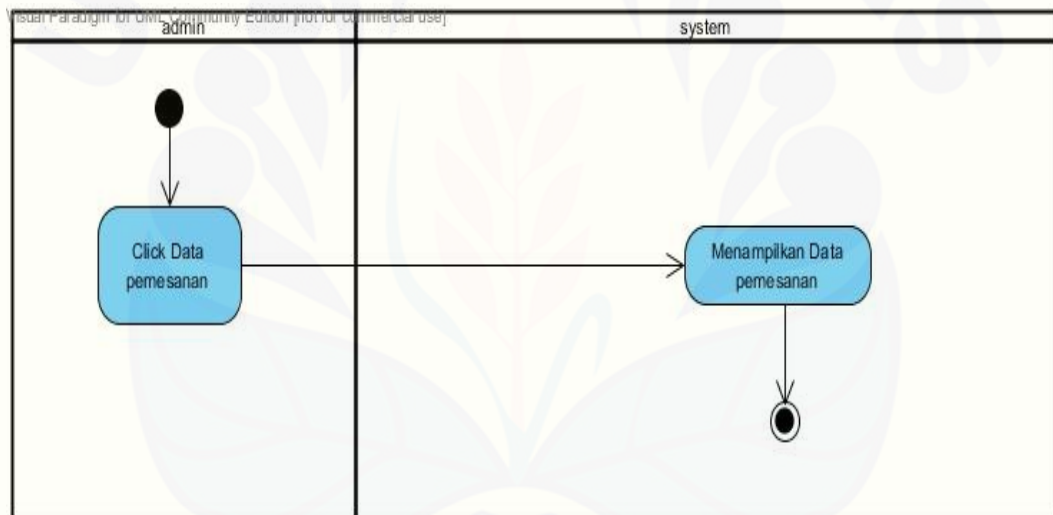
1.10 Activity Diagram Delete Data user



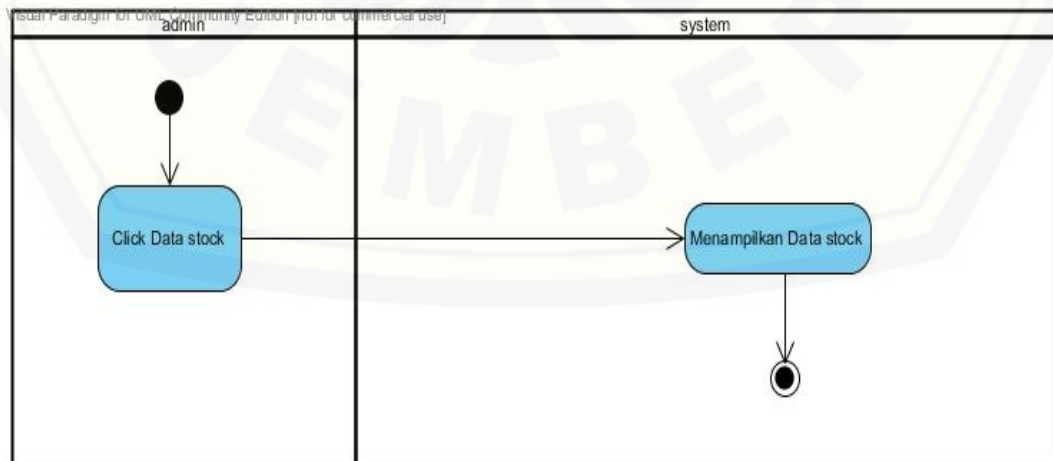
1.11 Activity Diagram View Data Kain

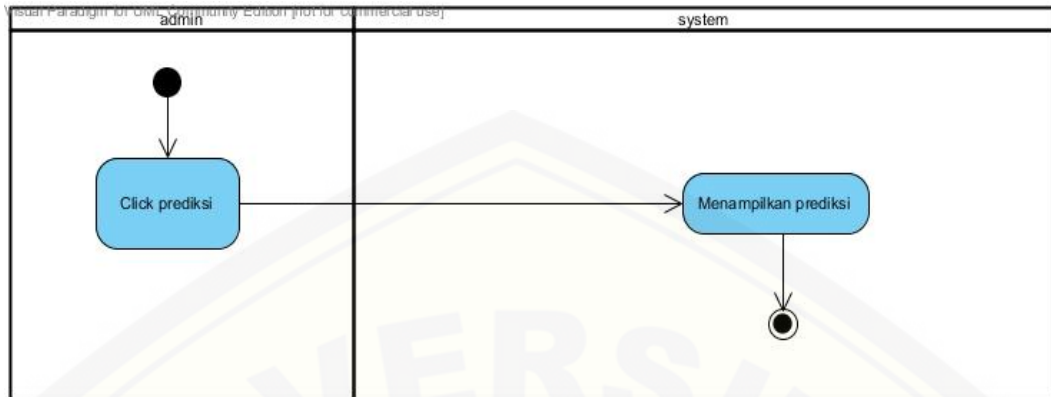
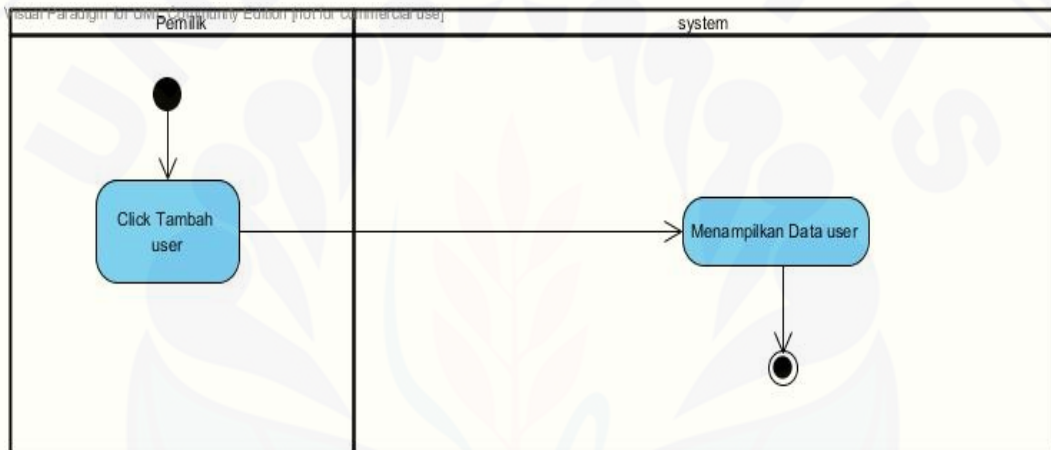


1.12 Activity Diagram View Data Pemesanan



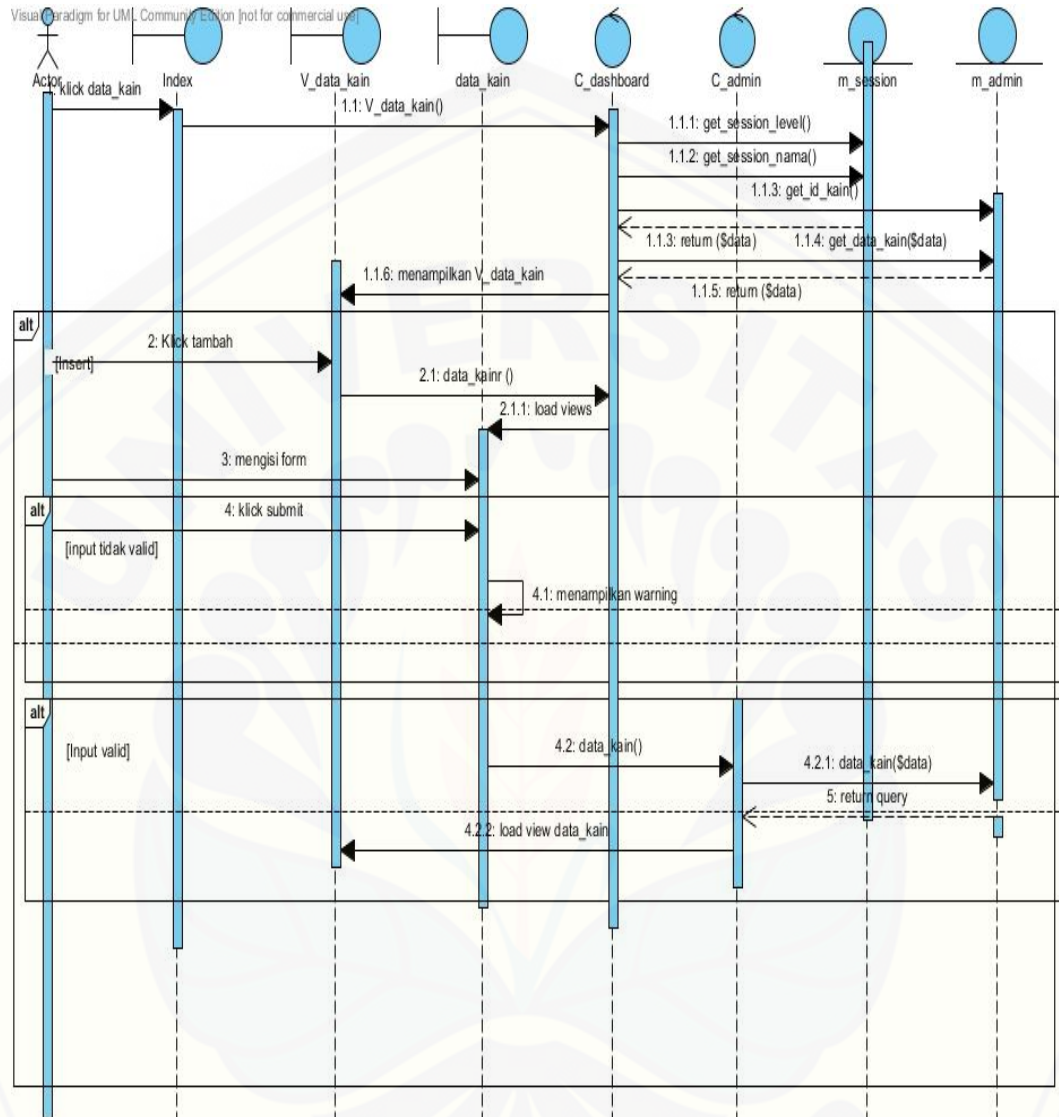
1.13 Activity Diagram View Data Stock



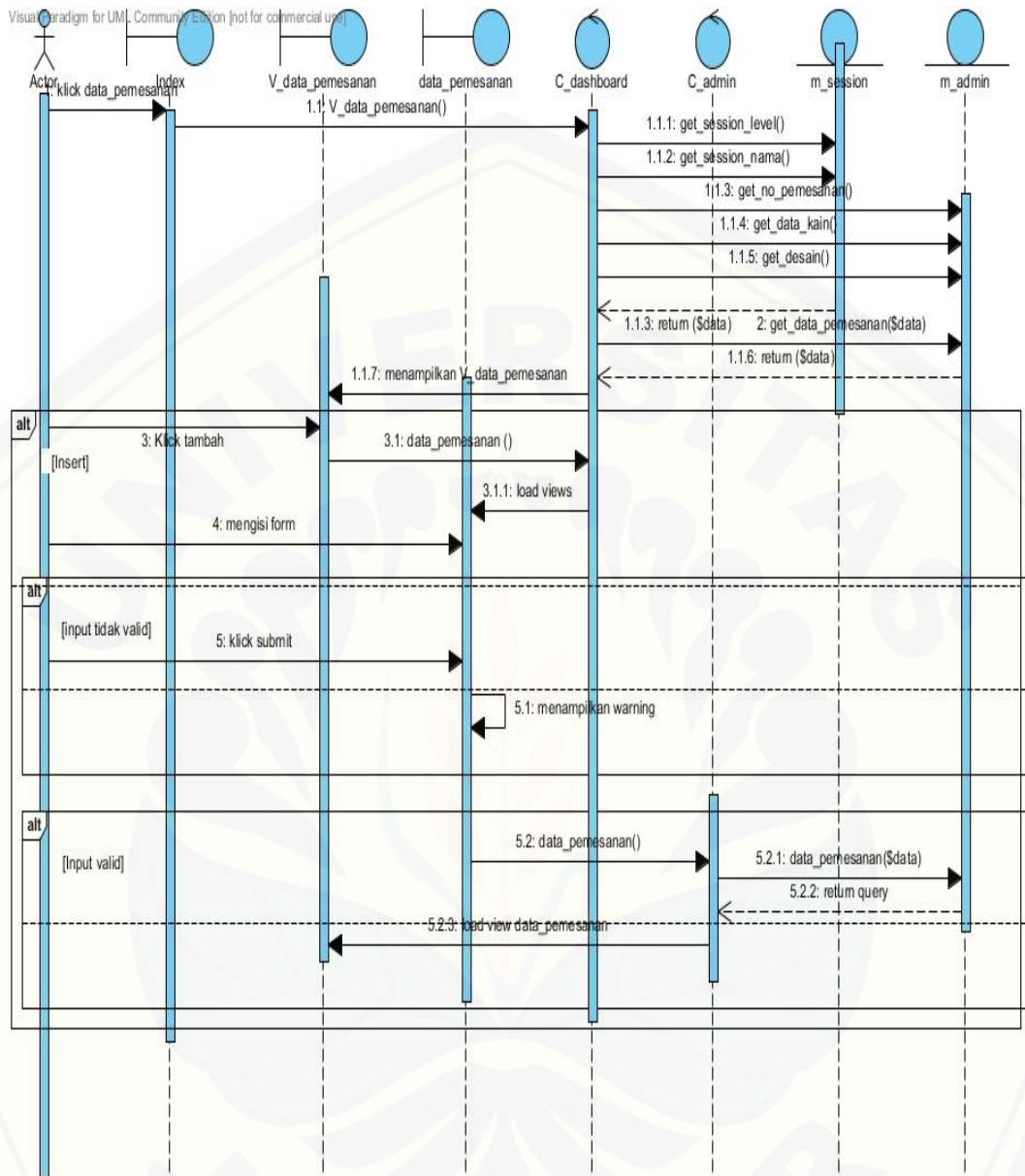
1.14 *Activity Diagram View Data Prediksi*1.15 *Activity Diagram View Data user*

2. Lampiran Sequence

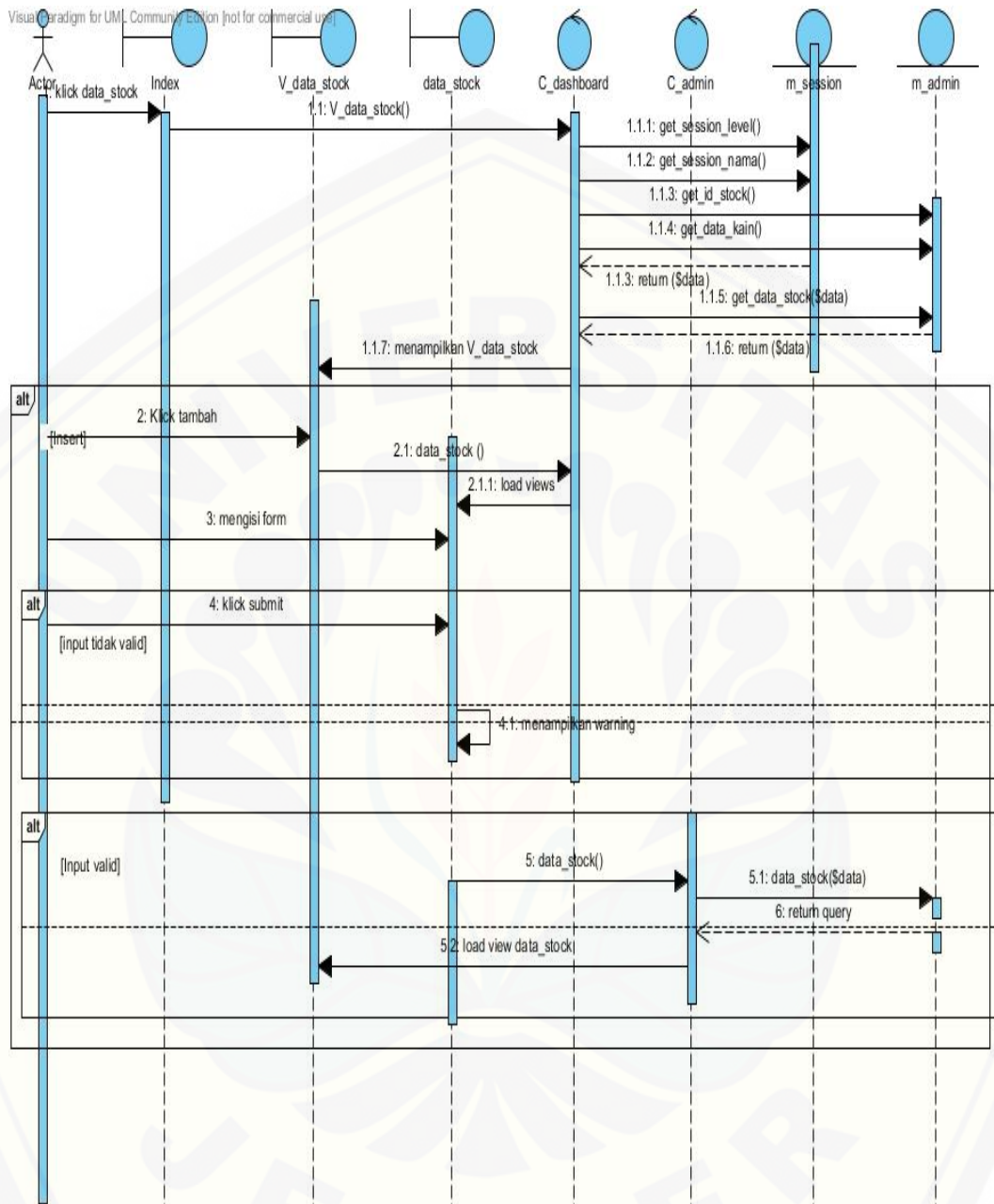
2.1 Sequence Diagram insert Data Kain



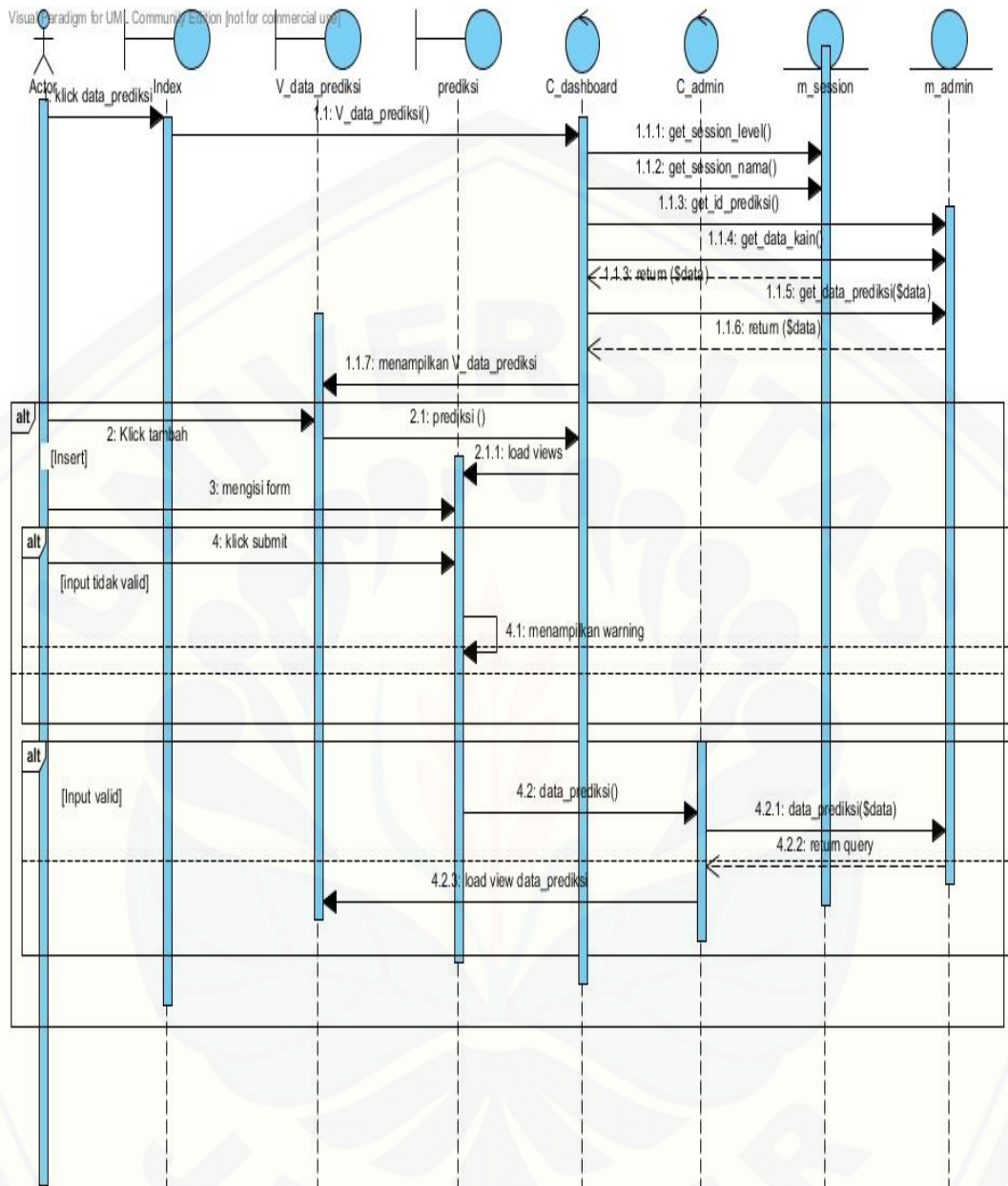
2.2 Sequence Diagram *insert* Data Pemesanan



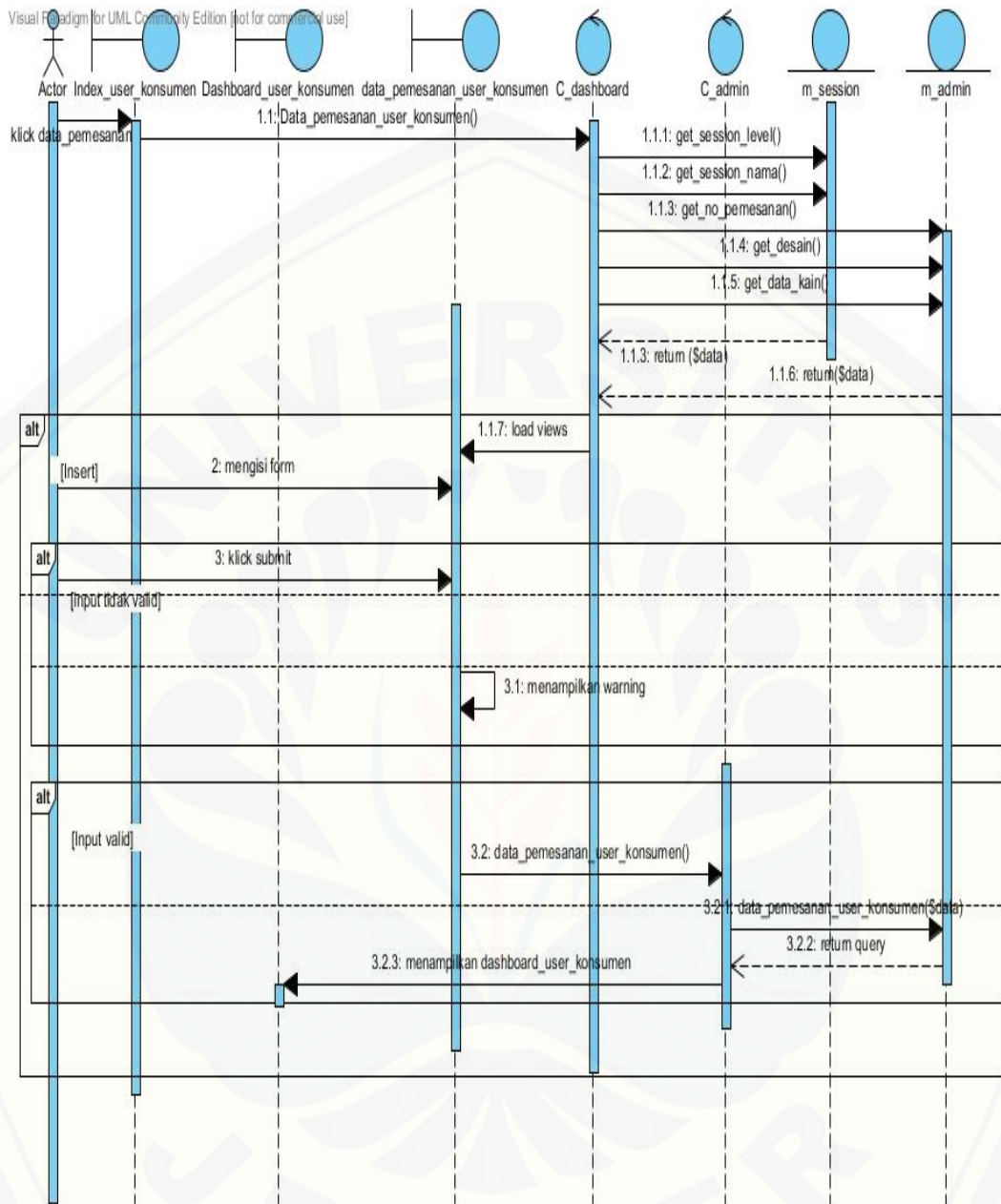
2.3 Sequence Diagram *insert Data Stock*



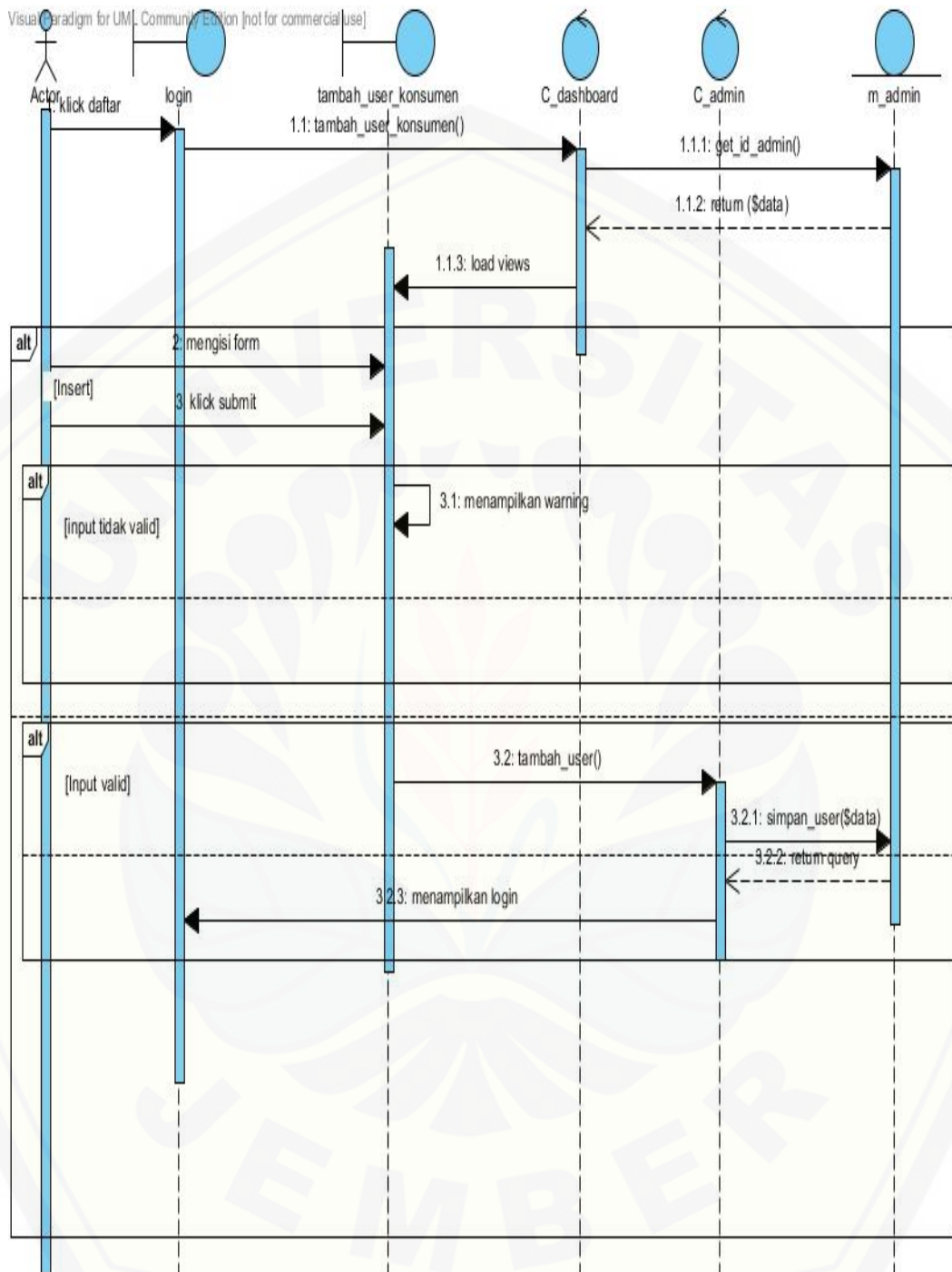
2.4 Sequence Diagram *insert* Data Prediksi



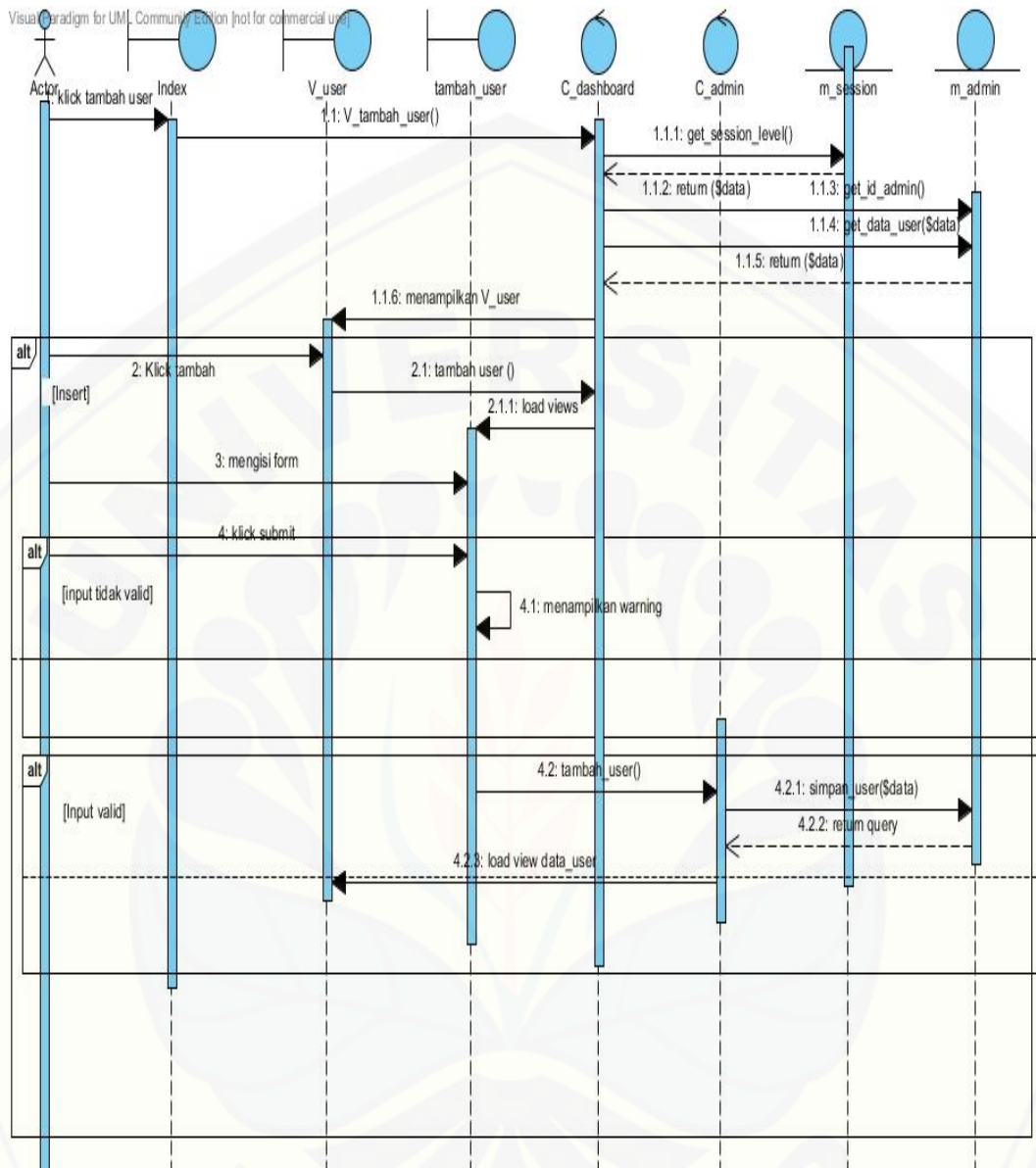
2.5 Sequence Diagram *insert* Data Pemesanan User Konsumen



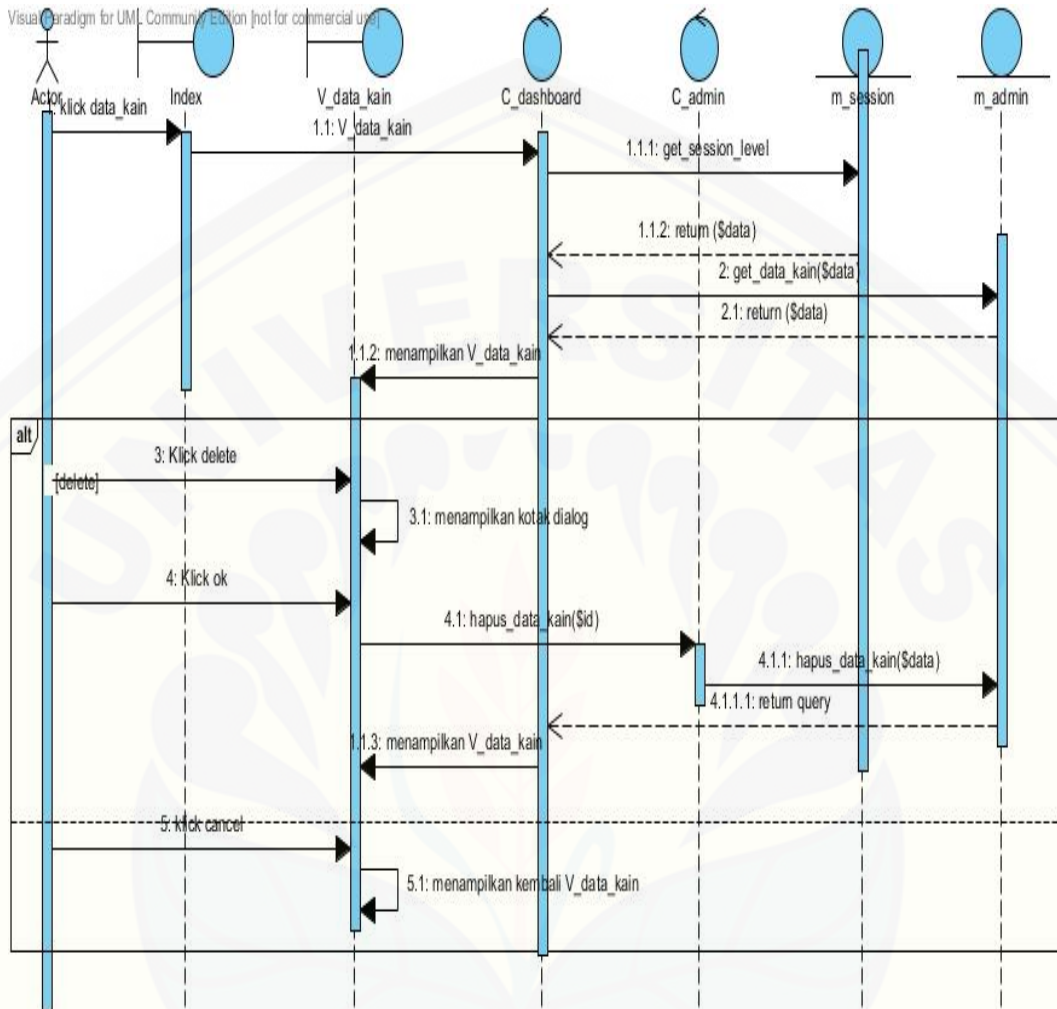
2.6 Sequence Diagram *insert Data User Konsumen*



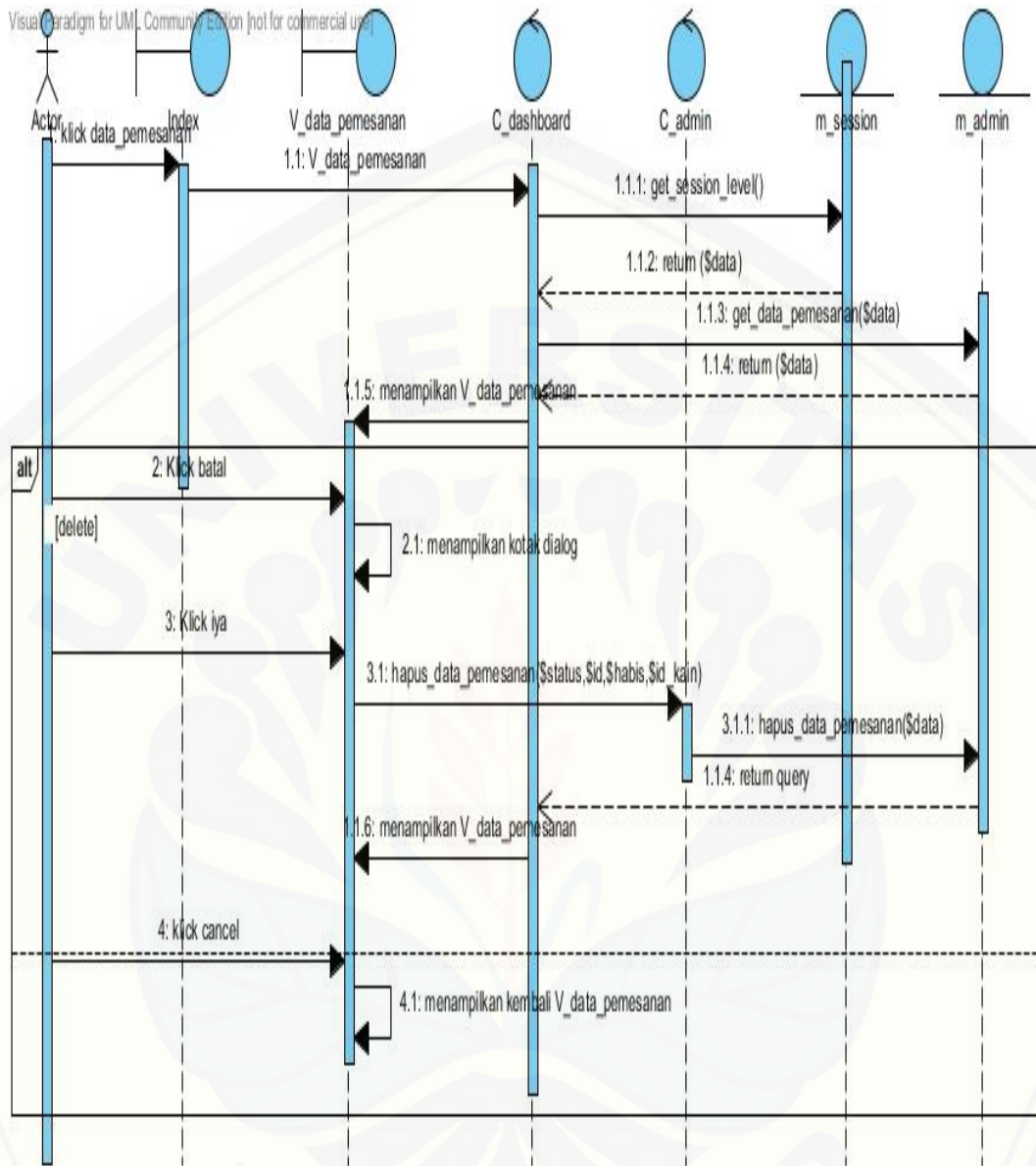
2.7 Sequence Diagram *insert Data User*



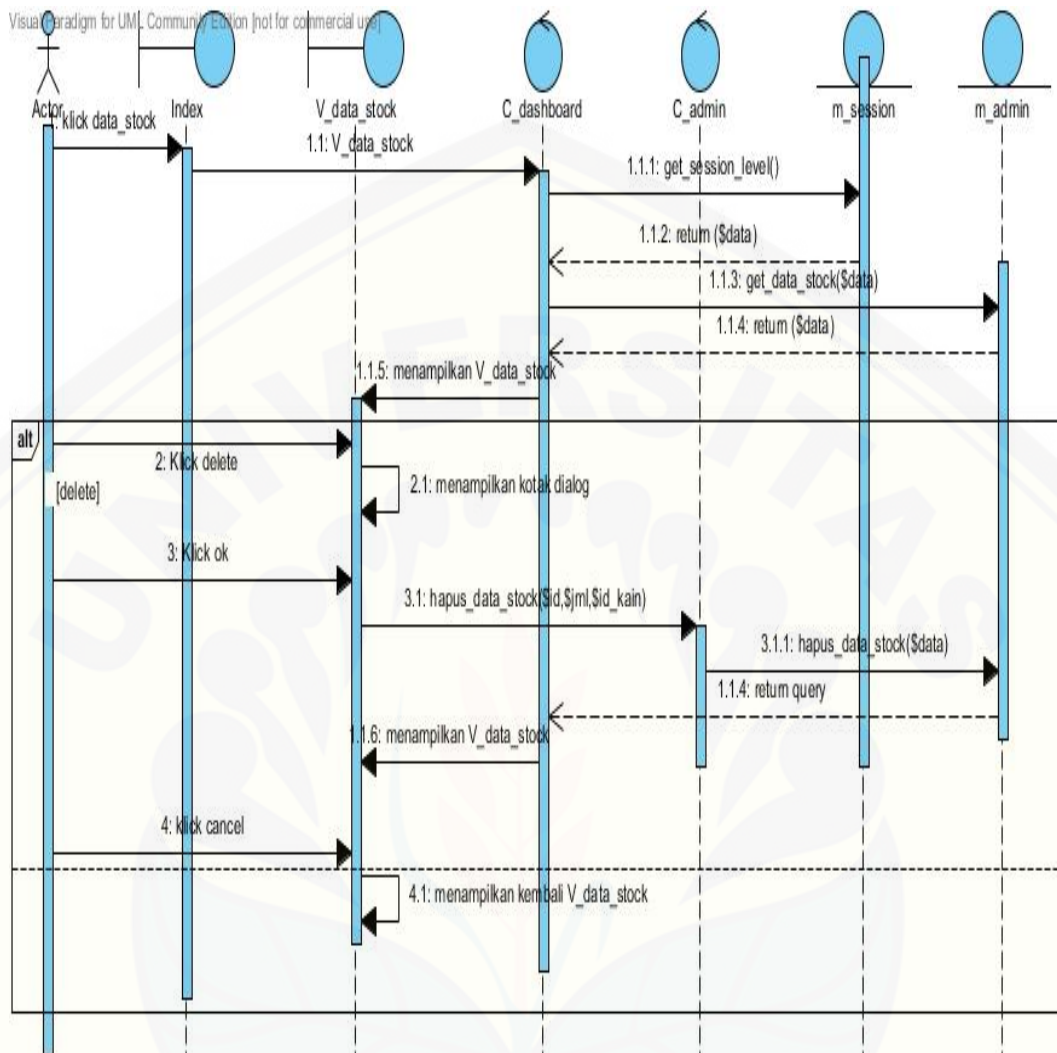
2.8 Sequence Diagram Delete Data Kain



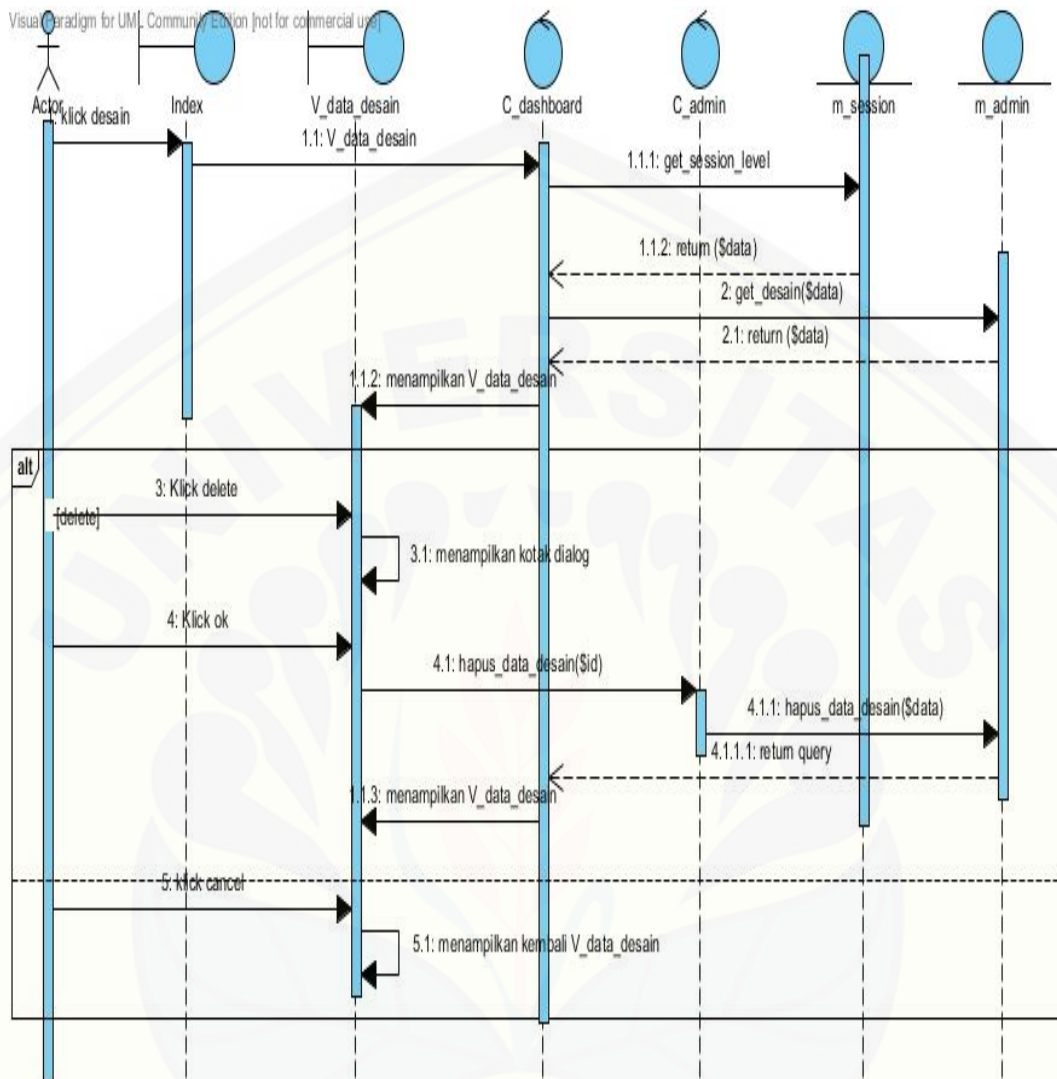
2.9 Sequence Diagram Delete Data Pemesanan



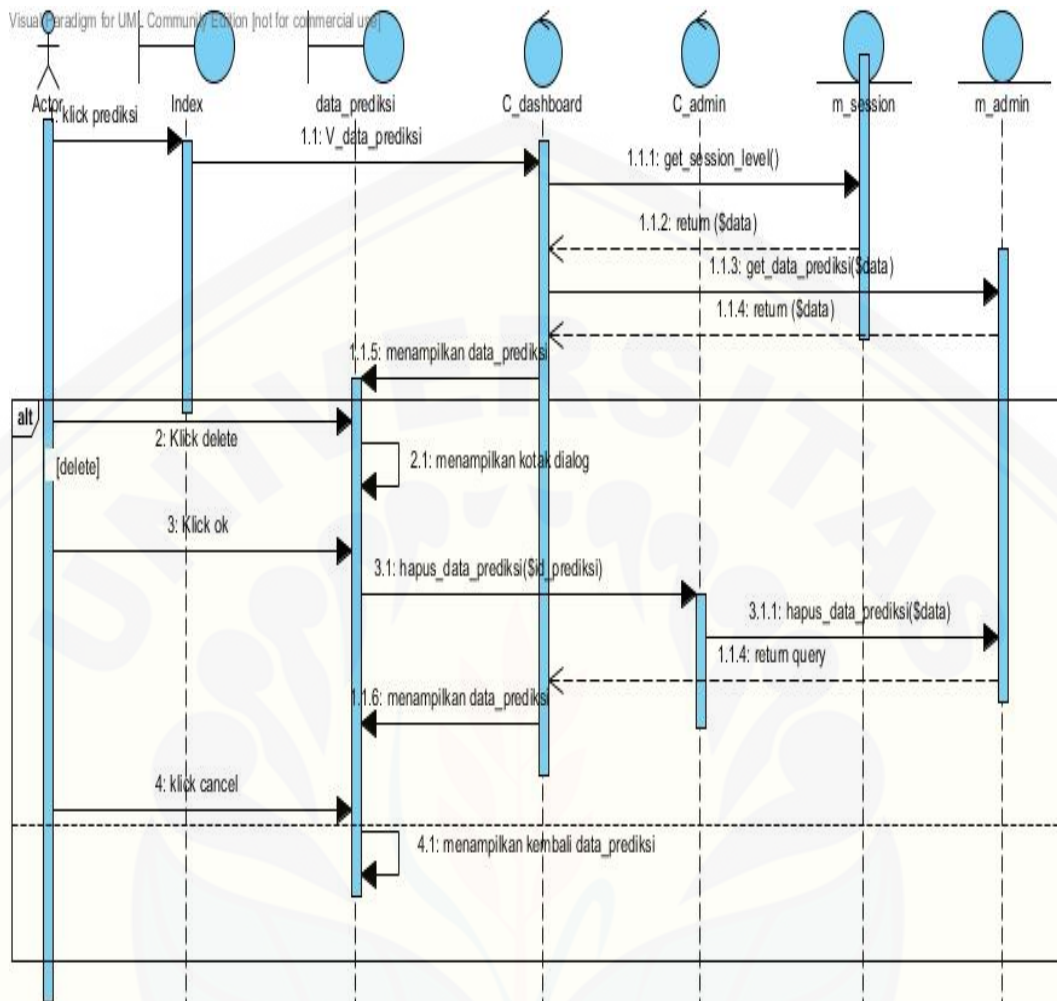
2.10 Sequence Diagram Delete Data Stock



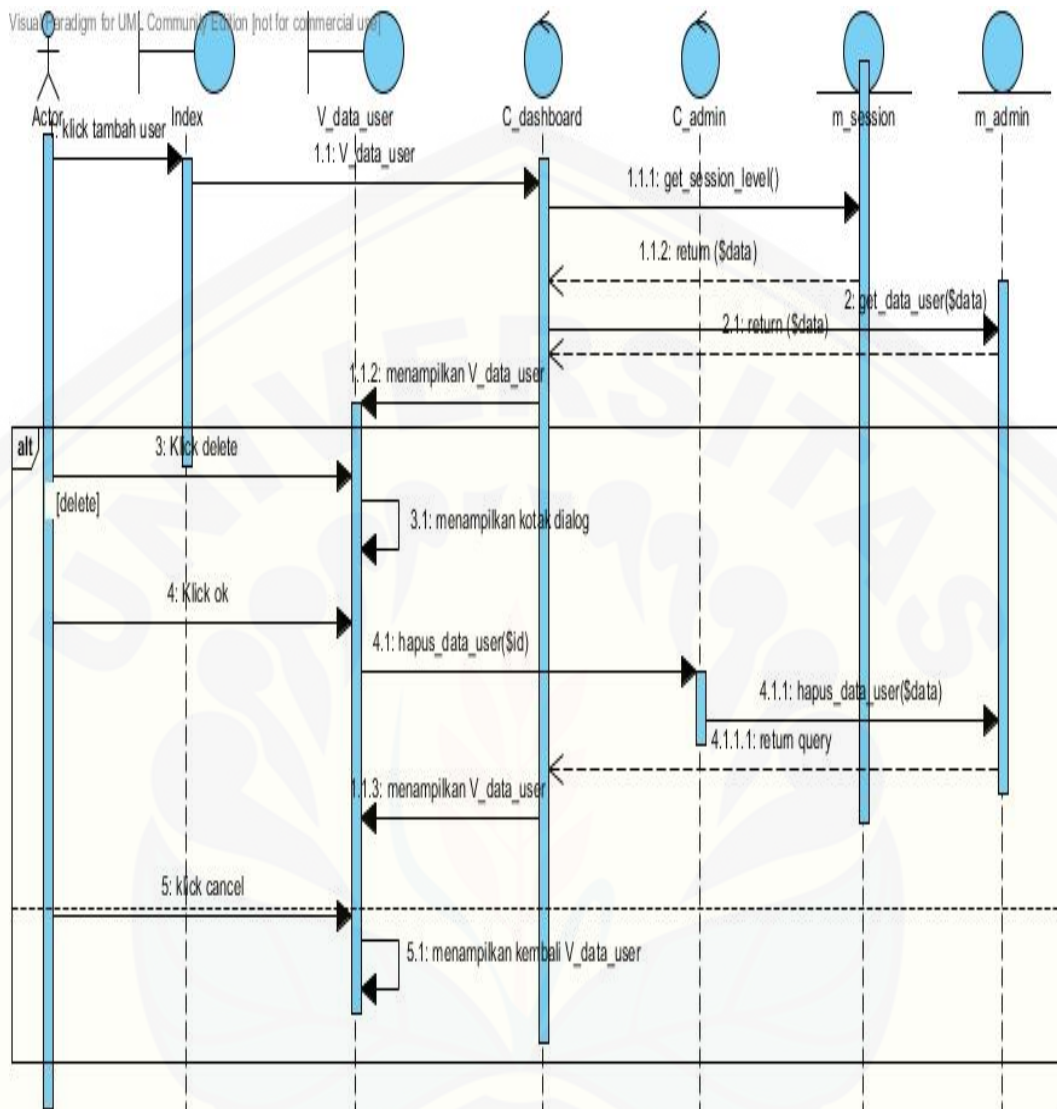
2.11 Sequence Diagram Delete Data Desain



2.12 Sequence Diagram Delete Data Prediksi

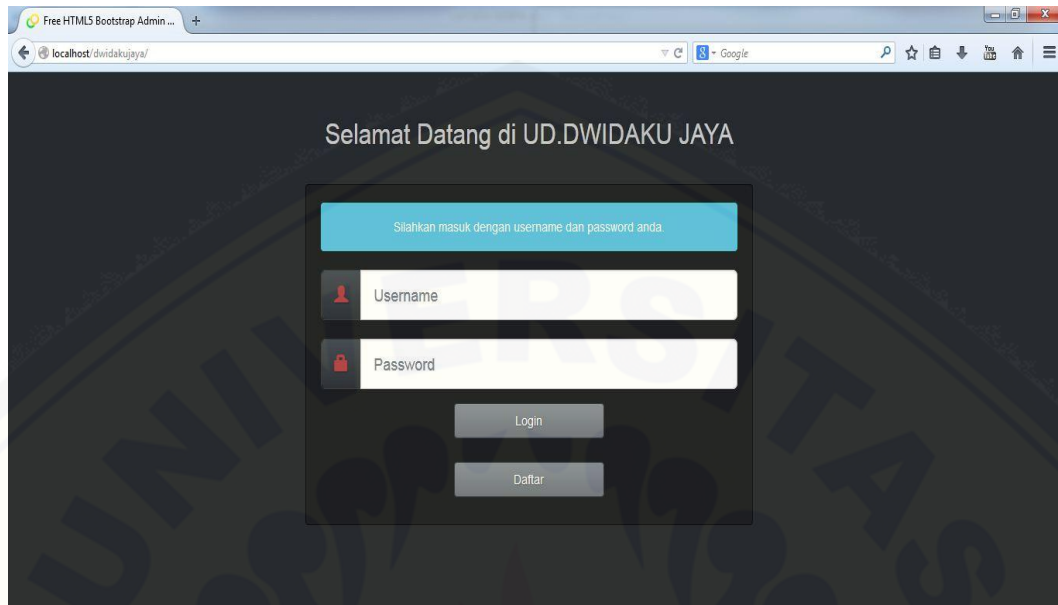


2.13 Sequence Diagram Delete Data User



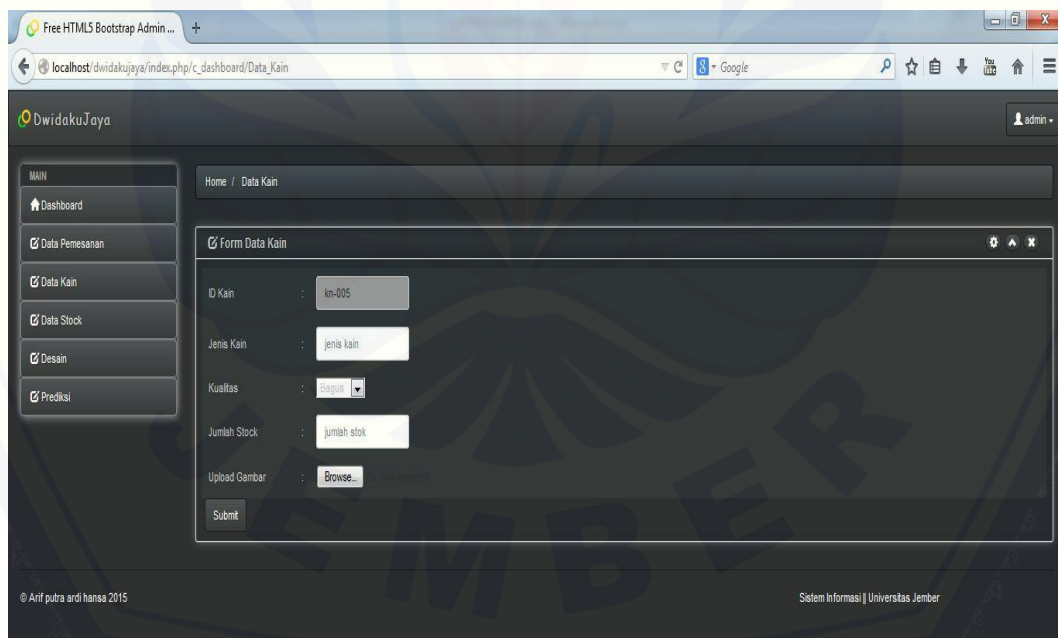
3. Lampiran Fitur

3.1 Fitur *Login*

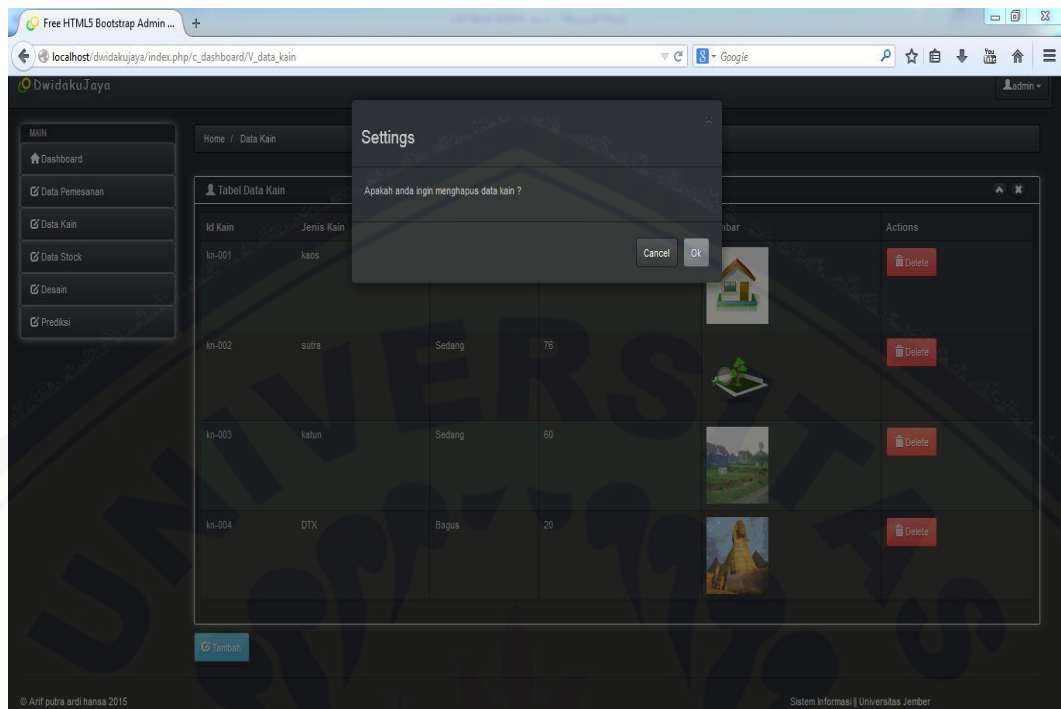


3.2 Fitur Manajemen Data Kain

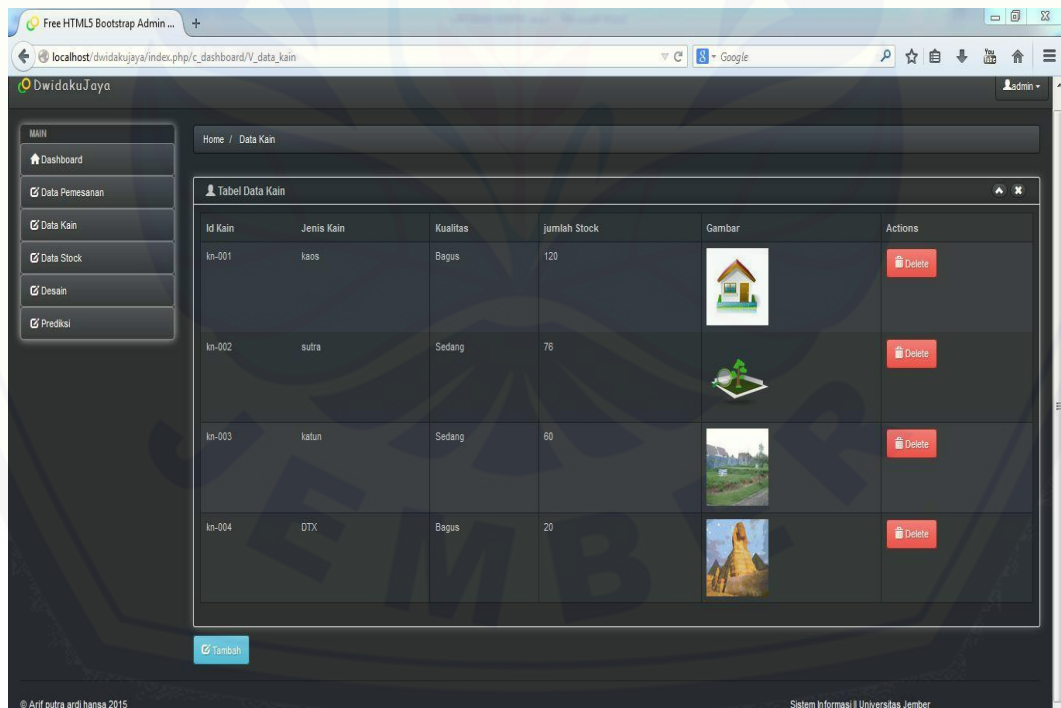
3.2.1 *Insert*



3.2.2 Delete



3.2.3 view



3.3 Fitur Manajemen Data *User*

3.3.1 *Insert*

The screenshot shows a web browser window displaying the 'Tambah User' form in the DwidakuJaya dashboard. The form is titled 'Form Tambah user' and contains the following fields:

- ID User:
- Nama Lengkap:
- Alamat:
- No telp:
- User Name:
- Password:
- Level:

A 'Submit' button is located at the bottom left of the form. The dashboard sidebar on the left includes a 'MAIN' menu with options: Dashboard, Data Pemesanan, Data Kain, Data Stock, Prediksi, Data User, and Desain. The breadcrumb trail at the top of the form area reads 'Home / Tambah User'.

3.3.2 *Edit*

The screenshot shows the 'Edit User' dialog box in the DwidakuJaya dashboard. The dialog box is titled 'Apakah anda ingin mengedit data user' and contains the following fields:

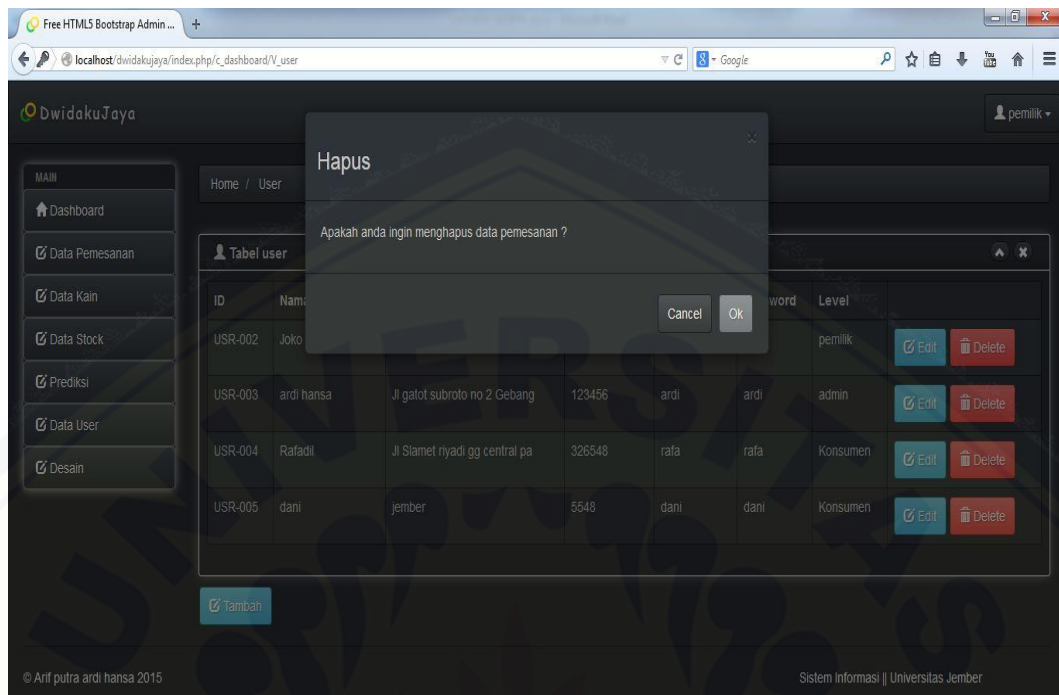
- Id_user:
- Username:
- Password:
- Level:

The dialog box has 'Cancel' and 'Simpan' buttons. In the background, a table of users is visible:

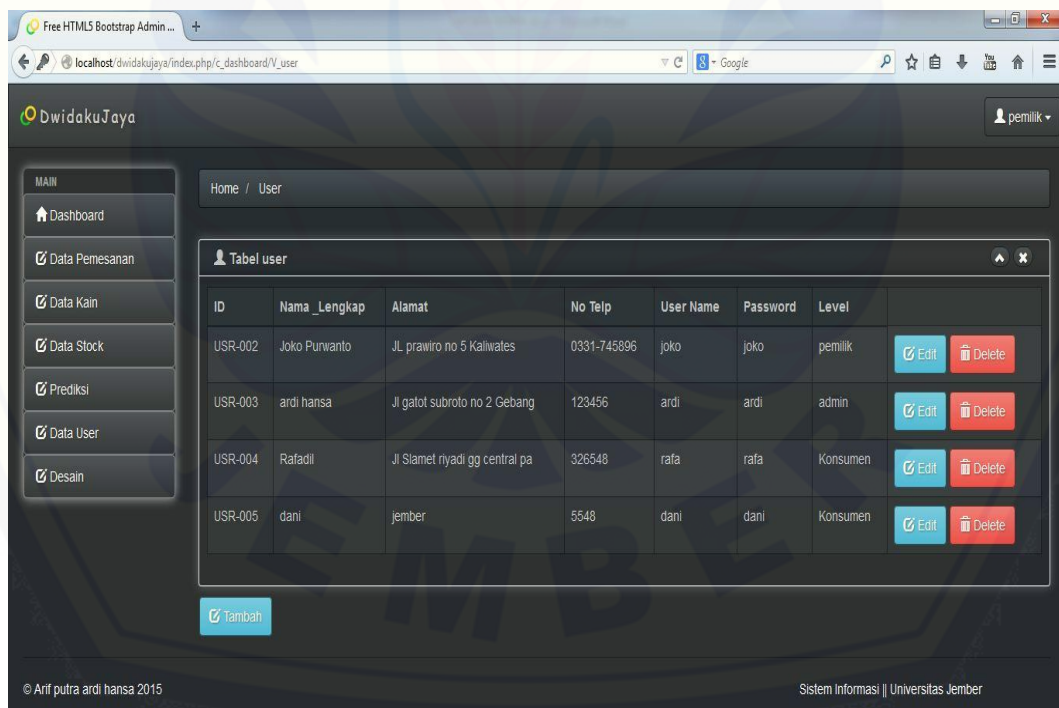
ID	Nama	password	Level	
USR-002	Joko		pemilik	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
USR-003	ardi		admin	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
USR-004	Rafa		Konsumen	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
USR-005	Idani		Konsumen	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

The dashboard sidebar and breadcrumb trail ('Home / User') are also visible in the background.

3.3.3 Delete

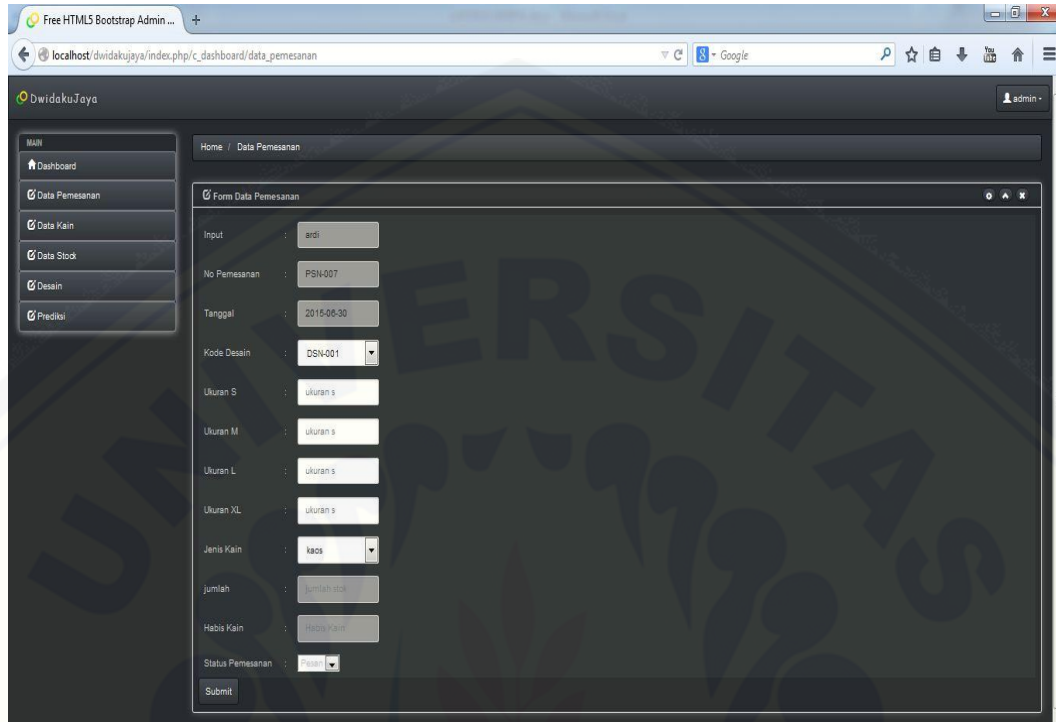


3.3.4 view

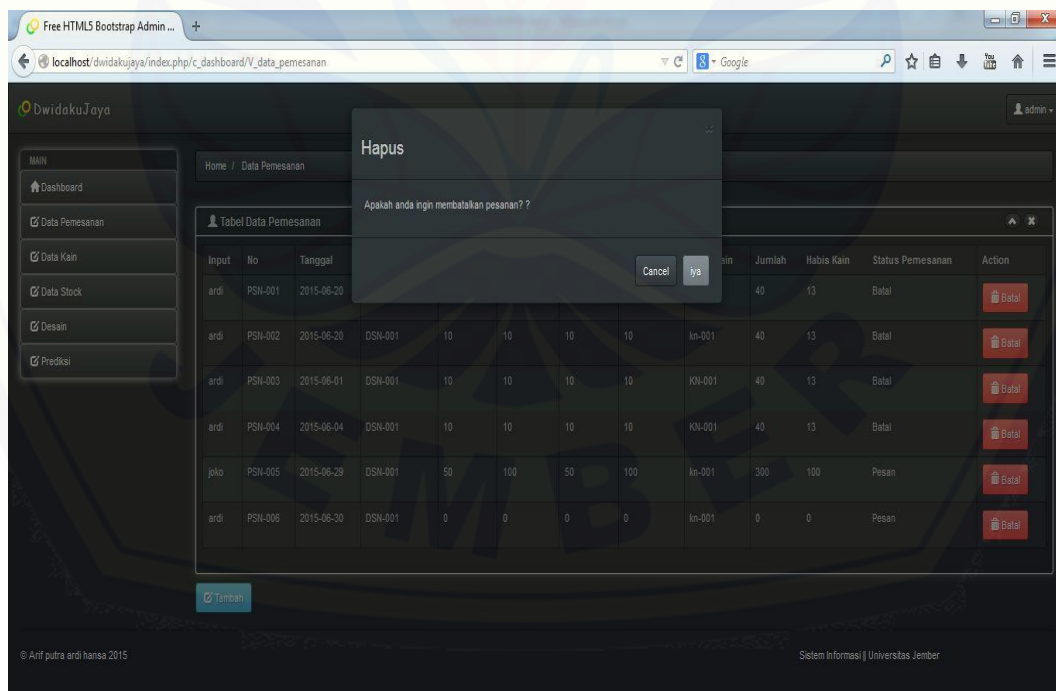


3.4 Fitur Manajemen Data Pemesanan

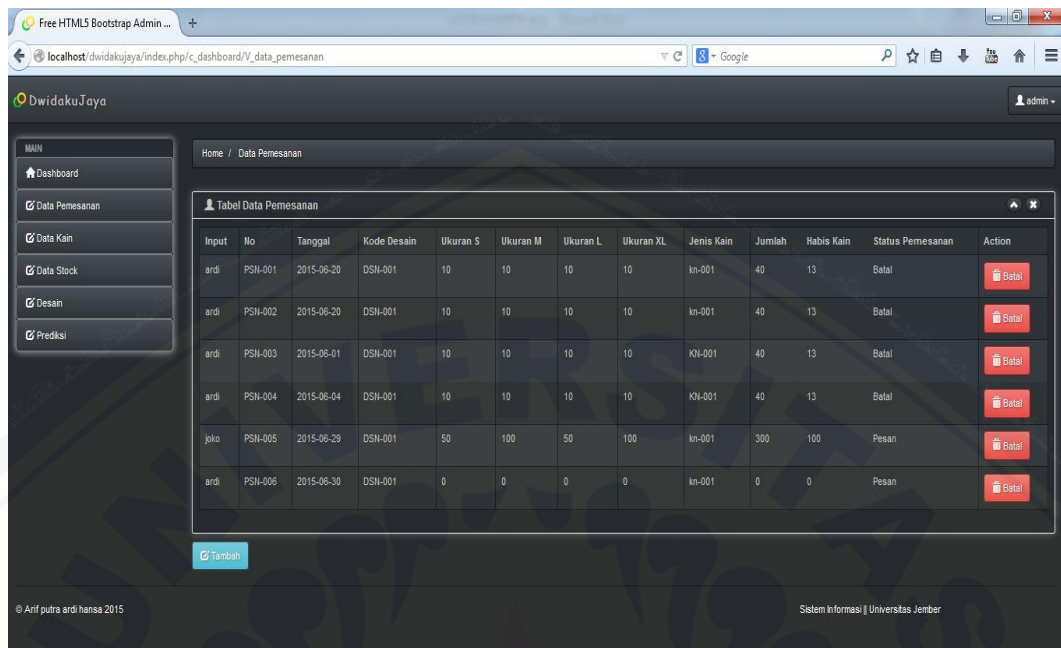
3.4.1 Insert



3.4.2 Delete

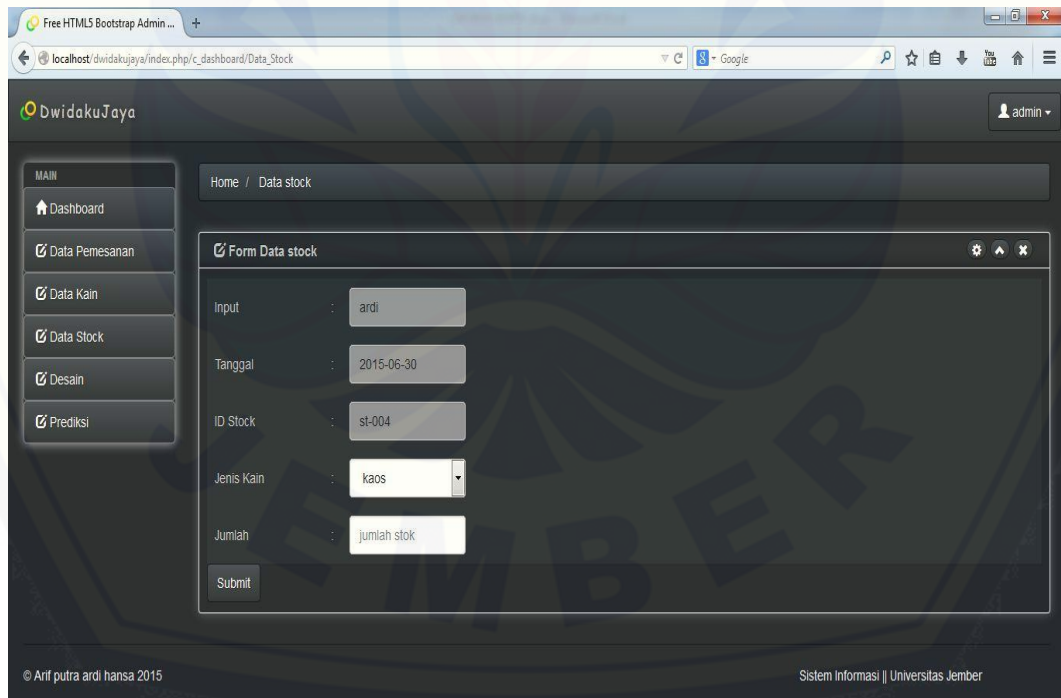


3.4.3 view

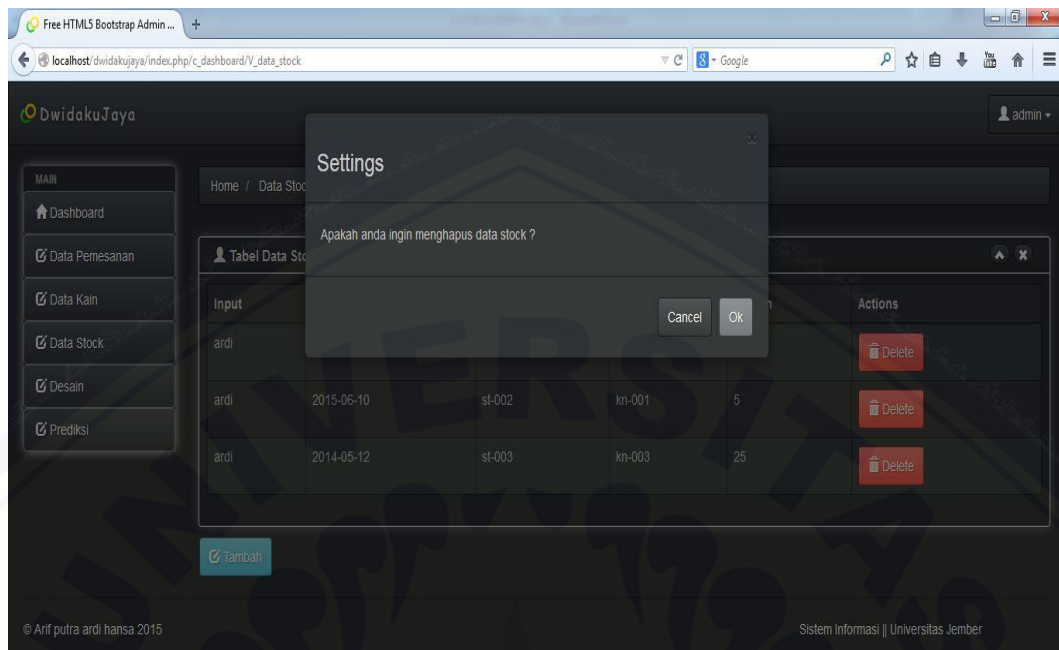


3.5 Fitur Manajemen Data Stock

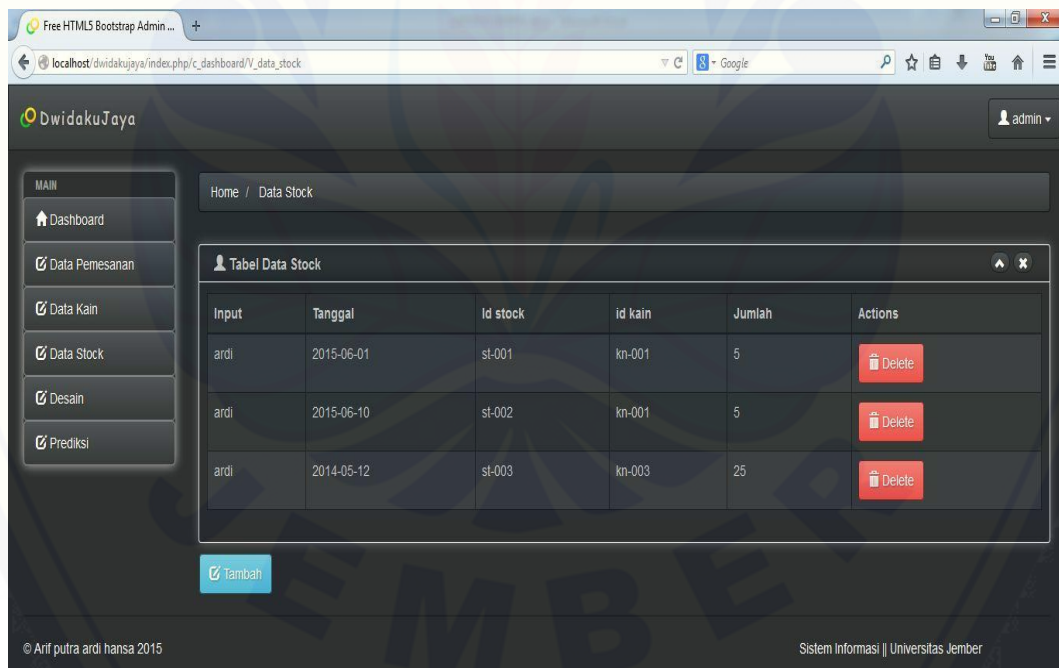
3.5.1 Insert



3.5.2 Delete

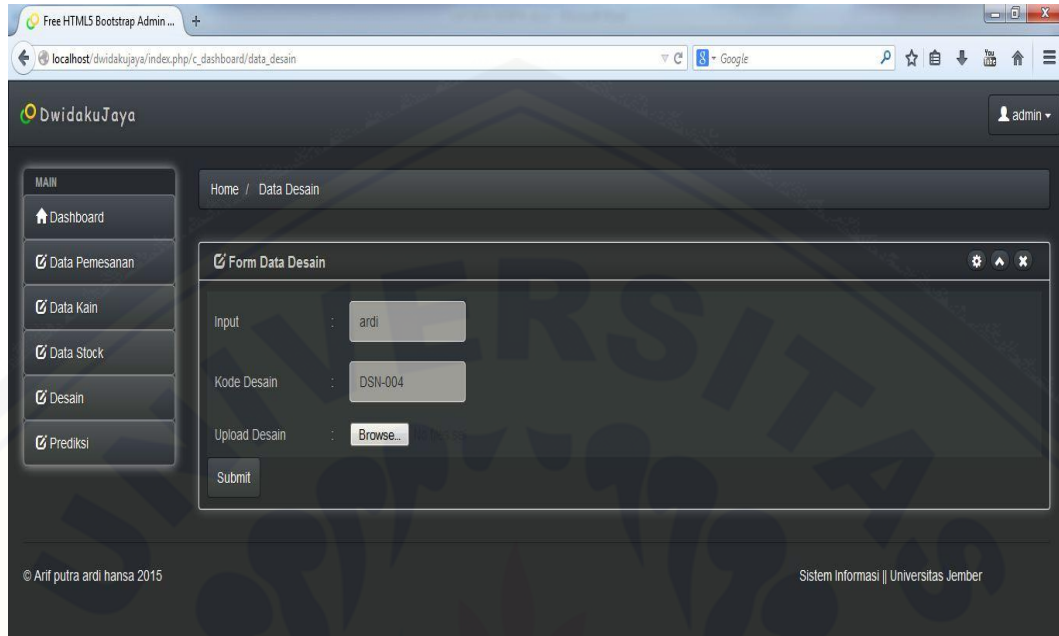


3.5.3 view

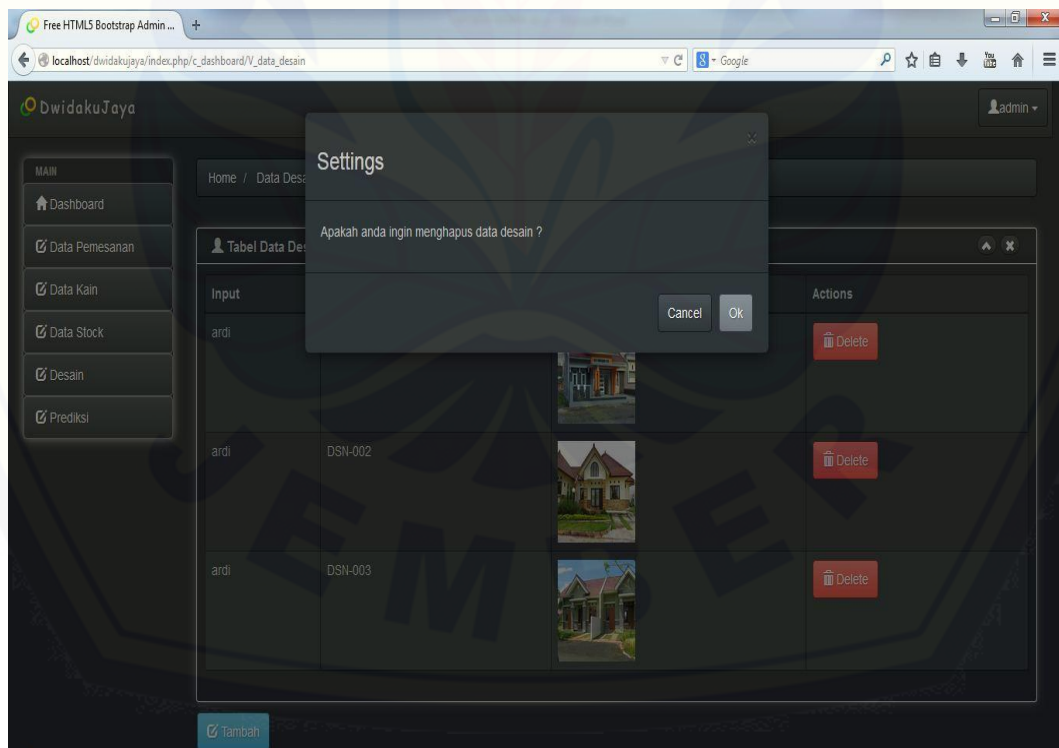


3.6 Fitur Manajemen Data Desain

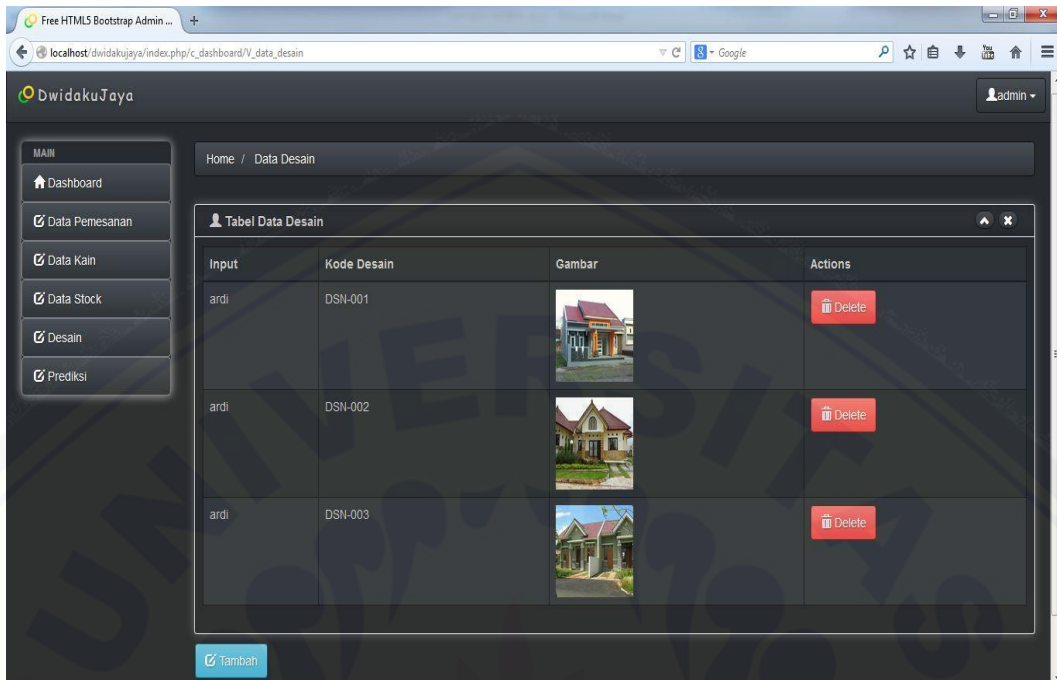
3.6.1 *Insert*






3.6.2 *Delete*



3.6.3 view



The screenshot displays a web application interface for 'DwidakuJaya'. The main content area shows a table titled 'Tabel Data Desain' with the following data:

Input	Kode Desain	Gambar	Actions
ardi	DSN-001		Delete
ardi	DSN-002		Delete
ardi	DSN-003		Delete

Below the table, there is a blue button labeled 'Tambah'.