

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan pembahasan pada bab-bab sebelumnya, maka dapat ditarik beberapa kesimpulan yaitu :

1. Perangkat lunak ASCaMaDa dibuat berbasis web menggunakan konsep *Supply Chain Management* sehingga menghubungkan distributor, perusahaan dan pemasok dalam suatu rantai produksi secara on-line.
2. Perangkat lunak ASCaMaDa dapat mengelola aliran data dari proses pencarian bahan baku, produksi sampai ke distribusi dengan efektif dan efisien.

5.2. Saran

Beberapa saran yang dapat diambil dari proses analisis sampai pada pembuatan tugas akhir ini adalah sebagai berikut :

1. Sistem diharapkan bisa dikembangkan lebih lanjut dengan fungsionalitas yang semakin baik seperti misalnya analisis permintaan pasar sehingga proses pembuatan schedule produksi semakin efektif.
2. Perangkat lunak ASCaMaDa dapat diintegrasikan dengan sistem lain diluar Supply Chain seperti sistem Intelejensi Bisnis sehingga sistem menjadi semakin berdayaguna.

DAFTAR PUSTAKA

Adnan, Risman, et.al. 2005. *Migrasi VB6 ke VB.NET*.
<http://otak.csharpindonesia.net>

Deitel. 2002. *C# How To Program*. Prentice Hall, Inc.
Englewood Cliffs, New Jersey.

Fowler, Martin. 2005. *UML Distilled - 3th Ed., Panduan Singkat Bahasa Pemrograman Objek Standar*. ANDI: Yogyakarta.

Hariman, Gunadi, A. Suhendar. *Visual Modeling Menggunakan UML dan RATIONAL ROSE*. Informatika Bandung: Bandung.

Kurniawan, Agus, et.al. 2004. *Pengenalan Bahasa C#*.
<http://otak.csharpindonesia.net>

Larman, Craig. 2005. *Applying UML and patterns : an introduction to object-oriented analysis and design and the Unified Process*. Prentice-Hall, Inc. Englewood Cliffs, New Jersey.

Nugroho, Adi. 2005. *Rational Rose untuk Pemrograman Berorientasi Objek*. Informatika Bandung.

Wisnubhadra, Irya. 2005. *Bahan Kuliah Basis Data dan Basis Data Lanjut*. Universitas Atma Jaya Yogyakarta.
<http://kuliah.inf.uajy.ac.id>

<http://www.lmfeui.com>

<http://www.hendriyusufbrothers.com/indonesia/articles>

SKPL

SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

Pembangunan Aplikasi Supply Chain Management

(Studi Kasus: PT. Sari Husada)

(ASCaMaDa)


Dipersiapkan oleh:

K. Joko Maryono

03 07 03937

Program Studi Teknik Informatika – Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		SKPL-ASCaMaDa		1/119
		Revisi		Tgl : 2-04-2007

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	
G	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh								
Diperiksa oleh								
Disetujui oleh								

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Daftar Isi

1	Pendahuluan.....	11
1.1	Tujuan.....	11
1.2	Lingkup Masalah.....	11
1.3	Istilah dan Singkatan.....	14
1.4	Referensi.....	14
1.5	Deskripsi umum (Overview).....	15
2	Deskripsi Keseluruhan.....	16
2.1	Perspektif Produk.....	16
2.2	Fungsi Produk.....	16
2.3	Karakteristik Pengguna.....	21
2.4	Batasan-batasan.....	21
2.5	Asumsi dan Ketergantungan.....	21
3	Deskripsi Rinci Kebutuhan.....	22
3.1	Spesifikasi Kebutuhan Fungsionalitas.....	22
3.1.1	<i>Use Case Diagram Administrator</i>	22
3.1.1.1	Spesifikasi Use Case : Login.....	22
3.1.1.2	Spesifikasi Use Case : Add User.....	23
3.1.1.3	Spesifikasi Use Case : Edit Password.....	24
3.1.1.4	Spesifikasi Use Case : Delete User.....	25
3.1.1.5	Spesifikasi Use Case : Edit User.....	26
3.1.1.6	Spesifikasi Use Case : Display User.....	27
3.1.2	<i>Use Case Diagram Production Planning and Inventory Control</i>	28
3.1.2.1	Spesifikasi Use Case : Entri Schedule.....	28
3.1.2.2	Spesifikasi Use Case : Entri Reschedule.....	29
3.1.2.3	Spesifikasi Use Case : Display Pesanan.....	30
3.1.2.4	Spesifikasi Use Case : Display Schedule Produksi.....	31
3.1.2.5	Spesifikasi Use Case : Display Reschedule Produksi.....	31
3.1.2.6	Spesifikasi Use Case : Display CRP.....	32
3.1.2.7	Spesifikasi Use Case : Display MRP.....	33
3.1.2.8	Spesifikasi Use Case : Entri Purchase Order.....	33
3.1.2.9	Spesifikasi Use Case : Display PO.....	34
3.1.2.10	Spesifikasi Use Case : Surat Perintah Produksi.....	34
3.1.2.11	Spesifikasi Use Case : Print SPP.....	35
3.1.3	<i>Use Case Diagram Production</i>	37
3.1.3.1	Spesifikasi Use Case : Add Produk.....	38
3.1.3.2	Spesifikasi Use Case : Edit Produk.....	38
3.1.3.3	Spesifikasi Use Case : Delete Produk.....	39
3.1.3.4	Spesifikasi Use Case : Display Produk.....	40
3.1.3.5	Spesifikasi Use Case : Entri Hasil Produksi.....	40
3.1.3.6	Spesifikasi Use Case : Edit Hasil Produksi.....	41
3.1.3.7	Spesifikasi Use Case : Delete Hasil Produksi.....	42
3.1.3.8	Spesifikasi Use Case : Display Hasil Produksi.....	43
3.1.3.9	Spesifikasi Use Case : Laporan Hasil Produksi.....	43
3.1.3.10	Spesifikasi Use Case : Print Laporan Produksi.....	44
3.1.3.11	Spesifikasi Use Case : Add Mesin.....	45
3.1.3.12	Spesifikasi Use Case : Edit Mesin.....	45
3.1.3.13	Spesifikasi Use Case : Delete Mesin.....	46
3.1.3.14	Spesifikasi Use Case : Edit Status Mesin.....	47
3.1.3.15	Spesifikasi Use Case : Display SPP.....	48
3.1.4	<i>Use Case Diagram Warehouse</i>	49
3.1.4.1	Spesifikasi Use Case : Add Bahan Produksi.....	50
3.1.4.2	Spesifikasi Use Case : Edit Bahan Produksi.....	50
3.1.4.3	Spesifikasi Use Case : Delete Bahan Produksi.....	51
3.1.4.4	Spesifikasi Use Case : Display Bahan Produksi.....	52
3.1.4.5	Spesifikasi Use Case : Entri Stock Keluar.....	53
3.1.4.6	Spesifikasi Use Case : Entri Stock Masuk.....	53

3.1.4.7	Spesifikasi Use Case : Add Kategori.....	54
3.1.4.8	Spesifikasi Use Case : Edit Kategori.....	55
3.1.4.9	Spesifikasi Use Case : Delete Kategori.....	56
3.1.4.10	Spesifikasi Use Case : Laporan Bahan Baku Produksi.....	57
3.1.4.11	Spesifikasi Use Case : Print Laporan Bahan Baku Produksi.....	57
3.1.4.12	Spesifikasi Use Case : Laporan Bahan Baku Produksi Keluar.....	58
3.1.4.13	Spesifikasi Use Case : Print Laporan Bahan Baku Produksi Keluar.....	59
3.1.5	<i>Use Case Diagram Quality Assurance</i>	60
3.1.5.1	Spesifikasi Use Case : Edit Status Bahan Baku.....	60
3.1.5.2	Spesifikasi Use Case : Edit Status Barang Produksi.....	60
3.1.6	<i>Use Case Diagram Marketing</i>	62
3.1.6.1	Spesifikasi Use Case : Add Penjualan.....	62
3.1.6.2	Spesifikasi Use Case : Edit Penjualan.....	63
3.1.6.3	Spesifikasi Use Case : Edit Harga Produk.....	64
3.1.6.4	Spesifikasi Use Case : Delete Penjualan.....	65
3.1.6.5	Spesifikasi Use Case : Display Penjualan.....	65
3.1.6.6	Spesifikasi Use Case : Print Penjualan.....	66
3.1.7	<i>Use Case Diagram Purchasing</i>	67
3.1.7.1	Spesifikasi Use Case : Add Supplier.....	68
3.1.7.2	Spesifikasi Use Case : Edit Supplier.....	69
3.1.7.3	Spesifikasi Use Case : Delete Supplier.....	69
3.1.7.4	Spesifikasi Use Case : Display Data Supplier.....	71
3.1.7.5	Spesifikasi Use Case : Edit Harga Bahan Baku.....	71
3.1.7.6	Spesifikasi Use Case : Order Bahan Baku.....	72
3.1.7.7	Spesifikasi Use Case : Display Order Pembelian.....	73
3.1.7.8	Spesifikasi Use Case : Laporan Pembelian.....	73
3.1.7.9	Spesifikasi Use Case : Print Laporan Pembelian.....	74
3.1.8	<i>Use Case Diagram Supplier</i>	75
3.1.8.1	Spesifikasi Use Case : Display Purchase Order.....	75
3.1.8.2	Spesifikasi Use Case : Print Purchase Order.....	76
3.1.9	<i>Use Case Diagram Distributor</i>	76
3.1.9.1	Spesifikasi Use Case : Entri Pesanan.....	77
3.1.9.2	Spesifikasi Use Case : Display Produk.....	78
3.2	Realisasi Use Case.....	79
3.2.1	<i>Static Structure Diagram</i>	79
3.2.1.1	Analysis Class Diagram : Package Dependencies.....	79
3.2.1.2	Analysis Class Diagram : Package Administrator.....	79
3.2.1.3	Analysis Class Diagram : Package PPIC.....	80
3.2.1.4	Analysis Class Diagram : Package Produksi.....	81
3.2.1.5	Analysis Class Diagram : Package Warehouse.....	82
3.2.1.6	Analysis Class Diagram : Package Quality Assurance.....	83
3.2.1.7	Analysis Class Diagram : Package Purchasing.....	84
3.2.1.8	Analysis Class Diagram : Package Marketing.....	85
3.2.1.9	Analysis Class Diagram : Package Distributor.....	85
3.2.1.10	Analysis Class Diagram : Package Supplier.....	86
3.2.2	<i>Interaction Diagram</i>	86
3.2.2.1	Analysis Collaboration Diagram : Use Case Login.....	86
3.2.2.2	Analysis Collaboration Diagram : Use Case Add User.....	87
3.2.2.3	Analysis Collaboration Diagram : Use Case Change Password.....	87
3.2.2.4	Analysis Collaboration Diagram : Use Case Delete User.....	88
3.2.2.5	Analysis Collaboration Diagram : Use Case Edit User.....	88
3.2.2.6	Analysis Collaboration Diagram : Use Case Display User.....	89
3.2.2.7	Analysis Collaboration Diagram : Use Case Entri Schedule Produksi.....	89
3.2.2.8	Analysis Collaboration Diagram : Use Case Entri Reschedule Produksi.....	90
3.2.2.9	Analysis Collaboration Diagram : Use Case Display Barang Pesanan.....	90
3.2.2.10	Analysis Collaboration Diagram : Use Case Display Schedule.....	91
3.2.2.11	Analysis Collaboration Diagram : Use Case Display Reschedule.....	91
3.2.2.12	Analysis Collaboration Diagram : Use Case Display CRP.....	92
3.2.2.13	Analysis Collaboration Diagram : Use Case Display MRP.....	92
3.2.2.14	Analysis Collaboration Diagram : Use Case Entri Purchase Order.....	93

3.2.2.15	Analysis Collaboration Diagram : Use Case Display Purchase Order.....	93
3.2.2.16	Analysis Collaboration Diagram : Use Case Surat Perintah Produksi.....	94
3.2.2.17	Analysis Collaboration Diagram : Use Case Print SPP.....	94
3.2.2.18	Analysis Collaboration Diagram : Use Case Add Produk.....	95
3.2.2.19	Analysis Collaboration Diagram : Use Case Edit Produk.....	95
3.2.2.20	Analysis Collaboration Diagram : Use Case Delete Produk.....	96
3.2.2.21	Analysis Collaboration Diagram : Use Case Display Produk.....	96
3.2.2.22	Analysis Collaboration Diagram : Use Case Entri Hasil Produksi.....	97
3.2.2.23	Analysis Collaboration Diagram : Use Case Edit Hasil Produksi.....	97
3.2.2.24	Analysis Collaboration Diagram : Use Case Delete Hasil Produksi.....	98
3.2.2.25	Analysis Collaboration Diagram : Use Case Display Hasil Produksi.....	98
3.2.2.26	Analysis Collaboration Diagram : Use Case Laporan Hasil Produksi.....	99
3.2.2.27	Analysis Collaboration Diagram : Use Case Print Laporan Produksi.....	100
3.2.2.28	Analysis Collaboration Diagram : Use Case Add Mesin.....	100
3.2.2.29	Analysis Collaboration Diagram : Use Case Edit Mesin.....	100
3.2.2.30	Analysis Collaboration Diagram : Use Case Delete Mesin.....	101
3.2.2.31	Analysis Collaboration Diagram : Use Case Edit Status Mesin.....	101
3.2.2.32	Analysis Collaboration Diagram : Use Case Display SPP.....	101
3.2.2.33	Analysis Collaboration Diagram : Use Case Add Bahan Produksi.....	102
3.2.2.34	Analysis Collaboration Diagram : Use Case Edit Bahan Produksi.....	102
3.2.2.35	Analysis Collaboration Diagram : Use Case Delete Bahan Produksi.....	103
3.2.2.36	Analysis Collaboration Diagram : Use Case Display Bahan Produksi.....	103
3.2.2.37	Analysis Collaboration Diagram : Use Case Entri Stok Keluar.....	104
3.2.2.38	Analysis Collaboration Diagram : Use Case Entri Stok Masuk.....	104
3.2.2.39	Analysis Collaboration Diagram : Use Case Add Kategori.....	105
3.2.2.40	Analysis Collaboration Diagram : Use Case Edit Kategori.....	105
3.2.2.41	Analysis Collaboration Diagram : Use Case Delete Kategori.....	105
3.2.2.42	Analysis Collaboration Diagram : Use Case Laporan BB Produksi.....	105
3.2.2.43	Analysis Collaboration Diagram : Use Case Print Laporan BB Produksi.....	106
3.2.2.44	Analysis Collaboration Diagram : Use Case Lap BB Produksi Keluar.....	106
3.2.2.45	Analysis Collaboration Diagram : Use Case Laporan Bahan Baku Produksi.....	106
3.2.2.46	Analysis Collaboration Diagram : Use Case Edit Status Bahan Baku Produksi.....	107
3.2.2.47	Analysis Collaboration Diagram : Use Case Edit Status Hasil Produksi.....	107
3.2.2.48	Analysis Collaboration Diagram : Use Case Add Penjualan.....	108
3.2.2.49	Analysis Collaboration Diagram : Use Case Edit Penjualan.....	108
3.2.2.50	Analysis Collaboration Diagram : Use Case Edit Harga Produk.....	108
3.2.2.51	Analysis Collaboration Diagram : Use Case Delete Penjualan.....	109
3.2.2.52	Analysis Collaboration Diagram : Use Case Display Penjualan.....	109
3.2.2.53	Analysis Collaboration Diagram : Use Case Print Penjualan.....	109
3.2.2.54	Analysis Collaboration Diagram : Use Case Add Supplier.....	110
3.2.2.55	Analysis Collaboration Diagram : Use Case Edit Supplier.....	110
3.2.2.56	Analysis Collaboration Diagram : Use Case Delete Supplier.....	111
3.2.2.57	Analysis Collaboration Diagram : Use Case Display Data Supplier.....	112
3.2.2.58	Analysis Collaboration Diagram : Use Case Edit Harga Bahan Baku.....	112
3.2.2.59	Analysis Collaboration Diagram : Use Case Order Bahan Produksi.....	113
3.2.2.60	Analysis Collaboration Diagram : Use Case Display Order Pembelian.....	114
3.2.2.61	Analysis Collaboration Diagram : Use Case Laporan Pembelian.....	114
3.2.2.62	Analysis Collaboration Diagram : Use Case Print Laporan Pembelian.....	115
3.2.2.63	Analysis Collaboration Diagram : Use Case Display Purchase Order.....	115
3.2.2.64	Analysis Collaboration Diagram : Use Case Print Purchase Order.....	116
3.2.2.65	Analysis Collaboration Diagram : Use Case Entri Pesanan.....	116
3.2.2.66	Analysis Collaboration Diagram : Use Case Display Produk.....	116
3.3	Spesifikasi Kebutuhan Non-Fungsionalitas.....	117
3.3.1	Antarmuka pemakai.....	117
3.3.2	Antarmuka perangkat keras.....	117
3.3.3	Antarmuka perangkat lunak.....	117
3.3.4	Antarmuka Komunikasi.....	118
4	Spesifikasi Kebutuhan Data.....	119
4.1	Entity Relationship Diagram (ERD).....	119

Daftar Gambar

Gambar 1.1 <i>Activity Diagram</i> Produksi.....	12
Gambar 1.2 <i>Activity Diagram</i> Pembelian.....	13
Gambar 1.3 <i>Activity Diagram</i> Penjualan.....	13
Gambar 1.4 <i>Activity Diagram</i> Supply Reschedule Produksi.....	13
Gambar 3.1 <i>Use Case Diagram</i> Administrator.....	22
Gambar 3.2 <i>Use Case Diagram</i> PPIC.....	28
Gambar 3.3 <i>Use Case Diagram</i> Production.....	37
Gambar 3.4 <i>Use Case Diagram</i> Warehouse.....	49
Gambar 3.5 <i>Use Case Diagram</i> Quality Assurance.....	60
Gambar 3.6 <i>Use Case Diagram</i> Marketing.....	62
Gambar 3.7 <i>Use Case Diagram</i> Purchasing.....	68
Gambar 3.8 <i>Use Case Diagram</i> Supplier.....	75
Gambar 3.9 <i>Use Case Diagram</i> Distributor.....	77
Gambar 3.10 Analysis Class Diagram : Package Dependencies.....	79
Gambar 3.11 Analysis Class Diagram : Package Data Administrator.....	79
Gambar 3.12 Analysis Class Diagram : Package PPIC.....	80
Gambar 3.13 Analysis Class Diagram : Package Produksi.....	81
Gambar 3.14 Analysis Class Diagram : Package Warehouse.....	82
Gambar 3.15 Analysis Class Diagram : Package Quality Assurance.....	83
Gambar 3.16 Analysis Class Diagram : Package Purchasing.....	84
Gambar 3.17 Analysis Class Diagram : Package Marketing.....	85
Gambar 3.18 Analysis Class Diagram : Package Distributor.....	85
Gambar 3.19 Analysis Class Diagram : Package Supplier.....	86
Gambar 3.20 Analysis Collaboration Diagram : Use Case Login.....	86
Gambar 3.21 Analysis Collaboration Diagram : Use Case Add User.....	87
Gambar 3.22 Analysis Collaboration Diagram : Use Case Change Password.....	87
Gambar 3.23 Analysis Collaboration Diagram : Use Case Delete User.....	88
Gambar 3.24 Analysis Collaboration Diagram : Use Case Edit User.....	88
Gambar 3.25 Analysis Collaboration Diagram : Use Case Display User.....	89
Gambar 3.26 Analysis Collaboration Diagram : Use Case Create Entri Schedule Produksi.....	89
Gambar 3.27 Analysis Collaboration Diagram : Use Case Entri Reschedule Produksi.....	90
Gambar 3.28 Analysis Collaboration Diagram : Use Case Display Barang Pesanan.....	90
Gambar 3.29 Analysis Collaboration Diagram : Use Case Display Schedule.....	91
Gambar 3.30 Analysis Collaboration Diagram : Use Case Display Reschedule.....	91
Gambar 3.31 Analysis Collaboration Diagram : Use Case Display CRP.....	92
Gambar 3.32 Analysis Collaboration Diagram : Use Case Display MRP.....	92
Gambar 3.33 Analysis Collaboration Diagram : Use Case Entri Purchase Order.....	93
Gambar 3.34 Analysis Collaboration Diagram : Use Case Display Purchase Order.....	93
Gambar 3.35 Analysis Collaboration Diagram : Use Case Surat Perintah Produksi.....	94
Gambar 3.36 Analysis Collaboration Diagram : Use Case Print SPP.....	94
Gambar 3.37 Analysis Collaboration Diagram : Use Case Add Produk.....	95
Gambar 3.38 Analysis Collaboration Diagram : Use Case Edit Produk.....	95
Gambar 3.39 Analysis Collaboration Diagram : Use Case Delete Produk.....	96
Gambar 3.40 Analysis Collaboration Diagram : Use Case Display Produk.....	96
Gambar 3.41 Analysis Collaboration Diagram : Use Case Entri Hasil Produksi.....	97
Gambar 3.42 Analysis Collaboration Diagram : Use Case Edit Hasil Produksi.....	97
Gambar 3.43 Analysis Collaboration Diagram : Use Case Delete Hasil Produksi.....	98
Gambar 3.44 Analysis Collaboration Diagram : Use Case Display Hasil Produksi.....	98
Gambar 3.45 Analysis Collaboration Diagram : Use Case Laporan Hasil Produksi.....	99
Gambar 3.46 Analysis Collaboration Diagram : Use Case Print Laporan Produksi.....	100
Gambar 3.47 Analysis Collaboration Diagram : Use Case Add Mesin.....	100
Gambar 3.48 Analysis Collaboration Diagram : Use Case Edit Mesin.....	100
Gambar 3.49 Analysis Collaboration Diagram : Use Case Delete Mesin.....	101
Gambar 3.50 Analysis Collaboration Diagram : Use Case Edit Status Mesin.....	101
Gambar 3.51 Analysis Collaboration Diagram : Use Case Display SPP.....	101
Gambar 3.52 Analysis Collaboration Diagram : Use Case Add Bahan Produksi.....	102

Gambar 3.53 Analysis Collaboration Diagram : Use Case Edit Bahan Produksi.....	102
Gambar 3.54 Analysis Collaboration Diagram : Use Case Delete Bahan Produksi	103
Gambar 3.55 Analysis Collaboration Diagram : Use Case Display Bahan Produksi.....	103
Gambar 3.56 Analysis Collaboration Diagram : Use Case Entri Stok Keluar	104
Gambar 3.57 Analysis Collaboration Diagram : Use Case Entri Stok Masuk.....	104
Gambar 3.58 Analysis Collaboration Diagram : Use Case Add Kategori	105
Gambar 3.59 Analysis Collaboration Diagram : Use Case Edit Kategori	105
Gambar 3.60 Analysis Collaboration Diagram : Use Case Delete Kategori	105
Gambar 3.61 Analysis Collaboration Diagram : Use Case Laporan BB Produksi	105
Gambar 3.62 Analysis Collaboration Diagram : Use Case Print Laporan BB Produksi	106
Gambar 3.63 Analysis Collaboration Diagram : Use Case Lap BB Produksi Keluar	106
Gambar 3.64 Analysis Collaboration Diagram : Use Case Laporan Bahan Baku	106
Gambar 3.65 Analysis Collaboration Diagram : Use Case Edit Status Bahan Baku	107
Gambar 3.66 Analysis Collaboration Diagram : Use Case Edit Status Hasil Produksi	107
Gambar 3.67 Analysis Collaboration Diagram : Use Case Add Penjualan.....	108
Gambar 3.68 Analysis Collaboration Diagram : Use Case Edit Penjualan.....	108
Gambar 3.69 Analysis Collaboration Diagram : Use Case Edit Harga Produk.....	108
Gambar 3.70 Analysis Collaboration Diagram : Use Case Delete Penjualan.....	109
Gambar 3.71 Analysis Collaboration Diagram : Use Case Display Penjualan	109
Gambar 3.72 Analysis Collaboration Diagram : Use Case Print Penjualan.....	109
Gambar 3.73 Analysis Collaboration Diagram : Use Case Add Supplier	110
Gambar 3.74 Analysis Collaboration Diagram : Use Case Edit Supplier.....	110
Gambar 3.75 Analysis Collaboration Diagram : Use Case Delete Supplier	111
Gambar 3.76 Analysis Collaboration Diagram : Use Case Display Data Supplier.....	112
Gambar 3.77 Analysis Collaboration Diagram : Use Case Edit Harga Bahan Baku.....	112
Gambar 3.78 Analysis Collaboration Diagram : Use Case Order Bahan Produksi.....	113
Gambar 3.79 Analysis Collaboration Diagram : Use Case Display Order Pembelian	114
Gambar 3.80 Analysis Collaboration Diagram : Use Case Laporan Pembelian.....	114
Gambar 3.81 Analysis Collaboration Diagram : Use Case Print Laporan Pembelian.....	115
Gambar 3.82 Analysis Collaboration Diagram : Use Case Display Purchase Order	115
Gambar 3.83 Analysis Collaboration Diagram : Use Case Print Purchase Order	116
Gambar 3.84 Analysis Collaboration Diagram : Use Case Entri Pesanan	116
Gambar 3.85 Analysis Collaboration Diagram : Use Case Display Produk.....	116
Gambar 3.86 Entity Relationship Diagram ASCaMaDa	119

Daftar Tabel

Tabel 1.1 Tabel Istilah dan Singkatan	14
Tabel 3.1 Spesifikasi Use Case : Login.....	22
Tabel 3.2 Spesifikasi Use Case : Add User.....	23
Tabel 3.3 Spesifikasi Use Case : Edit Password.....	24
Tabel 3.4 Spesifikasi Use Case : Delete User.....	25
Tabel 3.5 Spesifikasi Use Case : Edit User.....	26
Tabel 3.6 Spesifikasi Use Case : Display User	27
Tabel 3.7 Spesifikasi Use Case : Entri Schedule	28
Tabel 3.8 Spesifikasi Use Case : Entri Reschedule	29
Tabel 3.9 Spesifikasi Use Case : Display Pesanan.....	30
Tabel 3.10 Spesifikasi Use Case : Display Schedule Produksi	31
Tabel 3.11 Spesifikasi Use Case : Display Reschedule Produksi	31
Tabel 3.12 Spesifikasi Use Case : Display CRP	32
Tabel 3.13 Spesifikasi Use Case : Display MRP	33
Tabel 3.14 Spesifikasi Use Case : Entri Purchase Order.....	33
Tabel 3.15 Spesifikasi Use Case : Display PO.....	34
Tabel 3.16 Spesifikasi Use Case : Surat Perintah Produksi.....	34
Tabel 3.17 Spesifikasi Use Case : Print SPP.....	35
Tabel 3.18 Spesifikasi Use Case : Add Produk.....	38
Tabel 3.19 Spesifikasi Use Case : Edit Produk	38
Tabel 3.20 Spesifikasi Use Case : Delete Produk.....	39
Tabel 3.21 Spesifikasi Use Case : Display Produk.....	40
Tabel 3.22 Spesifikasi Use Case : Entri Hasil Produksi	40
Tabel 3.23 Spesifikasi Use Case : Edit Produk	41
Tabel 3.24 Spesifikasi Use Case : Delete Hasil Produksi	42
Tabel 3.25 Spesifikasi Use Case : Display Hasil Produksi	43
Tabel 3.26 Spesifikasi Use Case : Laporan Hasil Produksi.....	43
Tabel 3.27 Spesifikasi Use Case : Print Laporan Produksi	44
Tabel 3.28 Spesifikasi Use Case : Add Mesin.....	45
Tabel 3.29 Spesifikasi Use Case : Edit Mesin.....	45
Tabel 3.30 Spesifikasi Use Case : Delete Mesin.....	46
Tabel 3.31 Spesifikasi Use Case : Edit Status Mesin.....	47
Tabel 3.32 Spesifikasi Use Case : Display SPP.....	48
Tabel 3.33 Spesifikasi Use Case : Add Bahan Produksi.....	50
Tabel 3.34 Spesifikasi Use Case : Edit Bahan Produksi	50
Tabel 3.35 Spesifikasi Use Case : Delete Bahan Produksi.....	51
Tabel 3.36 Spesifikasi Use Case : Display Bahan Produksi.....	52
Tabel 3.37 Spesifikasi Use Case : Entri Stock Keluar.....	53
Tabel 3.38 Spesifikasi Use Case : Entri Stock Masuk.....	53
Tabel 3.39 Spesifikasi Use Case : Add Kategori.....	54
Tabel 3.40 Spesifikasi Use Case : Edit Kategori.....	55
Tabel 3.41 Spesifikasi Use Case : Delete Kategori.....	56
Tabel 3.42 Spesifikasi Use Case : Laporan Bahan Baku Produksi	57
Tabel 3.43 Spesifikasi Use Case : Print Laporan Bahan Produksi.....	57
Tabel 3.44 Spesifikasi Use Case : Laporan Bahan Baku Produksi Keluar.....	58
Tabel 3.45 Spesifikasi Use Case : Print Laporan Bahan Baku Produksi Keluar	59
Tabel 3.47 Spesifikasi Use Case : Edit Status Bahan Baku.....	60
Tabel 3.48 Spesifikasi Use Case : Edit Status Barang Produksi.....	60
Tabel 3.48 Spesifikasi Use Case : Add Penjualan	62
Tabel 3.49 Spesifikasi Use Case : Edit Penjualan.....	63
Tabel 3.50 Spesifikasi Use Case : Edit Harga Produk	64
Tabel 3.51 Spesifikasi Use Case : Delete Penjualan	65
Tabel 3.52 Spesifikasi Use Case : Display Penjualan.....	65
Tabel 3.53 Spesifikasi Use Case : Print Penjualan	66
Tabel 3.54 Spesifikasi Use Case : Add Supplier	68
Tabel 3.55 Spesifikasi Use Case : Edit Supplier	69

Tabel 3.56 Spesifikasi Use Case : Delete Supplier	69
Tabel 3.57 Spesifikasi Use Case : Display DataSupplier	71
Tabel 3.58 Spesifikasi Use Case : Edit Harga Bahan Baku	71
Tabel 3.59 Spesifikasi Use Case : Order Bahan Baku	72
Tabel 3.60 Spesifikasi Use Case : Display Order Pembelian	73
Tabel 3.61 Spesifikasi Use Case : Laporan Pembelian	73
Tabel 3.62 Spesifikasi Use Case : Print Laporan Pemeriksaan	74
Tabel 3.63 Spesifikasi Use Case : Display Purchase Order	75
Tabel 3.64 Spesifikasi Use Case : Print Purchase Order	76
Tabel 3.65 Spesifikasi Use Case : Entri Pesanan	77
Tabel 3.66 Spesifikasi Use Case : Display Produk	78



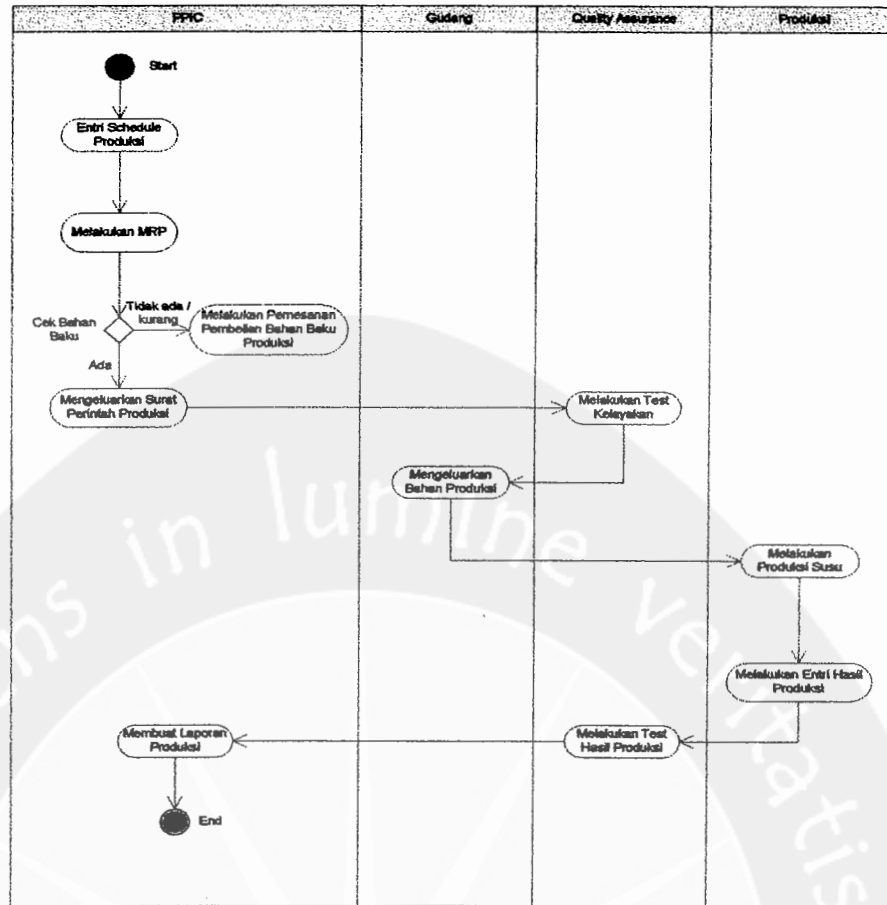
1 Pendahuluan

1.1 Tujuan

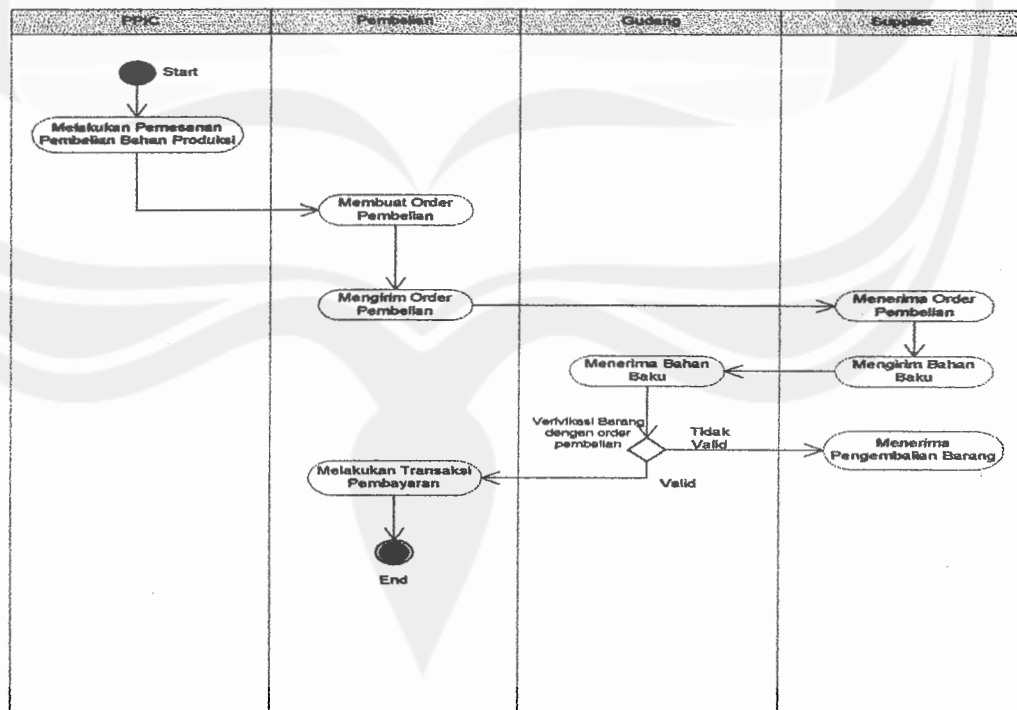
Dokumen ini berisi tentang penjelasan Spesifikasi Kebutuhan Perangkat Lunak (SKPL). Dokumen SKPL ini merupakan dokumen spesifikasi kebutuhan perangkat lunak ASCaMaDa yang akan dibangun. Dalam dokumen ini akan dijelaskan mengenai kebutuhan-kebutuhan yang harus tersedia agar perangkat lunak yang diharapkan dapat terwujud. Dokumen ini digunakan oleh pengembang perangkat lunak sebagai acuan teknis untuk pengembangan perangkat lunak ASCaMaDa.

1.2 Lingkup Masalah

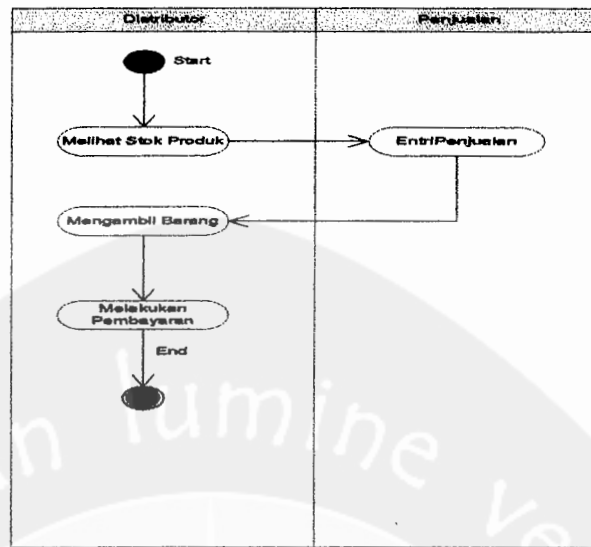
PT. Sari Husada dalam menjalankan bisnis prosesnya menggunakan aplikasi yang bersifat independent dalam artian hanya bisa dilihat oleh user dari perusahaan tersebut. Hubungan antara perusahaan dengan pihak diluar perusahaan berlangsung jika perusahaan mengirimkan informasi. Hal ini menyebabkan Chanel pada Supply Chain bersifat pasif. Oleh karena itu dibutuhkan sebuah perangkat lunak yang mampu menghubungkan Chanel pada Supply Chain sehingga aliran informasi tidak terputus. ASCaMaDa yang dibangun berbasis Web akan mampu menghubungkan antara pelaku-pelaku bisnis yaitu Supplier, Perusahaan dan Distributor secara *online* dan pertukaran informasi akan semakin efektif dan efisien.. Maka dari itu ASCaMaDa dibuat untuk menghubungkan Chanel pada Supply Chain Management yang telah diterapkan pada PT. Sari Husada melalui internet.



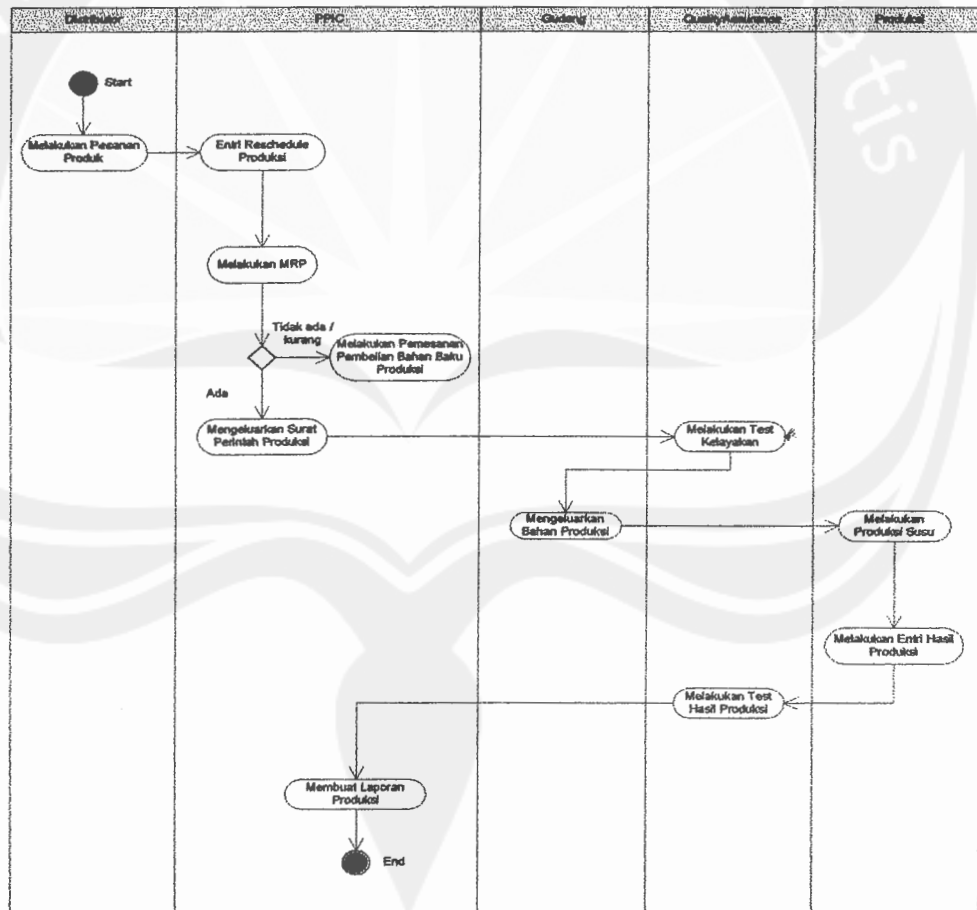
Gambar 1.1 Activity Diagram Produksi



Gambar 1.2 Activity Diagram Pembelian



Gambar 1.3 Activity Diagram Penjualan



Gambar 1.4 Activity Diagram Supply Reschedule Produksi

1.3 Istilah dan Singkatan

Daftar istilah dan singkatan yang digunakan :

Tabel 1.1 Tabel Istilah dan Singkatan

Keyword/Phrase	Definisi
SKPL	Spesifikasi Kebutuhan Perangkat Lunak, atau <i>SRS (Software Requirements Specification)</i> , merupakan spesifikasi perangkat lunak yang akan dirancang, dibuat dan dikembangkan.
ASCaMaDa	ASCaMaDa merupakan perangkat lunak yang dibangun untuk menghubungkan Chanel pada Supply Chain Management yang telah diterapkan pada PT. Sari Husada melalui internet.
SKPL-ASCaMaDa.K-XX	Kode yang merepresentasikan kebutuhan pada ASCaMaDa, dengan ASCaMaDa merupakan kode perangkat lunak, ASCaMaDa.K adalah kode fase dan XX adalah nomor kebutuhan.
SKPL-ASCaMaDa.UC-XX	Kode yang merepresentasikan <i>Use Case Diagram</i> pada ASCaMaDa, dengan ASCaMaDa merupakan kode perangkat lunak, ASCaMaDa.UC adalah <i>Use Case</i> dan XX adalah nomor urutan <i>Use Case</i> .

1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. A. Suhendar, Hariman Gunadi. *Visual Modeling Menggunakan UML dan RATIONAL ROSE*. Informatika Bandung. 2002.
2. Fowler, Martin. *UML Distilled 3th Ed., Panduan Singkat Bahasa Pemodelan Objek Standar*. ANDI Yogyakarta. 2005.
3. Larman, Craig. *Applying UML and patterns : an introduction to object-oriented analysis and design and the Unified Process*. Prentice-Hall, Inc. 2002.
4. Singgihraharja, Timotius Pamungkas. *Spesifikasi Kebutuhan Perangkat Lunak – Java Program Automatic Grading Tool*. Program Studi Teknik Informatika UAJY. 2005.

Program Studi Teknik Informatika	SKPL-ASCaMaDa	14/ 119
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1.5 Deskripsi umum (Overview)

Secara umum dokumen SKPL tersebut terbagi atas 4 bagian utama. Bagian pertama yaitu **Pendahuluan**, berisi penjelasan mengenai dokumen SKPL tersebut yang mencakup tujuan pembuatan SKPL, ruang lingkup masalah dalam pengembangan perangkat lunak tersebut, definisi, referensi dan deskripsi umum tentang dokumen SKPL ini. Bagian kedua adalah **Deskripsi Keseluruhan**, berisi penjelasan umum tentang perangkat lunak ASCaMaDa yang akan dikembangkan, mencakup perspektif produk yang akan dikembangkan, fungsi perangkat lunak, karakteristik pengguna, batasan dalam penggunaan perangkat lunak dan asumsi yang dipakai dalam pengembangan perangkat lunak ASCaMaDa tersebut. Bagian ketiga adalah **Deskripsi Rinci Kebutuhan**, berisi penjelasan secara lebih rinci tentang kebutuhan perangkat lunak ASCaMaDa yang akan dikembangkan. Bagian keempat adalah **Spesifikasi Kebutuhan Data**, berisi penjelasan tentang kebutuhan data yang akan digunakan.

2 Deskripsi Keseluruhan

2.1 Perspektif Produk

ASCaMaDa adalah perangkat lunak berbasis aplikasi Web yang dibangun untuk menghubungkan Chanel pada Supply Chain Management yang telah diterapkan pada PT. Sari Husada melalui internet Pengguna berinteraksi dengan sistem yang ditampilkan dalam bentuk halaman Web.

Secara garis besar, proses diawali dengan adanya *request* halaman Web dari *Client*, yang diteruskan dari browser ke *Web Server*, kemudian membuat *request* yang dikirimkan kepada *web server* tersebut. *Web server* mencari file yang berisi kumpulan instruksi tersebut. *Web server* mengenali tipe file yang di-*request* kemudian mengirimkannya kepada *handler* yang bertugas menangani tipe file tersebut. Yang selanjutnya dikompilasi, kemudian hasil eksekusinya dikirimkan kembali pada *browser* sebagai *response*.

2.2 Fungsi Produk

Fungsi produk perangkat lunak ASCaMaDa berdasarkan *user* adalah sebagai berikut :

A. ALL USER

A. 1 Fungsi Login (SKPL-ASCaMaDa.A-01), fungsi yang digunakan pengguna untuk masuk ke dalam sistem yang digunakan.

A. 2 Fungsi Change Password (SKPL-ASCaMaDa.A-02), fungsi yang digunakan pengguna untuk mengubah password.

B. ADMINISTRATOR

B. 1 Fungsi Add User (SKPL-ASCaMaDa.B-01), fungsi yang digunakan Administrator untuk menambah user.

B. 2 Fungsi Edit User (SKPL-ASCaMaDa.B-02), fungsi yang digunakan Administrator untuk mengubah data user.

B. 3 Fungsi Delete User (SKPL-ASCaMaDa.B-03), fungsi yang digunakan Administrator untuk mengubah data user.

B. 4 Fungsi Display User (SKPL-ASCaMaDa.B-04), fungsi yang digunakan Administrator untuk mengubah data user.

C. PPIC (*Production Planning and Inventory Control*)

Program Studi Teknik Informatika	SKPL-ASCaMaDa	16/ 119
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- C. 1 Fungsi Entri Schedule Produksi (SKPL-ASCaMaDa.C-01), fungsi yang digunakan Manager PPIC untuk melakukan penjadwalan produksi.
- C. 2 Fungsi Entri Reschedule Produksi (SKPL-ASCaMaDa.C-02), fungsi yang digunakan Manager PPIC untuk melakukan penjadwalan ulang produksi.
- C. 3 Fungsi Entri Purchase Order (SKPL-ASCaMaDa.C-03), fungsi yang digunakan Manager PPIC untuk melakukan permintaan pembelian bahan baku produksi.
- C. 4 Fungsi Surat Perintah Produksi (SKPL-ASCaMaDa.C-04), fungsi yang digunakan Manager PPIC untuk membuat Surat Perintah Produksi.
- C. 5 Fungsi Display Schedule Produksi (SKPL-ASCaMaDa.C-05), fungsi yang digunakan Manager PPIC untuk melihat Schedule Produksi.
- C. 6 Fungsi Display Reschedule Produksi (SKPL-ASCaMaDa.C-06), fungsi yang digunakan Manager PPIC untuk melihat Reschedule Produksi.
- C. 7 Fungsi Display Barang Pesanan (SKPL-ASCaMaDa.C-07), fungsi yang digunakan Manager PPIC untuk melihat barang pesanan Distributor.
- C. 8 Fungsi Display Purchase Order (SKPL-ASCaMaDa.C-08), fungsi yang digunakan Manager PPIC untuk melihat Purchase Order.
- C. 9 Fungsi Display Material Requirement Planning (MRP) (SKPL-ASCaMaDa.C-09), fungsi yang digunakan Manager PPIC untuk melihat MRP.
- C. 10 Fungsi Display Capacity Requirement Planning (CRP) (SKPL-ASCaMaDa.C-10), fungsi yang digunakan Manager PPIC untuk melihat CRP.
- C. 11 Fungsi Print SPP (SKPL-ASCaMaDa.C-11), fungsi yang digunakan Manager PPIC untuk mencetak Surat Perintah Produksi.
- D. Divisi Produksi
 - D. 1 Fungsi Add Produk (SKPL-ASCaMaDa.D-01), fungsi yang digunakan Manager Produksi untuk menambah produk baru.
 - D. 2 Fungsi Edit Produk (SKPL-ASCaMaDa.D-02), fungsi yang digunakan Manager Produksi untuk mengubah data produk.
 - D. 3 Fungsi Delete Produk (SKPL-ASCaMaDa.D-03), fungsi yang digunakan Manager Produksi untuk menghapus produk.

Program Studi Teknik Informatika	SKPL-ASCaMaDa	17/ 119
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- D. 4 Fungsi Display Produk (SKPL-ASCaMaDa.D-04), fungsi yang digunakan Manager Produksi untuk menampilkan data produk.
- D. 5 Fungsi Add Mesin (SKPL-ASCaMaDa.D-05), fungsi yang digunakan Manager Produksi untuk menambah data mesin.
- D. 6 Fungsi Edit Mesin (SKPL-ASCaMaDa.D-06), fungsi yang digunakan Manager Produksi untuk melakukan edit mesin.
- D. 7 Fungsi Delete Mesin (SKPL-ASCaMaDa.D-07), fungsi yang digunakan Manager Produksi untuk menghapus data mesin.
- D. 8 Fungsi Edit Status Mesin (SKPL-ASCaMaDa.D-08), fungsi yang digunakan Manager Produksi untuk mengubah status mesin.
- D. 9 Fungsi Entri Hasil Produksi (SKPL-ASCaMaDa.D-09), fungsi yang digunakan Manager Produksi untuk memasukkan data hasil produksi.
- D. 10 Fungsi Edit Hasil Produksi (SKPL-ASCaMaDa.D-10), fungsi yang digunakan Manager Produksi untuk mengubah hasil produksi.
- D. 11 Fungsi Delete Hasil Produksi (SKPL-ASCaMaDa.D-11), fungsi yang digunakan Manager Produksi untuk menghapus hasil produksi.
- D. 12 Fungsi Display Hasil Produksi (SKPL-ASCaMaDa.D-12), fungsi yang digunakan Manager Produksi untuk menampilkan hasil produksi.
- D. 13 Fungsi Laporan Hasil Produksi (SKPL-ASCaMaDa.D-13), fungsi yang digunakan Manager Produksi untuk membuat laporan hasil produksi.
- D. 14 Fungsi Display SPP (SKPL-ASCaMaDa.D-14), fungsi yang digunakan Manager Produksi untuk menampilkan SPP.

E. Divisi Warehouse

- E. 1 Fungsi Add Bahan Produksi (SKPL-ASCaMaDa.E-01), fungsi yang digunakan untuk Manager Warehouse menambah data bahan baku produksi.
- E. 2 Fungsi Edit Bahan Produksi (SKPL-ASCaMaDa.E-02), fungsi yang digunakan Manager Warehouse untuk mengubah data bahan baku produksi.
- E. 3 Fungsi Delete Bahan Produksi (SKPL-ASCaMaDa.E-03), fungsi yang digunakan Manager Warehouse untuk menghapus bahan baku produksi.

- E. 4 Fungsi Display Bahan Baku Produksi (SKPL-ASCaMaDa.E-04), fungsi yang digunakan Manager Warehouse untuk menampilkan data bahan baku produksi.
- E. 5 Fungsi Add Kategori Bahan Baku Produksi (SKPL-ASCaMaDa.E-05), fungsi yang digunakan Manager Warehouse untuk menambah data kategori bahan baku produksi.
- E. 6 Fungsi Edit Kategori Bahan Baku Produksi (SKPL-ASCaMaDa.E-06), fungsi yang digunakan Manager Warehouse untuk mengubah data kategori bahan baku produksi.
- E. 7 Fungsi Delete Kategori Bahan Baku Produksi (SKPL-ASCaMaDa.E-07), fungsi yang digunakan Manager Warehouse untuk menghapus kategori bahan baku produksi.
- E. 8 Fungsi Entri Stok Masuk (SKPL-ASCaMaDa.E-08), fungsi yang digunakan Manager Warehouse untuk memasukkan data stok masuk bahan baku produksi.
- E. 9 Fungsi Entri Stok Keluar (SKPL-ASCaMaDa.E-09), fungsi yang digunakan Manager Warehouse untuk memasukkan data stok keluar bahan baku produksi.
- E. 10 Fungsi Laporan Bahan Baku Produksi (SKPL-ASCaMaDa.E-10), fungsi yang digunakan Manager Warehouse untuk membuat Laporan Bahan Baku Produksi.
- E. 11 Fungsi Laporan Bahan Baku Produksi Keluar (SKPL-ASCaMaDa.E-11), fungsi yang digunakan Manager Warehouse untuk membuat Laporan Bahan Baku Produksi Keluar.
- F. Divisi Quality Assurance
- F. 1 Fungsi Edit Status Bahan Baku Produksi (SKPL-ASCaMaDa.F-01), fungsi yang digunakan Manager QA untuk mengubah status bahan baku produksi dari Quarantina menjadi Released atau Rejected.
- F. 2 Fungsi Edit Status Hasil Produksi (SKPL-ASCaMaDa.F-02), fungsi yang digunakan Manager QA untuk mengubah status hasil produksi dari Quarantine menjadi Released.
- G. Divisi Purchasing
- G. 1 Fungsi Add Supplier (SKPL-ASCaMaDa.G-01), fungsi yang digunakan Manager Pembelian untuk menambah data Supplier.

Program Studi Teknik Informatika	SKPL-ASCaMaDa	19/ 119
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- G. 2 Fungsi Edit Supplier (SKPL-ASCaMaDa.G-02), fungsi yang digunakan Manager Pembelian untuk mengubah data Supplier.
 - G. 3 Fungsi Delete Supplier (SKPL-ASCaMaDa.G-03), fungsi yang digunakan Manager Pembelian untuk menghapus data supplier.
 - G. 4 Fungsi Display Supplier (SKPL-ASCaMaDa.G-04), fungsi yang digunakan Manager Pembelian untuk menghapus data supplier.
 - G. 5 Fungsi Edit Harga Bahan Baku (SKPL-ASCaMaDa.G-05), fungsi yang digunakan Manager Pembelian untuk mengubah harga bahan baku.
 - G. 6 Fungsi Order Bahan Produksi (SKPL-ASCaMaDa.G-06), fungsi yang digunakan Manager Pembelian untuk melakukan Pembelian Bahan Baku Produksi.
 - G. 7 Fungsi Display Order Pembelian (SKPL-ASCaMaDa.G-07), fungsi yang digunakan Manager Pembelian untuk menampilkan data pembelian.
 - G. 8 Fungsi Laporan Pembelian (SKPL-ASCaMaDa.G-08), fungsi yang digunakan Manager Pembelian untuk membuat laporan pembelian.
- H. Divisi Marketing
- H. 1 Fungsi Add Penjualan (SKPL-ASCaMaDa.H-01), fungsi yang digunakan Manager Penjualan untuk menambah data penjualan.
 - H. 2 Fungsi Edit Penjualan (SKPL-ASCaMaDa.H-02), fungsi yang digunakan Manager Penjualan untuk mengubah data penjualan.
 - H. 3 Fungsi Edit Harga Produk (SKPL-ASCaMaDa.H-03), fungsi yang digunakan Manager Penjualan untuk mengubah harga produk.
 - H. 4 Fungsi Delete Penjualan (SKPL-ASCaMaDa.H-04), fungsi yang digunakan Manager Penjualan untuk menghapus data penjualan.
 - H. 5 Fungsi Display Penjualan (SKPL-ASCaMaDa.H-05), fungsi yang digunakan Manager Penjualan untuk menampilkan data penjualan.
 - H. 6 Fungsi Laporan Penjualan (SKPL-ASCaMaDa.H-06), fungsi yang digunakan Manager Penjualan untuk membuat laporan penjualan.
- I. *Supplier*
- I. 1 Fungsi Display Purchase Order (SKPL-ASCaMaDa.I-01), fungsi yang digunakan Supplier untuk menampilkan/mencetak permintaan pembelian Bahan Baku Produksi.
- J. Distributor

J. 1 Fungsi Entri Pesanan (SKPL-ASCaMaDa.J-01), fungsi yang digunakan Distributor untuk melakukan pemesanan Produk.

J. 2 Fungsi Display Produk (SKPL-ASCaMaDa.J-02), fungsi yang digunakan Distributor untuk melihat data produk yang telah siap diambil.

2.3 Karakteristik Pengguna

Pengguna perangkat lunak ASCaMaDa tersebut adalah *user* (Administrator, Manager PPIC, Manager Gudang, Manager Pembelian, Manager Penjualan, Manager Quality Assurance, Supplier dan Distributor) dengan karakteristik sebagai berikut :

1. *User*

- a. Pernah menggunakan aplikasi berbasis web.
- b. Mengerti dan memahami perangkat lunak yang digunakan.
- c. Memahami cara kerja sistem yang sedang dijalankan.

2.4 Batasan-batasan

Batasan-batasan dalam pengembangan perangkat lunak ASCaMaDa tersebut adalah :

1. Perangkat lunak ASCaMaDa hanya mencakup pengolahan terhadap data yang digunakan dalam Supply Chain Management pada PT. Sari Husada.
2. Berpedoman pada tujuan dari pengembangan perangkat lunak ASCaMaDa yaitu mendesain dan mengimplementasikan suatu sistem yang berguna untuk menangani pengolahan data dengan mudah, cepat dan efektif.
3. Distributor melakukan pemesanan secara on-line.

2.5 Asumsi dan Ketergantungan

Perangkat lunak ini dibangun dengan menggunakan teknologi *.NET*, khususnya *ASP.NET* untuk aplikasi *web* maka untuk sisi server dijalankan dengan sistem operasi windows 2000/XP/2003, sedangkan sisi client hanya membutuhkan web browser seperti Internet Explorer, Mozilla Firefox, Netscape Navigator atau web browser lainnya.

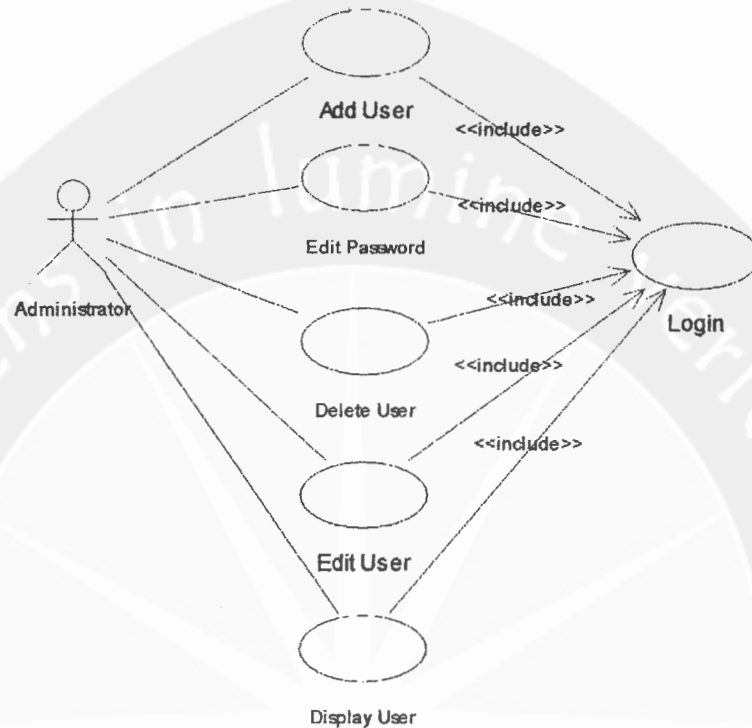
Program Studi Teknik Informatika	SKPL-ASCaMaDa	21/ 119
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

3 Deskripsi Rinci Kebutuhan

3.1 Spesifikasi Kebutuhan Fungsionalitas

3.1.1 Use Case Diagram Administrator

Secara umum fungsi-fungsi yang dimiliki oleh Administrator adalah:



Gambar 3.1 Use Case Diagram Administrator

3.1.1.1 Spesifikasi Use Case : Login

Tabel 3.1 Spesifikasi Use Case : Login

Use Case ID	SKPL-ASCaMaDa.UC-01
Use Case Name	Login
Use Case Type	Essential
Priority	High
Actors	User (Administrator, Supplier, Distributor, Manager PPIC, Manager Pembelian, Manager Penjualan, Manager Warehouse, Manager QA dan Manager Produksi)

Description	Use case ini digunakan user untuk memperoleh akses ke sistem. Login didasarkan pada sebuah username unik dan password yang berupa rangkaian karakter.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk login. 2. Aktor memasukkan username dan password. 3. Sistem memeriksa username dan password aktor. 4. Sistem memberikan akses ke aktor sesuai statusnya (Admin, Supplier, Distributor, Manager PPIC, Manager Pembelian, Manager Penjualan, Manager Warehouse, Manager QA dan Manager Produksi).
Alternative Paths	-
Postconditions	Aktor memasuki sistem dan dapat menggunakan fungsi-fungsi pada sistem.
Exception Paths	<ol style="list-style-type: none"> 1. Username atau password tidak sesuai (setelah Tabel 3.1 Basic Path 3) <ol style="list-style-type: none"> a. Sistem menampilkan peringatan bahwa username atau password tidak sesuai. b. Kembali ke Tabel 3.1 Basic Path 2.
Extends	-
Includes	-

3.1.1.2 Spesifikasi Use Case : Add User

Tabel 3.2 Spesifikasi Use Case : Add User

Use Case ID	SKPL- ASCaMaDa.UC-02
Use Case Name	Add User
Use Case Type	Essential
Priority	High
Actors	Administrator

Description	Use case ini digunakan Adminhistrator untuk menambahkan user baru.
Preconditions	Use Case : Login sudah dilaksanakan dan aktor sudah berhasil memasuki sistem.
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk memasukkan data user baru. 2. Aktor memasukkan detail data user baru. 3. Sistem memeriksa data-data yang dimasukkan aktor. 4. Aktor meminta sistem untuk menambahkan data baru ke basisdata. 5. Sistem menyimpan detail data yang telah diisi oleh aktor.
Alternative Paths	-
Postconditions	Data user berhasil disimpan.
Exception Paths	<ol style="list-style-type: none"> 1. Detail data yang diisikan belum lengkap (setelah Tabel 3.2 langkah 3) <ol style="list-style-type: none"> a. Sistem menampilkan peringatan bahwa data yang diisikan belum lengkap. b. Kembali ke Tabel 3.2 langkah 2.
Extends	-
Includes	Use Case : Login

3.1.1.3 Spesifikasi Use Case : Edit Password

Tabel 3.3 Spesifikasi Use Case : Edit Password

Use Case ID	SKPL-ASCaMaDa.UC-03
Use Case Name	Edit Password
Use Case Type	Essential
Priority	High
Actors	User
Description	Use case ini digunakan Administrator untuk mengubah password yang digunakan ketika Login ke

	sistem.
Preconditions	Use Case : Login sudah dilaksanakan dan aktor sudah berhasil memasuki sistem.
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk mengubah password. 2. Aktor memasukkan password lama lalu memasukkan password baru dan diulangi dua kali sebagai konfirmasi password baru. 3. Aktor meminta sistem untuk menyimpan data baru ke basisdata. 4. Sistem memeriksa validitas data. 5. Sistem menyimpan perubahan data password.
Alternative Paths	-
Postconditions	Data password baru berhasil disimpan.
Exception Paths	<ol style="list-style-type: none"> 1. Data yang dimasukkan tidak valid (setelah Tabel 3.3 langkah 4) <ol style="list-style-type: none"> a. Sistem menampilkan peringatan bahwa data tidak valid b. Kembali ke Tabel 3.3 langkah 2.
Extends	-
Includes	Use Case : Login

3.1.1.4 Spesifikasi Use Case : Delete User

Tabel 3.4 Spesifikasi Use Case : Delete User

Use Case ID	SKPL-ASCaMaDa.UC-04
Use Case Name	Delete User
Use Case Type	Essential
Priority	High
Actors	Administator
Description	Use case ini digunakan untuk menghapus data user.
Preconditions	-
Basic Path	1. Sistem menampilkan antar muka untuk

	<p>menghapus user.</p> <ol style="list-style-type: none"> 2. Sistem menampilkan data-data user. 3. Aktor memilih data user yang akan dihapus. 4. Aktor meminta system untuk menghapus data user dari basis data. 5. Sistem menghapus data user dari basis data.
Alternative Paths	-
Postconditions	Data user berhasil dihapus.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.1.5 Spesifikasi Use Case : Edit User

Tabel 3.5 Spesifikasi Use Case : Edit User

Use Case ID	SKPL-ASCaMaDa.UC-05
Use Case Name	Edit User
Use Case Type	Essential
Priority	High
Actors	Administrator
Description	Use case ini digunakan untuk mengubah data user.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Aktor memilih data user yang akan di-update. 2. Aktor memasukkan data baru 3. Aktor meminta sistem untuk menyimpan data yang baru. 4. Sistem menyimpan data yang telah diubah.
Alternative Paths	-
Postconditions	Data user berhasil diubah.
Exception Paths	<ol style="list-style-type: none"> 1. Data User melebihi 25 karakter (setelah Tabel 3.5 langkah 3) <ol style="list-style-type: none"> a. Sistem menampilkan peringatan bahwa data user terlalu panjang.

	b. Kembali ke Tabel 3.5 langkah 2.
Extends	-
Includes	Use Case : Login

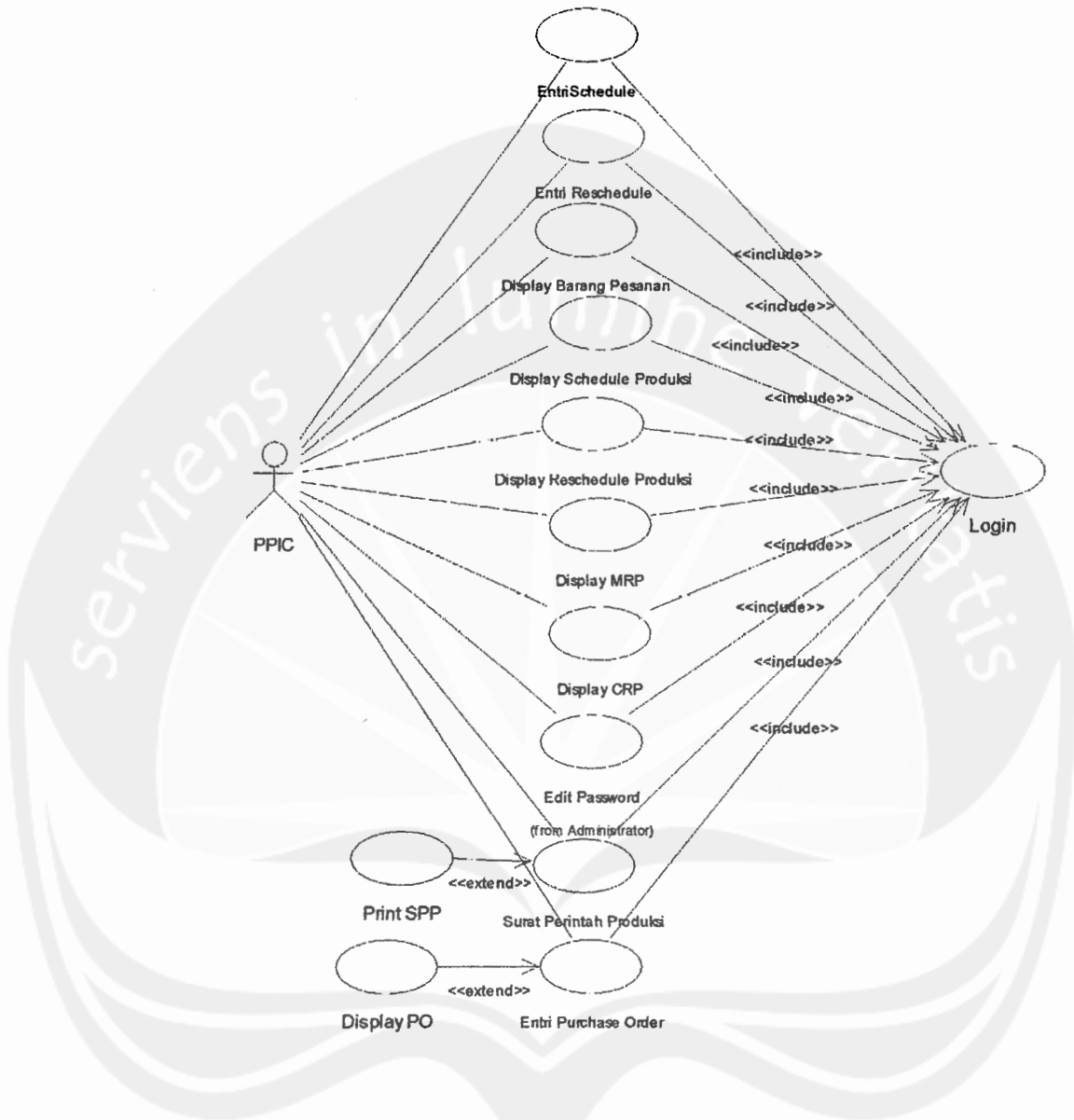
3.1.1.6 Spesifikasi Use Case : Display User

Tabel 3.6 Spesifikasi Use Case : Display User

Use Case ID	SKPL-ASCaMaDa.UC-06
Use Case Name	Display User
Use Case Type	Essential
Priority	High
Actors	Administrator
Description	Use case ini digunakan untuk melihat data user.
Preconditions	-
Basic Path	1. Sistem menampilkan data user.
Alternative Paths	-
Postconditions	Data user ditampilkan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.2 Use Case Diagram Production Planning and Inventory Control

Secara umum fungsi-fungsi yang dimiliki oleh PPIC adalah:



Gambar 3.2 Use Case Diagram PPIC

3.1.2.1 Spesifikasi Use Case : Entri Schedule

Tabel 3.7 Spesifikasi Use Case : Entri Schedule

Use Case ID	SKPL- ASCaMaDa.UC-07
Use Case Name	Entri Schedule

Use Case Type	Essential
Priority	High
Actors	Manager PPIC
Description	Use case ini digunakan Manage PPIC untuk melakukan entri data Schedule.
Preconditions	Use Case : Pengelolaan PPIC telah dijalankan dan Aktor telah masuk ke sistem antarmuka Entri Schedule.
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk melakukan entri data <i>schedule</i>. 2. Aktor memasukkan tanggal awal produksi, tanggal akhir produksi, barang yang akan diproduksi dan jumlah barang yang diproduksi. 3. Aktor meminta sistem untuk menyimpan data <i>schedule</i> ke basisdata. 4. Sistem menyimpan data <i>schedule</i> ke basis data.
Alternative Paths	-
Postconditions	Data <i>schedule</i> berhasil disimpan.
Exception Paths	<ol style="list-style-type: none"> 1. Detail data yang diisikan belum lengkap (setelah Tabel 3.8 langkah 3) <ol style="list-style-type: none"> a. Sistem menampilkan peringatan bahwa data yang diisikan belum lengkap. b. Kembali ke Tabel 3.8 langkah 2.
Extends	-
Includes	Use Case : Login

3.1.2.2 Spesifikasi Use Case : Entri Reschedule

Tabel 3.8 Spesifikasi Use Case : Entri Reschedule

Use Case ID	SKPL-ASCaMaDa.UC-08
Use Case Name	Entri Reschedule
Use Case Type	Essential
Priority	High

Actors	Manager PPIC
Description	Use case ini digunakan untuk melakukan penjadwalan ulang produksi yang telah dilakukan sebelumnya.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka Reschedule Produksi. 2. Aktor melakukan <i>entri</i> data schedule yang baru. 3. Aktor meminta sistem untuk melakukan perubahan jadwal produksi. 4. Sistem menyimpan data reschedule produksi pada basisdata.
Alternative Paths	-
Postconditions	Data reschedule produksi berhasil disimpan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.2.3 Spesifikasi Use Case : Display Pesanan

Tabel 3.9 Spesifikasi Use Case : Display Pesanan

Use Case ID	SKPL-ASCaMaDa.UC-09
Use Case Name	Display Pesanan
Use Case Type	Essential
Priority	High
Actors	Manager PPIC
Description	Use case ini digunakan untuk melihat data pesanan barang.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka Display Pesanan. 2. Sistem menampilkan data pesanan barang.

Alternative Paths	-
Postconditions	Data pesanan ditampilkan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.2.4 Spesifikasi Use Case : Display Schedule Produksi

Tabel 3.10 Spesifikasi Use Case : Display Schedule Produksi

Use Case ID	SKPL-ASCaMaDa.UC-10
Use Case Name	Display Schedule Produksi
Use Case Type	Essential
Priority	High
Actors	Manager PPIC
Description	Use case ini digunakan untuk melihat data <i>schedule</i> produksi.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka display schedule produksi. 2. Sistem menampilkan data schedule produksi.
Alternative Paths	-
Postconditions	Data schedule produksi ditampilkan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.2.5 Spesifikasi Use Case : Display Reschedule Produksi

Tabel 3.11 Spesifikasi Use Case : Display Reschedule Produksi

Use Case ID	SKPL-ASCaMaDa.UC-11
Use Case Name	Display Reschedule Produksi
Use Case Type	Essential

Priority	High
Actors	Manager PPIC
Description	Use case ini digunakan untuk melihat data <i>reschedule</i> produksi.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka display <i>reschedule</i> produksi. 2. Sistem menampilkan data <i>reschedule</i> produksi.
Alternative Paths	-
Postconditions	Data <i>reschedule</i> produksi ditampilkan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.2.6 Spesifikasi Use Case : Display CRP

Tabel 3.12 Spesifikasi Use Case : Display CRP

Use Case ID	SKPL-ASCaMaDa.UC-12
Use Case Name	Display CRP
Use Case Type	Essential
Priority	High
Actors	Manager PPIC
Description	Use case ini digunakan untuk melihat data CRP (Capacity Requirement Planning).
Preconditions	-
Basic Path	1. Sistem menampilkan data CRP yang terdapat dalam basisdata.
Alternative Paths	-
Postconditions	Data CRP ditampilkan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.2.7 Spesifikasi Use Case : Display MRP

Tabel 3.13 Spesifikasi Use Case : Display MRP

Use Case ID	SKPL-ASCaMaDa.UC-13
Use Case Name	Display MRP
Use Case Type	Essential
Priority	High
Actors	Manager PPIC
Description	Use case ini digunakan untuk melihat data MRP (Material Requirement Planning)
Preconditions	-
Basic Path	1. Sistem menampilkan data MRP yaitu bahan baku produksi yang tersedia di gudang.
Alternative Paths	-
Postconditions	Data MRP ditampilkan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.2.8 Spesifikasi Use Case : Entri Purchase Order

Tabel 3.14 Spesifikasi Use Case : Entri Purchase Order

Use Case ID	SKPL-ASCaMaDa.UC-14
Use Case Name	Entri Purchase Order
Use Case Type	Essential
Priority	High
Actors	Manager PPIC
Description	Use case ini digunakan untuk melakukan pemesanan bahan baku produksi pada supplier.
Preconditions	-
Basic Path	1. Sistem menampilkan antarmuka untuk melakukan pemesanan bahan baku produksi. 2. Aktor memasukkan data bahan produksi yang

	ingin dipesan. 3. Aktor meminta sistem untuk menyimpan data pesana ke basisdata. 4. Sistem menyimpan data pesanan ke basisdata.
Alternative Paths	-
Postconditions	Data pesanan bahan baku produksi berhasil disimpan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.2.9 Spesifikasi Use Case : Dispaly PO

Tabel 3.15 Spesifikasi Use Case : Display PO

Use Case ID	SKPL-ASCaMaDa.UC-15
Use Case Name	Display PO
Use Case Type	Essential
Priority	High
Actors	Manager PPIC
Description	Use case ini digunakan untuk menampilkan data pesanan bahan baku produksi.
Preconditions	-
Basic Path	1. Sistem menampilkan data PO.
Alternative Paths	-
Postconditions	Data pesanan bahan baku produksi ditampilkan.
Exception Paths	-
Extends	Use Case : Purchase Order
Includes	-

3.1.2.10 Spesifikasi Use Case : Surat Perintah Produksi

Tabel 3.16 Spesifikasi Use Case : Surat Perintah Produksi

Use Case ID	SKPL-ASCaMaDa.UC-16
-------------	---------------------

Program Studi Teknik Informatika	SKPL-ASCaMaDa	34/ 119
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Use Case Name	Surat Perintah Produksi
Use Case Type	Essential
Priority	High
Actors	Manager PPIC
Description	Use case ini digunakan untuk membuat surat perintah produksi.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk membuat surat perintah produksi. 2. Aktor memasukkan kode schedule produksi atau kode reschedule produksi, data produk dan bahan baku yang diperlukan untuk melakukan produksi. 3. Sistem menyimpan data ke basisdata.
Alternative Paths	-
Postconditions	Data berhasil disimpan di database.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.2.11 Spesifikasi Use Case : Print SPP

Tabel 3.17 Spesifikasi Use Case : Print SPP

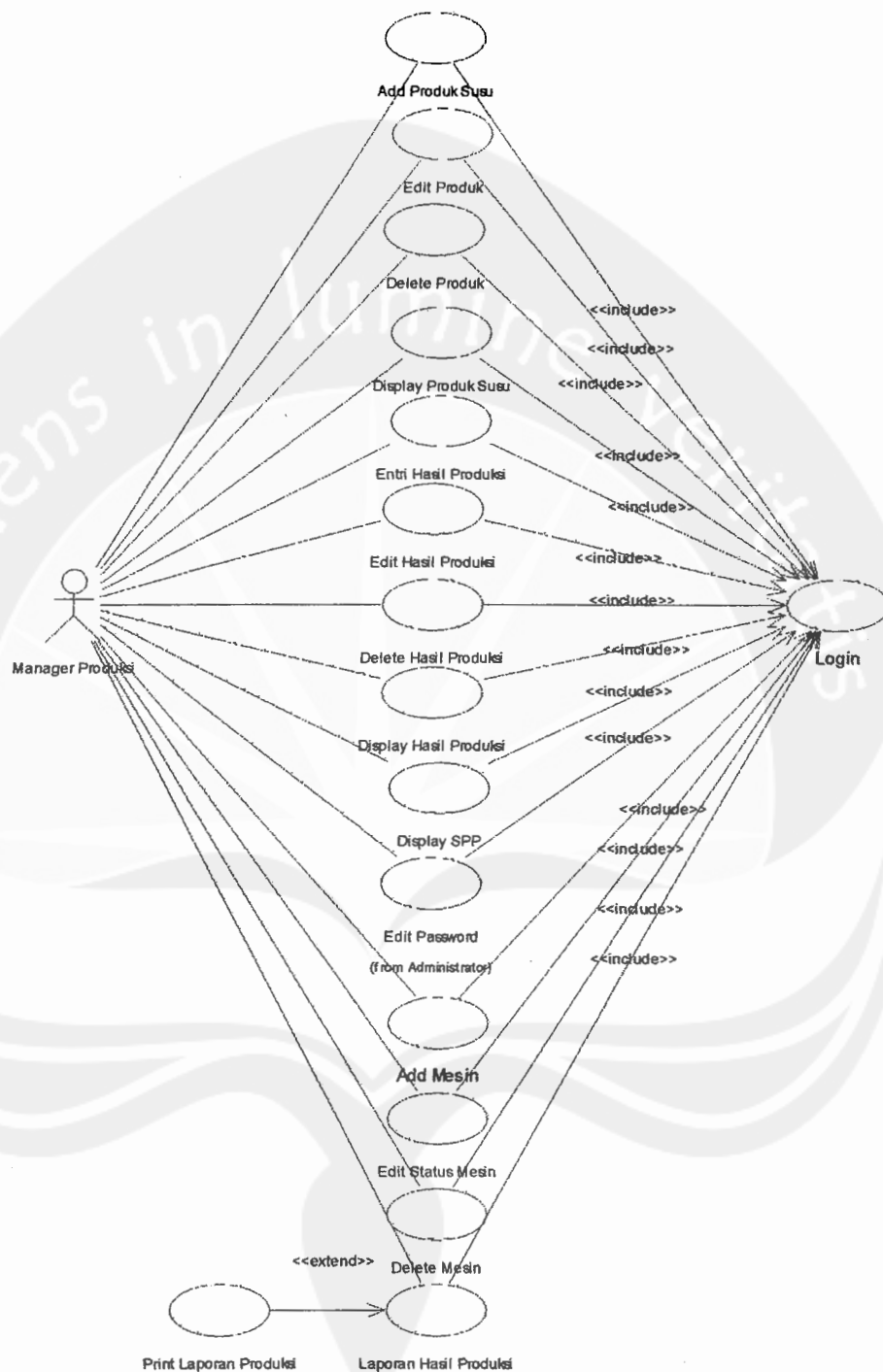
Use Case ID	SKPL-ASCaMaDa.UC-17
Use Case Name	Print SPP
Use Case Type	Essential
Priority	High
Actors	Manager PPIC
Description	Use case ini digunakan untuk mencetak surat perintah produksi pada kertas.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Aktor meminta sistem untuk mencetak surat perintah produksi pada kertas.

	2. Sistem mencetak surat perintah produks pada kertas.
Alternative Paths	-
Postconditions	SPP berhasil dicetak pada kertas.
Exception Paths	-
Extends	Use Case : Surat Perintah Produksi
Includes	-



3.1.3 Use Case Diagram Production

Secara umum fungsi-fungsi yang dimiliki oleh Production adalah:



Gambar 3.3 Use Case Diagram Production

3.1.3.1 Spesifikasi Use Case : Add Produk

Tabel 3.18 Spesifikasi Use Case : Add Produk

Use Case ID	SKPL- ASCaMaDa.UC-18
Use Case Name	Add Produk
Use Case Type	Essential
Priority	High
Actors	Manager Produksi
Description	Use case ini digunakan Manager Produksi untuk menambahkan data produk susu baru.
Preconditions	-
Basic Path	<ol style="list-style-type: none">1. Sistem menampilkan antarmuka untuk memasukkan data produk baru.2. Aktor memasukkan detail data produk baru.3. Sistem memeriksa data-data yang dimasukkan aktor.4. Aktor meminta sistem untuk menambahkan data baru ke basisdata.5. Sistem menyimpan detail data yang telah diisi oleh aktor.
Alternative Paths	-
Postconditions	Data produk berhasil disimpan.
Exception Paths	<ol style="list-style-type: none">1. Detail data yang diisikan belum lengkap (setelah Tabel 3.20 langkah 3)<ol style="list-style-type: none">a. Sistem menampilkan peringatan bahwa data yang diisikan belum lengkap.b. Kembali ke Tabel 3.20 langkah 2.
Extends	-
Includes	Use Case : Login

3.1.3.2 Spesifikasi Use Case : Edit Produk

Tabel 3.19 Spesifikasi Use Case : Edit Produk

Program Studi Teknik Informatika	SKPL-ASCaMaDa	38/ 119
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Use Case ID	SKPL-ASCaMaDa.UC-19
Use Case Name	Edit Produk
Use Case Type	Essential
Priority	High
Actors	Manager Produksi
Description	Use case ini digunakan Manager Produksi untuk mengubah data produk yang telah ada.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk mengubah data produk. 2. Aktor memasukkan data produk yang baru. 3. Aktor meminta sistem untuk menyimpan data baru ke basisdata. 4. Sistem menyimpan perubahan data produk.
Alternative Paths	-
Postconditions	Data produk yang baru berhasil disimpan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.3.3 Spesifikasi Use Case : Delete Produk

Tabel 3.20 Spesifikasi Use Case : Delete Produk

Use Case ID	SKPL-ASCaMaDa.UC-20
Use Case Name	Delete Produk
Use Case Type	Essential
Priority	High
Actors	Manager Produksi
Description	Use case ini digunakan untuk menghapus data produk.
Preconditions	-
Basic Path	1. Sistem menampilkan antar muka untuk

	<p>menghapus produk.</p> <ol style="list-style-type: none"> 2. Sistem menampilkan data-data produk. 3. Aktor memilih data produk yang akan dihapus. 4. Aktor meminta sistem untuk menghapus data produk dari basis data. 5. Sistem menghapus data produk dari basis data.
Alternative Paths	-
Postconditions	Data produk berhasil dihapus.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.3.4 Spesifikasi Use Case : Display Produk

Tabel 3.21 Spesifikasi Use Case : Display Produk

Use Case ID	SKPL-ASCaMaDa.UC-21
Use Case Name	Display Produk
Use Case Type	Essential
Priority	High
Actors	Manager Produksi
Description	Use case ini digunakan untuk melihat data produk.
Preconditions	-
Basic Path	1. Sistem menampilkan data produk yang tersimpan di database.
Alternative Paths	-
Postconditions	Data produk ditampilkan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.3.5 Spesifikasi Use Case : Entri Hasil Produksi

Tabel 3.22 Spesifikasi Use Case : Entri Hasil Produksi

Program Studi Teknik Informatika	SKPL-ASCaMaDa	40/ 119
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Use Case ID	SKPL- ASCaMaDa.UC-22
Use Case Name	Entri Hasil Produksi
Use Case Type	Essential
Priority	High
Actors	Manager Produksi
Description	Use case ini digunakan Manager Produksi untuk menambahkan hasil produksi.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk memasukkan data hasil produksi. 2. Aktor memasukkan detail data produksi. 3. Aktor meminta sistem untuk menambahkan data ke basisdata. 4. Sistem menyimpan detail data yang telah diisi oleh aktor.
Alternative Paths	-
Postconditions	Data produk berhasil disimpan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.3.6 Spesifikasi Use Case : Edit Hasil Produksi

Tabel 3.23 Spesifikasi Use Case : Edit Produk

Use Case ID	SKPL-ASCaMaDa.UC-23
Use Case Name	Edit Hasil Produksi
Use Case Type	Essential
Priority	High
Actors	Manager Produksi
Description	Use case ini digunakan Manager Produksi untuk mengubah data hasil produksi yang telah ada.
Preconditions	-

Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk mengubah data hasil produksi. 2. Aktor memasukkan data yang baru. 3. Aktor meminta sistem untuk menyimpan data baru ke basisdata. 4. Sistem menyimpan perubahan data hasil produksi.
Alternative Paths	-
Postconditions	Data hasil produksi yang baru berhasil disimpan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.3.7 Spesifikasi Use Case : Delete Hasil Produksi

Tabel 3.24 Spesifikasi Use Case : Delete Hasil Produksi

Use Case ID	SKPL-ASCaMaDa.UC-24
Use Case Name	Delete Hasil Produksi
Use Case Type	Essential
Priority	High
Actors	Manager Produksi
Description	Use case ini digunakan untuk menghapus data hasil produksi.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antar muka untuk menghapus data hasil produksi. 2. Sistem menampilkan data-data hasil produk. 3. Aktor memilih data yang akan dihapus. 4. Aktor meminta sistem untuk menghapus data basis data. 5. Sistem menghapus data dari basis data.
Alternative Paths	-
Postconditions	Data hasil produksi berhasil dihapus.

Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.3.8 Spesifikasi Use Case : Display Hasil Produksi

Tabel 3.25 Spesifikasi Use Case : Display Hasil Produksi

Use Case ID	SKPL-ASCaMaDa.UC-25
Use Case Name	Display Hasil Produksi
Use Case Type	Essential
Priority	High
Actors	Manager Produksi
Description	Use case ini digunakan untuk melihat data hasil produk.
Preconditions	-
Basic Path	1. Sistem menampilkan data hasil produksi yang tersimpan di datrabase.
Alternative Paths	-
Postconditions	Data hasil produksi ditampilkan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.3.9 Spesifikasi Use Case : Laporan Hasil Produksi

Tabel 3.26 Spesifikasi Use Case : Laporan Hasil Produksi

Use Case ID	SKPL-ASCaMaDa.UC-26
Use Case Name	Laporan Hasil Produksi
Use Case Type	Essential
Priority	High
Actors	Manager Produksi
Description	Use case ini digunakan untuk membuat Laporan

	Hasil Produksi.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk membuat laporan hasil produksi. 2. Aktor memasukkan tanggal produksi. 3. Sistem menampilkan laporan hasil produksi dalam bentuk report.
Alternative Paths	-
Postconditions	Laporan hasil produksi berhasil dibuat.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.3.10 Spesifikasi Use Case : Print Laporan Produksi

Tabel 3.27 Spesifikasi Use Case : Print Laporan Produksi

Use Case ID	SKPL-ASCaMaDa.UC-27
Use Case Name	Print Laporan Produksi
Use Case Type	Essential
Priority	High
Actors	Manager Produksi
Description	Use case ini digunakan untuk mencetak laporan produksi pada kertas.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Aktor meminta sistem untuk mencetak laporan produksi. 2. Sistem mencetak laporan produksi pada kertas.
Alternative Paths	-
Postconditions	Laporan produksi berhasil dicetak pada kertas.
Exception Paths	-
Extends	Use Case : Laporan Hasil Produksi
Includes	-

3.1.3.11 Spesifikasi Use Case : Add Mesin

Tabel 3.28 Spesifikasi Use Case : Add Mesin

Use Case ID	SKPL- ASCaMaDa.UC-28
Use Case Name	Add Mesin
Use Case Type	Essential
Priority	High
Actors	Manager Produksi
Description	Use case ini digunakan Manager Produksi untuk menambahkan data mesin baru.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk memasukkan data mesin baru. 2. Aktor memasukkan detail data mesin baru. 3. Sistem memeriksa data-data yang dimasukkan aktor. 4. Aktor meminta sistem untuk menambahkan data baru ke basisdata. 5. Sistem menyimpan detail data yang telah diisi oleh aktor.
Alternative Paths	-
Postconditions	Data mesin berhasil disimpan.
Exception Paths	<ol style="list-style-type: none"> 1. Detail data yang diisikan belum lengkap (setelah Tabel 3.30 langkah 3) <ol style="list-style-type: none"> a. Sistem menampilkan peringatan bahwa data yang diisikan belum lengkap. b. Kembali ke Tabel 3.30 langkah 2.
Extends	-
Includes	Use Case : Login

3.1.3.12 Spesifikasi Use Case : Edit Mesin

Tabel 3.29 Spesifikasi Use Case : Edit Mesin

Use Case ID	SKPL-ASCaMaDa.UC-29
-------------	---------------------

Program Studi Teknik Informatika	SKPL-ASCaMaDa	45/ 119
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Use Case Name	Edit Mesin
Use Case Type	Essential
Priority	High
Actors	Manager Produksi
Description	Use case ini digunakan Manager Produksi untuk mengubah data mesin yang telah ada.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk mengubah data mesin. 2. Aktor memasukkan data mesin yang baru. 3. Aktor meminta sistem untuk menyimpan data baru ke basisdata. 4. Sistem menyimpan perubahan data mesin.
Alternative Paths	-
Postconditions	Data mesin yang baru berhasil disimpan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.3.13 Spesifikasi Use Case : Delete Mesin

Tabel 3.30 Spesifikasi Use Case : Delete Mesin

Use Case ID	SKPL-ASCaMaDa.UC-30
Use Case Name	Delete Mesin
Use Case Type	Essential
Priority	High
Actors	Manager Produksi
Description	Use case ini digunakan untuk menghapus data mesin.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antar muka untuk menghapus mesin.

	<ol style="list-style-type: none"> 2. Sistem menampilkan data-data mesin. 3. Aktor memilih data mesin yang akan dihapus. 4. Aktor meminta sistem untuk menghapus data mesin dari basis data. 5. Sistem menghapus data mesin dari basis data.
Alternative Paths	-
Postconditions	Data mesin berhasil dihapus.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.3.14 Spesifikasi Use Case : Edit Status Mesin

Tabel 3.31 Spesifikasi Use Case : Edit Status Mesin

Use Case ID	SKPL- ASCaMaDa.UC-31
Use Case Name	Edit Status Mesin
Use Case Type	Essential
Priority	High
Actors	Manager Produksi
Description	Use case ini digunakan Manager Produksi untuk mengubah status mesin dari <i>use</i> ke <i>ready</i> atau sebaliknya.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk melakukan perubahan status mesin. 2. Aktor memasukkan data status yang baru. 3. Aktor meminta sistem untuk menambahkan data baru ke basisdata. 4. Sistem meng-<i>update</i> data yang telah diisi oleh aktor.
Alternative Paths	-
Postconditions	Data mesin berhasil di- <i>update</i> .
Exception Paths	-

Extends	-
Includes	Use Case : Login

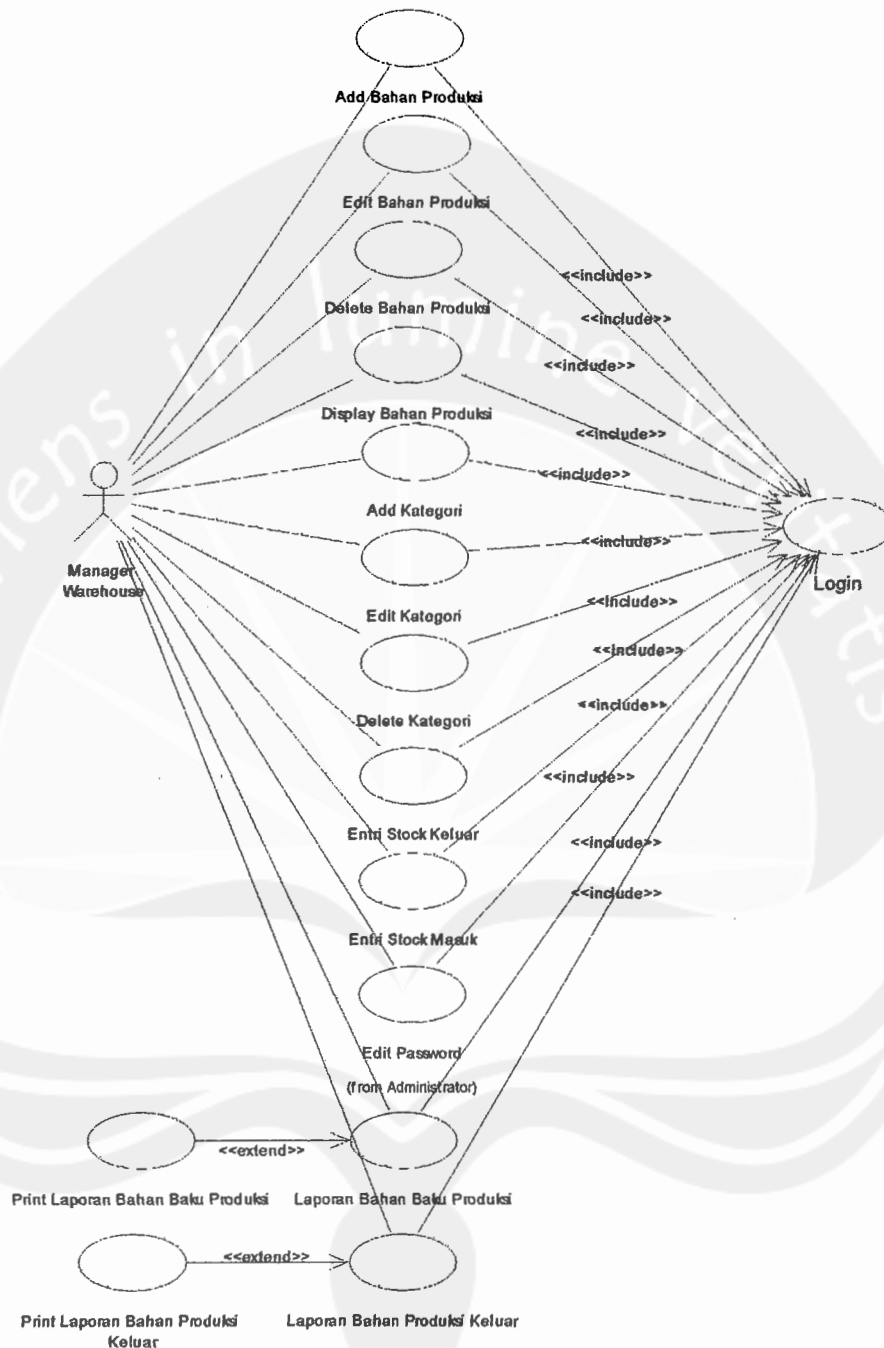
3.1.3.15 Spesifikasi Use Case : Display SPP

Tabel 3.32 Spesifikasi Use Case : Display SPP

Use Case ID	SKPL-ASCaMaDa.UC-32
Use Case Name	Display SPP
Use Case Type	Essential
Priority	High
Actors	Manager Produksi
Description	Use case ini digunakan Manager Produksi untuk meliah data Surat Perintah Produksi
Preconditions	-
Basic Path	1. Sistem menampilkan antarmuka untuk melihat data SPP 2. Sistem menampilkan data SPP.
Alternative Paths	-
Postconditions	Data SPP ditampilkan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.4 Use Case Diagram Warehouse

Secara umum fungsi-fungsi yang dimiliki oleh Warehouse adalah:



Gambar 3.4 Use Case Diagram Warehouse

3.1.4.1 Spesifikasi Use Case : Add Bahan Produksi

Tabel 3.33 Spesifikasi Use Case : Add Bahan Produksi

Use Case ID	SKPL- ASCaMaDa.UC-33
Use Case Name	Add Bahan Produksi
Use Case Type	Essential
Priority	High
Actors	Manager Warehouse
Description	Use case ini digunakan Manager Warehouse untuk menambahkan data bahan produksi baru.
Preconditions	-
Basic Path	<ol style="list-style-type: none">1. Sistem menampilkan antarmuka untuk memasukkan data bahan produksi.2. Aktor memasukkan detail data bahan produksi.3. Sistem memeriksa data-data yang dimasukkan aktor.4. Aktor meminta sistem untuk menambahkan data baru ke basisdata.5. Sistem menyimpan detail data yang telah diisi oleh aktor.
Alternative Paths	-
Postconditions	Data bahan produksi berhasil disimpan.
Exception Paths	<ol style="list-style-type: none">1. Detail data yang diisikan belum lengkap (setelah Tabel 3.36 langkah 3)<ol style="list-style-type: none">a. Sistem menampilkan peringatan bahwa data yang diisikan belum lengkap.b. Kembali ke Tabel 3.36 langkah 2.
Extends	-
Includes	Use Case : Login

3.1.4.2 Spesifikasi Use Case : Edit Bahan Produksi

Tabel 3.34 Spesifikasi Use Case : Edit Bahan Produksi

Use Case ID	SKPL-ASCaMaDa.UC-34
-------------	---------------------

Program Studi Teknik Informatika	SKPL-ASCaMaDa	50/ 119
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Use Case Name	Edit Bahan Produksi
Use Case Type	Essential
Priority	High
Actors	Manager Warehouse
Description	Use case ini digunakan Manager Warehouse untuk mengubah data bahan produksi yang telah ada.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk mengubah data bahan produksi. 2. Aktor memasukkan data bahan produksi yang baru. 3. Aktor meminta sistem untuk menyimpan data baru ke basisdata. 4. Sistem menyimpan perubahan data bahan produksi.
Alternative Paths	-
Postconditions	Data bahan produksi yang baru berhasil disimpan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.4.3 Spesifikasi Use Case : Delete Bahan Produksi

Tabel 3.35 Spesifikasi Use Case : Delete Bahan Produksi

Use Case ID	SKPL-ASCaMaDa.UC-35
Use Case Name	Delete Bahan Produksi
Use Case Type	Essential
Priority	High
Actors	Manager Warehouse
Description	Use case ini digunakan untuk menghapus data bahan produksi.
Preconditions	-

Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antar muka untuk menghapus data bahan produksi. 2. Sistem menampilkan data-data bahan produksi. 3. Aktor memilih data bahan produksi yang akan dihapus. 4. Aktor meminta sistem untuk menghapus data bahan produksi dari basis data. 5. Sistem menghapus data bahan produksi dari basis data.
Alternative Paths	-
Postconditions	Data data bahan produksi berhasil dihapus.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.4.4 Spesifikasi Use Case : Display Bahan Produksi

Tabel 3.36 Spesifikasi Use Case : Display Bahan Produksi

Use Case ID	SKPL-ASCaMaDa.UC-36
Use Case Name	Display Bahan Produksi
Use Case Type	Essential
Priority	High
Actors	Manager Warehouse
Description	Use case ini digunakan untuk melihat data bahan produksi.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan data bahan produksi yang tersimpan di database.
Alternative Paths	-
Postconditions	Data bahan produksi ditampilkan.
Exception Paths	-
Extends	-

Includes	Use Case : Login
----------	------------------

3.1.4.5 Spesifikasi Use Case : Entri Stock Keluar

Tabel 3.37 Spesifikasi Use Case : Entri Stock Keluar

Use Case ID	SKPL-ASCaMaDa.UC-37
Use Case Name	Entri Stock Keluar
Use Case Type	Essential
Priority	High
Actors	Manager Warehouse
Description	Use case ini digunakan untuk melakukan entri data bahan baku yang keluar dari gudang untuk keperluan produksi.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka entri stock bahan baku keluar. 2. Aktor memasukkan data barang dan jumlah barang yang keluar. 3. Aktor meminta sistem untuk menyimpan dalam database. 4. Sistem menyimpan data tersebut dalam database.
Alternative Paths	-
Postconditions	Data stock keluar berhasil disimpan dan stock persediaan bahan baku terupdate.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.4.6 Spesifikasi Use Case : Entri Stock Masuk

Tabel 3.38 Spesifikasi Use Case : Entri Stock Masuk

Use Case ID	SKPL-ASCaMaDa.UC-38
-------------	---------------------

Program Studi Teknik Informatika	SKPL-ASCaMaDa	53/ 119
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Use Case Name	Entri Stock Masuk
Use Case Type	Essential
Priority	High
Actors	Manager Warehouse
Description	Use case ini digunakan untuk melakukan entri data barang yang masuk ke gudang dari supplier.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka entri stock bahan baku masuk. 2. Aktor memasukkan data barang dan jumlah barang yang masuk. 3. Aktor meminta sistem untuk menyimpan dalam database. 4. Sistem menyimpan data tersebut dalam database.
Alternative Paths	-
Postconditions	Data stock masuk berhasil disimpan dan stock persediaan bahan baku terupdate.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.4.7 Spesifikasi Use Case : Add Kategori

Tabel 3.39 Spesifikasi Use Case : Add Kategori

Use Case ID	SKPL- ASCaMaDa.UC-39
Use Case Name	Add Kategori
Use Case Type	Essential
Priority	High
Actors	Manager Warehouse
Description	Use case ini digunakan Manager Warehouse untuk menambahkan data kategori bahan baku produksi.

Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk memasukkan data kategori bahan produksi. 2. Aktor memasukkan detail data kategori bahan produksi. 3. Aktor meminta sistem untuk menambahkan data baru ke basisdata. 4. Sistem menyimpan detail data yang telah diisi oleh aktor.
Alternative Paths	-
Postconditions	Data bahan produksi berhasil disimpan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.4.8 Spesifikasi Use Case : Edit Kategori

Tabel 3.40 Spesifikasi Use Case : Edit Kategori

Use Case ID	SKPL-ASCaMaDa.UC-40
Use Case Name	Edit Kategori
Use Case Type	Essential
Priority	High
Actors	Manager Warehouse
Description	Use case ini digunakan Manager Warehouse untuk mengubah data kategori bahan produksi yang telah ada.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk mengubah data kategori bahan produksi. 2. Aktor memasukkan data kategori bahan produksi yang baru. 3. Aktor meminta sistem untuk menyimpan data kategori baru ke basisdata.

	4. Sistem menyimpan perubahan data kategori bahan produksi.
Alternative Paths	-
Postconditions	Data kategori bahan produksi yang baru berhasil disimpan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.4.9 Spesifikasi Use Case : Delete Kategori

Tabel 3.41 Spesifikasi Use Case : Delete Kategori

Use Case ID	SKPL-ASCaMaDa.UC-41
Use Case Name	Delete Kategori Bahan Produksi
Use Case Type	Essential
Priority	High
Actors	Manager Warehouse
Description	Use case ini digunakan untuk menghapus data kategori bahan produksi.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antar muka untuk menghapus data kaegori bahan produksi. 2. Sistem menampilkan data-data kategori bahan produksi. 3. Aktor memilih data kategori bahan produksi yang akan dihapus. 4. Aktor meminta sistem untuk menghapus data kategori bahan produksi dari basis data. 5. Sistem menghapus data kategori bahan produksi dari basis data.
Alternative Paths	-
Postconditions	Data kategori bahan produksi berhasil dihapus.
Exception Paths	-

Extends	-
Includes	Use Case : Login

3.1.4.10 Spesifikasi Use Case : Laporan Bahan Baku Produksi

Tabel 3.42 Spesifikasi Use Case : Laporan Bahan Baku Produksi

Use Case ID	SKPL-ASCaMaDa.UC-42
Use Case Name	Laporan Bahan Baku Produksi
Use Case Type	Essential
Priority	High
Actors	Manager Warehouse
Description	Use case ini digunakan untuk membuat Laporan Bahan Baku Produksi.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk membuat laporan bahan baku produksi. 2. Aktor meminta sistem untuk membuat laporan bahan baku produksi. 3. Sistem menampilkan laporan bahan baku produksi dalam bentuk report.
Alternative Paths	-
Postconditions	Laporan bahan baku produksi berhasil dibuat.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.4.11 Spesifikasi Use Case : Print Laporan Bahan Baku Produksi

Tabel 3.43 Spesifikasi Use Case : Print Laporan Bahan Produksi

Use Case ID	SKPL-ASCaMaDa.UC-43
Use Case Name	Print Laporan Bahan Baku Produksi
Use Case Type	Essential
Priority	High

Actors	Manager Warehouse
Description	Use case ini digunakan untuk mencetak laporan bahan baku produksi pada kertas.
Preconditions	-
Basic Path	1. Aktor meminta sistem untuk mencetak laporan bahan baku produksi. 2. Sistem mencetak laporan bahan baku produksi pada kertas.
Alternative Paths	-
Postconditions	Laporan bahan baku produksi berhasil dicetak pada kertas.
Exception Paths	-
Extends	Use Case : Laporan Bahan Produksi
Includes	-

3.1.4.12 Spesifikasi Use Case : Laporan Bahan Baku Produksi Keluar

Tabel 3.44 Spesifikasi Use Case : Laporan Bahan Baku Produksi Keluar

Use Case ID	SKPL-ASCaMaDa.UC-44
Use Case Name	Laporan Bahan Baku Produksi Keluar.
Use Case Type	Essential
Priority	High
Actors	Manager Warehouse
Description	Use case ini digunakan untuk membuat Laporan Bahan Baku Produksi Keluar.
Preconditions	-
Basic Path	1. Sistem menampilkan antarmuka untuk membuat laporan bahan baku produksi keluar. 2. Aktor meminta sistem untuk membuat laporan bahan baku produksi keluar. 3. Sistem menampilkan laporan bahan baku produksi keluar dalam bentuk report.
Alternative Paths	-

Postconditions	Laporan laporan bahan baku produksi keluar berhasil dibuat.
Exception Paths	-
Extends	-
Includes	Use Case : Login

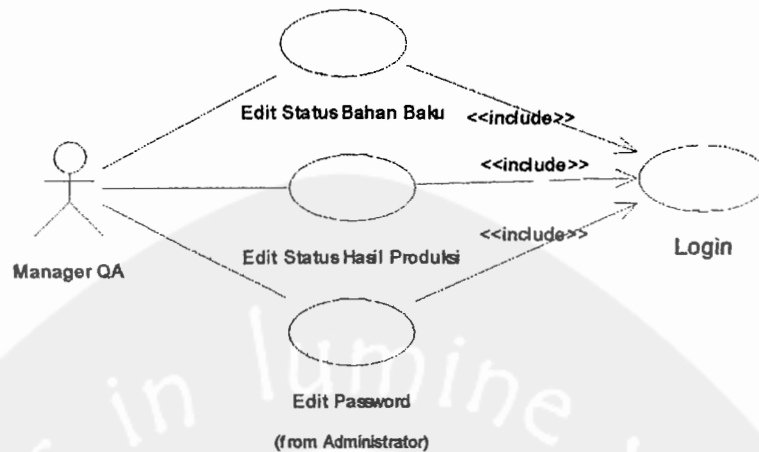
3.1.4.13 Spesifikasi Use Case : Print Laporan Bahan Baku Produksi Keluar

Tabel 3.45 Spesifikasi Use Case : Print Laporan Bahan Baku Produksi Keluar

Use Case ID	SKPL-ASCaMaDa.UC-45
Use Case Name	Print Laporan Bahan Baku Produksi
Use Case Type	Essential
Priority	High
Actors	Manager Warehouse
Description	Use case ini digunakan untuk mencetak laporan bahan baku produksi keluar pada kertas.
Preconditions	-
Basic Path	1. Aktor meminta sistem untuk mencetak laporan bahan baku produksi. 2. Sistem mencetak laporan bahan baku produksi pada kertas.
Alternative Paths	-
Postconditions	Laporan bahan baku produksi berhasil dicetak pada kertas.
Exception Paths	-
Extends	Use Case : Laporan Bahan Baku Produksi
Includes	-

3.1.5 Use Case Diagram Quality Assurance

Secara umum fungsi-fungsi yang dimiliki oleh Quality Assurance adalah:



Gambar 3.5 Use Case Diagram Quality Assurance

3.1.5.1 Spesifikasi Use Case : Edit Status Bahan Baku

Tabel 3.47 Spesifikasi Use Case : Edit Status Bahan Baku

Use Case ID	SKPL-ASCaMaDa.UC-46
Use Case Name	Edit Status Bahan Baku
Use Case Type	Essential
Priority	High
Actors	Manager QA
Description	Use case ini digunakan Manager QA untuk mengubah status bahan baku produksi dari <i>Quarantine</i> menjadi <i>released</i> atau <i>rejected</i> .
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk mengubah status bahan baku produksi. 2. Aktor memilih data bahan baku produksi yang akan diberi status baru. 3. Aktor memasukkan data status bahan baku produksi yang baru. 4. Aktor meminta sistem untuk menyimpan data status bahan baku produksi ke basisdata.

	5. Sistem menyimpan perubahan data status bahan baku produksi.
Alternative Paths	-
Postconditions	Data status bahan baku produksi yang baru berhasil disimpan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.5.2 Spesifikasi Use Case : Edit Status Barang Produksi

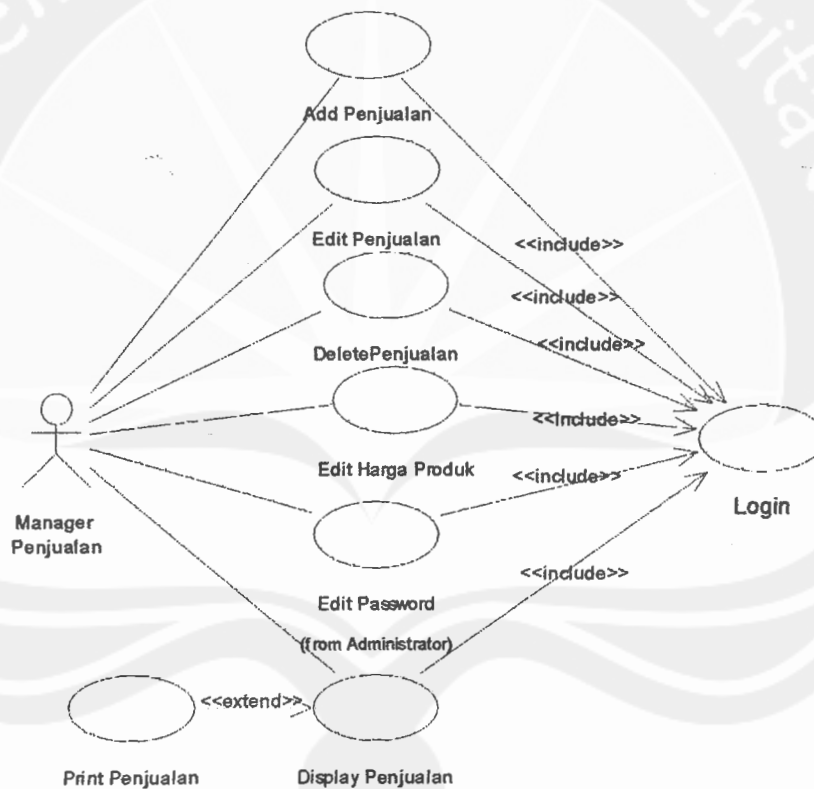
Tabel 3.48 Spesifikasi Use Case : Edit Status Barang Produksi

Use Case ID	SKPL-ASCaMaDa.UC-47
Use Case Name	Edit Status Barang Produksi
Use Case Type	Essential
Priority	High
Actors	Manager QA
Description	Use case ini digunakan Manager QA untuk mengubah status barang produksi dari <i>Quarantine</i> menjadi <i>released</i> atau <i>rejected</i> .
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk mengubah status barang produksi. 2. Aktor memilih data status barang produksi yang akan diberi status baru. 3. Aktor memasukkan data status barang produksi yang baru. 4. Aktor meminta sistem untuk menyimpan data status barang produksi ke basisdata. 5. Sistem menyimpan perubahan data status barang produksi.
Alternative Paths	-
Postconditions	Data status barang produksi yang baru berhasil

	disimpan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.6 Use Case Diagram Marketing

Secara umum fungsi-fungsi yang dimiliki oleh Marketing adalah:



Gambar 3.6 Use Case Diagram Marketing

3.1.6.1 Spesifikasi Use Case : Add Penjualan

Tabel 3.48 Spesifikasi Use Case : Add Penjualan

Program Studi Teknik Informatika	SKPL-ASCaMaDa	62/ 119
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Use Case ID	SKPL- ASCaMaDa.UC-48
Use Case Name	Add penjualan
Use Case Type	Essential
Priority	High
Actors	Manager Penjualan
Description	Use case ini digunakan Manager Penjualan untuk menambahkan data penjualan barang hasil produksi ke distributor.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk memasukkan data penjualan. 2. Aktor memasukkan detail data penjualan. 3. Aktor meminta sistem untuk menyimpan detail data yang telah diisi oleh aktor. 4. Sistem menyimpan detail data yang telah diisi oleh aktor.
Alternative Paths	-
Postconditions	Data penjualan berhasil disimpan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.6.2 Spesifikasi Use Case : Edit Penjualan

Tabel 3.49 Spesifikasi Use Case : Edit Penjualan

Use Case ID	SKPL-ASCaMaDa.UC-49
Use Case Name	Edit Penjualan
Use Case Type	Essential
Priority	High
Actors	Manager Penjualan
Description	Use case ini digunakan Manager Penjualan untuk mengubah data penjualan yang telah ada.

Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk mengubah data penjualan. 2. Aktor memasukkan data penjualan yang baru. 3. Aktor meminta sistem untuk menyimpan data baru ke basisdata. 4. Sistem menyimpan perubahan data penjualan.
Alternative Paths	-
Postconditions	Data penjualan yang baru berhasil disimpan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.6.3 Spesifikasi Use Case : Edit Harga Produk

Tabel 3.50 Spesifikasi Use Case : Edit Harga Produk

Use Case ID	SKPL-ASCaMaDa.UC-50
Use Case Name	Edit Harga Produk
Use Case Type	Essential
Priority	High
Actors	Manager Penjualan
Description	Use case ini digunakan Manager Penjualan untuk mengubah data harga produk yang telah ada.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk mengubah data harga produk. 2. Aktor memasukkan data harga yang baru. 3. Aktor meminta sistem untuk menyimpan data baru ke basisdata. 4. Sistem menyimpan perubahan data harga produk.
Alternative Paths	-
Postconditions	Data harga produk yang baru berhasil disimpan.

Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.6.4 Spesifikasi Use Case : Delete Penjualan

Tabel 3.51 Spesifikasi Use Case : Delete Penjualan

Use Case ID	SKPL-ASCaMaDa.UC-51
Use Case Name	Delete Penjualan
Use Case Type	Essential
Priority	High
Actors	Manager Penjualan
Description	Use case ini digunakan untuk menghapus data penjualan.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antar muka untuk menghapus data penjualan. 2. Sistem menampilkan data-data penjualan. 3. Aktor memilih data penjualan yang akan dihapus. 4. Sistem menghapus data penjualan dari basis data.
Alternative Paths	-
Postconditions	Data penjualan berhasil dihapus.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.6.5 Spesifikasi Use Case : Display Penjualan

Tabel 3.52 Spesifikasi Use Case : Display Penjualan

Use Case ID	SKPL-ASCaMaDa.UC-52
Use Case Name	Display Penjualan

Use Case Type	Essential
Priority	High
Actors	Manager Penjualan
Description	Use case ini digunakan untuk melihat data penjualan.
Preconditions	-
Basic Path	1. Sistem menampilkan data penjualan yang tersimpan di database.
Alternative Paths	-
Postconditions	Data penjualan ditampilkan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.6.6 Spesifikasi Use Case : Print Penjualan

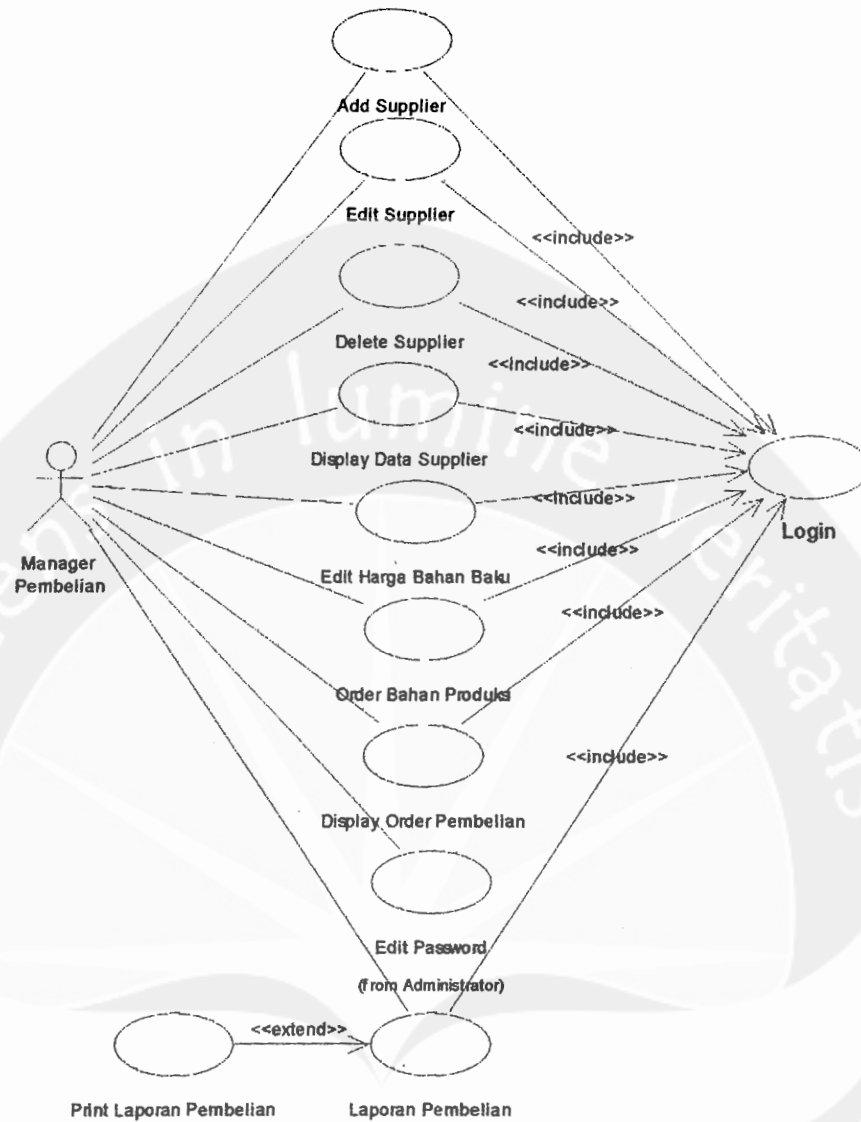
Tabel 3.53 Spesifikasi Use Case : Print Penjualan

Use Case ID	SKPL-ASCaMaDa.UC-53
Use Case Name	Print Penjualan
Use Case Type	Essential
Priority	High
Actors	Manager Warehouse
Description	Use case ini digunakan untuk mencetak penjualan.
Preconditions	-
Basic Path	1. Aktor memilih tanggal penjualan. 2. Aktor meminta sistem untuk mencetak data penjualan. 3. Sistem mencetak data penjualan ke kertas.
Alternative Paths	-
Postconditions	Data penjualan berhasil dicetak pada kertas.
Exception Paths	-
Extends	Use Case : Display Penjualan
Includes	-

3.1.7 Use Case Diagram Purchasing

Secara umum fungsi-fungsi yang dimiliki oleh Purchasing adalah:





Gambar 3.7 Use Case Diagram Purchasing

3.1.7.1 Spesifikasi Use Case : Add Supplier

Tabel 3.54 Spesifikasi Use Case : Add Supplier

Use Case ID	SKPL- ASCaMaDa.UC-54
Use Case Name	Add Supplier
Use Case Type	Essential
Priority	High
Actors	Manager Pembelian

Description	Use case ini digunakan Manager Pembelian untuk menambahkan data supplier bahan baku produksi.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk memasukkan data supplier. 2. Aktor memasukkan data detail supplier. 3. Sistem memeriksa data-data yang dimasukkan oleh aktor. 4. Aktor meminta sistem untuk menyimpan detail data yang dimasukkan oleh aktor. 5. Sistem menyimpan detail data yang dimasukkan oleh aktor.
Alternative Paths	-
Postconditions	Data supplier berhasil disimpan.
Exception Paths	<ol style="list-style-type: none"> 1. Detail data yang diisikan belum lengkap (setelah Tabel 3.61 langkah 3) <ol style="list-style-type: none"> a. Sistem menampilkan peringatan bahwa data yang diisikan belum lengkap. b. Kembali ke Tabel 3.61 langkah 2.
Extends	-
Includes	Use Case : Login

3.1.7.2 Spesifikasi Use Case : Edit Supplier

Tabel 3.55 Spesifikasi Use Case : Edit Supplier

Use Case ID	SKPL-ASCaMaDa.UC-55
Use Case Name	Edit Supplier
Use Case Type	Essential
Priority	High
Actors	Manager Pembelian.
Description	Use case ini digunakan Manager Pembelian untuk mengubah data supplier yang telah ada.
Preconditions	-
Basic Path	1. Sistem menampilkan antarmuka untuk mengubah

	<p>data supplier.</p> <ol style="list-style-type: none"> Aktor memasukkan data supplier yang baru. Aktor meminta sistem untuk menyimpan data baru ke basisdata. Sistem menyimpan perubahan data supplier.
Alternative Paths	-
Postconditions	Data supplier yang baru berhasil disimpan.
Exception Paths	
Extends	-
Includes	Use Case : Login

3.1.7.3 Spesifikasi Use Case : Delete Supplier

Tabel 3.56 Spesifikasi Use Case : Delete Supplier

Use Case ID	SKPL-ASCaMaDa.UC-56
Use Case Name	Delete Supplier
Use Case Type	Essential
Priority	High
Actors	Manager Pembelian
Description	Use case ini digunakan untuk menghapus data supplier.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> Sistem menampilkan antar muka untuk menghapus data supplier. Sistem menampilkan data-data supplier. Aktor memilih data supplier yang akan dihapus. Sistem menghapus data supplier dari basis data.
Alternative Paths	-
Postconditions	Data supplier berhasil dihapus.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.7.4 Spesifikasi Use Case : Display Data Supplier

Tabel 3.57 Spesifikasi Use Case : Display DataSupplier

Use Case ID	SKPL-ASCaMaDa.UC-57
Use Case Name	Display Data Supplier
Use Case Type	Essential
Priority	High
Actors	Manager Pembelian
Description	Use case ini digunakan untuk melihat data supplier.
Preconditions	-
Basic Path	1. Sistem menampilkan data supplier yang tersimpan di database.
Alternative Paths	-
Postconditions	Data supplier ditampilkan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.7.5 Spesifikasi Use Case : Edit Harga Bahan Baku

Tabel 3.58 Spesifikasi Use Case : Edit Harga Bahan Baku

Use Case ID	SKPL- ASCaMaDa.UC-58
Use Case Name	Edit Harga Bahan Baku
Use Case Type	Essential
Priority	High
Actors	Manager Pembelian
Description	Use case ini digunakan Manager Pembelian untuk melakukan edit harga bahan baku.
Preconditions	-
Basic Path	1. Sistem menampilkan antarmuka untuk melakukan edit harga bahan baku. 2. Aktor memasukkan data harga bahan baku yang baru.

	<p>3. Aktor meminta sistem untuk menyimpan data harga bahan baku</p> <p>4. Sistem menyimpan data bahan baku yang dimasukkan oleh aktor.</p>
Alternative Paths	-
Postconditions	Data harga bahan baku ter-update.
Exception Paths	-
Extends	-.
Includes	Use Case : Login

3.1.7.6 Spesifikasi Use Case : Order Bahan Baku

Tabel 3.59 Spesifikasi Use Case : Order Bahan Baku

Use Case ID	SKPL- ASCaMaDa.UC-59
Use Case Name	Order Bahan Baku
Use Case Type	Essential
Priority	High
Actors	Manager Pembelian
Description	Use case ini digunakan Manager Pembelian untuk melakukan order pembelian bahan baku produksi.
Preconditions	-
Basic Path	<p>1. Sistem menampilkan antarmuka untuk melakukan order pembelian.</p> <p>2. Aktor memasukkan data pembelian bahan baku produksi.</p> <p>3. Aktor meminta sistem untuk menyimpan data order pembelian.</p> <p>4. Sistem menyimpan data order pembelian yang dimasukkan oleh aktor.</p>
Alternative Paths	-
Postconditions	Data order pembelian berhasil disimpan.
Exception Paths	-
Extends	-

Includes	Use Case : Login
----------	------------------

3.1.7.7 Spesifikasi Use Case : Display Order Pembelian

Tabel 3.60 Spesifikasi Use Case : Display Order Pembelian

Use Case ID	SKPL-ASCaMaDa.UC-60
Use Case Name	Display Order Pembelian
Use Case Type	Essential
Priority	High
Actors	Manager Pembelian
Description	Use case ini digunakan untuk menampilkan Order Pembelian.
Preconditions	-
Basic Path	1. Sistem menampilkan data order pembelian yang tersimpan di database.
Alternative Paths	-
Postconditions	Data order pembelian ditampilkan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.1.7.8 Spesifikasi Use Case : Laporan Pembelian

Tabel 3.61 Spesifikasi Use Case : Laporan Pembelian

Use Case ID	SKPL-ASCaMaDa.UC-61
Use Case Name	Laporan Pembelian
Use Case Type	Essential
Priority	High
Actors	Manager Pembelian
Description	Use case ini digunakan untuk membuat Laporan Pembelian.
Preconditions	-
Basic Path	1. Sistem menampilkan antarmuka untuk membuat

	<p>laporan pembelian.</p> <ol style="list-style-type: none"> 2. Aktor memasukkan data tanggal pembelian. 3. Aktor meminta sistem untuk membuat laporan pembelian. 4. Sistem menampilkan laporan pembelian dalam bentuk report.
Alternative Paths	-
Postconditions	Laporan Pembelian berhasil dibuat.
Exception Paths	-
Extends	-
Includes	Use Case : Login

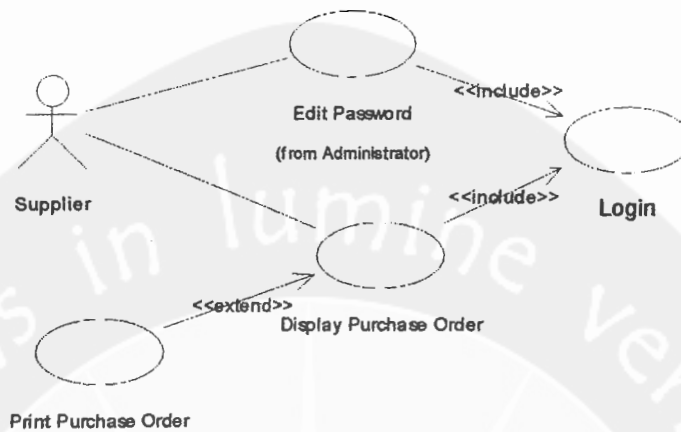
3.1.7.9 Spesifikasi Use Case : Print Laporan Pembelian

Tabel 3.62 Spesifikasi Use Case : Print Laporan Pemeriksaan

Use Case ID	SKPL-ASCaMaDa.UC-62
Use Case Name	Print Laporan Pembelian
Use Case Type	Essential
Priority	High
Actors	Manager Pembelian
Description	Use case ini digunakan untuk mencetak laporan pembelian bahan baku produksi kertas.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Aktor meminta sistem untuk mencetak laporan pembelian bahan baku produksi pada kertas. 2. Sistem mencetak laporan pembelian bahan baku produksi pada kertas.
Alternative Paths	-
Postconditions	Laporan pembelian bahan baku produksi tercetak di kertas.
Exception Paths	-
Extends	Use Case : Laporan Pembelian.
Includes	-

3.1.8 Use Case Diagram Supplier

Secara umum fungsi-fungsi yang dimiliki oleh Supplier adalah:



Gambar 3.8 Use Case Diagram Supplier

3.1.8.1 Spesifikasi Use Case : Display Purchase Order

Tabel 3.63 Spesifikasi Use Case : Display Purchase Order

Use Case ID	SKPL-ASCaMaDa.UC-63
Use Case Name	Display Purchase Order
Use Case Type	Essential
Priority	High
Actors	Supplier
Description	Use case ini digunakan untuk menampilkan Purchase Order Bahan Baku Produksi.
Preconditions	-
Basic Path	1. Sistem menampilkan data Order Bahan Baku Produksi yang tersimpan di database.
Alternative Paths	-
Postconditions	Data Purchase Order ditampilkan.
Exception Paths	-
Extends	

Includes	- Use Case : Login
----------	--------------------

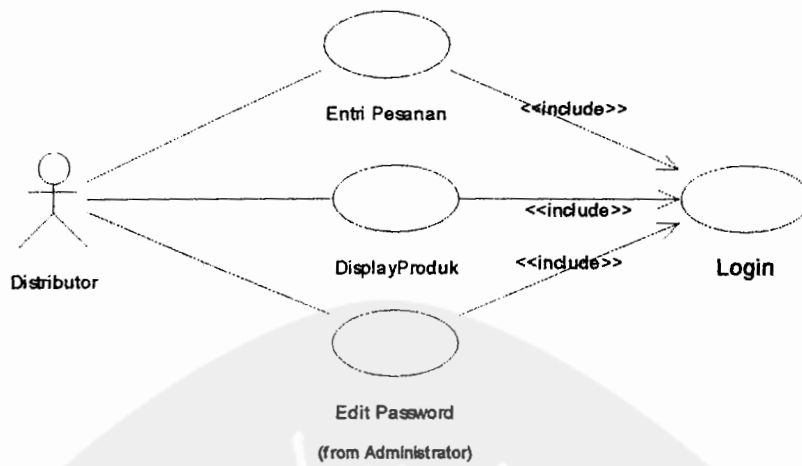
3.1.8.2 Spesifikasi Use Case : Print Purchase Order

Tabel 3.64 Spesifikasi Use Case : Print Purchase Order

Use Case ID	SKPL-ASCaMaDa.UC-64
Use Case Name	Print Purchase Order
Use Case Type	Essential
Priority	High
Actors	Supplier
Description	Use case ini digunakan untuk mencetak Order Bahan Baku Produksi pada kertas.
Preconditions	-
Basic Path	1. Aktor meminta sistem untuk mencetak Order Bahan Baku Produksi pada kertas. 2. Sistem mencetak Order Bahan Baku Produksi pada kertas.
Alternative Paths	-
Postconditions	Order Bahan Baku Produksi tercetak pada kertas.
Exception Paths	-
Extends	Use Case : Display Purchase Order
Includes	-

3.1.9 Use Case Diagram Distributor

Secara umum fungsi-fungsi yang dimiliki oleh Distributor adalah:



Gambar 3.9 Use Case Diagram Distributor

3.1.9.1 Spesifikasi Use Case : Entri Pesanan

Tabel 3.65 Spesifikasi Use Case : Entri Pesanan

Use Case ID	SKPL- ASCaMaDa.UC-65
Use Case Name	Entri pesanan
Use Case Type	Essential
Priority	High
Actors	Distributor
Description	Use case ini digunakan distributor untuk menambahkan data pesanan barang ke pabrik.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka untuk memasukkan data pemesanan. 2. Aktor memasukkan detail data pemesanan. 3. Aktor meminta sistem untuk menyimpan detail data yang telah diisi oleh aktor. 4. Sistem menyimpan detail data yang telah diisi oleh aktor.
Alternative Paths	-
Postconditions	Data pemesanan berhasil disimpan.
Exception Paths	-
Extends	-

Includes	Use Case : Login
----------	------------------

3.1.9.2 Spesifikasi Use Case : Display Produk

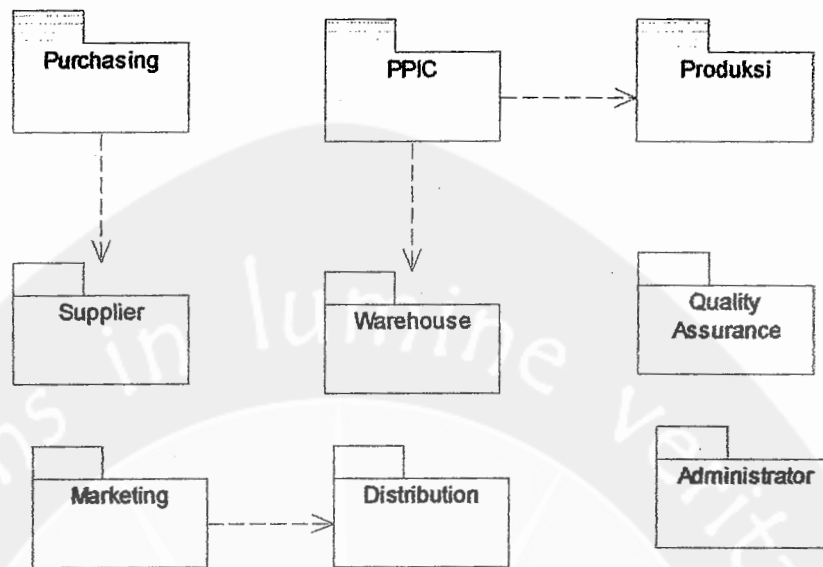
Tabel 3.66 Spesifikasi Use Case : Display Produk

Use Case ID	SKPL-ASCaMaDa.UC-66
Use Case Name	Display Produk
Use Case Type	Essential
Priority	High
Actors	Distributor
Description	Use case ini digunakan untuk melihat data produk yang dijual.
Preconditions	-
Basic Path	1. Sistem menampilkan data produk yang tersimpan di database.
Alternative Paths	-
Postconditions	Data produk ditampilkan.
Exception Paths	-
Extends	-
Includes	Use Case : Login

3.2 Realisasi Use Case

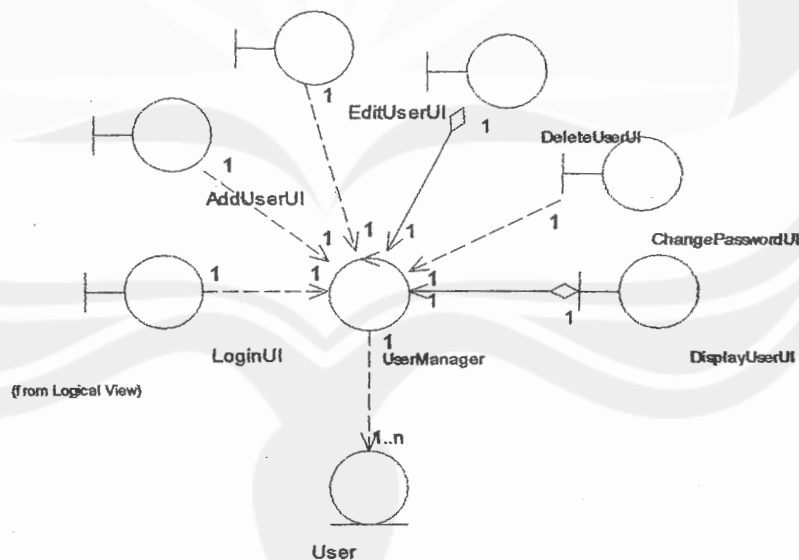
3.2.1 Static Structure Diagram

3.2.1.1 Analysis Class Diagram : Package Dependencies



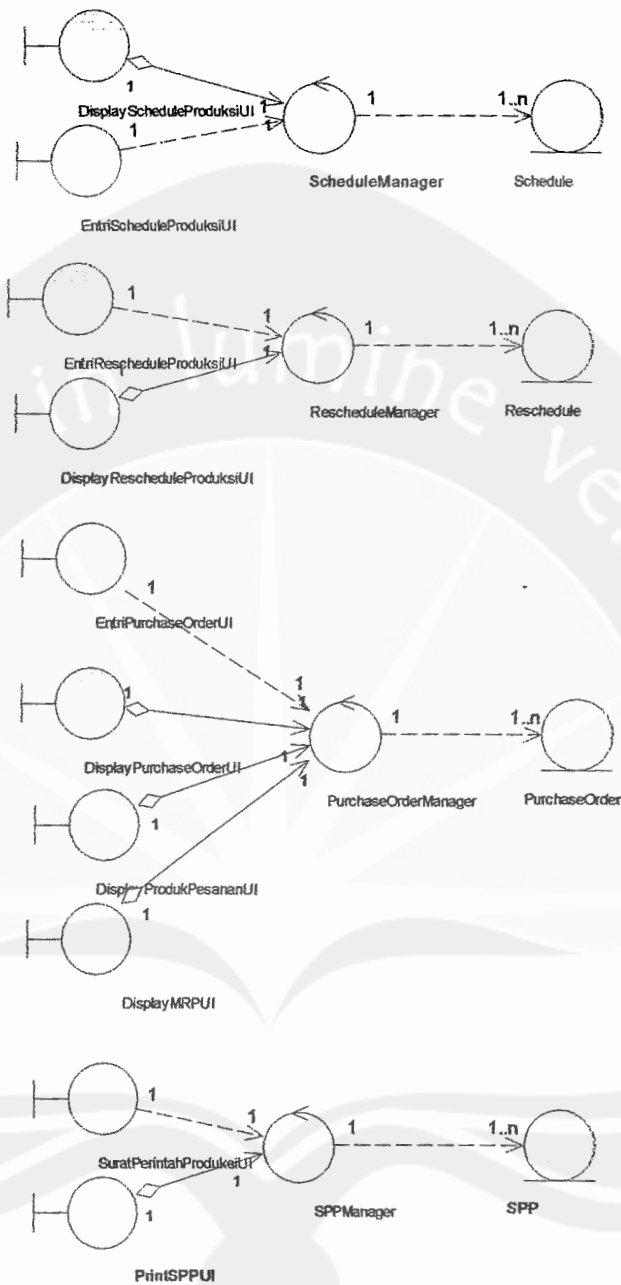
Gambar 3.10 Analysis Class Diagram : Package Dependencies

3.2.1.2 Analysis Class Diagram : Package Administrator



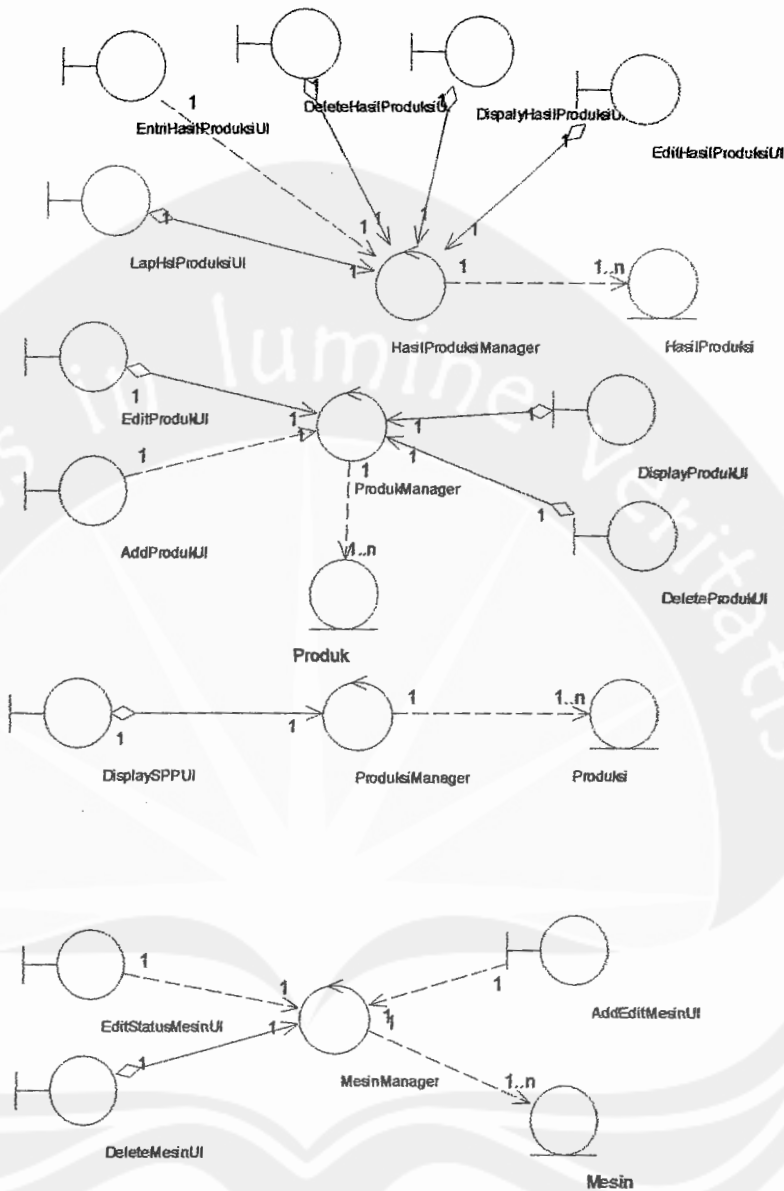
Gambar 3.11 Analysis Class Diagram : Package Data Administrator

3.2.1.3 Analysis Class Diagram : Package PPIC



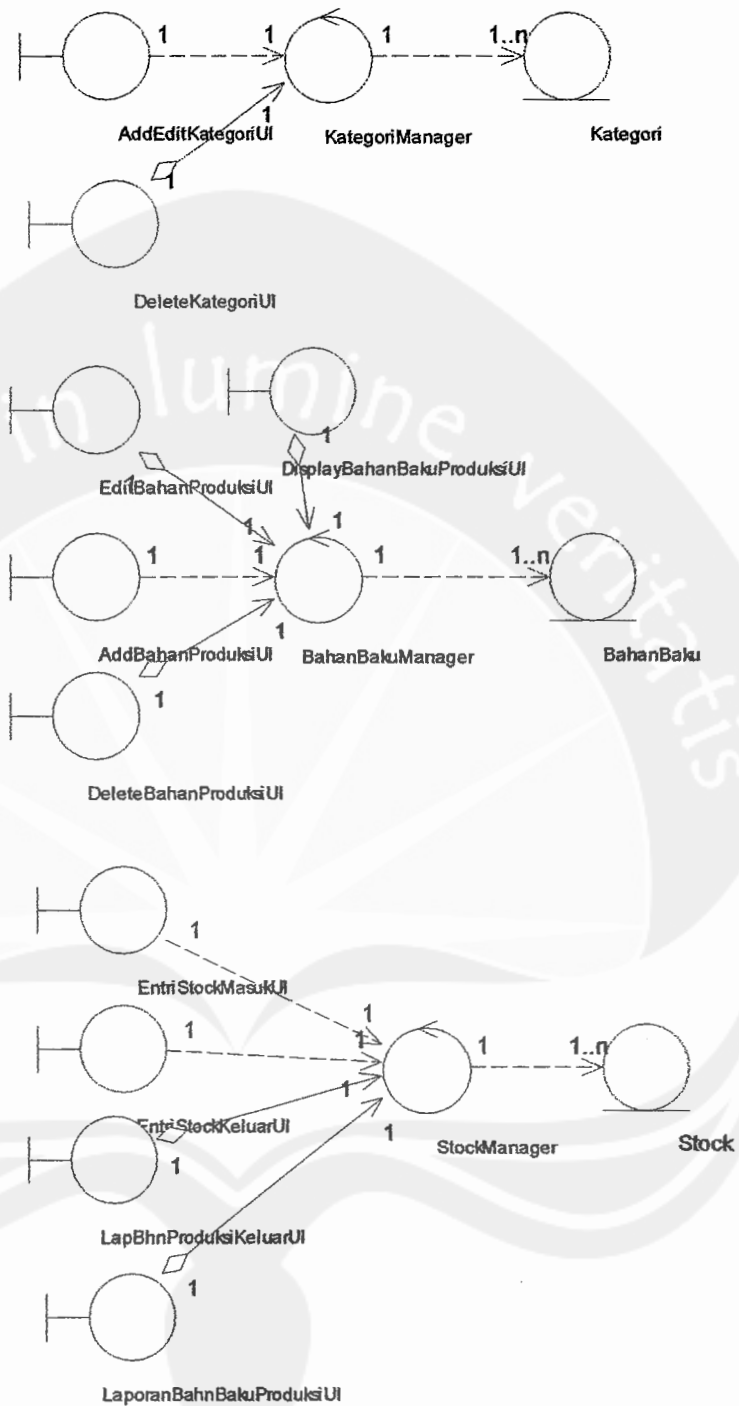
Gambar 3.12 Analysis Class Diagram : Package PPIC

3.2.1.4 Analysis Class Diagram : Package Produksi



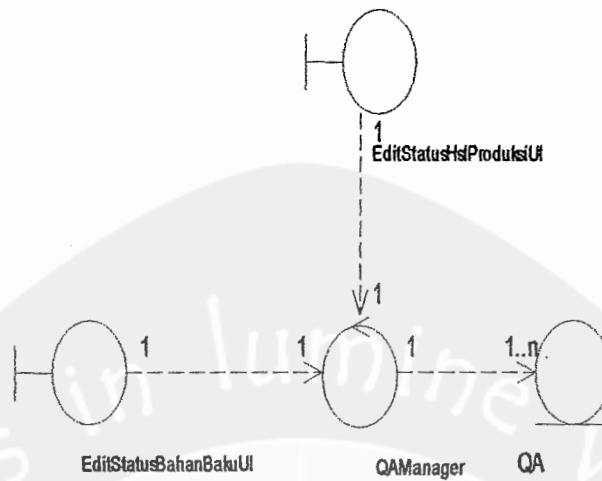
Gambar 3.13 Analysis Class Diagram : Package Produksi

3.2.1.5 Analysis Class Diagram : Package Warehouse



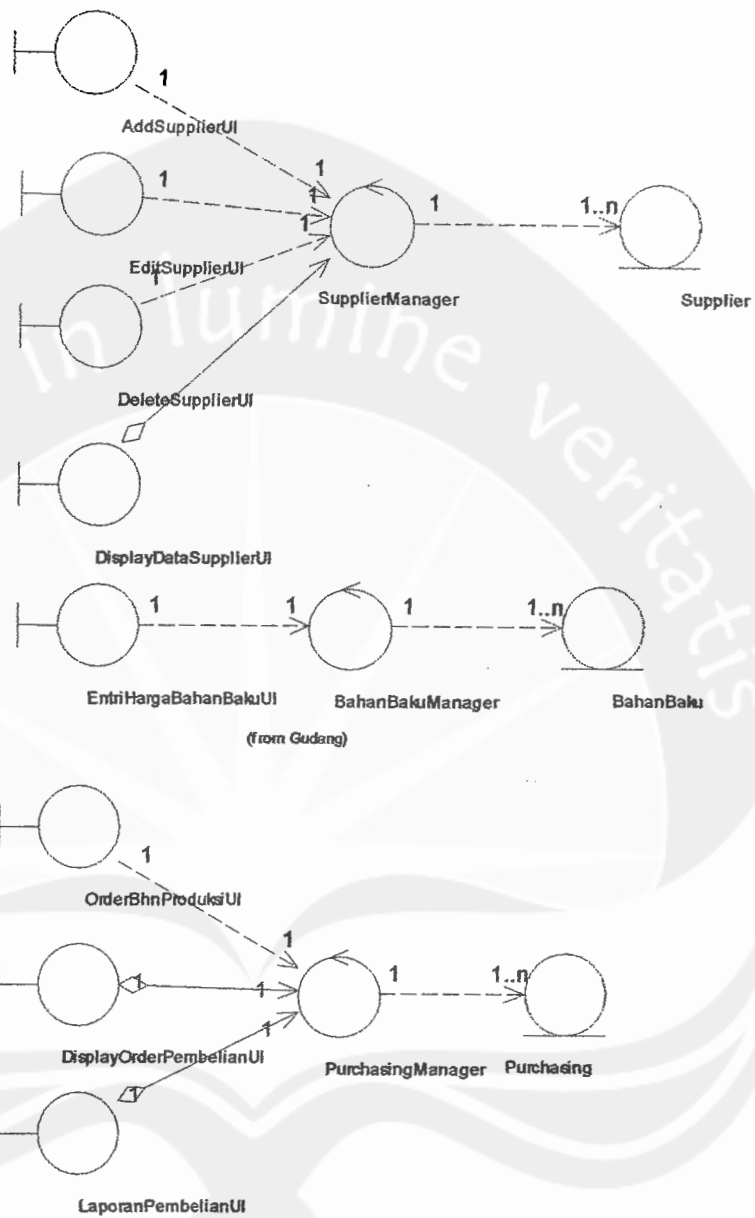
Gambar 3.14 Analysis Class Diagram : Package Warehouse

3.2.1.6 Analysis Class Diagram : Package QualityAssurance



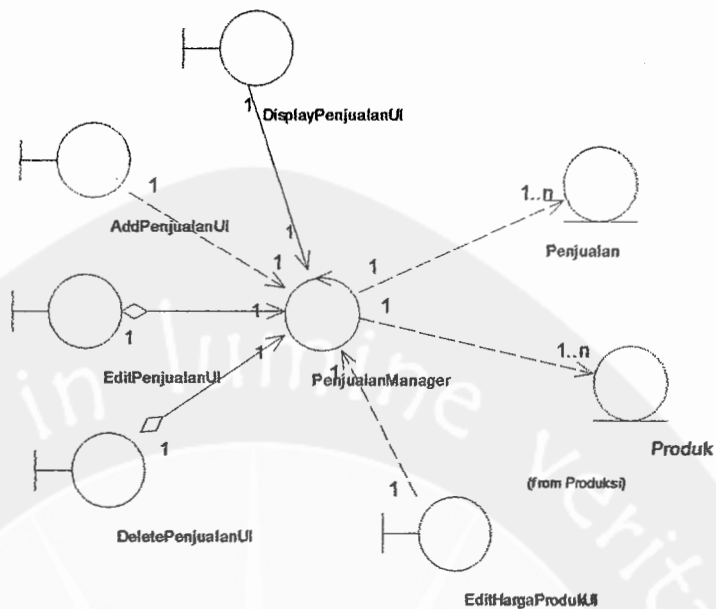
Gambar 3.15 Analysis Class Diagram : Package QualityAssurance

3.2.1.7 Analysis Class Diagram : Package Purchasing



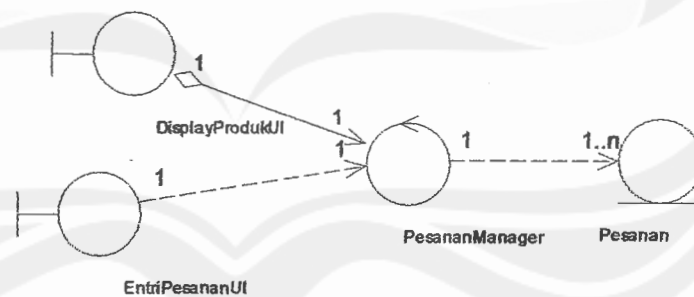
Gambar 3.16 Analysis Class Diagram : Package Purchasing

3.2.1.8 Analysis Class Diagram : Package Marketing



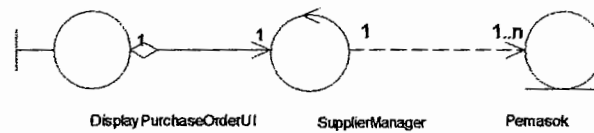
Gambar 3.17 Analysis Class Diagram : Package Marketing

3.2.1.9 Analysis Class Diagram : Package Distributor



Gambar 3.18 Analysis Class Diagram : Package Distributor

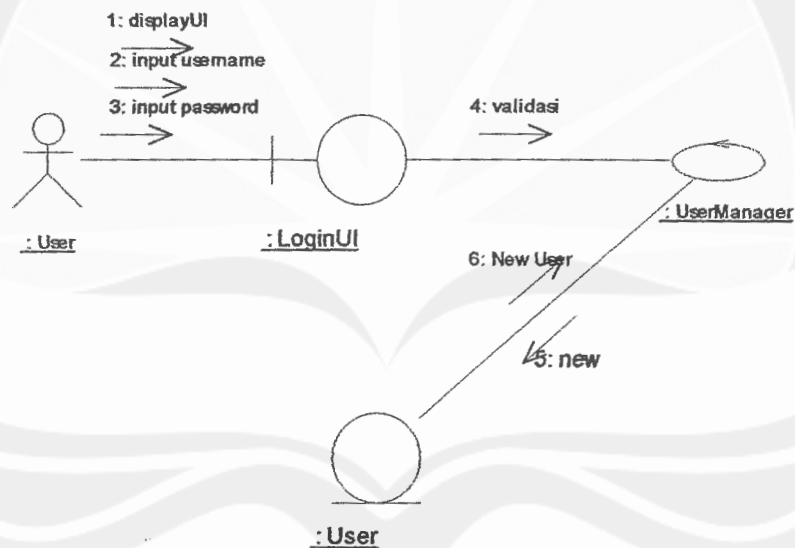
3.2.1.10 Analysis Class Diagram : Package Supplier



Gambar 3.19 Analysis Class Diagram : Package Supplier

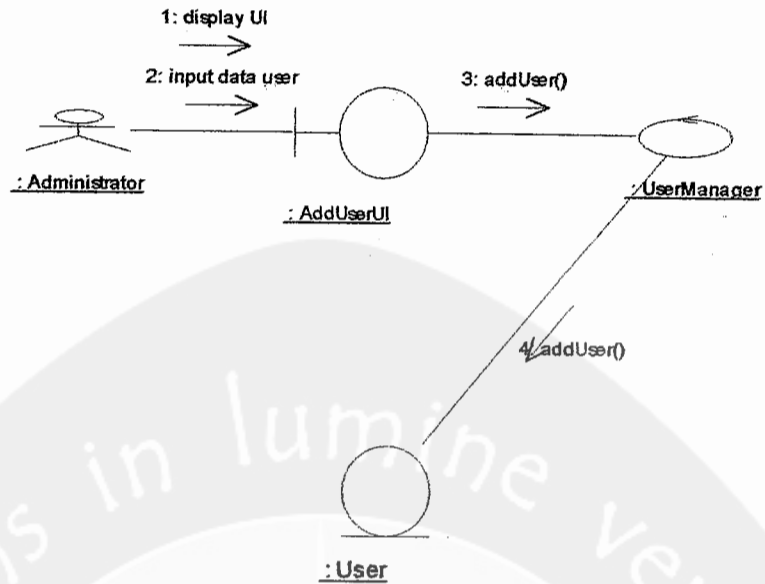
3.2.2 Interaction Diagram

3.2.2.1 Analysis Collaboration Diagram : Use Case Login



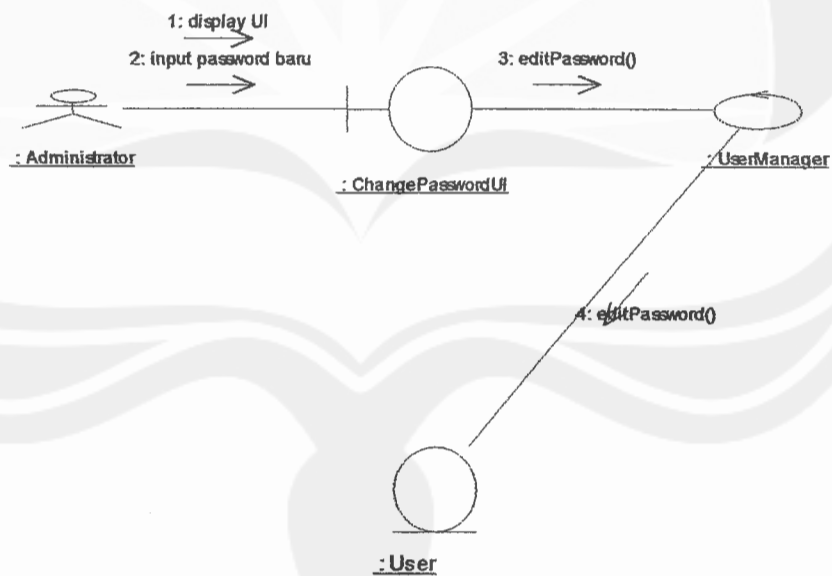
Gambar 3.20 Analysis Collaboration Diagram : Use Case Login

3.2.2.2 Analysis Collaboration Diagram : Use Case Add User



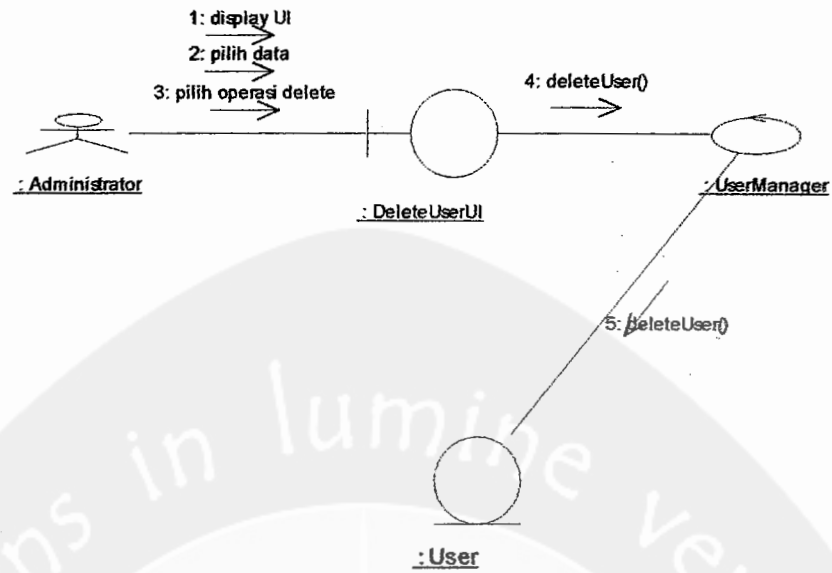
Gambar 3.21 Analysis Collaboration Diagram : Use Case Add User

3.2.2.3 Analysis Collaboration Diagram : Use Case Change Password



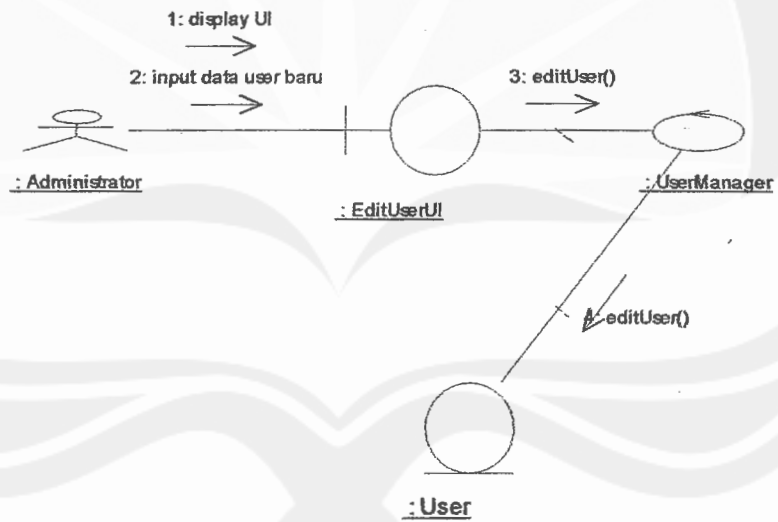
Gambar 3.22 Analysis Collaboration Diagram : Use Case Change Password

3.2.2.4 Analysis Collaboration Diagram : Use Case Delete User



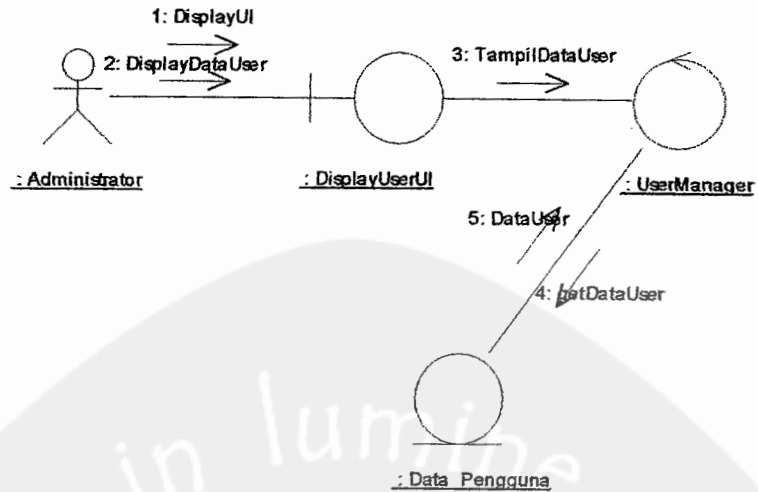
Gambar 3.23 Analisis Collaboration Diagram : Use Case Delete User

3.2.2.5 Analisis Collaboration Diagram : Use Case Edit User



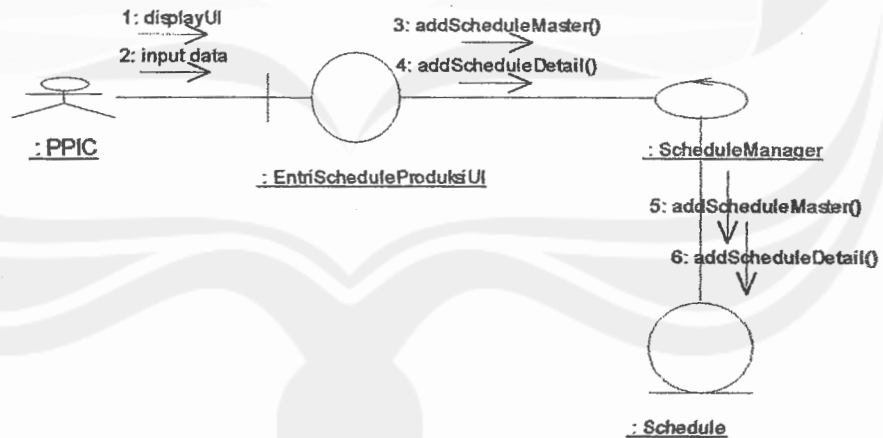
Gambar 3.24 Analisis Collaboration Diagram : Use Case Edit User

3.2.2.6 Analysis Collaboration Diagram : Use Case Display User



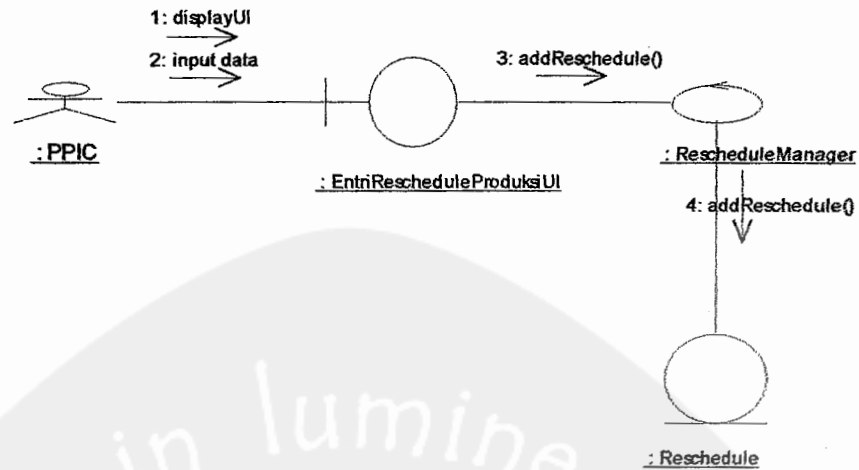
Gambar 3.25 Analysis Collaboration Diagram : Use Case Display User

3.2.2.7 Analysis Collaboration Diagram : Use Case Entri Schedule Produksi



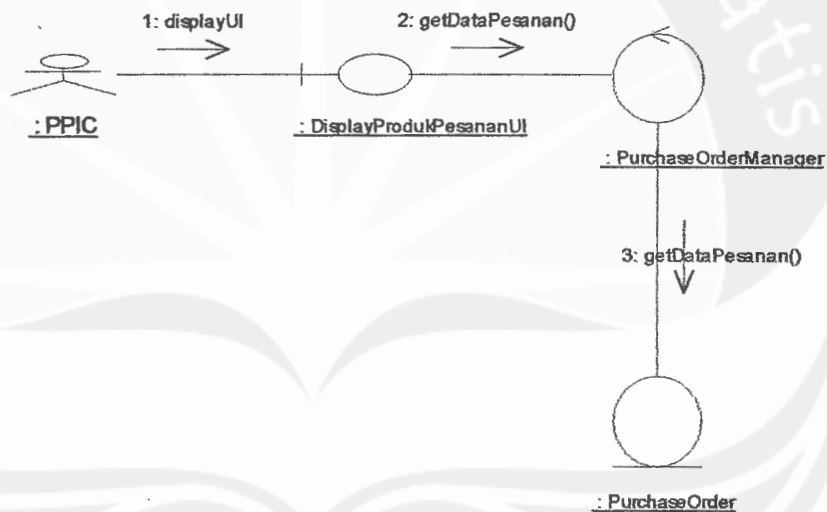
Gambar 3.26 Analysis Collaboration Diagram : Use Case Create Entri Schedule Produksi

3.2.2.8 Analysis Collaboration Diagram : Use Case Entri Reschedule Produksi



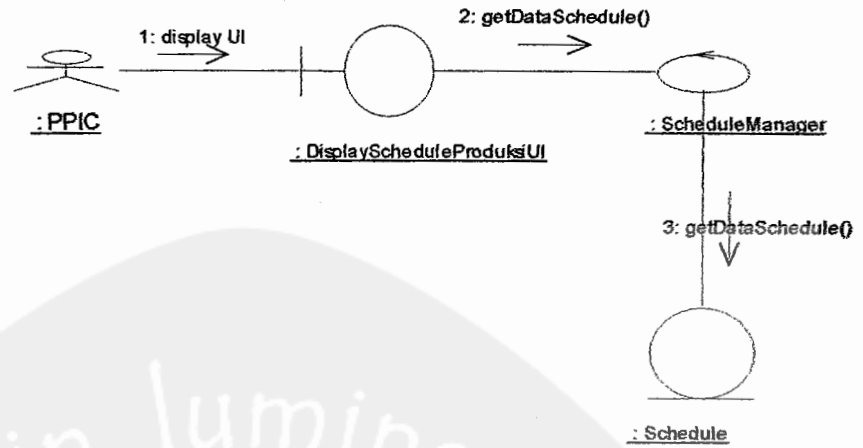
Gambar 3.27 Analysis Collaboration Diagram : Use Case Entri Reschedule Produksi

3.2.2.9 Analysis Collaboration Diagram : Use Case Display Barang Pesanan



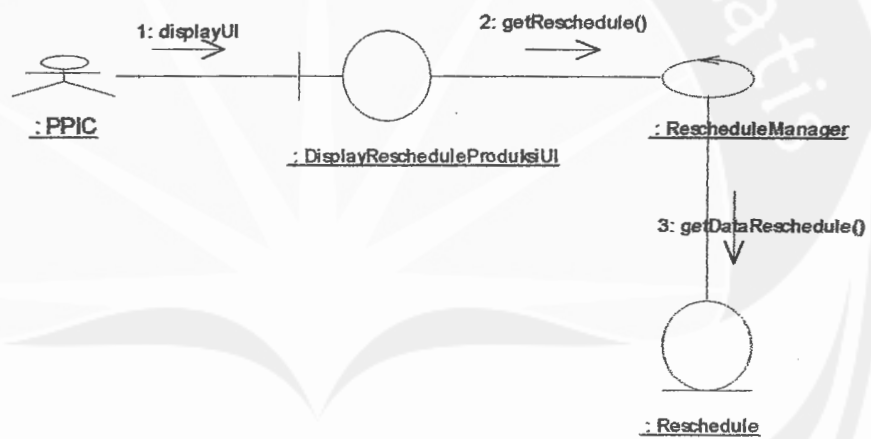
Gambar 3.28 Analysis Collaboration Diagram : Use Case Display Barang Pesanan

3.2.2.10 Analysis Collaboration Diagram : Use Case Display Schedule



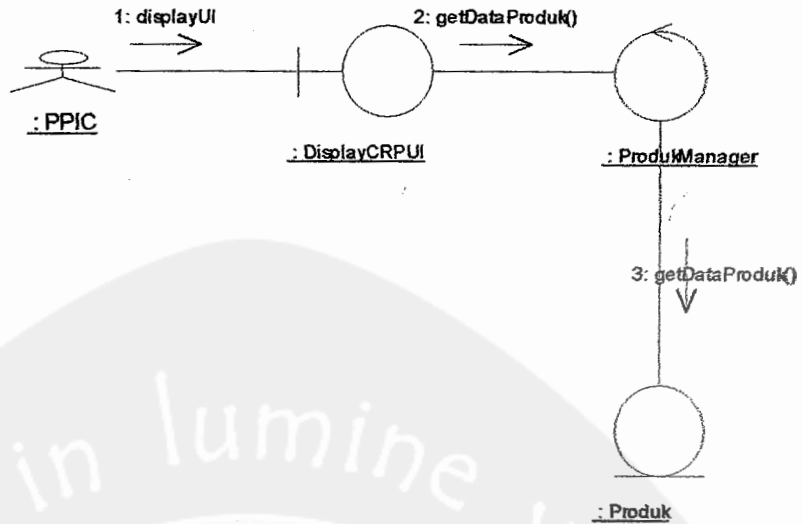
Gambar 3.29 Analysis Collaboration Diagram : Use Case Display Schedule

3.2.2.11 Analysis Collaboration Diagram : Use Case Display Reschedule



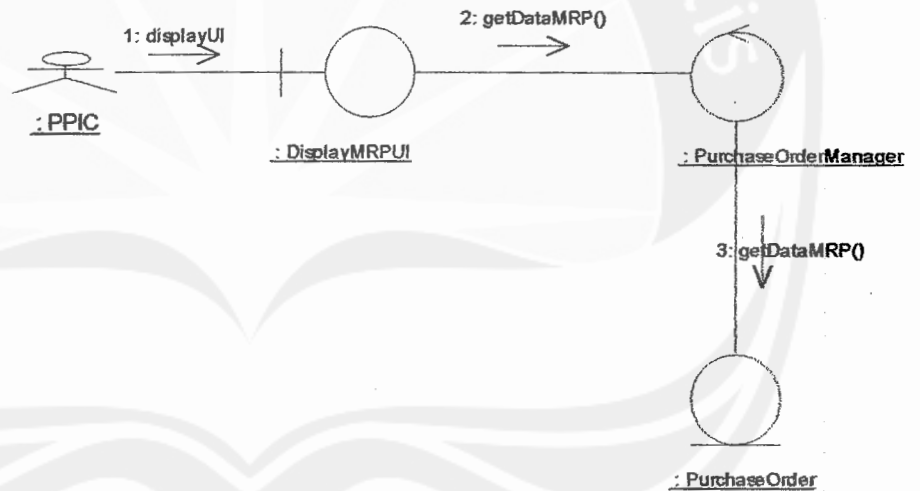
Gambar 3.30 Analysis Collaboration Diagram : Use Case Display Reschedule

3.2.2.12 Analysis Collaboration Diagram : Use Case Display CRP



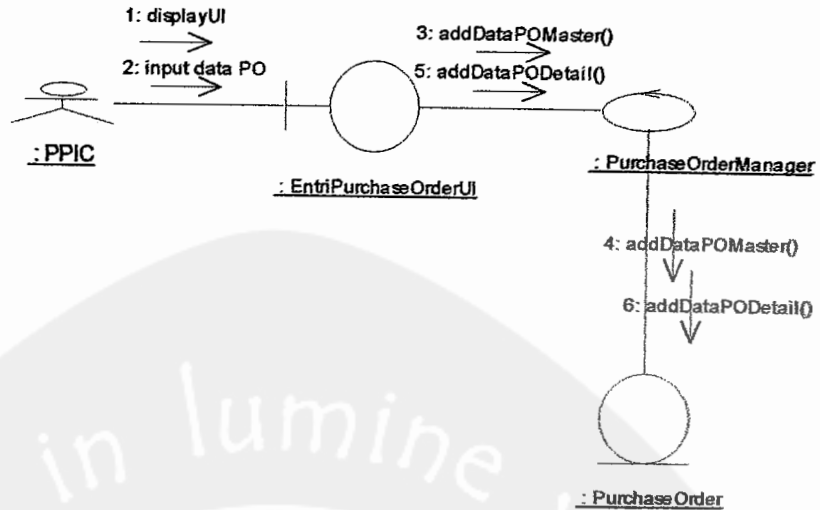
Gambar 3.31 Analysis Collaboration Diagram : Use Case Display CRP

3.2.2.13 Analysis Collaboration Diagram : Use Case Display MRP



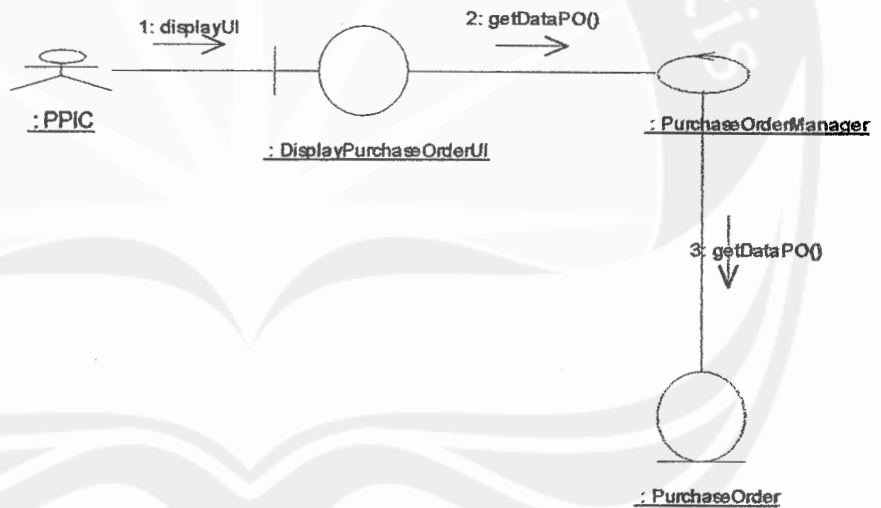
Gambar 3.32 Analysis Collaboration Diagram : Use Case Display MRP

3.2.2.14 Analysis Collaboration Diagram : Use Case Entri Purchase Order



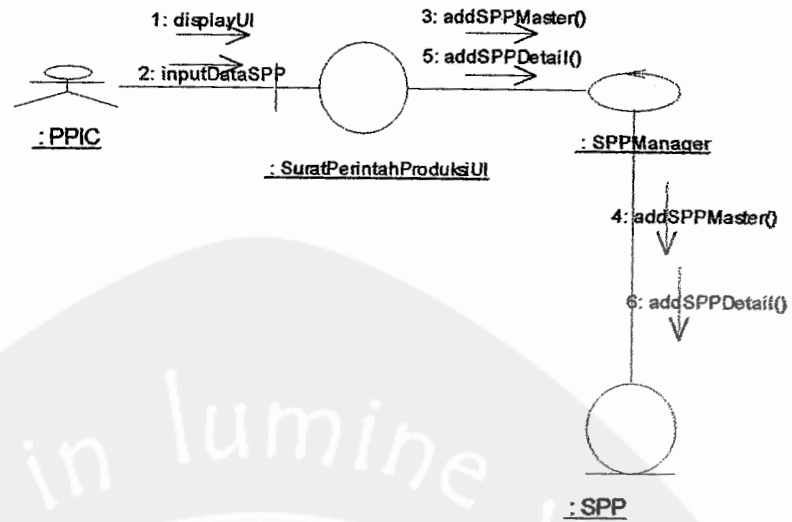
Gambar 3.33 Analysis Collaboration Diagram : Use Case Entri Purchase Order

3.2.2.15 Analysis Collaboration Diagram : Use Case Display Purchase Order



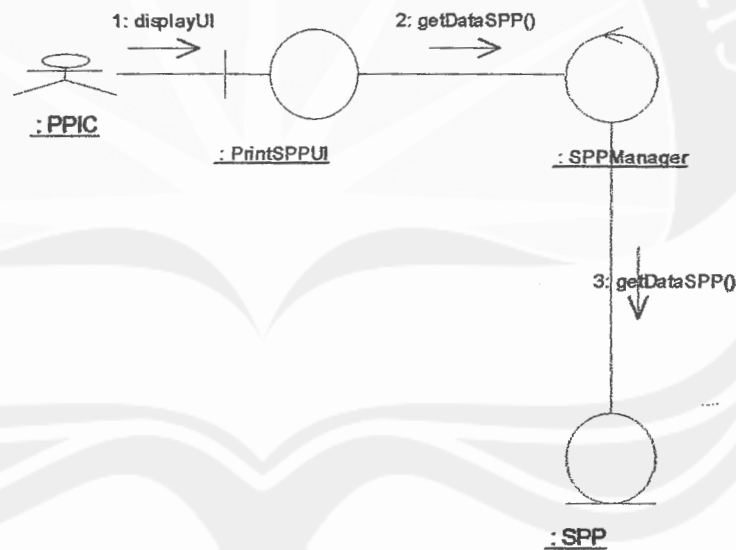
Gambar 3.34 Analysis Collaboration Diagram : Use Case Display Purchase Order

3.2.2.16 Analysis Collaboration Diagram : Use Case Surat Perintah Produksi



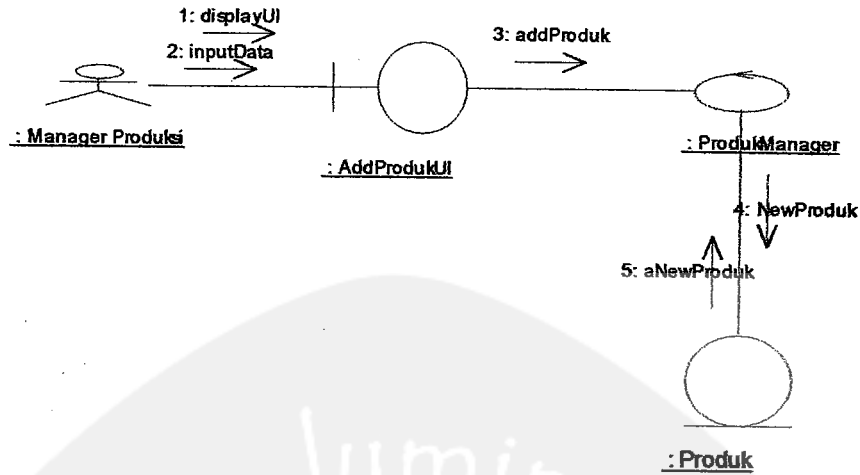
Gambar 3.35 Analysis Collaboration Diagram : Use Case Surat Perintah Produksi

3.2.2.17 Analysis Collaboration Diagram : Use Case Print SPP



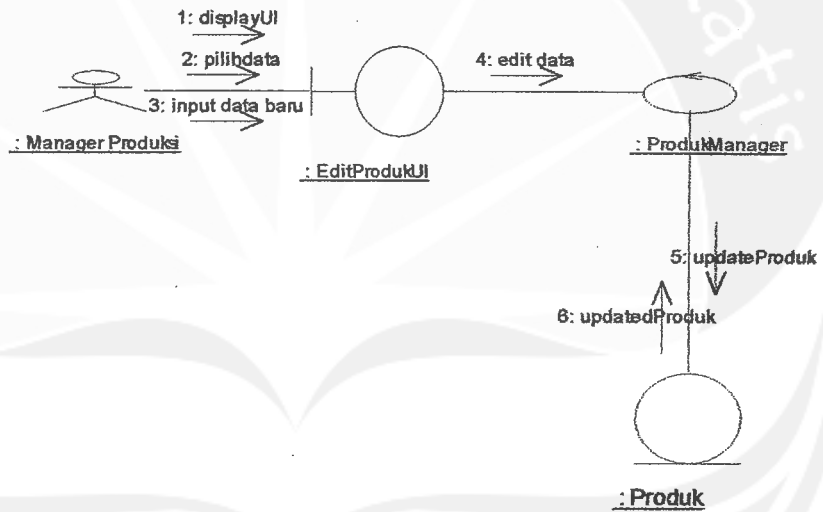
Gambar 3.36 Analysis Collaboration Diagram : Use Case Print SPP

3.2.2.18 Analysis Collaboration Diagram : Use Case Add Produk



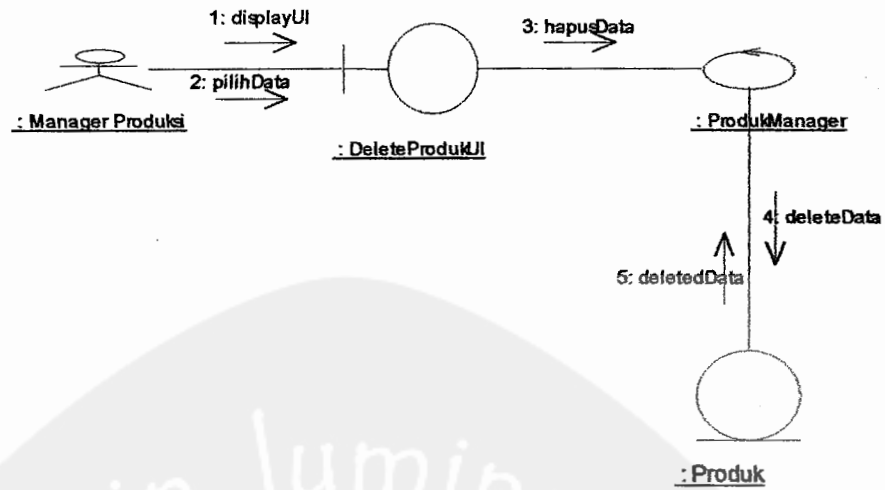
Gambar 3.37 Analysis Collaboration Diagram : Use Case Add Produk

3.2.2.19 Analysis Collaboration Diagram : Use Case Edit Produk



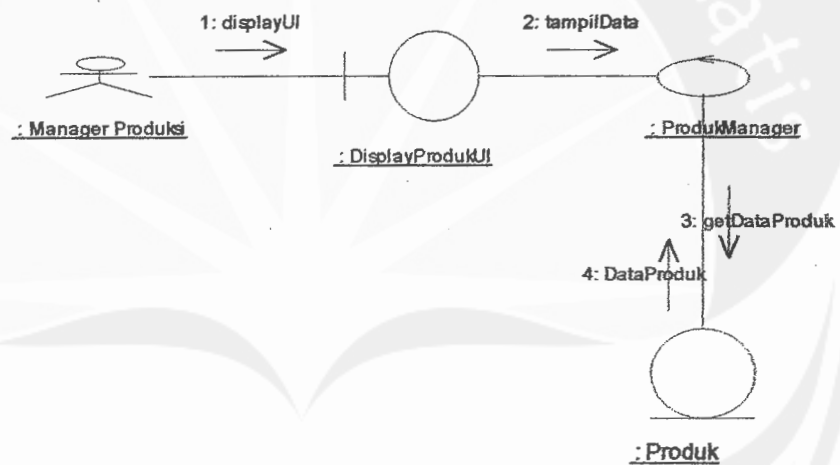
Gambar 3.38 Analysis Collaboration Diagram : Use Case Edit Produk

3.2.2.20 Analysis Collaboration Diagram : Use Case Delete Produk



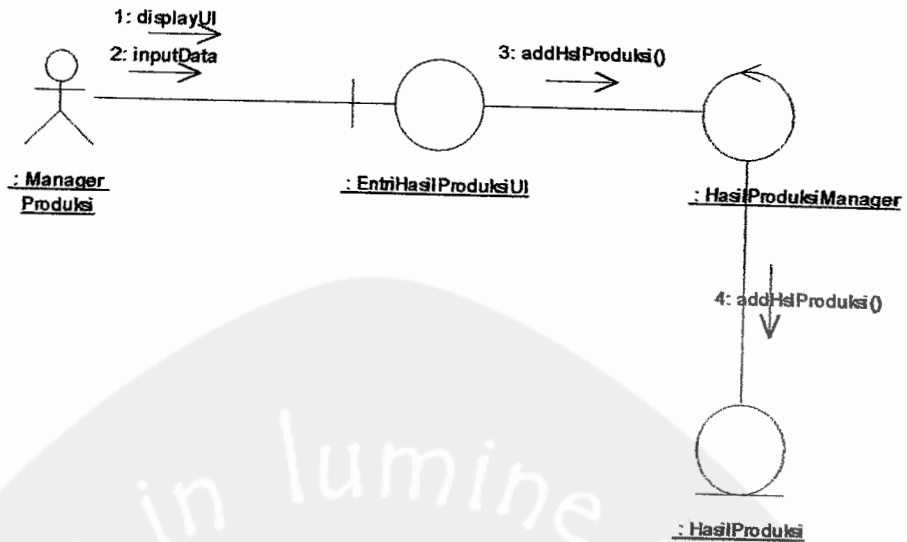
Gambar 3.39 Analysis Collaboration Diagram : Use Case Delete Produk

3.2.2.21 Analysis Collaboration Diagram : Use Case Display Produk



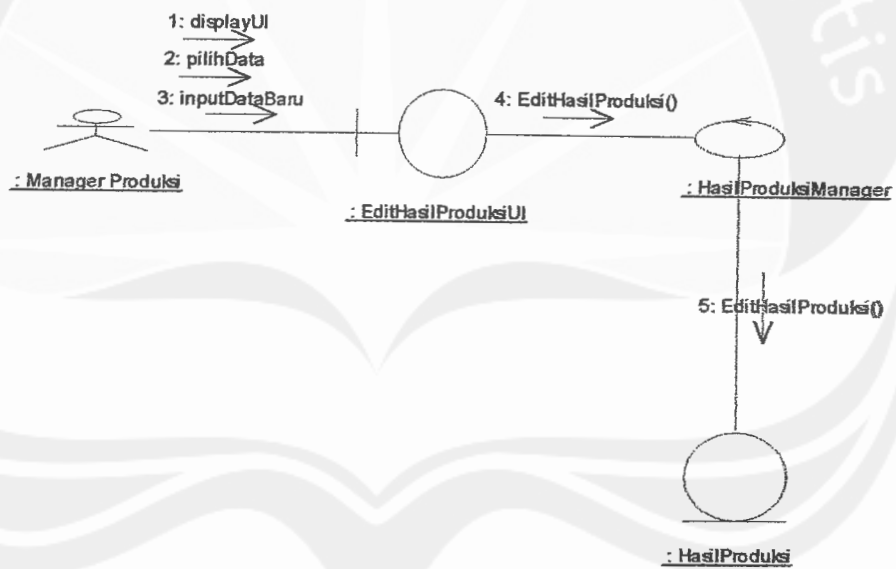
Gambar 3.40 Analysis Collaboration Diagram : Use Case Display Produk

3.2.2.22 Analysis Collaboration Diagram : Use Case Entri Hasil Produksi



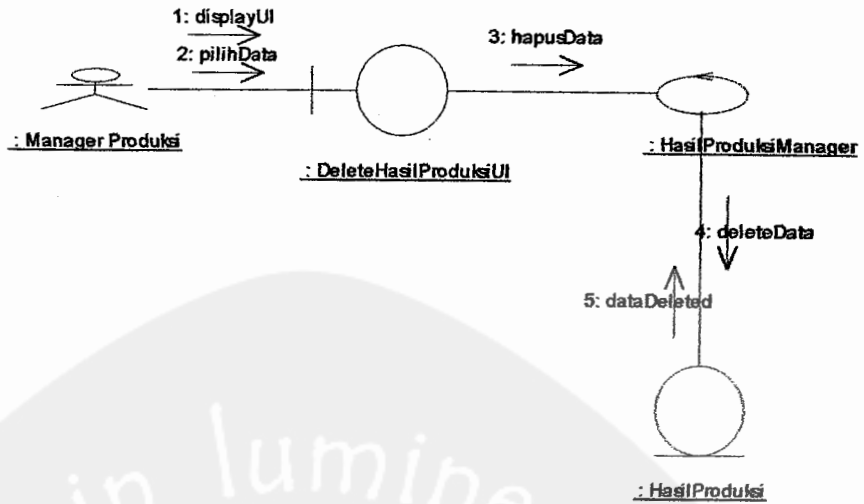
Gambar 3.41 Analysis Collaboration Diagram : Use Case Entri Hasil Produksi

3.2.2.23 Analysis Collaboration Diagram : Use Case Edit Hasil Produksi



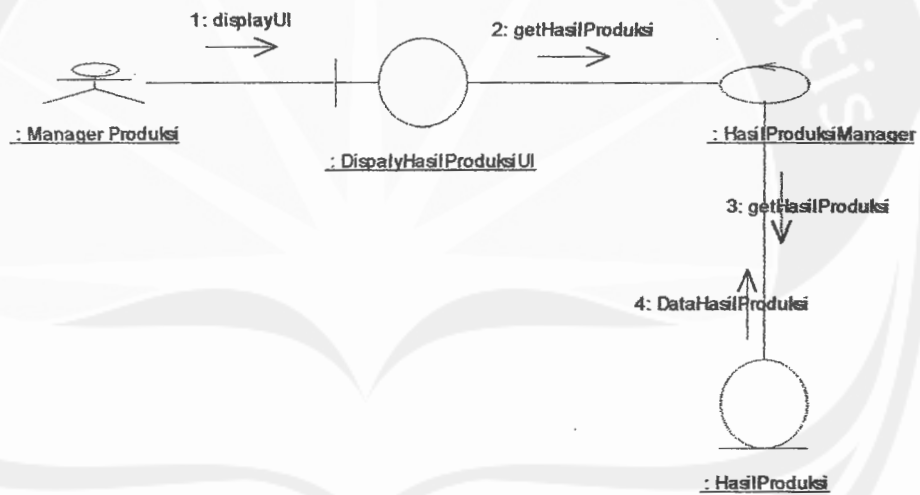
Gambar 3.42 Analysis Collaboration Diagram : Use Case Edit Hasil Produksi

3.2.2.24 Analysis Collaboration Diagram : Use Case Delete Hasil Produksi



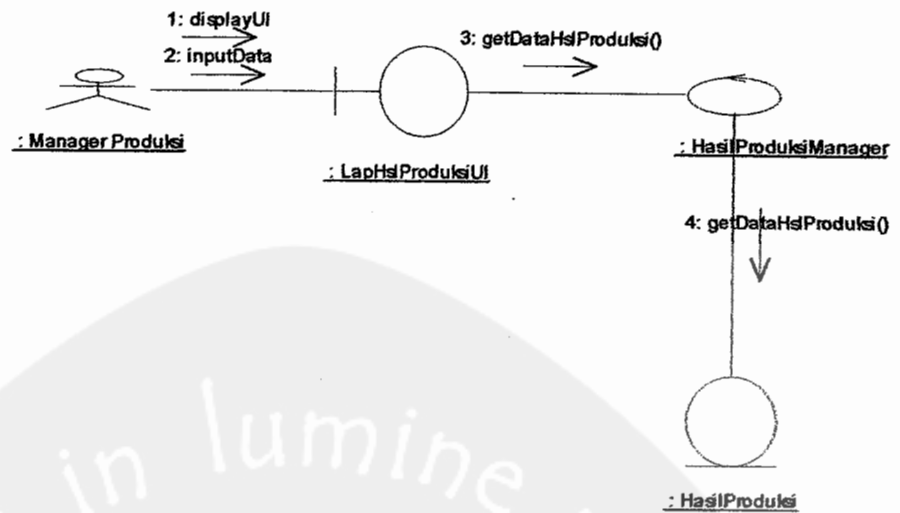
Gambar 3.43 Analysis Collaboration Diagram : Use Case Delete Hasil Produksi

3.2.2.25 Analysis Collaboration Diagram : Use Case Display Hasil Produksi



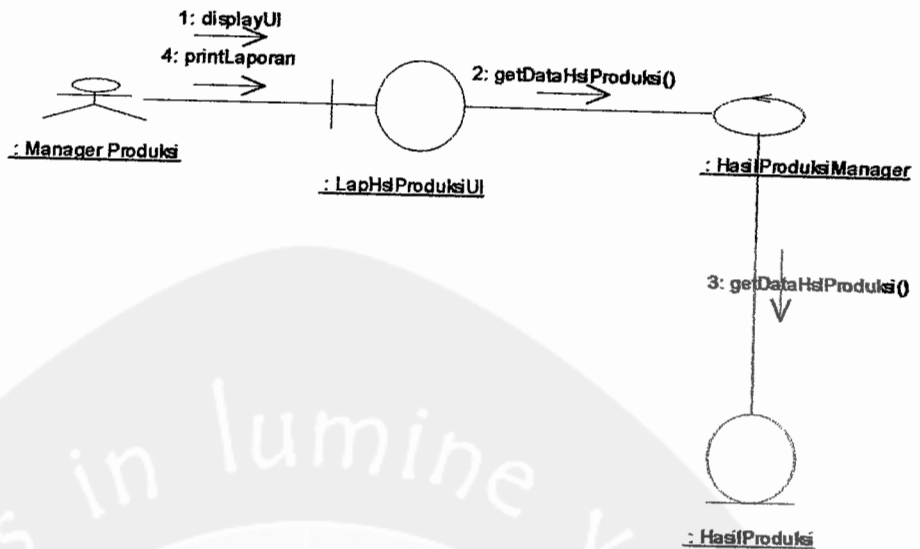
Gambar 3.44 Analysis Collaboration Diagram : Use Case Display Hasil Produksi

3.2.2.26 Analysis Collaboration Diagram : Use Case Laporan Hasil Produksi



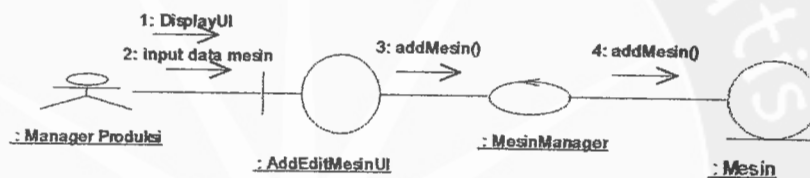
Gambar 3.45 Analysis Collaboration Diagram : Use Case Laporan Hasil Produksi

3.2.2.27 Analysis Collaboration Diagram : Use Case Print Laporan Produksi



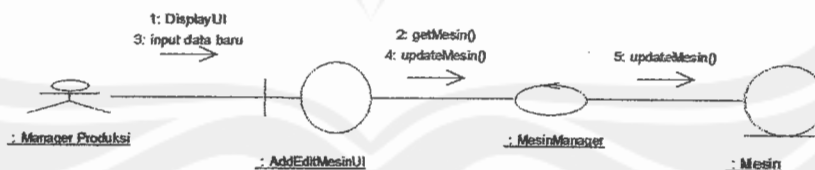
Gambar 3.46 Analysis Collaboration Diagram : Use Case Print Laporan Produksi

3.2.2.28 Analysis Collaboration Diagram : Use Case Add Mesin



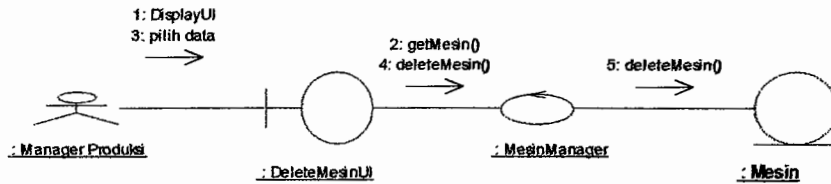
Gambar 3.47 Analysis Collaboration Diagram : Use Case Add Mesin

3.2.2.29 Analysis Collaboration Diagram : Use Case Edit Mesin



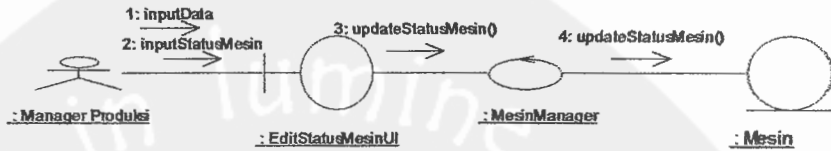
Gambar 3.48 Analysis Collaboration Diagram : Use Case Edit Mesin

3.2.2.30 Analysis Collaboration Diagram : Use Case Delete Mesin



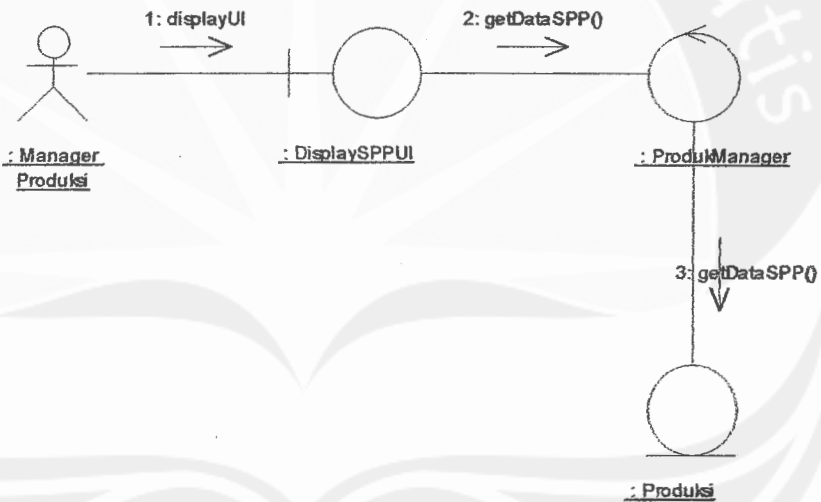
Gambar 3.49 Analysis Collaboration Diagram : Use Case Delete Mesin

3.2.2.31 Analysis Collaboration Diagram : Use Case Edit Status Mesin



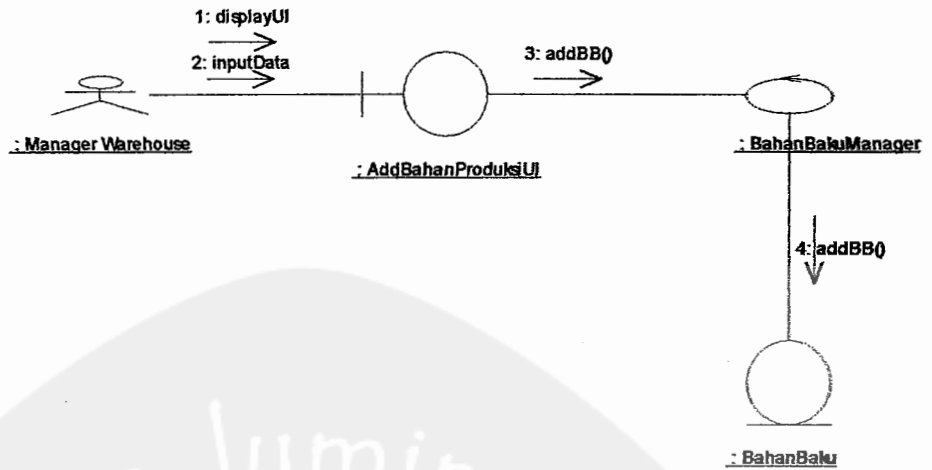
Gambar 3.50 Analysis Collaboration Diagram : Use Case Edit Status Mesin

3.2.2.32 Analysis Collaboration Diagram : Use Case Display SPP



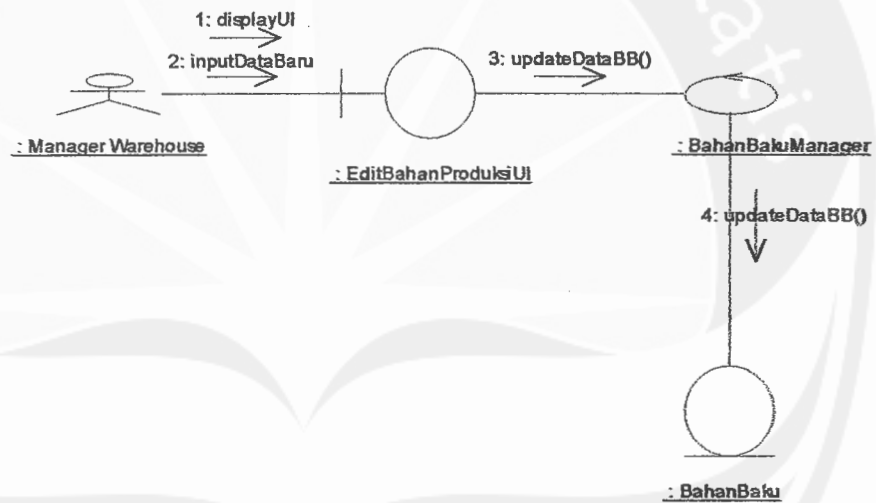
Gambar 3.51 Analysis Collaboration Diagram : Use Case Display SPP

3.2.2.33 Analysis Collaboration Diagram : Use Case Add Bahan Produksi



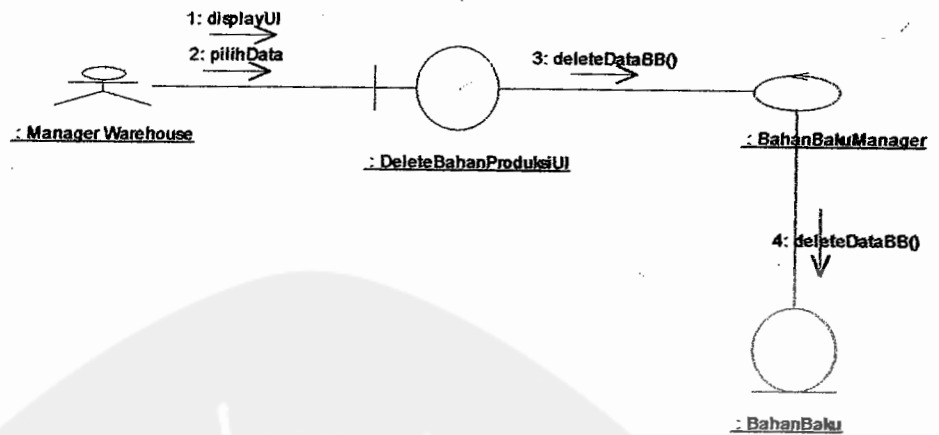
Gambar 3.52 Analysis Collaboration Diagram : Use Case Add Bahan Produksi

3.2.2.34 Analysis Collaboration Diagram : Use Case Edit Bahan Produksi



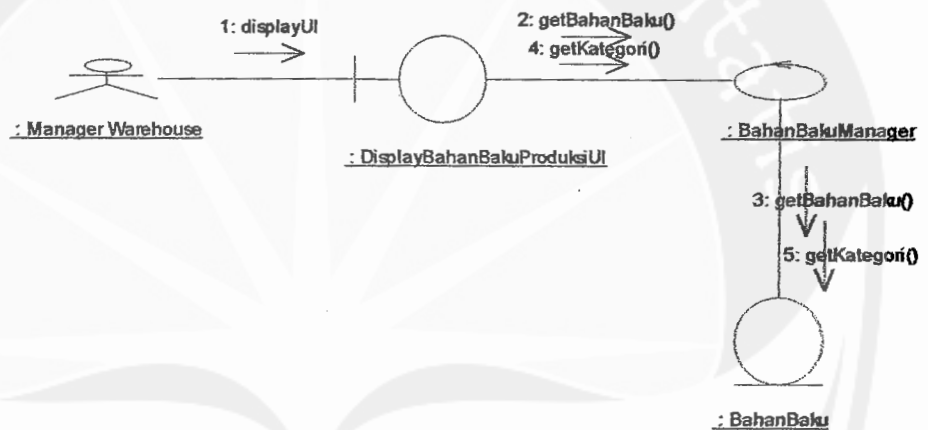
Gambar 3.53 Analysis Collaboration Diagram : Use Case Edit Bahan Produksi

3.2.2.35 Analysis Collaboration Diagram : Use Case Delete Bahan Produksi



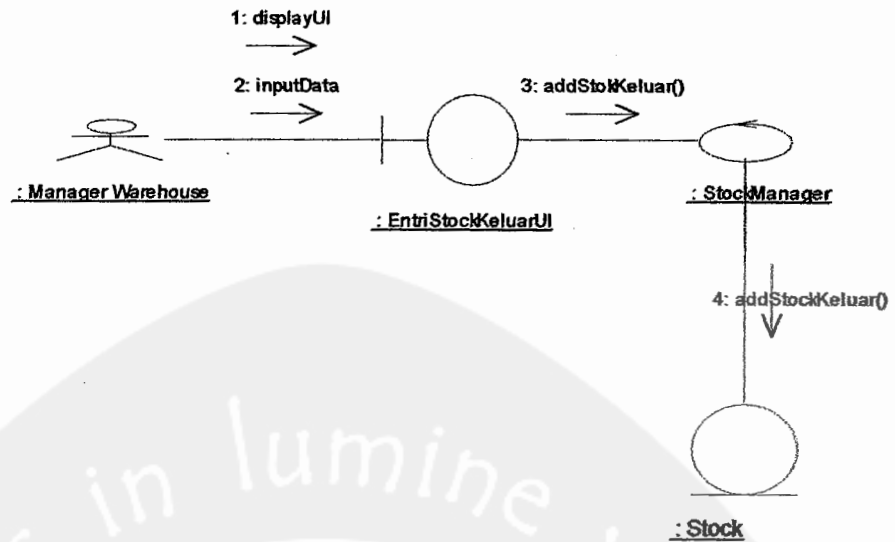
Gambar 3.54 Analysis Collaboration Diagram : Use Case Delete Bahan Produksi

3.2.2.36 Analysis Collaboration Diagram : Use Case Display Bahan Produksi



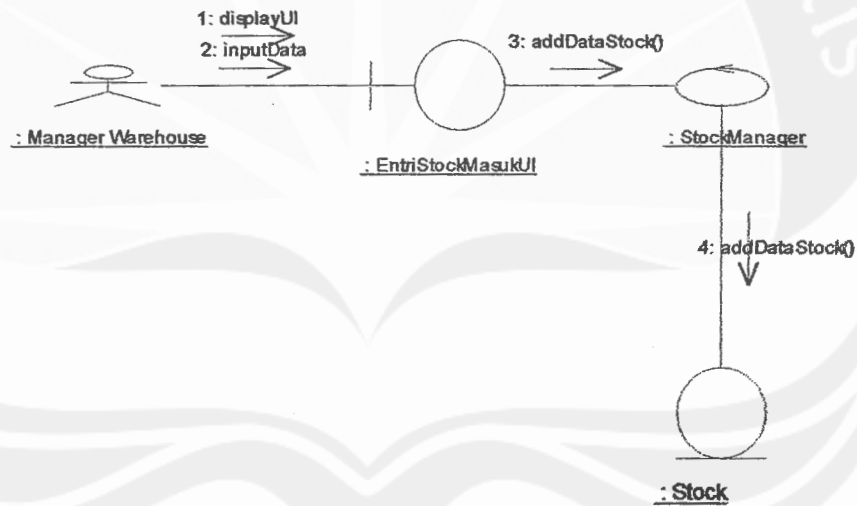
Gambar 3.55 Analysis Collaboration Diagram : Use Case Display Bahan Produksi

3.2.2.37 Analysis Collaboration Diagram : Use Case Entri Stok Keluar



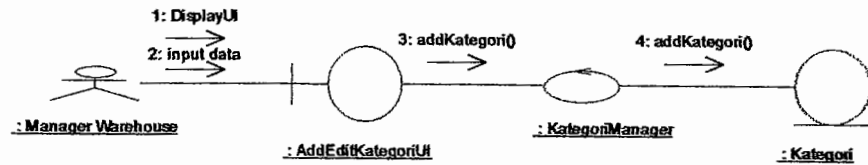
Gambar 3.56 Analysis Collaboration Diagram : Use Case Entri Stok Keluar

3.2.2.38 Analysis Collaboration Diagram : Use Case Entri Stok Masuk



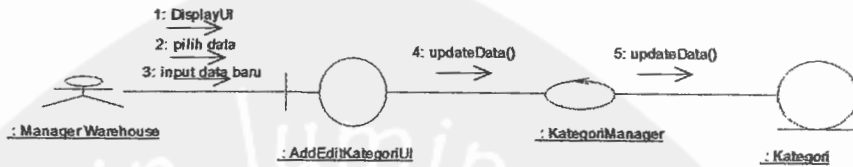
Gambar 3.57 Analysis Collaboration Diagram : Use Case Entri Stok Masuk

3.2.2.39 Analysis Collaboration Diagram : Use Case Add Kategori



Gambar 3.58 Analysis Collaboration Diagram : Use Case Add Kategori

3.2.2.40 Analysis Collaboration Diagram : Use Case Edit Kategori



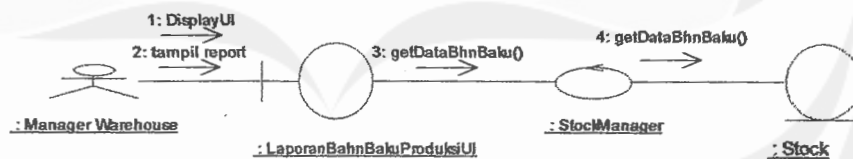
Gambar 3.59 Analysis Collaboration Diagram : Use Case Edit Kategori

3.2.2.41 Analysis Collaboration Diagram : Use Case Delete Kategori



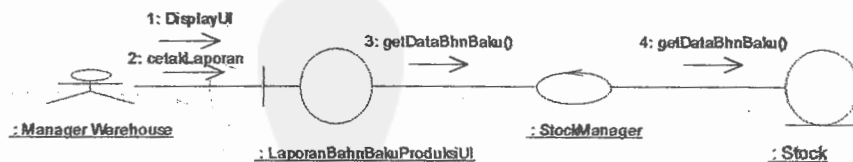
Gambar 3.60 Analysis Collaboration Diagram : Use Case Delete Kategori

3.2.2.42 Analysis Collaboration Diagram : Use Case Laporan BB Produksi



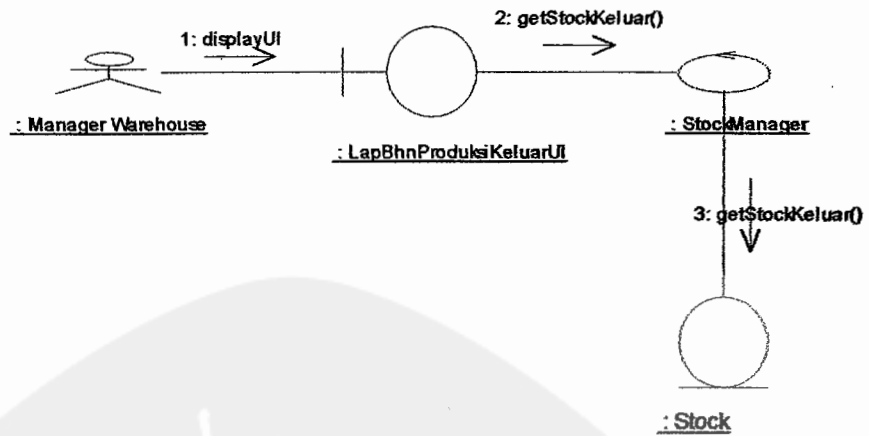
Gambar 3.61 Analysis Collaboration Diagram : Use Case Laporan BB Produksi

3.2.2.43 Analysis Collaboration Diagram : Use Case Print Laporan BB Produksi



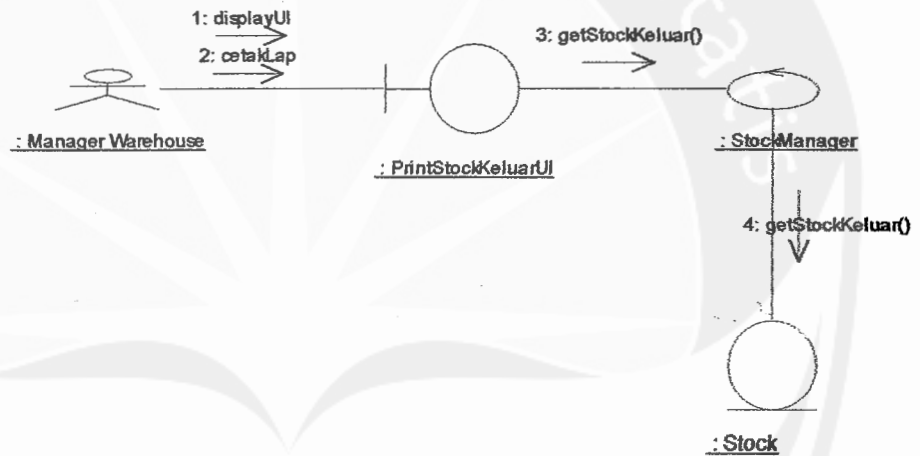
Gambar 3.62 Analysis Collaboration Diagram : Use Case Print Laporan BB Produksi

3.2.2.44 Analysis Collaboration Diagram : Use Case Lap BB Produksi Keluar



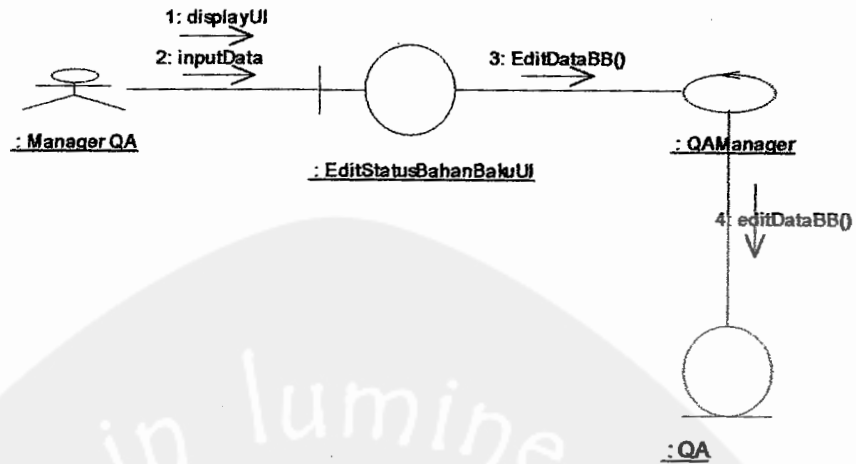
Gambar 3.63 Analysis Collaboration Diagram : Use Case Lap BB Produksi Keluar

3.2.2.45 Analysis Collaboration Diagram : Use Case Print Laporan Bahan Baku Produksi Keluar



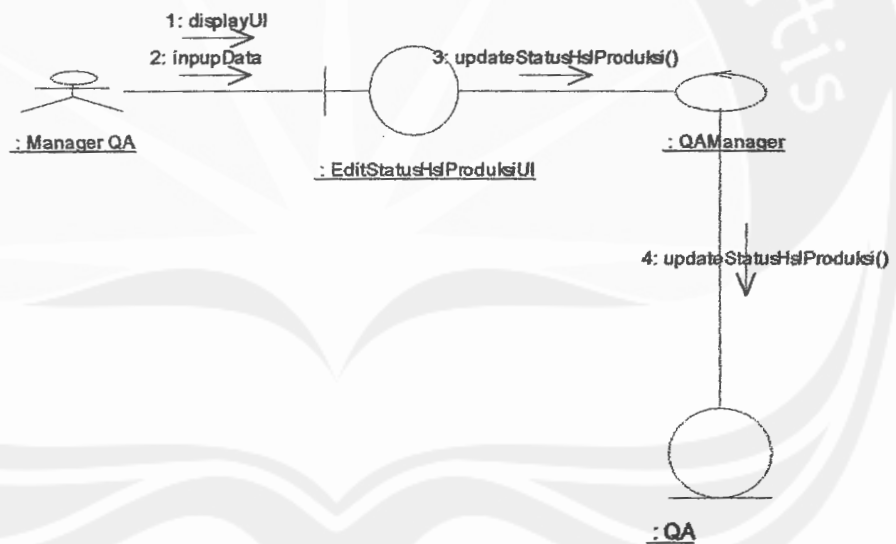
Gambar 3.64 Analysis Collaboration Diagram : Use Case Print Laporan Bahan Baku Produksi Keluar

3.2.2.46 Analysis Collaboration Diagram : Use Case Edit Status Bahan Baku Produksi



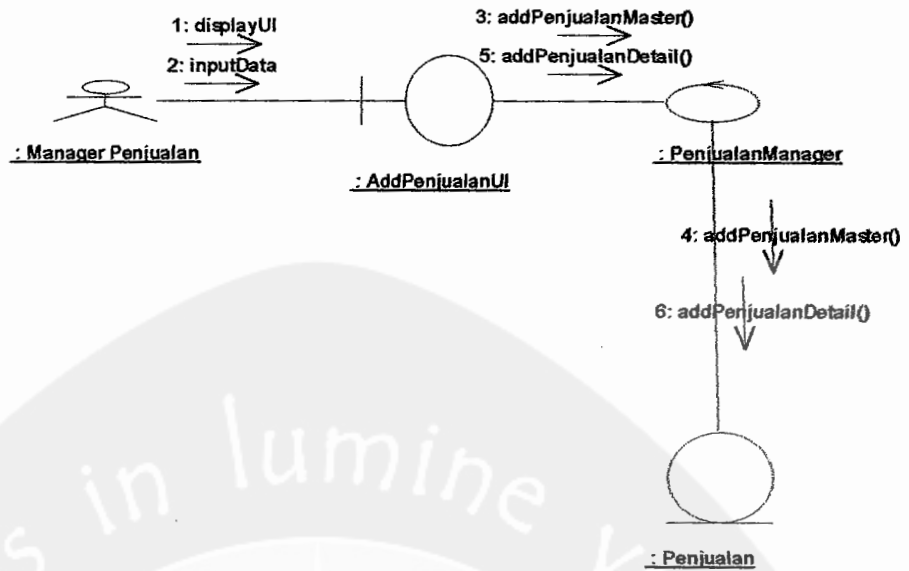
Gambar 3.65 Analysis Collaboration Diagram : Use Case Edit Status Bahan Baku

3.2.2.47 Analysis Collaboration Diagram : Use Case Edit Status Hasil Produksi



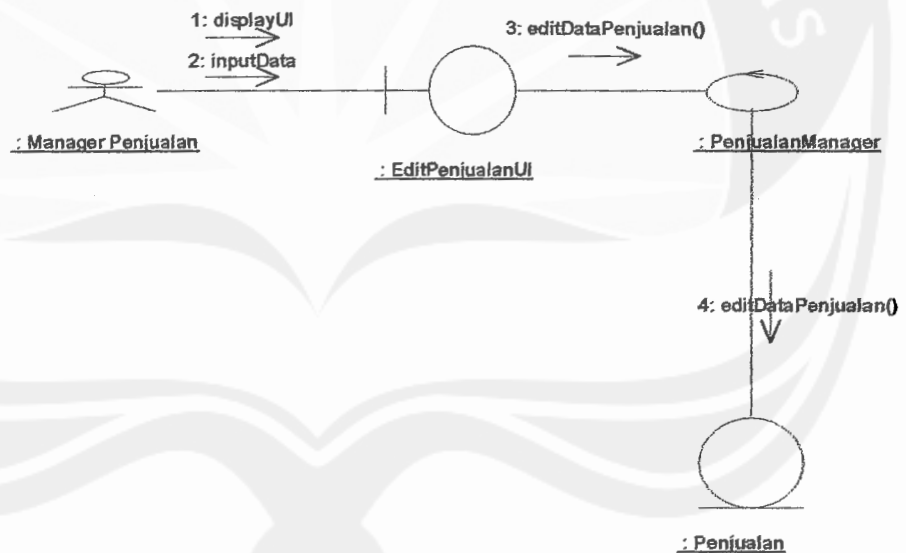
Gambar 3.66 Analysis Collaboration Diagram : Use Case Edit Status Hasil Produksi

3.2.2.48 Analysis Collaboration Diagram : Use Case Add Penjualan



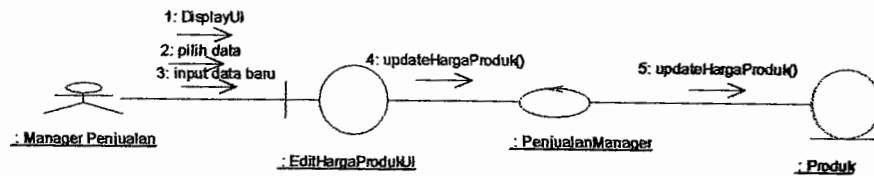
Gambar 3.67 Analysis Collaboration Diagram : Use Case Add Penjualan

3.2.2.49 Analysis Collaboration Diagram : Use Case Edit Penjualan



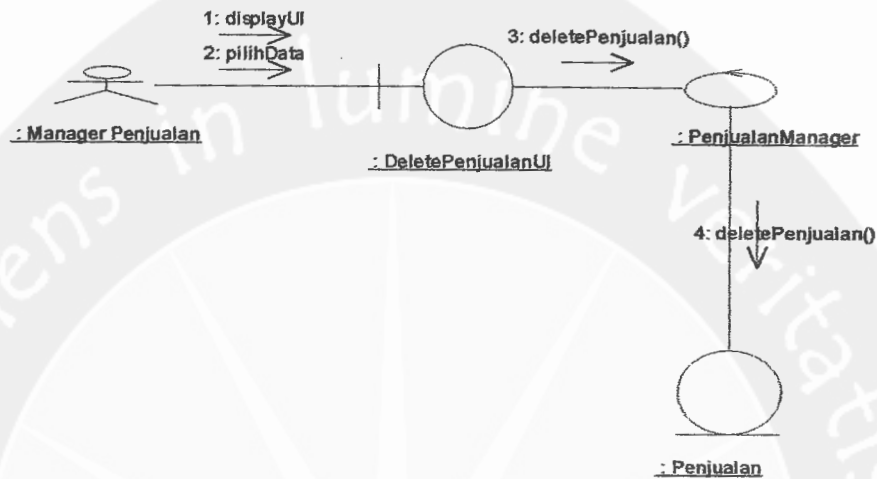
Gambar 3.68 Analysis Collaboration Diagram : Use Case Edit Penjualan

3.2.2.50 Analysis Collaboration Diagram : Use Case Edit Harga Produk



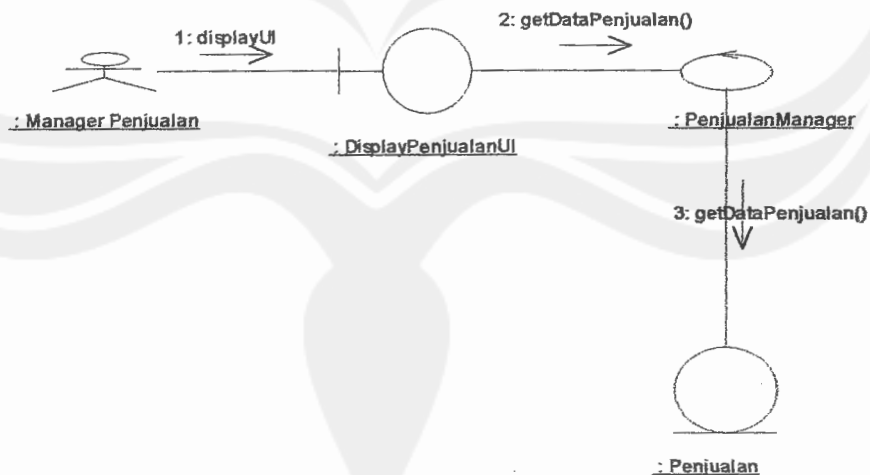
Gambar 3.69 Analysis Collaboration Diagram : Use Case Edit Harga Produk

3.2.2.51 Analysis Collaboration Diagram : Use Case Delete Penjualan



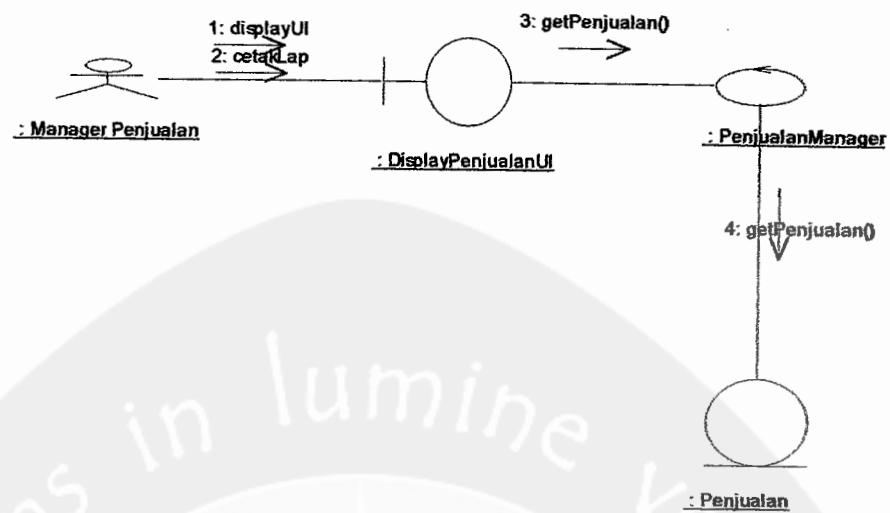
Gambar 3.70 Analysis Collaboration Diagram : Use Case Delete Penjualan

3.2.2.52 Analysis Collaboration Diagram : Use Case Display Penjualan



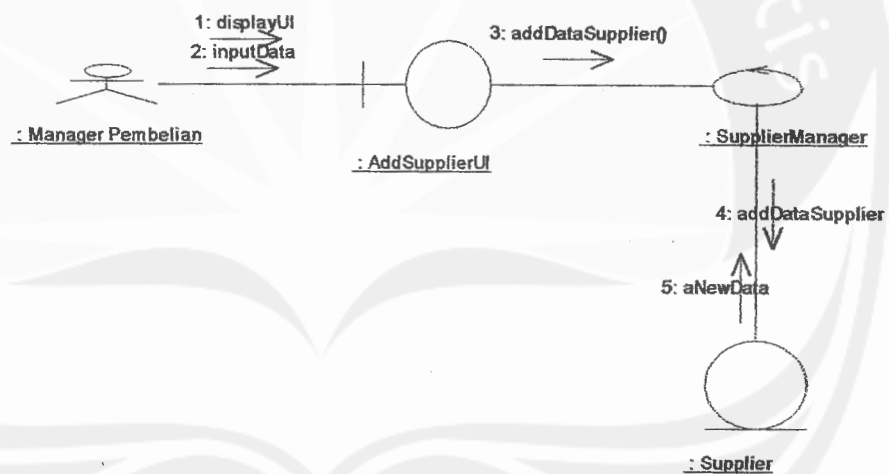
Gambar 3.71 Analysis Collaboration Diagram : Use Case Display Penjualan

3.2.2.53 Analysis Collaboration Diagram : Use Case Print Penjualan



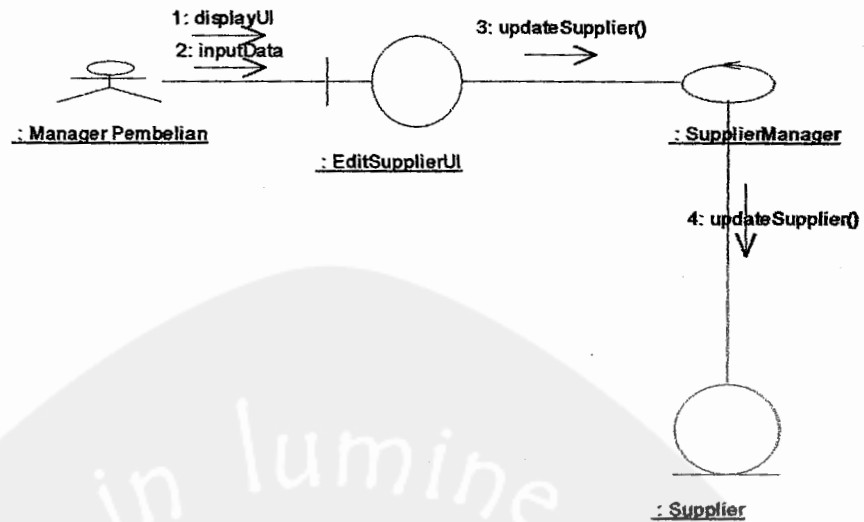
Gambar 3.72 Analysis Collaboration Diagram : Use Case Print Penjualan

3.2.2.54 Analysis Collaboration Diagram : Use Case Add Supplier



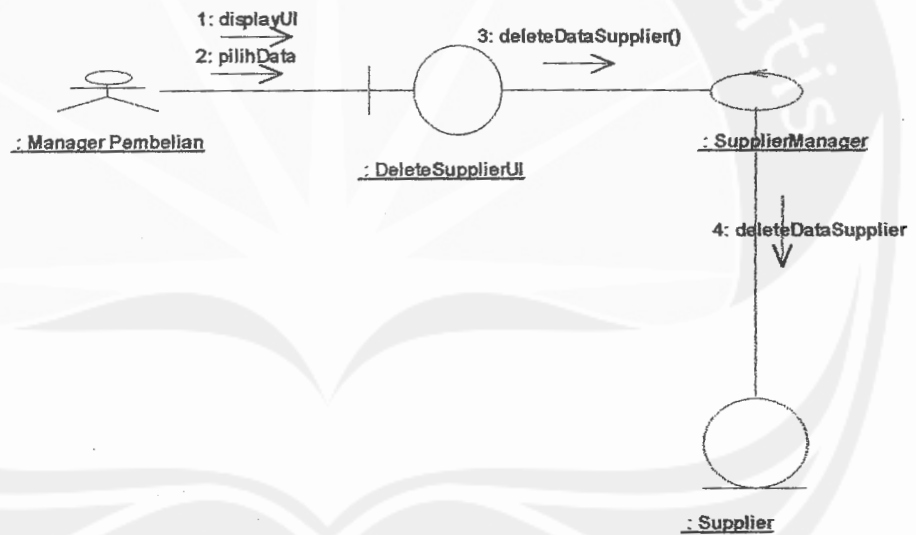
Gambar 3.73 Analysis Collaboration Diagram : Use Case Add Supplier

3.2.2.55 Analysis Collaboration Diagram : Use Case Edit Supplier



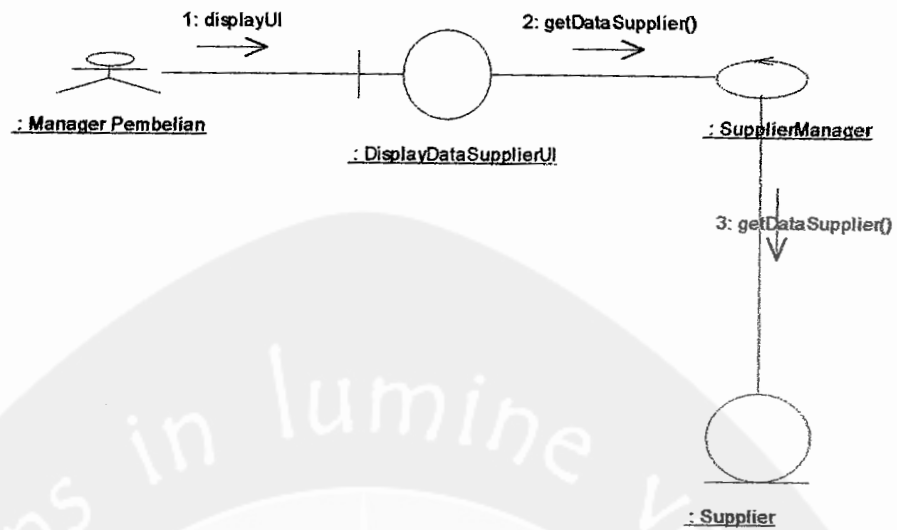
Gambar 3.74 Analysis Collaboration Diagram : Use Case Edit Supplier

3.2.2.56 Analysis Collaboration Diagram : Use Case Delete Supplier



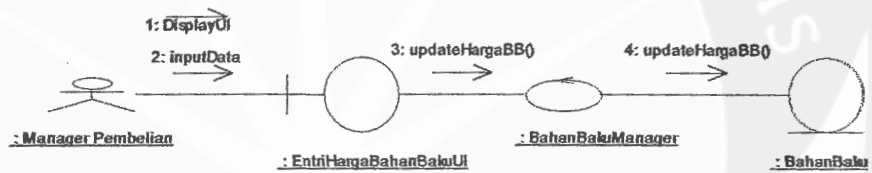
Gambar 3.75 Analysis Collaboration Diagram : Use Case Delete Supplier

3.2.2.57 Analisis Collaboration Diagram : Use Case Display Data Supplier



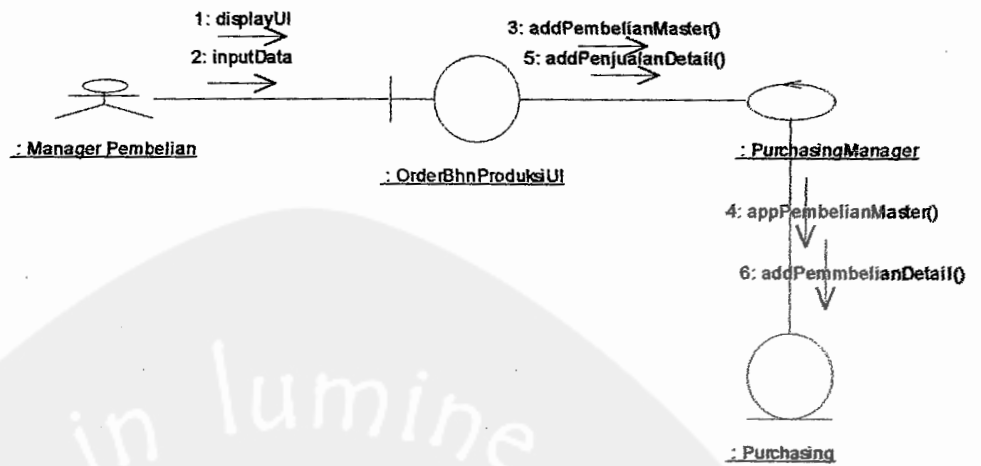
Gambar 3.76 Analisis Collaboration Diagram : Use Case Display Data Supplier

3.2.2.58 Analisis Collaboration Diagram : Use Case Edit Harga Bahan Baku



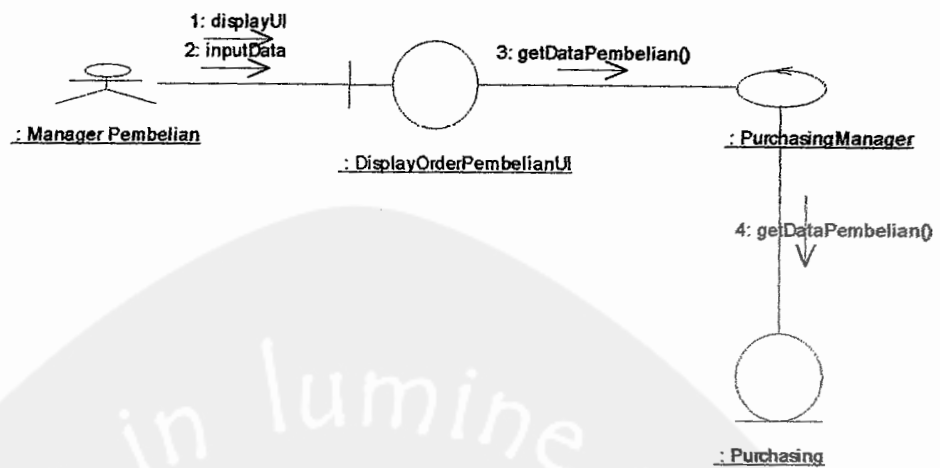
Gambar 3.77 Analisis Collaboration Diagram : Use Case Edit Harga Bahan Baku

3.2.2.59 Analysis Collaboration Diagram : Use Case Order Bahan Produksi



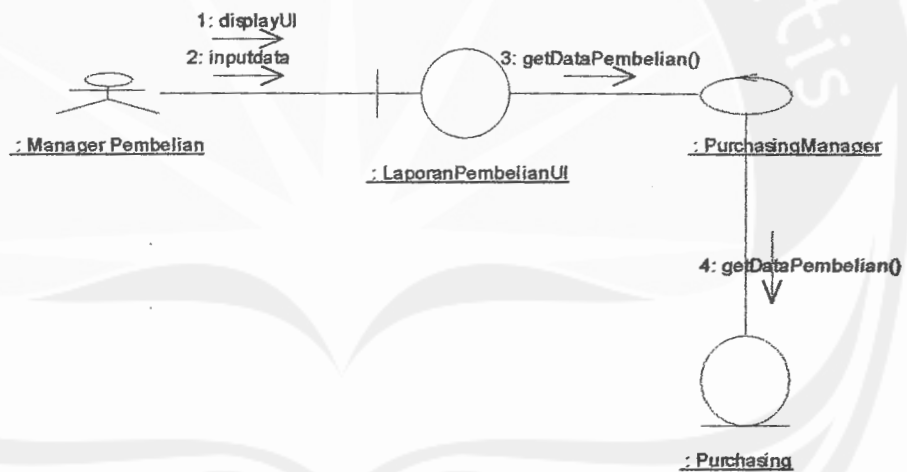
Gambar 3.78 Analysis Collaboration Diagram : Use Case Order Bahan Produksi

3.2.2.60 Analysis Collaboration Diagram : Use Case Display Order Pembelian



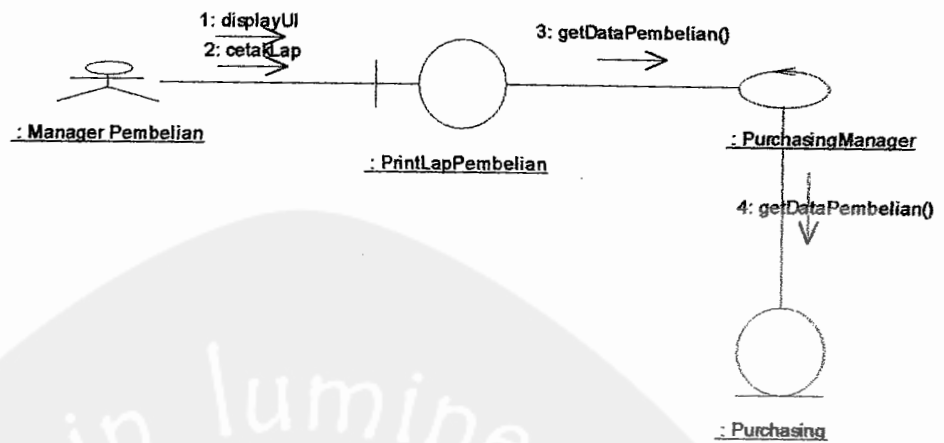
Gambar 3.79 Analysis Collaboration Diagram : Use Case Display Order Pembelian

3.2.2.61 Analysis Collaboration Diagram : Use Case Laporan Pembelian



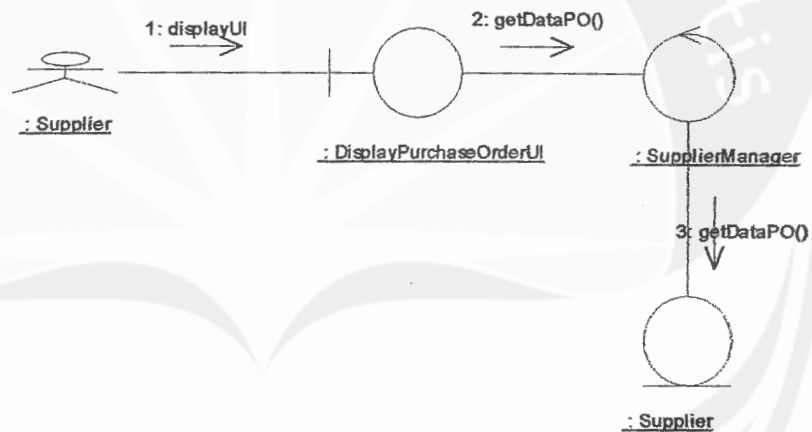
Gambar 3.80 Analysis Collaboration Diagram : Use Case Laporan Pembelian

3.2.2.62 Analysis Collaboration Diagram : Use Case Print Laporan Pembelian



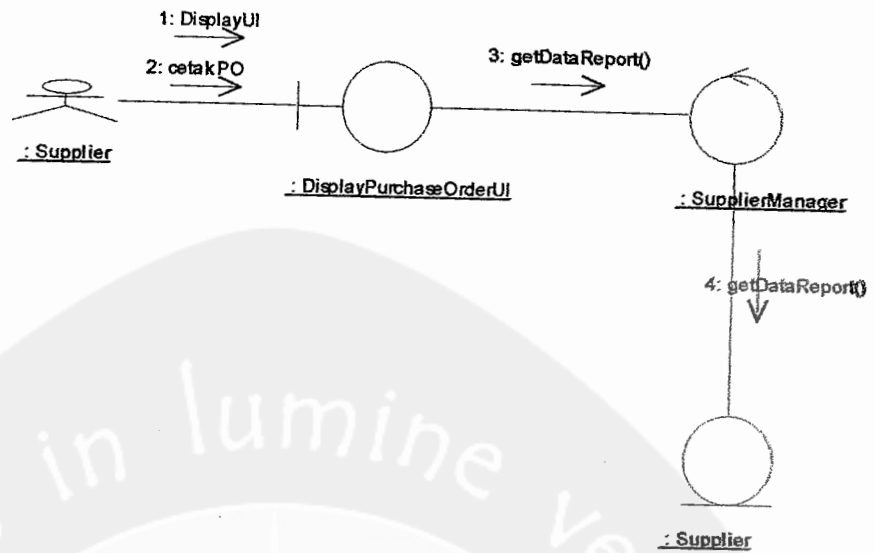
Gambar 3.81 Analysis Collaboration Diagram : Use Case Print Laporan Pembelian

3.2.2.63 Analysis Collaboration Diagram : Use Case Display Purchase Order



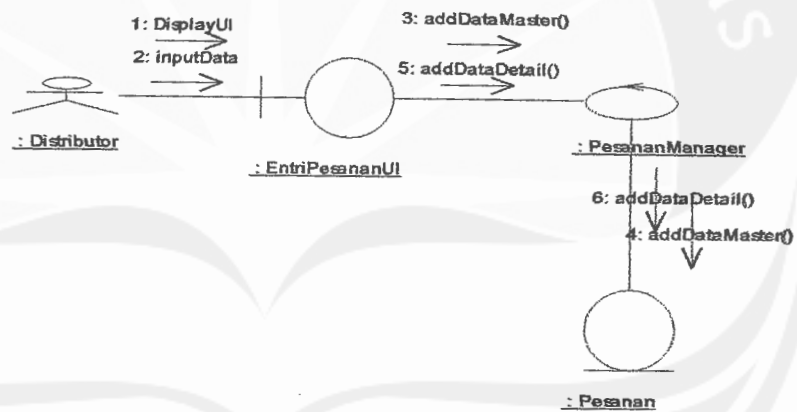
Gambar 3.82 Analysis Collaboration Diagram : Use Case Display Purchase Order

3.2.2.64 Analysis Collaboration Diagram : Use Case Print Purchase Order



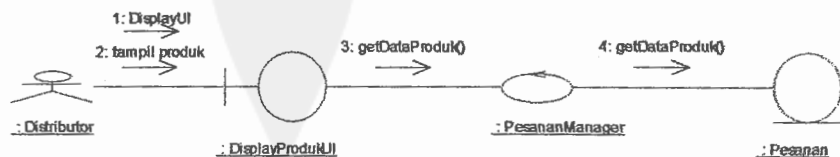
Gambar 3.83 Analysis Collaboration Diagram : Use Case Print Purchase Order

3.2.2.65 Analysis Collaboration Diagram : Use Case Entri Pesanan



Gambar 3.84 Analysis Collaboration Diagram : Use Case Entri Pesanan

3.2.2.66 Analysis Collaboration Diagram : Use Case Display Produk



Gambar 3.85 Analysis Collaboration Diagram : Use Case Display Produk

3.3 Spesifikasi Kebutuhan Non-Fungsionalitas

3.3.1 Antarmuka pemakai

Pemakai berinteraksi langsung dengan perangkat lunak melalui masukan perintah yang diketikan langsung dari *keyboard* atau penggunaan *mouse* pada *link* yang sudah disediakan, sedangkan hasil keluaran akan ditampilkan langsung ke layar *monitor* dalam format halaman Web.

3.3.2 Antarmuka perangkat keras

Antarmuka perangkat keras yang digunakan dalam perangkat lunak ASCaMaDa adalah:

1. *PC IBM Compatible Pentium 4*
2. *RAM minimal 256 MB*
3. *Harddisk*
4. *Layar Monitor*
5. *Keyboard dan mouse*

3.3.3 Antarmuka perangkat lunak

Perangkat lunak yang dibutuhkan untuk mengoperasikan perangkat lunak ASCaMaDa adalah :

1. Nama : *Visual C#.NET, ASP.NET*

Sumber : *Microsoft Visual Studio .NET*

Sebagai tool perancangan yang dibutuhkan dalam pengembangan perangkat lunak ASCaMaDa.

2. Nama : *SQL Server 2005*

Sumber : *Microsoft*

Sebagai database yang dibutuhkan dalam mengoperasikan perangkat lunak ASCaMaDa.

3. Nama : *Windows 2000/XP/2003*

Sumber : *Microsoft*

Sebagai sistem operasi komputer.

4. Nama : *Internet Explorer 6.0*

Sumber : *Microsoft*

Sebagai *browser* Internet bagi *client*.

5. Nama : *Internet Information Services (IIS)*

Sumber : *Microsoft*

Sebagai *Web Server*.

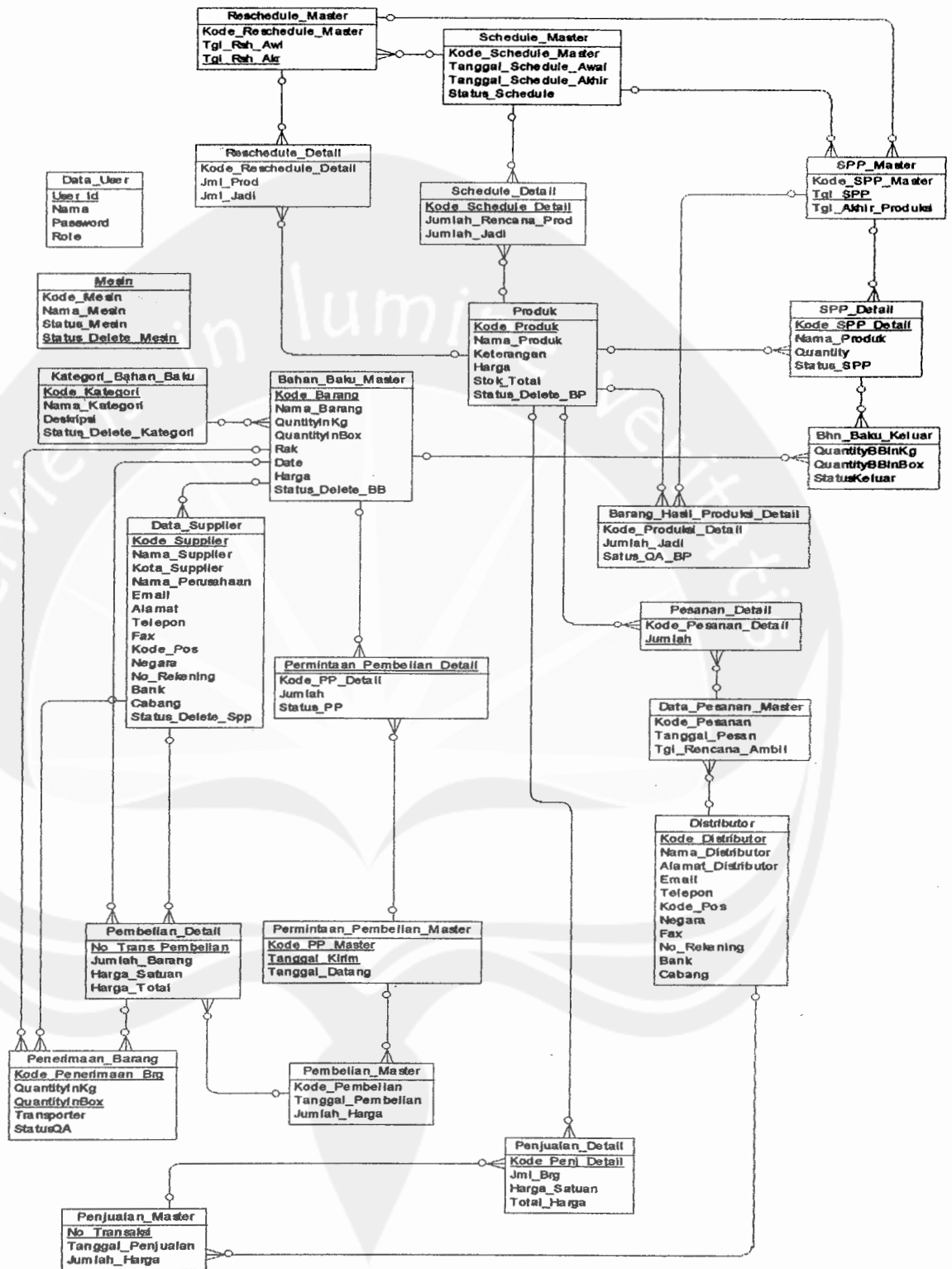
3.3.4 Antarmuka Komunikasi

Antarmuka komunikasi perangkat lunak ASCaMaDa menggunakan *protocol* TCP/IP.



4 Spesifikasi Kebutuhan Data

4.1 Entity Relationship Diagram (ERD)



Gambar 3.86 Entity Relationship Diagram ASCaMaDa

DPPL

DESKRIPSI PERANCANGAN PERANGKAT LUNAK
Pembangunan Aplikasi Supply Chain Management

(Studi Kasus: PT. Sari Husada)


(ASCaMaDa)

Dipersiapkan oleh:

K. JOKO MARYONO

03 07 03937

Program Studi Teknik Informatika – Fakultas Teknologi Industri
Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		DPPL- ASCaMaDa		1/277
		Revisi		Tgl : 28-06-2007

Program Studi Teknik Informatika	DPPL-ASCaMaDa	1/277
----------------------------------	---------------	-------

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika

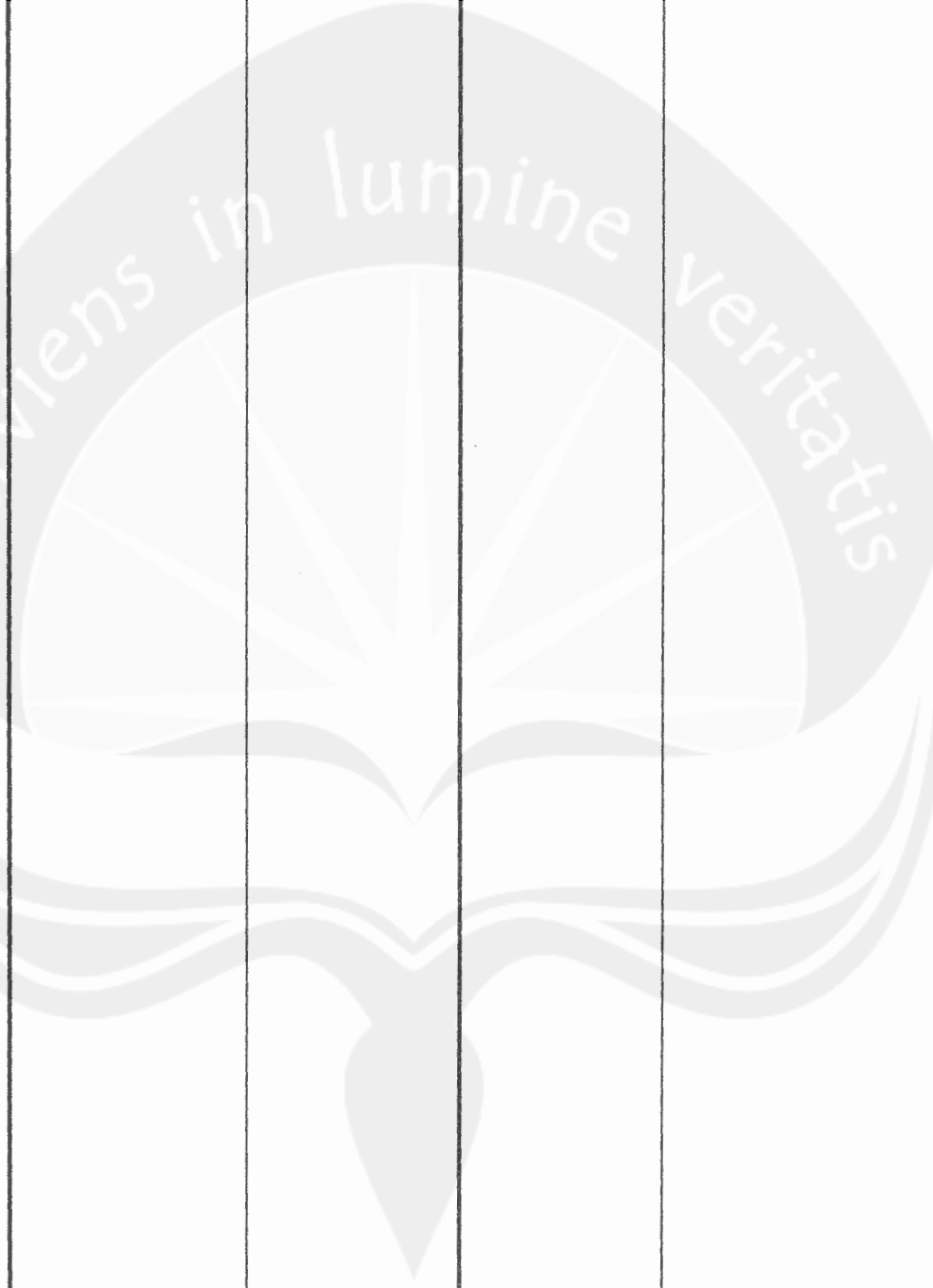
DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	
G	

INDEX TGL	-	A	B	C	D	E	F
Ditulis oleh							
Diperiksa oleh							
Disetujui oleh							

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi



Daftar Isi

Daftar Tabel	13
1 Pendahuluan.....	14
1.1 Tujuan	14
1.2 Lingkup Dokumen	14
1.3 Istilah dan Singkatan	14
1.4 Referensi.....	15
1.5 Deskripsi umum (Overview)	15
2 Deskripsi Perancangan Arsitektural	16
2.1 Deployment Diagram	16
2.1.1 Node : Client.....	16
2.1.2 Node : Web Server	16
2.1.3 Node : Database Server	17
2.2 Design Class.....	17
2.2.1 Pengantar	17
2.2.2 Package Dependencies.....	18
2.2.3 Package Data Administrator (ASCM.Administrator)	18
2.2.3.1 Class diagram package ASCM.Administrator	19
2.2.3.2 Class ASCM.Administrator.User	19
2.2.3.3 Class ASCM.Administrator.UserManager	21
2.2.3.4 Class ASCM.Administrator.LoginUI	23
2.2.3.5 Class ASCM.Administrator.AddUserUI	24
2.2.3.6 Class ASCM.Administrator.ChangePasswordUI	25
2.2.3.7 Class ASCM.Administrator.DeleteUserUI	25
2.2.3.8 Class ASCM.Administrator.EditUserUI.....	26
2.2.3.9 Class ASCM.Administrator.DisplayUserUI	26
2.2.4 Package PPIC (ASCM.PPIC)	27
2.2.4.1 Class diagram package ASCM.PPIC	27
2.2.4.2 Class ASCM.PPIC.Schedule.....	28
2.2.4.3 Class ASCM.PPIC.ScheduleManager	30
2.2.4.4 Class ASCM.PPIC.EntriScheduleUI	31
2.2.4.5 Class ASCM.PPIC.DisplayScheduleUI	32
2.2.4.6 Class ASCM.PPIC.Reschedule	32
2.2.4.7 Class ASCM.PPIC.RescheduleManager.....	35
2.2.4.8 Class ASCM.PPIC.EntriRescheduleUI.....	37
2.2.4.9 Class ASCM.PPIC.DisplayRescheduleUI	37
2.2.4.10 Class ASCM.PPIC.PurchaseOrde	38
2.2.4.11 Class ASCM.PPIC.PurchaseOrderManager	40
2.2.4.12 Class ASCM.PPIC.EntriPurchaseOrderUI	42
2.2.4.13 Class ASCM.PPIC.DisplayPurchaseOrderUI	42
2.2.4.14 Class ASCM.PPIC.SPP	43
2.2.4.15 Class ASCM.PPIC.SPPManager	46
2.2.4.16 Class ASCM.PPIC.SuratPerintahProduksiUI	47
2.2.4.17 Class ASCM.PPIC.PrintSPPUI	48
2.2.4.18 Class ASCM.PPIC.DisplayMRPUI	48
2.2.4.19 Class ASCM.PPIC.DisplayCRPUI.....	49
2.2.4.20 Class ASCM.PPIC.DisplayProdukPesanan	50
2.2.5 Package Produksi (ASCM.Produksi)	51
2.2.5.1 Class diagram package ASCM.Produksi.....	51
2.2.5.2 Class ASCM.Produksi.Produk.....	52
2.2.5.3 Class ASCM.Produksi.ProdukManager.....	53
2.2.5.4 Class ASCM.Produksi.AddProdukUI.....	54
2.2.5.5 Class ASCM.Produksi.EditProdukUI.....	55

2.2.5.6	Class ASCM.Produksi.DeleteProdukUI.....	55
2.2.5.7	Class ASCM.Produksi.DisplayProdukUI.....	56
2.2.5.8	Class ASCM.Produksi.Mesin.....	57
2.2.5.9	Class ASCM.Produksi.MesinManager.....	58
2.2.5.10	Class ASCM.Produksi.AddMesinUI.....	59
2.2.5.11	Class ASCM.Produksi.EditStatusMesinUI.....	60
2.2.5.12	Class ASCM.Produksi.DeleteMesinUI.....	60
2.2.5.13	Class ASCM.Produksi.HasilProduksi.....	61
2.2.5.14	Class ASCM.Produksi.HasilProduksiManager.....	62
2.2.5.15	Class ASCM.Produksi.EntriHasilProduksiUI.....	63
2.2.5.16	Class ASCM.Produksi.EditHasilProduksiUI.....	64
2.2.5.17	Class ASCM.Produk.DisplayHasilProduksiUI.....	65
2.2.5.18	Class ASCM.Produk.DeleteHasilProduksiUI.....	65
2.2.5.19	Class ASCM.Produk.LapHslProduksiUI.....	66
2.2.6	Package Warehouse (ASCM.Warehouse).....	67
2.2.6.1	Class diagram package ASCM.Warehouse.....	67
2.2.6.2	Class ASCM.Warehouse.Stock.....	68
2.2.6.3	Class ASCM.Warehouse.StockManager.....	70
2.2.6.4	Class ASCM.Warehouse.EntriStockMasukUI.....	72
2.2.6.5	Class ASCM.Warehouse.EntriStockKeluarUI.....	72
2.2.6.6	Class ASCM.Warehouse.LaporanBhnProduksiKeluarUI.....	73
2.2.6.7	Class ASCM.Warehouse.LapBhnBakuProduksiUI.....	73
2.2.6.8	Class ASCM.Warehouse.Kategori.....	74
2.2.6.9	Class ASCM.Warehouse.KategoriManager.....	75
2.2.6.10	Class ASCM.Warehouse.AddEditKategoriUI.....	76
2.2.6.11	Class ASCM.Warehouse.DeleteKategoriUI.....	77
2.2.6.12	Class ASCM.Warehouse.BahanBaku.....	77
2.2.6.13	Class ASCM.Warehouse.BahanBakuManager.....	79
2.2.6.14	Class ASCM.Warehouse.AddBahanProduksiUI.....	80
2.2.6.15	Class ASCM.Warehouse.EditBahanBakuUI.....	81
2.2.6.16	Class ASCM.Warehouse.DeleteBahanProduksiUI.....	81
2.2.6.17	Class ASCM.Warehouse.DisplayBahanBakuProduksiUI.....	82
2.2.7	Package QualityAssurance (ASCM.QualityAssurance).....	83
2.2.7.1	Class diagram package ASCM.QualityAssurance.....	83
2.2.7.2	Class ASCM.QualityAssurance.QualityControl.....	83
2.2.7.3	Class ASCM.QualityAssurance.QualityControlManager.....	85
2.2.7.4	Class ASCM.QualityAssurance.EditStatusbahanBakuUI.....	86
2.2.7.5	Class ASCM.QualityAssurance.EditStatusHslProduksiUI.....	87
2.2.8	Package Marketing (ASCM.Marketing).....	88
2.2.8.1	Class diagram package ASCM.Marketing.....	88
2.2.8.2	Class ASCM.Marketing.Penjualan.....	88
2.2.8.3	Class ASCM.Marketing.PenjualanManager.....	92
2.2.8.4	Class ASCM.Marketing.AddPenjualanUI.....	94
2.2.8.5	Class ASCM.Marketing.EditPenjualanUI.....	94
2.2.8.6	Class ASCM.Marketing.DeletePenjualanUI.....	95
2.2.8.7	Class ASCM.Marketing.DisplayPenjualanUI.....	95
2.2.8.8	Class ASCM.Marketing.EditHargaProdukUI.....	96
2.2.9	Package Purchasing (ASCM.Purchasing).....	96
2.2.9.1	Class diagram package ASCM.Purchasing.....	97
2.2.9.2	Class ASCM.Purchasing.Purchasing.....	98
2.2.9.3	Class ASCM.Purchasing.PurchasingManager.....	101
2.2.9.4	Class ASCM.Purchasing.OrderBahanProduksiUI.....	103
2.2.9.5	Class ASCM.Purchasing.DisplayOrderPembelianUI.....	104
2.2.9.6	Class ASCM.Purchasing.LaporanPembelianUI.....	104
2.2.9.7	Class ASCM.Purchasing.Supplier.....	105

2.2.9.8	<i>Class ASCM.Purchasing.SupplierManager</i>	107
2.2.9.9	<i>Class ASCM.Purchasing.AddSupplierUI</i>	109
2.2.9.10	<i>Class ASCM.Purchasing.EditSupplierUI</i>	109
2.2.9.11	<i>Class ASCM.Purchasing.DeleteSupplierUI</i>	110
2.2.9.12	<i>Class ASCM.Purchasing.DisplayDataSupplierUI</i>	110
2.2.9.13	<i>Class ASCM.Purchasing.EditHargaBahanBakuUI</i>	111
2.2.10	Package Supplier (ASCM.Supplier).....	112
2.2.10.1	<i>Class diagram package ASCM.Supplier</i>	112
2.2.10.2	<i>Class ASCM.Supplier.SupplierUI</i>	112
2.2.11	Package Distributor (ASCM.Distributor).....	113
2.2.11.1	<i>Class diagram package ASCM.Distributor</i>	113
2.2.11.2	<i>Class ASCM.Distributor.Pesanan</i>	113
2.2.11.3	<i>Class ASCM.Distributor.PesananManager</i>	115
2.3	Realisasi Use Case.....	116
2.3.1	Use Case : Login.....	116
2.3.2	Use Case : Ubah Password.....	117
2.3.3	Use Case : Add User.....	118
2.3.4	Use Case : Edit User.....	119
2.3.5	Use Case : Delete User.....	120
2.3.6	Use Case : Display User.....	121
2.3.7	Use Case : Entri Schedule Produksi.....	122
2.3.8	Use Case : EntriReschedule.....	123
2.3.9	Use Case : Display Pesanan.....	124
2.3.10	Use Case : Display Schedule Produksi.....	125
2.3.11	Use Case : Display Reschedule Produksi.....	126
2.3.12	Use Case : Display Capacity Requirement Planning.....	127
2.3.13	Use Case : Display Material Requirement Planning.....	128
2.3.14	Use Case : Entri Purchase Order.....	129
2.3.15	Use Case : Display Purchase Order.....	130
2.3.16	Use Case : Surat Perintah Produksi.....	131
2.3.17	Use Case : Print SPP.....	132
2.3.18	Use Case : AddProduk.....	133
2.3.19	Use Case : EditProduk.....	134
2.3.20	Use Case : Delete Produk.....	135
2.3.21	Use Case : Display Produk.....	136
2.3.22	Use Case : Entri Hasil Produksi.....	137
2.3.23	Use Case : Edit Hasil Produksi.....	138
2.3.24	Use Case : Delete Hasil Produksi.....	139
2.3.25	Use Case : Display Hasil Produksi.....	140
2.3.26	Use Case : Laporan Hasil Produksi.....	141
2.3.27	Use Case : Print Laporan Hasil Produksi.....	142
2.3.28	Use Case : Add Mesin.....	143
2.3.29	Use Case : Edit Mesin.....	144
2.3.30	Use Case : Delete Mesin.....	145
2.3.31	Use Case : Edit Status Mesin.....	146
2.3.32	Use Case : Display SPP.....	147
2.3.33	Use Case : Add Bahan Produksi.....	148
2.3.34	Use Case : Edit Bahan Produksi.....	149
2.3.35	Use Case : Delete Bahan Produksi.....	150
2.3.36	Use Case : Display Bahan Produksi.....	151
2.3.37	Use Case : Entri Stock Keluar.....	152
2.3.38	Use Case : Entri Stock Masuk.....	153
2.3.39	Use Case : Add Kategori.....	154
2.3.40	Use Case : Edit Kategori.....	155
2.3.41	Use Case : Delete Kategori.....	156

2.3.42	Use Case : Laporan Bahan Baku Produksi.....	157
2.3.43	Use Case : Print Laporan Bahan Baku Produksi.....	158
2.3.44	Use Case : Laporan Bahan Baku Produksi Keluar.....	159
2.3.45	Use Case : Print Laporan Bahan Baku Produksi Keluar.....	160
2.3.46	Use Case : Edit Status Bahan Baku.....	161
2.3.47	Use Case : Edit Status Barang Produksi.....	162
2.3.48	Use Case : Add Penjualan.....	163
2.3.49	Use Case : Edit Penjualan.....	164
2.3.50	Use Case : Edit Harga Produk.....	165
2.3.51	Use Case : Delete Penjualan.....	166
2.3.52	Use Case : Display Penjualan.....	167
2.3.53	Use Case : Print Penjualan.....	168
2.3.54	Use Case : Add Supplier.....	169
2.3.55	Use Case : Edit Supplier.....	170
2.3.56	Use Case : Delete Supplier.....	171
2.3.57	Use Case : Display Supplier.....	172
2.3.58	Use Case : Edit Harga Bahan Baku.....	173
2.3.59	Use Case : Order Bahan Baku.....	174
2.3.60	Use Case : Laporan Pembelian.....	175
2.3.61	Use Case : Print Laporan Pembelian.....	176
2.3.62	Use Case : Display Purchase Order.....	177
2.3.63	Use Case : Print Purchase Order.....	178
2.3.64	Use Case : Entri Pesanan.....	179
2.3.65	Use Case : Display Produk.....	180
3	Deskripsi Perancangan Persistent Data.....	181
3.1	Basis Data.....	181
3.1.1	Data_Pengguna.....	182
3.1.2	Schedule_Master.....	182
3.1.3	Schedule_Detail.....	182
3.1.4	Reschedule_Master.....	183
3.1.5	Rechedule_Detail.....	183
3.1.6	SPP_Master.....	184
3.1.7	SPP_Detail.....	184
3.1.8	Bhn_Baku_Keluar.....	185
3.1.9	Produk.....	185
3.1.10	Barang_Hasil_Produksi.....	186
3.1.11	Data_Pesanan_Master.....	186
3.1.12	Pesanan_Detail.....	187
3.1.13	Distributor.....	187
3.1.14	Penjualan_Master.....	188
3.1.15	Penjualan_Detail.....	188
3.1.16	Bahan_Baku_Master.....	189
3.1.17	Kategori_Bahan_Baku.....	189
3.1.18	Permintaan_Pembelian_Master.....	190
3.1.19	Permintaan_Pembelian_Detail.....	190
3.1.20	Pembelian_Master.....	191
3.1.21	Pembelian_Detail.....	191
3.1.22	Penerimaan_Barang.....	192
3.1.23	Data_Supplier.....	192
3.1.24	Mesin.....	193
4	Deskripsi Perancangan Antarmuka.....	194
4.1	Use Case : Login.....	194
4.2	Use Case : Ubah Password.....	195
4.3	Use Case : Add User.....	197
4.4	Use Case : Display User.....	198

4.5	Use Case : Edit User.....	199
4.6	Use Case : Delete User.....	201
4.7	Use Case : Entri Schedule Produksi	202
4.8	Use Case : Entri Reschedule Produksi	204
4.9	Use Case : Display Pesanan.....	206
4.10	Use Case : Display Schedule Produksi.....	207
4.11	Use Case : Display Reschedule Produksi.....	208
4.12	Use Case : Display CRP (Capacity Requirement Planning).....	209
4.13	Use Case : Display MRP (Material Requirement Planning)	210
4.14	Use Case : Entri Purchase Order.....	211
4.15	Use Case : Display PO.....	213
4.16	Use Case : Surat Perintah Produksi.....	214
4.17	Use Case : Print SPP.....	216
4.18	Use Case : Add Produk.....	218
4.19	Use Case : Edit Produk.....	219
4.20	Use Case : Delete Produk.....	220
4.21	Use Case : Display Produk	221
4.22	Use Case : Entri Hasil Produksi	222
4.23	Use Case : Edit Hasil Produksi	224
4.24	Use Case : Delete Hasil Produksi	225
4.25	Use Case : Display Hasil Produksi	226
4.26	Use Case : Laporan Hasil Produksi	227
4.27	Use Case: Print Hasil Produksi	228
4.28	Use Case : Add Mesin.....	230
4.29	Use Case : Edit Mesin.....	231
4.30	Use Case : Delete Mesin.....	232
4.31	Use Case : Edit Status Mesin.....	233
4.32	Use Case : Display SPP.....	235
4.33	Use Case : Add Bahan Produksi.....	236
4.34	Use Case : Edit Bahan Baku.....	237
4.35	Use Case : Delete Bahan Baku.....	239
4.36	Use Case : Display Bahan Produksi	240
4.37	Use Case : Entri Stock Keluar.....	241
4.38	Use Case : Entri Stock Masuk.....	242
4.39	Use Case : Add Kategori.....	244
4.40	Use Case : Edit Kategori.....	245
4.41	Use Case : Delete Kategori.....	246
4.42	Use Case : Laporan Bahan Baku Produksi.....	248
4.43	Use Case : Print Laporan Bahan Baku Produksi.....	249
4.44	Use Case : Laporan Bahan Baku Produksi Keluar.....	250
4.45	Use Case : Print Laporan Bahan Baku Produksi Keluar	252
4.46	Use Case : Edit Status Bahan Baku.....	253
4.47	Use Case : Edit Status Barang Produksi.....	255
4.48	Use Case : Add Penjualan	256
4.49	Use Case : Edit Penjualan	258
4.50	Use Case : Edit Harga Produk	259
4.51	Use Case : Delete Penjualan	260
4.52	Use Case : Add Supplier.....	261
4.53	Use Case : Edit Supplier.....	262
4.54	Use Case : Delete Supplier.....	264
4.55	Use Case : Display Supplier.....	265
4.56	Use Case : Edit Harga Bahan Baku.....	266
4.57	Use Case : Order Bahan Baku.....	267
4.58	Use Case : Display Order Pembelian.....	269
4.59	Use Case : Laporan Pembelian	270

4.60	Use Case : Print Laporan Pembelian	271
4.61	Use Case : Display Purchase Order	273
4.62	Use Case : Print Purchase Order.....	274
4.63	Use Case : Entri Pesanan	275
4.64	Use Case : Display Produk	277



Daftar Gambar

Gambar 2.1 Deployment Diagram ASCaMaDa.....	16
Gambar 2.2 Package Dependencies ASCaMaDa.....	18
Gambar 2.3 Class Diagram Package ASCM.Administrator.....	19
Gambar 2.4 Class ASCM.Administrator.User	19
Gambar 2.15 Class ASCM.PPIC.EntriScheduleUI	31
Gambar 2.16 Class ASCM.PPIC.DsisplayScheduleUI	32
Gambar 2.17 Class ASCM.PPIC.Reshcedule.....	32
Gambar 2.18 Class ASCM.PPIC.RescheduleManager.....	35
Gambar 2.24 Class ASCM.PPIC.SPP	43
Gambar 2.43 Class Diagram Package ASCM.Produksi.....	51
Gambar 2.46 Class Diagram Package ASCM.Warehouse	67
Gambar 2.46 Class Diagram Package ASCM.QualityAssurance.....	83
Gambar 2.46 Class Diagram Package ASCM.Marketing	88
Gambar 2.46 Class Diagram Package ASCM.Purchasing.....	97
Gambar 2.46 Class Diagram Package ASCM.Supplier.....	112
Gambar 2.46 Class Diagram Package ASCM.Distributor	113
Gambar 2.49 Design Sequence Diagram Use Case Login	116
Gambar 2.50 Design Sequence Diagram Use Case Ubah Password.....	117
Gambar 2.51 Design Sequence Diagram Use Case Add User.....	118
Gambar 2.52 Design Sequence Diagram Use Case Edit User	119
Gambar 2.53 Design Sequence Diagram Use Case Delet User.....	120
Gambar 2.54 Design Sequence Diagram Use Case Display User.....	121
Gambar 2.55 Design Sequence Diagram Use Case Entri Schedule Produksi.....	122
Gambar 2.56 Design Sequence Diagram Use Case Entri Reschedule.....	123
Gambar 2.57 Design Sequence Diagram Use Case Display Pesanan	124
Gambar 2.58 Design Sequence Diagram Use Case Display Schedule Produksi	125
Gambar 2.59 Design Sequence Diagram Use Case Display Reschedule Produksi	126
Gambar 2.60 Design Sequence Diagram Display Use Case CRP.....	127
Gambar 2.61 Design Sequence Diagram Use Case Display MRP	128
Gambar 2.62 Design Sequence Diagram Use Case Entri Purchase Order	129
Gambar 2.63 Design Sequence Diagram Use Case Display Puchase Order	130
Gambar 2.64 Design Sequence Diagram Use Case Surat Perintah Produksi	131
Gambar 2.65 Design Sequence Diagram Use Case Print SPP	132
Gambar 2.66 Design Sequence Diagram Use Case Add Produk	133
Gambar 2.67 Design Sequence Diagram Use Case Edit Produk	134
Gambar 2.68 Design Sequence Diagram Use Case Delete Produk	135
Gambar 2.69 Design Sequence Diagram Use Case Display Produk.....	136
Gambar 2.70 Design Sequence Diagram Use Case Entri Hasil Produksi.....	137
Gambar 2.71 Design Sequence Diagram Use Case Edit Hasil Produksi.....	138
Gambar 2.72 Design Sequence Diagram Use Case Delete Hasil Produksi.....	139
Gambar 2.73 Design Sequence Diagram Use Case Display Hasil Produksi	140
Gambar 2.74 Design Sequence Diagram Use Case Laporan Hasil Produksi.....	141
Gambar 2.75 Design Sequence Diagram Use Case Print Laporan Hasil Produksi.....	142
Gambar 2.76 Design Sequence Diagram Use Case Add Mesin	143
Gambar 2.77 Design Sequence Diagram Use Case Edit Mesin	144
Gambar 2.78 Design Sequence Diagram Use Case Delete Mesin	145
Gambar 2.79 Design Sequence Diagram Use Case Edit Status Mesin	146
Gambar 2.80 Design Sequence Diagram Use Case Display SPP	147
Gambar 2.81 Design Sequence Diagram Use Case Add Bahan Produksi	148
Gambar 2.82 Design Sequence Diagram Use Case Edit Bahan Produksi	149
Gambar 2.83 Design Sequence Diagram Use Case Delete Bahan Produksi	150
Gambar 2.84 Design Sequence Diagram Use Case Display Bahan Produksi	151

Program Studi Teknik Informatika	DPPL-ASCaMaDa	10/ 277
----------------------------------	---------------	---------

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika

Gambar 2.85 Design Sequence Diagram Use Case Entri Stock Keluar	152
Gambar 2.86 Design Sequence Diagram Use Case Entri Stock Masuk	153
Gambar 2.87 Design Sequence Diagram Use Case Add Kategori	154
Gambar 2.88 Design Sequence Diagram Use Case Edit Kategori	155
Gambar 2.89 Design Sequence Diagram Use Case Delete Kategori	156
Gambar 2.90 Design Sequence Diagram Use Case Laporan Bahan Baku Produksi	157
Gambar 2.91 Design Sequence Diagram Use Case Print Lap Bahan Baku Produksi ...	158
Gambar 2.92 Design Sequence Diagram Use Case Lap Bhn Baku Produksi Keluar	159
Gambar 2.93 Design Sequence Diagram Use Case Print Lap Bhn Baku Prod Keluar ..	160
Gambar 2.94 Design Sequence Diagram Use Case Edit Status Bahan Baku	161
Gambar 2.95 Design Sequence Diagram Use Case Edit Status Barang Produksi	162
Gambar 2.96 Design Sequence Diagram Use Case Add Penjualan.....	163
Gambar 2.97 Design Sequence Diagram Use Case Edit Penjualan.....	164
Gambar 2.98 Design Sequence Diagram Use Case Edit Harga Produk.....	165
Gambar 2.99 Design Sequence Diagram Use Case Delete Penjualan.....	166
Gambar 2.100 Design Sequence Diagram Use Case Display Penjualan	167
Gambar 2.101 Design Sequence Diagram Use Case Print Penjualan.....	168
Gambar 2.102 Design Sequence Diagram Use Case Add Supplier	169
Gambar 2.103 Design Sequence Diagram Use Case Edit Supplier.....	170
Gambar 2.104 Design Sequence Diagram Use Case Delete Supplier	171
Gambar 2.105 Design Sequence Diagram Use Case Display Supplier.....	172
Gambar 2.106 Design Sequence Diagram Use Case Edit Harga Bahan Baku.....	173
Gambar 2.107 Design Sequence Diagram Use Case Order Bahan Baku	174
Gambar 2.108 Design Sequence Diagram Use Case Laporan Pembelian.....	175
Gambar 2.109 Design Sequence Diagram Use Case Print Laporan Pembelian.....	176
Gambar 2.110 Design Sequence Diagram Use Case Display Purchase Order.....	177
Gambar 2.111 Design Sequence Diagram Use Case Print Purchase Order	178
Gambar 2.112 Design Sequence Diagram Use Case Entri Pesanan	179
Gambar 2.113 Design Sequence Diagram Use Case Display Produk.....	180
Gambar 3.1 Database Diagram ASCaMaDa.....	181
Gambar 4.1 Rancangan Halaman Login	194
Gambar 4.2 Rancangan Halaman Ubah Password	195
Gambar 4.3 Rancangan Add User	197
Gambar 4.4 Rancangan Halaman Display User	198
Gambar 4.5 Rancangan Halaman Editing User	199
Gambar 4.6 Rancangan Halaman Delete User.....	201
Gambar 4.7 Rancangan Halaman Entri Schedule Produksi.....	202
Gambar 4.8 Rancangan Halaman Entri Reschedule Produksi.....	204
Gambar 4.9 Rancangan Halaman Display Pesanan.....	206
Gambar 4.10 Rancangan Halaman Display ScheduleProduksi	207
Gambar 4.11 Rancangan Halaman Display Reshcedule Produksi	208
Gambar 4.12 Rancangan Halaman Display CRP	209
Gambar 4.13 Rancangan Halaman Display MRP	210
Gambar 4.14 Rancangan Halaman Entri Purchase Order	211
Gambar 4.15 Rancangan Halaman Display PO.....	213
Gambar 4.16 Rancangan Halaman Surat Perintah Produksi.....	214
Gambar 4.17 Rancangan Halaman Print SPP	216
Gambar 4.18 Rancangan Halaman Add Produk.....	218
Gambar 4.19 Rancangan Halaman Edit Produk	219
Gambar 4.20 Rancangan Halaman Delete Produk	220
Gambar 4.21 Rancangan Halaman Display Produk.....	221
Gambar 4.22 Rancangan Halaman Entri Hasil Produksi.....	222
Gambar 4.23 Rancangan Halaman Editing Hasil Produksi	224
Gambar 4.24 Rancangan Halaman Delete Hasil Produksi	225
Gambar 4.25 Rancangan Halaman Display Hasil Produksi	226

Gambar 4.26 Rancangan Halaman Laporan Hasil Produksi.....	227
Gambar 4.27 Rancangan Halaman Print Hasil Produksi.....	228
Gambar 4.28 Rancangan Halaman Add Mesin.....	230
Gambar 4.29 Rancangan Halaman Edit Mesin.....	231
Gambar 4.30 Rancangan Halaman Delet Mesin.....	232
Gambar 4.31 Rancangan Halaman Edit Status Mesin.....	233
Gambar 4.32 Rancangan Halaman Display SPP.....	235
Gambar 4.33 Rancangan Halaman Add Bahan Produksi.....	236
Gambar 4.34 Rancangan Halaman Edit Bahan Baku.....	237
Gambar 4.35 Rancangan Halaman Delete Bahan Baku.....	239
Gambar 4.36 Rancangan Halaman Display Bahan Produksi.....	240
Gambar 4.37 Rancangan Halaman Entri Stock Keluar.....	241
Gambar 4.38 Rancangan Halaman Entri Stock Masuk.....	242
Gambar 4.39 Rancangan Halaman Add Kategori.....	244
Gambar 4.40 Rancangan Halaman Edit Kategori.....	245
Gambar 4.41 Rancangan Halaman Delete Kategori.....	246
Gambar 4.42 Rancangan Halaman Laporan Bahan Baku Produksi.....	248
Gambar 4.43 Rancangan Halaman Print Laporan Bahan Baku Produksi.....	249
Gambar 4.44 Rancangan Halaman Laporan Bahan Baku Keluar.....	250
Gambar 4.45 Rancangan Halaman Laporan Bahan Baku Keluar.....	252
Gambar 4.46 Rancangan Halaman Edit Status Bahan Baku.....	253
Gambar 4.47 Rancangan Halaman Edit Status Barang Produksi.....	255
Gambar 4.48 Rancangan Halaman Add Penjualan.....	256
Gambar 4.49 Rancangan Halaman Edit Penjualan.....	258
Gambar 4.50 Rancangan Halaman Edit Harga Produk.....	259
Gambar 4.5 Rancangan Halaman Editing User.....	260
Gambar 4.52 Rancangan Halaman Add Supplier.....	261
Gambar 4.53 Rancangan Halaman Pengelolaan Kategori.....	262
Gambar 4.54 Rancangan Halaman Delete Supplier.....	264
Gambar 4.45 Rancangan Halaman Display Supplier.....	265
Gambar 4.56 Rancangan Halaman Edit Harga Bahan Baku.....	266
Gambar 4.57 Rancangan Halaman Order Bahan Produksi.....	267
Gambar 4.58 Rancangan Halaman Display Order Pembelian.....	269
Gambar 4.59 Rancangan Halaman Laporan Pembelian.....	270
Gambar 4.60 Rancangan Halaman Print Laporan Pembelian.....	271
Gambar 4.61 Rancangan Halaman Display Purchase Order.....	273
Gambar 4.62 Rancangan Halaman Print Purchase Order.....	274
Gambar 4.63 Rancangan Halaman Entri Pesanan.....	275
Gambar 4.64 Rancangan Halaman Display Produk.....	277

Daftar Tabel

Tabel 1.1 Tabel Istilah dan Singkatan	14
Tabel 4.1 Tabel Data Pengguna	182
Tabel 4.2 Tabel Data Schedule Master	182
Tabel 4.3 Tabel Data Schedule Detail	182
Tabel 4.4 Tabel Data Reschedule Master	183
Tabel 4.5 Tabel Data Reschedule Detail	183
Tabel 4.6 Tabel Data SPP Master	184
Tabel 4.7 Tabel Data SPP Detail	184
Tabel 4.8 Tabel Data Bahan Baku Keluar	185
Tabel 4.9 Tabel Data Produk	185
Tabel 4.10 Tabel Barang Hasil Produksi	186
Tabel 4.11 Tabel Data Pesanan Master	186
Tabel 4.12 Tabel Pesanan Detail	187
Tabel 4.13 Tabel Distributor	187
Tabel 4.14 Tabel Penjualan Master	188
Tabel 4.15 Tabel Penjualan Detail	188
Tabel 4.16 Tabel Bahan Baku Master	189
Tabel 4.17 Tabel Kategori Bahan Baku	189
Tabel 4.18 Tabel Permintaan Pembelian Master	190
Tabel 4.19 Tabel Permintaan Pembelian Detail	190
Tabel 4.20 Tabel Pembelian Master	191
Tabel 4.21 Tabel Pembelian Detail	191
Tabel 4.22 Tabel Penerimaan Barang	192
Tabel 4.23 Tabel Data Supplier	192
Tabel 4.24 Tabel Mesin	193

1 Pendahuluan

1.1 Tujuan

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) ini dibuat untuk menyediakan deskripsi lengkap mengenai perancangan perangkat lunak ASCaMaDa yang dibuat untuk PT. Sari Husada. Dokumen ini khususnya ditujukan untuk pengembang perangkat lunak sebagai acuan untuk implementasi pada tahap selanjutnya dan orang-orang lain yang tertarik untuk mengembangkan perangkat lunak ini lebih lanjut.

1.2 Lingkup Dokumen

Dokumen DPPL ini menyediakan deskripsi lengkap perancangan perangkat lunak untuk ASCaMaDa. Perancangan ini merupakan arsitektur sistem yang dijelaskan melalui perancangan class/modul, detail operasi apa yang akan dilakukan oleh masing-masing class/modul, dan layout database. Metodologi yang digunakan dalam perancangan adalah metode USDP (Unified Software Development Process) dari Rational Software.

1.3 Istilah dan Singkatan

Daftar istilah dan singkatan yang digunakan dalam dokumen ini adalah:

Tabel 1.1 Tabel Istilah dan Singkatan

Keyword/Phrase	Definisi
DPPL	Deskripsi Perancangan Perangkat Lunak disebut juga <i>Software Design Description (SDD)</i> merupakan deskripsi dari perancangan produk/perangkat lunak yang akan dikembangkan.
ASCaMaDa	ASCaMaDa merupakan perangkat lunak yang dibangun untuk menghubungkan Chanel pada Supply Chain Management

	yang telah diterapkan pada PT. Sari Husada melalui internet.
--	--

1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

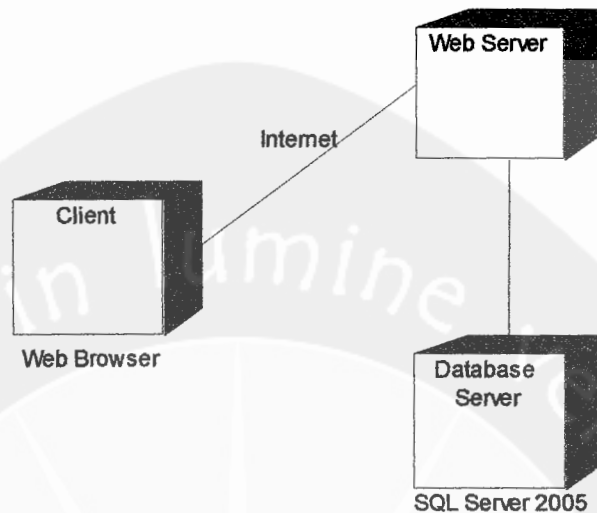
1. A. Suhendar, Hariman Gunadi. *Visual Modeling Menggunakan UML dan RATIONAL ROSE*. Informatika Bandung. 2002.
2. Fowler, Martin. *UML Distilled 3th Ed., Panduan Singkat Bahasa Pemodelan Objek Standar*. ANDI Yogyakarta. 2005.
3. Larman, Craig. *Applying UML and patterns : an introduction to object-oriented analysis and design and the Unified Process*. Prentice-Hall, Inc. 2002.

1.5 Deskripsi umum (Overview)

Dokumen DPPL ini terdiri dari 4 bab. Bab pertama adalah **Pendahuluan**, yang berisi deskripsi dokumen. Bab kedua adalah **Deskripsi Perancangan Arsitektural**, yang berisi deskripsi arsitektur sistem. Bab ketiga adalah **Deskripsi Perancangan Persistent Data**, yang berisi deskripsi data-data yang akan disimpan pada *persistent storage*. Bab keempat adalah **Deskripsi Perancangan Antarmuka**, yang berisi deskripsi rancangan GUI yang digunakan sistem untuk berinteraksi dengan user.

2 Deskripsi Perancangan Arsitektural

2.1 Deployment Diagram



Gambar 2.1 Deployment Diagram ASCaMaDa

Deployment diagram ini dibuat untuk menunjukkan semua node pada sistem, hubungan di antara mereka, dan proses yang akan dijalankan di masing-masing node.

2.1.1 Node : Client

Client merupakan komputer yang digunakan oleh user untuk melakukan pengelolaan terhadap dokumen. Proses yang ada didalamnya adalah:

1. Web browser, digunakan untuk melakukan permintaan proses yang dijalankan pada aplikasi ASCaMaDa.

2.1.2 Node : Web Server

Web Server merupakan komputer yang menyediakan layanan ke web bagi *client* yang mengakses Internet, dimana Web server ini akan mengakses basis data untuk operasi yang melibatkan data-data pada basis data.

Program Studi Teknik Informatika	DPPL-ASCaMaDa	16/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2.1.3 Node : Database Server

Database Server merupakan komputer yang dipakai untuk menyediakan koneksi ke basis data dan mengautentikasi Web server dan tidak akan mengizinkannya melihat informasi atau menjalankan prosedur kecuali jika ia memiliki hak yang sesuai.

2.2 Design Class

2.2.1 Pengantar

Nama class yang digunakan dalam design class adalah nama class yang valid, termasuk nama packagenya. Untuk class-class yang berasal dari framework .NET juga digunakan nama class dengan package lengkap. Untuk penjelasan tipe data yang utuh dapat dilihat pada bagian deskripsi class, sedangkan gambar design class tidak akan menggunakan nama package yang lengkap.

Stereotype yang digunakan dalam design class adalah :

a. << boundary >>

Boundary class merupakan class yang berfungsi untuk menghubungkan sistem dengan user di luar sistem.

b. << control >>

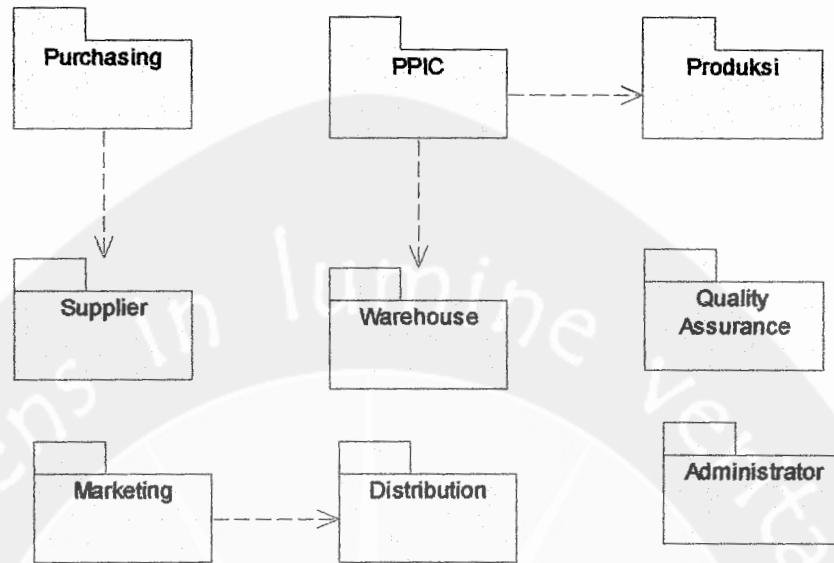
Control class adalah suatu class yang objeknya melakukan interaksi antar sekelompok objek lain. Control class biasanya memiliki karakteristik yang spesifik untuk satu use case, dan objek class ini biasanya hanya aktif pada realisasi use case.

c. << entity >>

Entity class adalah class yang bersifat pasif, dalam arti class tersebut tidak memulai interaksi dengan class lain. Entity class ini biasanya merepresentasikan suatu objek yang disimpan dalam persistent storage.

Program Studi Teknik Informatika	DPPL-ASCaMaDa	17/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2.2.2 Package Dependencies

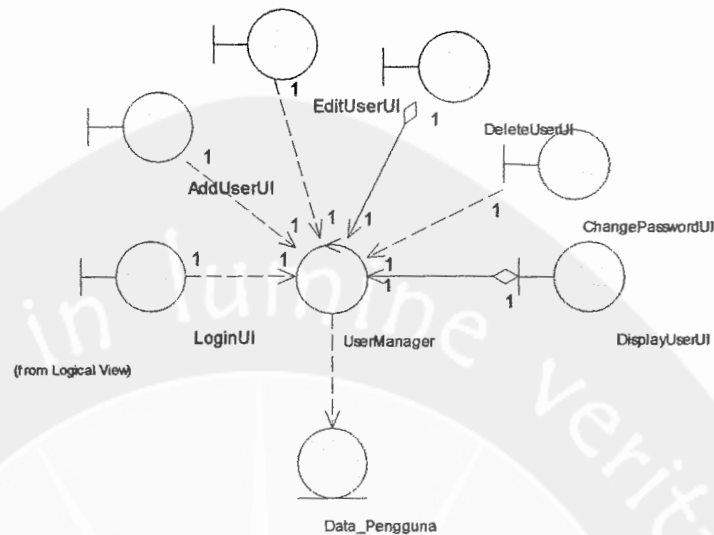


Gambar 2.2 Package Dependencies ASCaMaDa

2.2.3 Package Data Administrator (ASCM.Administrator)

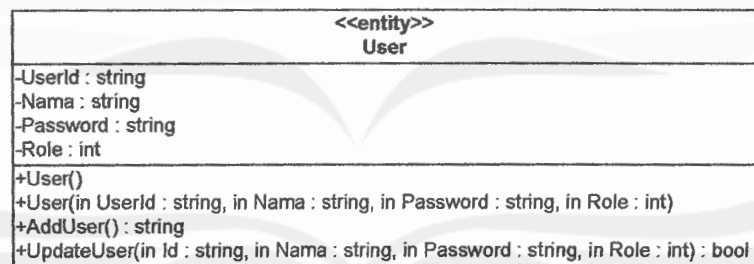
Package ini menyediakan class-class yang diperlukan untuk melakukan pengelolaan yang berhubungan dengan data pengguna/user.

2.2.3.1 Class diagram package ASCM.Administrator



Gambar 2.3 Class Diagram Package ASCM.Administrator

2.2.3.2 Class ASCM.Administrator.User



Gambar 2.4 Class ASCM.Administrator.User

Deskripsi

Class ini merepresentasikan user/pengguna sistem ASCaMaDa.

Atribut

- - UserId : string

Merepresentasikan username pengguna.

- - Nama : string

Merepresentasikan nama pengguna.

- - Password : string

Merepresentasikan password pengguna.

- - Role : integer

Merepresentasikan role pengguna.

Method

- + User()

Konstruktor class ASCM.Administrator.User. Buat instance baru ASCM.Administrator.User tanpa atribut terdefinisi.

- + User(UserId : string, Nama : string, Password : string, Role : integer)

Konstruktor class ASCM.Administrator.User. Buat instance baru ASCM.Administrator.User. dengan atribut terdefinisi.

- + AddUser() : string

Digunakan untuk menambahkan user baru.

Returns:

"Sukses" jika User berhasil ditambahkan.

"Error" jika User tidak berhasil ditambahkan.

- + UpdateUser(Id : string, nama : string, password : string, role :int) : bool

Digunakan untuk mengubah data pada tabel Data_Pengguna.

Parameters :

Id - Merepresentasikan username pengguna.

nama - Merepresentasikan nama pengguna.

password - Merepresentasikan password pengguna.

role - Merepresentasikan role pengguna.

Returns:

true jika tabel yang bernama Data_Pengguna berhasil diubah.

2.2.3.3 Class ASCM.Administrator.UserManager

<<control>> userManager
<pre> +userManager() +AddUserAccount(in id : string, in nama : string, in password : string, in role : int) : string +UpdateUser(in id : string, in nama : string, in password : string, in role : int) : bool +GetDataUser(in Kode : string) : <<entity>> User +GetRole(in Kode : string) : int +ValidateKode(in Kode : string) : bool +ValidateLoginId(in Kode : string, in Pass : string) : bool +ValidatePassword(in Pass : string) : bool </pre>

Gambar 2.5 Class ASCM.Administrator. UserManager

Deskripsi

Class yang berperan sebagai control class untuk aksi yang berhubungan dengan pengelolaan data user/pengguna pada ASCaMaDa.

Atribut

-

Method

- + userManager()

Konstruktor class ASCM.Administrator.User. Buat instance baru ASCM.Administrator.User.

- + AddUserAccount(id : string, nama : string, password : string , role : int) : string

Digunakan untuk menambahkan data user baru ke database.

Parameters :

id – merepresentasikan username.

nama – merepresentasikan nama user.

password – merepresentasikan password.

role – merepresentasikan role user.

Returns:

“Sukses” jika User berhasil ditambahkan.

“Error” jika User tidak berhasil ditambahkan.

- + UpdateUser(Id : string, nama : string, password : string, role :int) : bool

Digunakan untuk mengubah data pada tabel

Data_Pengguna. **Parameters :**

Id - Merepresentasikan username pengguna.

nama - Merepresentasikan nama pengguna.

password - Merepresentasikan password pengguna.

role - Merepresentasikan role pengguna.

Returns:

true jika tabel yang bernama Data_Pengguna berhasil diubah.

- + ValidateLoginID(Kode : string, Pass : string) : bool

Validasi username dan password apakah sudah ada dalam basis data.

Parameters :

Kode – username yang akan divalidasi.

Pass – password yang akan divalidasi.

Returns:

true jika dan hanya jika username dan password ada dalam database dan valid.

- + ValidateKode(kode : string) : bool

Validasi username apakah ada dalam basis data.

Parameters :

Kode – username yang akan divalidasi.

Returns:

true jika dan hanya jika username yang diinputkan ada dalam database.

- + ValidatePassword(Pass : string) : bool

Validasi password apakah sudah ada dalam basis data.

Parameters :

Pass – password yang akan divalidasi.

Returns:

true jika dan hanya jika password ada dalam database.

- + GetRole(kode : string) : int

Memperoleh data role yang terdapat dalam database.

Parameters :

kode – username yang akan diambil role-nya.

Returns:

true jika dan hanya jika role ada dalam database.

- + GetDataUser(kode : string) :
ASCM.Administrator.User

Memperoleh class ASCM.Administrator.User untuk username tertentu.

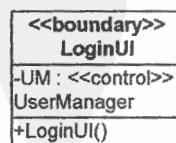
Parameters :

kode – username yang akan diambil datanya.

Returns:

ASCM.Administrator.User untuk username kode, atau null jika username tidak ditemukan.

2.2.3.4 Class ASCM.Administrator.LoginUI



Gambar 2.6 Class ASCM.Administrator.LoginUI

Deskripsi

Class ini merupakan GUI yang digunakan untuk validasi username dan password untuk login (SKPL-ASCaMaDa.UC-01).

Atribut

- - UM : ASCM.Administrator.UserManager
Merupakan control class untuk melakukan validasi username dan password.

Method

- + LoginUI()
Buat instance baru ASCM.Administrator.LoginUI

2.2.3.5

Class ASCM.Administrator.AddUserUI

<<boundary>> AddUserUI
-UM : <<control>> UserManager
+AddUserUI()

Gambar 2.7 Class ASCM.Administrator.AddUserUI

Deskripsi

Class ini merupakan GUI yang digunakan untuk menambah data user baru ke database (SKPL- ASCaMaDa.UC-02).

Atribut

- - UM : ASCM.Administrator.UserManager
Merupakan control class untuk melakukan pengelolaan data user.

Method

- + AddUserUI()
Buat instance baru ASCM.Administrator.AddUserUI

Returns:

“sukses” jika data berhasil ditambahkan ke database.

- + AddRescheduleDetail(kodeRescheduleMaster : int, kodeRescheduleDetail : int, Produk : int, jml : int) : string

Menambah data reschedule pada tabel reschedule detail.

Parameters :

- kodeRescheduleMaster : int

Merepresentasikan kode reschedule master.

- kodeRescheduleDetail : int

Merepresentasikan kode reschedule detail.

- Produk : int

Merepresentasikan kode produk.

- jml : int

Merepresentasikan jumlah produk.

Returns:

“sukses” jika data berhasil ditambahkan ke database.

- + GenerateIdRescheduleMaster() : int

Membuat id reschedule master secara otomatis

Parameters :

-

Returns:

Nilai Id reschedule master unik.

- + GenerateIdRescheduleDetail() : bool

Membuat id reschedule detail secara otomatis.

Parameters :

-

Returns:

Nilai Id reschedule detail unik.

- + UpdateScheduleMaster() : bool

Melakukan update status schedule jika terjadi reschedule.

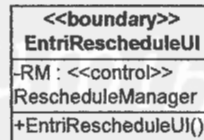
Parameters :

-

Returns:

True jika data berhasil diupdate.

2.2.4.8 Class ASCM.PPIC.EntriRescheduleUI



Gambar 2.19 Class ASCM.PPIC.EntriRescheduleUI

Deskripsi

Class yang berperan sebagai boundary class/GUI untuk proses pembuatan data reschedule produksi pada ASCaMaDa.

Atribut

- - RM : RecheduleManager

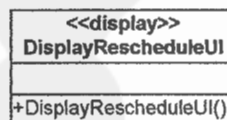
Merepresentasikan object control class untuk entri reschedule produksi .

Method

- + EntriRescheduleUI()

Konstruktor class ASCM.PPIC.EntriRescheduleUI. Buat instance baru tanpa atribut terdefinisi.

2.2.4.9 Class ASCM.PPIC.DisplayRescheduleUI



Gambar 2.20 Class ASCM.PPIC.DisplayRescheduleUI

Deskripsi

Class yang berperan sebagai boundary class/GUI untuk menampilkan data Reschedule produksi.

Atribut

Program Studi Teknik Informatika	DPPL-ASCaMaDa	37/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Method

- + DisplayRescheduleUI()

Konstruktor class ASCM.PPIC.DisplayRescheduleUI. **Buat** instance baru tanpa atribut terdefinisi.

2.2.4.10 Class ASCM.PPIC.PurchaseOrder

<<entity>> PurchaseOrder
-KodePPMaster : int -KodePPDetail : int -Tglkirim : string -Tgl datang : string -BahanProduksi : string -Jml : int
+PurchaseOrder() +PurchaseOrder(in kodePPMaster : int, in kodePPDetail : int, in tglkirim : string, in tgl datang : string, in bahanProduksi : string, in jml : int) +AddPPMaster() +AddPPDetail() +GetLastidPPMaster() : int +GetLastidPPDetail() : int

Gambar 2.21 Class ASCM.PPIC.PurchaseOrder

Deskripsi

Class ini merepresentasikan data-data purchase order.

Atribut

- - KodePPMaster : int
Merepresentasikan kode purchase order master.
- - KodePPDetail : int
Merepresentasikan kode purchase order detail.
- - Tglkirim : string
Merepresentasikan tanggal pengiriman bahan baku.
- - Tgl datang : string
Merepresentasikan tanggal dating bahan baku.
- - BahanBaku : string
Merepresentasikan kode bahan baku produksi.
- - Jml : int
Merepresentasikan jumlah bahan baku.

Method

- + PurchaseOrder()

Konstruktor class ASCM.PPIC.PurchaseOrder. **Buat instance baru tanpa atribut terdefinisi.**

- + PurchaseOrder(kodePPMaster : int, kodePPDetail : int, tglKirim : string, tglDatang : int, bahanProduksi : string, jml : int)

Konstruktor class ASCM.PPIC.PurchaseOrder. **Buat instance baru dengan atribut terdefinisi.**

- + AddPPMaster()

Menambah data permintaan pembelian pada tabel permintaan_pembelian_master.

- + AddPPDetail()

Menambah data permintaan pembelian pada tabel permintaan_pembelian_detail.

- + GetLastIdPPMaster()

Ambil Id permintaan pembelian master yang terakhir dari database.

Parameters :

-

Returns:

Nilai id PP master.

- + GetLastIdPPDetail()

Ambil Id permintaan pembelian detail yang terakhir dari database.

Parameters :

-

Returns:

Nilai Id PP detail.

2.2.4.11 Class ASCM.PPIC.PurchaseOrderManager

<<control>> PurchaseOrderManager
+PurchaseOrderManager() +AddPPMaster(in kodePPMaster : int, in tglKirim : string, in tglDatang : string) : string +AddPPDetail(in kodePPDetail : int, in kodeBarang : string, in kodePPMaster : int, in jml : int) : string +GenerateIdPPMaster() : int +GenerateIdPPDetail() : int

Gambar 2.22 Class ASCM.PPIC.PurchaseOrderManager

Deskripsi

Class yang berperan sebagai control class untuk aksi yang berhubungan dengan pengelolaan data Purchase Order pada ASCaMaDa.

Atribut

-

Method

- + PurchaseOrderManager ()

Konstruktor class ASCM.PPIC.PurchaseOrderManager.

Buat instance baru tanpa atribut terdefinisi.

- + AddPPMaster(kodePPMaster : int, tglKirim: string, tglDatang : string) : string

Menambah data permintaan pembelian pada tabel permintaan pembelian master.

Parameters :

- kodePPMaster : int

Merepresentasikan kode permintaan pembelian master.

- tglKirim : string

Merepresentasikan tanggal kirim bahan baku.

- tglDatang : string

Merepresentasikan tanggal datang bahan baku.

Returns:

“sukses” jika data berhasil ditambahkan ke database.

- + AddPPDetail(kodePPDetail : int, kodeBarang : string, kodePPMaster : int, jml : int) : string

Menambah data permintaan pembelian detail pada tabel permintaan pembelian detail.

Parameters :

- kodePPDetail : int

Merepresentasikan kode permintaan pembelian detail.

- kodeBarang : string

Merepresentasikan kode barang.

- kodePPMaster : int

Merepresentasikan kode permintaan pembelian master.

- jml : int

Merepresentasikan jumlah barang.

Returns:

“sukses” jika data berhasil ditambahkan ke database.

- + GenerateIdPPMaster() : int

Membuat id permintaan pembelian master secara otomatis

Parameters :

-

Returns:

Nilai Id PP master unik.

- + GenerateIdPPDetail() : int

Membuat id permintaan pembelian detail secara otomatis.

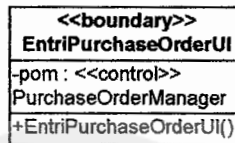
Parameters :

-

Returns:

Nilai Id permintaan pembelian detail unik.

2.2.4.12 Class ASCM.PPIC.EntriPurchaseOrderUI



Gambar 2.23 Class ASCM.PPIC.EntriPurchaseOrderUI

Deskripsi

Class yang berperan sebagai boundary class/GUI untuk proses entri data permintaan pembelian.

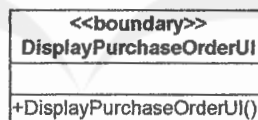
Atribut

- - pom : PurchaseOrderManager
Merepresentasikan object control class untuk pengelolaan Permintaan Pembelian.

Method

- + EntriPurchaseOrderManager()
Konstruktor class ASCM.PPIC.EntriPurchaseOrderUI.
Buat instance baru tanpa atribut terdefinisi.

2.2.4.13 Class ASCM.PPIC.DisplayPurchaseOrderUI



Gambar 2.23 Class ASCM.PPIC.DisplayPurchaseOrderUI

Deskripsi

Class yang berperan sebagai boundary class/GUI untuk proses menampilkan data permintaan pembelian

Atribut

-

Method

- + DisplayPurchaseOrderManager()

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk membuat Surat Perintah Produksi.

Atribut

- - sppm : SPPManager

Merepresentasikan object control class untuk pengelolaan SPP.

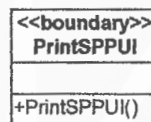
Method

- + SuratPerintahProduksiUI()

Konstruktor class ASCM.PPIC.SuratPerintahProduksiUI.

Buat instance baru tanpa atribut terdefinisi.

2.2.4.17 Class ASCM.PPIC.PrintSPPUI



Gambar 2.27 Class ASCM.PPIC.PrintSPPUI

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk menampilkan dan mencetak Surat Peintah Produksi.

Atribut

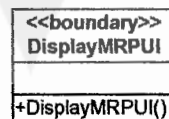
-

Method

- + PrintSPPUI()

Konstruktor class ASCM.PPIC.PrintSPPUI. Buat instance baru tanpa atribut terdefinisi.

2.2.4.18 Class ASCM.PPIC.DisplayMRPUI



Gambar 2.27 Class ASCM.PPIC.DiplayMRPUI

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk menampilkan Material Requirement Planning.

Atribut

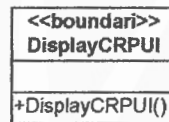
-

Method

- + DisplayMRPUI ()

Konstruktor class ASCM.PPIC.DisplayMRPUI. Buat instance baru tanpa atribut terdefinisi.

2.2.4.19 Class ASCM.PPIC.DisplayCRPUI



Gambar 2.27 Class ASCM.PPIC.DisplayCRPUI

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk menampilkan Capacity Requirement Planning.

Atribut

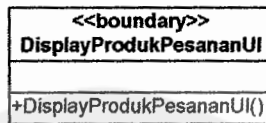
-

Method

- + DisplayCRPUI ()

Konstruktor class ASCM.PPIC.DisplayCRPUI. Buat instance baru tanpa atribut terdefinisi.

2.2.4.20 Class ASCM.PPIC.DisplayProdukPesanan



Gambar 2.27 Class ASCM.PPIC.DisplayProdukPesananUI

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk menampilkan pesanan distributor.

Atribut

-

Method

- + DisplayProdukPesananUI ()

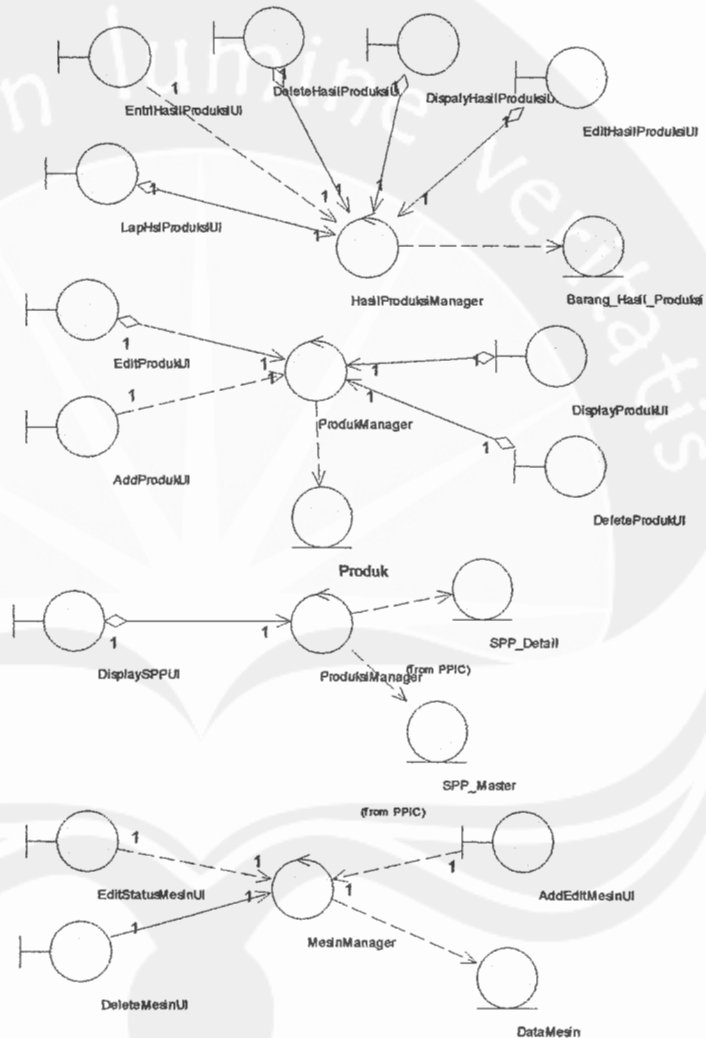
Konstruktor class ASCM.PPIC.DisplayProdukPesananUI.

Buat instance baru tanpa atribut terdefinisi.

2.2.5 Package Produksi (ASCM.Produksi)

Package ini menyediakan class-class yang diperlukan untuk melakukan pengelolaan yang berhubungan dengan data produksi.

2.2.5.1 Class diagram package ASCM.Produksi



Gambar 2.43 Class Diagram Package ASCM.Produksi

2.2.5.2 Class ASCM.Produksi.Produk

<<entity>> Produk
-KodeProduk : int -NamaProduk : string -Keterangan : string
+Produk() +Produk(in kodeproduk : int, in namaproduk : string, in keterangan : string) +AddProduk() : string +DeleteUpdateProduk() : bool +GetLastIdProduk() : int

Gambar 2. 39 Class ASCM.Produksi.Produk

Deskripsi

Class ini merepresentasikan data Produk pada sistem ASCaMaDa.

Atribut

- - KodeProduk : int
Merepresentasikan id produk
- - NamaProduk : string
Merepresentasikan nama produk.
- - Keterangan : string
Merepresentasikan keterangan produk.

Method

- + Produk()
Konstruktor class ASCM.Produksi.Produk. Buat instance baru tanpa atribut terdefinisi.
- + Produk(kodeProduk : int, namaproduk : string, keterangan : string)
Konstruktor class ASCM.Produksi.Produk. Buat instance baru dengan atribut terdefinisi.
- + AddProduk() : String
Menambahkan data produk baru ke basis data.

Parameters :

-

Returns:

Program Studi Teknik Informatika	DPPL-ASCaMaDa	52/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

“sukses” jika data produk baru berhasil ditambahkan.

- + DeleteUpdateProduk() : bool

Menghapus data produk pada database.

Parameters :

-

Returns:

“true” jika data berhasil dihapus

- + GetLastIdProduk() : int

Mengambil id produk yang terakhir pada database.

Parameters :

-

Returns:

Nilai id produk yang terakhir.

2.2.5.3 Class ASCM.Produksi.ProdukManager

<<control>> ProdukManager
+ProdukManager() +AddProduk(in kodeProduk : int, in namaProduk : string, in ket : string) : string +DeleteUpdateProduk(in kode : int) : bool +GenerateIdProduk() : int

Gambar 2. 40 Class ASCM.Produksi.ProdukManager

Deskripsi

Class ini merepresentasikan control class untuk data produk pada sistem ASCaMaDa.

Atribut

-

Method

- + ProdukManager()

Konstruktor class ASCM.Produksi.ProdukManager. Buat instance baru tanpa atribut terdefinisi.

Program Studi Teknik Informatika	DPPL-ASCaMaDa	53/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- + AddProduk(kodeproduk : int, namaproduk : String, keterangan : String) : String

Menambahkan data produk baru ke basis data.

Parameters : -

kodeproduk – id produk

namaproduk – nama produk

keterangan – keterangan

Returns:

“sukses” jika data produk baru berhasil ditambahkan.

- + DeleteUpdateProduk(kode : int) : Boolean

Menghapus data produk pada database.

Parameters :

kode – id produk

Returns:

True jika data produk berhasil dihapus.

- + GenerateIdProduk() : int

Membuat id produk secara otomatis dan unik.

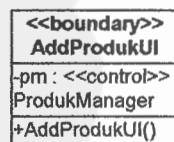
Parameters :

-

Returns:

Nilai id produk unik.

2.2.5.4 Class ASCM.Produksi.AddProdukUI



Gambar 2. 41 Class ASCM.Produksi.AddProduk

Deskripsi

Program Studi Teknik Informatika	DPPL-ASCaMaDa	54/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Class ini merepresentasikan class boundary/GUI untuk memasukkan data produk baru dalam sistem ASCaMaDa.

Atribut

- - pm : ProdukManager

Merepresentasikan object control class untuk pengelolaan tambah produk.

Method

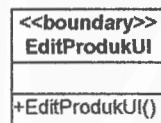
- + AddProdukUI()

Konstruktor class `ASCM.Produk.AddProdukUI`.

Buat instance baru tanpa atribut terdefinisi.

2.2.5.5

Class `ASCM.Produksi.EditProdukUI`



Gambar 2. 42 Class `ASCM.Produksi.EditProdukUI`

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk mengubah data produk dalam sistem ASCaMaDa.

Atribut

-

Method

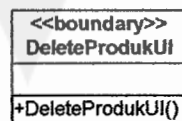
- + EditProdukUI()

Konstruktor class `ASCM.Produk.EditProdukUI`.

Buat instance baru tanpa atribut terdefinisi.

2.2.5.6

Class `ASCM.Produksi.DeleteProdukUI`



Gambar 2. 42 Class ASCM.Produksi.DeleteProdukUI

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk menghapus data produk dalam sistem ASCaMaDa.

Atribut

-

Method

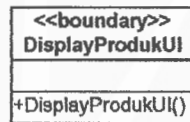
- + DeleteProdukUI ()

Konstruktor class ASCM.Produk.DeleteProdukUI.

Buat instance baru tanpa atribut terdefinisi.

2.2.5.7

Class ASCM.Produksi.DisplayProdukUI



Gambar 2. 42 Class ASCM.Produksi.DisplayProdukUI

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk menampilkan data produk dalam sistem ASCaMaDa.

Atribut

-

Method

- + DisplayProdukUI ()

Konstruktor class ASCM.Produk.DsisplayProdukUI.

Buat instance baru tanpa atribut terdefinisi.

2.2.5.8 Class ASCM.Produksi.Mesin

<<entity>> Mesin
-KodeMesin : string -NamaMesin : string
+Mesin() +Mesin(in kodemesin : string, in namamesin : string) +AddMesin() : string +UpdateStatusMesin(in kode : string, in status : string) : bool +DeleteUpdateMesin() : bool

Gambar 2. 42 Class ASCM.Produksi.Mesin

Deskripsi

Class ini merepresentasikan data pengelolaan mesin pada sistem ASCaMaDa.

Atribut

- - KodeMesin : string
Merepresentasikan kode mesin.
- - NamaMesin : string
Merepresentasikan nama mesin.

Method

- + Mesin()
Konstruktor class ASCM.Produk.Mesin. Buat instance baru tanpa atribut terdefinisi.
- + Mesin(kodeMesin : String, namaMesin : String)
Konstruktor class ASCM.Produk.Mesin. Buat instance baru dengan atribut terdefinisi.
- + AddMesin() : String
Menambahkan data mesin baru ke basisdata.

Parameters :

-

Returns:

"sukses" jika data mesin berhasil ditambahkan.

- + UpdateStatusMesin(kode : String, status : String)
: Boolean

Program Studi Teknik Informatika	DPPL-ASCaMaDa	57/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Mengubah status mesin dengan status yang baru pada basisdata.

Parameters :

kode – kode mesin

status – status mesin

Returns:

True jika statua mesin berhasil diubah

- + DeleteUpdateMesin() : Boolean

Menghapus data mesin pada database.

Parameters :

-

Returns:

True jika data mesin berhasil dihapus.

2.2.5.9 Class ASCM.Produksi.MesinManager

<<control>> MesinManager
+MesinManager() +AddMesin(in kodeMesin, in namaMesin) : string +DeleteUpdateMesin(in kode : string) : bool +UpdateStatusMesin(in kode : string, in status : string) : bool

Gambar 2. 43 Class ASCM.Produksi.MesinManager

Deskripsi

Class ini merepresentasikan control class untuk data pengelolaan mesin pada system ASCaMaDa..

Atribut

-

Method

- + MesinManager ()

Konstruktor class ASCM.Produksi.ProduksiManager. Buat instance baru tanpa atribut terdefinisi.

- + AddMesin(kodeMesin : string, namaMesin : string)
: String

Program Studi Teknik Informatika	DPPL-ASCaMaDa	58/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Menambahkan data mesin baru ke basisdata.

Parameters :

-

Returns:

“sukses” jika data mesin berhasil ditambahkan.

- + DeleteUpdateMesin(kode : string) : Boolean

Menghapus data mesin pada database.

Parameters :

kode – kode mesin

Returns:

True jika data mesin berhasil dihapus.

- + UpdateStatusMesin(kode : String, status : String) : Boolean

Mengubah status mesin dengan status yang baru pada basisdata.

Parameters :

kode – kode mesin

status – status mesin

Returns:

True jika statua mesin berhasil diubah

2.2.5.10 Class ASCM.Produksi.AddMesinUI

```
<<boundary>>
AddEditMesinUI
-mm : <<control>>
MesinManager
+AddEditMesinUI()
```

Gambar 2. 44 Class ASCM.Produksi.AddEditMesinUI

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk memasukkan data mesin baru dalam ASCaMaDa.

Atribut

- - mm : MesinManager

Program Studi Teknik Informatika	DPPL-ASCaMaDa	59/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Merepresentasikan object control class untuk pengelolaan tambah mesin.

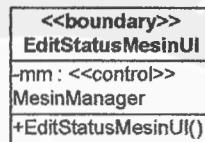
Method

- + AddMesinUI()

Konstruktor class `ASCM.Produk.AddMesinUI`.

Buat instance baru tanpa atribut terdefinisi.

2.2.5.11 Class `ASCM.Produksi.EditStatusMesinUI`



Gambar 2. 44 Class `ASCM.Produksi.EditStatusMesinUI`

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk memasukkan data status mesin yang baru dalam ASCaMaDa.

Atribut

- - mm : MesinManager

Merepresentasikan object control class untuk pengelolaan edit status mesin.

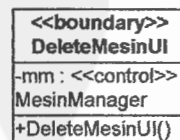
Method

- + EditStatusMesinUI()

Konstruktor class `ASCM.Produk.EditStatusMesinUI`.

Buat instance baru tanpa atribut terdefinisi.

2.2.5.12 Class `ASCM.Produksi.DeleteMesinUI`



Gambar 2. 44 Class `ASCM.Produksi.DeleteMesinUI`

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk menghapus data mesin dalam ASCaMaDa.

Atribut

- - mm : MesinManager

Merepresentasikan object control class untuk pengelolaan hapus mesin.

Method

- + DeleteMesinUI()

Konstruktor class ASCM.Produk.DeleteMesinUI.

Buat instance baru tanpa atribut terdefinisi.

2.2.5.13 Class ASCM.Produksi.HasilProduksi

<<entity>> HasilProduksi
-KodeSPPMaster : int -KodeProduksi : int -KodeProduk : int -JmlJadi : int
+HasilProduksi() +HasilProduksi(in kodeproduksi : int, in kodeproduk : int, in jmljadi : int, in kodeSPPMaster : int) +AddHasilProduksi() : string +GetLastIdBrgHslProduksi() : int

Gambar 2. 42 Class ASCM.Produksi.HasilProduksi

Deskripsi

Class ini merepresentasikan data pengelolaan hasil produksi pada sistem ASCaMaDa.

Atribut

- - KodeSPPMaster : int

Merepresentasikan kode SPP master.

- - KodeProduksi : int

Merepresentasikan kode produksi.

- - KodeProduk : int

Merepresentasikan kode produk.

- - JmlJadi : int

Merepresentasikan jumlah produksi.

Method

- + HasilProduksi()

Konstruktor class `ASCM.Produk.HasilProduksi`. Buat instance baru tanpa atribut terdefinisi.

- + HasilProduksi(kodeproduksi : int, kodeproduk : int, jmljadi : int, kodeSPPMaster : int)

Konstruktor class `ASCM.Produk.HasilProduksi`. Buat instance baru dengan atribut terdefinisi.

- + AddHasilProduksi() : String

Menambahkan data hasil produksi ke basisdata.

Parameters :

-

Returns:

“sukses” jika data hasil produksi berhasil ditambahkan.

- + GetLastIdBrgHslPProduksi() : int

Mengambil nilai id brg hasil produksi yang terakhir pada database.

Parameters :

-

Returns:

Nilai id hasil produksi unik.

2.2.5.14 Class `ASCM.Produksi.HasilProduksiManager`

<<control>> HasilProduksiManager
<code>+HasilProduksiManager()</code> <code>+AddHasilProduksi(in kodeproduksi : int, in kodeproduk : int, in jmljadi : int, in kodeSPPMaster : int) : string</code> <code>+GenerateldHasilProduksi() : int</code>

Program Studi Teknik Informatika	DPPL-ASCaMaDa	62/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Gambar 2. 43 Class ASCM.Produksi.HasilProduksiManager

Deskripsi

Class ini merepresentasikan control class untuk data pengelolaan hasil produksi pada sistem ASCaMaDa..

Atribut

-

Method

- + HasilProduksi()
Konstruktor class ASCM.Produksi.HasilProduksiManager.
Buat instance baru tanpa atribut terdefinisi.
- + AddHasilProduksi(kodeproduksi : int, kodeproduk : int, jmljadi : int, kodeSPPMaster : int) : String
Menambahkan data hasil produksi ke basisdata.

Parameters :

kodeProduksi - kode produksi
kodeproduk - kode produk
jmljadi - jumlah hasil produksi
kodeSPPMaster - kode surat perintah produksi

Returns:

"sukses" jika data hasil produksi berhasil ditambahkan.

- + GenerateIdHasilProduksi() : int
Membuat id hasil produksi secara otomatis dan unik.

Parameters :

-

Returns:

Nilai id hasil produksi unik.

2.2.5.15 Class ASCM.Produksi.EntriHasilProduksiUI

<<boundary>>
EntriHasilProduksiUI
-hpm : <<control>>
HasilProduksiManager
+EntriHasilProduksiUI()

Gambar 2. 44 Class ASCM.Produksi.EntriHasilProduksiUI

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk menambah data hasil produksi yang baru dalam ASCaMaDa.

Atribut

- - hpm : HasilProduksiManager

Merepresentasikan object control class untuk pengelolaan entri hasil produksi.

Method

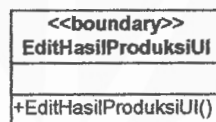
- + EntriHasilProduksiUI ()

Konstruktor class ASCM.Produk.EntriHasilProduksiUI.

Buat instance baru tanpa atribut terdefinisi.

2.2.5.16

Class ASCM.Produksi.EditHasilProduksiUI



Gambar 2. 44 Class ASCM.Produksi.EditHasilProduksiUI

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk mengubah data hasil produksi dalam ASCaMaDa.

Atribut

-

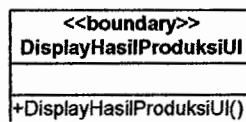
Method

- + EditHasilProduksiUI ()

Konstruktor class ASCM.Produk.EditHasilProduksiUI.

Buat instance baru tanpa atribut terdefinisi.

2.2.5.17 Class ASCM.Produk.DisplayHasilProduksiUI



Gambar 2. 44 Class ASCM.Produksi.DisplayHasilProduksiUI

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk menampilkan data hasil produksi dalam ASCaMaDa.

Atribut

-

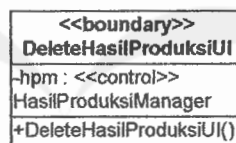
Method

- + DisplayHasilProduksiUI ()

Konstruktor

class
ASCM.Produk.DisplayHasilProduksiUI. Buat instance baru
tanpa atribut terdefinisi.

2.2.5.18 Class ASCM.Produk.DeleteHasilProduksiUI



Gambar 2. 44 Class ASCM.Produksi.DeleteHasilProduksiUI

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk menghapus data hasil produksi dalam ASCaMaDa.

Atribut

- - hpm : HasilProduksiManager

Merepresentasikan object control class untuk
pengelolaan delete hasil produksi.

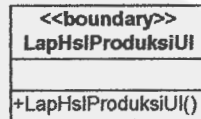
Method

- + DeleteHasilProduksiUI()

Konstruktor class ASCM.Produk.DeleteHasilProduksiUI.

Buat instance baru tanpa atribut terdefinisi.

2.2.5.19 Class ASCM.Produk.LapHslProduksiUI



Gambar 2. 44 Class ASCM.Produksi.LapHslProduksiUI

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk membuat laporan data hasil produksi dalam ASCaMaDa.

Atribut

-

Method

- + LapHslProduksiUI()

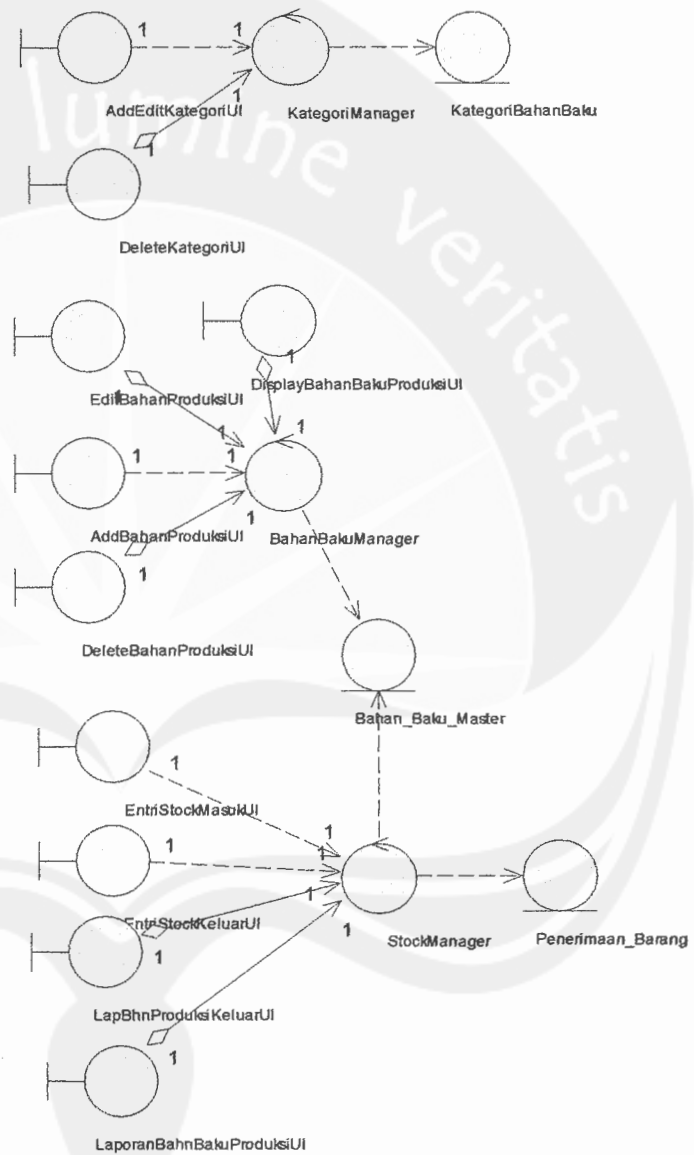
Konstruktor class ASCM.Produk.LapHslProduksiUI.

Buat instance baru tanpa atribut terdefinisi.

2.2.6 Package Warehouse (ASCM.Warehouse)

Package ini menyediakan class-class yang diperlukan untuk melakukan pengelolaan gudang.

2.2.6.1 Class diagram package ASCM.Warehouse



Gambar 2.46 Class Diagram Package ASCM.Warehouse

2.2.6.2 Class ASCM.Warehouse.Stock

<<entity>> Stock
-KodeBarang : string -KodePenerimaanBB : int -KodeSupplier : string -KodeTransporter : string -QinBox : int -QinKg : int
+Stock() +Stock(in kodePenerimaanBB : int, in qinkg : int, in qinbox : int, in kodeSupplier : string, in kodeBarang : string, in kodeTransporter : string) +AddPenerimaanBarang() : string +UpdateStockMasuk() : string +UpdateStatusBBKeluar(in kodeSPP : int, in kodeBB : string) : string +GetLastIdPenerimaanBarang() : int +GetQuantityBB(in kodeSPPdet : int, in kodeBrg : string) : int

Gambar 2. 47 Class ASCM.Warehouse.Stok

Deskripsi

Class ini merepresentasikan data stock pada sistem ASCaMaDa.

Atribut

- - KodeBarang : string
Merepresentasikan kode barang.
- - KodePenerimaanBB : int
Merepresentasikan kode penerimaan bahan baku.
- - KodeSupplier : string
Merepresentasikan kode supplier.
- - KodeTransporter : string
Merepresentasikan kode barang.
- - QinBox : string
Merepresentasikan jumlah dalam box.
- - QinKg : string
Merepresentasikan jumlah dalam kilogram.

Method

- + Stock()
Konstruktor class ASCM.Warehouse.Stock. Buat instance baru tanpa atribut terdefinisi.

Program Studi Teknik Informatika	DPPL-ASCaMaDa	68/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- + Stock(kodepenerimaanBB : int, qinkg : int, qinbox : int, kodeSupplier : string, kodeBarang : string, kodeTransporter : string)

Konstruktor class ASCM.Warehouse.Stock. **Buat instance** baru dengan atribut terdefinisi.

- + AddPenerimaanBarang() : String

Menambah data penerimaan barang dari supplier.

Parameters :

-

Returns:

“sukses” jika data berhasil ditambahkan.

- + UpdateStokMasuk() : String

Meng-update data stok yang masuk pada basisdata.

Parameters :

-

Returns:

“sukses” jika data stok berhasil di update.

- + UpdateStatusBBKeluar(kodeSPP : int, kodeBB : string) : String

Meng-update status bahan baku yang telah keluar dari gudang untuk keperluan produksi.

Parameters :

kodeSPP - kode SPP

kodeBB - kode bahan baku

Returns:

“sukses” jika data stok berhasil di update.

- + GetLastIdPenerimaanBarang() : int

Mengambil data id penerimaan barang yang terakhir.

Parameters :

-

Returns:

Nilai id penerimaan barang berikutnya dan unik.

- + GetQuantityBB(kodeSPPdet : int, kodeBrg : string)
: int

Mengambil data jumlah bahan baku produksi.

Parameters :

-

Returns:

Nilai jumlah bahan baku produksi.

2.2.6.3 Class ASCM.Warehouse.StockManager

<<control>> StockManager
+StockManager() +AddBarangMasuk(in kodePenerimaanBB : int, in qinkg : int, in qinbox : int, in kodeSupplier : string, in kodeBarang : string, in kodeTransporter : string) : string +GenerateIdPenerimaanBrg() : int +GetQuantityBB(in kodeSPPDet : int, in kodeBrg : string) : int +UpdateStatusBBKeluar(in kodeSPP : int, in kodeBB : string) : string

Gambar 2. 48 Class ASCM.Warehouse.StockManagerUI

Deskripsi

Class ini merepresentasikan control class untuk pengelolaan data stock pada sistem ASCaMaDa.

Atribut

-

Method

- + StockManager ()
Konstruktor class ASCM.Warehouse.StockManager. **Buat instance baru tanpa atribut terdefinisi.**
- + AddBarangMasuk(kodepenerimaanBB : int, qinkg : int, qinbox : int, kodeSupplier : string, kodeBarang : string, kodeTransporter : string) : String
Menambahkan data barang yang masuk ke gudang dari supplier.

Parameters :

Program Studi Teknik Informatika	DPPL-ASCaMaDa	70/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

kodePenerimaanBB - kode penerimaan bahan baku
qinBox - jumlah bahan baku dalam box
qinKg - jumlah bahan baklu dalam Kg
kodeSupplier - kode supplier
kodebarang - kode bahan baku
kodeTransporter - kode kendaraan pengangkut

Returns:

“sukses” jika data berhasil ditambahkan ke database.

- + GenerateIdPenerimaanBrg() : int

Meng-generate id penerimaan barang secara otomatis dan unik.

Parameters :

-

Returns:

Nilai id penerimaan barang unik.

- + GetQuantityBB(kodeSPPDet : int, kodeBrg : string) : int

Mengambil nilai jumlah bahan baku.

Parameters :

kodeSPPDet - kode surat perintah produksi

kodeBrg - kode bahan baku

Returns:

Nilai jumlah bahan baku.

- + UpdateStatusBBKeluar(kodeSPP : int, kodeBB: string) : string

Meng-update status bahan baku produksi keluar.

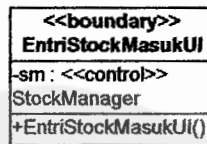
Parameters :

-

Returns:

“sukses” jika data berhasil diupdate

2.2.6.4 Class ASCM.Warehouse.EntriStockMasukUI



Gambar 2. 44 Class ASCM.Warehouse.EntriStockMasukUI

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk memasukkan data stok masuk dari kiriman supplier..

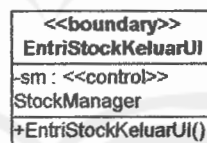
Atribut

- - sm : StockManager
Merepresentasikan object control class untuk pengelolaan stock masuk.

Method

- + EntriStockMasukUI ()
Konstruktor class ASCM.Warehouse.EntriStockMasukUI.
Buat instance baru tanpa atribut terdefinisi.

2.2.6.5 Class ASCM.Warehouse.EntriStockKeluarUI



Gambar 2. 44 Class ASCM.Warehouse.EntriStockKeluarUI

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk memasukkan data stok keluar untuk keperluan produksi.

Atribut

- - sm : StockManager
Merepresentasikan object control class untuk

pengelolaan stock keluar.

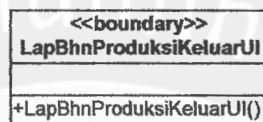
Method

- + EntriStockKeluarUI()

Konstruktor class ASCM.Warehouse.EntriStockKeluarUI.

Buat instance baru tanpa atribut terdefinisi.

2.2.6.6 Class ASCM.Warehouse.LaporanBhnProduksiKeluarUI



Gambar 2. 44 Class ASCM.Warehouse.LapBhnProduksiKeluarUI

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk membuat laporan dan mencetak bahan baku produksi keluar.

Atribut

-

Method

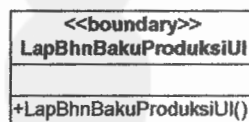
- + LapBhnProduksiKeluarUI()

Konstruktor

class

ASCM.Warehouse.LapBhnProduksiKeluarUI. **Buat instance** baru tanpa atribut terdefinisi.

2.2.6.7 Class ASCM.Warehouse.LapBhnBakuProduksiUI



Gambar 2. 44 Class ASCM.Warehouse.LapBhnBakuProduksiUI

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk membuat laporan dan mencetak bahan baku produksi.

Atribut

-

Method

- + LapBhnProduksiUI()
Konstruktor class ASCM.Warehouse.LapBhnProduksiUI.
Buat instance baru tanpa atribut terdefinisi.

2.2.6.8 Class ASCM.Warehouse.Kategori

<<entity>> Kategori
-KodeKategori : string -NamaKategori : string -Deskripsi : string
+Kategori() +Kategori(in kodekategori : string, in namakategori : string, in deskripsi : string) +AddKategori() : string +DeleteUpdateKategori() : bool

Gambar 2. 47 Class ASCM.Warehouse.Kategori

Deskripsi

Class ini merepresentasikan data kategori pada sistem ASCaMaDa.

Atribut

- - KodeKategori : string
Merepresentasikan kode kategori.
- - NamaKategori : string
Merepresentasikan nama kategori.
- - Diskripsi : string
Merepresentasikan deskripsi kategori.

Method

- + Kategori()
Konstruktor class ASCM.Warehouse.Kategori. Buat instance baru tanpa atribut terdefinisi.

- + Kategori(kodekategori : string, namakategori : int, deskripsi : string)

Konstruktor class ASCM.Warehouse.Kategori. **Buat instance baru dengan atribut terdefinisi.**

- + AddKategori() : String

Menambah data kategori baru barang dari supplier.

Parameters :

-

Returns:

“sukses” jika data berhasil ditambahkan.

- + DeleteUpdateKategori() : Boolean

Menghapus data kategori pada tabel kategori.

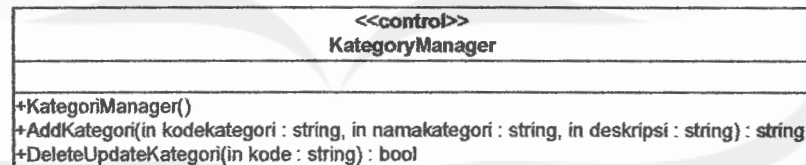
Parameters :

-

Returns:

True jika data stok berhasil di dihapus.

2.2.6.9 Class ASCM.Warehouse.KategoriManager



Gambar 2. 48 Class ASCM.Warehouse.KategoriManagerUI

Deskripsi

Class ini merepresentasikan control class untuk pengelolaan data kategori pada sistem ASCaMaDa.

Atribut

-

Method

- + KategoriManager()

Konstruktor class ASCM.Warehouse.KategoriManager.

Buat instance baru tanpa atribut terdefinisi.

- + AddKategori(kodekategori : string, namakategori : string, deskripsi : string) : String

Menambahkan data kategori pada tabel kategori.

Parameters :

kodekategori - kode kategori
 namakategori - nama kategori
 deskripsi - deskripsi kaegori

Returns:

“sukses” jika data berhasil ditambahkan ke database.

- + DeleteUpdateKategori(kode : string) : Boolean

Menghapus kategori dari database.

Parameters :

kode - kode kategori

Returns:

True jika data berhasil dihapus.

2.2.6.10 Class ASCM.Warehouse.AddEditKategoriUI

<<boundary>> AddEditKategoriUI
-km : <<control>> KategoryManager
+AddEditkategoriUI()

Gambar 2. 44 Class ASCM.Warehouse.AddEditKategoriUI

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk memasukkan data dan mengubah data kategori.

Atribut

- - km : KategoriManager

Merepresentasikan object control class untuk pengelolaan kategori bahan baku.

Method

- + AddEditKategoriUI ()

Konstruktor class `ASCM.Warehouse.AddEditkategoriUI`.

Buat instance baru tanpa atribut terdefinisi.

2.2.6.11 Class `ASCM.Warehouse.DeleteKategoriUI`



Gambar 2. 44 Class `ASCM.Warehouse.DeleteKategoriUI`

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk menghapus data ketegori.

Atribut

- - km : KategoriManager

Merepresentasikan object control class untuk

Menghapus data kategori.

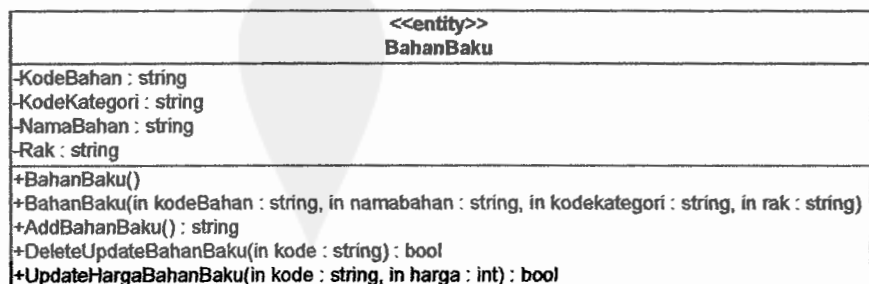
Method

- + DeleteKategoriUI ()

Konstruktor class `ASCM.Warehouse.DeleteKategoriUI`.

Buat instance baru tanpa atribut terdefinisi.

2.2.6.12 Class `ASCM.Warehouse.BahanBaku`



Gambar 2. 47 Class ASCM.Warehouse.BahanBaku

Deskripsi

Class ini merepresentasikan data Bahan Baku pada sistem ASCaMaDa.

Atribut

- - KodeBahan : string
Merepresentasikan kode bahan.
- - KodeKategori : string
Merepresentasikan kode kategori bahan baku.
- - NamaBahan : string
Merepresentasikan nama kategori.
- - Rak : string
Merepresentasikan rak tempat bahan baku.

Method

- + BahanBaku()
Konstruktor class ASCM.Warehouse.BahanBaku. Buat instance baru tanpa atribut terdefinisi.
- + BahanBaku(kodeBahan : string, namabahan : string, kodekategori : string, rak : string)
Konstruktor class ASCM.Warehouse.BahanBaku. Buat instance baru dengan atribut terdefinisi.
- + AddBahanBaku() : String
Menambah data bahan baku baru ke tabel bahan_baku_master.
Parameters :
-
Returns:
"sukses" jika data berhasil ditambahkan.
- + DeleteUpdateBahanBaku(kode : string) : Boolean
Menghapus data bahan baku produksi.

Parameters :

kode - kode bahan baku

Returns:

True jika data berhasil di hapus.

- + UpdateHargaBahanBaku(kode : string, harga : int)
: Boolean

Meng-update harga bahan baku produksi.

Parameters :

kode - kode bahan baku

harga - harga bahan baku

Returns:

"sukses" jika data stok berhasil di update.

2.2.6.13 Class ASCM.Warehouse.BahanBakuManager

<<control>> BahanBakuManager
+BahanBakuManager() +AddBahan(in kodeBahan : string, in namaBahan : string, in kodeKategori : string, in rak : string) +UpdateHargaBahan(in kode : string, in harga : int) : bool +DeleteUpdateBahan(in kode : string) : bool

Gambar 2. 48 Class ASCM.Warehouse.BahanBakuManagerUI

Deskripsi

Class ini merepresentasikan control class untuk pengelolaan data bahan baku pada sistem ASCaMaDa.

Atribut

-

Method

- + BahanBakuManager ()
Konstruktor class ASCM.Warehouse.BahanBakuManager.
Buat instance baru tanpa atribut terdefinisi.
- + AddBahan(kodeBahan : string, namabahan : string,
kodekategori : string, rak : string) : String

Program Studi Teknik Informatika	DPPL-ASCaMaDa	79/ 277
----------------------------------	---------------	---------

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika

Menambahkan data bahan baku ke tabel bahan_baku_master.

Parameters :

kodeBahan - kode bahan baku
namaBahan - nama bahan baku
rak - rak tempat bahan baku

Returns:

“sukses” jika data berhasil ditambahkan ke database.

- + UpdateHargaBahan(kode : string, harga : int) : Boolean

Meng-update harga bahan baku produksi.

Parameters :

kode - kode bahan baku
harga - harga bahan baku

Returns:

True jika harga berhasil diupdate.

- + DeletUpdateBahan(kode : string) : Boolean

Menghapus data bahan baku dari database.

Parameters :

kode - kode bahan baku

Returns:

True jika bahan baku berhasil dihapus.

2.2.6.14 Class ASCM.Warehouse.AddBahanProduksiUI

<<boundary>> AddBahanProduksiUI
-bbm : <<control>> BahanBakuManager
+AddBahanProduksiUI()

Gambar 2. 44 Class ASCM.Warehouse.AddBahanProduksiUI

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk memasukkan data bahan baku produksi yang baru.

Program Studi Teknik Informatika	DPPL-ASCaMaDa	80/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Atribut

- - bbm : BahanBakuManager

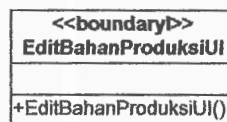
Merepresentasikan object control class untuk pengelolaan bahan baku.

Method

- + AddBahanBakuUI ()

Konstruktor class `ASCM.Warehouse.AddBahanBakuUI`.
Buat instance baru tanpa atribut terdefinisi.

2.2.6.15 Class `ASCM.Warehouse.EditBahanBakuUI`



Gambar 2. 44 Class `ASCM.Warehouse.EditBahanProduksiUI`

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk mengubah data bahan produksi.

Atribut

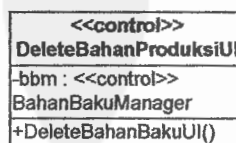
-

Method

- + EditBahanProduksiUI ()

Konstruktor class `ASCM.Warehouse.EditBahanProduksiUI`.
Buat instance baru tanpa atribut terdefinisi.

2.2.6.16 Class `ASCM.Warehouse.DeleteBahanProduksiUI`



Gambar 2. 44 Class `ASCM.Warehouse.DeleteBahanProduksiUI`

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk menghapus bahan produksi dari database.

Atribut

- - bbm : BahanBakuManager

Merepresentasikan object control class untuk pengelolaan bahan baku.

Method

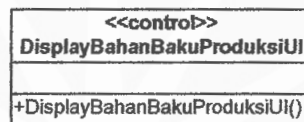
- + DeleteBahanProduksiUI()

Konstruktor

class

ASCM.Warehouse.DeleteBahanProduksiUI. Buat instance baru tanpa atribut terdefinisi.

2.2.6.17 Class *ASCM.Warehouse.DisplayBahanBakuProduksiUI*



Gambar 2. 44 Class *ASCM.Warehouse.DisplayBahanBakuProduksiUI*

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk menampilkan data bahan baku produksi.

Atribut

-

Method

- + DisplayBahanBakuProduksiUI()

Konstruktor

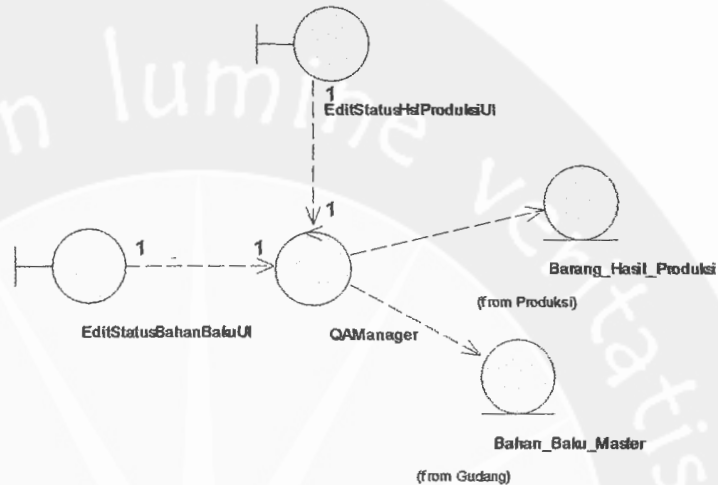
class

ASCM.Warehouse.DisplayBahanBakuProduksiUI. Buat instance baru tanpa atribut terdefinisi.

2.2.7 Package QualityAssurance (ASCM.QualityAssurance)

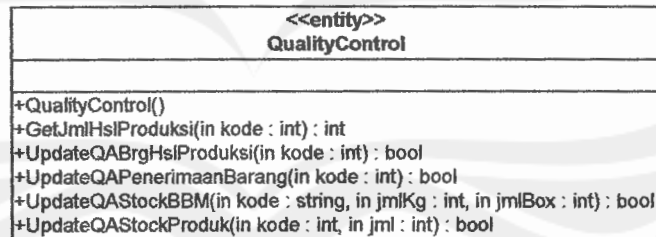
Package ini menyediakan class-class yang diperlukan untuk melakukan pengelolaan Quality Control.

2.2.7.1 Class diagram package ASCM.QualityAssurance



Gambar 2.46 Class Diagram Package ASCM.QualityAssurance

2.2.7.2 Class ASCM.QualityAssurance.QualityControl



Gambar 2.47 Class ASCM.QualityAssurance.QualityControl

Deskripsi

Class ini merepresentasikan data Quality Assurance pada sistem ASCaMaDa.

Atribut

-

Method

- + `QualityControl()`

Konstruktor

class

`ASCM.QualityAssurance.QualityControl`. **Buat instance baru tanpa atribut terdefinisi.**

- + `GetJmlHslProduksi(kode : int) : int`

Mengambil data jumlah hasil produksi.

Parameters :

kode - kode produksi

Returns:

Nilai jumlah hasil produksi.

- + `UpdateQAbrgHslProduksi(kode : int) : Boolean`

Meng-update barang hasil produksi bahwa barang hasil produksi sudah dicek.

Parameters :

kode - kode produk

Returns:

True jika data berhasil diupdate

- + `UpdateQAPenerimaanBarang(kode : int) : Boolean`

Meng-update bahan baku produksi pada tabel penerimaan barang bahwa bahan baku sudah dicek.

Parameters :

kode - kode bahan baku

Returns:

True jika data berhasil diupdate

- + `UpdateQAStockBBM(kode : string, jmlKg : int, jmlBox : int) : Boolean`

Meng-update stock bahan baku produksi.

Parameters :

kode - kode bahan baku

jmlKg - jumlah bahan baku dlm Kg

jmlBox - jumlah bahan baku dlm Box

Returns:

True jika data berhasil diupdate

- + UpdateQAStockProduk(kode : int, jml : int) : Boolean

Meng-update produk hasil produksi.

Parameters :

kode - kode produk
 jml - jumlah produk

Returns:

True jika data berhasil diupdate

2.2.7.3

Class ASCM.QualityAssurance.QualityControlManager

<<control>> QualityControlManager
+QualityControlManager() +getJmlHslProduksi(in kode : int) : int +UpdateQAHslProduksi(in kodeProduksi : int, in kodeProduk : int, in jml : int) : bool +UpdateQAPenerimaanBarang(in kodePenerimaan : int, in kodeBarang : string, in jmlKg : int, in jmlBox : int) : bool

Gambar 2. 48 Class ASCM.QualityAssurance.QualityControlManager

Deskripsi

Class ini merepresentasikan control class untuk pengelolaan data Quality Assurance pada sistem ASCaMaDa.

Atribut

-

Method

- + QualityControlManager() class
 Konstruktor Buat
 ASCM.QualityAssurance.QualityControlManager.
 instance baru tanpa atribut terdefinisi.
- + getJmlHslProduksi(kode : int) : int
 Mengambil data jumlah hasil produksi.

Parameters :

kode - kode produksi

Returns:

Nilai jumlah hasil produksi.

- + UpdateQAHaslProduksi (kodeProduksi : int, kodeProduk : int, jml : int) : Boolean

Meng-update barang hasil produksi.

Parameters :

kode - kode produk

Returns:

True jika data berhasil diupdate

- + UpdateQAPenerimaanBarang (kodePenerimaan : int, kodeBarang : string, jmlKg : int, jmlBox : int) : Boolean

Meng-update bahan baku produksi pada tabel penerimaan barang bahwa bahan baku sudah dicek dan mengupdate tabel bahan_baku_master.

Parameters :

kode - kode bahan baku

Returns:

True jika data berhasil diupdate.

2.2.7.4 Class ASCM.QualityAssurance.EditStatusbahanBakuUI

<<boundary>> EditStatusbahanBakuUI
-qam : <<control>> QualityControlManager
+EditStatusBahanBakuUI()

Gambar 2. 44 Class ASCM.QualityAssurance.EditStatusBahanBakuUI

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk mengubah status bahan baku produksi.

Atribut

- - qam : QualityControlManager

Merepresentasikan object control class untuk pengelolaan Quality Control.

Method

- + EditStatusbahanBakuUI ()

Konstruktor

class

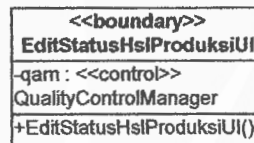
ASCM.QualityAssurance.EditStatusbahanBakuUI.

Buat

instance baru tanpa atribut terdefinisi.

2.2.7.5

Class ASCM. QualityAssurance.EditStatusHslProduksiUI



Gambar 2. 44 Class ASCM.QualityAssurance.EditStatusHslProduksiUI

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk mengubah status hasil produksi.

Atribut

- - qam : QualityControlManager

Merepresentasikan object control class untuk pengelolaan Quality Control.

Method

- + EditStatusHslProduksiUI ()

Konstruktor

class

ASCM.QualityAssurance.EditStatusHslProduksiUI.

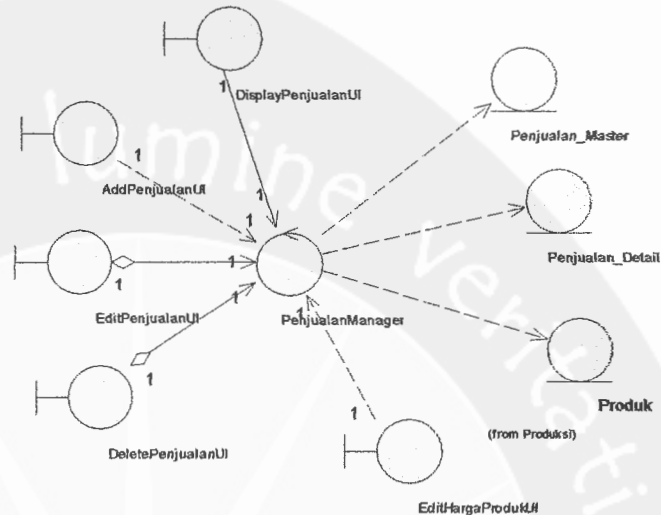
Buat

instance baru tanpa atribut terdefinisi.

2.2.8 Package Marketing (ASCM.Marketing)

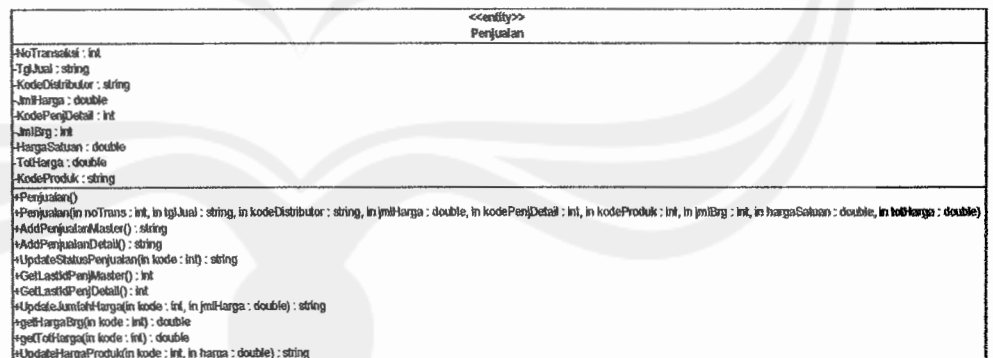
Package ini menyediakan class-class yang diperlukan untuk melakukan pengelolaan marketing.

2.2.8.1 Class diagram package ASCM.Marketing



Gambar 2.46 Class Diagram Package ASCM.Marketing

2.2.8.2 Class ASCM.Marketing.Penjualan



Gambar 2. 47 Class ASCM.Marketing.Penjualan

Deskripsi

Class ini merepresentasikan data Penjualan pada sistem ASCaMaDa.

Atribut

- - NoTransaksi : int
Merepresentasikan no transaksi.
- - TglJual : string
Merepresentasikan tgl penjualan.
- - KodeDistributor : string
Merepresentasikan kode distributor.
- - JmlHarga : double
Merepresentasikan harga satuan.
- - KodePenjDetail : int
Merepresentasikan kode penjualan detail.
- - JmlBrg : int
Merepresentasikan jumlah barang.
- - HargaSatuan : double
Merepresentasikan harga barang satuan.
- - TotHarga : double
Merepresentasikan total harga.
- - KodeProduk : string
Merepresentasikan kode produk.

Method

- + Penjualan()
Konstruktor class ASCM.Marketing.Penjualan. Buat instance baru tanpa atribut terdefinisi.
- + Penjualan(noTrans : int, tglJual: string, kodeDistributor: string, jmlHarga : double, kodePenjDetail : int, kodeProduk : int, jmlBrg : int, hargaSatuan : double, totHarga : double)

Program Studi Teknik Informatika	DPPL-ASCaMaDa	89/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Konstruktor class `ASCM.Marketing.Penjualan`. **Buat instance baru dengan atribut terdefinisi.**

- `+ AddPenjualanMaster() : String`

Menambah data penjualan ke tabel penjualan_master.

Parameters :

-

Returns:

“sukses” jika data berhasil ditambahkan.

- `+ AddPenjualanDetail() : String`

Menambah data penjualan ke tabel penjualan_detail.

Parameters :

-

Returns:

“sukses” jika data berhasil ditambahkan.

- `+ UpdateStatusPenjualan(kode : int) : String`

Meng-update status produk telah layak dijual.

Parameters :

kode - kode produk

Returns:

“sukses” jika data berhasil diupdate.

- `+ GetLastIdPenjMaster() : int`

Mengambil data id penjualan master yang terakhir.

Parameters :

-

Returns:

Nilai id penjualan barang berikutnya dan unik.

- `+ GetLastIdPenjDetail() : int`

Mengambil data id penjualan detail yang terakhir.

Parameters :

-

Returns:

Program Studi Teknik Informatika	DPPL-ASCaMaDa	90/ 277
----------------------------------	---------------	---------

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika

Nilai id penjualan detail berikutnya dan unik.

- + UpdateJumlahHarga(kode : int, jmlHarga : double)
: String

Meng-update jumlah harga pembelian.

Parameters :

kode - kode produk

jmlHarga - jumlah harga penjualan

Returns:

“sukses” jika data penjualan berhasil di update.

- + getHargaBrg(kode : int) : double

Mengambil harga produk.

Parameters :

kode - kode produk

Returns:

Nilai harga produk.

- + getTotHarga(kode : int) : double

Mengambil total harga produk.

Parameters :

kode - kode produk

Returns:

Nilai total harga produk.

- + UpdateHargaProduk(kode : int, harga : double) :
String

Meng-update harga produk.

Parameters :

kode - kode produk

harga - harga produk

Returns:

“sukses” jika data berhasil diupdate.

2.2.8.3 Class ASCM.Marketing.PenjualanManager

<<control>> PenjualanManager
<pre> +PenjualanManager() +AddPenjMaster(in notransaksi : int, in tgljual : string, in kodedistri : string) : string +AddPenjDetail(in kodepenj : int, in notrans : int, in kodeproduk : int, in jmlbrg : int, in hargasatuan : double, in totalharga : double) : string +GenerateIdPenjMaster() : int +GenerateIdPenjDetail() : int +updateJmlHarga(in kode : int, in totl-Harga : double) : string +updateHarga(in kode : int, in harga : double) : string +getTotl-Harga(in kode : int) : double +get-Harga(in kode : int) : double </pre>

Gambar 2. 48 Class ASCM.Marketing.PenjualanManager

Deskripsi

Class ini merepresentasikan control class untuk pengelolaan data penjualan pada sistem ASCaMaDa.

Atribut

-

Method

- + PenjualanManager()

Konstruktor class ASCM.Marketing.PenjualanManager.

Buat instance baru tanpa atribut terdefinisi.

- + AddPenjMaster(notransaksi : int, tgljual : string, kodedistri : string) : String

Menambahkan data penjualan ke tabel penjualan_master.

Parameters :

notransaksi - no transaksi

tgljual - tanggal penjualan

kodeDistri - kode distributor

Returns:

“sukses” jika data berhasil ditambahkan ke database.

- + AddPenjDetail(kodepenj : int, notrans : int, kodeproduk : int, jmlbrg : int, hargasatuan : double, totalharga : double) : String

Menambahkan data penjualan ke tabel penjualan_detail.

Program Studi Teknik Informatika	DPPL-ASCaMaDa	92/ 277
<p>Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika</p>		

Parameters :

kodepenj - kode penjualan
notrans - no transaksi penjualan
kodeproduk - kode produk
hargasatuan - harga satuan
totalharga - harga total

Returns:

“sukses” jika data berhasil ditambahkan ke database.

- + GenerateIdPenjMaster() : int
Membuat id penjualan master secara otomatis dan unik.

Parameters :

-

Returns:

Nilai Id penjualan master.

- + GenerateIdPenjDetail() : int
Membuat id penjualan detail secara otomatis dan unik.

Parameters :

-

Returns:

Nilai Id penjualan detail.

- + UpdateJmlHarga(kode : int, totHarga : double) :
String

Meng-update jumlah harga pada database..

Parameters :

kode - kode produk
totHarga - total harga

Returns:

“sukses” jika data berhasil diupdate.

- + getHarga(kode : int) : double
Mengambil harga produk.

Parameters :

kode - kode produk

Returns:

Program Studi Teknik Informatika	DPPL-ASCaMaDa	93/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Nilai harga produk.

- + getTotHarga(kode : int) : double

Mengambil total harga produk.

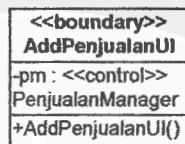
Parameters :

kode - kode produk

Returns:

Nilai total harga produk.

2.2.8.4 Class ASCM.Marketing.AddPenjualanUI



Gambar 2. 44 Class ASCM.Marketing.AddPenjualanUI

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk memasukkan data penjualan.

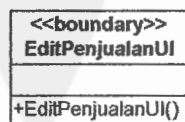
Atribut

- - pm : PenjualanManager
Merepresentasikan object control class untuk pengelolaan penjualan.

Method

- + AddPenjualanUI()
Konstruktor class ASCM.Marketing.AddPenjualanUI.
Buat instance baru tanpa atribut terdefinisi.

2.2.8.5 Class ASCM.Marketing.EditPenjualanUI



Gambar 2. 44 Class ASCM.Marketing.EditPenjualanUI

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk mengubah data penjualan.

Atribut

-

Method

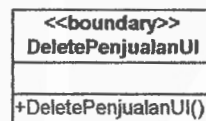
- + EditPenjualanUI()

Konstruktor class `ASCM.Marketing.EditPenjualanUI`.

Buat instance baru tanpa atribut terdefinisi.

2.2.8.6

Class `ASCM.Marketing.DeletePenjualanUI`



Gambar 2. 44 Class `ASCM.Marketing.DeletePenjualanUI`

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk menghapus data penjualan.

Atribut

-

Method

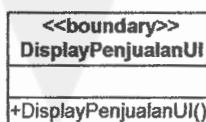
- + DeletePenjualanUI()

Konstruktor class `ASCM.Marketing.DeletePenjualanUI`.

Buat instance baru tanpa atribut terdefinisi.

2.2.8.7

Class `ASCM.Marketing.DisplayPenjualanUI`



Gambar 2. 44 Class `ASCM.Marketing.DisplayPenjualanUI`

- + `getHarga(kode : int) : double`

Mengambil harga bahan baku.

Parameters :

kode - kode bahan baku

Returns:

Nilai harga bahan baku.

- + `getJmlBarang(kode : String) : int`

Mengambil jumlah bahan baku permintaan pembelian.

Parameters :

kode - kode bahan baku

Returns:

Nilai jumlah bahan baku.

- + `getTotHarga(kode : int) : double`

Mengambil total harga pembelian.

Parameters :

kode - kode pembelian master

Returns:

Nilai total harga pembelian.

- + `GetLastIdPembMaster() : int`

Mengambil id pembelian master yang terakhir.

Parameters :

-

Returns:

Nilai id pembelian master yang terakhir.

- + `GetLastIdPembDetail() : int`

Mengambil id pembelian detail yang terakhir.

Parameters :

-

Returns:

Nilai id pembelian detail yang terakhir.

- + `UpdateTotalHarga(kode : int, jmlharga : double) :`

String

Meng-update total harga pembelian.

Parameters :

kode - kode pembelian master

jlmlharga - jumlah harga pembelian

Returns:

“sukses” jika data berhasil di update.

2.2.9.3 Class ASCM.Purchasing.PurchasingManager

<<control>> PurchasingManager
+PurchasingManager() +AddPembMaster(in kodepembelian : int, in kodePPMaster : int, in tgl : string) : string +AddPembDetail(in kodepembelian : int, in kodePembDetail : int, in jml : int, in hargaSat : double, in hargaTot : double, in kodeSupp : string, in kodeBarang : string) : string +GenerateIdPembDetail() : int +GenerateIdPembMaster() : int +getHargaBahanBaku(in kode : string) : double +getJumlahBahanBaku(in kode : string) : int +getTotalHarga(in kode : int) : double +updateTotHarga(in kode : int, in totHarga : double) : string

Gambar 2. 48 Class ASCM.Purchasing.PurchasingManager

Deskripsi

Class ini merepresentasikan control class untuk pengelolaan data pembelian pada sistem ASCaMaDa.

Atribut

-

Method

- + PurchasingManager()
Konstruktor class ASCM.Purchasing.PurchasingManager.
Buat instance baru tanpa atribut terdefinisi.
- + AddPembMaster(kodepembelian : int, kodePPmas : int, tgl : String) : String
Menambah data pembelian ke tabel pembelian_master.

Parameters :

kodepembelian - kode pembelian

kodePPMas - kode permintaan pembelian master

tgl - tanggal pembelian

Program Studi Teknik Informatika	DPPL-ASCaMaDa	101/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program StudiTeknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Returns:

“sukses” jika data berhasil ditambahkan.

- + AddPembelianDetail() : String

Menambah data pembelian ke tabel pembelian_detail.

Parameters :

-

Returns:

“sukses” jika data berhasil ditambahkan.

- + GenerateIdPembMaster() : int

Membuat id pembelian master secara otomatis dan unik.

Parameters :

-

Returns:

Nilai id pembelian master.

- + GenerateIdPembDetail() : int

Membuat id pembelian detail secara otomatis dan unik.

Parameters :

-

Returns:

Nilai id pembelian detail.

- + getHargaBahanBaku(kode : string) : double

Mengambil harga bahan baku.

Parameters :

kode - kode bahan baku

Returns:

Nilai harga bahan baku.

- + getJumlahBahanBaku(kode : String) : int

Mengambil jumlah bahan baku permintaan pembelian.

Parameters :

kode - kode bahan baku

Returns:

Nilai jumlah bahan baku.

- + getTotHarga(kode : int) : double

Mengambil total harga pembelian.

Parameters :

kode - kode pembelian master

Returns:

Nilai total harga pembelian.

- + updateTotalHarga(kode : int, totHarga : double) : String

Meng-update total harga pembelian.

Parameters :

kode - kode pembelian master

totharga - jumlah harga pembelian

Returns:

“sukses” jika data berhasil di update.

2.2.9.4 Class ASCM.Purchasing.OrderBahanProduksiUI

<<boundary>> OrderBahanProduksiUI
-pm : <<control>> PurchasingManager
+OrderBahanProduksiUI()

Gambar 2. 44 Class ASCM.Purchasing.OrderBahanProduksiUI

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk memasukkan data pembelian bahan baku produksi.

Atribut

- - pm : PurchasingManager

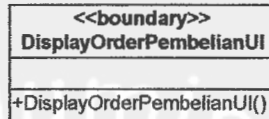
Merepresentasikan object control class untuk pengelolaan purchasing.

Method

- + OrderBahanProduksiUI()

Konstruktor **class** `ASCM.Purchasing.`
`OrderBahanProduksiUI`. **Buat instance baru tanpa atribut**
terdefinisi.

2.2.9.5 *Class `ASCM.Purchasing.DisplayOrderPembelianUI`*



Gambar 2. 44 Class `ASCM.Purchasing.DisplayOrderPembelianUI`

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk menampilkan data pembelian.

Atribut

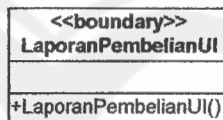
-

Method

- + `DisplayOrderPembelianUI()`

Konstruktor **class** `ASCM.Purchasing.`
`DisplayOrderPembelianUI`. **Buat instance baru tanpa**
atribut terdefinisi.

2.2.9.6 *Class `ASCM.Purchasing.LaporanPembelianUI`*



Gambar 2. 44 Class `ASCM.Purchasing.LaporanPembelianUI`

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk membuat dan mencetak laporan pembelian.

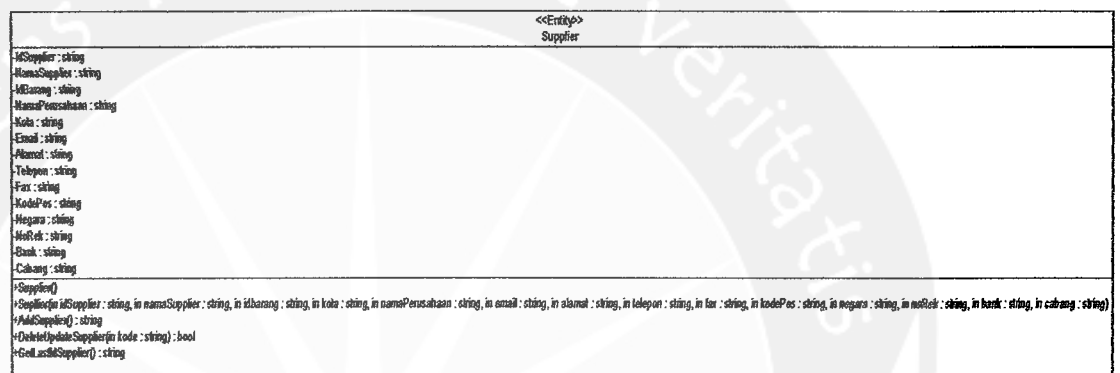
Atribut

Method

- + LaporanPembelianUI()

Konstruktor **class** **ASCM.Purchasing.**
LaporanPembelianUI. Buat instance baru tanpa atribut terdefinisi.

2.2.9.7 Class *ASCM.Purchasing.Supplier*



Gambar 2. 47 Class *ASCM.Purchasing.Supplier*

Deskripsi

Class ini merepresentasikan data supplier pada sistem ASCaMaDa.

Atribut

- - IdSupplier : string
Merepresentasikan id supplier.
- - NamaSupplier : string
Merepresentasikan nama supplier..
- - Id Barang : string
Merepresentasikan kode barang.
- - NamaPerusahaan : string
Merepresentasikan nama perusahaan.

- - Kota : string
Merepresentasikan nama kota supplier.
- - Email : string
Merepresentasikan alamat email supplier.
- - Alamat : string
Merepresentasikan alamat supplier.
- - Telepon : string
Merepresentasikan nomor telepon supplier.
- - Fax : string
Merepresentasikan no fax.
- - KodePos : string
Merepresentasikan kode pos.
- - Negara : string
Merepresentasikan degara asal supplier.
- - NoRek : string
Merepresentasikan no rekening supplier.
- - Bank : string
Merepresentasikan nama bank supplier.
- - Cabang : string
Merepresentasikan nama cabang tempat bank supplier.

Method

- + Supplier()
Konstruktor class ASCM.Purchasing.Supplier. **Buat instance baru tanpa atribut terdefinisi.**
- + Supplier(idSupplier : string, namaSupplier : string, idBarang : string, namaPerusahaan : string, kota : string, email : string, Alamat : string, telepon : string, fax : string, kodepos : string, negara : string, norek : string, bank : string, cabang : string)
Konstruktor class ASCM.Purchasing.Supplier. **Buat instance baru dengan atribut terdefinisi.**

- + AddSupplier() : String

Menambah data supplier ke tabel data_supplier.

Parameters :

-

Returns:

“sukses” jika data berhasil ditambahkan.

- + DeleteUpdateSupplier(kode : string) : Boolean

Menghapus data supplier dari tabel data_supplier.

Parameters :

kode - kode supplier

Returns:

True jika data berhasil dihapus.

- + GetLastIdSupplier() : string

Mengambil id supplier yang terakhir.

Parameters :

-

Returns:

Nilai id supplier yang terakhir.

2.2.9.8 Class ASCM.Purchasing.SupplierManager

<<control>> SupplierManager
<pre> +SupplierManager() +AddSupplier(in idSupplier: string, in namaSupplier: string, in kbarang: string, in kota: string, in namaPerusahaan: string, in email: string, in alamat: string, in telepon: string, in fax: string, in kodePos: string, in negara: string, in noRek: string, in bank: string, in cabang: string): string +DeleteSupplier(in kode: string): bool +GenerateIdSupplier(): string </pre>

Gambar 2. 48 Class ASCM.Purchasing.SupplierManagerUI

Deskripsi

Class ini merepresentasikan control class untuk pengelolaan data supplier pada sistem ASCaMaDa.

Atribut

-

Program Studi Teknik Informatika	DPPL-ASCaMaDa	1071 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Method

- + SupplierManager()

Konstruktor class ASCM.Purchasing.SupplierManager.

Buat instance baru tanpa atribut terdefinisi.

- + AddSupplier(idSupplier : string, namaSupplier : string, idBarang : string, namaPerusahaan : string, kota : string, email : string, Alamat : string, telepon : string, fax : string, kodepos : string, negara : string, norek : string, bank : string, cabang : string) : String

Menambah data supplier ke tabel data_supplier.

Parameters :

idSupplier - id supplier

namaSupplier - nama supplier

idBarang - kode barang

namaPerusahaan - nama perusahaan supplier

email - alamat email supplier

alamat - alamat supplier

telepon - nomor telepon supplier

fax - no faximile supplier

kodepos - kode pos supplier

negara - negara asal supplier

norek - nomor rekening bank supplier

bank - nama bank supplier

cabang - cabang bank supplier

Returns:

“sukses” jika data berhasil ditambahkan.

- + DeleteSupplier(kode : string) : Boolean

Menghapus data supplier dari tabel data_supplier.

Parameters :

kode - kode supplier

Returns:

True jika data berhasil dihapus.

- + GenerateIdSupplier() : string

Membuat id supplier secara otomatis dan unik .

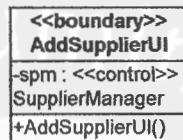
Parameters :

-

Returns:

Nilai id supplier unik.

2.2.9.9 Class *ASCM.Purchasing.AddSupplierUI*



Gambar 2. 44 Class *ASCM.Purchasing.AddSupplierUI*

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk memasukkan data supplier.

Atribut

- - spm : SupplierManager

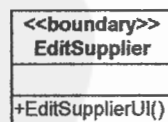
Merepresentasikan object control class untuk pengelolaan supplier.

Method

- + AddSupplierUI()

Konstruktor class *ASCM.Purchasing.AddSupplierUI*. Buat instance baru tanpa atribut terdefinisi.

2.2.9.10 Class *ASCM.Purchasing.EditSupplierUI*



Gambar 2. 44 Class *ASCM.Purchasing.EditSupplierUI*

Deskripsi

Program Studi Teknik Informatika	DPPL-ASCaMaDa	109/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Class ini merepresentasikan class boundary/GUI untuk melakukan edit supplier.

Atribut

-

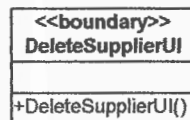
Method

- + EditSupplierUI()

Konstruktor class `ASCM.Purchasing.EditSupplierUI`.

Buat instance baru tanpa atribut terdefinisi.

2.2.9.11 Class `ASCM.Purchasing.DeleteSupplierUI`



Gambar 2. 44 Class `ASCM.Purchasing.DeleteSupplierUI`

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk menghapus data supplier.

Atribut

-

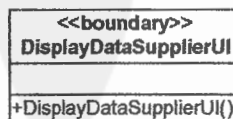
Method

- + DeleteSupplierUI()

Konstruktor class `ASCM.Purchasing.DeleteSupplierUI`.

Buat instance baru tanpa atribut terdefinisi.

2.2.9.12 Class `ASCM.Purchasing.DisplayDataSupplierUI`



Gambar 2. 44 Class `ASCM.Purchasing.DisplayDataSupplierUI`

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk menampilkan data supplier.

Atribut

-

Method

- + DisplayDataSupplierUI ()

Konstruktor class `ASCM.Purchasing.DisplayDataSupplierUI`. Buat instance baru tanpa atribut terdefinisi.

2.2.9.13 Class `ASCM.Purchasing.EditHargaBahanBakuUI`

<code><<boundary>></code>
<code>EditHargaBahanBakuUI</code>
<code>-bbm : <<control>></code>
<code>BahanBakuManager</code>
<code>+EntriHargaBahanBakuUI()</code>

Gambar 2. 44 Class `ASCM.Purchasing.EditHargaBahanBakuUI`

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk mengubah harga bahan baku.

Atribut

- - bbm : BahanBakuManager

Merepresentasikan object control class untuk pengelolaan bahan baku.

Method

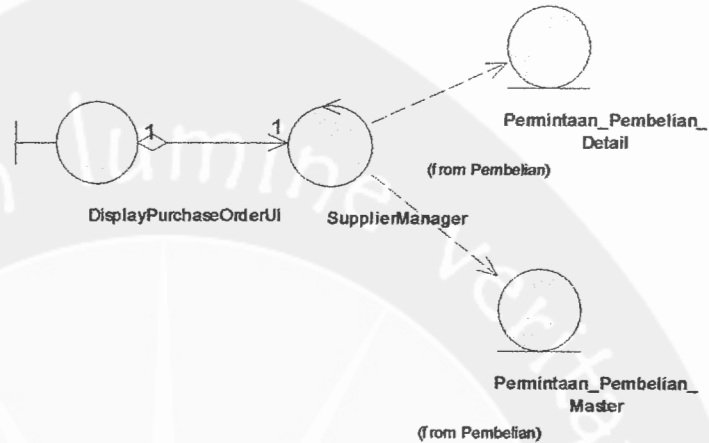
- + EditBahanBakuUI ()

Konstruktor class `ASCM.Purchasing.EditBahanBakuUI`. Buat instance baru tanpa atribut terdefinisi.

2.2.10 Package Supplier (ASCM.Supplier)

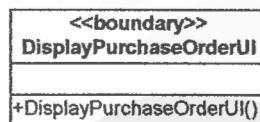
Package ini menyediakan class-class yang diperlukan untuk melakukan pengelolaan supplier.

2.2.10.1 Class diagram package ASCM.Supplier



Gambar 2.46 Class Diagram Package ASCM.Supplier

2.2.10.2 Class ASCM.Supplier.SupplierUI



Gambar 2. 47 Class ASCM.Supplier.SupplierUI

Deskripsi

Class ini merepresentasikan class boundary/GUI untuk menampilkan dan mencetak permintaan pembelian.

Atribut

-

Method

- + DisplayPurchaseOrderUI ()

Konstruktor

class

ASCM.Purchasing.DisplayPurchaseOrderUI.

Buat

instance baru tanpa atribut terdefinisi.

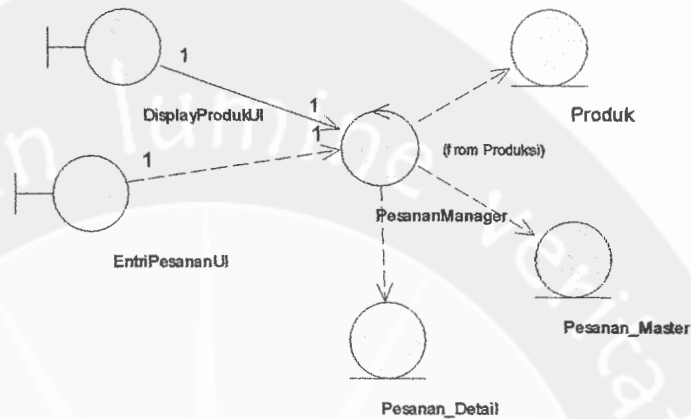
Program Studi Teknik Informatika	DPPL-ASCaMaDa	112/ 277
----------------------------------	---------------	----------

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika

2.2.11 Package Distributor (ASCM.Distributor)

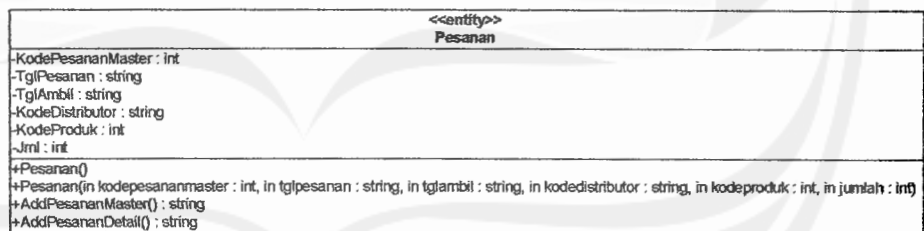
Package ini menyediakan class-class yang diperlukan untuk melakukan pengelolaan distributor.

2.2.11.1 Class diagram package ASCM.Distributor



Gambar 2.46 Class Diagram Package ASCM.Distributor

2.2.11.2 Class ASCM.Distributor.Pesanan



Gambar 2. 47 Class ASCM.Distributor.Pesanan

Deskripsi

Class ini merepresentasikan data pesanan pada sistem ASCaMaDa.

Atribut

- - kodePesananMaster : int

Merepresentasikan kode pesanan master.

Program Studi Teknik Informatika	DPPL-ASCaMaDa	113/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- - TglPesanan : string
Merepresentasikan tanggal pesanan.
- - TglAmbil : string
Merepresentasikan tanggal rencana ambil.
- - KodeDistributor : string
Merepresentasikan id distributor.
- - Jml : int
Merepresentasikan jumlah pesanan.

Method

- + Pesanan()
Konstruktor class ASCM.Distributor.Pesanan. Buat instance baru tanpa atribut terdefinisi.
- + Pesanan(kodepesananmaster : int, tglpesanan : string, tglambil : string, kodedistributor : string, kodeproduk : string, jumlah : int)
Konstruktor class ASCM.Distributor.Pesanan. Buat instance baru dengan atribut terdefinisi.
- + AddPesananMaster() : String
Menambah data pesanan ke tabel data_pesanan_master.
Parameters :
-
Returns:
"sukses" jika data berhasil ditambahkan.
- + AddPesananDetail() : String
Menambah data pesanan ke tabel data_pesanan_detail.
Parameters :
-
Returns:
"sukses" jika data berhasil ditambahkan.

Program Studi Teknik Informatika	DPPL-ASCaMaDa	114/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2.2.11.3 Class ASCM.Distributor.PesananManager

<<control>> PesananManager
+PesananManager() +AddPesananMaster(in kodedistributor : string, in tglpesanan : string, in tglambil : string) : string +AddPesananDetail(in kodepesananmaster : int, in kodeproduk : int, in jumlah : int) : string

Gambar 2. 48 Class ASCM.Distributor.PesananManager

Deskripsi

Class ini merepresentasikan control class untuk pengelolaan data pesanan pada sistem ASCaMaDa.

Atribut

-

Method

- + PesananManager ()

Konstruktor class ASCM.Distributor.PesananManager.

Buat instance baru tanpa atribut terdefinisi.

- + AddPesananMaster (kodedistributor : string, tglpesanan : string, tglambil : string) : String

Menambahkan data pesanan ke tabel data_pesanan_master.

Parameters :

kodedistributor - kode distributor

tglpesanan - tanggal pesanan

tglambil - tanggal rencana ambil

Returns:

“sukses” jika data berhasil ditambahkan ke database.

- + AddPesananDetail(kodepesananmaster : string, kodeproduk : int, jumlah : int) : String

Menambahkan data pesanan ke tabel data_pesanan_detail.

Parameters :

Program Studi Teknik Informatika	DPPL-ASCaMaDa	115/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

kodepesananmaster - kode pesanan master

kodeproduk - kode produk

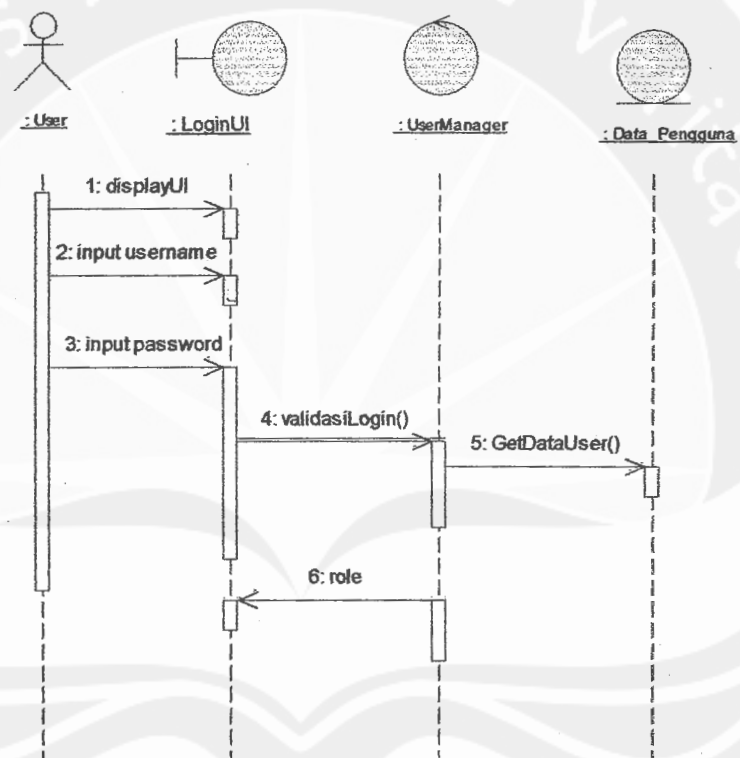
jumlah - jumlah produk

Returns:

“sukses” jika data berhasil ditambahkan ke database.

2.3 Realisasi Use Case

2.3.1 Use Case : Login



Gambar 2.49 Design Sequence Diagram Use Case Login

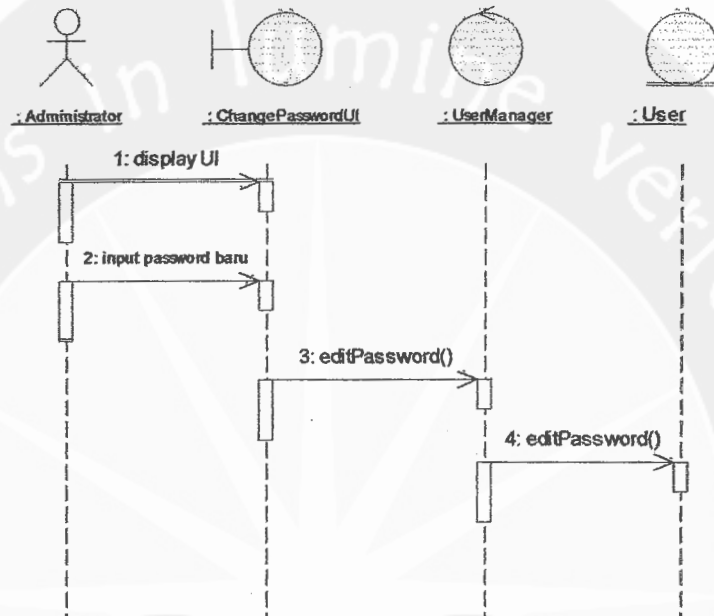
Flow of events :

1. User menampilkan antarmuka untuk login yaitu boundary class LoginUI. Kemudian user memasukkan nama user dan password.
2. LoginUI memanggil fungsi validasiLogin() milik control class UserManager yang di dalamnya juga dipanggil fungsi GetDataUser() milik entity class User untuk mengambil data user

dari basis data yang kemudian diproses untuk memvalidasi masukan nama dan password.

3. UserManager mengembalikan role dari user hasil proses validasi ke class boundary LoginUI.

2.3.2 Use Case : Ubah Password

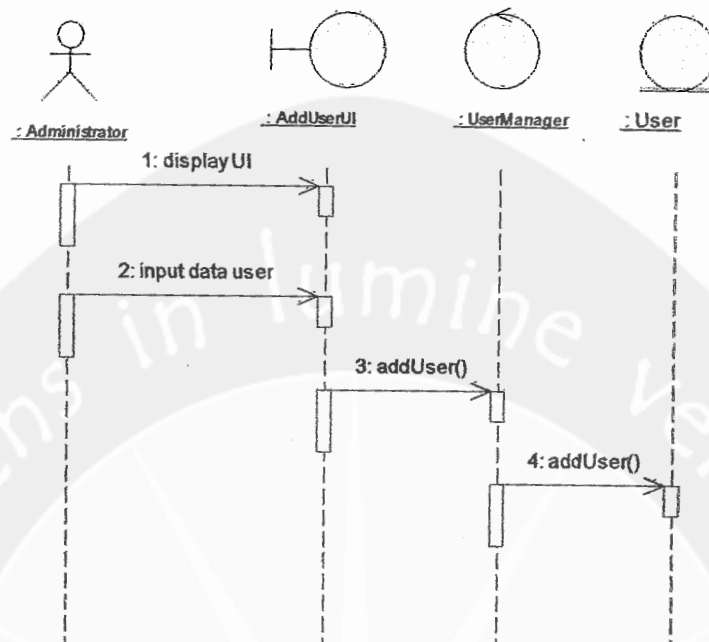


Gambar 2.50 Design Sequence Diagram Use Case Ubah Password

Flow of events :

1. User menampilkan antarmuka untuk mengubah password yaitu boundary class ChangePasswordUI, kemudian user memasukkan password lama dan password baru.
2. ChangePasswordUI memanggil fungsi UpdateUser() milik control class UserManager yang di dalamnya juga dipanggil fungsi UpdateUser() milik entity class User untuk mengubah password lama di basis data ke password yang baru dimasukkan oleh user.

2.3.3 Use Case : Add User

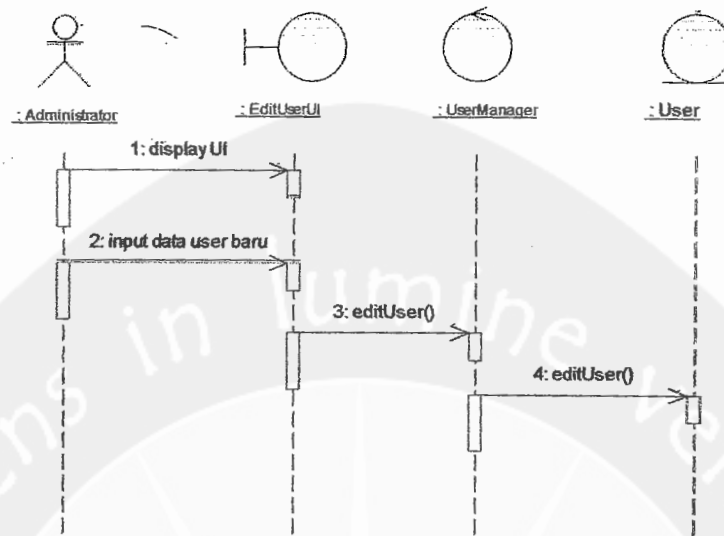


Gambar 2.51 Design Sequence Diagram Use Case Add User

Flow of events :

1. User menampilkan antarmuka untuk menambah data user baru ke basis data yaitu boundary class AddUserUI. Kemudian memasukkan detail data user baru, yaitu id user, nama, password dan role untuk user.
2. Setelah itu class AddUserUI memanggil fungsi `addUserAccount()` dimana di dalamnya juga memanggil fungsi `addUserAccount()` milik entity class User.
3. `UserManager()` menyimpan data user baru ke basisdata.

2.3.4 Use Case : Edit User

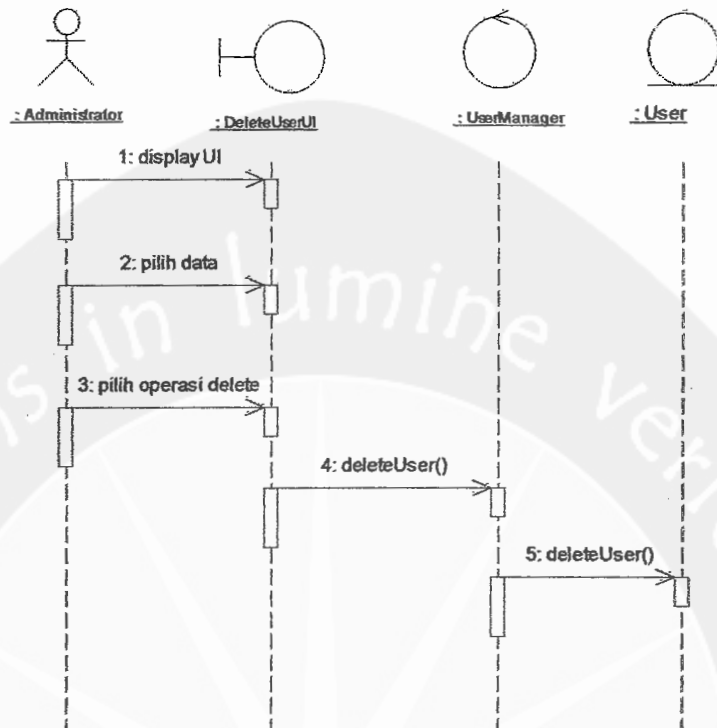


Gambar 2.52 Design Sequence Diagram Use Case Edit User

Flow of events :

1. User menampilkan antarmuka untuk mengubah data user di basis data yaitu boundary class `EditUserUI`. Kemudian user memasukkan kunci pencarian data user yang ingin diubah datanya.
2. `EditUserUI` menampilkan data user yang ingin dicari dengan memanggil fungsi `GetDataUserById()` milik control class `UserManager` yang di dalamnya juga dipanggil fungsi `GetDataUserById()` milik entity class `User`.
3. User mengubah data user yang telah ditampilkan, lalu data user akan disimpan dengan cara `EditUserUI` memanggil fungsi `updateUserAccount()` milik control class `UserManager` dimana didalamnya juga dilakukan pemanggilan fungsi `updateUserAccount()` milik entity class `User`.

2.3.5 Use Case : Delete User

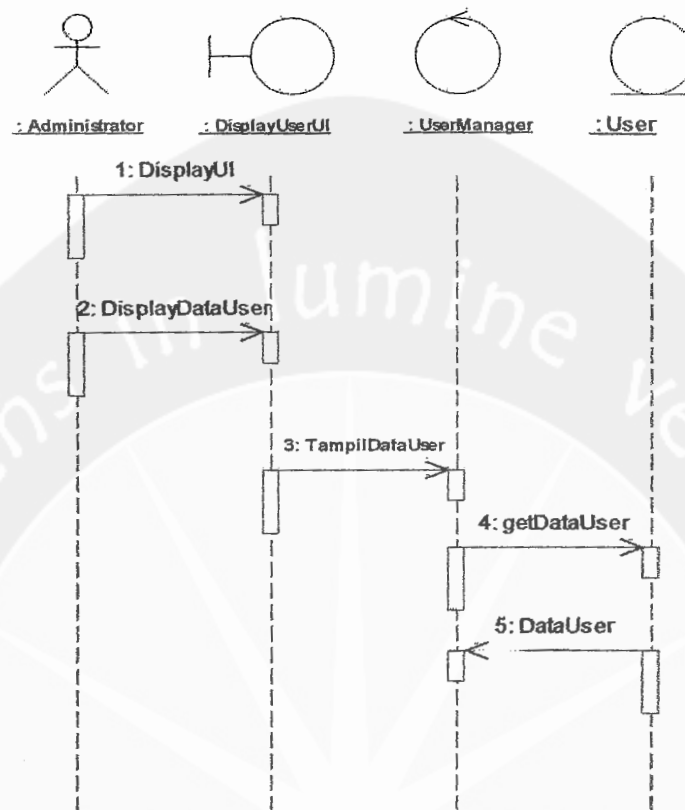


Gambar 2.53 Design Sequence Diagram Use Case Delet User

Flow of events :

1. User menampilkan antarmuka untuk menghapus data user dari basis data yaitu boundary class DeleteUserUI. Kemudian user memasukkan kunci pencarian data user yang ingin diubah datanya.
2. DeleteUserUI menampilkan data user yang ingin dicari dengan memanggil fungsi GetDataUserById() milik control class UserManager yang di dalamnya juga dipanggil fungsi GetDataUserById() milik entity class User.
3. User menghapus data user yang telah ditampilkan dimana EditUserUI memanggil fungsi DeleteUser() milik control class UserManager dimana di dalamnya juga dilakukan pemanggilan fungsi DeleteUser() milik entity class User.

2.3.6 Use Case : Display User

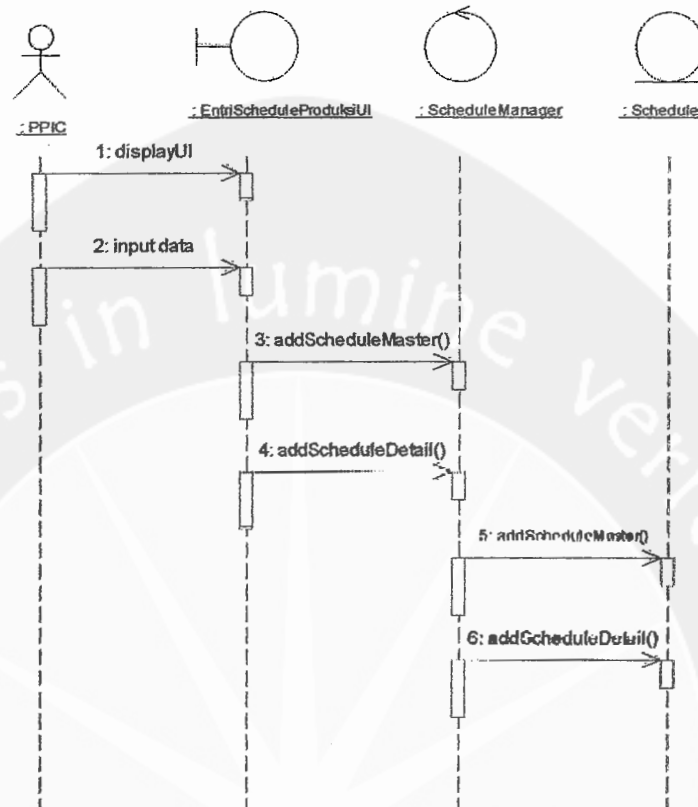


Gambar 2.54 Design Sequence Diagram Use Case Display User

Flow of events :

1. User menampilkan antarmuka untuk menampilkan data user dari basis data yaitu boundary class `DisplayUserUI`. Kemudian user memasukkan kunci pencarian data user yang ingin diubah datanya.
2. `DisplayUserUI` menampilkan data user yang ingin dicari dengan memanggil fungsi `GetDataUser()` milik control class `UserManager` yang di dalamnya juga dipanggil fungsi `GetDataUser()` milik entity class `User`.

2.3.7 Use Case : Entri Schedule Produksi

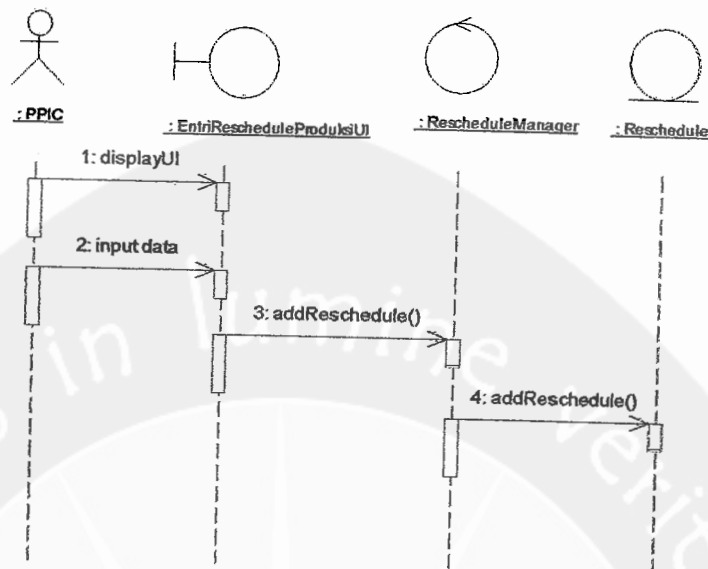


Gambar 2.55 Design Sequence Diagram Use Case Entri Schedule Produksi

Flow of events :

1. User menampilkan antarmuka untuk menambah data Schedule Produksi yaitu boundary class `EntriScheduleProduksiUI`.
2. `EntriScheduleProduksiUI` memanggil fungsi `addScheduleMaster()` milik control class `ScheduleManager` dimana di dalamnya dipanggil fungsi `addScheduleMaster()` milik entity class `Schedule` dan fungsi `addScheduleDetail()` dimana di dalamnya dipanggil fungsi `addScheduleDetail()` milik entity class `Schedule`.
3. User memasukkan detail data schedule produksi

2.3.8 Use Case : EntriReschedule

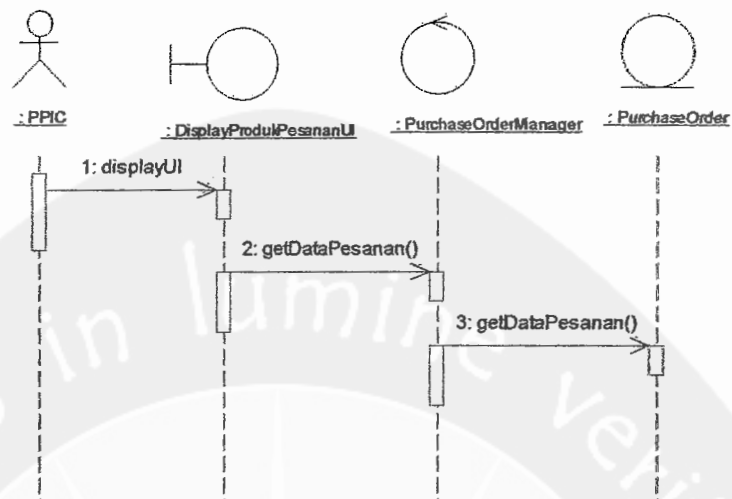


Gambar 2.56 Design Sequence Diagram Use Case Entri Reschedule

Flow of events :

1. User menampilkan antarmuka untuk menambah data Reschedule Produksi yaitu boundary class `EntriRescheduleProduksiUI`.
2. `EntriRescheduleProduksiUI` memanggil fungsi `updateScheduleMaster()` milik control class `RescheduleManager` dimana di dalamnya dipanggil fungsi `updateScheduleMaster()` milik entity class `Schedule`. Kemudian memanggil fungsi `addRescheduleMaster()` dimana di dalamnya dipanggil fungsi `addRescheduleMaster()` dan memanggil fungsi `addRescheduleDetail()` dimana di dalamnya dipanggil fungsi `addRescheduleDetail()` milik entity class `Reschedule`.
3. User memasukkan detail data Reschedule produksi

2.3.9 Use Case : Display Pesanan

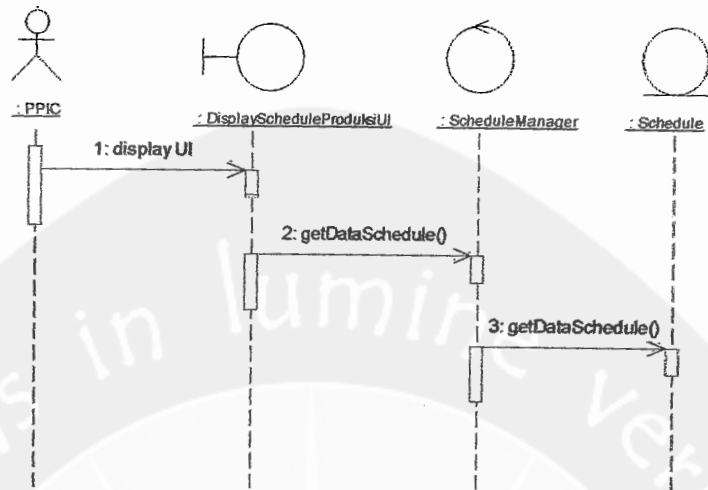


Gambar 2.57 Design Sequence Diagram Use Case Display Pesanan

Flow of events :

1. User menampilkan antarmuka untuk menampilkan data pesanan yaitu boundary class DisplayProdukPesananUI.
2. DisplayProdukPesananUI memanggil fungsi getDataPesanan() milik control class PPICManager untuk menampilkan id transaksi pasang baru ke class DisplayProdukPesananUI dimana di dalamnya dipanggil fungsi getDataPesanan() milik entity class PPIC.

2.3.10 Use Case : Display Schedule Produksi

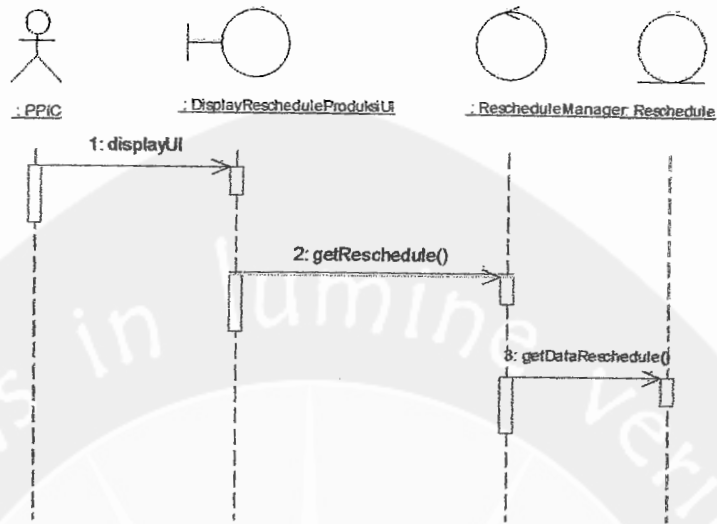


Gambar 2.58 Design Sequence Diagram Use Case Display Schedule Produksi

Flow of events :

1. User menampilkan antarmuka untuk menampilkan data schedule yaitu boundary class `DisplayScheduleProduksiUI`.
2. `DisplayScheduleProduksiUI` memanggil fungsi `getDataScheduleMaster()` dan `getDataScheduleDetail()` milik control class `ScheduleManager` untuk menampilkan data schedule ke class `DisplayScheduleProduksiUI` dimana di dalamnya dipanggil fungsi `getDataScheduleMaster()` dan `getDataScheduleDetail` milik entity class `Schedule`.

2.3.11 Use Case : Display Reschedule Produksi

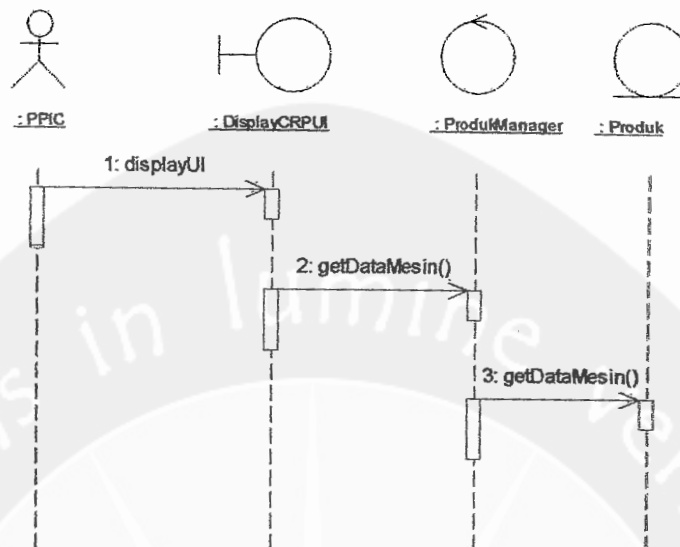


Gambar 2.59 Design Sequence Diagram Use Case Display Reschedule Produksi

Flow of events :

1. User menampilkan menampilkan antarmuka untuk menampilkan data reschedule yaitu boundary class DisplayRescheduleProduksiUI.
2. DisplayRescheduleProduksiUI memanggil fungsi getDataReschedule () milik control class RescheduleManager untuk menampilkan data reschedule ke class DisplayRescheduleProduksiUI dimana di dalamnya dipanggil fungsi getDataReschedule () milik entity class Reschedule

2.3.12 Use Case : Display Capacity Requirement Planning

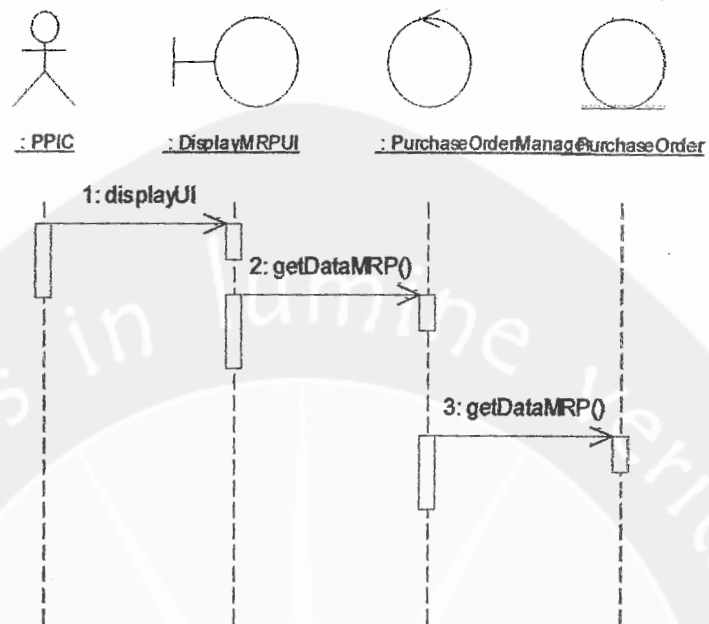


Gambar 2.60 Design Sequence Diagram Display Use Case CRP

Flow of events :

1. User menampilkan antarmuka untuk menampilkan data CRP yaitu boundary class DisplayCRPUI.
2. DisplayCRPUI memanggil fungsi getDataMesin() milik control class PPICManager untuk menampilkan data CRP ke class DisplayCRPUI dimana di dalamnya dipanggil fungsi getDataMesin() milik entity class PPIC.

2.3.13 Use Case : Display Material Requirement Planning

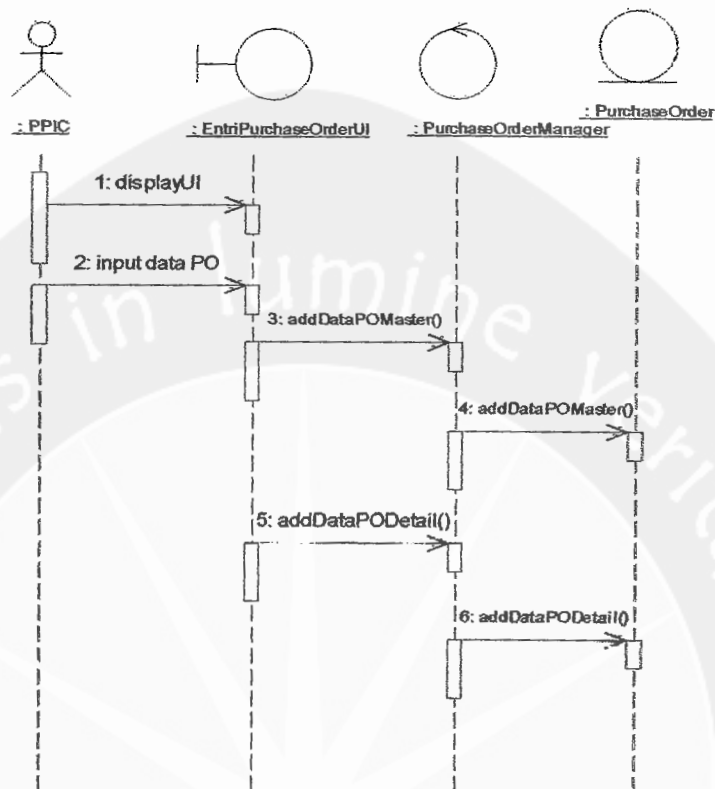


Gambar 2.61 Design Sequence Diagram Use Case Display MRP

Flow of events :

1. User menampilkan antarmuka untuk menampilkan data MRP yaitu boundary class `DisplayMRPUI`.
2. `DisplayMRPUI` memanggil fungsi `getDataMRP()` milik control class `PPICManager` untuk menampilkan data *material requirement planning* ke class `DisplayMRPUI` dimana di dalamnya dipanggil fungsi `getDataMRP()` milik entity class `PPIC`.

2.3.14 Use Case : Entri Purchase Order

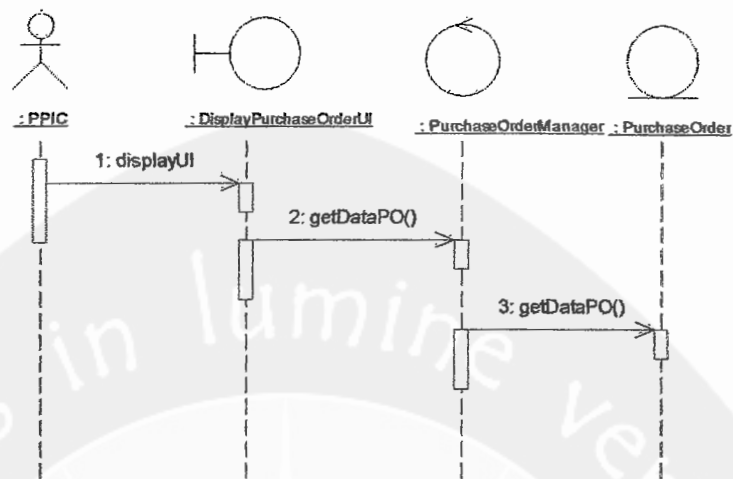


Gambar 2.62 Design Sequence Diagram Use Case Entri Purchase Order

Flow of events :

1. User menampilkan antarmuka untuk melakukan entri data Purchase Order yaitu boundary class EntriPurchaseOrderUI lalu melakukan input data PurchaseOrder.
2. EntriPurchaseOrderUI memanggil fungsi addDataPOMaster() milik control class PPICManager dimana di dalamnya dipanggil fungsi addDataPOMaster() milik entity class PPIC.
3. Kemudian memanggil fungsi addDataPODetail() milik control class PPICManager dimana di dalamnya dipanggil fungsi addDataPODetail() .

2.3.15 Use Case : Display Purchase Order

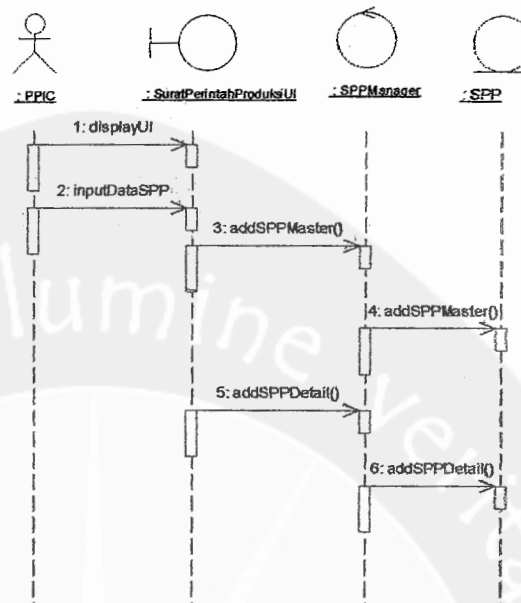


Gambar 2.63 Design Sequence Diagram Use Case Display Purchase Order

Flow of events :

1. User menampilkan antarmuka untuk menampilkan data Purchase Order yaitu boundary class DisplayPurchaseOrderUI.
2. DisplayPurchaseOrderUI memanggil fungsi getDataPO () milik control class PPICManager untuk menampilkan data PPIC ke class DisplayPurchaseOrderUI dimana di dalamnya dipanggil fungsi getDataPO () milik entity class PPIC.

2.3.16 Use Case : Surat Perintah Produksi

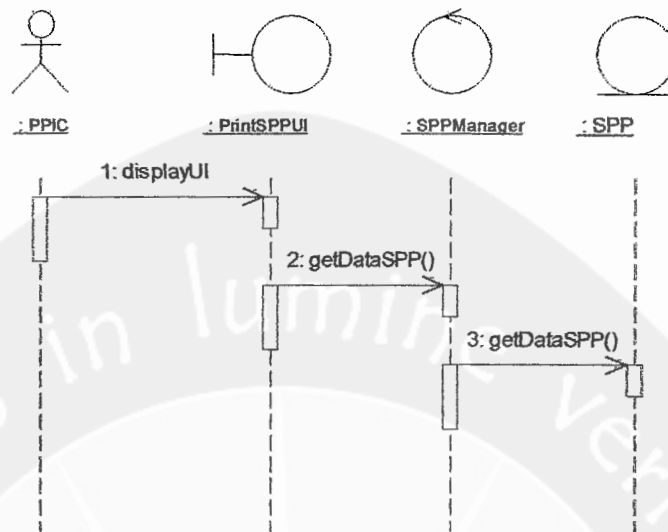


Gambar 2.64 Design Sequence Diagram Use Case Surat Perintah Produksi

Flow of events :

1. User menampilkan antarmuka untuk melakukan entri data Surat Perintah Produksi yaitu boundary class SuratPerintahProduksiUI lalu melakukan input data Surat Perintah Produksi.
2. SuratPerintahProduksiUI memanggil fungsi addSPPMaster() milik control class SPPManager dimana di dalamnya dipanggil fungsi addSPPMaster() milik entity class SPP.
3. SuratPerintahProduksiUI memanggil fungsi addSPPDetail() milik control class SPPManager dimana di dalamnya dipanggil fungsi addSPPDetail() milik entity class SPP.

2.3.17 Use Case : Print SPP

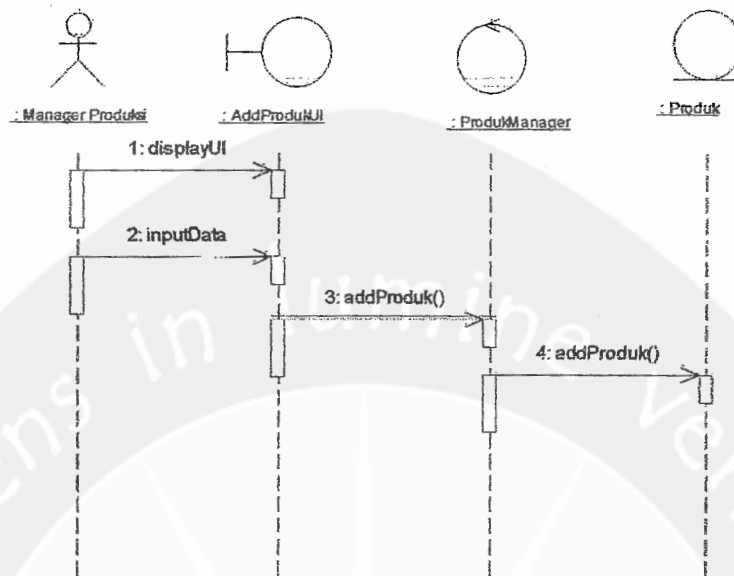


Gambar 2.65 Design Sequence Diagram Use Case Print SPP

Flow of events :

1. User menampilkan antarmuka untuk menampilkan data pesanan yaitu boundary class `PrintSPPUI`.
2. `PrintSPPUI` memanggil fungsi `getDataSPP()` milik control class `ScheduleManager` untuk menampilkan data Surat Perintah Produksi ke class `PrintSPPUI` dimana di dalamnya dipanggil fungsi `getDataSPP()` milik entity class `SPP`.

2.3.18 Use Case : AddProduk

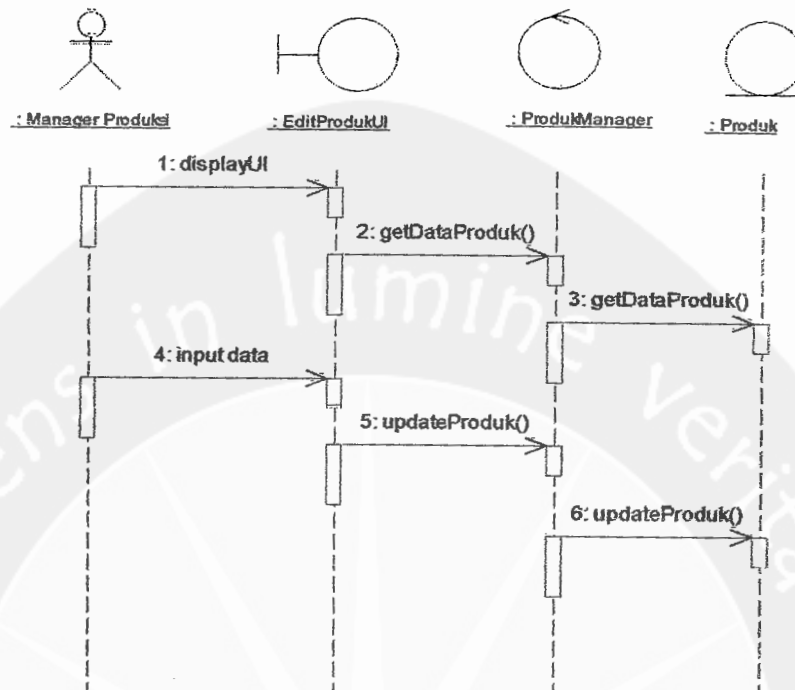


Gambar 2.66 Design Sequence Diagram Use Case Add Produk

Flow of events :

1. User menampilkan antarmuka untuk melakukan add data produk yaitu boundary class AddProdukUI .
2. User memasukkan data produk baru.
3. AddProdukUI memanggil fungsi addProduk () milik control class ProdukManager dimana didalamnya dipanggil fungsi addProduk () milik entity class Produk.

2.3.19 Use Case : EditProduk

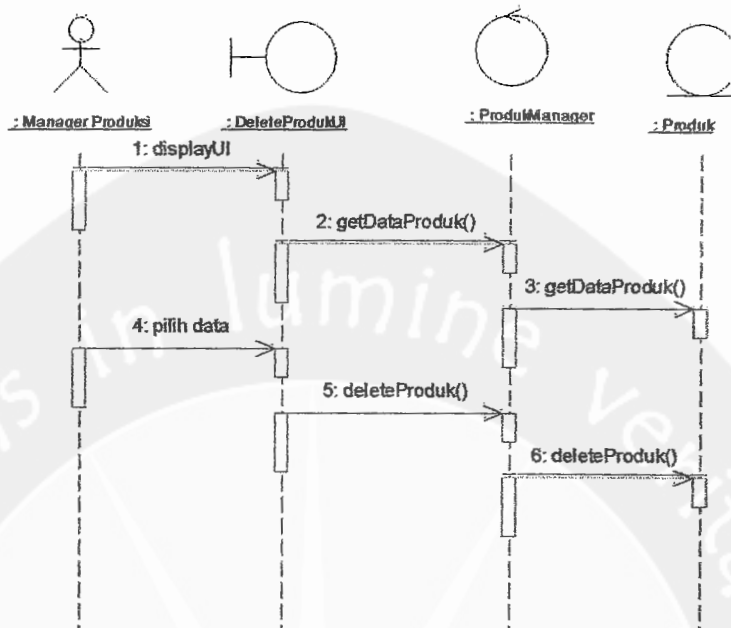


Gambar 2.67 Design Sequence Diagram Use Case Edit Produk

Flow of events :

4. User menampilkan antarmuka untuk melakukan edit data produk yaitu boundary class EditProdukUI.
5. EditProdukUI memanggil fungsi getDataProduk() milik control class ProdukManager untuk menampilkan data produk ke class EditProdukUI dimana di dalamnya dipanggil fungsi getDataProduk() milik entity class Produk.
6. User memasukkan data produk baru.
7. EditProdukUI memanggil fungsi updateProduk() milik control class ProdukManager dimana didalamnya dipanggil fungsi updateProduk() milik entity class Produk.

2.3.20 Use Case : Delete Produk

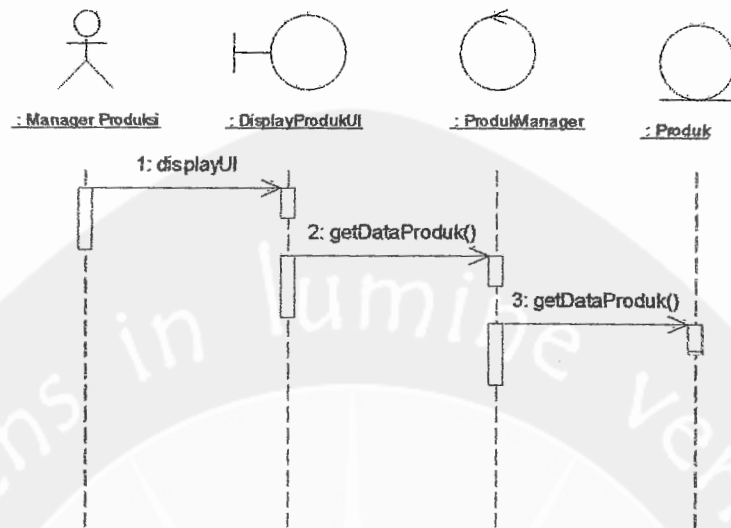


Gambar 2.68 Design Sequence Diagram Use Case Delete Produk

Flow of events :

1. User menampilkan antarmuka untuk menghapus data produk dari basis data yaitu boundary class DeleteProdukUI.
2. DeleteProdukUI memanggil fungsi getDataProduk() milik control class ProdukManager dimana didalamnya dipanggil fungsi getDataProduk() milik entity class Produk.
3. User memilih data yang akan dihapus.
4. DeleteProdukUI memanggil fungsi deleteProduk() milik control class ProdukManager dimana didalamnya dipanggil fungsi deleteProduk() milik entity class Produk.

2.3.21 Use Case : Display Produk

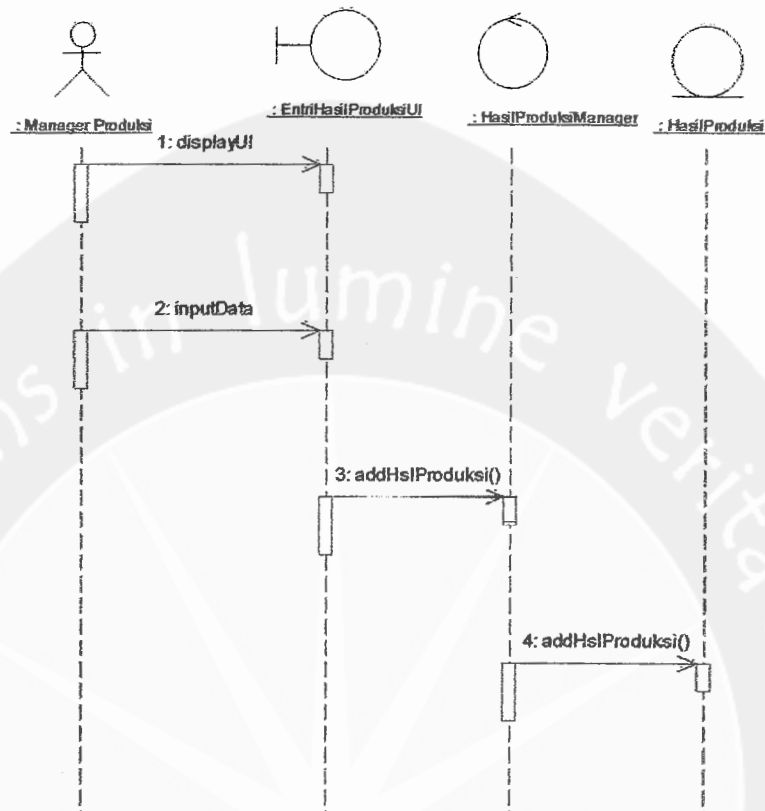


Gambar 2.69 Design Sequence Diagram Use Case Display Produk

Flow of events :

1. User menampilkan antarmuka untuk menampilkan data produk yaitu boundary class DisplayProdukUI.
2. DisplayProdukUI memanggil fungsi getDataProduk() milik control class ProdukManager dimana didalamnya dipanggil fungsi getDataProduk() milik entity class Produk.

2.3.22 Use Case : Entri Hasil Produksi

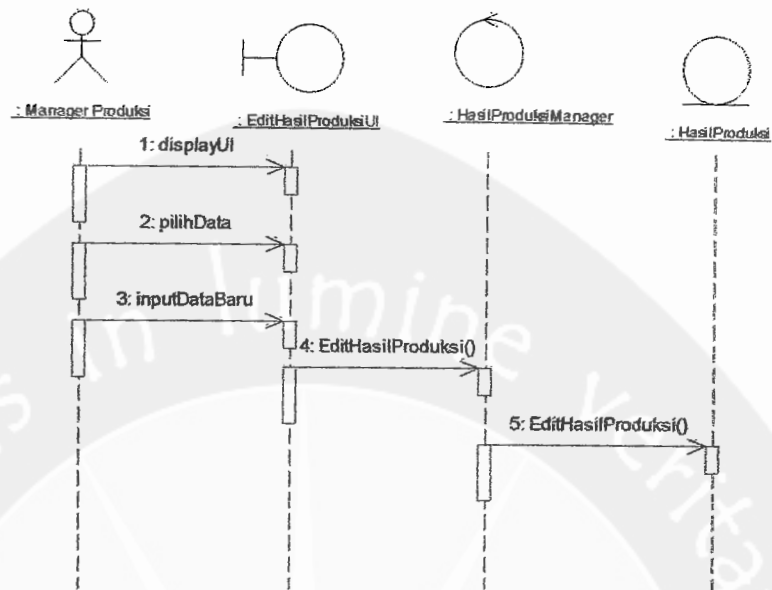


Gambar 2.70 Design Sequence Diagram Use Case Entri Hasil Produksi

Flow of events :

1. User menampilkan antarmuka untuk melakukan entri data hasil produksi yaitu boundary class `EntriHasilProduksiUI`.
2. `EntriHasilProduksiUI` memanggil fungsi `addHasilProduksi()` milik control class `HasilProduksimanager` dimana didalamnya dipanggil fungsi `addHasilProduksi()` milik entity class `HasilProduksi`.

2.3.23 Use Case : Edit Hasil Produksi

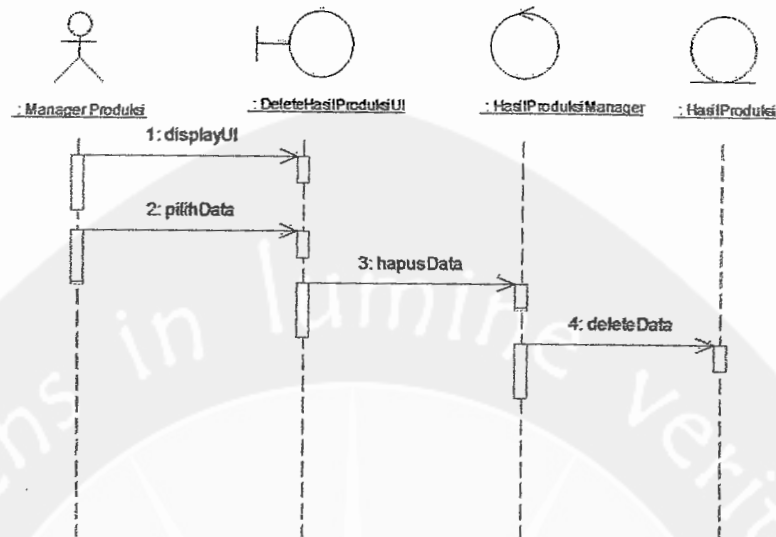


Gambar 2.71 Design Sequence Diagram Use Case Edit Hasil Produksi

Flow of events :

1. User menampilkan antarmuka untuk mengubah data hasil produksi yaitu boundary class `EditHasilProduksiUI`, user memilih data lalu memasukkan data baru.
2. `EditHasilProduksiUI` memanggil fungsi `editHasilProduksi()` milik control class `HasilProduksiManager` dimana didalamnya dipanggil fungsi `editHasilProduksi()` milik entity class `HasilProduksi`.
3. User memilih data dan kemudian memasukkan data baru.

2.3.24 Use Case : Delete Hasil Produksi

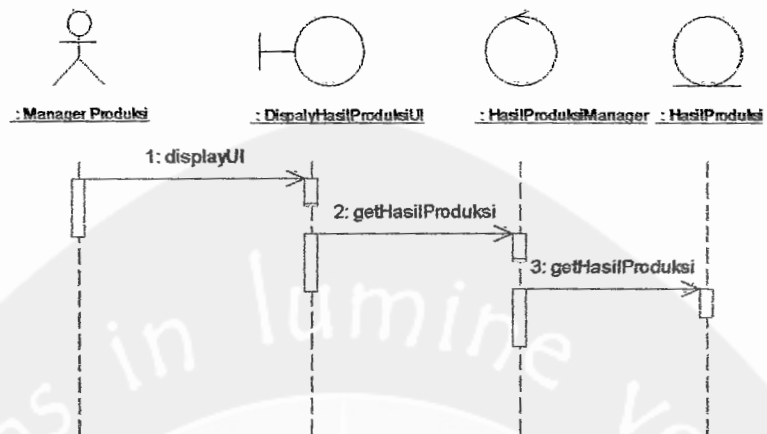


Gambar 2.72 Design Sequence Diagram Use Case Delete Hasil Produksi

Flow of events :

1. User menampilkan antarmuka untuk menampilkan data hasil produksi yaitu boundary class `DeleteHasilProduksiUI`, user memilih data yang akan dihapus.
2. `DeleteHasilProduksiUI` memanggil fungsi `hapusData()` milik control class `HasilProduksiManager` dimana didalamnya dipanggil fungsi `deleteData()` milik entity class `HasilProduksi`.

2.3.25 Use Case : Display Hasil Produksi

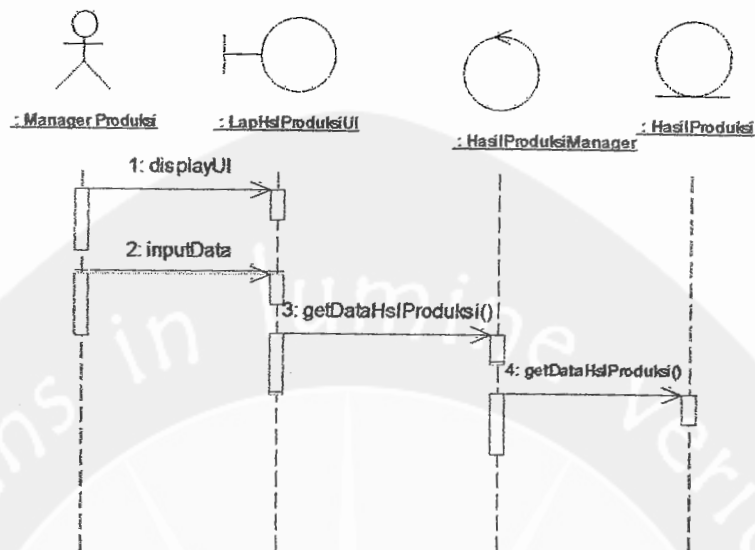


Gambar 2.73 Design Sequence Diagram Use Case Display Hasil Produksi

Flow of events :

1. User menampilkan antarmuka untuk menampilkan data hasil produksi yaitu boundary class `DisplayHasilProduksiUI`.
2. `DisplayHasilProduksiUI` memanggil fungsi `getHasilProduksi()` milik control class `HasilProduksiManager` dimana didalamnya dipanggil fungsi `getHasilProduksi()` milik entity class `HasilProduksi`.

2.3.26 Use Case : Laporan Hasil Produksi

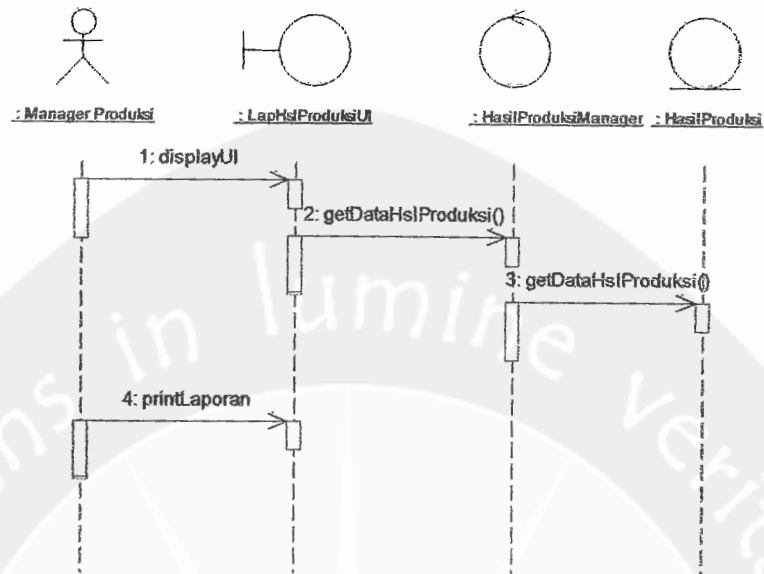


Gambar 2.74 Design Sequence Diagram Use Case Laporan Hasil Produksi

Flow of events :

1. User menampilkan antarmuka untuk menampilkan laporan hasil produksi yaitu boundary class `LapHslProduksiUI`.
2. User memasukkan data berupa tanggal.
3. `LapHslProduksiUI` memanggil fungsi `getDataHslProduksiByTgl()` milik `HasilProduksiManager` control class.
4. `HasilProduksiManager` memanggil fungsi `getDataHslProduksiByTgl()` milik `HasilProduksi` entity class.

2.3.27 Use Case : Print Laporan Hasil Produksi

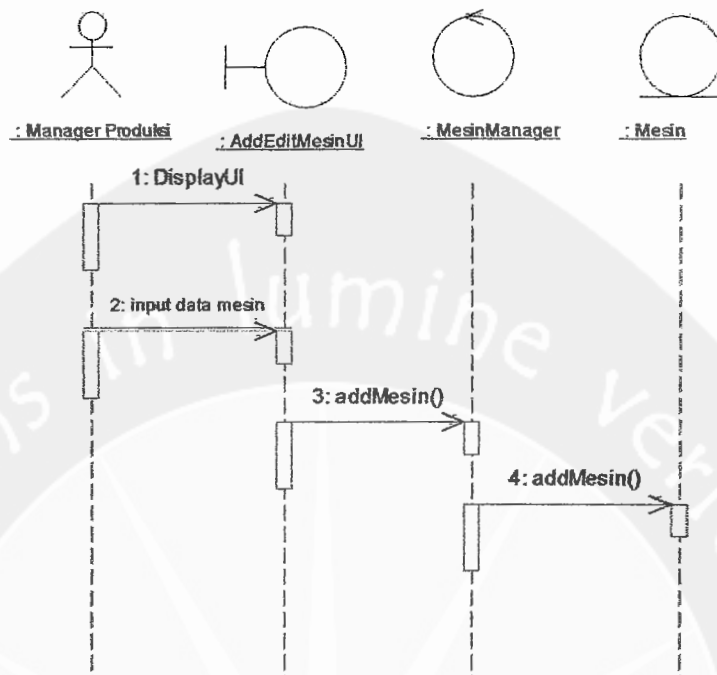


Gambar 2.75 Design Sequence Diagram Use Case Print Laporan Hasil Produksi

Flow of events :

1. User menampilkan antarmuka untuk menampilkan laporan hasil produksi yaitu boundary class `LapHslProduksiUI`.
2. `LapHslProduksiUI` memanggil fungsi `getDataHslProduksi()` milik control class `HasilProduksiManager`.
3. `HasilProduksiManager` memanggil fungsi `getDataHslProduksi()` milik entity class `HasilProduksi`.
4. User memasukkan data berupa perintah mencetak.
5. `LapHslProduksiUI` memanggil fungsi `printLaporan()`.

2.3.28 Use Case : Add Mesin

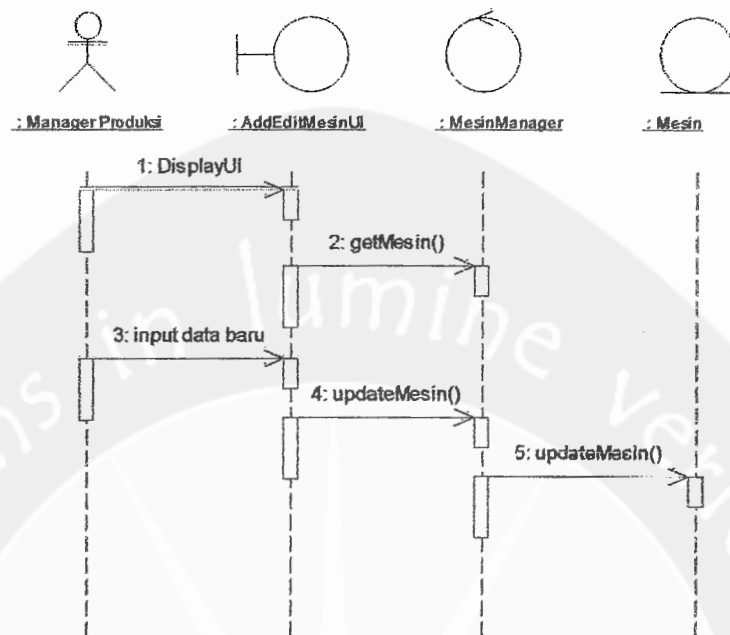


Gambar 2.76 Design Sequence Diagram Use Case Add Mesin

Flow of events :

1. User menampilkan antarmuka untuk melakukan tambah data mesin yaitu boundary class AddEditMesinUI.
2. User memasukkan data mesin yang baru.
3. AddEditMesinUI memanggil fungsi addMesin() milik control class MesinManager dimana didalamnya dipanggil fungsi addMesin() milik entity class Mesin.

2.3.29 Use Case : Edit Mesin

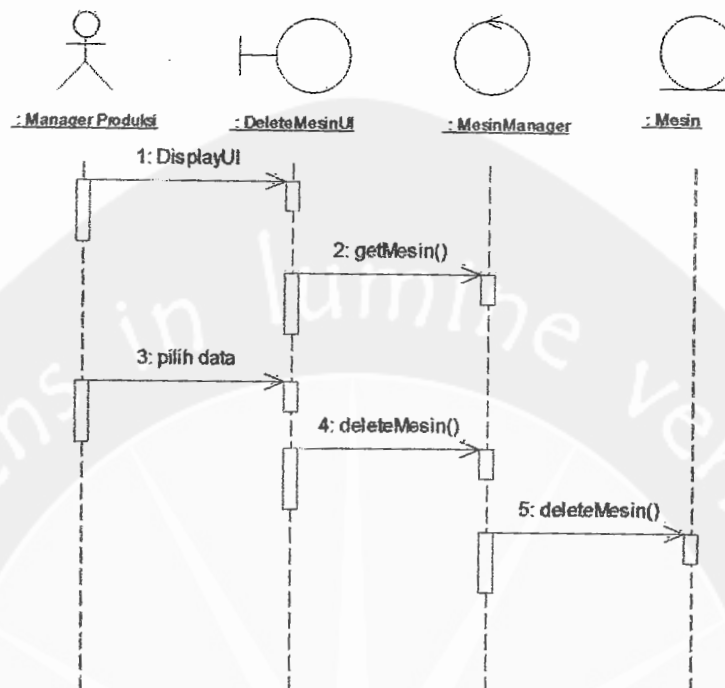


Gambar 2.77 Design Sequence Diagram Use Case Edit Mesin

Flow of events :

1. User menampilkan antarmuka untuk menampilkan data hasil produksi yaitu boundary class AddEditMesinUI..
2. AddEditMesinUI memanggil fungsi getMesin() milik control class MesinManager dimana didalamnya dipanggil fungsi getMesin() milik entity class Mesin.
3. User memasukkan data mesin yang baru.
4. AddEditMesinUI memanggil fungsi updateMesin() milik control class MesinManager dimana didalamnya dipanggil fungsi updateMesin() milik entity class Mesin.

2.3.30 Use Case : Delete Mesin

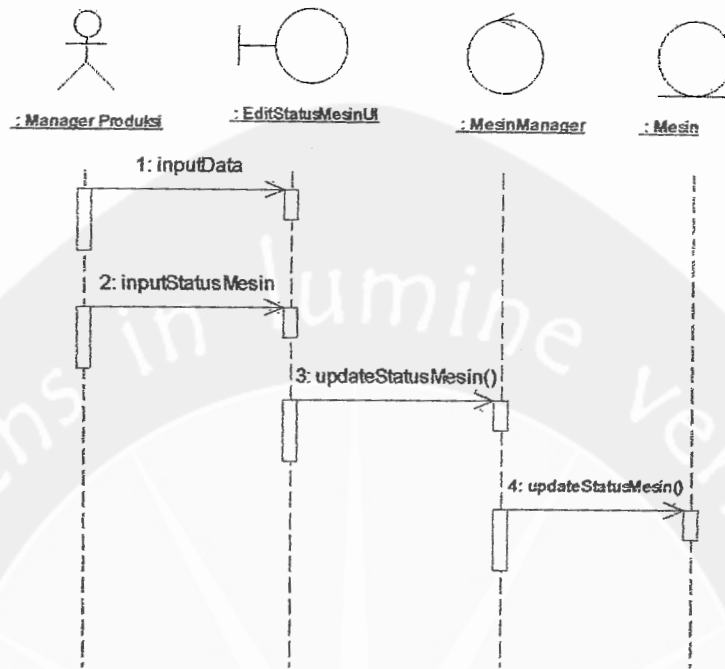


Gambar 2.78 Design Sequence Diagram Use Case Delete Mesin

Flow of events :

1. User menampilkan antarmuka untuk melkakukan hapus mesin yaitu boundary class DeleteMesinUI..
2. DeleteMesinUI memanggil fungsi getMesin() milik control class MesinManager dimana didalamnya dipanggil fungsi getMesin() milik entity class Mesin.
3. User memilih data mesin yang akan dihapus.
4. DeleteMesinUI memanggil fungsi deleteMesin() milik control class MesinManager dimana didalamnya dipanggil fungsi deleteMesin() milik entity class Mesin.

2.3.31 Use Case : Edit Status Mesin

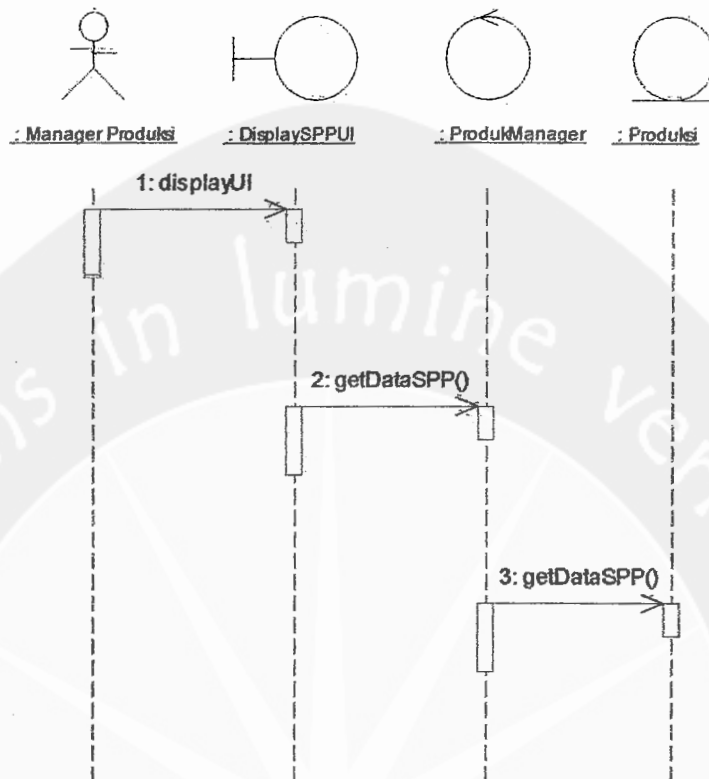


Gambar 2.79 Design Sequence Diagram Use Case Edit Status Mesin

Flow of events :

1. User menampilkan antarmuka untuk melakukan perubahan status mesin yaitu boundary class `EditStatusMesinUI`.
2. User memasukkan data status mesin yang baru.
3. `EditStatusMesinUI` memanggil fungsi `updateStatusMesin()` milik control class `MesinManager` dimana didalamnya dipanggil fungsi `updateStatusMesin()` milik entity class `Mesin`.

2.3.32 Use Case : Display SPP

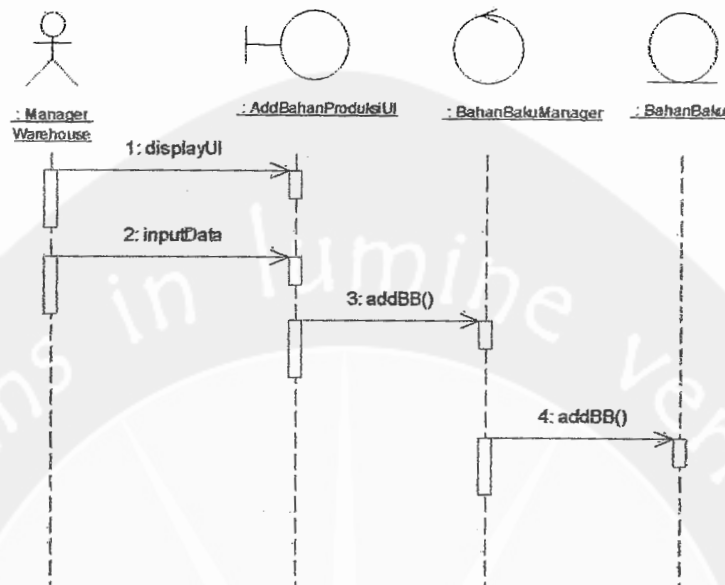


Gambar 2.80 Design Sequence Diagram Use Case Display SPP

Flow of events :

1. User menampilkan antarmuka untuk menampilkan data SPP yaitu boundary class DisplaySPPUI.
2. User memasukkan data tanggal.
3. DisplaySPPUI memanggil fungsi getDataSPP() milik control class ProdukManager dimana didalamnya dipanggil fungsi getDataSPP() milik entity class Produk. ...

2.3.33 Use Case : Add Bahan Produksi

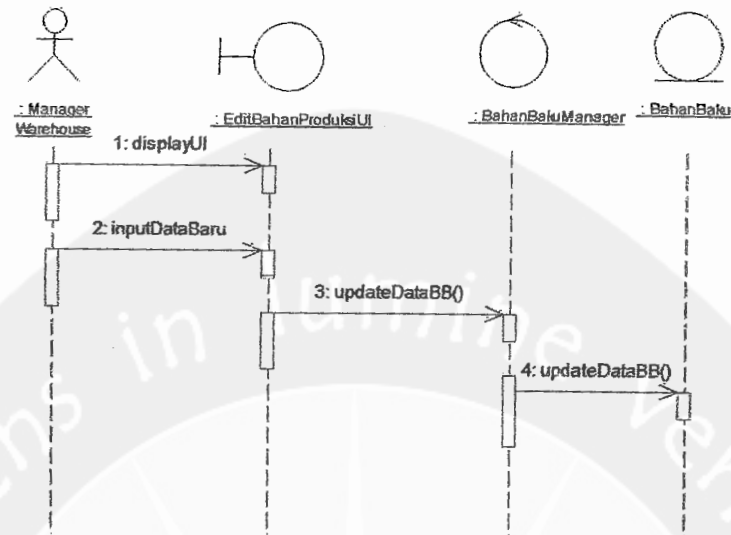


Gambar 2.81 Design Sequence Diagram Use Case Add Bahan Produksi

Flow of events :

1. User menampilkan antarmuka untuk menambah data bahan baku produksi baru ke basis data yaitu boundary class AddBahanProduksiUI.
2. User memasukkan data bahan baku yang baru.
3. AddBahanProduksiUI memanggil fungsi addBB () milik control class BahanBakuManager.
4. BahanBakuManager memanggil fungsi addBB () milik entity class BahanBaku.

2.3.34 Use Case : Edit Bahan Produksi

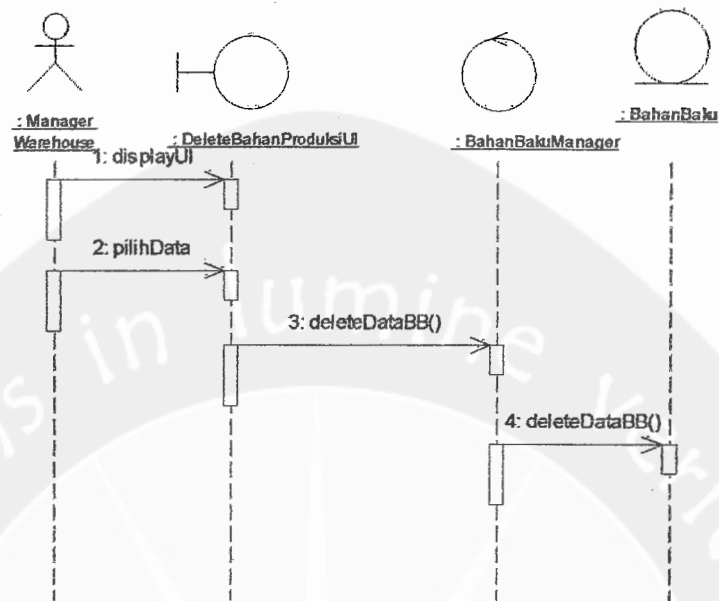


Gambar 2.82 Design Sequence Diagram Use Case Edit Bahan Produksi

Flow of events :

1. User menampilkan antarmuka untuk mengubah data bahan baku yaitu boundary class `EditBahanProduksiUI` lalu memasukkan data bahan baku yang baru.
2. User memasukkan data bahan baku yang baru
3. `EditBahanProduksiUI` memanggil fungsi `updateDataBB()` milik control class `BahanBakuManager`.
4. `BahanBakuManager` memanggil fungsi `updateDataBB()` milik entity class `BahanBaku`.

2.3.35 Use Case : Delete Bahan Produksi

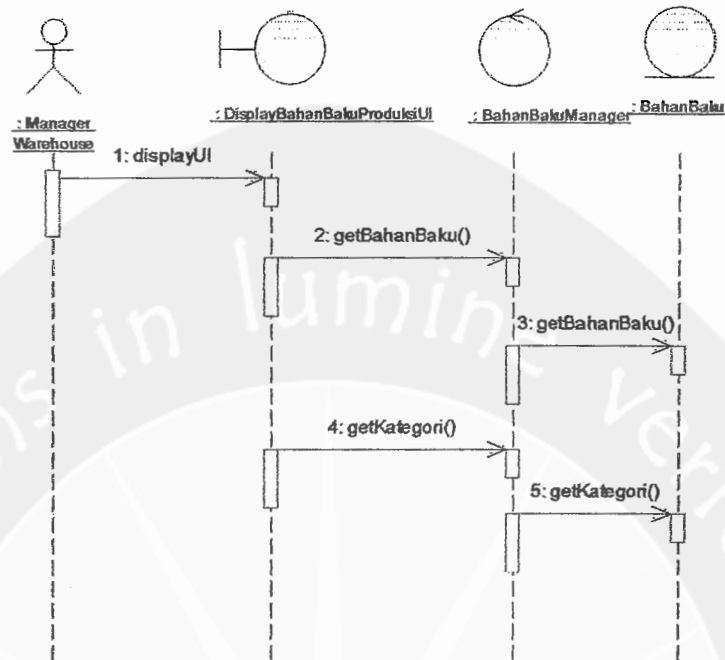


Gambar 2.83 Design Sequence Diagram Use Case Delete Bahan Produksi

Flow of events :

1. User menampilkan antarmuka untuk menghapus data bahan baku yaitu boundary class DeleteBahanProduksiUI.
2. User memilih data yang akan dihapus.
3. DeleteBahanProduksiUI memanggil fungsi deleteDataBB() milik control class BahanBakuManager.
4. BahanBakuManager memanggil fungsi deleteDataBB() milik entity class BahanBaku.

2.3.36 Use Case : Display Bahan Produksi

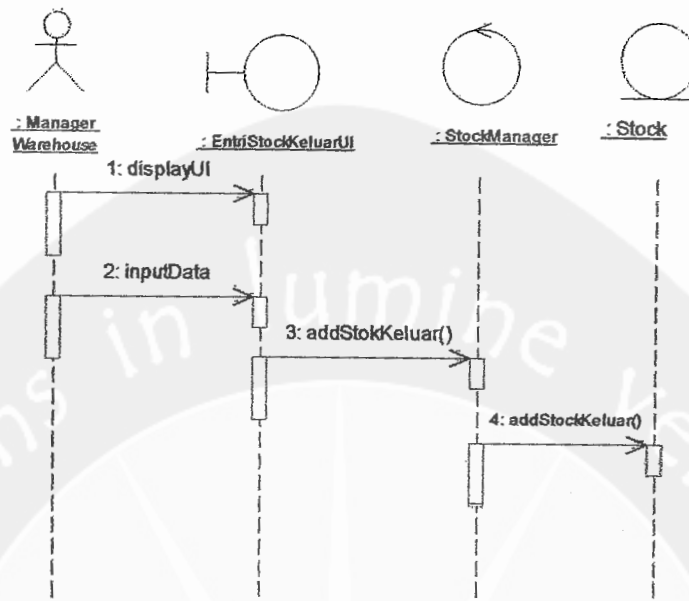


Gambar 2.84 Design Sequence Diagram Use Case Display Bahan Produksi

Flow of events :

1. User menampilkan antarmuka untuk menampilkan data bahan baku produksi yaitu boundary class `DisplayBahanBakuProduksiUI`.
2. `DisplayBahanBakuProduksiUI` memanggil fungsi `getBahanBaku()` milik control class `BahanBakuManager`.
3. `BahanBakuManager` memanggil fungsi `getBahanBaku()` milik entity class `BahanBaku`.
4. `DisplayBahanBakuProduksiUI` memanggil fungsi `getKategori()` milik control class `BahanBakuManager`.
5. `BahanBakuManager` memanggil fungsi `getKategori()` milik entity class `BahanBaku`.

2.3.37 Use Case : Entri Stock Keluar

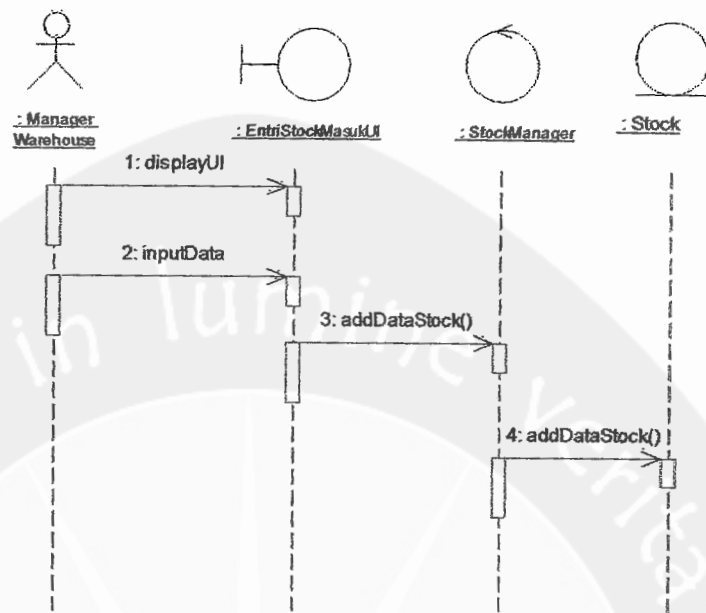


Gambar 2.85 Design Sequence Diagram Use Case Entri Stock Keluar

Flow of events :

1. User menampilkan antarmuka untuk melakukan entri data stock bahan baku keluar yaitu boundary class `EntriStockKeluarUI`.
2. User memasukkan data bahan baku keluar.
3. `EntriStockKeluarUI` memanggil fungsi `addStockKeluar()` milik control class `StockManager` dimana didalamnya dipanggil fungsi `addStockKeluar()` milik entity class `Stock`.

2.3.38 Use Case : Entri Stock Masuk

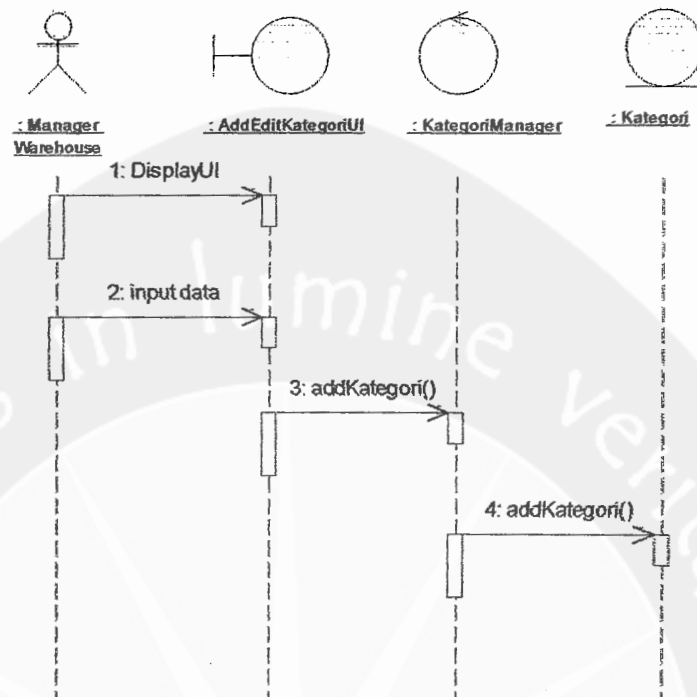


Gambar 2.86 Design Sequence Diagram Use Case Entri Stock Masuk

Flow of events :

1. User menampilkan antarmuka untuk melakukan entri data stock bahan baku keluar yaitu boundary class `EntriStockMasukUI`.
2. User memasukkan data bahan baku keluar.
3. `EntriStockMasukUI` memanggil fungsi `addDataStock()` milik control class `StockManager`.
4. `StockManager` memanggil fungsi `addDataStock()` milik entity class `Stock`.

2.3.39 Use Case : Add Kategori

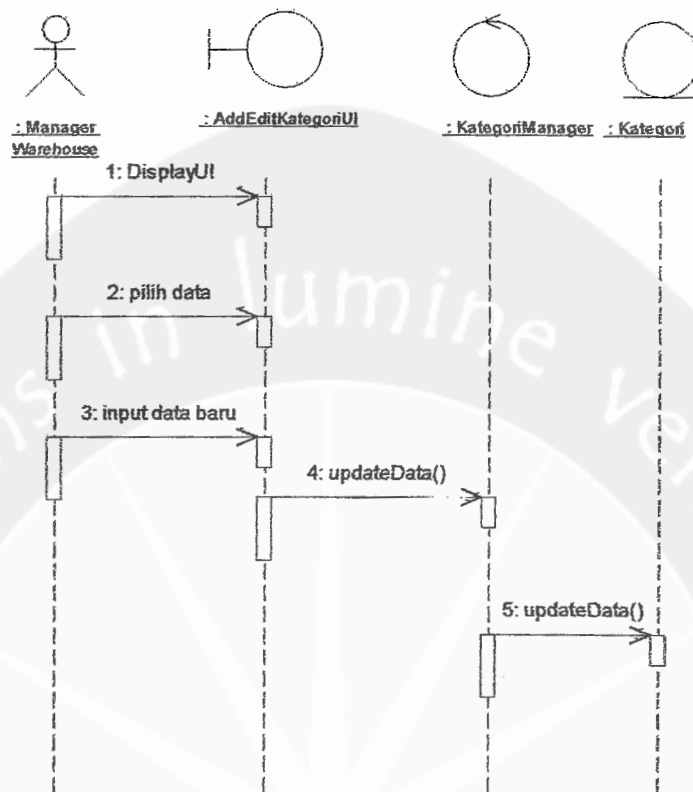


Gambar 2.87 Design Sequence Diagram Use Case Add Kategori

Flow of events :

1. User menampilkan antarmuka untuk menambah data kategori ke basis data yaitu boundary class AddEditKategoriUI.
2. User memasukkan data kategori baru.
3. AddEditKategoriUI memanggil fungsi addKategori() milik control class KategoriManager.
4. KategoriManager memanggil fungsi addKategori () milik entity class Kategori.

2.3.40 Use Case : Edit Kategori

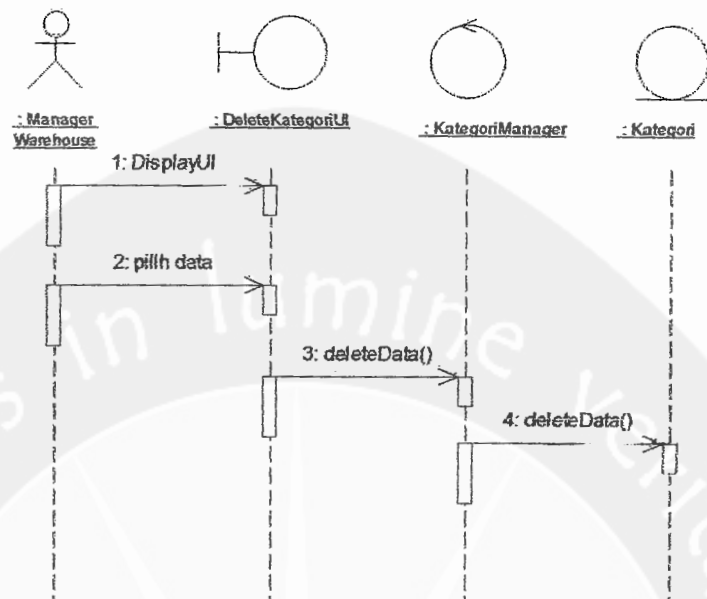


Gambar 2.88 Design Sequence Diagram Use Case Edit Kategori

Flow of events :

1. User menampilkan antarmuka untuk mengubah data kategori yaitu boundary class AddEditKategoriUI.
2. User memasukkan data kategori baru.
3. AddEditKategoriUI memanggil fungsi updateData() milik control class KategoriManager.
4. KategoriManager memanggil fungsi updateData() milik entity class Kategori.

2.3.41 Use Case : Delete Kategori

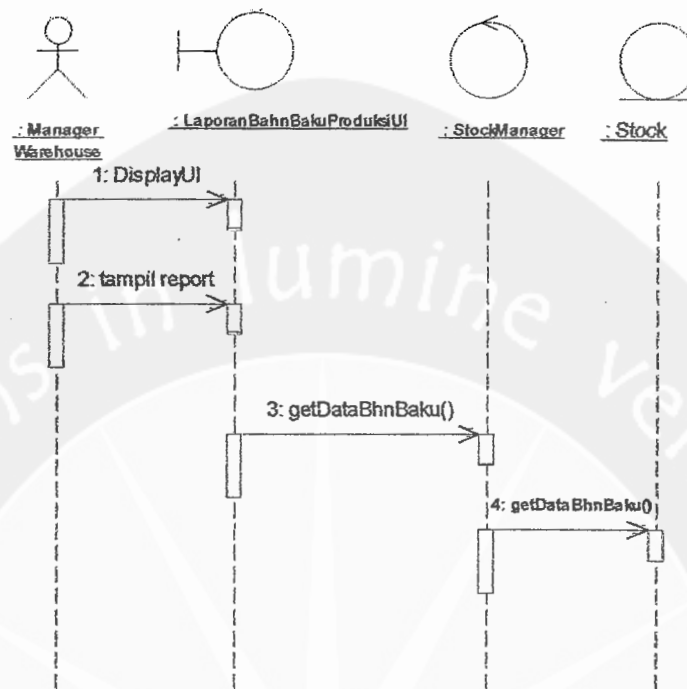


Gambar 2.89 Design Sequence Diagram Use Case Delete Kategori

Flow of events :

1. User menampilkan antarmuka untuk menghapus data kategori yaitu boundary class DeleteKategoriUI.
2. User memilih data kategori yang akan dihapus..
3. DeleteKategoriUI memanggil fungsi deleteData() milik control class KategoriManager.
4. KategoriManager memanggil fungsi deleteData() milik entity class Kategori.

2.3.42 Use Case : Laporan Bahan Baku Produksi

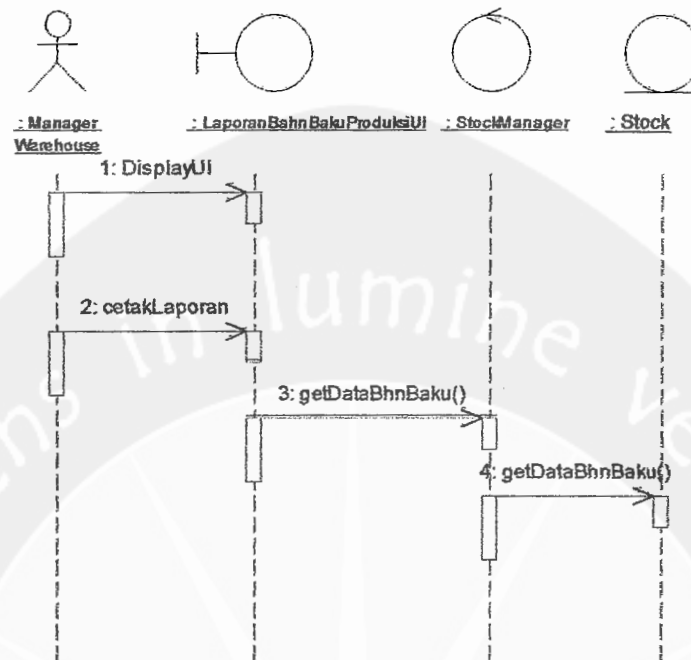


Gambar 2.90 Design Sequence Diagram Use Case Laporan Bahan Baku Produksi

Flow of events :

1. User menampilkan antarmuka untuk membuat laporan bahan baku yaitu boundary class LaporanBahanBakuProduksiUI.
2. LaporanBahanBakuProduksiUI memanggil fungsi getDataBhnBaku() milik control class StockManager.
3. StockManager memanggil fungsi getDataBhnBaku() milik entity class Stock.

2.3.43 Use Case : Print Laporan Bahan Baku Produksi

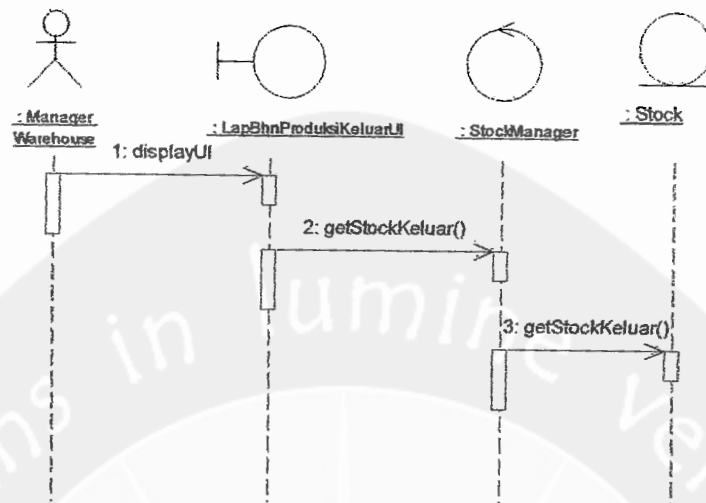


Gambar 2.91 Design Sequence Diagram Use Case Print Lap Bahan Baku Produksi

Flow of events :

1. User menampilkan antarmuka untuk mencetak laporan bahan baku yaitu boundary class LaporanBahanBakuProduksiUI dan meminta system mencetak laporan.
2. LaporanBahanBakuProduksiUI memanggil fungsi getDataBhnBaku() milik control class StockManager.
3. StockManager memanggil fungsi getDataBhnBaku() milik entity class Stock.
4. LaporanBahanBakuProduksiUI memanggil fungsi cetakLaporan().

2.3.44 Use Case : Laporan Bahan Baku Produksi Keluar

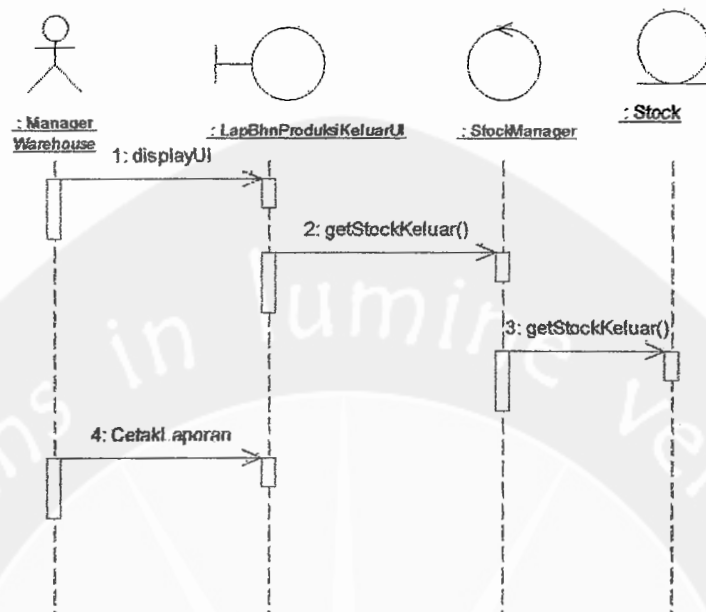


Gambar 2.92 Design Sequence Diagram Use Case Lap Bhn Baku Produksi Keluar

Flow of events :

1. User menampilkan antarmuka untuk membuat laporan bahan baku keluar yaitu boundary class LaporanBhnBakuProduksiKeluarUI.
2. LaporanBhnBakuProduksiKeluarUI memanggil fungsi getStockKeluar() milik control class StockManager.
3. StockManager memanggil fungsi getStockKeluar() milik entity class Stock.

2.3.45 Use Case : Print Laporan Bahan Baku Produksi Keluar

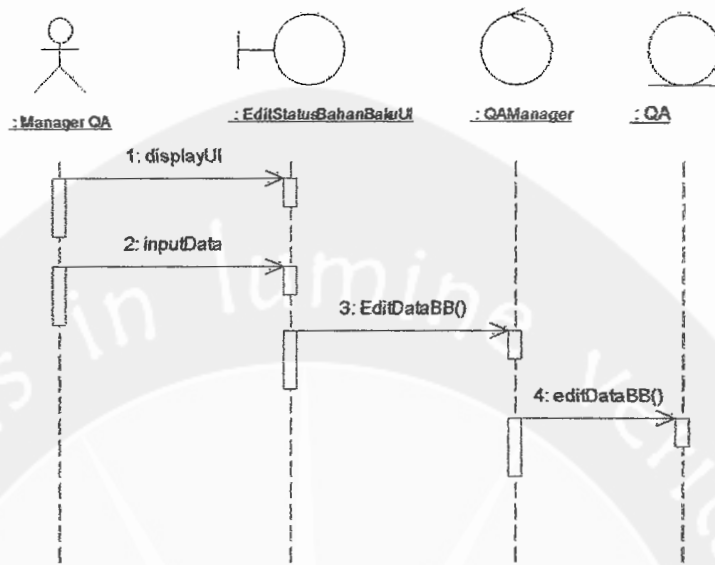


Gambar 2.93 Design Sequence Diagram Use Case Print Lap Bhn Baku Prod Keluar

Flow of events :

1. User menampilkan antarmuka untuk mencetak laporan bahan baku keluar yaitu boundary class LaporanBhnBakuProduksiKeluarUI.
2. LaporanBhnBakuProduksiKeluarUI memanggil fungsi getStockKeluar() milik control class StockManager.
3. StockManager memanggil fungsi getStockKeluar() milik entity class Stock.
4. User meminta system mencetak laporan
5. LaporanBhnBakuProduksiKeluarUI memanggil fungsi CetakLaporan().

2.3.46 Use Case : Edit Status Bahan Baku

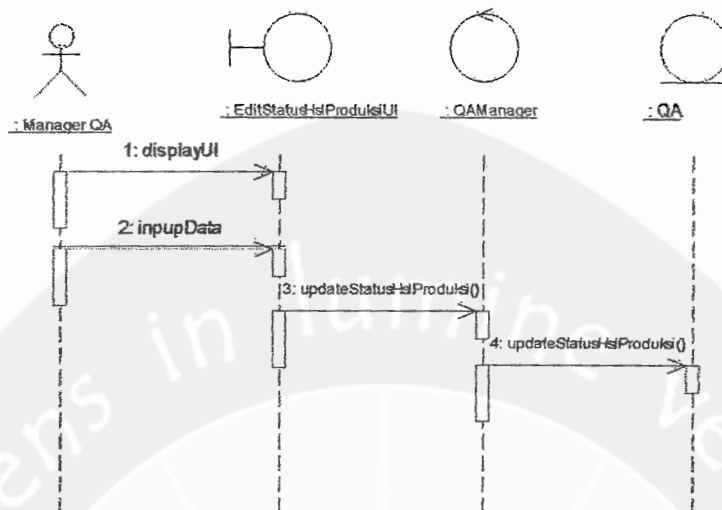


Gambar 2.94 Design Sequence Diagram Use Case Edit Status Bahan Baku

Flow of events :

1. User menampilkan antarmuka untuk mengubah status bahan baku produksi dan meng-update stock bahan baku yaitu boundary class EditStatusBahanBakuUI.
2. User memasukkan data bahan baku dan status.
3. EditStatusBahanBakuUI akan memanggil fungsi EditDataBB() milik control class QAManager.
4. QAManager memanggil fungsi updateStatusBB() dan EditDataBB() milik entity class QA.

2.3.47 Use Case : Edit Status Barang Produksi

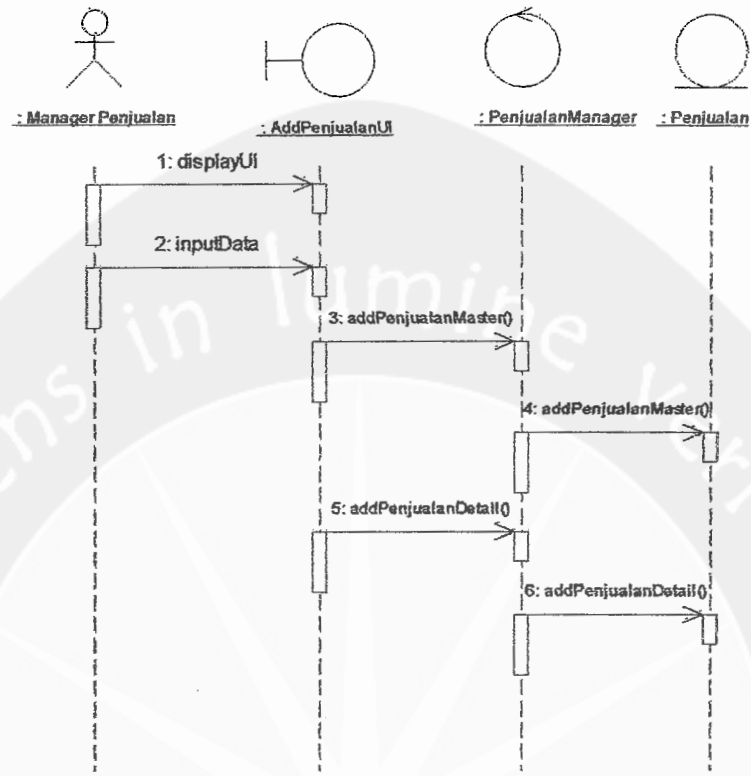


Gambar 2.95 Design Sequence Diagram Use Case Edit Status Barang Produksi

Flow of events :

1. User menampilkan antarmuka untuk mengubah status barang hasil produksi dan meng-update stock produk yaitu boundary class `EditStatusHslProduksiUI`.
2. User memasukkan data hasil produksi dan status.
3. `EditStatusHslProduksiUI` akan memanggil fungsi `updateStatusHslProduksi()` milik control class `QAManager`.
4. `QAManager` memanggil fungsi `updateStatusBB()` dan `updateStatusHslProduksi()` milik entity class `QA`.

2.3.48 Use Case : Add Penjualan

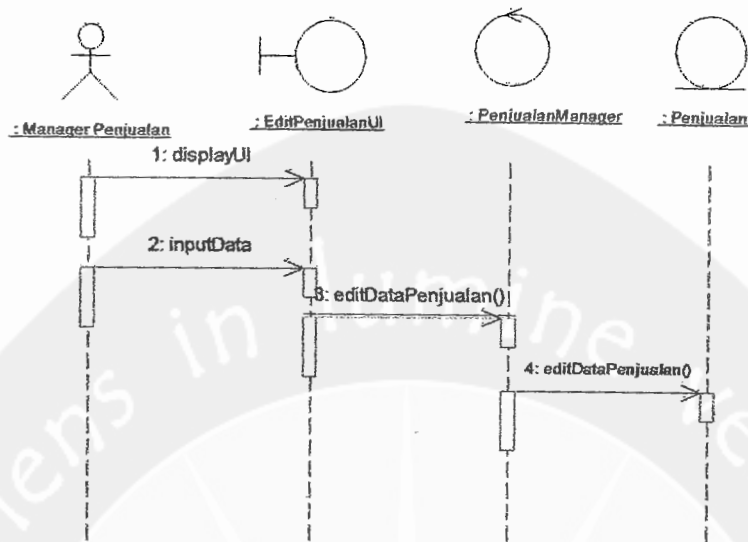


Gambar 2.96 Design Sequence Diagram Use Case Add Penjualan

Flow of events :

1. User menampilkan antarmuka untuk melakukan penjualan produk yaitu boundary class AddPenjualanUI
2. User memasukkan data penjualan.
3. AddPenjualanUI memanggil fungsi addPenjualanMaster() milik control class PenjualanManager dimana di dalamnya dipanggil fungsi addPenjualanMaster() milik entity class Penjualan.
4. AddPenjualanUI memanggil fungsi addPenjualanDetail() milik control class PenjualanManager dimana di dalamnya dipanggil fungsi addPenjualanDetail() dan updateStock() milik entity class Penjualan.

2.3.49 Use Case : Edit Penjualan

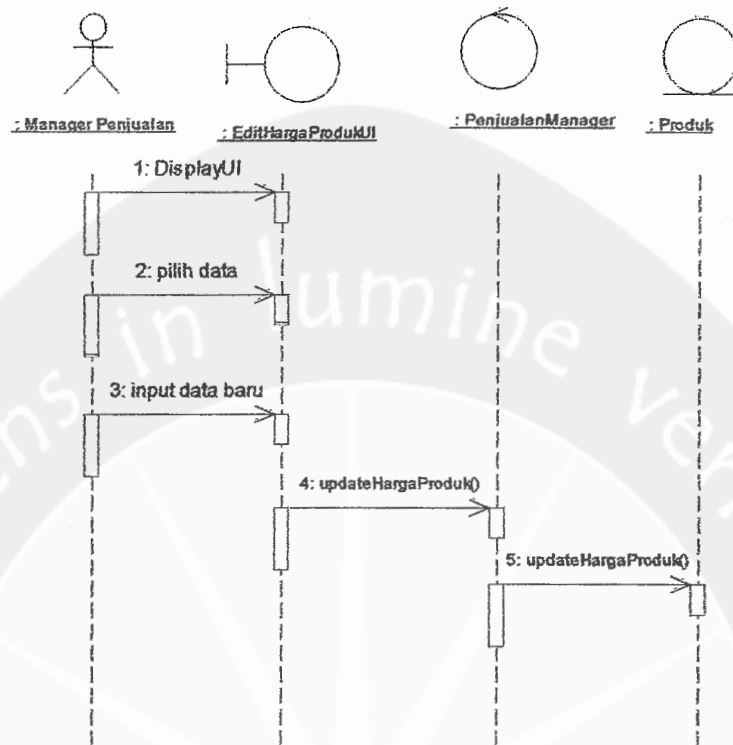


Gambar 2.97 Design Sequence Diagram Use Case Edit Penjualan

Flow of events :

1. User menampilkan antarmuka untuk melakukan edit penjualan produk yaitu boundary class EditPenjualanUI
2. User memasukkan data penjualan.
3. EditPenjualanUI memanggil fungsi editDataPenjualan() milik control class PenjualanManager dimana di dalamnya dipanggil fungsi editDataPenjualan () milik entity class Penjualan.

2.3.50 Use Case : Edit Harga Produk

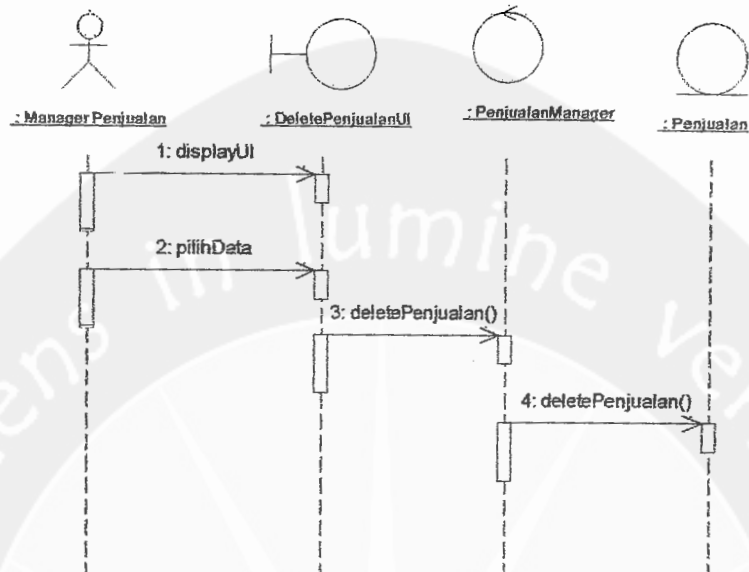


Gambar 2.98 Design Sequence Diagram Use Case Edit Harga Produk

Flow of events :

1. User menampilkan antarmuka untuk mengubah harga produk yaitu boundary class `EditHargaProdukUI()`.
2. `EditHargaProdukUI` akan menampilkan data produk dengan cara memanggil fungsi `getDataProduk()` milik control class `PenjualanManager` dimana di dalamnya dipanggil fungsi `getDataProduk()` milik entity class `Penjualan`.
3. User memasukkan data harga produk.
4. `EditHargaProdukUI` memanggil fungsi `updateHargaProduk()` milik control class `PenjualanManager` dimana di dalamnya dipanggil fungsi `updateHargaProduk()` milik entity class `Penjualan`.

2.3.51 Use Case : Delete Penjualan

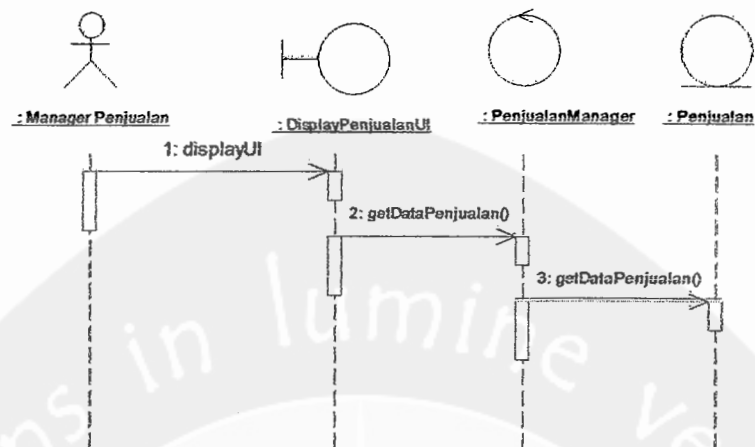


Gambar 2.99 Design Sequence Diagram Use Case Delete Penjualan

Flow of events :

1. User menampilkan antarmuka untuk melakukan delete penjualan produk yaitu boundary class DeletePenjualanUI.
2. User memilih data penjualan.
3. DeletePenjualanUI memanggil fungsi deletePenjualan() milik control class PenjualanManager dimana di dalamnya dipanggil fungsi deletePenjualan() milik entity class Penjualan.

2.3.52 Use Case : Display Penjualan

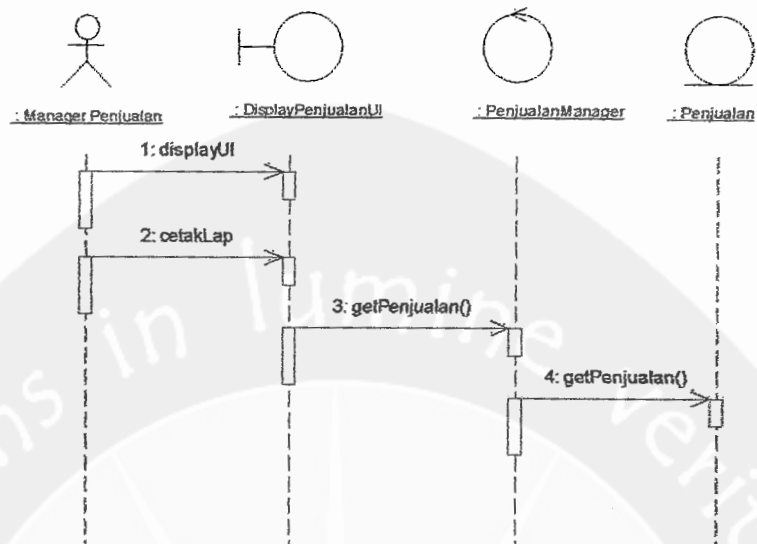


Gambar 2.100 Design Sequence Diagram Use Case Display Penjualan

Flow of events :

1. User menampilkan antarmuka untuk menampilkan data penjualan berdasarkan tanggal penjualan yaitu boundary class DisplayPenjualanUI.
2. DisplayPenjualanUI memanggil fungsi getDataPenjualan() milik control class PenjualanManager dimana di dalamnya dipanggil fungsi getDataPenjualan() dan getNamaProduk() milik entity class Penjualan.

2.3.53 Use Case : Print Penjualan

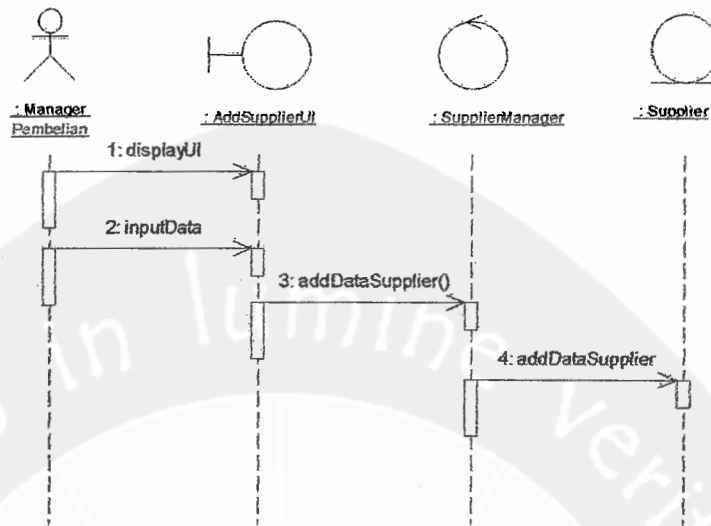


Gambar 2.101 Design Sequence Diagram Use Case Print Penjualan

Flow of events :

1. User menampilkan antarmuka untuk menampilkan data penjualan berdasarkan tanggal penjualan yaitu boundary class DisplayPenjualanUI.
2. DisplayPenjualanUI memanggil fungsi getPenjualan() milik control class PenjualanManager dimana di dalamnya dipanggil fungsi getPenjualan() milik entity class Penjualan.
3. DisplayPenjualanUI memanggil fungsi cetakLaporan() untuk mencetak laporan.

2.3.54 Use Case : Add Supplier

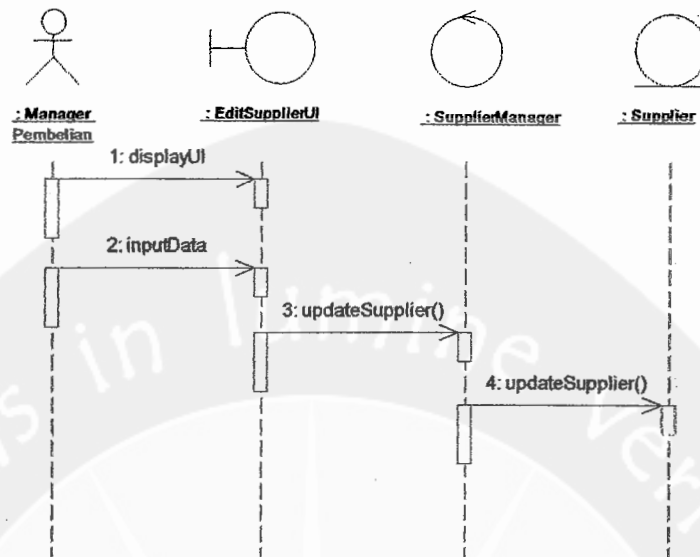


Gambar 2.102 Design Sequence Diagram Use Case Add Supplier

Flow of events :

1. User menampilkan antarmuka untuk menambah data supplier ke basis data yaitu boundary class AddSupplierUI.
2. User memasukkan data supplier baru.
3. AddSupplierUI memanggil fungsi addSupplier() milik control class SupplierManager.
4. SupplierManager memanggil fungsi addSupplier() milik entity class Supplier.

2.3.55 Use Case : Edit Supplier

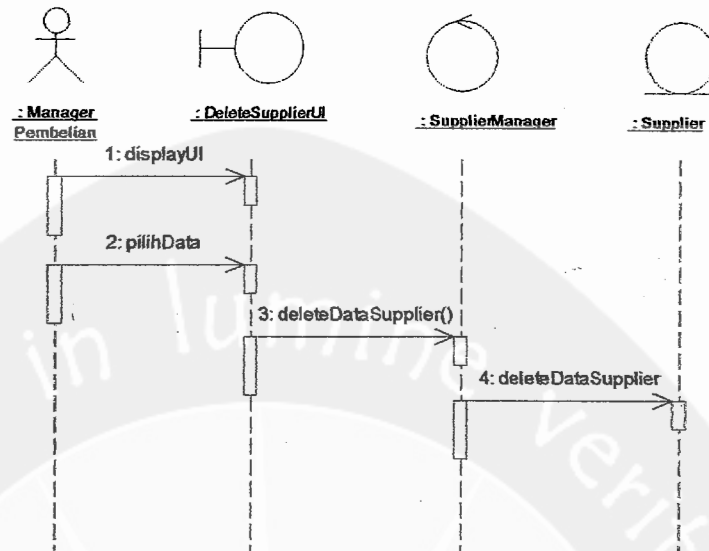


Gambar 2.103 Design Sequence Diagram Use Case Edit Supplier

Flow of events :

1. User menampilkan antarmuka untuk mengubah data supplier yaitu boundary class `EditSupplierUI` kemudian user memasukkan data pencarian supplier.
2. User memasukkan data supplier baru.
3. `EditSupplierUI` memanggil fungsi `updateSupplier()` milik control class `SupplierManager`.
4. `SupplierManager` memanggil fungsi `updateSupplier ()` milik entity class `Supplier`.

2.3.56 Use Case : Delete Supplier

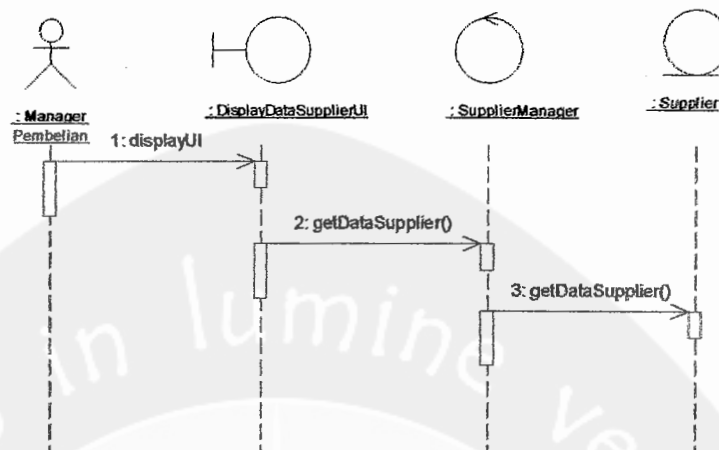


Gambar 2.104 Design Sequence Diagram Use Case Delete Supplier

Flow of events :

1. User menampilkan antarmuka untuk menghapus data supplier yaitu boundary class DeleteSupplierUI.
2. User memilih data supplier yang akan dihapus..
3. DeleteSupplierUI memanggil fungsi deleteDataSupplier () milik control class SupplierManager.
4. SupplierManager memanggil fungsi deleteDataSupplier () milik entity class Supplier.

2.3.57 Use Case : Display Supplier

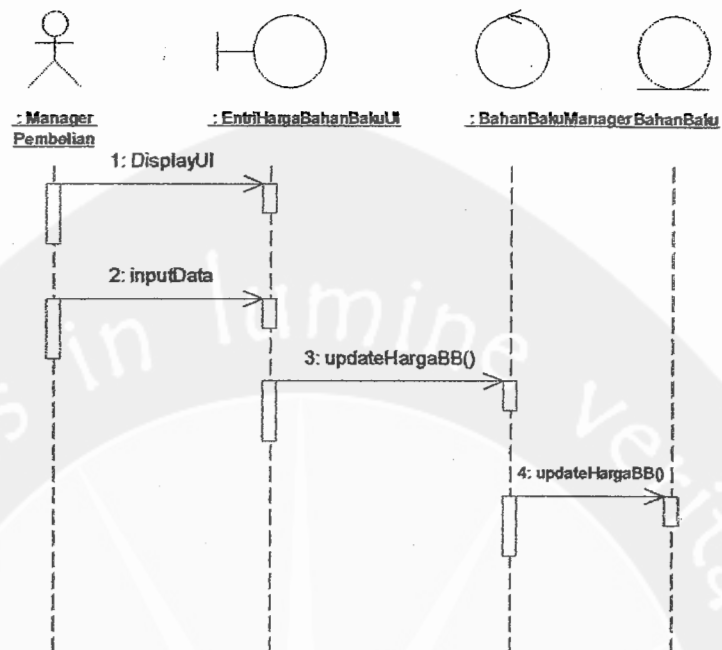


Gambar 2.105 Design Sequence Diagram Use Case Display Supplier

Flow of events :

1. User menampilkan antarmuka untuk menampilkan data supplier yaitu boundary class DisplaySupplierUI.
2. DisplaySupplierUI memanggil fungsi getDataSupplier() milik control class SupplierManager.
3. SupplierManager memanggil fungsi getDataSupplier() milik entity class Supplier.

2.3.58 Use Case : Edit Harga Bahan Baku

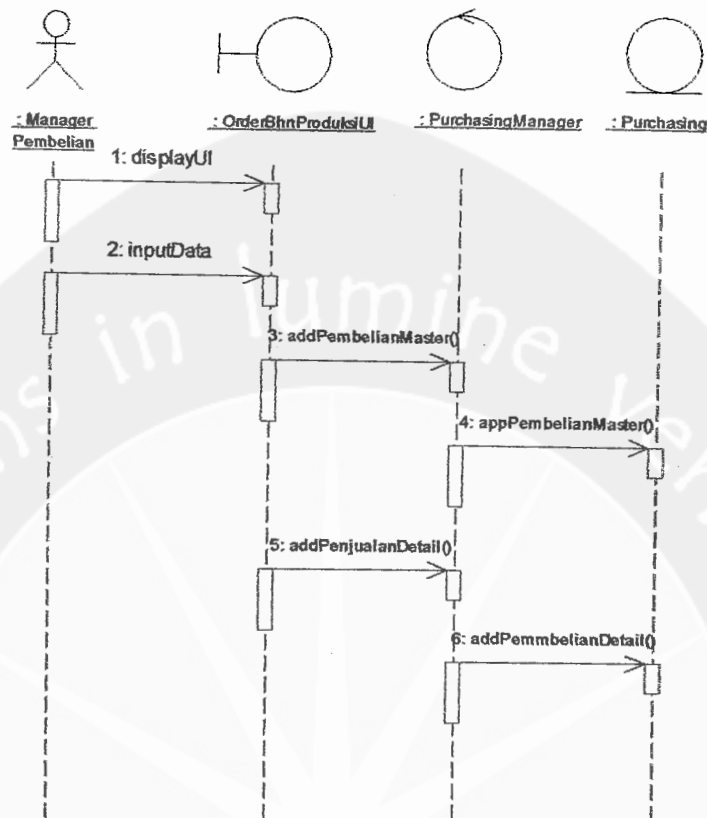


Gambar 2.106 Design Sequence Diagram Use Case Edit Harga Bahan Baku

Flow of events :

1. User menampilkan antarmuka untuk mengubah harga bahan baku yaitu boundary class `EditHargaBahanBakuUI`.
2. User memasukkan data harga bahan baku.
3. `EditHargaBahanBakuUI` memanggil fungsi `updateHargaBB()` milik control class `BahanBakuManager` dimana di dalamnya dipanggil fungsi `updateHargaBB()` milik entity class `BahanBaku`.

2.3.59 Use Case : Order Bahan Baku

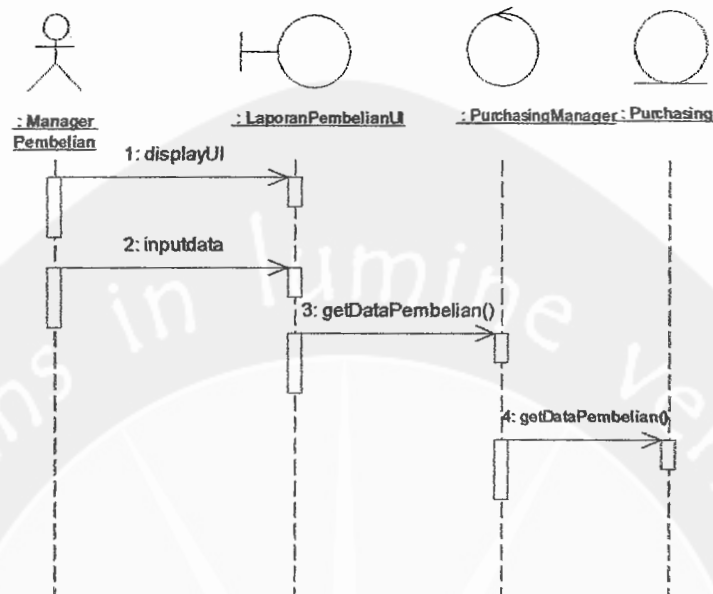


Gambar 2.107 Design Sequence Diagram Use Case Order Bahan Baku

Flow of events :

1. User menampilkan antarmuka untuk melakukan order bahan baku produksi yaitu boundary class `OrderBhnProduksiUI`.
2. User memasukkan data pembelian bahan baku.
3. `OrderBhnProduksiUI` memanggil fungsi `addPembelianMaster()` dan `addPembelianDetail()` milik control class `PurchasingManager` dimana di dalamnya dipanggil fungsi `addPembelianMaster()` dan `addPembelianDetail()` milik entity class `Purchasing`.

2.3.60 Use Case : Laporan Pembelian

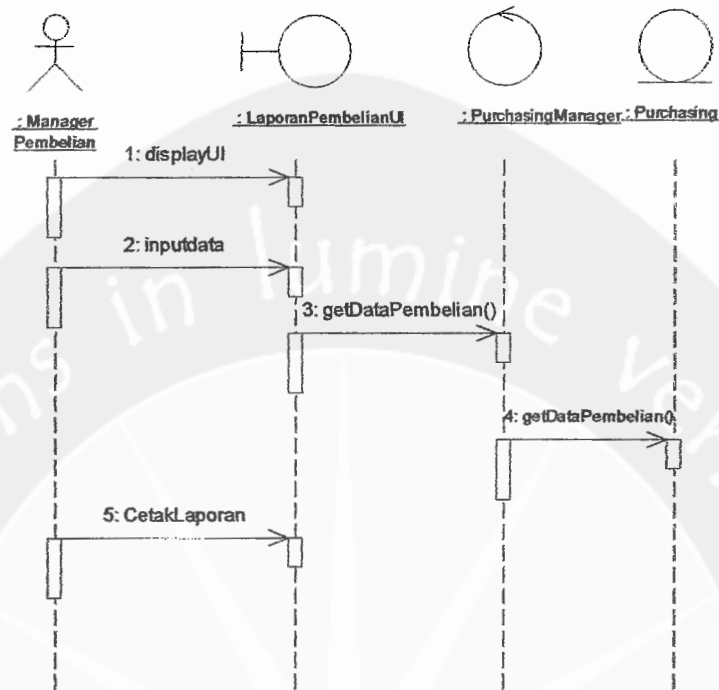


Gambar 2.108 Design Sequence Diagram Use Case Laporan Pembelian

Flow of events :

1. User menampilkan antarmuka untuk membuat laporan pembelian yaitu boundary class LaporanPembelianUI.
2. User memasukkan tanggal pembelian.
3. LaporanPembelianUI akan menampilkan data pembelian dengan cara memanggil fungsi `getDataPembelian()` milik control class PurchasingManager dimana di dalamnya dipanggil fungsi `getDataPembelian()` milik entity class Purchasing.

2.3.61 Use Case : Print Laporan Pembelian

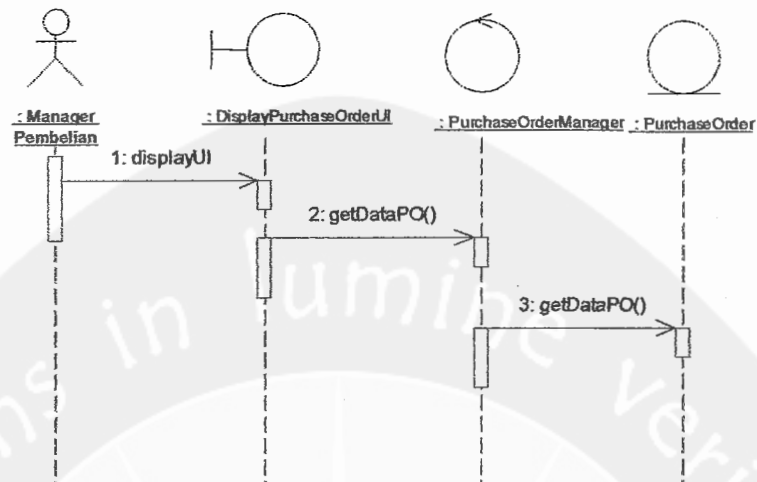


Gambar 2.109 Design Sequence Diagram Use Case Print Laporan Pembelian

Flow of events :

1. User menampilkan antarmuka untuk membuat laporan pembelian yaitu boundary class `LaporanPembelianUI`.
2. User memasukkan tanggal pembelian.
3. `LaporanPembelianUI` akan menampilkan data pembelian dengan cara memanggil fungsi `getDataPembelian()` milik control class `PurchasingManager` dimana di dalamnya dipanggil fungsi `getDataPembelian()` milik entity class `Purchasing`.
4. Untuk mencetak ke kertas dengan memanggil fungsi `cetakLaporan()`.

2.3.62 Use Case : Display Purchase Order

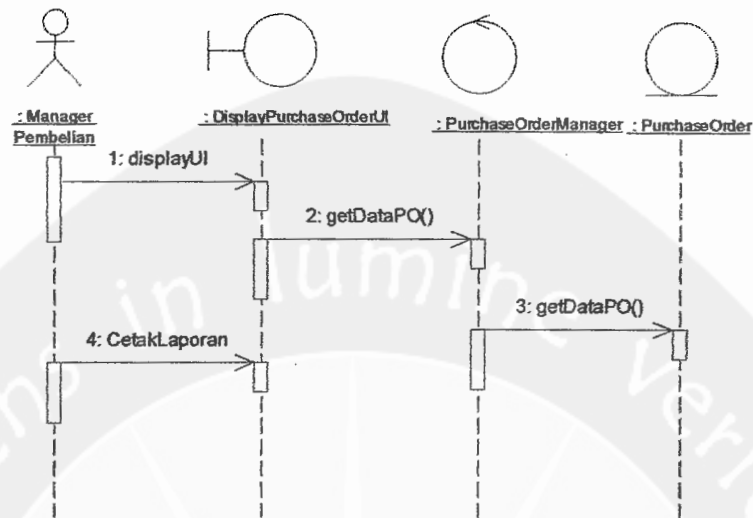


Gambar 2.110 Design Sequence Diagram Use Case Display Purchase Order

Flow of events :

1. User menampilkan antarmuka untuk menampilkan data purchase order yaitu boundary class `DisplayPurchaseOrderUI`.
2. `DisplayPurchaseOrderUI` akan menampilkan data purchase order dengan cara memanggil fungsi `getDataPO()` milik control class `PurchaseOrderManager` dimana di dalamnya dipanggil fungsi `getDataPO()` milik entity class `PurchaseOrder`.

2.3.63 Use Case : Print Purchase Order

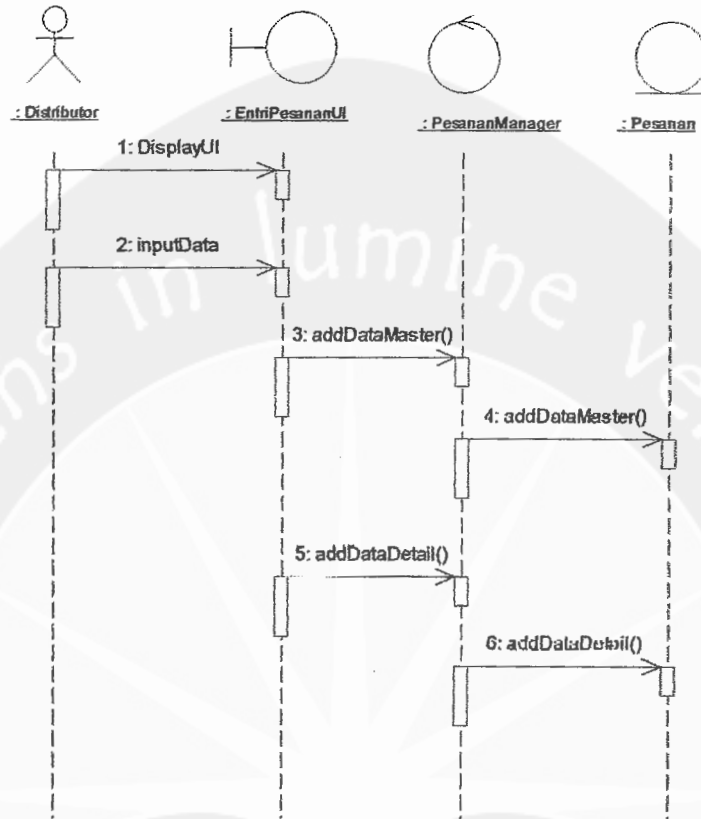


Gambar 2.111 Design Sequence Diagram Use Case Print Purchase Order

Flow of events :

1. User menampilkan antarmuka untuk menampilkan data purchase order yaitu boundary class `DisplayPurchaseOrderUI`.
2. `DisplayPurchaseOrderUI` akan menampilkan data purchase order dengan cara memanggil fungsi `getDataPO()` milik control class `PurchaseOrderManager` dimana di dalamnya dipanggil fungsi `getDataPO()` milik entity class `PurchaseOrder`.
3. `DisplayPurchaseOrderUI` akan memanggil fungsi `cetakLaporan()` untuk mencetak ke kertas.

2.3.64 Use Case : Entri Pesanan

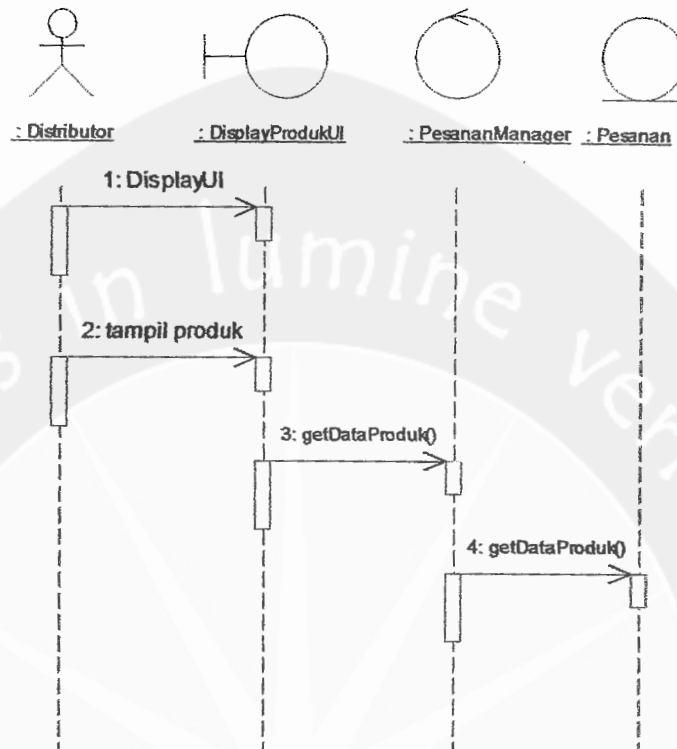


Gambar 2.112 Design Sequence Diagram Use Case Entri Pesanan

Flow of events :

1. User menampilkan antarmuka untuk membuat pesanan produk yaitu boundary class `EntriPesananUI`.
2. User melakukan input data pesanan.
3. `EntriPesananUI` akan memanggil fungsi `addDataMaster()` dan `addDataDetail()` milik control class `PesananManager` dimana di dalamnya dipanggil fungsi `addDataMaster()` dan `addDataDetail` milik entity class `Pesanan`.

2.3.65 Use Case : Display Produk



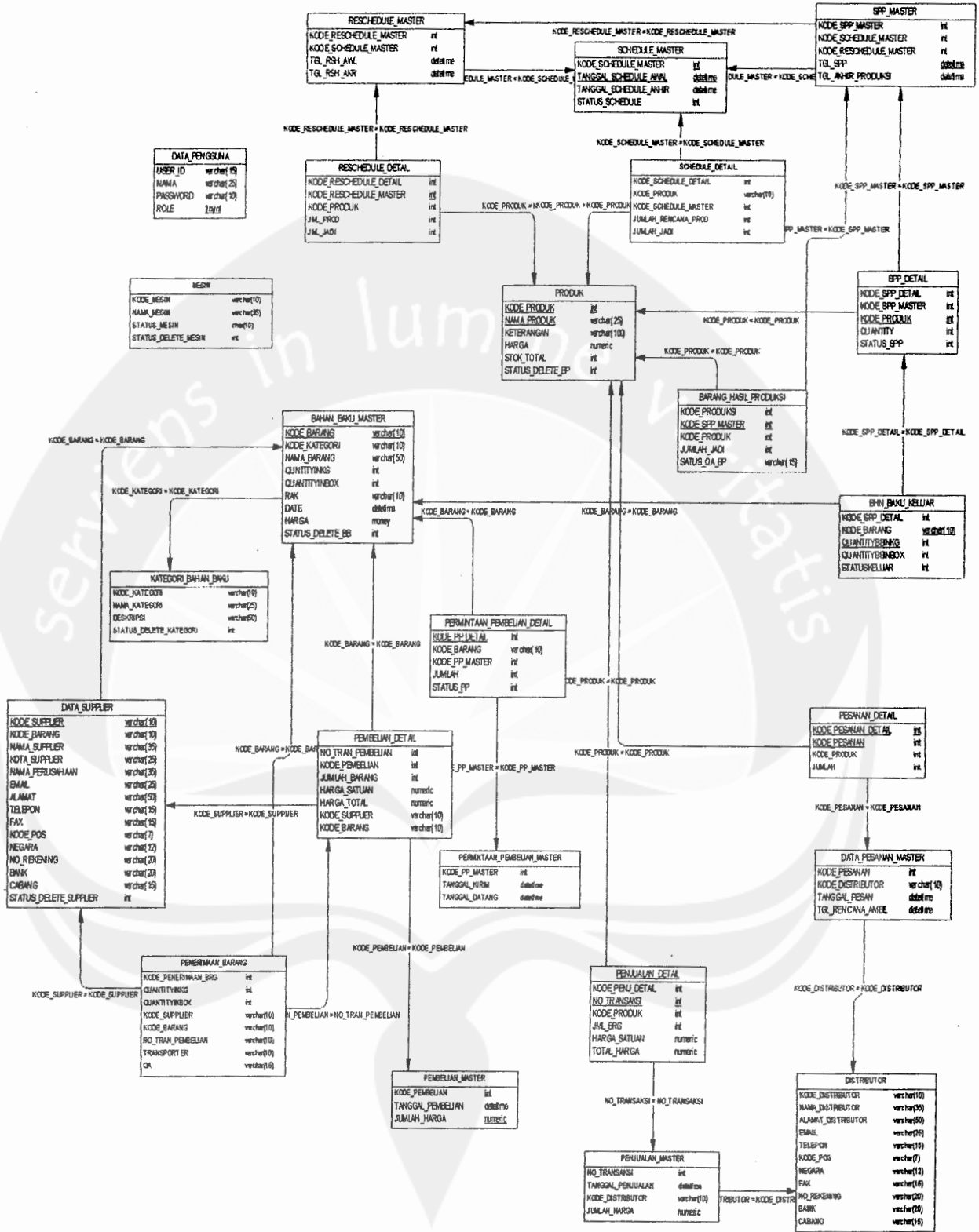
Gambar 2.113 Design Sequence Diagram Use Case Display Produk

Flow of events :

1. User menampilkan antarmuka untuk menampilkan data produk yaitu boundary class `DisplayProdukUI`.
2. `DisplayProdukUI` akan menampilkan data produk dengan cara memanggil fungsi `getDataProduk()` milik control class `PesananManager` dimana di dalamnya dipanggil fungsi `getDataProduk()` milik entity class `Pesanan`.

3 Deskripsi Perancangan Persistent Data

3.1 Basis Data



Gambar 3.1 Database Diagram ASCaMaDa

3.1.1 Data_Pengguna

Tabel ini merepresentasikan entitas Data Pengguna

Tabel 4.1 Tabel Data Pengguna

Field	Tipe Data	NULL	Deskripsi
User_ID	varchar(15)	No	Id pengguna sistem. Primary Key
Nama	varchar(25)	No	Nama user
Password	varchar(10)	No	Password
Role	tinyint	No	Role

3.1.2 Schedule_Master

Tabel ini merepresentasikan entitas schedule master.

Tabel 4.2 Tabel Data Schedule Master

Field	Tipe Data	NULL	Deskripsi
Kode_Schedule_Master	Integer	No	Kode schedule master. Primary Key
Tanggal_Schedule_Awal	datetime	No	Tanggal schedule awal
Tanggal_Schedule_Akhir	Datetime	No	Tanggal schedule akhir
Status_Schedule	int	No	Status schedule jika terjadi reschedule

3.1.3 Schedule_Detail

Tabel ini merepresentasikan entitas schedule detail.

Tabel 4.3 Tabel Data Schedule Detail

Field	Tipe Data	NULL	Deskripsi
Kode_Schedule_Detail	Integer	No	Kode schedule master. Primary Key
Kode_Schedule_Master	Integer	No	Kode schedule master. Foreign

			Key
Kode_Produk	Integer	No	Kode Produk
Jumlah_Rencana_Produksi	Integer	No	Jumlah rencana produksi
Jumlah_Jadi	Integer	No	Jumlah hasil produksi

3.1.4 Reschedule_Master

Tabel ini merepresentasikan entitas `reschedule master`.

Tabel 4.4 Tabel Data Reschedule Master

Field	Tipe Data	NULL	Deskripsi
Kode_Reschedule_Master	Integer	No	Kode reschedule master. Primary Key
Kode_Schedule_Master	Integer	No	Kode schedule master. Foreign Key
Tgl_Rsh_Awl	varchar(10)	No	Tanggal reschedule awal
Tgl_Rsh_Akr	Integer	No	Tanggal reschedule akhir

3.1.5 Rechedule_Detail

Tabel ini merepresentasikan entitas `reschedule detail`.

Tabel 4.5 Tabel Data Reschedule Detail

Field	Tipe Data	NULL	Deskripsi
Kode_Reschedule_Detail	Integer	No	Kode reschedule master. Primary Key
Kode_Reschedule_Master	Integer	No	Kode reschedule master. Foreign

			Key
Kode_Produk	Integer	No	Kode Produk
Jml_Prod	Integer	No	Jumlah rencana produksi
Jml_Jadi	Integer	No	Jumlah hasil produksi

3.1.6 SPP_Master

Tabel ini merepresentasikan entitas SPP Master.

Tabel 4.6 Tabel Data SPP Master

Field	Tipe Data	NULL	Deskripsi
Kode_SPP_Master	Integer	No	Kode surat perintah produksi master. Primary Key
Kode_Schedule_Master	Integer	No	Kode schedule master. Foreign Key
Kode_Rechedule_Master	Integer	No	Kode reschedule master. Foreign Key
Tgl_SPP	Datetime	No	Tanggal SPP dibuat
Tgl_Akhir_Produksi	Datetime	No	Tanggal akhir produksi

3.1.7 SPP_Detail

Tabel ini merepresentasikan entitas SPP Detail.

Tabel 4.7 Tabel Data SPP Detail

Field	Tipe Data	NULL	Deskripsi
Kode_SPP_Detail	Integer	No	Kode surat perintah produksi detail. Primary Key
Kode_SPP_Master	Integer	No	Kode spp master. Foreign Key
Kode_Produk	Integer	No	Kode produk
Quantity	Integer	No	Jumlah produk

Status_SPP	Integer	No	Status selesai produksi atau belum
------------	---------	----	------------------------------------

3.1.8 Bhn_Baku_Keluar

Tabel ini merepresentasikan entitas Bahan Baku Keluar.

Tabel 4.8 Tabel Data Bahan Baku Keluar

Field	Tipe Data	NULL	Deskripsi
Kode_SPP_Detail	Integer	No	Kode surat perintah produksi detail. Foreign Key
Kode_Barang	Varchar(10)	No	Kode bahan baku. Foreign Key
QuantityBBInKg	Integer	No	Jumlah produk dalam kilogram
QuantityBBInBox	Integer	No	Jumlah produk dalam box
Statuskeluar	Integer	No	Status bahan baku keluar dari gudang

3.1.9 Produk

Tabel ini merepresentasikan entitas Produk.

Tabel 4.9 Tabel Data Produk

Field	Tipe Data	NULL	Deskripsi
Kode_Produk	Integer	No	Kode produk. Primary Key
Nama_Produk	Varchar(25)	No	Nama produk
Keterangan	Varchar(100)	No	Keterangan mengenai produk
Harga	Integer	No	Harga produk
Stok_Total	Integer	No	Jumlah stok produk
Status_Delete_BP	Integer		Status penghapusan

3.1.10 Barang_Hasil_Produksi

Tabel ini merepresentasikan entitas Barang Hasil Produksi.

Tabel 4.10 Tabel Barang Hasil Produksi

Field	Tipe Data	NULL	Deskripsi
Kode_Produksi	Integer	No	Kode produksi. Primary Key
Kode_SPP_Master	Integer	No	Kode surat perintah produksi master. Foreign Key
Kode_Produk	Integer	No	Kode Produk
Jumlah_Jadi	Integer	No	Jumlah hasil produksi
Status_QA_BP	Varchar(15)	No	Status pengecekan dari quality control

3.1.11 Data_Pesanan_Master

Tabel ini merepresentasikan entitas Data Pesanan Master.

Tabel 4.11 Tabel Data Pesanan Master

Field	Tipe Data	NULL	Deskripsi
Kode_Pesanan	Integer	No	Kode pesanan Primary Key
Kode_Distributor	Varchar(10)	No	Kode distributor Foreign Key
Tanggal_Pesan	Integer	No	Tanggal pesanan
Tgl_Rencana_Ambil	Integer	No	Tanggal rencana ambil

3.1.12 Pesanan_Detail

Tabel ini merepresentasikan entitas Pesanan Detail.

Tabel 4.12 Tabel Pesanan Detail

Field	Tipe Data	NULL	Deskripsi
Kode_Pesanan_Detail	Integer	No	Kode pesanan detail. Primary Key
Kode_Pesanan	Integer	No	Kode pesanan. Foreign Key
Kode_Produk	Integer	No	Kode produk. Foreign Key
Jumlah	Integer	No	Jumlah produk pesanan

3.1.13 Distributor

Tabel ini merepresentasikan entitas Distributor.

Tabel 4.13 Tabel Distributor

Field	Tipe Data	NULL	Deskripsi
Kode_Distributor	Varchar(10)	No	Kode distributor Primary Key
Nama_Distributor	Varchar(10)	No	Nama distributor
Alamat_Distributor	Varchar(10)	No	Alamat distributor
Email	Varchar(10)	No	Alamat email
Telepon	Varchar(10)	No	Nomor telepon
Kode_Pos	Varchar(10)	No	Kode Pos
Negara	Varchar(10)	No	Negara
Fax	Varchar(10)	No	Fax
No_Rekening	Varchar(10)	No	No Rekening
Bank	Varchar(10)	No	Bank
Cabang	Varchar(10)	No	Cabang

3.1.14 Penjualan_Master

Tabel ini merepresentasikan entitas Penjualan Master.

Tabel 4.14 Tabel Penjualan Master

Field	Type Data	NULL	Deskripsi
No_Transaksi	Integer	No	Nomor Transaksi Primary Key
Tanggal_Penjualan	Datetime	No	Tanggal penjualan
Kode_Distributor	Varchar(10)	No	Kode distributor Foreign Key
Jumlah_Harga	Numeric	No	Jumlah harga bayar

3.1.15 Penjualan_Detail

Tabel ini merepresentasikan entitas Penjualan Detail.

Tabel 4.15 Tabel Penjualan Detail

Field	Type Data	NULL	Deskripsi
Kode_Penj_Detail	Integer	No	Kode penjualan detail. Primary Key
No_Transaksi	Integer	No	No transaksi penjualan. Foreign Key
Kode_Produk	Integer	No	Kode produk Foreign Key
Jml_Brg	Integer	No	Jumlah barang
Harga_Satuan	Numeric	No	Harga satuan barang
Total_Harga	Numeric	No	Total harga

3.1.16 Bahan_Baku_Master

Tabel ini merepresentasikan entitas Bahan baku master.

Tabel 4.16 Tabel Bahan Baku Master

Field	Tipe Data	NULL	Deskripsi
Kode_Barang	Varchar(10)	No	Kode bahan baku Primary Key
Kode_Kategori	Varchar(10)	No	Kode kategori bahan baku. Foreign Key
Nama_Barang	Varchar(10)	No	Nama bahan baku
Quantityinkg	Varchar(10)	No	Jumlah dalam Kilogram
Quantityinbox	Varchar(10)	No	Jumlah dalam box
Rak	Varchar(10)	No	Rak tempat bahan baku disimpan
Date	Varchar(10)	No	Tanggal bahan baku ditambahkan
Harga	Varchar(10)	No	Harga bahan baku
Status_Delete_BB	Varchar(10)	No	Status hapus bahan baku

3.1.17 Kategori_Bahan_Baku

Tabel ini merepresentasikan entitas Kategori Bahan baku.

Tabel 4.17 Tabel Kategori Bahan Baku

Field	Tipe Data	NULL	Deskripsi
Kode_Kategori	Varchar(10)	No	Kode kategori Primary Key
Nama_Kategori	Varchar(10)	No	Nama kategori
Deskripsi	Varchar(10)	No	Deskripsi kategori
Status_Delete_Kategori	Integer	No	Status hapus kategori

3.1.18 Permintaan_Pembelian_Master

Tabel ini merepresentasikan entitas Permintaan Pembelian Master.

Tabel 4.18 Tabel Permintaan Pembelian Master

Field	Tipe Data	NULL	Deskripsi
Kode_PP_Master	Integer	No	Kode permintaan pembelian bahan baku master. Primary Key
Tanggal_Kirim	Datetime	No	Tanggal pengiriman bahan baku
Tanggal_Datang	Datetime	No	Tanggal datang bahan baku

3.1.19 Permintaan_Pembelian_Detail

Tabel ini merepresentasikan entitas Permintaan Pembelian Detail.

Tabel 4.19 Tabel Permintaan Pembelian Detail

Field	Tipe Data	NULL	Deskripsi
Kode_PP_Detail	Integer	No	Kode permintaan pembelian bahan baku detail. Primary Key
Kode_PP_Master	Integer	No	Kode permintaan pembelian bahan baku master. Foreign Key
Kode_Barang	Varchar(10)	No	Kode bahan baku
Jumlah	Integer	No	Jumlah barang
Status_PP	Integer	No	Status permintaan pembelian

3.1.20 Pembelian_Master

Tabel ini merepresentasikan entitas Pembelian Master.

Tabel 4.20 Tabel Pembelian Master

Field	Type Data	NULL	Deskripsi
Kode_Pembelian	Integer	No	Kode permintaan pembelian bahan baku master. Primary Key
Tanggal_Pembelian	Datetime	No	Tanggal pengiriman bahan baku
Jumlah_Harga	Datetime	No	Tanggal datang bahan baku

3.1.21 Pembelian_Detail

Tabel ini merepresentasikan entitas Pembelian Detail.

Tabel 4.21 Tabel Pembelian Detail

Field	Type Data	NULL	Deskripsi
No_Trans_Pembelian	Integer	No	No transaks pembelian. Primary Key
Kode_Pembelian	Integer	No	Kode pembelian bahan baku master. Foreign Key
Kode_Barang	Varchar(10)	No	Kode bahan baku. Foreign Key
Kode_Supplier	Varchar(10)	No	Kode supplier. Foreign Key
Jumlah_Barang	Integer	No	Jumlah bahan baku
Harga_Satuan	Numeric	No	Harga satuan
Harga_Total	Numeric	No	Harga Total

3.1.22 Penerimaan_Barang

Tabel ini merepresentasikan entitas Penerimaan Barang.

Tabel 4.22 Tabel Penerimaan Barang

Field	Tipe Data	NULL	Deskripsi
Kode_Penerimaan_Brg	Integer	No	Kode penerimaan barang. Primary Key
No_Tran_Pembelian	Integer		No transaksi pembelian. Foreign Key
Kode_Barang	Integer	No	Kode bahan baku master. Foreign Key
Kode_Supplier	Varchar(10)	No	Kode supplier. Foreign Key
Quantityinkg	Integer	No	Jumlah barang dalam Kilogram
Quantityinbox	Integer	No	Jumlah barang dalam Box
Transporter	Varchar(10)	No	No Transporter
QA	Varchar(15)	No	Status cek dari Quality Control

3.1.23 Data_Supplier

Tabel ini merepresentasikan entitas Data Supplier.

Tabel 4.23 Tabel Data Supplier

Field	Tipe Data	NULL	Deskripsi
Kode_Supplier	Varchar(10)	No	Kode Supplier. Primary Key
Nama_Supplier	Varchar(35)	No	Nama Supplier
Kota_Supplier	Varchar(25)	No	Kota Supplier
Nama_Perusahaan	Varchar(35)	No	Nama Perusahaan
Email	Varchar(25)	No	Alamat Email

Alamat	Varchar(50)	No	Alamat
Telepon	Varchar(15)	No	Nomor Telepon
Fax	Varchar(15)	No	Fax
Kode_Pos	Varchar(7)	No	Kode Pos
Negara	Varchar(12)	No	Negara
No_Rekening	Varchar(20)	No	No Rekening
Bank	Varchar(20)	No	Nama Bank
Cabang	Varchar(15)	No	Cabang
Status_Delete_Supplier	Integer	No	Status penghapusan data supplier

3.1.24 Mesin

Tabel ini merepresentasikan entitas Mesin.

Tabel 4.24 Tabel Mesin

Field	Type Data	NULL	Deskripsi
Kode_Mesin	Integer	No	Kode mesin Primary Key
Nama_Mesin	Integer		Nama mesin produksi.
Status_Mesin	Integer	No	Status mesin sedang dipakai atau tidak.
Status_Delete_Mesin	Varchar(10)	No	Status penghapusan data mesin

4 Deskripsi Perancangan Antarmuka

4.1 Use Case : Login



Gambar 4.1 Rancangan Halaman Login

Deskripsi

- Halaman ini digunakan untuk melakukan login bagi user. Hak akses terhadap operasi yang dapat dilakukan tergantung dari login user.
- Terdapat satu tombol Login yang digunakan untuk masuk ke dalam sistem.

Event

- Login

```
onClick_Login()  
{  
    SQL Statement :  
    SELECT * FROM Data_Pengguna  
    WHERE NamaUser = '[txtNama]' and Password  
    = '[txtPassword]'  
}
```

Urutan aksi yang terjadi :

Program Studi Teknik Informatika	DPPL-ASCaMaDa	194/ 277
----------------------------------	---------------	----------

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika

1. User memasukkan username dan password pada textbox yang tersedia.
2. User menekan tombol Login.
3. Sistem memeriksa di basis data, apakah data user dengan nama yang sama dengan nama user dari langkah 1.
4. Sistem membandingkan nama user dan password dari query yang diperoleh pada langkah 3 dengan password input user. Jika password sesuai, user masuk ke sistem sesuai rolenya. Jika password tidak sesuai, sistem menampilkan pesan kesalahan.

4.2 Use Case : Ubah Password

Gambar 4.2 Rancangan Halaman Ubah Password

Deskripsi

- Halaman ini digunakan untuk melakukan perubahan password user.

- Tombol Change digunakan untuk menyimpan password yang baru.

Event

- Ubah Password

```
OnClick_Chgange()  
{  
    SQL Statement :  
    UPDATE Data_Pengguna  
    SET Password = [txtPasswordBaru]  
    WHERE IdUser = 'Session [IdUser]'  
}
```

Urutan aksi yang terjadi :

1. User memasukkan password yang lama.
2. User memasukkan password baru dan konfirmasinya
3. User menekan tombol Change.
4. Sistem akan melakukan perubahan password lama dengan password baru namun terlebih dahulu akan dicek password yang lama.
5. Jika perubahan password berhasil maka sistem akan menampilkan pesan, dan jika gagal sistem akan menampilkan pesan kesalahan.

4.3 Use Case : Add User

The screenshot shows a web application interface for adding a user. At the top right, the logo for 'SARI HUSADA' is visible with the tagline 'The Pioneer & Manufacturer of Palm & Nut Based Foods'. The main form area contains the following elements:

- User ID:** A text input field.
- Nama:** A text input field.
- Role:** A dropdown menu with a list of roles: 1. Administrator, 2. PPIC, 3. Produksi, 4. Warehouse, 5. QA, 6. Pembelian, 7. Pengujian, 8. Supplier, 9. Distributor.
- Password:** A text input field.
- Confirm Password:** A text input field.
- Buttons:** 'Add' and 'Cancel' buttons.

Gambar 4.3 Rancangan Add User

Deskripsi

- Halaman ini digunakan untuk melakukan penambahan user sebagai pengguna sistem.
- Tombol Add digunakan untuk menyimpan data user baru.

Event

- Tambah User

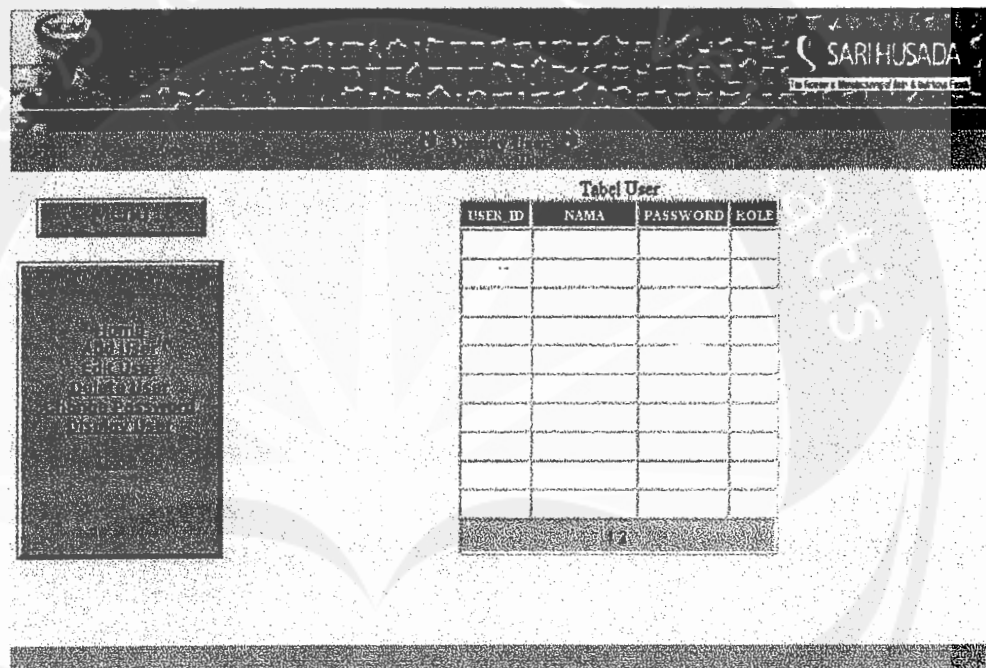
```
OnClick_Add()  
{  
    SQL Statement :  
    INSERT INTO Data_Pengguna  
    VALUES (id, [txtNama], [txtPassword], [ddlRole])  
}
```

Urutan aksi yang terjadi :

1. User memasukkan user id, nama, password, dan role sebagai data user yang baru.
2. User menekan tombol Add.
3. Sistem akan melakukan pengecekan terhadap user id yang dimasukkan.

4. Apabila user id dan inputan lainnya valid maka sistem akan menambahkan data user yang baru ke dalam basis data.
5. Jika penambahan data user berhasil maka sistem akan menampilkan pesan, dan jika gagal sistem akan menampilkan pesan kesalahan.

4.4 Use Case : Display User



Gambar 4.4 Rancangan Halaman Display User

Deskripsi

- Halaman ini digunakan untuk menampilkan data user di basis data.

Event

- Halaman di *load*

```

Page_Load()
{
    SQL Statement :
    SELECT * FROM Data_Pengguna
}

```

Urutan aksi yang terjadi :

1. Sistem akan menampilkan data user yang terdapat di basisdata.

4.5 Use Case : Edit User

The screenshot shows a web interface for editing a user. At the top right, there is a logo for 'SARIHUSADA' with the tagline 'The Power of Manufacturing Goes to the Next Level'. The main form area contains the following elements:

- A search input field labeled 'User ID' with a 'Cari' button next to it.
- A text input field labeled 'Nama'.
- A dropdown menu labeled 'Role'.
- A password input field labeled 'Password'.
- 'Update' and 'Cancel' buttons at the bottom of the form.
- A list of roles on the right side: 1. Administrator, 2. PPIC, 3. Prodikel, 4. Warehouse, 5. QA, 6. Pembelian, 7. Penjualan, 8. Supplier, 9. Distributor.

Gambar 4.5 Rancangan Halaman Editing User

Deskripsi

- Halaman ini digunakan untuk melakukan proses *editing* terhadap data user di basis data.
- Tombol Cari digunakan untuk melakukan pencarian data user berdasarkan kata kunci.
- Tombol Update digunakan untuk melakukan perubahan data user.
- Tombol Cancel digunakan untuk menghapus inputan data.

Event

- Cari User

```
OnClick_Search()  
{  
    SQL Statement :  
    SELECT * FROM Data_Pengguna
```



```
WHERE [idUser] = [txtid]
}
```

Urutan aksi yang terjadi :

1. User memasukkan kata kunci pencarian.
2. Sistem akan mencari data berdasarkan kata kunci yang diinputkan kemudian menampilkannya di tempat yang tersedia.

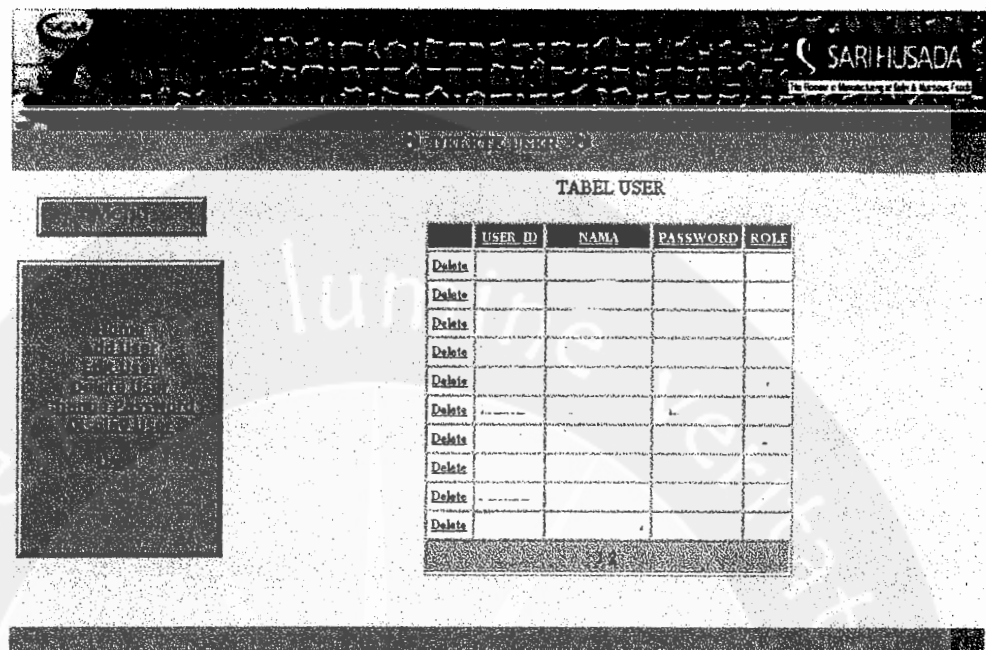
- **Ubah User**

```
OnClick_Update()
{
    SQL Statement :
    UPDATE Data_Pengguna
    SET NamaUser = [txtNamaUser],
        Password = [txtPassword],
        Role = [ddlRole]
    WHERE IdUser = [txtId]
}
```

Urutan aksi yang terjadi :

1. User menekan tombol Update, kemudian melakukan perubahan terhadap data tersebut.
2. Sistem akan melakukan perubahan data user yang dipilih.
3. Jika pengubahan data user berhasil maka sistem akan menampilkan pesan, dan jika gagal sistem akan menampilkan pesan kesalahan.

4.6 Use Case : Delete User



Gambar 4.6 Rancangan Halaman Delete User

Deskripsi

- Halaman ini digunakan untuk menghapus data user di basis data.

Event

- Delete User

```
OnClick_Delete()  
{  
    SQL Statement :  
    Delete From Data_Pengguna  
    WHERE [idUser] = dg.SelectedIndex()  
}
```

Urutan aksi yang terjadi :

1. User memilih data yang akan dihapus.
2. Sistem akan menghapus data sesuai dengan pilihan user.

4.7 Use Case : Entri Schedule Produksi

The screenshot shows a web application interface for entering production schedules. At the top, there is a header with the logo 'SARIHUSADA' and the tagline 'The Essence of Indonesian Food'. Below the header, there is a sidebar menu on the left. The main content area contains a form with the following fields and buttons:

- ID Schedule**: A text input field.
- Tgl Produksi Awal**: A date input field.
- Tgl Produksi Akhir**: A date input field.
- Add**: A button to submit the form.
- Tgl Produksi Awal**: A calendar widget for July 2007.
- Tgl Produksi Akhir**: A calendar widget for July 2007.
- ID Schedule Detail**: A text input field.
- Kode Produk**: A dropdown menu.
- Jumlah Rencana Produksi**: A text input field.

Gambar 4.7 Rancangan Halaman Entri Schedule Produksi

Deskripsi

- Halaman ini digunakan untuk membuat schedule produksi
- Tombol Add digunakan untuk menambahkan data schedule produksi ke basis data.

Event

- Entri Schedule Master

```
OnClick_Add1()
{
    SQL Statement :
    INSERT INTO Schedule_Master
    VALUES ([txtid], [txttglAwal], [txttglAkhir])
}
```

Urutan aksi yang terjadi :

1. User memasukkan detail data schedule master.
2. User menekan tombol Add.
3. Sistem akan menambahkan data schedule master ke dalam basis data.

4. Jika penambahan data schedule master berhasil maka sistem akan menampilkan pesan dan user akan memasukkan data schedule detail, dan jika gagal sistem akan menampilkan pesan kesalahan.

- Entri Schedule Detail

```
OnClick_Add2()  
{  
    SQL Statement :  
    INSERT INTO Schedule_Detail  
    VALUES ([txtidDet],[ddlProduk],[txtid],[txtjml],0)  
}
```

Urutan aksi yang terjadi :

1. User memasukkan detail data schedule detail.
2. User menekan tombol Add.
3. Sistem akan menambahkan data schedule detail ke dalam basis data.
4. Jika penambahan data schedule detail berhasil maka sistem akan menampilkan pesan dan jika gagal sistem akan menampilkan pesan kesalahan.

4.8 Use Case : Entri Reschedule Produksi

The screenshot shows a web application interface for entering production reschedule data. The form is titled 'Entri Reschedule Produksi' and is part of the 'SARIHUSADA' system. It contains several input fields and a list of existing records on the left. The main form fields are: 'ID Reschedule' (text input), 'ID Schedule Master' (dropdown menu), 'Tgl Produksi Awal' (calendar picker), and 'Tgl Produksi Akhir' (calendar picker). Below these are 'ID Reschedule Detail', 'Kode Produk', and 'Jumlah Rencana Produksi' (text input). An 'Add' button is positioned to the right of the date pickers. The page header includes the 'SARIHUSADA' logo and the tagline 'The Future is Coming from Java & Software Tools'.

Gambar 4.8 Rancangan Halaman Entri Reschedule Produksi

Deskripsi

- Halaman ini digunakan untuk membuat reschedule produksi
- Tombol Add digunakan untuk menambahkan data reschedule produksi ke basis data.

Event

- Entri Reschedule Master

```
OnClick_Add1 ()  
{  
    SQL Statement :  
    INSERT INTO Rechedule Master  
        VALUES ([txtid], [ddlidSche] [txttglAwal], [txttglAkhir])  
}
```

Urutan aksi yang terjadi :

1. User memasukkan detail data reschedule master.
2. User menekan tombol Add.
3. Sistem akan menambahkan data reschedule master ke dalam basis data.

4. Jika penambahan data reschedule master berhasil maka sistem akan menampilkan pesan dan user akan memasukkan data reschedule detail, dan jika gagal sistem akan menampilkan pesan kesalahan.

- Entri Reschedule Detail

```
OnClick_Add2()  
{  
    SQL Statement :  
    INSERT INTO Schedule_Detail  
    VALUES ([txtidDet],[ddlProduk],[txtid],[txtjml],0  
    )  
}
```

Urutan aksi yang terjadi :

1. User memasukkan detail data reschedule detail.
2. User menekan tombol Add
3. Sistem akan menambahkan data reschedule detail ke dalam basis data.
4. Jika penambahan data reschedule detail berhasil maka sistem akan menampilkan pesan dan jika gagal sistem akan menampilkan pesan kesalahan

4.9 Use Case : Display Pesanan

Friday, July 06, 2007 Dari: TIGARASA

NO	TGL PESAN	TGL AMBIL	ID	KODE PRODUK	NAMA PRODUK	JUMLAH
1	3/3/2007	6/6/2007	1	101	SGM2	10
3	6/19/2007	6/21/2007	2	101	SGM2	10
			3	100	SGM1	100
4	6/21/2007	6/30/2007	4	101	SGM2	300
			5	100	SGM1	300
6	7/1/2007	6/30/2007	6	101	SGM2	1000

Gambar 4.9 Rancangan Halaman Display Pesanan

Deskripsi

- Halaman ini digunakan untuk menampilkan dan mencetak data pesanan distributor.
- Tombol Export digunakan untuk mengubah report ke dalam bentuk excel atau pdf.
- Fungsi Print bisa digunakan jika report telah di export ke dalam bentuk excel atau pdf.

Event

- Halaman di *load*

Page_Load()

{

SQL Statement :

SELECT * FROM Data_Pesanan_Master INNER JOIN

Data_Pesanan_Detail ON

Data_Pesanan_Master.Kode_Pesanan =

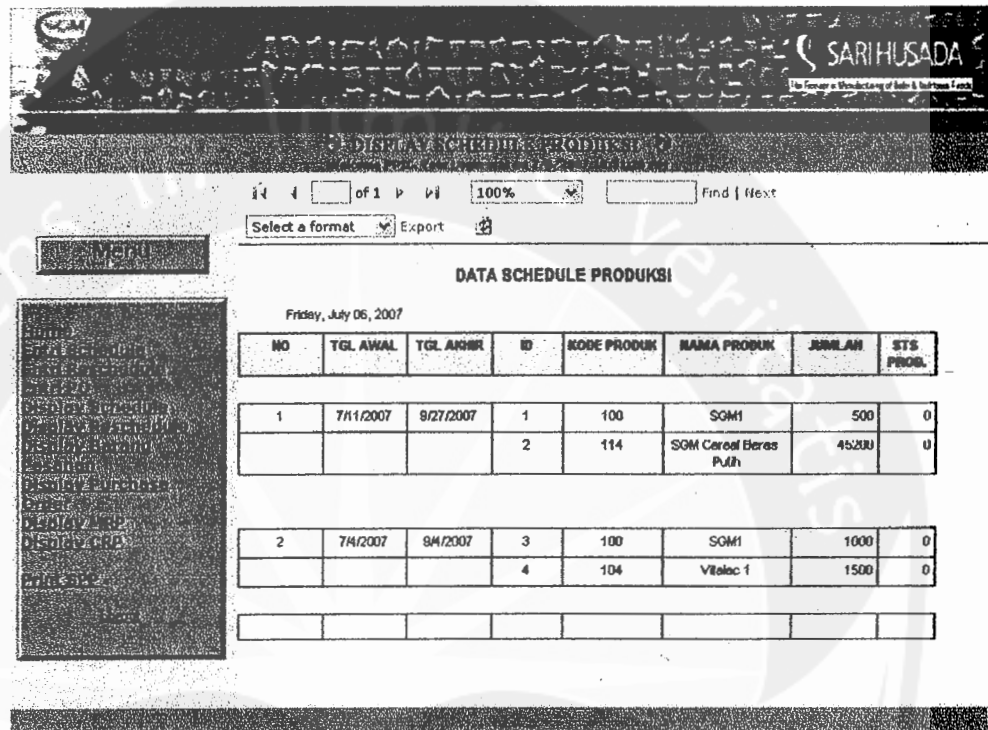
Data_Pesanan_Detail.Kode_Pesanan

}

Urutan aksi yang terjadi :

1. Sistem akan menampilkan data pesanan distributor dalam bentuk report .

4.10 Use Case : Display Schedule Produksi



Gambar 4.10 Rancangan Halaman Display ScheduleProduksi

Deskripsi

- Halaman ini digunakan untuk menampilkan dan mencetak data schedule produksi.
- Tombol Export digunakan untuk mengubah report ke dalam bentuk excel atau pdf.
- Fungsi Print bisa digunakan jika report telah di export ke dalam bentuk excel atau pdf.

Event

- Halaman di *load*

```
Page_Load()
```

```
{
```

```
SQL Statement :
```

Program Studi Teknik Informatika	DPPL-ASCaMaDa	207/ 277
----------------------------------	---------------	----------

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika


```

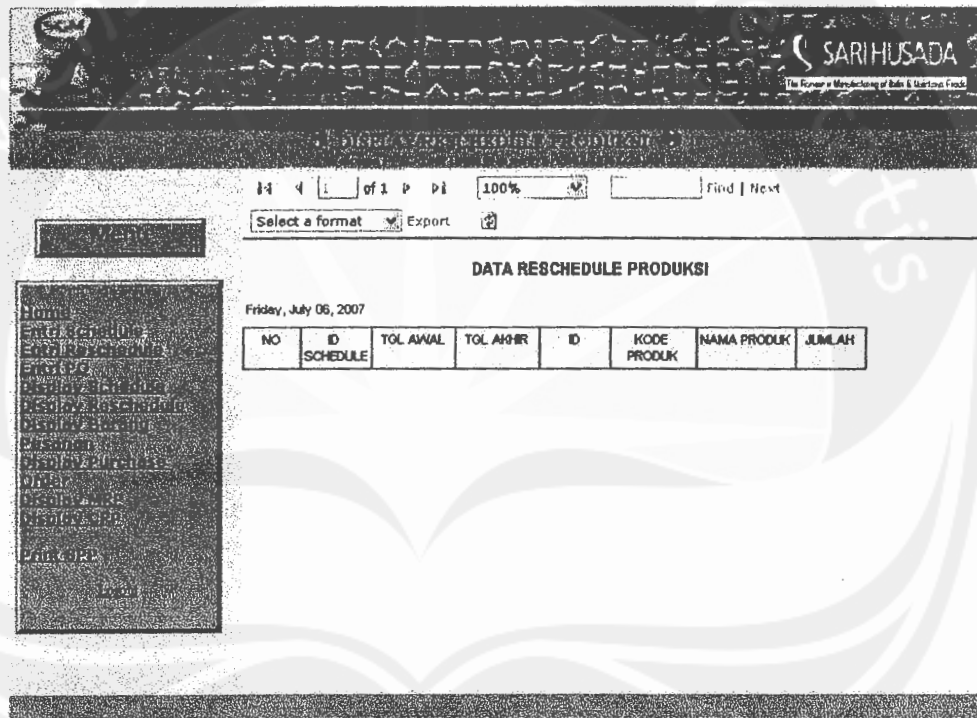
SELECT * FROM Data_Schedule_Master INNER JOIN
Data_Schedule_Detail ON Data_Schedule
_Master.Kode_Schedule_Master = Data_Schedule
_Detail.Kode_Schedule_Master WHERE
Data_Schedule_Master.Status_Schedule = 0
)

```

Urutan aksi yang terjadi :

1. Sistem akan menampilkan data schedule dalam bentuk report .

4.11 Use Case : Display Reschedule Produksi



Gambar 4.11 Rancangan Halaman Display Reschedule Produksi

Deskripsi

- Halaman ini digunakan untuk menampilkan dan mencetak data reschedule produksi.
- Tombol Export digunakan untuk mengubah report ke dalam bentuk excel atau pdf.

- Fungsi Print bisa digunakan jika report telah di export ke dalam bentuk excel atau pdf.

Event

- Halaman di *load*

```

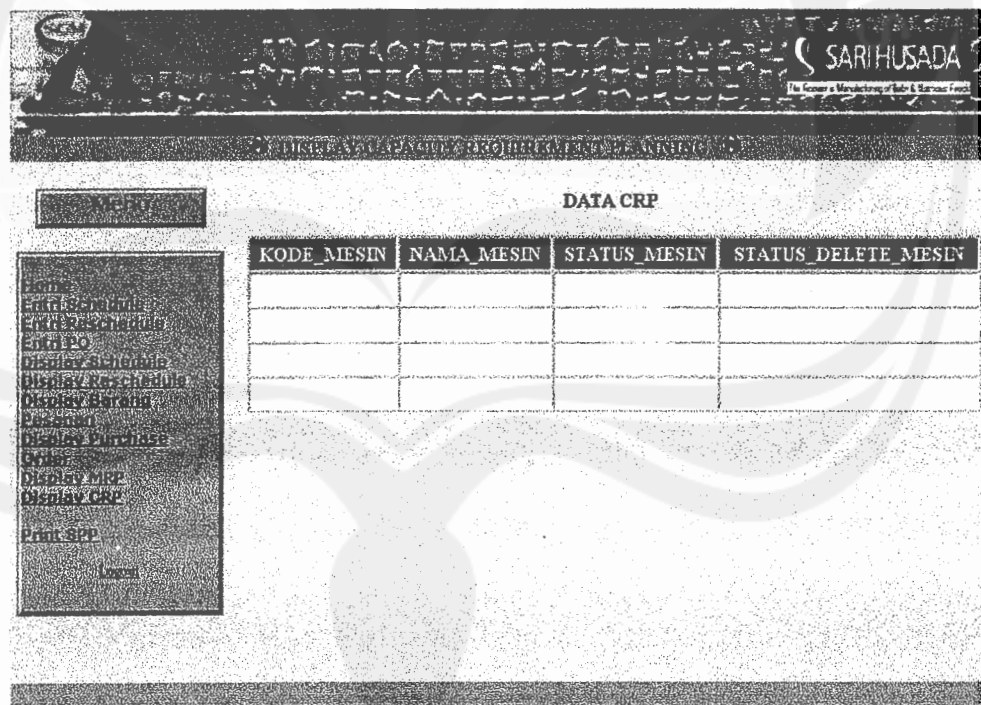
Page_Load()
{
    SQL Statement :
    SELECT * FROM Data_Rechedule_Master INNER JOIN
    Data_Rechedule_Detail ON Data_Reschedule
    _Master.Kode_Rechedule_Master = Data_Rechedule
    _Detail.Kode_Rechedule_Master
}

```

Urutan aksi yang terjadi :

1. Sistem akan menampilkan data reschedule dalam bentuk report .

4.12 Use Case : Display CRP (Capacity Requirement Planning)



Gambar 4.12 Rancangan Halaman Display CRP

Deskripsi

Program Studi Teknik Informatika	DPPL-ASCaMaDa	209/ 277
----------------------------------	---------------	----------

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika

- Halaman ini digunakan untuk menampilkan data CRP (Capacity Requirement Planning).

Event

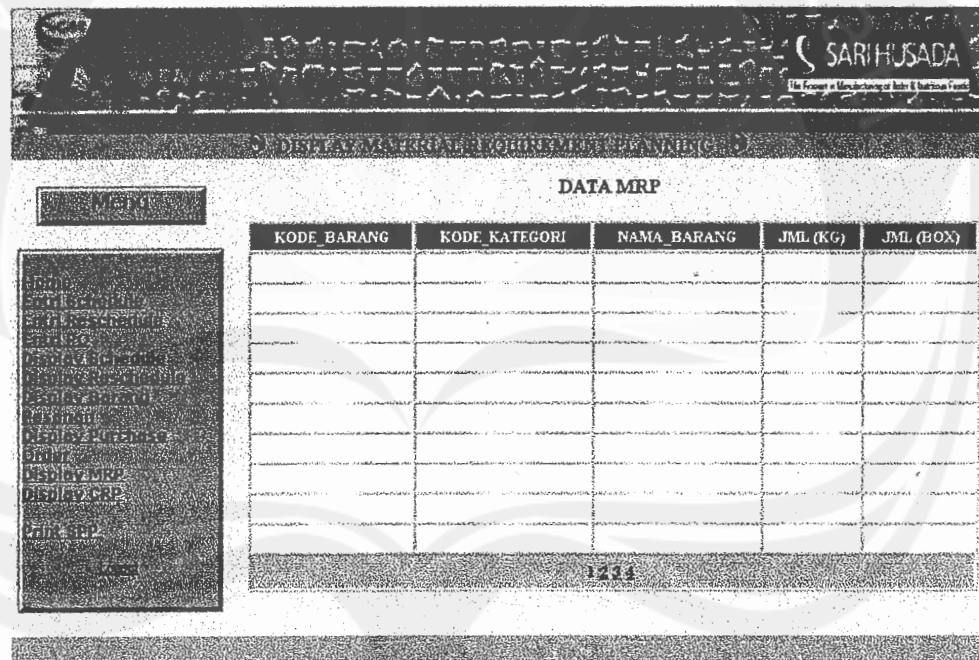
- Halaman di *load*

```
Page_Load()
{
    SQL Statement :
    SELECT * FROM Mesin WHERE Status_Delete_Mesin = 0
}
```

Urutan aksi yang terjadi :

1. Sistem akan menampilkan data CRP .

4.13 Use Case : Display MRP (Material Requirement Planning)



Gambar 4.13 Rancangan Halaman Display MRP

Deskripsi

- Halaman ini digunakan untuk menampilkan data MRP (Material Requirement Planning).

Event

- Halaman di *load*

```

Page_Load()
{
    SQL Statement :
    SELECT
    Kode_Barang,Kode_Kategori>Nama_Barang,Quantityingk,
    quantityinbox FROM Bahan_Baku_Master WHERE
    Status_Delete_BB = 0
}

```

Urutan aksi yang terjadi :

1. Sistem akan menampilkan data MRP

4.14 Use Case : Entri Purchase Order

Gambar 4.14 Rancangan Halaman Entri Purchase Order

Deskripsi

- Halaman ini digunakan untuk membuat permintaan pembelian bahan baku produksi
- Tombol Generate digunakan untuk meng-generate id.
- Tombol Add digunakan untuk menambahkan data permintaan pembelian bahan baku produksi ke basis data.

Event

- Entri PO Master

```

OnClick_Add1()
{
    SQL Statement :
    INSERT INTO Permintaan_Pembelian_Master
        VALUES ([txtid], [txttglKirim], [txttglDatang])
}

```

Urutan aksi yang terjadi :

1. User memasukkan detil data permintaan pembelian master.
2. User menekan tombol Add.
3. Sistem akan menambahkan data permintaan pembelian master ke dalam basis data.
4. Jika penambahan data permintaan pembelian master berhasil maka sistem akan menampilkan pesan dan user akan memasukkan data permintaan pembelian detail, dan jika gagal sistem akan menampilkan pesan kesalahan.

- Entri permintaan pembelian Detail

```

OnClick_Add2()
{
    SQL Statement :
    INSERT INTO Permintaan_Pembelian_Detail
        VALUES ([txtidDet], [ddlBarang], [txtid], [txtjml], 0
    )
}

```

Urutan aksi yang terjadi :

1. User memasukkan detil data permintaan pembelian detail.
2. User menekan tombol Add
3. Sistem akan menambahkan data permintaan pembelian detail ke dalam basis data.
4. Jika penambahan data permintaan pembelian detail berhasil maka sistem akan menampilkan pesan dan jika gagal sistem akan menampilkan pesan kesalahan.

4.15 Use Case : Display PO

PERMINTAAN PEMBELIAN

Friday, July 06, 2007

ID PESANAN	TANGGAL	TGL AMBIL	ID	KODE PRODUK	NAMA PRODUK	JUMLAH
1	3/3/2007	6/6/2007	1	101	SOM2	10
3	6/19/2007	0/21/2007	2	101	SOM2	10
			3	100	SGM1	100
4	6/21/2007	6/30/2007	4	101	SOM2	300
			5	100	SGM1	300
5	7/2/2007	6/29/2007	6	101	SOM2	1000
6	7/3/2007	7/20/2007	7	101	SOM2	2000

Gambar 4.15 Rancangan Halaman Display PO

Deskripsi

- Halaman ini digunakan untuk menampilkan dan mencetak data Permintaan pembelian.
- Tombol Export digunakan untuk mengubah report ke dalam bentuk excel atau pdf.
- Fungsi Print bisa digunakan jika report telah di export ke dalam bentuk excel atau pdf.

Event

- Halaman di *load*

```
Page_Load()
{
```

```
    SQL Statement :
    SELECT * FROM Permintaan_Pembelian_Master INNER
    JOIN Permintaan_Pembelian_Detail ON
    Permintaan_Pembelian_Master.Kode_PP_Master =
```

Program Studi Teknik Informatika	DPPL-ASCaMaDa	213/ 277
----------------------------------	---------------	----------

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UJAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika

```

Permintaan_Pembelian_Detail.Kode_PP_Master WHERE
Status_PP = 0
}

```

Urutan aksi yang terjadi :

1. Sistem akan menampilkan data permintaan pembelian dalam bentuk report .

4.16 Use Case : Surat Perintah Produksi

The screenshot shows a web-based form for creating a Production Order (SPP). The form is titled 'Surat Perintah Produksi' and is part of the 'SARIHUSADA' system. The header includes the company name 'SARIHUSADA' and the tagline 'The Power of Manufacturing with Electronic Form'. The form contains several input fields for identifying the SPP and its schedule. A central section is dedicated to managing raw materials ('Jumlah Bahan Baku'), featuring a list of materials, a 'Save' button, and navigation arrows to add or remove items from the list.

Gambar 4.16 Rancangan Halaman Surat Perintah Produksi

Deskripsi

- Halaman ini digunakan untuk membuat surat perintah produksi.
- Tombol Add digunakan untuk menambahkan data surat perintah produksi ke basis data.
- Tombol panah ke kanan (>) digunakan untuk menambahkan data bahan baku ke list di sebelah kanan, sedangkan tombol panah ke kiri (<) digunakan untuk membatalkan pemilihan.

Event

Program Studi Teknik Informatika	DPPL-ASCaMaDa	214/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- **Entri SPP Master**

```
OnClick_Add1()  
{  
    SQL Statement :  
    INSERT INTO SPP_Master  
        VALUES([txtid],[txtidSche],[txtidResche],[txttgl  
Spp],[txttglAkhir])  
}
```

Urutan aksi yang terjadi :

1. User memasukkan detil data SPP master.
2. User menekan tombol Add.
3. Sistem akan menambahkan data SPP master ke dalam basis data.
4. Jika penambahan data SPP master berhasil maka sistem akan menampilkan pesan dan user akan memasukkan data SPP detail, dan jika gagal sistem akan menampilkan pesan kesalahan.

- **Entri SPP Detail**

```
OnClick_Add2()  
{  
    SQL Statement :  
    INSERT INTO SPP_Detail  
        VALUES([txtidDet],[txtid],[ddlBarang],[txtjml],0  
    )  
}
```

Urutan aksi yang terjadi :

1. User memasukkan detil data SPP detail.
2. User menekan tombol Add
3. Sistem akan menambahkan data SPP detail ke dalam basis data.
4. Jika penambahan data SPP detail berhasil maka sistem akan menampilkan pesan dan user akan memasukkan data Bahan baku yang diperlukan dan jika gagal sistem akan menampilkan pesan kesalahan.

- **Entri Bahan Baku**

```
OnClick_Save()
```

Program Studi Teknik Informatika	DPPL-ASCaMaDa	215/ 277
----------------------------------	---------------	----------

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika


```

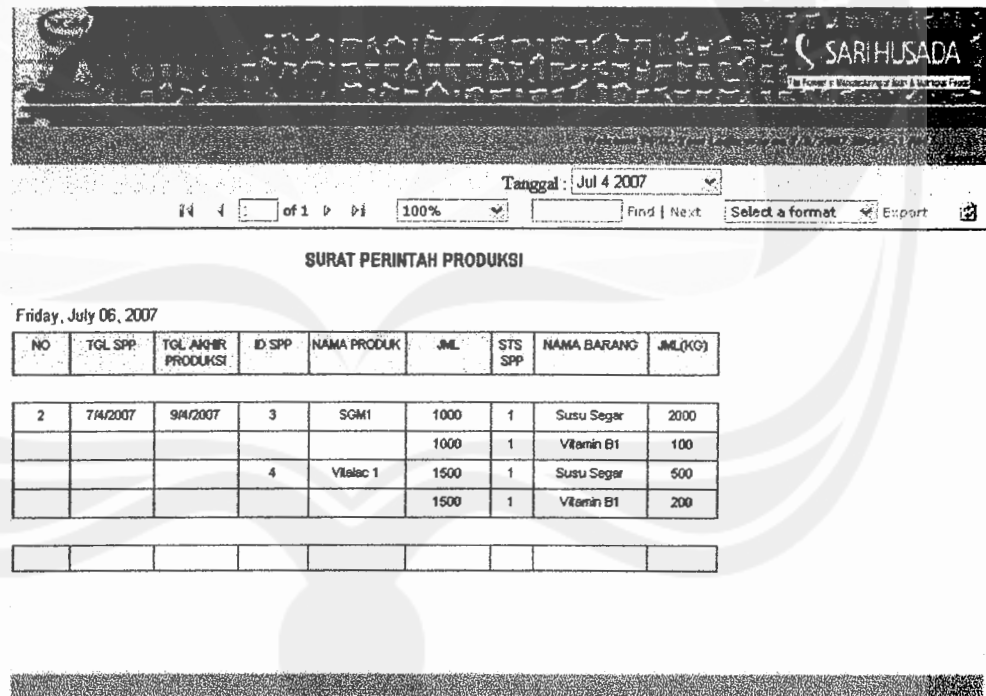
{
  SQL Statement :
  INSERT INTO Bahan_Baku_Keluar
    VALUES ([txtidDet],[idBrg],[Qinkg],[Qinbox],0)
}

```

Urutan aksi yang terjadi :

1. User memasukkan detail data bahan baku produksi.
2. User menekan tombol Save
3. Sistem akan menambahkan data bahan baku ke dalam basis data.
4. Jika penambahan data bahan baku berhasil maka sistem akan menampilkan pesan dan jika gagal sistem akan menampilkan pesan kesalahan.

4.17 Use Case : Print SPP



Gambar 4.17 Rancangan Halaman Print SPP

Deskripsi

- Halaman ini digunakan untuk menampilkan dan mencetak surat perintah produksi.

- Combobox digunakan untuk memilih tanggal SPP.
- Tombol Export digunakan untuk mengubah report ke dalam bentuk excel atau pdf.
- Fungsi Print bisa digunakan jika report telah di export ke dalam bentuk excel atau pdf.

Event

- Tampil Report

```

OnClick_Refresh()
{
    SQL Statement :.
    SELECT * FROM SPP_Master INNER JOIN SPP_Detail ON
    SPP_Master.Kode_SPP_Master =
    SPP_Detail.Kode_SPP_Master INNER JOIN
    BHN_BAKU_KELUAR ON SPP_Detail.KodeSPP_Detail =
    Bhn_Baku_Keluar.Kode_SPP_Detail WHERE
    SPP_Master.TGL_SPP = [ddlTgl]
}

```

Urutan aksi yang terjadi :

1. User memilih tanggal SPP kemudian menekan tombol refres.
2. Sistem akan menampilkan data permintaan SPP dalam bentuk report .
3. User mengubah format report menjadi pdf dengan menekan tombol export, lalu memilih print.

4.18 Use Case : Add Produk

The screenshot shows a web interface for adding a product. At the top, there is a header with the text 'SARIHUSADA' and a logo. Below the header, there is a form with three input fields: 'ID Produk' (with a small dropdown menu), 'Nama Produk', and 'Keterangan'. Below these fields are two buttons: 'Add' and 'Cancel'. To the left of the form, there is a large, dark, illegible area, possibly representing a list of existing products or a sidebar menu.

Gambar 4.18 Rancangan Halaman Add Produk

Deskripsi

- Halaman ini digunakan untuk melakukan penambahan produk.
- Tombol Add digunakan untuk menyimpan data produk baru.

Event

- Tambah Produk

```
OnClick_Add()  
{  
    SQL Statement :  
    INSERT INTO Produk  
    VALUES ([id],[txtNama],[txtKet],0,0,0,0,0,0)  
}
```

Urutan aksi yang terjadi :

1. User memasukkan nama produk dan keterangan, id produk akan di-generate secara otomatis.
2. User menekan tombol Add.
3. Sistem akan menambahkan data produk yang baru ke basis data.

4. Jika penambahan data produk berhasil maka sistem akan menampilkan pesan, dan jika gagal sistem akan menampilkan pesan kesalahan.

4.19 Use Case : Edit Produk

DATA PRODUK

	ID	NAMA	KETERANGAN	HARGA	STOK TOTAL	STS DEL	STS PENJUALAN
Edit	100	SGM1	Bayi dibawah 6 bulan	25000	3500	0	1
Edit	101	SGM2	Bayi 6 bula -3 tahun	14000	2300	0	1
Edit	102	SGM 3	Anak usia 1 -5 tahun ke atas	12000	1200	0	0
Edit	103	SGM 4	Anak usia 4 athun ke atas	20000	2100	0	1
Edit	104	Vitalac 1	Bayi dibawah 6 bulan	0	2500	0	0
Edit	105	Vitalac 2	Bayi 6 bulan -3 tahun	0	200	0	0
Edit	106	Vitalac 3	Anak usia 1 tahun ke atas	0	0	0	0
Edit	107	Lactamil Coklat 1	Ibu selama menyusui	0	0	0	0
Edit	108	Lactamil Coklat 2	Ibu selama kehamilan	0	0	0	0
Edit	109	Lactamil Vanilla 1	Ibu selama menyusui		0	0	0

Gambar 4.19 Rancangan Halaman Edit Produk

Deskripsi

- Halaman ini digunakan untuk melakukan proses *editing* terhadap data produk di basis data.
- Tombol Edit digunakan untuk melakukan perubahan data produk.

Event

- Edit Produk

OnClick_Edit()

{

SQL Statement :

UPDATE Produk

SET Nama_Produk = [NamaProduk],

```

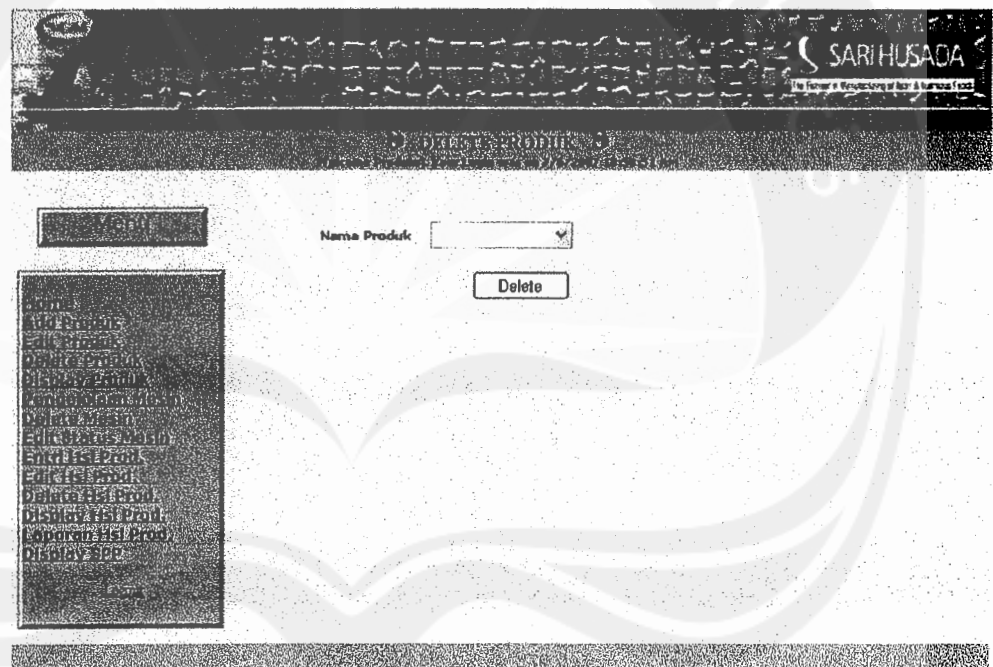
Keterangan = [Ket],
Harga = [Harga]
WHERE Kode_Produk = [idProduk]
}

```

Urutan aksi yang terjadi :

1. User menekan tombol Edit, kemudian melakukan perubahan terhadap data tersebut.
2. Sistem akan melakukan perubahan data produk yang dipilih.
3. Jika pengubahan data produk berhasil maka datagrid akan menampilkan data hasil perubahan.

4.20 Use Case : Delete Produk



Gambar 4.20 Rancangan Halaman Delete Produk

Deskripsi

- Halaman ini digunakan untuk menghapus data produk di basis data.
- Combobox digunakan untuk memilih produk yang akan dihapus.
- Tombol Delete digunakan untuk menghapus data produk.

Program Studi Teknik Informatika	DPPL-ASCaMaDa	220/ 277
----------------------------------	---------------	----------

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika

Event

- Delete User

```

OnClick_Delete()
{
    SQL Statement :
    UPDATE Produk
    SET Status_Delete_BP = 1
    WHERE Kode_Produk = [ddlProduk]
}
    
```

Urutan aksi yang terjadi :

1. User memilih data yang akan dihapus.
2. User menekan tombol Delete untuk menghapus produk.
3. Jika *delete* data produk berhasil maka sistem akan menampilkan pesan, dan jika gagal sistem akan menampilkan pesan kesalahan.

4.21 Use Case : Display Produk

ID	NAMA	KETERANGAN	HARGA	STOK	STS DEL	STS PNJ
100	SGM1	Bayi dibawah 6 bulan	25000	3500	0	1
101	SGM2	Bayi 6 bula - 3 tahun	14000	2300	0	1
102	SGM 3	Anak usia 1 - 5 tahun ke atas	12000	1200	0	0
103	SGM 4	Anak usia 4 athun ke atas	20000	2100	0	1
104	Vitalac 1	Bayi dibawah 6 bulan	0	2500	0	0
105	Vitalac 2	Bayi 6 bulan - 3 tahun	0	200	0	0
106	Vitalac 3	Anak usia 1 tahun ke atas	0	0	0	0
107	Lactamil Coklat 1	Ibu selama menyusui	0	0	0	0
108	Lactamil Coklat 2	Ibu selama kehamilan	0	0	0	0
109	Lactamil Vanilla 1	Ibu selama menyusui	0	0	0	0

Gambar 4.21 Rancangan Halaman Display Produk

Deskripsi

Program Studi Teknik Informatika	DPPL-ASCaMaDa	2211/277
----------------------------------	---------------	----------

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika

- Tombol Cek digunakan untuk mengambil data produk sesuai dengan id SPP.
- Tombol Add digunakan untuk menambahkan data hasil produksi ke basis data.

Event

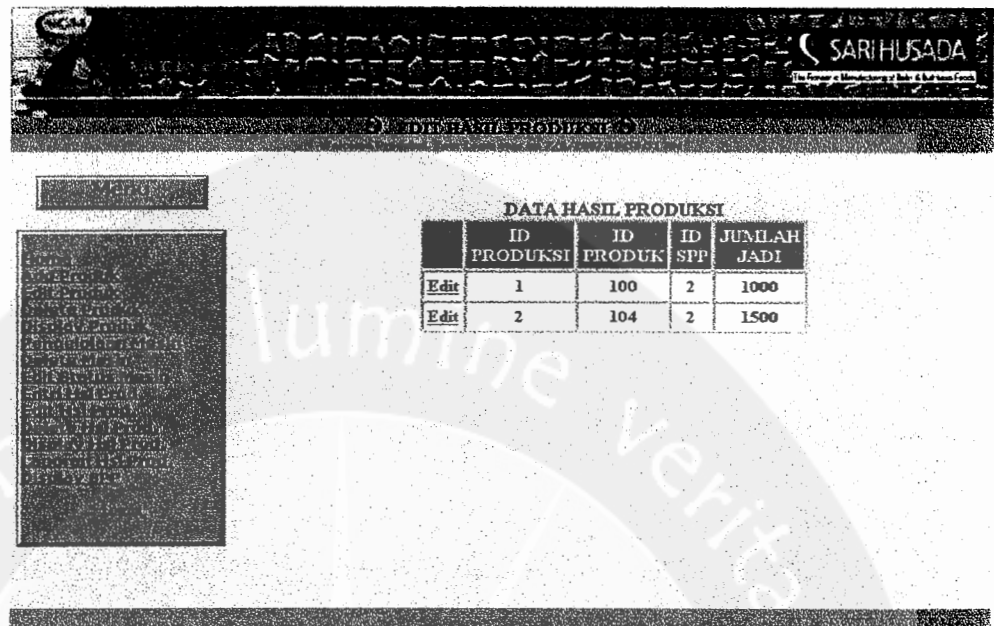
- Entri Hasil Produksi

```
OnClick_Add()
{
    SQL Statement :
    INSERT INTO Barang_Hasil_Produksi
    VALUES ([txtid], [txtidspp], [ddlproduk], [txtjml], 0
    )
}
```

Urutan aksi yang terjadi :

1. User memasukkan detail data hasil produksi.
2. User menekan tombol Add.
3. Sistem akan menambahkan data hasil produksi ke dalam basis data.
4. Jika penambahan data hasil produksi berhasil maka sistem akan menampilkan pesan dan jika gagal sistem akan menampilkan pesan kesalahan.

4.23 Use Case : Edit Hasil Produksi



Gambar 4.23 Rancangan Halaman Editing Hasil Produksi

Deskripsi

- Halaman ini digunakan untuk melakukan proses *editing* terhadap data hasil produksi di basis data.
- Tombol Edit digunakan untuk melakukan perubahan data hasil produksi.

Event

- Edit Hasil Produksi

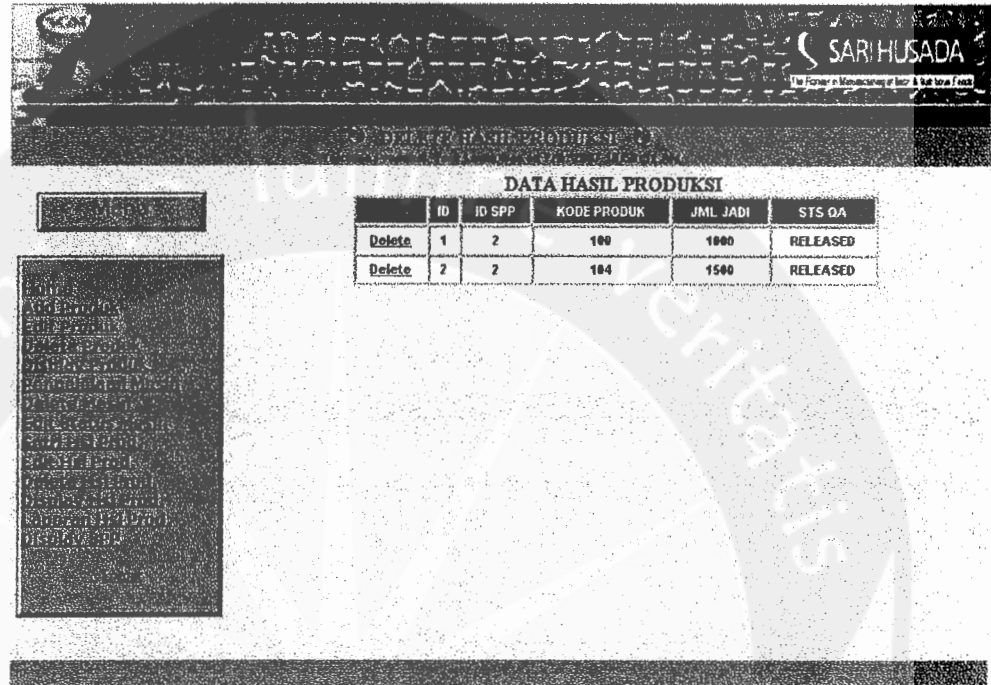
```
onClick_Edit()  
{  
  SQL Statement :  
  UPDATE Barang_Hasil_Produksi  
  SET Jumlah_Jadi = [jml],  
  WHERE Kode_Produksi = [idProduksi]  
}
```

Urutan aksi yang terjadi :

1. User menekan tombol Edit, kemudian memasukkan data hasil produksi yang baru.
2. Sistem akan melakukan perubahan data hasil produksi yang dipilih.

3. Jika perubahan data hasil produksi berhasil maka datagrid akan menampilkan data hasil perubahan.

4.24 Use Case : Delete Hasil Produksi



Gambar 4.24 Rancangan Halaman Delete Hasil Produksi

Deskripsi

- Halaman ini digunakan untuk menghapus data hasil produksi di basis data.
- Combobox digunakan untuk memilih produk yang akan dihapus.
- Tombol Delete digunakan untuk menghapus data produk.

Event

- Delete User

```

OnClick_Delete ()
{
    SQL Statement :
    UPDATE Produk
    SET Status_Delete_BP = 1
    WHERE Kode_Produk = [ddlProduk]
}

```

Urutan aksi yang terjadi :

4. User memilih data yang akan dihapus.
5. User menekan tombol Delete untuk menghapus produk.
6. Jika *delete* data produk berhasil maka sistem akan menampilkan pesan, dan jika gagal sistem akan menampilkan pesan kesalahan.

4.25 Use Case : Display Hasil Produksi

ID	STP	ID PRODUK	PRODUK	JUMLAH	STS QA
1	2	100	SGMI	1000	RELEASED
2	2	104	Vitalac 1	1500	RELEASED

Gambar 4.25 Rancangan Halaman Display Hasil Produksi

Deskripsi

- Halaman ini digunakan untuk menampilkan data hasil produksi di basis data.

Event

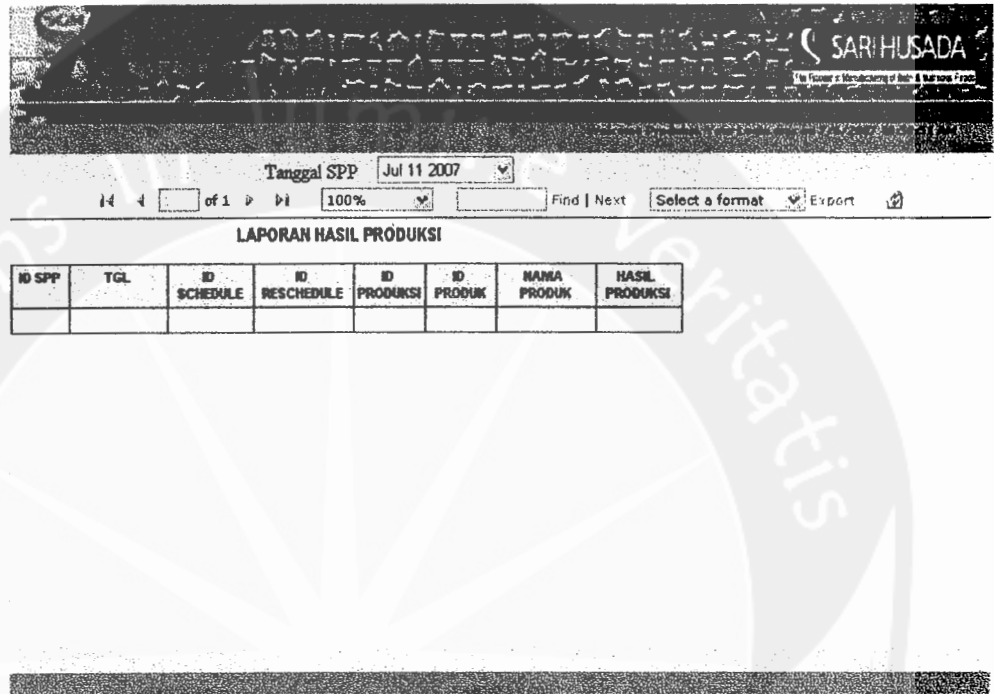
- Halaman di *load*

```
Page_Load()  
{  
    SQL Statement :  
    SELECT * FROM Barang_Hasil_Produksi  
}
```

Urutan aksi yang terjadi :

1. Sistem akan menampilkan data hasil produksi yang terdapat di basisdata.

4.26 Use Case : Laporan Hasil Produksi



Gambar 4.26 Rancangan Halaman Laporan Hasil Produksi

Deskripsi

- Halaman ini digunakan untuk menampilkan dan mencetak laporan hasil produksi.
- Combobox digunakan untuk memilih tanggal Laporan hasil produksi.
- Tombol Export digunakan untuk mengubah report ke dalam bentuk excel atau pdf.
- Fungsi Print bisa digunakan jika report telah di export ke dalam bentuk excel atau pdf.

Event

Program Studi Teknik Informatika	DPPL-ASCaMaDa	2271/277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- Tampil Report

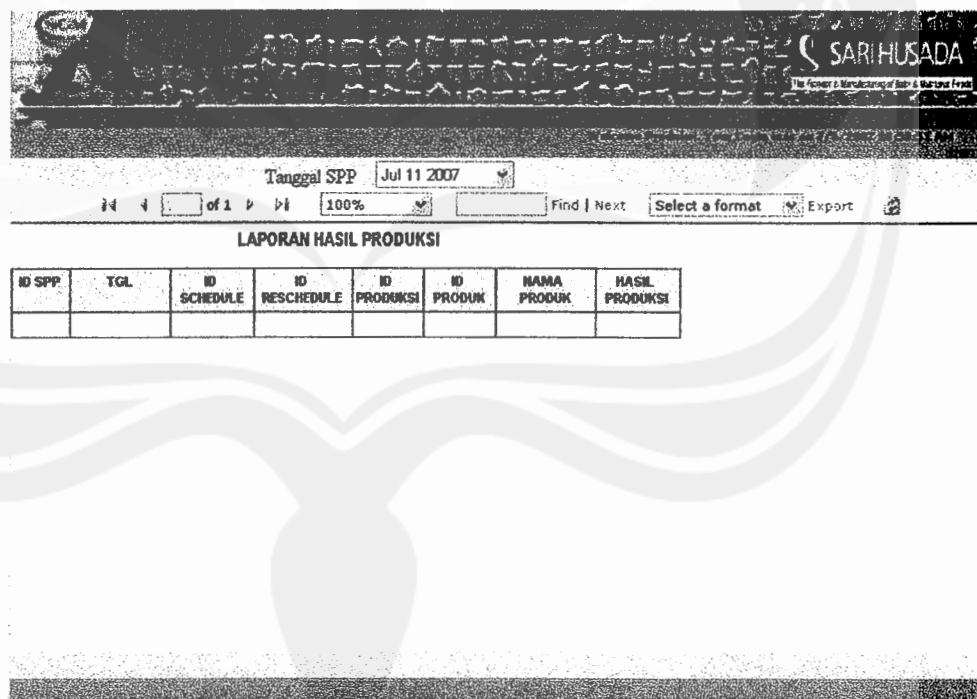
OnClick_Refresh()

```
{
  SQL Statement :
  SELECT * FROM SPP_Master INNER JOIN SPP_Detail ON
  SPP_Master.Kode_SPP_Master =
  SPP_Detail.Kode_SPP_Master INNER JOIN
  Barang_Hasil_Produksi ON SPP_Master.KodeSPP_Master
  = Barang_Hasil_Produksi.Kode_SPP_Master WHERE
  SPP_Master.TGL_SPP = [ddlTgl] AND
  Barang_Hasil_Produksi.Status_QA_BP='RELEASED'
}
```

Urutan aksi yang terjadi :

- 1 User memilih tanggal Laporan kemudian menekan tombol refresh.
- 2 Sistem akan menampilkan data hasil produksi dalam bentuk report .

4.27 Use Case: Print Hasil Produksi



Gambar 4.27 Rancangan Halaman Print Hasil Produksi

Deskripsi

Program Studi Teknik Informatika	DPPL-ASCaMaDa	228/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- Halaman ini digunakan untuk menampilkan dan mencetak laporan hasil produksi.
- Combobox digunakan untuk memilih tanggal Laporan hasil produksi.
- Tombol Export digunakan untuk mengubah report ke dalam bentuk excel atau pdf.
- Fungsi Print bisa digunakan jika report telah di export ke dalam bentuk excel atau pdf.

Event

- Tampil Report

OnClick_Refresh()

```
{
    SQL Statement :
    SELECT * FROM SPP_Master INNER JOIN SPP_Detail ON
    SPP_Master.Kode_SPP_Master =
    SPP_Detail.Kode_SPP_Master INNER JOIN
    Barang_Hasil_Produksi ON SPP_Master.KodeSPP_Master
    = Barang_Hasil_Produksi.Kode_SPP_Master WHERE
    SPP_Master.TGL_SPP = [ddlTgl] AND
    Barang_Hasil_Produksi.Status_QA_BP='RELEASED'
}
```

Urutan aksi yang terjadi :

1. User memilih tanggal Laporan kemudian menekan tombol refresh.
2. Sistem akan menampilkan data hasil produksi dalam bentuk report .
3. User menekan tombol export untuk mengubah ke format pdf lalu print

4.28 Use Case : Add Mesin

	KODE MESIN	NAMA MESIN	STS MESIN	STS DEL
Edt	aa	asf	READY	1
Edt	B2314DEF	Boller Freze	READY	0
Edt	M456DF	MIXER	READY	0
Edt	SUS23	Suspensi	Use	1

Gambar 4.28 Rancangan Halaman Add Mesin

Deskripsi

- Halaman ini digunakan untuk melakukan penambahan data mesin.
- Tombol Add digunakan untuk menyimpan data mesin baru.

Event

- Tambah Mesin

```
OnClick_Add()  
{  
    SQL Statement :  
    INSERT INTO Mesin  
    VALUES ([id],[txtNama], 'READY', 0)
```

Urutan aksi yang terjadi :

1. User memasukkan id mesin dan nama mesin.
2. User menekan tombol Add.

3. Sistem akan menambahkan data mesin yang baru ke basis data.
4. Jika penambahan data produk berhasil maka sistem akan menampilkan pesan, dan jika gagal sistem akan menampilkan pesan kesalahan.

4.29 Use Case : Edit Mesin

DATA MESIN PRODUKSI				
	KODE MESIN	NAMA MESIN	STS MESIN	STS DEL
Edit	as	asf	READY	1
Edit	B2314DEF	Boiler Freze	READY	0
Edit	M455DF	MIXER	READY	0
Edit	SUS23	Suspensi	Use	1

Gambar 4.29 Rancangan Halaman Edit Mesin

Deskripsi

- Halaman ini digunakan untuk melakukan proses *editing* terhadap data mesin di basis data.
- Tombol Edit digunakan untuk melakukan perubahan data mesin.

Event

- Edit Mesin

- Combobox digunakan untuk memilih mesin yang akan dihapus.
- Tombol Delete digunakan untuk menghapus data mesin.

Event

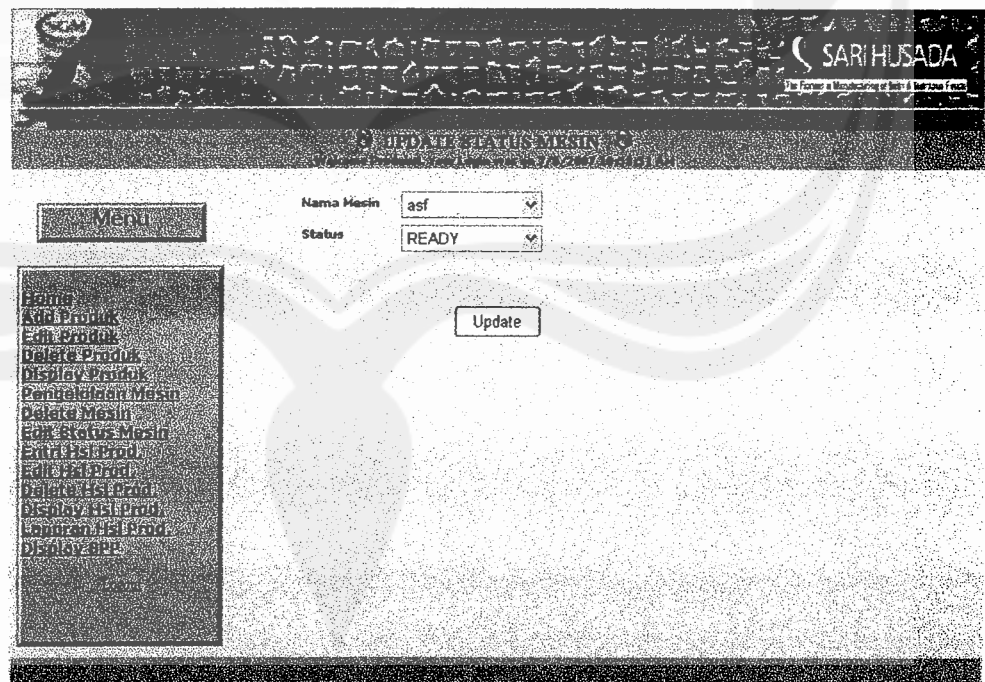
- Delete Mesin

```
OnClick_Delete()
{
    SQL Statement :
    UPDATE Mesin
    SET Status_Delete_Mesin = 1
    WHERE Kode_Mesin = [idMesin]
}
```

Urutan aksi yang terjadi :

1. User memilih data yang akan dihapus.
2. User menekan tombol Delete untuk menghapus mesin.
3. Jika *delete* data mesin berhasil maka sistem akan menampilkan pesan, dan jika gagal sistem akan menampilkan pesan kesalahan.

4.31 Use Case : Edit Status Mesin



Gambar 4.31 Rancangan Halaman Edit Status Mesin

Deskripsi

- Halaman ini digunakan untuk mengubah status mesin dari Ready ke use atau sebaliknya.
- Combobox digunakan untuk memilih mesin dan status mesin.
- Tombol Update digunakan untuk melakukan perubahan data.

Event

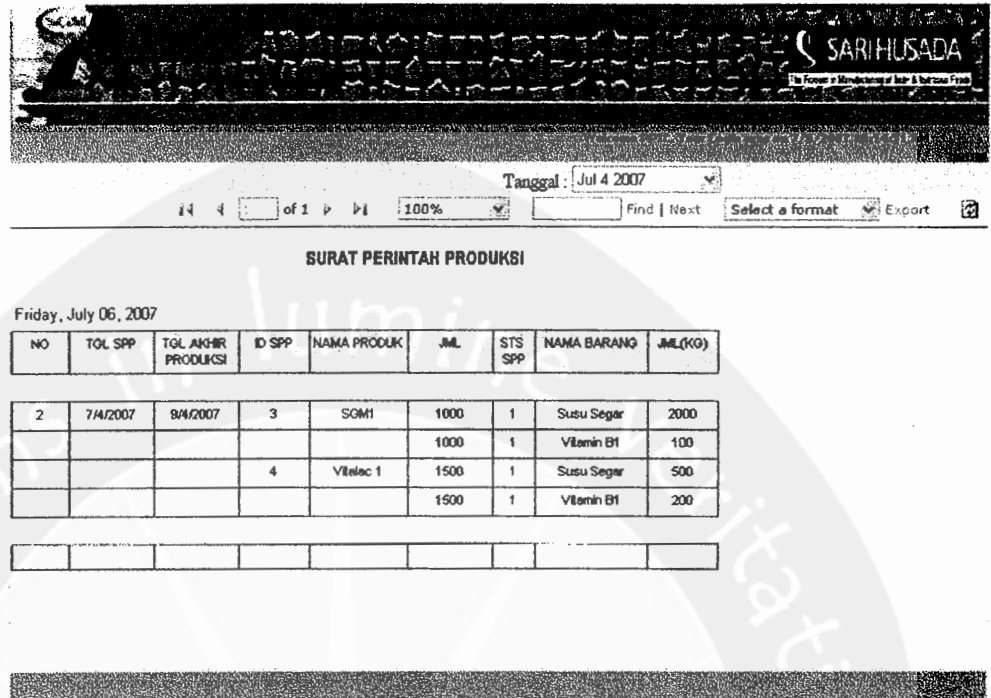
- Update Mesin

```
Page_Load()  
{  
    SQL Statement :  
    UPDATE Mesin SET Status_Mesin = [dllStatus]  
    WHERE Kode_Mesin = [idMesin]  
}
```

Urutan aksi yang terjadi :

1. User memilih data yang akan diubah dan melakukan input data perubahan.
2. User menekan tombol Update untuk mengubah status mesin.
3. Jika *update* data status mesin berhasil maka sistem akan menampilkan pesan, dan jika gagal sistem akan menampilkan pesan kesalahan.

4.32 Use Case : Display SPP



Gambar 4.32 Rancangan Halaman Display SPP

Deskripsi

- Halaman ini digunakan untuk menampilkan dan mencetak surat perintah produksi.
- Combobox digunakan untuk memilih tanggal SPP.
- Tombol Export digunakan untuk mengubah report ke dalam bentuk excel atau pdf.
- Fungsi Print bisa digunakan jika report telah di export ke dalam bentuk excel atau pdf.

Event

- Tampil Report

```
OnClick_Refresh()  
{
```

```
    SQL Statement :  
    SELECT * FROM SPP_Master INNER JOIN SPP_Detail ON  
    SPP_Master.Kode_SPP_Master =  
    SPP_Detail.Kode_SPP_Master INNER JOIN  
    BHN_BAKU_KELUAR ON SPP_Detail.KodeSPP_Detail =  
    Bhn_Baku_Keluar.Kode_SPP_Detail WHERE  
    SPP_Master.TGL_SPP = [ddlTgl]
```

Urutan aksi yang terjadi :

1. User memilih tanggal SPP kemudian menekan tombol refresh.
2. Sistem akan menampilkan data permintaan SPP dalam bentuk report .
3. User mengubah format report menjadi pdf dengan menekan tombol export untuk mencetak SPP lalu memilih print.

4.33 Use Case : Add Bahan Produksi

The screenshot shows a web application interface for adding production materials. At the top, there is a header with the logo of 'SARI HUSADA' and the text 'The Power of Manufacturing of Better & Better Food'. Below the header, the page title is 'Add Bahan Produksi'. The main form area contains the following fields:

- ID Bahan Produk:
- Nama Bahan Produk:
- Rak:
- Kategori:

At the bottom right of the form are two buttons: 'Add' and 'Cancel'. On the left side, there is a sidebar menu with the following items:

- Home
- Dashboard
- Menu
- Produk
- Order
- Supplier
- Log
- Logout

Gambar 4.33 Rancangan Halaman Add Bahan Produksi

Deskripsi

- Halaman ini digunakan untuk melakukan penambahan data bahan produksi.
- Combobox digunakan untuk memilih kategori bahan baku.

- Tombol Add digunakan untuk menyimpan data bahan baku baru.

Event

- Tambah Bahan Baku

```
OnClick_Add()
{
```

```
    SQL Statement :
    INSERT INTO Bahan_Baku_Master
    VALUES ([id],[ddlKtgr],[txtNama],0,0,[txtRak],0,0,0)
```

Urutan aksi yang terjadi :

1. User memasukkan nama bahan baku, rak dan kategori.
2. User menekan tombol Add.
3. Sistem akan menambahkan data bahan baku yang baru ke basis data.
4. Jika penambahan data bahan baku berhasil maka sistem akan menampilkan pesan, dan jika gagal sistem akan menampilkan pesan kesalahan.

4.34 Use Case : Edit Bahan Baku

	ID	KTRG	NAMA	JML(KG)	JML(BOX)	HARGA	RAK	DATE	STS BEL
Edit	B1001	K100	Susu Segar	5000	0	1000	R1001	Jun 20 2007	0
Edit	B1002	K100	Deminal	5500	0	1500	R1002	Jun 21 2007	0
Edit	B1003	K100	Lactose	2000	0	1300	R1003	Jun 20 2007	0
Edit	B1004	K100	Sodium	3200	0	2000	R1004	Jun 20 2007	0
Edit	B1005	K100	SMP	4000	0	5200	R1005	Jun 20 2007	0
Edit	B1006	K100	FCMP	1000	0	1000	R1006	Jun 20 2007	0
Edit	B2001	K200	Beras Merah	1200	0	1000	R2001	Juz 20 2007	0
Edit	B2002	K200	Beras Putih	1600	0	2000	R2002	Jun 20 2007	0
Edit	B2003	K200	Cocoa Powder	3000	0	3000	R2003	Jun 20 2007	0
Edit	B2004	K200	Gula Pasir	10000	0	1500	R2004	Jun 20 2007	0

Gambar 4.34 Rancangan Halaman Edit Bahan Baku

Deskripsi

- Halaman ini digunakan untuk melakukan proses *editing* terhadap data bahan baku di basis data.
- Tombol Edit digunakan untuk melakukan perubahan data mesin.

Event

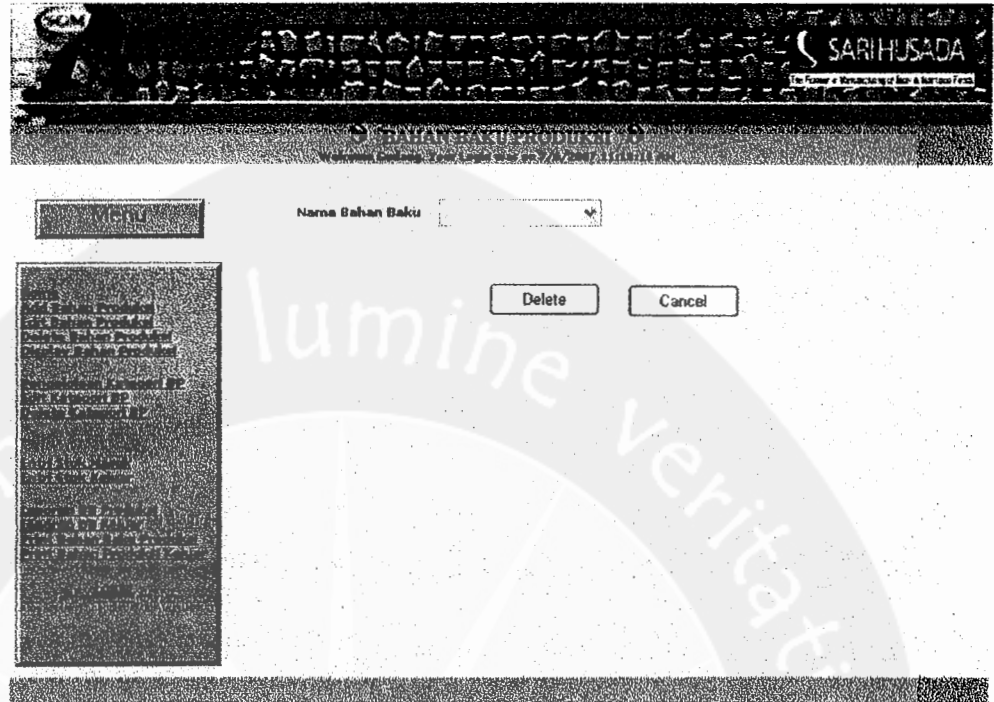
- Edit Bahan Baku

```
OnClick_Edit()  
{  
    SQL Statement :  
    UPDATE Bahan_Baku_Master  
    SET Nama_Barang = [Nama], Rak = [rak],  
    Status_Delete_BB = [status]  
    WHERE Kode_Barang = [idBarang]  
}
```

Urutan aksi yang terjadi :

1. User menekan tombol Edit, kemudian melakukan perubahan terhadap data tersebut.
2. Sistem akan melakukan perubahan data bahan baku yang dipilih.
3. Jika perubahan data bahan baku berhasil maka datagrid akan menampilkan data hasil perubahan.

4.35 Use Case : Delete Bahan Baku



Gambar 4.35 Rancangan Halaman Delete Bahan Baku

Deskripsi

- Halaman ini digunakan untuk menghapus data bahan baku di basis data.
- Combobox digunakan untuk memilih bahan baku yang akan dihapus.
- Tombol Delete digunakan untuk menghapus data bahan baku.

Event

- Delete Bahan Baku

```
OnClick_Delete()  
{  
    SQL Statement :  
    UPDATE Bahan_Baku_Master  
    SET Status_Delete_BB = 1  
    WHERE Kode_Barang = [idBarang]  
}
```

Urutan aksi yang terjadi :

1. User memilih data yang akan dihapus.

2. User menekan tombol Delete untuk menghapus bahan baku.
3. Jika *delete* data bahan baku berhasil maka sistem akan menampilkan pesan, dan jika gagal sistem akan menampilkan pesan kesalahan.

4.36 Use Case : Display Bahan Produksi

ID	KTRR	NAMA	JML (KG)	JML (BOX)	RAK	TANGGAL	HARGA
B1001	K100	Susu Segar	4000	0	R1001	Jun 20 2007	1000
B1002	K100	Deminal	5500	0	R1002	Jun 21 2007	1500
B1003	K100	Lactose	2000	0	R1003	Jun 20 2007	1300
B1004	K100	Sodium	3200	0	R1004	Jun 20 2007	2000
B1005	K100	SMP	4000	0	R1005	Jun 20 2007	5200
B1006	K100	FCMP	1000	0	R1006	Jun 20 2007	1000
B2001	K200	Beras Merah	1200	0	R2001	Jun 20 2007	1000
B2002	K200	Beras Putih	1000	0	R2002	Jun 20 2007	2000
B2003	K200	Cocoa Powder	3000	0	R2003	Jun 20 2007	3000
B2004	K200	Gula Pasir	10000	0	R2004	Jun 20 2007	1500

Gambar 4.36 Rancangan Halaman Display Bahan Produksi

Deskripsi

- Halaman ini digunakan untuk menampilkan data bahan produksi di basis data.

Event

- Halaman di *load*

```

Page_Load()
{
    SQL Statement :
    SELECT * FROM Bahan_Baku_Master
}

```

Urutan aksi yang terjadi :

1. Sistem akan menampilkan data bahan baku yang terdapat di basisdata.

4.37 Use Case : Entri Stock Keluar

The screenshot shows a web application interface for 'Entri Stock Keluar'. At the top, there is a header with the 'SARIHUSADA' logo and the text 'SARIPUSADA'. Below the header, there is a table with columns for 'No', 'Nama Produk', 'Nama Bahan', 'Jumlah In KG', and 'Jumlah In Box'. To the right of the table, there is a form with four input fields: 'Nama Produk' (a dropdown menu), 'Nama Bahan' (a dropdown menu), 'Jumlah In KG' (a text input), and 'Jumlah In Box' (a text input). Below the form, there are two buttons: 'Add' and 'Cancel'.

Gambar 4.37 Rancangan Halaman Entri Stock Keluar

Deskripsi

- Halaman ini digunakan untuk memasukkan data stok bahan baku keluar untuk proses produksi.
- Combobox digunakan untuk memilih data produk dan bahan baku.
- Tombol Add digunakan untuk menambahkan data stock keluar ke basis data.

Event

- Entri Stock Keluar
onClick_Add()

```

{
  SQL Statement :
  UPDATE Bhn_Baku_Keluar
  SET STATUSKELUAR = '1'
  WHERE Kode_SPP_Detail = [kodeSPP]
}

```

Urutan aksi yang terjadi :

1. User memasukkan detil data stock bahan baku keluar.
2. User menekan tombol Add.
3. Sistem akan menambahkan data stock bahan baku keluar ke dalam basis data.
4. Jika penambahan data berhasil maka sistem akan menampilkan pesan dan jika gagal sistem akan menampilkan pesan kesalahan

4.38 Use Case : Entri Stock Masuk

Gambar 4.38 Rancangan Halaman Entri Stock Masuk

Deskripsi

Program Studi Teknik Informatika	DPPL-ASCaMaDa	242/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- Halaman ini digunakan untuk memasukkan data stok bahan baku masuk kiriman dari supplier.
- Combobox digunakan untuk memilih data bahan baku dan supplier.
- Tombol Add digunakan untuk menambahkan data stock masuk ke basis data.

Event

- Entri Stock Masuk

```
OnClick_Add()
{
    SQL Statement :
    UPDATE Bahan_Baku_Master
    SET Quantitiinkg = [txtqinkg],Quantityinbox =
    [txtqinbox]
    WHERE Kode_Barang = [ddlBrg]
}
```

Urutan aksi yang terjadi :

1. User memasukkan detil data stock bahan baku masuk.
2. User menekan tombol Add.
3. Sistem akan menambahkan data stock bahan baku masuk ke dalam basis data.
4. Jika penambahan data berhasil maka sistem akan menampilkan pesan dan jika gagal sistem akan menampilkan pesan kesalahan.

4.39 Use Case : Add Kategori

	KODE_KATEGORI	NAMA_KATEGORI	DESKRIPSI	STATUS_DELETE_KATEGORI
Edit	AA	ASAS	ASASA	1
Edit	K100	Susu	Produk Susu	0
Edit	K200	Sereal	Bahan Baku Sereal	0
Edit	K300	Minyak	Bahan Miaya	0
Edit	K400	Vitamin	Bahan Minyak	0
Edit	K500	Mineral	Bahan Mineral	0
Edit	K600	Lain-lain	Bahan pelangrep	0
Edit	K700	Pengemas	Bahan pengemas	0
Edit	S1	Susu Utamax	Ini bahan baku loh	0
Edit	U1	Utamax	Bahan Utama	0

Gambar 4.39 Rancangan Halaman Add Kategori

Deskripsi

- Halaman ini digunakan untuk melakukan penambahan data kategori bahan produksi.
- Tombol Add digunakan untuk menyimpan data kategori bahan baku baru.

Event

- Tambah Kategori

OnClick_Add()

{

SQL Statement :

INSERT INTO Kategori_Bahan_Baku

VALUES ([txtid], [txtNama], [txtDesc], 0)

Urutan aksi yang terjadi :

1. User memasukkan detail data kategori.
2. User menekan tombol Add.

3. Sistem akan menambahkan data kategori yang baru ke basis data.
4. Jika penambahan data kategori berhasil maka sistem akan menampilkan pesan, dan jika gagal sistem akan menampilkan pesan kesalahan.

4.40 Use Case : Edit Kategori

	KODE KATEGORI	NAMA KATEGORI	DESKRIPSI	STATUS DELETE KATEGORI
Edit	AA	ASAS	ASASA	1
Edit	K100	Susu	Produk Susu	0
Edit	K200	Sereal	Bahan Baku Sereal	0
Edit	K300	Minyak	Bahan Minyak	0
Edit	K400	Vitamin	Bahan Minyak	0
Edit	K500	Mineral	Bahan Mineral	0
Edit	K600	Lain-lain	Bahan pelengkap	0
Edit	K700	Pengemas	Bahan pengemas	0
Edit	S1	Susu Utamax	Ini bahan baku loh	0
Edit	U1	Utamax	Bahan Utama	0

Gambar 4.40 Rancangan Halaman Edit Kategori

Deskripsi

- Halaman ini digunakan untuk melakukan proses *editing* terhadap data kategori bahan baku di basis data.
- Tombol Edit digunakan untuk melakukan pengubahan data kategori.

Event

- Edit Kategori
OnClick_Edit()

```

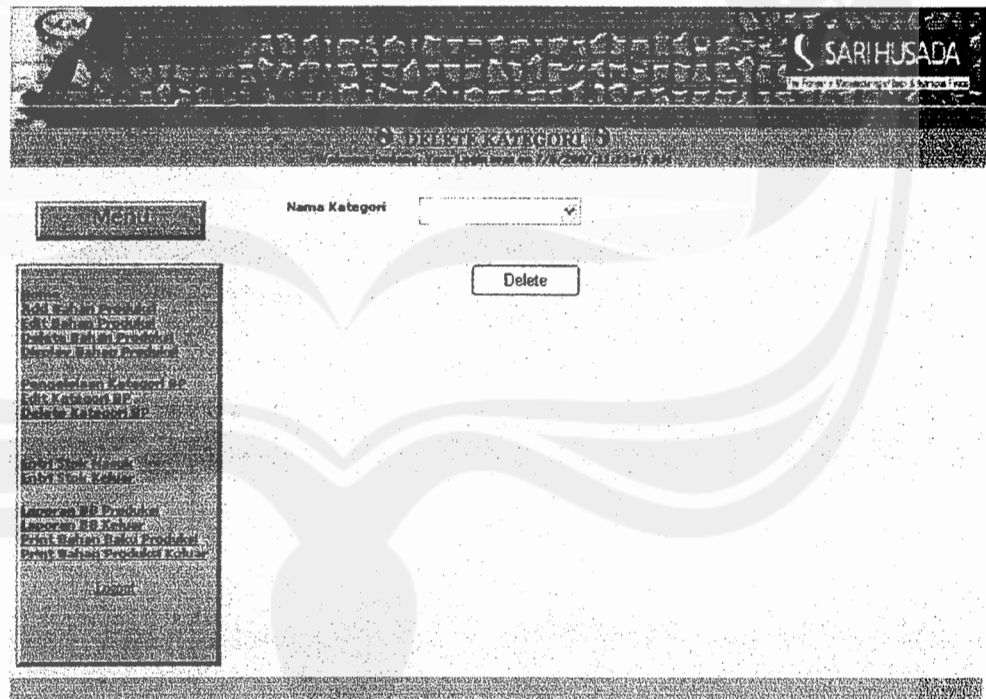
{
SQL Statement :
UPDATE Kategori_Bahan_Baku
SET Nama_Kategori = [Nama], Deskripsi = [desc],
Status_Delete_Kategori = [status]
WHERE Kode_Kategori = [idKtgr]
}

```

Urutan aksi yang terjadi :

1. User menekan tombol Edit, kemudian melakukan perubahan terhadap data tersebut.
2. Sistem akan melakukan perubahan data kategori bahan baku yang dipilih.
3. Jika pengubahan data kategori bahan baku berhasil maka datagrid akan menampilkan data hasil perubahan.

4.41 Use Case : Delete Kategori



Gambar 4.41 Rancangan Halaman Delete Kategori

Deskripsi

Program Studi Teknik Informatika	DPPL-ASCaMaDa	246/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- Halaman ini digunakan untuk menghapus data kategori bahan baku di basis data.
- Combobox digunakan untuk memilih kategori bahan baku yang akan dihapus.
- Tombol Delete digunakan untuk menghapus data kategori bahan baku.

Event

- Delete Kategori

```
OnClick_Delete()
{
    SQL Statement :
    UPDATE Kategori_Bahan_Baku
    SET Status_Delete_Kategori = 1
    WHERE Kode_Kategori = [ddlKtgr]
}
```

Urutan aksi yang terjadi :

1. User memilih data yang akan dihapus.
2. User menekan tombol Delete untuk menghapus kategori bahan baku.
3. Jika *delete* data bahan kategori baku berhasil

4.42 Use Case : Laporan Bahan Baku Produksi

ID KTOR	NAMA KATEGORI	ID BRG	NAMA BARANG	JML (KG)	JML (BOX)	RAK	HARGA
K100	Susu	B1001	Susu Segar	5000	0	R1001	1000
		B1002	Deminal	5500	0	R1002	1500
		B1003	Lactose	2000	0	R1003	1300
		B1004	Sodium	3200	0	R1004	2000
		B1005	SMP	4000	0	R1005	5200
		B1006	FCMP	1000	0	R1006	1000
K200	Sereal	B2001	Beras Merah	1200	0	R2001	1000
		B2002	Beras Putih	1000	0	R2002	2000
		B2003	Cocoa Powder	3000	0	R2003	3000
		B2004	Gula Pasir	10000	0	R2004	1500
		B2005	Kedelai	4000	0	R2005	4000
		B2006	Madu	6000	0	R2006	1200
K300	Minyak	B3001	AMF	400	0	R3001	1600

Gambar 4.42 Rancangan Halaman Laporan Bahan Baku Produksi

Deskripsi

- Halaman ini digunakan untuk menampilkan laporan bahan baku produksi.
- Tombol Export digunakan untuk mengubah report ke dalam bentuk excel atau pdf.
- Fungsi Print bisa digunakan jika report telah di export ke dalam bentuk excel atau pdf.

Event

- Tampil Report

```

OnClick_Refresh()
{
    SQL Statement :
    SELECT * FROM Bahan_Baku_Master INNER JOIN
    Kategori_Bahan_Baku ON
    Bahan_Baku_Master.Kode_Kategori =
    Kategori_Bahan_Baku.Kode_Kategori WHERE
    Bahan_Baku_Master.Status_Delete_BB = 0
}
    
```

Urutan aksi yang terjadi :

1. Sistem akan menampilkan data laporan bahan baku dalam bentuk report .

4.43 Use Case : Print Laporan Bahan Baku Produksi

Tanggal : Jul 4 2007

of 1 100% Find | Next Select a format Export

SURAT PERINTAH PRODUKSI

Friday, July 06, 2007

NO	TGL SPP	TGL AKHIR PRODUKSI	ID SPP	NAMA PRODUK	JML	STS SPP	NAMA BARANG	JML(100)
2	7/4/2007	9/4/2007	3	SGM1	1000	1	Susu Segar	2000
					1000	1	Vitamin B1	100
			4	Vitalec 1	1500	1	Susu Segar	500
					1500	1	Vitamin B1	200

Gambar 4.43 Rancangan Halaman Print Laporan Bahan Baku Produksi

Deskripsi

- Halaman ini digunakan untuk menampilkan dan mencetak laporan bahan baku produksi.
- Tombol Export digunakan untuk mengubah report ke dalam bentuk excel atau pdf.
- Fungsi Print bisa digunakan jika report telah di export ke dalam bentuk excel atau pdf.

Event

- Tampil Report

OnClick_Refresh()

{

SQL Statement :

```

SELECT * FROM Bahan_Baku_Master INNER JOIN
Kategori_Bahan_Baku ON
Bahan_Baku_Master.Kode_Kategori =
Kategori_Bahan_Baku.Kode_Kategori WHERE
Bahan_Baku_Master.Status_Delete_BB = 0
}

```

Urutan aksi yang terjadi :

1. Sistem akan menampilkan data laporan bahan baku dalam bentuk report .
2. User mengubah format report menjadi pdf dengan menekan tombol export untuk mencetak laporan lalu memilih print.

4.44 Use Case : Laporan Bahan Baku Produksi Keluar

Tanggal SPP 7/11/2007

of 1 100% Find | Next Select a format Export

LAPORAN BAHAN BAKU KELUAR

ID SPP	TGL	KATEGORI	NAMA BARANG	JML (KG)	JML (BOX)	ID KTRR	ID BRG
1	7/11/2007	Susu	Susu Segar	4500	0	K100	B1001
			Lactose	6000	0		B1003
		Sereal	Beras Merah	7600	0	K200	B2001
		Susu	Sodum	4500	0	K100	B1004

Gambar 4.44 Rancangan Halaman Laporan Bahan Baku Keluar

Deskripsi

Program Studi Teknik Informatika	DPPL-ASCaMaDa	250/ 277
----------------------------------	---------------	----------

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika

```

SELECT * FROM Bahan_Baku_Master INNER JOIN
Kategori_Bahan_Baku ON
Bahan_Baku_Master.Kode_Kategori =
Kategori_Bahan_Baku.Kode_Kategori WHERE
Bahan_Baku_Master.Status_Delete_BB = 0
)

```

Urutan aksi yang terjadi :

1. Sistem akan menampilkan data laporan bahan baku dalam bentuk report .
2. User mengubah format report menjadi pdf dengan menekan tombol export untuk mencetak laporan lalu memilih print.

4.44 Use Case : Laporan Bahan Baku Produksi Keluar

Tanggal SPP 7/11/2007

of 1 100% Find Next Select a format Export

LAPORAN BAHAN BAKU KELUAR

ID SPP	TGL	KATEGORI	NAMA BARANG	JML (KG)	JML (BOX)	ID KTGR	ID BRG
1	7/11/2007	Susu	Susu Segar	4500	0	K100	B1001
			Lactose	6000	0		B1003
		Sereal	Beras Merah	7800	0	K200	B2001
		Susu	Sodum	4500	0	K100	B1004

Gambar 4.44 Rancangan Halaman Laporan Bahan Baku Keluar

Deskripsi

Program Studi Teknik Informatika	DPPL-ASCaMaDa	250/ 277
----------------------------------	---------------	----------

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika

- Halaman ini digunakan untuk menampilkan laporan bahan baku produksi keluar.
- Tombol Export digunakan untuk mengubah report ke dalam bentuk excel atau pdf.
- Fungsi Print bisa digunakan jika report telah di export ke dalam bentuk excel atau pdf.

Event

- Tampil Report

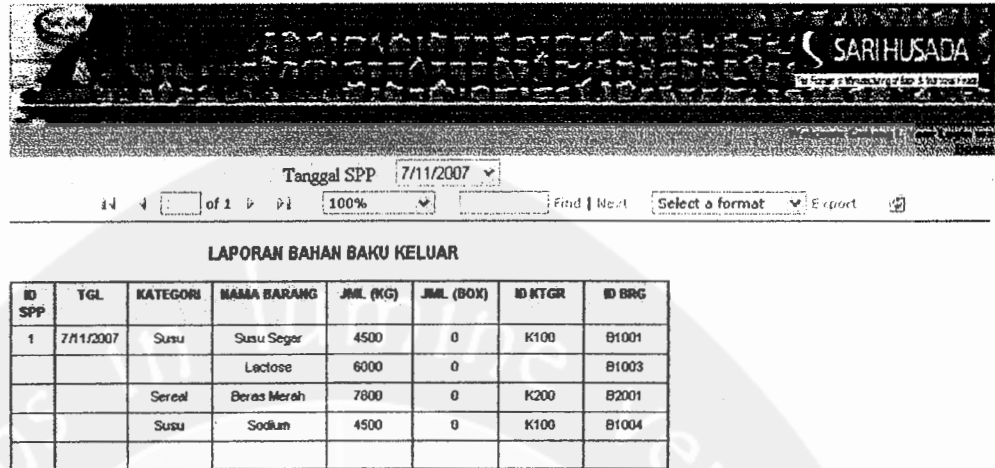
OnClick_Refresh()

```
{
  SQL Statement :
  SELECT * FROM SPP_Master INNER JOIN SPP_Detail ON
  SPP_Master.Kode_SPP_Master =
  SPP_Detail.Kode_SPP_Master INNER JOIN
  Bhn_Baku_Keluar ON SPP_Detail.Kode_SPP_Detail =
  Bhn_Baku_Keluar.Kode_SPP_Detail INNER JOIN
  Bahan_Baku_Master ON Bahan_Baku_Keluar.Kode_Barang
  = Bhn_Baku_Keluar.Kode_Barang WHERE
  SPP_Master.TGL_SPP = [ddltgl] AND
  Bhn_Baku_Keluar.Statuskeluar = 1;
}
```

Urutan aksi yang terjadi :

1. User memilih tanggal SPP
2. Sistem akan menampilkan data laporan bahan baku keluar dalam bentuk report .

4.45 Use Case : Print Laporan Bahan Baku Produksi Keluar



ID SPP	TGL	KATEGORI	NAMA BARANG	JML (KG)	JML (BOX)	ID KTR	ID BRG
1	7/11/2007	Susu	Susu Segar	4500	0	K100	B1001
			Lactose	6000	0		B1003
		Sereal	Beras Merah	7800	0	K200	B2001
		Susu	Sodaum	4500	0	K100	B1004

Gambar 4.45 Rancangan Halaman Laporan Bahan Baku Keluar

Deskripsi

- Halaman ini digunakan untuk menampilkan dan mencetak laporan bahan baku produksi keluar.
- Tombol Export digunakan untuk mengubah report ke dalam bentuk excel atau pdf.
- Fungsi Print bisa digunakan jika report telah di export ke dalam bentuk excel atau pdf.

Event

- Tampil Report

OnClick_Refresh()

{

SQL Statement :

```
SELECT * FROM SPP_Master INNER JOIN SPP_Detail ON  
SPP_Master.Kode_SPP_Master =  
SPP_Detail.Kode_SPP_Master INNER JOIN  
Bhn_Baku_Keluar ON SPP_Detail.Kode_SPP_Detail =
```

```

Bhn_Baku_Keluar.Kode_SPP_Detail INNER JOIN
Bahan_Baku_Master ON Bahan_Baku_Keluar.Kode_Barang
= Bhn_Baku_Keluar_Kode_Barang WHERE
SPP_Master.TGL_SPP = [ddltgl] AND
Bhn_Baku_Keluar.Statuskeluar = 1;
}

```

Urutan aksi yang terjadi :

1. User memilih tanggal SPP
2. Sistem akan menampilkan data laporan bahan baku keluar dalam bentuk report .
3. User mengubah format laporan dengan menekan tombol eksport menjadi pdf lalu menekan print.

4.46 Use Case : Edit Status Bahan Baku

Gambar 4.46 Rancangan Halaman Edit Status Bahan Baku

Deskripsi

- Halaman ini digunakan untuk melakukan proses *editing* terhadap status QA data bahan baku di basis data.

- Tombol Released digunakan untuk melakukan perubahan status.

Event

- Edit Status

```
OnClick_Released()  
{  
    SQL Statement :  
    UPDATE Penerimaan_Barang  
    SET QA = 1  
    WHERE Kode_Penerimaan_Brg = [ddlId]  
    UPDATE Bahan_Baku_Master  
    SET Quantityinkg = [qinkg],Quantityinbox = [qinbox]  
    WHERE Kode_Barang = [ddlbrg]  
}
```

Urutan aksi yang terjadi :

1. User memasukkan data bahan baku, jumlah dan status QA.
2. User menekan tombol Released
3. Sistem akan melakukan perubahan status data bahan baku yang dipilih.
4. Jika perubahan status data bahan baku berhasil maka sistem akan menampilkan pesan, dan jika gagal sistem akan menampilkan pesan kesalahan.

4.47 Use Case : Edit Status Barang Produksi

Menu

Nama Produk:

Kode Produk:

Jumlah:

Released Cancel

Gambar 4.47 Rancangan Halaman Edit Status Barang Produksi

Deskripsi

- Halaman ini digunakan untuk melakukan proses *editing* terhadap status QA hasil produksi di basis data.
- Tombol Released digunakan untuk melakukan perubahan status.

Event

- Edit Status

```
OnClick_Released()  
{  
    SQL Statement :  
    UPDATE Barang_Kasil_Produksi  
    SET QA = 1  
    WHERE Kode_Produksi = [ddlId]  
    UPDATE Produk  
    SET Stok_Total = [jml]  
    WHERE Kode_Produk = [ddlProduk]  
}
```

Urutan aksi yang terjadi :

1. User memasukkan data produk dan jumlah.
2. User menekan tombol Released

3. Sistem akan melakukan perubahan status produk yang dipilih.
4. Jika pengubahan status produk berhasil maka sistem akan menampilkan pesan, dan jika gagal sistem akan menampilkan pesan kesalahan.

4.48 Use Case : Add Penjualan

Gambar 4.48 Rancangan Halaman Add Penjualan

Deskripsi

- Halaman ini digunakan untuk melakukan penambahan data penjualan produk.
- Tombol Add digunakan untuk menyimpan data penjualan

Event

- Tambah Penjualan Master

OnClick_Add1 ()

{

SQL Statement :

INSERT INTO Penjualan_Master

VALUES ([lblid], [lbltgl], [lbldistri], 0)

Urutan aksi yang terjadi :

Program Studi Teknik Informatika	DPPL-ASCaMaDa	256/ 277
----------------------------------	---------------	----------

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika

1. User memasukkan detail data penjualan.
2. User menekan tombol Add.
3. Sistem akan menambahkan data penjualan master ke basis data.
4. Jika penambahan data reschedule master berhasil maka sistem akan menampilkan pesan dan user akan memasukkan data pembelian detail, dan jika gagal sistem akan menampilkan pesan kesalahan.

- Entri Penjualan Detail

```

OnClick_Add2 ()
{
    SQL Statement :
    INSERT INTO Penjualan_Detail
    VALUES ([Kode_Penj] , [noTrans] , [ddlProd] , [txtjml] , [tx
    thargaSat] , [txttotHarga])
    Update Produk set Stok_Total = Stok_Total -
    [txtjml] Where Kode_Produk = [ddlProd]
}

```

Urutan aksi yang terjadi :

1. User memasukkan data penjualan detail.
2. User menekan tombol Add
3. Sistem akan menambahkan data penjualan detail ke dalam basis data.
4. Jika penambahan data penjualan detail berhasil maka sistem akan menampilkan penjualan ke datagrid dan jika gagal sistem akan menampilkan pesan kesalahan

4.49 Use Case : Edit Penjualan

The screenshot shows a web application interface for editing sales data. At the top, there is a header with the 'SARI HUSADA' logo and a navigation menu on the left. The main content area contains a form with the following fields: 'Id Penjualan', 'Tanggal', 'Distributor', 'Produk', 'Jumlah', 'Harga Satuan', and 'Jumlah Harga'. There are 'Edit' and 'Delete' buttons. Below the form is a data table with the following columns: NO, ID PRODUK, NAMA PRODUK, BANYAKNYA, HARGA SATUAN, and JUMLAH.

Gambar 4.49 Rancangan Halaman Edit Penjualan

Deskripsi

- Halaman ini digunakan untuk melakukan proses *editing* terhadap data penjualan.
- Tombol Edit digunakan untuk melakukan perubahan data penjualan.

Event

- Edit Penjualan

OnClick_Edit()

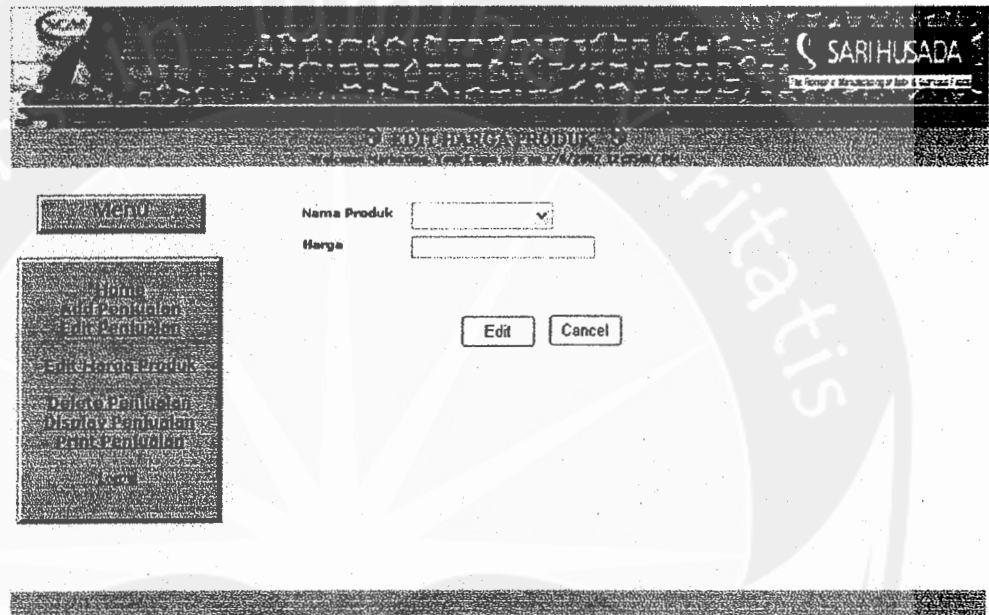
```
{
    SQL Statement :
    UPDATE Penjualan_Detail
    SET Kode_Produk = [ddlProd], Jml_Brg = [txtjml],
    Harga_Satuan = [txthargaSat], Total_Harga =
    [txttotharga] WHERE Kode_Pen_Detail = [idPnjl]
    UPDATE Produk SET Stok_Total = Stok_Total + [dgJml]
    WHERE Kode_Produk = [dgProd]
}
```

Urutan aksi yang terjadi :

1. User memilih data yang mau diubah dengan menekan tombol select
2. User memasukkan data yang baru.

3. User menekan tombol Edit untuk mengubah data penjualan.
4. Sistem akan melakukan perubahan data penjualan.
5. Jika pengubahan data penjualan berhasil maka datagrid akan berubah.

4.50 Use Case : Edit Harga Produk



Gambar 4.50 Rancangan Halaman Edit Harga Produk

Deskripsi

- Halaman ini digunakan untuk melakukan proses *editing* terhadap harga produk di basis data.
- Tombol Edit digunakan untuk melakukan pengubahan data harga produk.

Event

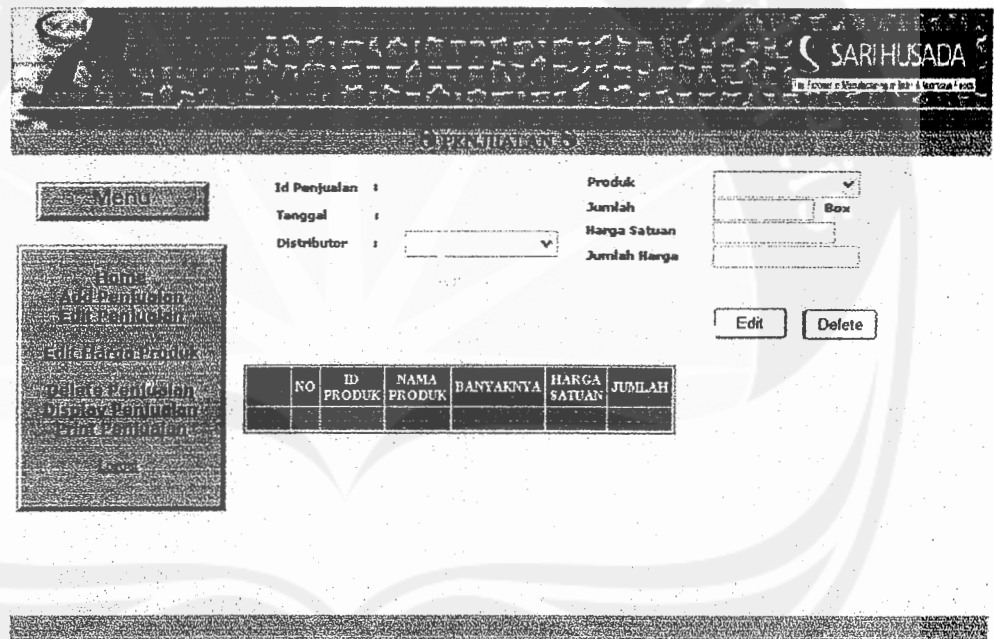
- Edit Harga

```
OnClick_Edit()
{
    SQL Statement :
    UPDATE Produk
    SET Harga = [txtHarga]
    WHERE Kode_Produk = [ddlProduk]
```

}
Urutan aksi yang terjadi :

1. User memasukkan data harga produk.
2. User menekan tombol Edit.
3. Sistem akan melakukan perubahan data harga produk.
4. Jika pengubahan harga produk maka sistem akan menampilkan pesan, dan jika gagal sistem akan menampilkan pesan kesalahan.

4.51 Use Case : Delete Penjualan



Gambar 4.5 Rancangan Halaman Editing User

Deskripsi

- Halaman ini digunakan untuk menghapus data penjualan.
- Datagrid digunakan untuk memilih data yang akan dihapus.
- Tombol Delete digunakan untuk menghapus data penjualan.

Event

- Delete Penjualan
onClick_Delete()

```

{
SQL Statement :
DELETE Penjualan_Detail Where Kode_Penj_Detail =
[dgidpenj]
UPDATE Produk SET Stok_Total = Stok_Total + [dgJml]
WHERE Kode_Produk = [dgProd]
}

```

Urutan aksi yang terjadi :

1. User memilih data yang akan dihapus.
2. User menekan tombol Delete untuk menghapus penjualan
3. Jika *delete* data penjualan berhasil, maka item pada datagrid akan berkurang.

4.52 Use Case : Add Supplier

Gambar 4.52 Rancangan Halaman Add Supplier

Deskripsi

- Halaman ini digunakan untuk melakukan penambahan data supplier.
- Tombol Add digunakan untuk menyimpan data supplier

Event

Program Studi Teknik Informatika	DPPL-ASCaMaDa	261/ 277
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- **Tambah Supplier**

OnClick_Add()

{

SQL Statement :

INSERT INTO Data_Supplier

VALUES ([txtid],[txtidbrg],[txtNama],[txtkota],[txtn
amaperusahaan],[txtemail],[txtalamat],[txttelepon],
[txtfax],[txtkodepos],[txtnegara],[txtnorek],[txtba
nk],[cabang],0)

Urutan aksi yang terjadi :

1. User memasukkan detail data supplier.
2. User menekan tombol Add.
3. Sistem akan menambahkan data supplier yang baru ke basis data.
4. Jika penambahan data supplier berhasil maka sistem akan menampilkan pesan, dan jika gagal sistem akan menampilkan pesan kesalahan.

4.53 Use Case : Edit Supplier

The screenshot shows a web application interface for 'SARIHUSADA'. At the top, there is a navigation bar with the company name and logo. Below it, there is a header for 'DATA KATEGORI BAHAN BAKU'. The main content area features a table with the following data:

	KODE KATEGORI	NAMA KATEGORI	DESKRIPSI	STATUS DELETE_KATEGORI
Edit	AA	ASAS	ASASA	1
Edit	K100	Susu	Produk Susu	0
Edit	K200	Sereal	Bahan Baku Sereal	0
Edit	K300	Minyak	Bahan Minyak	0
Edit	K400	Vitamin	Bahan Minyak	0
Edit	K500	Mineral	Bahan Mineral	0
Edit	K600	Lain-lain	Bahan pelengkap	0
Edit	K700	Pengemas	Bahan pengemas	0
Edit	S1	Susu Utamax	Ini bahan baku loh	0
Edit	U1	Utamax	Bahan Utama	0

On the left side, there is a 'MENU' sidebar with various options like 'Home', 'Add Bahan Produk', 'Edit Bahan Produk', etc. At the top right, there are input fields for 'ID Kategori', 'Nama Kategori', and 'Deskripsi', along with 'Add' and 'Cancel' buttons.

Gambar 4.53 Rancangan Halaman Pengelolaan Kategori

Deskripsi

- Halaman ini digunakan untuk melakukan proses *editing* terhadap data kategori bahan baku di basis data.
- Tombol Edit digunakan untuk melakukan perubahan data kategori.

Event

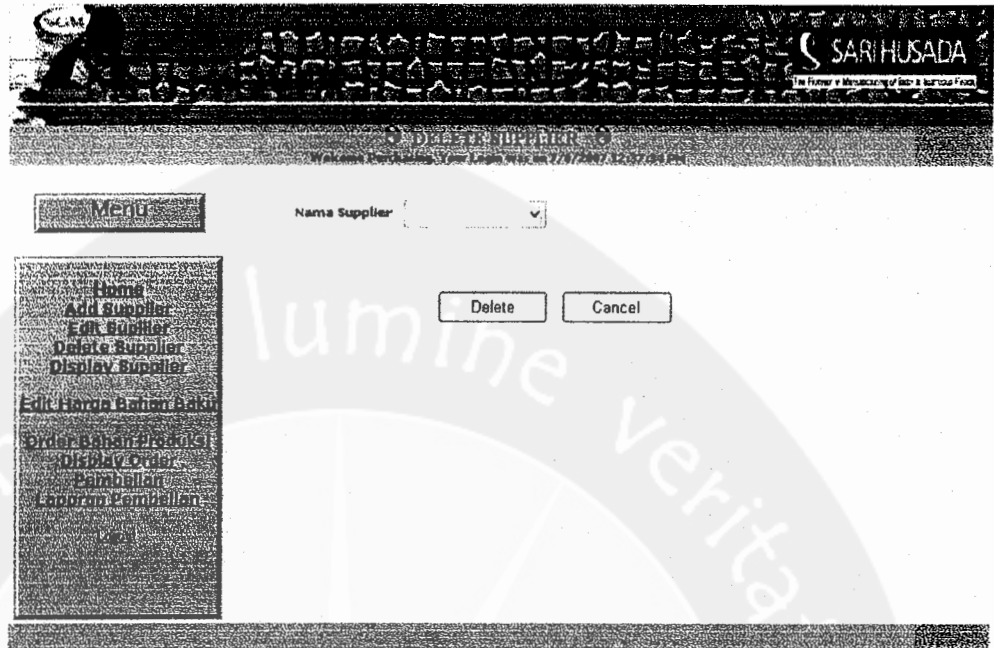
- Edit Kategori

```
OnClick_Edit()  
{  
    SQL Statement :  
    UPDATE Kategori_Bahan_Baku  
    SET Nama_Kategori = [Nama], Deskripsi = [desc],  
    Status_Delete_Kategori = [status]  
    WHERE Kode_Kategori = [idKtgr]  
}
```

Urutan aksi yang terjadi :

4. User menekan tombol Edit, kemudian melakukan perubahan terhadap data tersebut.
5. Sistem akan melakukan perubahan data kategori bahan baku yang dipilih.
6. Jika perubahan data kategori bahan baku berhasil maka datagrid akan menampilkan data hasil perubahan.

4.54 Use Case : Delete Supplier



Gambar 4.54 Rancangan Halaman Delete Supplier

Deskripsi

- Halaman ini digunakan untuk menghapus data supplier di basis data.
- Combobox digunakan untuk memilih supplier yang akan dihapus.
- Tombol Delete digunakan untuk menghapus data supplier

Event

- Delete Supplier

```
OnClick_Delete()  
{  
    SQL Statement :  
    UPDATE Data_Supplier  
    SET Status_Delete_SPP = 1  
    WHERE Kode_Supplier = [ddlidSupp]  
}
```

Urutan aksi yang terjadi :

1. User memilih data supplier yang akan dihapus.
2. User menekan tombol Delete untuk menghapus supplier

3. Jika *delete* data supplier berhasil maka sistem akan menampilkan pesan, dan jika gagal sistem akan menampilkan pesan kesalahan.

4.55 Use Case : Display Supplier

ID	NAMA	ID BUKU-CG	ALAMAT	EMAIL	STATUS PERUSAHAAN	NOTA	NEGERI	KOD POS	FAK	TELEPON	CABANG	SIKIL	NO REKENING
5001	KUD Jember Makmur	E1001	Cangkubing Suman	jaranamakmur@yahoo.com	KUD Larian Makmur	Jember	Indonesia	626511	0274565246	0274592455	Jember	BNI	1254094514744
5002	KUD Wajaya Mulya	E1001	Tan. Suman	wajayamulya@yahoo.com	KUD Wajaya Mulya	Jember	Indonesia	622511	0274145267	0274255134	Jember	BNI	11254928541254
5003	KUM Jember	E1001	Jember, Jember	jarum11@yahoo.com	KUM Jember	Jember	Indonesia	641208	0271254234	0271254234	Jember	BNI	1025555145678
5004	KUD Lippo	E1001	Cepog Kiyok	lippo@yahoo.com	KUD Lippo	Jember	Indonesia	651208	0271254234	0271254234	Cepog	BNI	1025555145678
5005	PT. Wenasung	E1005	Balaru	wenasung@yahoo.com	PT. Wenasung	Balaru	Indonesia	560120	0272445271	0272445271	Balaru	BCA	2025455145678
5006	Gondang Baru	E2004	Gondang, Klaten	Gondang@yahoo.com	PT. Gondang Baru	Klaten	Indonesia	622450	0271254234	0271254234	Klaten	BNI	1025555145678
5007	PT. Demasal Seroja	E1002	Lamongan 6 Buar Jember	demasal@yahoo.com	PT. Demasal Seroja	Buar	Indonesia	265321	027222651	027222651	Buar	BCA	20251225211232
5008	PT. FCHP Seroja	E1008	Lamongan 6 Buar Jember	demasal@yahoo.com	PT. FCHP Seroja	Buar	Indonesia	265321	027222651	027222651	Buar	BCA	20251225211232
5009	PT. Gagah Perkasa	E2001	Jl. Pajay 12 Kudus	gagah@yahoo.com	PT. Gagah Perkasa	Kudus	Indonesia	707514	0271254234	0271254234	Kudus	BNI	5152545514567

Gambar 4.45 Rancangan Halaman Display Supplier

Deskripsi

- Halaman ini digunakan untuk menampilkan data supplier di basis data.

Event

- Halaman di *load*

Page_Load ()

```
{
    SQL Statement :
    SELECT * FROM Supplier
    WHERE Status_Delete_Supplier = 0
}
```

Urutan aksi yang terjadi :

1. Sistem akan menampilkan data bahan baku yang terdapat di basisdata.

8. Jika pengubahan data kategori bahan baku berhasil maka sistem akan menampilkan pesan, dan jika gagal sistem akan menampilkan pesan kesalahan.

4.57 Use Case : Order Bahan Baku

Gambar 4.57 Rancangan Halaman Order Bahan Produksi

Deskripsi

- Halaman ini digunakan untuk membuat pembelian bahan baku produksi
- Tombol Add digunakan untuk menambahkan data pembelian ke basis data.

Event

- Entri PO Master

```

OnClick_Add1()
{
    SQL Statement :
    INSERT INTO Pembelian Master
    VALUES ([txtid], [txttgl], 0)
}

```

Urutan aksi yang terjadi :

1. User memasukkan detil data permintaan pembelian master.
2. User menekan tombol Add.
3. Sistem akan menambahkan data pembelian master ke dalam basis data.
4. Jika penambahan data pembelian master berhasil maka sistem akan menampilkan pesan dan user akan memasukkan data pembelian detail, dan jika gagal sistem akan menampilkan pesan kesalahan.

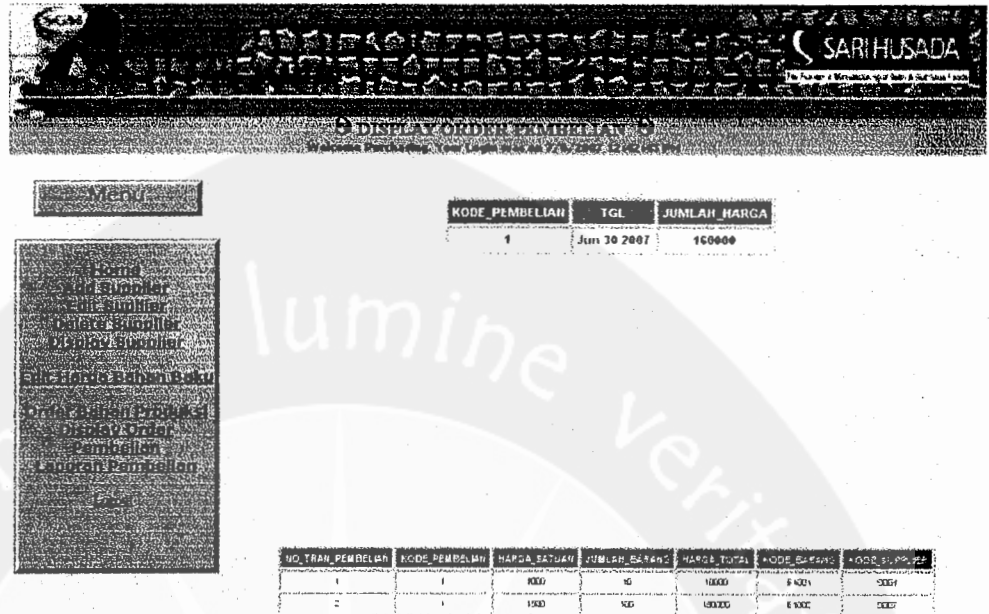
- Entri PO Detail

```
OnClick_Add2()  
{  
    SQL Statement :  
    INSERT INTO Pembelian_Detail  
    VALUES ([idDet], [id], [txtjml], [hargasat], [hargatot],  
    [ddlSupp], [ddlbrg])  
    UPDATE Pembelian_Master  
    SET Jumlah_Harga = [lblSumHarga]  
}
```

Urutan aksi yang terjadi :

1. User memasukkan detil data pembelian detail.
2. User menekan tombol Add
3. Sistem akan menambahkan data pembelian detail ke dalam basis data.
4. Jika penambahan data pembelian detail berhasil maka sistem akan menampilkan pesan dan jika gagal sistem akan menampilkan pesan kesalahan.

4.58 Use Case : Display Order Pembelian



Gambar 4.58 Rancangan Halaman Display Order Pembelian

Deskripsi

- Halaman ini digunakan untuk menampilkan data pembelian.

Event

- Halaman di *load*

Page_Load ()

```

{
    SQL Statement :
    SELECT * FROM Pembelian_Master INNER JOIN
    Pembelian_Detail ON
    Pembelian_Master.Kode_Pembelian =
    Pembelian_Detail.Kode_Pembelian
}
    
```

Urutan aksi yang terjadi :

1. Sistem akan menampilkan data pembelian dalam data grid .

4.59 Use Case : Laporan Pembelian

ID	TGL	NAMA KATEGORI	KODE BARANG	NAMA BARANG	NAMA SUPPLIER	JUMLAH BARANG	HARGA SATUAN	HARGA TOTAL
1	6/30/2007	Susu	B1001	Susu Segar	KUD Serana Makmur	10	1,000.00	10,000.00
			B1002	Deminat	PT Deminat Sentosa	100	1,500.00	150,000.00
								168,000.00

Gambar 4.59 Rancangan Halaman Laporan Pembelian

Deskripsi

- Halaman ini digunakan untuk menampilkan laporan pembelian
- Combobox digunakan untuk memilih tanggal pembelian.
- Tombol Export digunakan untuk mengubah report ke dalam bentuk excel atau pdf.
- Fungsi Print bisa digunakan jika report telah di export ke dalam bentuk excel atau pdf.

Event

- Tampil Pembelian

OnClick_Refresh()

```
{
    SQL Statement :
    SELECT * FROM Pembelian_Master INNER JOIN
    Pembelian_Detail ON
    Pembelian_Master.Kode_Pembelian =
    Pembelian_Detail.Kode_Pembelian
    INNER JOIN Bahan_Baku_Master ON
    Pembelian_Detail.Kode_Barang =
```

```

Bahan_Master.Kode_Barang =
Bhn_Baku_Keluar_Kode_Barang WHERE
Pembelian_Master.Tanggal_Pembelian = [ddltgl]
}

```

Urutan aksi yang terjadi :

4. Sistem akan menampilkan laporan pembelian terbaru dalam bentuk report .
5. User dapat memilih tanggal pembelian dengan memilih tanggal pada combobox.

4.60 Use Case : Print Laporan Pembelian

ID	TGL	NAMA KATEGORI	KODE BARANG	NAMA BARANG	NAMA SUPPLIER	JUMLAH BARANG	HARGA SATUAN	HARGA TOTAL
1	6/30/2007	Susu	B1001	Susu Segar	KUD Sarena Makmur	10	1,000.00	10,000.00
			B1002	Deminal	PT. Deminal Sentosa	100	1,500.00	150,000.00
								160,000.00

Gambar 4.60 Rancangan Halaman Print Laporan Pembelian

Deskripsi

- Halaman ini digunakan untuk menampilkan dan mencetak laporan pembelian
- Combobox digunakan untuk memilih tanggal pembelian.
- Tombol Export digunakan untuk mengubah report ke dalam bentuk excel atau pdf.

Program Studi Teknik Informatika	DPPL-ASCaMaDa	271/ 277
----------------------------------	---------------	----------

Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika

- Fungsi Print bisa digunakan jika report telah di export ke dalam bentuk excel atau pdf.

Event

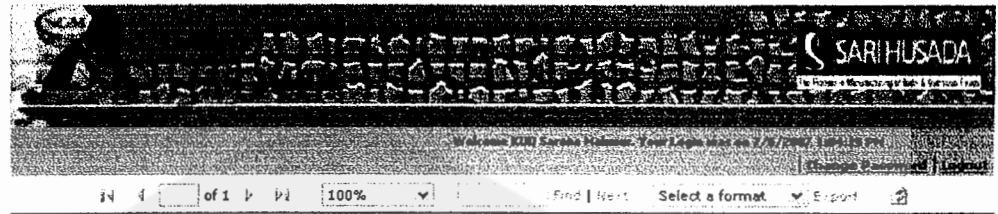
- Tampil Pembelian

```
OnClick_Refresh()  
{  
    SQL Statement :  
    SELECT * FROM Pembelian_Master INNER JOIN  
    Pembelian_Detail ON  
    Pembelian_Master.Kode_Pembelian =  
    Pembelian_Detail.Kode_Pembelian  
    INNER JOIN Bahan_Baku_Master ON  
    Pembelian_Detail.Kode_Barang =  
    Bahan_Master.Kode_Barang =  
    Bhn_Baku_Keluar_Kode_Barang WHERE  
    Pembelian_Master.Tanggal_Pembelian = [ddltgl]  
}
```

Urutan aksi yang terjadi :

1. Sistem akan menampilkan laporan pembelian terbaru dalam bentuk report .
2. User dapat memilih tanggal pembelian dengan memilih tanggal pada combobox.
3. User menekan tombol eksport untuk mengubah format laporan menjadi pdf kemudian menekan print untuk mencetak.

4.61 Use Case : Display Purchase Order



Gambar 4.61 Rancangan Halaman Display Purchase Order

Deskripsi

- Halaman ini digunakan untuk menampilkan purchase order atau pembelian bahan baku dalam bentuk report.

Event

- Tampil Pembelian

OnClick_Refresh()

{

SQL Statement :

```
SELECT * FROM Pembelian_Master INNER JOIN
```

```
Pembelian_Detail ON
```

```
Pembelian_Master.Kode_Pembelian =
```

```
Pembelian_Detail.Kode_Pembelian
```

```
INNER JOIN Bahan_Baku_Master ON
```

```
Pembelian_Detail.Kode_Barang =
```

```
Bahan_Master.Kode_Barang =
```

```
Bhn_Baku_Keluar_Kode_Barang WHERE Pembelian_Detail
```

```
= session[idSupp]
```

}

Urutan aksi yang terjadi :

1. Sistem akan menampilkan laporan pembelian terbaru dalam bentuk report .

4.62 Use Case : Print Purchase Order



PURCHASE ORDER

Friday, July 06, 2007 Kode: S001

NO	TANGGAL	ID BARANG	NAMA BARANG	JUMLAH	HARGA	TOTAL
1	6/30/2007	B1001	Susu Seger	10	1000	10000

Gambar 4.62 Rancangan Halaman Print Purchase Order

Deskripsi

- Halaman ini digunakan untuk menampilkan purchase order atau pembelian bahan baku.
- Tombol Export digunakan untuk mengubah report ke dalam bentuk excel atau pdf.
- Fungsi Print bisa digunakan jika report telah di export ke dalam bentuk excel atau pdf.

Event

- Tampil Pembelian

`OnClick_Refresh()`

{

 SQL Statement :

 SELECT * FROM Pembelian_Master INNER JOIN
 Pembelian_Detail ON

```

Pembelian_Master.Kode_Pembelian =
Pembelian_Detail.Kode_Pembelian
INNER JOIN Bahan_Baku_Master ON
Pembelian_Detail.Kode_Barang =
Bahan_Master.Kode_Barang =
Bhn_Baku_Keluar_Kode_Barang WHERE Pembelian_Detail
= session[idSupp]
}

```

Urutan aksi yang terjadi :

1. Sistem akan menampilkan laporan pembelian terbaru dalam bentuk report .
2. User menekan tombol ekspor untuk mengubah report ke format pdf lalu memilih fungsi print.

4.63 Use Case : Entri Pesanan

Gambar 4.63 Rancangan Halaman Entri Pesanan

Deskripsi

- Halaman ini digunakan untuk membuat produk.

- Tombol Add digunakan untuk menambahkan data pesanan produk ke basis data.

Event

- Entri Pesanan Master

```
OnClick_Add1()
{
    SQL Statement :
    INSERT INTO Data_Pesanan_Master
    VALUES ([txtid], session[distri], [txttgl], [txttglambi
    1])
}
```

Urutan aksi yang terjadi :

1. User memasukkan detail data pesanan master.
2. User menekan tombol Add.
3. Sistem akan menambahkan data pesanan master ke dalam basis data.
4. Jika penambahan data pesanan master berhasil maka sistem akan menampilkan pesan dan user akan memasukkan data pesanan detail, dan jika gagal sistem akan menampilkan pesan kesalahan.

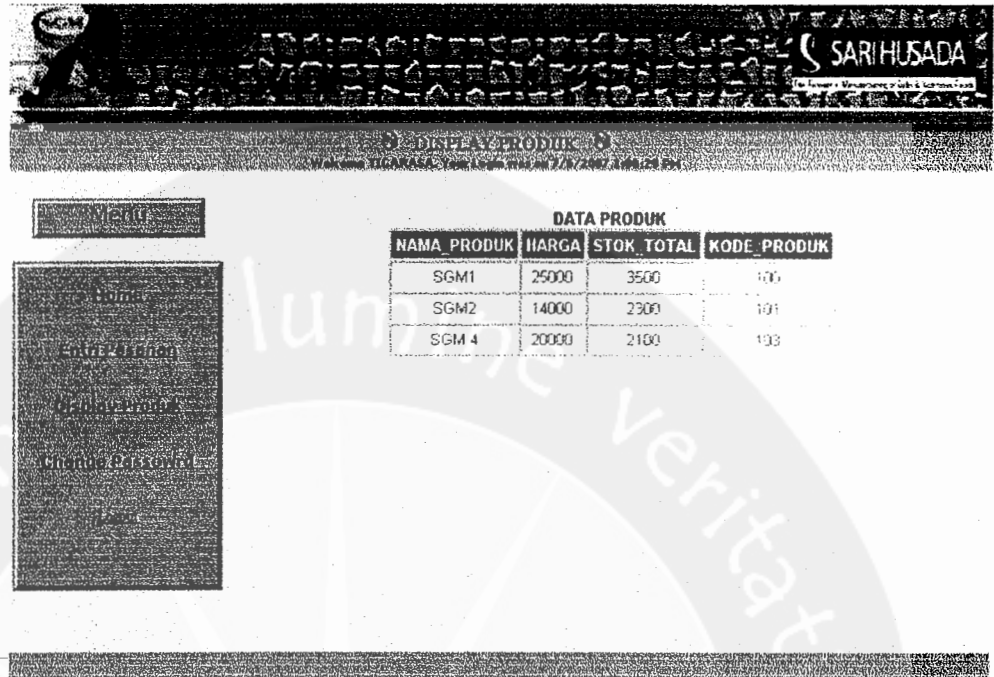
- Entri Pesanan Detail

```
OnClick_Add2()
{
    SQL Statement :
    INSERT INTO Pesanan_Detail
    VALUES ([idDet], [txtid], [ddlProduk], [txtjml])
}
```

Urutan aksi yang terjadi :

1. User memasukkan detail data pesanan.
2. User menekan tombol Add
3. Sistem akan menambahkan data pesanan detail ke dalam basis data.
4. Jika penambahan data pesanan detail berhasil maka sistem akan menampilkan pesan dan jika gagal sistem akan menampilkan pesan kesalahan.

4.64 Use Case : Display Produk



Gambar 4.64 Rancangan Halaman Display Produk

Deskripsi

- Halaman ini digunakan untuk menampilkan data produk.

Event

- Halaman di *load*

```
Page_Load()  
{  
    SQL Statement :  
    SELECT Kode_Produk ,Nama_Produk ,Harga ,Stok_Total  
    FROM Produk  
}
```

Urutan aksi yang terjadi :

1. Sistem akan menampilkan data pesanan dalam data grid .