

**RANCANG BANGUN SISTEM INFORMASI  
LAPORAN KUALITAS *PLATTING* BERBASIS WEB  
MENGUNAKAN PHP 5.4 DAN MYSQL 5.5  
PADA PT SURYA SUNJAYA INDUSTRI**

**TUGAS AKHIR**

Untuk Memenuhi Sebagian Syarat Penyelesaian Jenjang Diploma Empat (D4)  
Program Studi Sistem Informasi Pada Politeknik STMI Jakarta

**OLEH**

**FAHRU ROZI**

**1309040**



**POLITEKNIK STMI JAKARTA  
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA  
JAKARTA  
2016**

**POLITEKNIK STMI JAKARTA**  
**KEMENTERIAN PERINDUSTRIAN R.I.**

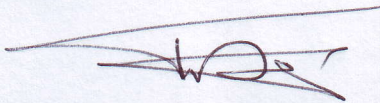
**LEMBAR PENGESAHAN**

Telah Diuji Oleh Tim Sidang Tugas Akhir Program Studi Sistem Informasi  
Politeknik STMI Jakarta Kementerian Perindustrian R.I.  
Pada Hari Senin 10 Oktober 2016

Menyetujui

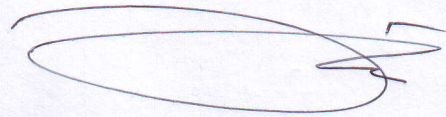
Jakarta, Oktober 2016

Dosen Pembimbing



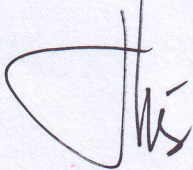
Dedy Trisanto, SKom, MMSI  
NIP : 197805052005021002

Dosen Penguji



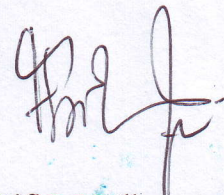
Drs. Jacob Saragih, MM  
NIP : 19404281986031002

Dosen Penguji



Ulil Hamida, ST, MT  
NIP : 198103272005022001

Dosen Penguji



Fifi L Hadianastuti, S.Kom, M. Kes  
NIP : 197310162005022001

**POLITEKNIK STMI JAKARTA**  
**KEMENTERIAN PERINDUSTRIAN R.I.**

**TANDA PERSETUJUAN DOSEN PEMBIMBING**

Judul Tugas Akhir : **Rancang Bangun Aplikasi Laporan Kualitas  
*Platting* Berbasis Web Menggunakan PHP 5.4  
Dan MYSQL 5.5 Pada PT Surya Sunjaya  
Industri**

Disusun Oleh :

Nama . : Fahru Rozi

NIM : 1309040

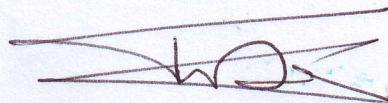
Program Studi : Sistem Informasi

Tanggal Seminar : 8 September 2016

Tanggal Sidang : 10 Oktober 2016

Tanggal Lulus : 10 Oktober 2016

Menyetujui,  
Jakarta, Oktober 2016  
Dosen Pembimbing



Dedy Trisanto, SKom, MMSI  
NIP : 197805052005021002

**POLITEKNIK STMI JAKARTA**  
**KEMENTERIAN PERINDUSTRIAN R.I.**

**TANDA PERSETUJUAN ASISTEN DOSEN PEMBIMBING**

Judul Tugas Akhir : **Rancang Bangun Aplikasi Laporan Kualitas  
*Platting* Berbasis Web Menggunakan PHP 5.4  
Dan MYSQL 5.5 Pada PT Surya Sunjaya  
Industri**

Disusun Oleh :

Nama : Fahru Rozi

NIM : 1309040

Program Studi : Sistem Informasi

Tanggal Seminar : 8 September 2016

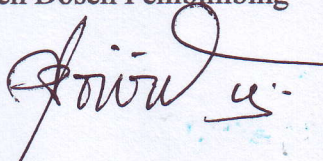
Tanggal Sidang : 10 Oktober 2016

Tanggal Lulus : 10 Oktober 2016

Menyetujui,

Jakarta, Oktober 2016

Asisten Dosen Pembimbing













Noveriza Yuliasari, SSI, MT

NIP: 197811212009012003

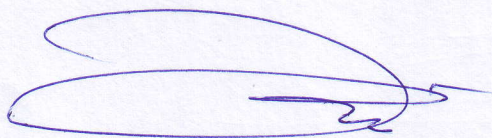


**LEMBAR BIMBINGAN TUGAS AKHIR**

Nama : Fahru Rozi  
NIM : 1309040  
Judul TA : Rancang Bangun Sistem Informasi Laporan Kualitas *Plating* Berbasis Web Menggunakan PHP 5.4 Dan MySQL 5.5 Pada PT Surya Sunjaya Industri  
Pembimbing : Dedy Trisanto, S.Kom, MMSI

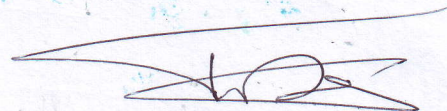
Tanggal	Keterangan	Paraf
03-09-2014	Konsultasi BAB I - BAB III	
05-09-2014	Revisi BAB I	
08-09-2014	Revisi BAB II dan BAB III	
10-09-2014	Pengajuan dan pembahasan BAB IV	
15-09-2014	Revisi BAB IV	
02-03-2016	Pengajuan dan pembahasan BAB V	
15-03-2016	Revisi BAB V	
24-03-2016	Pengajuan BAB IV dan presentasi program	
08-06-2016	Revisi program	
30-08-2016	Review Full BAB, program, dan ACC	

Mengetahui,  
Ketua Prodi  
Sistem Informasi Industri Otomotif



(Drs. Jacob Saragih, MM)  
NIP: 195404281986031002

Dosen Pembimbing



(Dedy Trisanto, S.Kom, MMSI)  
NIP. 197805052005021002















## LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Fahru Rozi

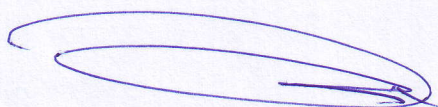
NIM : 1309040

Judul TA : Rancang Bangun Sistem Informasi Laporan Kualitas *Plating* Berbasis Web Menggunakan PHP 5.4 Dan MySQL 5.5 Pada PT Surya Sunjaya Industri

Pembimbing : Noveriza Yuliasari, SSi, MT

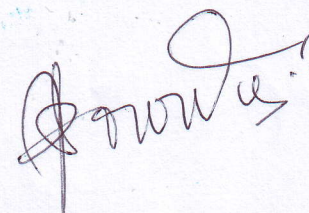
Tanggal	Keterangan	Paraf
03-09-2014	Diskusi awal Tugas Akhir	
05-09-2014	Konsultasi BAB I - BAB III	
11-09-2014	Revisi BAB I – BAB III	
19-09-2014	Pengajuan dan pembahasan BAB IV	
24-09-2014	Revisi BAB IV	
31-10-2014	Pengajuan dan pembahasan BAB V	
11-11-2014	Pembahasan BAB V	
13-01-2015	Revisi BAB V	
27-5-2015	Revisi BAB V	
06-04-2016	Revisi BAB V	
23-08-2016	Revisi BAB V dan Pembahasan BAB VI	
24-8-2016	Presentasi dan demo program	
30-08-2016	Review Full BAB, program, dan ACC	

Mengetahui,  
Ketua Prodi  
Sistem Informasi Industri Otomotif



(Drs. Jacob Saragih, MM)  
NIP: 195404281986031002

Asisten Dosen Pembimbing



(Noveriza Yuliasari, SSi, MT)  
NIP: 197 811212009012003



## LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan dibawah ini:

Nama : Fahu Rozi

NIM : 1309040

Berstatus sebagai mahasiswa jurusan program studi Sistem Informasi Industri Otomotif pada Politeknik STMI Jakarta Kementrian Perindustrian R.I. Dengan ini menyatakan bahwa hasil karya Tugas Akhir yang saya buat dengan judul :

“RANCANG BANGUN SISTEM INFORMASI LAPORAN KUALITAS PLATTING BERBASIS WEB MENGGUNAKAN PHP 5.4 DAN MYSQL 5.5 PADA PT SURYA SUNJAYA INDUSTRI”.

**Dibuat** dan diselesaikan sendiri, dengan menggunakan literatur hasil kuliah, survey lapangan, dibantu oleh dosen pembimbing maupun asisten dosen pembimbing, serta buku – buku maupun jurnal – jurnal ilmiah yang menjadi bahan acuan yang tertera dalam referensi karya Tugas Akhir ini.

**Bukan** merupakan hasil duplikasi karya tulis yang sudah dipublikasikan atau yang pernah dipakai sebelumnya untuk mendapatkan gelar sarjana di Universitas atau Perguruan Tinggi lain, kecuali yang telah disebutkan sumbernya dan dicantumkan pada referensi karya Tugas Akhir ini.

**Bukan** merupakan karya tulis hasil terjemahan dari kumpulan buku atau jurnal acuan yang tertera dalam referensi karya Tugas Akhir ini.

Jika terbukti saya tidak memenuhi apa yang telah saya nyatakan di atas, maka saya bersedia menerima sanksi atas apa yang telah saya lakukan sesuai dengan peraturan yang berlaku.

Jakarta, 30 Agustus 2016  
Yang menyatakan,



Fahu Rozi  
NIM. 1309040

## ABSTRAK

PT Surya Sunjaya Industri adalah perusahaan yang bergerak di bidang pembuatan *component manufacturing*, yang memegang komitmen secara profesional guna kepuasan pelanggan, dan menggunakan *standard management* sesuai dengan permintaan *customer* serta memiliki konsep **Kaizen** (*Continuous Improvement / perbaikan secara terus menerus*). PT Surya Sunjaya Industri membutuhkan sebuah sistem untuk mendata laporan kualitas *plating*. Sistem informasi laporan kualitas *plating* pada PT Surya Sunjaya Industri saat ini sebagian masih dilakukan secara manual yaitu menggunakan kertas untuk mencatat laporan kualitas *plating*, yang selanjutnya diketik kembali ke dalam *Microsoft Excel* dan disimpan di dalam komputer. Sistem yang ada pada perusahaan belum terintegrasi sehingga dalam penyampaian informasi mengenai data kualitas *plating* yang disajikan tidak langsung *ter-update*. Rancang bangun sistem informasi laporan kualitas *plating* berbasis web ini merupakan suatu rancang bangun yang sangat diperlukan perusahaan dalam melakukan pelaporan kualitas *plating* yang bersifat *real time*. Proses akan menjadi lebih mudah dan informasi menjadi lebih *update*, sehingga membantu bagian-bagian di dalam sistem yang terkait untuk saling bertukar informasi dan mengambil keputusan dengan cepat. Perancangan sistem informasi laporan kualitas *plating* ini menggunakan pengembangan sistem *evolutionary prototype*. Pemodelan sistem menggunakan *flowchart*, *Unified Modeling Language*(UML), HIPO, dan pembuatan aplikasi sistem laporan kualitas *plating* menggunakan *Hypertext PreProcessor*(PHP) dan serta perangkat lunak basis data menggunakan *My Structure Query Language*(SQL). Sistem informasi laporan kualitas *plating* yang dirancang dan dibangun untuk dapat memudahkan perusahaan dalam mengelola dan melihat informasi laporan kualitas *plating* dan dapat memudahkan dalam pengambilan keputusan jika diperlukan. Aplikasi ini diharapkan dapat membantu karyawan bagian *quality control* dalam pembuatan laporan kualitas *plating*.

**Kata kunci** : sistem informasi, laporan kualitas *plating*, *real time*, *Hypertext PreProcessor*(PHP), *My Structure Query Language*(SQL), *Unified Modeling Language*(UML)

## KATA PENGANTAR

Puji Syukur Kehadirat Allah, yang telah melimpahkan nikmat-Nya, terutama nikmat iman, sehat sehingga dapat menyelesaikan Tugas Akhir ini dengan baik dan lancar.

Tugas Akhir ini disusun untuk memenuhi persyaratan akademis yang telah ditetapkan oleh Politeknik STMI Jakarta dengan bobot 4 SKS. Penulis mengambil pokok bahasan untuk laporan dengan judul "**Rancang Bangun Sistem Informasi Laporan Kualitas *Plating* Berbasis Web Menggunakan PHP 5.4 Dan MySQL 5.5 Pada PT Surya Sunjaya Industri**". Penulisan ini juga merupakan penerapan atau latihan untuk mengaplikasikan teori-teori yang pernah penulis dapatkan pada mata kuliah yang dipelajari dibangku kuliah dalam kenyataan di lapangan.

Dalam penulisan laporan ini, penulis banyak mendapatkan bantuan, bimbingan, petunjuk dan dukungan dari berbagai pihak. Dengan ketulusan dan kerendahan hati, perkenankanlah penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Untuk kedua orang tua ku dan kakak-kakak ku yang selalu mendoakan dan memberikan semangat maupun dukungan dalam penyusunan tugas akhir .
2. Bapak Dr. Mustofa, ST, MT selaku Direktur Politeknik STMI Jakarta.
3. Bapak Drs. Jacob Saragih, MM. selaku Ketua Program Studi Sistem Informasi di Politeknik STMI Jakarta yang telah bersedia memberikan pengarahan kepada penulis untuk menyelesaikan Tugas Akhir ini.
4. Ibu Noveriza Yuliasari, SSi, MT selaku sekretaris program studi sistem informasi di Politeknik STMI, dan selaku Asisten Dosen Pembimbing yang telah memberikan arahan, bimbingan serta penjelasan dalam penyusunan Tugas Akhir.
5. Bapak Dedy Trisanto, S.Kom, MMSI selaku Dosen Pembimbing yang telah memberikan arahan, bimbingan serta penjelasan dalam penyusunan Tugas Akhir.

6. Bapak Gunari Yanto, serta seluruh pegawai PT Surya Sunjaya Industri yang telah memberi arahan serta informasi untuk pengumpulan data.
7. Seluruh jajaran dosen dan staf karyawan Politeknik STMI Jakarta yang telah membantu dalam kelancaran pembuatan penulisan ini.
8. Rekan-rekan mahasiswa Politeknik STMI Jakarta Program Studi Sistem Informasi terutama rekan-rekan SA02 2009 atas kebersamaan dan motivasinya selama ini.

Dengan segala kemampuan dan keterbatasan, penulis menyadari segala kekurangan yang dalam penulisan, karena itu penulis sangat mengharapkan segala kritik atau saran yang dapat membangun dari semua pihak. Dan juga berharap penulisan ini dapat berguna bagi diri pribadi maupun pihak-pihak lain yang membacanya.

Jakarta, 30 Agustus 2016

Penulis

# DAFTAR ISI

	Halaman
Halaman Judul	
Lembar Persetujuan Dosen Pembimbing	
Lembar Persetujuan Asisten Dosen Pembimbing	
Lembar Bimbingan Tugas Akhir	
Lembar Pernyataan Keaslian	
Abstraksi .....	i
Kata Pengantar .....	ii
Daftar Isi .....	iv
Daftar Gambar.....	vii
Daftar Tabel .....	x
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Pokok Permasalahan .....	2
1.3. Tujuan Penelitian .....	2
1.4. Batasan Masalah.....	3
1.5. Manfaat Tugas Akhir .....	3
1.6. Sistematika Penulisan .....	4
<b>BAB II LANDASAN TEORI .....</b>	<b>6</b>
2.1. Rancang Bangun .....	6
2.2. Definisi Sistem.....	6
2.2.1. Karakteristik Sistem .....	7
2.2.2. Klasifikasi Sistem .....	9
2.3. Konsep Dasar Informasi .....	11
2.3.1. Siklus Informasi .....	11
2.3.2. Kualitas Informasi.....	12
2.4. Pengertian Sistem Informasi .....	13
2.5. Siklus Hidup Pengembangan Sistem .....	13

2.5.1. Metodologi Pengembangan Sistem.....	14
2.6. Produk .....	18
2.7. Produk Gagal .....	19
2.8. Pengendalian .....	20
2.9. Laporan .....	21
2.9.1. Jenis Laporan .....	21
2.10. Kualitas .....	23
2.11. <i>Quality Control</i> .....	24
2.12. <i>Flowchart</i> .....	24
2.12.1. Jenis-Jenis <i>Flowchart</i> .....	26
2.13. <i>Unified Modeling Language (UML)</i> .....	28
2.13.1. Jenis-Jenis Diagram Pada UML.....	28
2.14. <i>Multiplicity</i> .....	38
2.15. Kamus Data.....	42
2.16. <i>Hierarchy plus Input-Proses-Output (HIPO)</i> .....	43
2.16.1 Diagram Dalam Paket HIPO .....	43
2.17. MySQL .....	46
2.17.1 Kelebihan MySQL .....	46
2.17.2. Tipe Data Pada MySQL .....	48
2.18. PHP .....	50
2.18.1 Kelebihan PHP .....	51
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>52</b>
3.1. Metodologi Penelitian .....	52
3.2. Metode Pengumpulan Data .....	52
3.3. Metode Pengembangan Sistem .....	53
3.4. Kerangka Pemecahan Masalah .....	54
<b>BAB IV PENGUMPULAN DAN PENGOLAHAN DATA .....</b>	<b>58</b>
4.1. Sejarah Singkat Perusahaan .....	58
4.2. Metode Pengumpulan Data .....	58
4.3. Produk .....	59
4.4. Proses Produksi .....	59

4.5. Proses Kualitas Platting .....	60
4.6. Perencanaan Aplikasi Menggunakan UML .....	62
4.6.1. <i>Use Case</i> .....	62
4.7. Dokumen-Dokumen Yang Digunakan .....	66
<b>BAB V ANALISIS DAN PEMBAHASAN .....</b>	<b>68</b>
5.1. Analisis Kebutuhan <i>User</i> .....	68
5.2. Sistem Informasi Lapran <i>Platting</i> yang Diusulkan.....	69
5.3. Analisis Sistem .....	71
5.3.1. <i>Use Case Diagram</i> yang Diusulkan .....	71
5.3.2. <i>Activity Diagram</i> yang Diusulkan .....	75
5.3.3. <i>Sequence Diagram</i> yang Diusulkan .....	80
5.3.4. <i>Collaboration Diagram</i> yang Diusulkan .....	86
5.4. Perancangan Data .....	92
5.4.1. <i>Class Diagram</i> yang Diusulkan .....	92
5.4.2. Kamus Data .....	93
5.4.3. <i>Deployment Diagram</i> yang Diusulkan .....	97
5.5. Perancangan Sistem .....	98
5.5.1. <i>Hierachy plus Input-Proses-Output (HIPO)</i> .....	98
5.5.2. <i>Flowchart</i> Program .....	99
5.5.3. Perancangan <i>Interface</i> Sistem .....	102
5.6. Testing Dan Implementasi Sistem .....	111
5.6.1. Implementasi Perangkat Keras Dan Lunak.....	111
<b>BAB VI KESIMPULAN DAN SARAN .....</b>	<b>113</b>
6.1. Kesimpulan .....	113
6.2. Saran .....	113
<b>DAFTAR PUSTAKA .....</b>	<b>114</b>

## DAFTAR GAMBAR

		Halaman
Gambar II.1	Gambaran Karakteristik Sistem .....	9
Gambar II.2	Gambar Siklus Informasi.....	12
Gambar II.3	Model <i>Waterfall</i> Rosa Dan Salahuddin.....	14
Gambar II.4	Model <i>Waterfall</i> Sommerville.....	15
Gambar II.5	Langkah-Langkah Pengembangan <i>Prototipe Evolusioner</i> .....	18
Gambar II.6	UML <i>Diagram</i> .....	29
Gambar II.7	Contoh <i>Use Case Diagram</i> .....	30
Gambar II.8	Contoh <i>Statechart Diagram</i> .....	34
Gambar II.9	Contoh <i>Activity Diagram</i> .....	36
Gambar II.10	Contoh <i>Component Diagram</i> .....	39
Gambar II.11	Contoh <i>Deployment Diagram</i> .....	40
Gambar II.12	<i>Visual Table Of Content</i> (HIPO) .....	44
Gambar II.13	<i>Overview Diagram</i> .....	45
Gambar III.1	Kerangka Penelitian .....	57
Gambar IV.1	Produk Yang Dihasilkan .....	60
Gambar IV.2	<i>Flowmap</i> Laporan Kualitas Hasil <i>Platting</i> .....	63
Gambar IV.3	<i>Use Case Diagram</i> Penerimaan Dan Pengecekan Kualitas <i>Platting</i> .....	65
Gambar IV.4	Laporan Penerimaan Barang Pada PT Surya Sunjaya Industri	66
Gambar IV.5	Laporan Kualitas <i>Platting</i> Pada PT Surya Sunjaya Industri....	67
Gambar V.1	<i>Flowmap</i> Pendataan Kualitas <i>Platting</i> Yang Diusulkan .....	70
Gambar V.2	<i>Use Case Diagram</i> Proses Laporan Kualitas <i>Platting</i> Yang Diusulkan.....	72
Gambar V.3	<i>Activity Diagram</i> Proses <i>Login</i> Yang Diusulkan .....	75
Gambar V.4	<i>Activity Diagram</i> Proses Mengolah Data Master Yang Diusulkan.....	76
Gambar V.5	<i>Activity Diagram</i> Proses Mengolah Data Barang Masuk Yang	

	Diusulkan.....	77
Gambar V.6	<i>Activity Diagram</i> Membuat Laporan Penerimaan Barang Yang Diusulkan.....	78
Gambar V.7	<i>Activity Diagram</i> Mengolah Data Kualitas Hasil Produksi Yang Diusulkan.....	79
Gambar V.8	<i>Activity Diagram</i> Mengolah Data Periodik Yang Diusulkan..	80
Gambar V.9	<i>Sequence Diagram</i> Login Yang Diusulkan.....	81
Gambar V.10	<i>Sequence Diagram</i> Pendataan Data Master Yang Diusulkan .	82
Gambar V.11	<i>Sequence Diagram</i> Pendataan Barang Masuk Yang Diusulkan .....	83
Gambar V.12	<i>Sequence Diagram</i> Laporan Penerimaan Barang Yang Diusulkan.....	84
Gambar V.13	<i>Sequence Diagram</i> Mengolah Data Kualitas Hasil Produksi Yang Diusulkan .....	85
Gambar V.14	<i>Sequence Diagram</i> Mengolah Laporan Periodik Yang Diusulkan.....	86
Gambar V.15	<i>Collaboration Diagram</i> Proses Login Yang Diusulkan.....	87
Gambar V.16	<i>Collaboration Diagram</i> Pendataan Data Master Yang Diusulkan.....	88
Gambar V.17	<i>Collaboration Diagram</i> Mengolah Data Barang Masuk Yang Diusulkan .....	89
Gambar V.18	<i>Collaboration Diagram</i> Laporan Barang Masuk Yang Diusulkan .....	90
Gambar V.19	<i>Collaboration Diagram</i> Mengolah Hasil Produksi Yang Diusulkan .....	91
Gambar V.20	<i>Collaboration Diagram</i> Mengolah Data Periodik Yang Diusulkan .....	92
Gambar V.21	<i>Class Diagram</i> Yang Diusulkan .....	93
Gambar V.22	<i>Deployment Diagram</i> Yang Diusulkan .....	97
Gambar V.23	Hirarki Menu Aplikasi .....	99
Gambar V.24	<i>Flowchart</i> Program Yang Diusulkan .....	100

Gambar V.25	<i>Flowchart</i> Program Yang Diusulkan (Lanjutan) .....	101
Gambar V.26	Rancangan <i>Form Login</i> .....	102
Gambar V.27	Rancangan Menu Utama .....	103
Gambar V.28	Rancangan Data <i>Master User</i> .....	104
Gambar V.29	Rancangan Data Jenis Barang .....	104
Gambar V.30	Rancangan Data Barang .....	105
Gambar V.31	Rancangan Data Spesifikasi <i>Control</i> .....	106
Gambar V.32	Rancangan Data <i>Vendor</i> .....	107
Gambar V.33	Rancangan Data Transaksi Barang Masuk.....	108
Gambar V.34	Rancangan Data Transaksi Kualitas Barang .....	109
Gambar V.35	Rancangan Data Laporan Barang Masuk.....	110
Gambar V.36	Rancangan Data Laporan Kualitas Barang .....	111

## DAFTAR TABEL

	Halaman
Tabel II.1 Simbol-Simbol Diagram Alur .....	25
Tabel II.1 Simbol-Simbol Diagram Alur (Lanjutan) .....	26
Tabel II.2 Simbol-Simbol <i>Use Case Diagram</i> .....	30
Tabel II.2 Simbol-Simbol <i>Use Case Diagram</i> (Lanjutan).....	31
Tabel II.3 Simbol-Simbol <i>Class Diagram</i> .....	32
Tabel II.3 Simbol-Simbol <i>Class Diagram</i> (Lanjutan).....	33
Tabel II.4 Simbol Yang Sering Digunakan Pada <i>Activity Diagram</i> .....	35
Tabel II.4 Simbol Yang Sering Digunakan Pada <i>Activity Diagram</i> (Lanjutan) .....	36
Tabel II.5 Simbol-Simbol <i>Sequence Diagram</i> .....	37
Tabel II.5 Simbol-Simbol <i>Sequence Diagram</i> (Lanjutan).....	38
Tabel II.6 Simbol-Simbol <i>Deployment Diagram</i> .....	40
Tabel II.6 Simbol-Simbol <i>Deployment Diagram</i> (Lanjutan).....	41
Tabel II.7 Simbol <i>Multiplicity</i> .....	41
Tabel II.7 Simbol <i>Multiplicity</i> (Lanjutan).....	42
Tabel II.8 Tipe Data Numerik.....	48
Tabel II.9 Besar Kebutuhan Memori Penyimpanan Tipe Data.....	49
Tabel II.10 Tipe Data <i>String</i> .....	49
Tabel II.10 Tipe Data <i>String</i> (Lanjutan) .....	50
Tabel II.11 Tipe Data Waktu .....	50
Tabel IV.1 Definisi <i>Actor</i> .....	62
Tabel IV.1 Definisi <i>Actor</i> (Lanjutan).....	63
Tabel V.1 Identifikasi Kebutuhan <i>User</i> .....	68
Tabel V.1 Identifikasi Kebutuhan <i>User</i> (Lanjutan) .....	69
Tabel V.2 Definisi <i>Actor</i> .....	72
Tabel V.3 Skenario <i>Use Case Login</i> .....	73
Tabel V.4 Skenario <i>Use Case</i> Mengolah Data Master .....	73
Tabel V.5 Skenario <i>Use Case</i> Mengolah Data Barang Masuk .....	73

Tabel V.5	Skenario <i>Use Case</i> Mengolah Data Barang Masuk(Lanjutan) .....	74
Tabel V.6	Skenario <i>Use Case</i> Membuat Laporan Penerimaan Barang .....	74
Tabel V.7	Skenario <i>Use Case</i> Mengolah Kualitas Hasil Produksi .....	74
Tabel V.8	Skenario <i>Use Case</i> Mengolah Laporan Periodik .....	74
Tabel V.8	Skenario <i>Use Case</i> Mengolah Laporan Periodik(Lanjutan) .....	75
Tabel V.9	Spesifikasi Tabel Barang .....	94
Tabel V.10	Spesifikasi Tabel Barang Masuk .....	94
Tabel V.11	Spesifikasi Tabel Detail <i>Quality Control</i> .....	95
Tabel V.12	Spesifikasi Tabel Jenis Barang .....	95
Tabel V.13	Spesifikasi Tabel Detail <i>Quality Control</i> .....	95
Tabel V.14	Spesifikasi Tabel Spesifikasi .....	96
Tabel V.15	Spesifikasi Tabel <i>Supplier</i> .....	96
Tabel V.16	Spesifikasi Tabel <i>User</i> .....	96

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Perkembangan teknologi informasi saat ini sangat pesat. Seiring dengan pesatnya kemajuan teknologi informasi dewasa ini, informasi merupakan hal yang sangat penting dalam menjalankan sesuatu pekerjaan dan kegiatan. Teknologi informasi merupakan sarana yang sangat penting dan menunjang bagi suatu instansi dalam skala kecil, sedang maupun besar. Sehingga dengan informasi dapat diharapkan mempermudah pekerjaan dan tujuan dapat tercapai secara maksimal.

Perusahaan-perusahaan besar saat ini telah banyak yang menggunakan sistem informasi secara maksimal. Sistem informasi banyak digunakan untuk mengolah proses persediaan bahan baku, proses absensi, proses produksi, proses penggajian, proses laporan, dan lain sebagainya. Sehingga proses pengolahan data lebih akurat, cepat dalam memproses data, dan dapat menyelesaikan kegiatan kerja dengan hasil yang maksimal.

Laporan merupakan gambaran hasil produksi yang berjalan. Terkadang suatu laporan menjadi masalah yang cukup kompleks, ketika pertama kali manajemen harus merumuskan secara cermat kebijakan yang terkait dengan sistem laporan yang ada di perusahaan atau instansi.

Pembuatan laporan merupakan salah satu kegiatan yang sebaiknya dikerjakan dengan sistem terkomputerisasi. Dengan menggunakan sistem manual, dapat terjadi hilangnya beberapa data dan adanya data yang sama. Selain itu sistem ini juga banyak memakan waktu pada saat kita mencari data yang kita inginkan. Dengan menggunakan sistem terkomputerisasi, dapat mengurangi kesalahan-kesalahan yang mungkin terjadi dan proses pencarian data menjadi lebih cepat dan mudah.

PT Surya Sunjaya Industri merupakan suatu perusahaan yang bergerak dalam bidang *Plating Nickel Chrome*, yaitu teknik pelapisan logam(*electroplating*) dengan menggunakan bantuan arus listrik dan senyawa kimia tertentu untuk menghasilkan benda yang terlihat mengkilap. Pembuatan produk yang dilakukan oleh PT Surya Sunjaya Industri yaitu dengan menggunakan produk sesuai dengan pesanan pelanggan yang biasa disebut *Make to Order*.

Laporan produksi PT Surya Sunjaya Industri masih menggunakan kertas lembar(*form*) dan *Microsoft Excel* dalam penyimpanan data proses laporan produksi. Kertas-kertas *form* itu disimpan dalam sebuah map, sehingga resiko kehilangan data sangat besar. Selain itu, penyimpanan data menggunakan *Microsoft Excel* membuat proses pencarian data membutuhkan waktu yang cukup lama.

Untuk mengatasi masalah tersebut, diperlukan teknologi informasi untuk membantu melakukan penyimpanan data hasil laporan yang cepat dan tepat yang pada akhirnya mempercepat pula dalam pengambilan keputusan atau kebijakan manajemen.

## **1.2 Pokok Permasalahan**

Berdasarkan latar belakang tersebut, maka dapat diidentifikasi permasalahan yang terjadi di bagian produksi PT Surya Sunjaya Industri sebagai berikut:

1. Penulisan laporan hasil produksi masih dicatat secara manual.
2. Media penyimpanan laporan masih menggunakan kertas *form* yang disimpan dalam map dan *Microsoft Excel* sebagai penyimpanan data hasil produksi.

## **1.3 Tujuan Penelitian**

Adapun tujuan penelitian dari Tugas Akhir ini adalah sebagai berikut:

1. Membuat aplikasi laporan kualitas hasil *plating* yang datanya disimpan dalam suatu *database*.
2. Membuat aplikasi penyimpanan data berbasis web menggunakan bahasa pemrograman PHP 5.4 dan MySQL 5.5, sehingga mudah dalam proses pencarian data dan dapat mengurangi kehilangan data..

#### **1.4 Batasan Masalah**

Agar dalam penulisan Tugas Akhir ini lebih fokus dan lebih terarah, maka perlu diadakan batasan-batasan sebagai berikut:

1. Hanya mencakup pada data kualitas hasil *plating* PT Surya Sunjaya Industri.
2. Laporan yang dihasilkan tidak membahas masalah bahan baku produksi dan proses produksi.
3. Penelitian dilakukan di PT Surya Sunjaya Industri Cikarang pada bulan Maret 2013.

#### **1.5 Manfaat Tugas Akhir**

Hasil dari pembuatan Tugas Akhir ini diharapkan dapat bermanfaat:

1. Bagi Perusahaan  
Rancang bangun sistem informasi laporan produksi diharapkan dapat mempermudah mengetahui kualitas hasil *plating* pada PT Surya Sunjaya Industri serta dapat menjadi masukan yang bermanfaat untuk perbaikan dan pengembangan sistem informasi kualitas *plating*.
2. Bagi Penulis  
Rancang bangun sistem informasi laporan produksi memberikan pembelajaran untuk menerapkan ilmu yang diperoleh selama perkuliahan serta menambah wawasan dan pengetahuan penulis, khususnya dalam analisis dan perancangan sistem informasi.

## 1.6 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam Tugas Akhir ini terbagi dalam beberapa pokok bahasan, yaitu:

### **BAB I            PENDAHULUAN**

Bab ini membahas tentang hal-hal yang bersifat umum dalam latar belakang masalah, tujuan penelitian, pembahasan masalah, serta sistematika penulisan.

### **BAB II            LANDASAN TEORI**

Bab ini membahas tentang pengertian sistem, informasi, produksi, produk, *flowchart*, *Unified Modeling Language*(UML), HIPO, MySQL, dan PHP sebagai alat bantu untuk membuat rancangan konseptual, serta teori-teori lain yang berhubungan dengan perancangan sistem.

### **BAB III            METODOLOGI PENELITIAN**

Bab ini membahas tentang jenis dan sumber data, metode pengumpulan data, serta langkah-langkah yang akan dilakukan dalam pemecahan masalah termasuk metodologi pengembangan sistem yang digunakan yakni metode *Evolutionary Prototype*.

### **BAB IV            PENGUMPULAN DAN PENGOLAHAN DATA**

Bab ini mengurai tentang hasil pengamatan yang telah dilakukan, seperti fungsi produksi, *flow* proses produksi, dan laporan-laporan yang dibuat pada bagian produksi.

### **BAB V            ANALISIS DAN PEMBAHASAN**

Bab ini berisi tentang perancangan sistem, mulai dari analisis sistem rinci dari pengolahan data, yakni mulai dari analisis *flowchart*, *Unified Modeling Language* sampai dengan rancangan *interface* program dan implementasi aplikasi pendataan kualitas hasil *plating*.

**BAB VI KESIMPULAN DAN SARAN**

Bab ini berisikan kesimpulan dari penelitian serta saran-saran yang diperlukan bagi perusahaan dan peneliti selanjutnya.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Rancang Bangun**

Rancang bangun adalah tahap dari setelah analisis dari siklus pengembangan sistem yang merupakan pendefinisian dari kebutuhan-kebutuhan fungsional, serta menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, termasuk menyangkut mengkonfigurasi dari komponen-komponen perangkat keras dan perangkat lunak dari suatu sistem (Jogiyanto,2005).

#### **2.2 Definisi Sistem**

Kata sistem awalnya berasal dari bahasa Yunani (*sustēma*) dan bahasa Latin (*systema*). Berikut ini ada beberapa pengertian sistem yang diambil dari berbagai sumber. Sistem menurut Jogiyanto (2005) adalah “Suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk menyelesaikan suatu sasaran tertentu”. Sistem menurut Kadir (2003) adalah “Sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan”.

Berdasarkan definisi di atas dapat disimpulkan bahwa sistem pada dasarnya mempunyai sifat umum yaitu: kumpulan dari elemen-elemen atau komponen-komponen atau sub-sub sistem yang saling berhubungan untuk mencapai suatu tujuan tertentu. Setiap bagian subsistem dari sebuah sistem memiliki suatu tujuan dan memberikan pengaruh untuk mencapai tujuan tersebut.

### 2.2.1. Karakteristik Sistem

Suatu sistem mempunyai sifat atau karakteristik tertentu, yaitu mempunyai komponen-komponen (*components*), batasan sistem (*boundary*), lingkungan luar sistem (*environments*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolah (*process*), dan sasaran (*objectives*) atau tujuan (*goal*). (Jogiyanto, 2005).

#### 1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem tidak peduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai suatu yang lebih besar lagi yang disebut *supra system*.

#### 2. Batas Sistem

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas suatu sistem menunjukkan ruang lingkup dari sistem tersebut.

#### 3. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

#### 4. Penghubung Sistem

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Keluaran dari satu subsistem akan menjadi masukan untuk subsistem yang lainnya melalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

#### 5. Masukan Sistem

Masukan adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh di dalam sistem komputer, program adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan data adalah *signal input* untuk diolah menjadi informasi.

#### 6. Keluaran Sistem

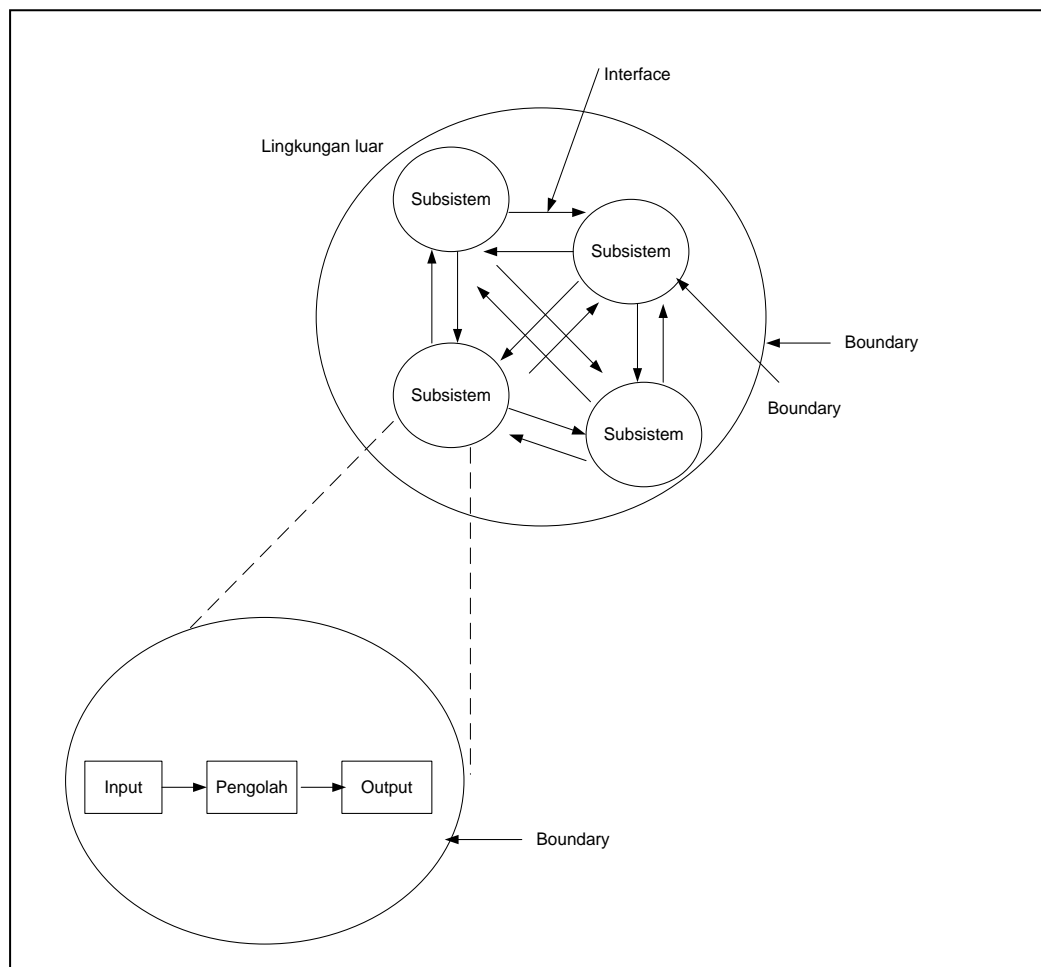
Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem. Misalnya untuk sistem komputer, panas yang dihasilkan adalah keluaran yang tidak berguna merupakan hasil sisa pembuangan, sedang informasi adalah keluaran yang dibutuhkan.

#### 7. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan yang lain menjadi keluaran berupa barang jadi.

## 8. Sasaran Sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.



Gambar II.1 Karakteristik Sistem

Sumber: Jogiyanto, 2005

### 2.2.2. Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut(Jogiyanto, 2005):

1. Sistem diklasifikasikan sebagai sistem abstrak (*abstract system*) dan sistem fisik (*physical system*). Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik misalnya sistem teologia, yaitu sistem yang berupa pemikiran-pemikiran hubungan antara manusia dengan Tuhan. Sistem fisik merupakan sistem yang ada secara fisik, misalnya sistem komputer, sistem akuntansi, dan sistem produksi.
2. Sistem diklasifikasikan sebagai sistem alamiah (*natural system*) dan sistem buatan manusia (*human made system*). Sistem alamiah adalah sistem yang terjadi melalui proses alam dan tidak dibuat manusia, sedangkan sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi antara manusia dengan mesin disebut dengan *human-machine system* atau ada yang menyebut dengan *man-machine system*. Sistem informasi akuntansi merupakan contoh *man-machine system*, karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.
3. Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*). Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Sistem komputer adalah contoh dari sistem tertentu yang tingkah lakunya dapat dipastikan berdasarkan program-program yang dijalankan. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.
4. Sistem diklasifikasi sebagai sistem tertutup (*closed system*) dan sistem terbuka (*open system*). Sistem tertutup merupakan sistem yang tidak berhubungan dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teoritis sistem tertutup ini ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup). Sistem terbuka adalah sistem

yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lainnya.

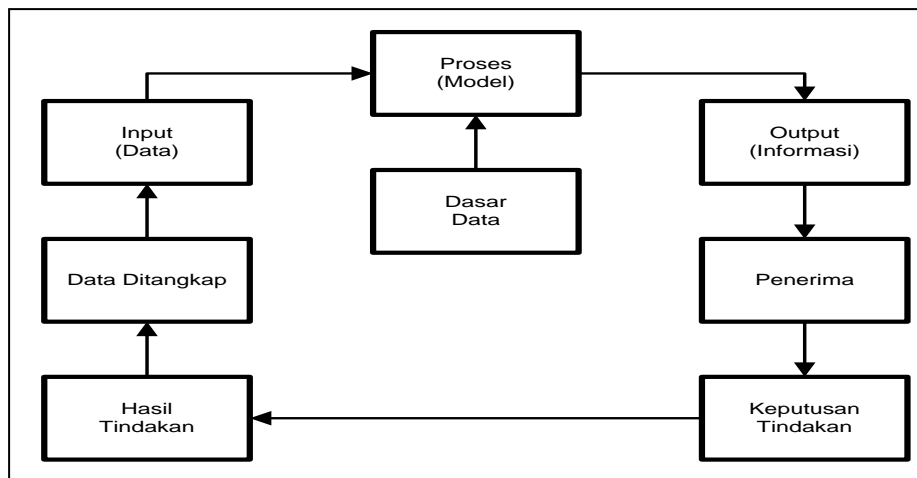
### **2.3 Konsep Dasar Informasi**

Dalam suatu organisasi atau perusahaan, informasi merupakan sesuatu yang memiliki arti yang sangat penting dalam mendukung proses pengambilan keputusan oleh pihak manajemen. Informasi dibutuhkan oleh manajemen untuk menghindari proses *entropy*. Proses *entropy* adalah proses berakhirnya keberadaan suatu sistem manajemen yang didahului kondisi tanpa pola dan tidak menentu. Informasi adalah hasil pengolahan data yang dapat digunakan sebagai dasar pengambilan keputusan. Secara umum informasi dapat didefinisikan sebagai data yang diolah menjadi bentuk yang lebih berarti bagi yang menerimanya (Jogiyanto, 2005).

#### **2.3.1 Siklus Informasi**

Data merupakan bentuk yang masih mentah, belum dapat bercerita banyak sehingga perlu diolah lebih lanjut. Data diolah melalui suatu metode untuk menghasilkan informasi. Data dapat berbentuk simbol-simbol semacam huruf, angka, bentuk suara, gambar, dsb(Jogiyanto, 2005).

Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus(Jogiyanto, 2005). Siklus informasi ini dapat dilihat pada Gambar II.2.



Gambar II.2 Siklus Informasi  
Sumber: Jogiyanto (2005)

### 2.3.2 Kualitas Informasi

Menurut Jogiyanto(2005) kualitas informasi (*quality of information*) sangat dipengaruhi atau ditentukan oleh tiga hal, yaitu :

a. Relevan

Informasi harus memberikan manfaat bagi pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda. Misalnya informasi mengenai sebab akan kerusakan mesin produksi yang diajukan kepada akuntan perusahaan, akan lebih relevan bila ditujukan kepada ahli teknik perusahaan.

b. Akurat

Informasi harus bebas dari kesalahan-kesalahan, tidak menyesatkan, dan harus mencerminkan maksudnya. Ketidakakuratan dapat terjadi karena sumber informasi(data) mengalami gangguan/kesengajaan sehingga merusak/merubah data-data asli tersebut.

c. Tepat waktu

Informasi yang dihasilkan/dibutuhkan tidak boleh terlambat. Informasi yang terlambat tidak mempunyai nilai yang baik, sehingga kalau digunakan sebagai dasar dalam pengambilan keputusan akan berakibat fatal/kesalahan dalam keputusan dan tindakan. Kondisi demikian

menyebabkan mahalnya nilai suatu informasi, sehingga kecepatan untuk mendapatkan, mengolah, dan mengirimkannya memerlukan teknologi-teknologi terbaru.

#### **2.4 Pengertian Sistem Informasi**

Terdapat berbagai macam pengertian Sistem Informasi menurut beberapa ahli, diantaranya:

1. Sistem informasi adalah suatu sistem didalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategis dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan(Jogiyanto, 2005).
2. Sistem Informasi adalah kumpulan elemen yang saling berhubungan satu sama lain untuk membentuk suatu kesatuan untuk mengintegrasikan data, memproses dan menyimpan serta mendistribusikan informasi tersebut(Sutejo, 2006).
3. Sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan, dan mengolah data serta menyediakan informasi keluaran kepada pemakai(Kadir, 2003).

Jadi definisi sistem informasi adalah suatu kumpulan sumber daya manusia atau alat yang terpadu serta modal yang bertanggung jawab untuk mengumpulkan data dan mengolah data demi menghasilkan suatu informasi yang berguna bagi seluruh tingkat operasi untuk kegiatan perencanaan, pelaksanaan, pekerjaan, pengendalian, dan pengambilan keputusan dalam sebuah organisasi.

#### **2.5 Siklus Hidup Pengembangan Sistem**

Menurut Rosa dan Shalahuddin (2011) Siklus Hidup Pengembangan Sistem atau *System Development Life Cycle* (SDLC) adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan

menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik).

Tahapan utama siklus hidup pengembangan sistem terdiri dari beberapa tahap yaitu (Jogiyanto, 2005):

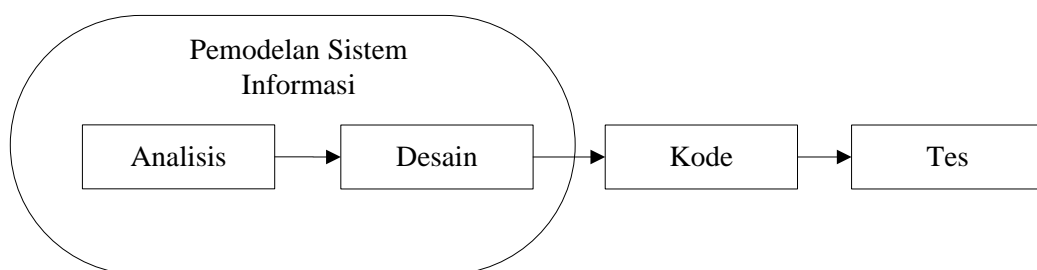
1. Perencanaan sistem (*system planning*)
2. Analisis sistem (*system analysis*)
3. Desain sistem (*system design*)
4. Seleksi sistem (*system selection*)
5. Implementasi sistem (*system implementation*)
6. Perawatan sistem (*system maintenance*)

### 2.5.1 Metodologi Pengembangan Sistem

Metodologi pengembangan sistem terdiri dari beberapa model, yaitu:

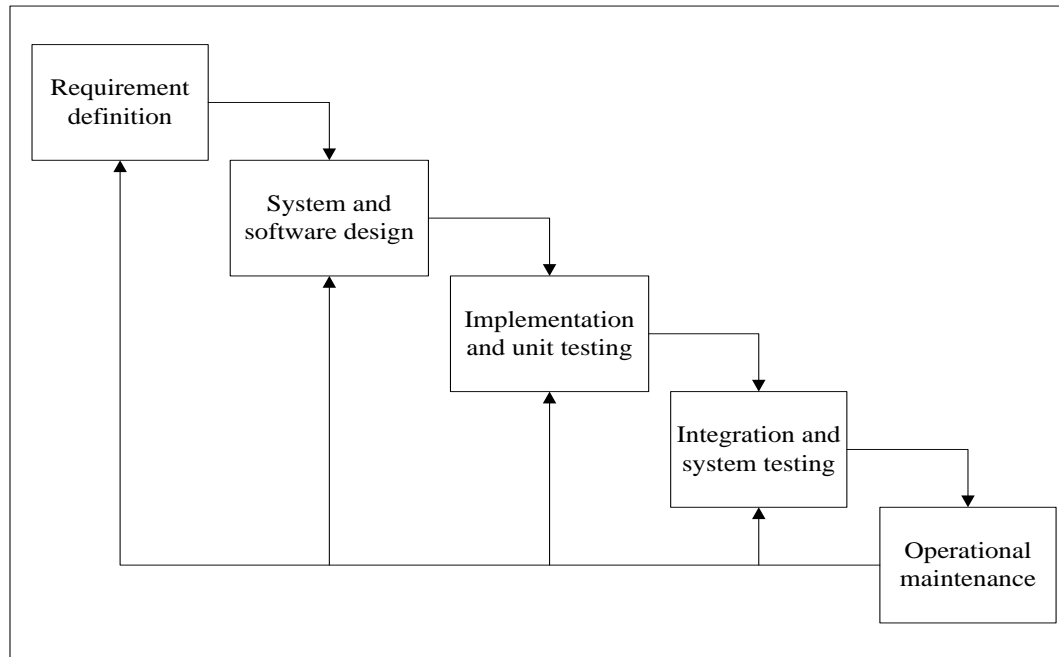
1. Model *Waterfall*

Model air terjun (*waterfall*) sering juga disebut model sekuensial linear (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahapan pendukung (*support*). Berikut adalah gambar model air terjun (Rosa dan Shalahuddin, 2011):



Gambar II.3 Model *Waterfall*  
Sumber: Rosa & Shalahuddin (2011)

Fase-fase dalam Waterfall Model menurut referensi Sommerville (2003), yaitu:



Gambar II.4 Model *Waterfall*

Sumber: Sommerville (2003)

Penjelasannya adalah sebagai berikut:

- a. *Requirements analysis and definition*: Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Fase ini harus dikerjakan secara lengkap untuk bisa menghasilkan desain yang lengkap.
- b. *System and software design*: Desain dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap.
- c. *Implementation and unit testing*: Desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibangun langsung diuji baik secara unit.

- d. *Integration and system testing*: Penyatuan unit-unit program kemudian diuji secara keseluruhan (*system testing*).
- e. *Operation and maintenance*: mengoperasikan program di lingkungannya dan melakukan pemeliharaan, seperti penyesuaian atau perubahan karena adaptasi dengan situasi sebenarnya.

## 2. *Prototyping Model*

*Prototyping model* merupakan salah satu metode pengembangan sistem dan perangkat lunak yang banyak digunakan. Dengan metode *prototyping* ini pengembang dan pengguna dapat saling berinteraksi selama proses pembuatan sistem.

### a. Jenis-Jenis *Prototyping*

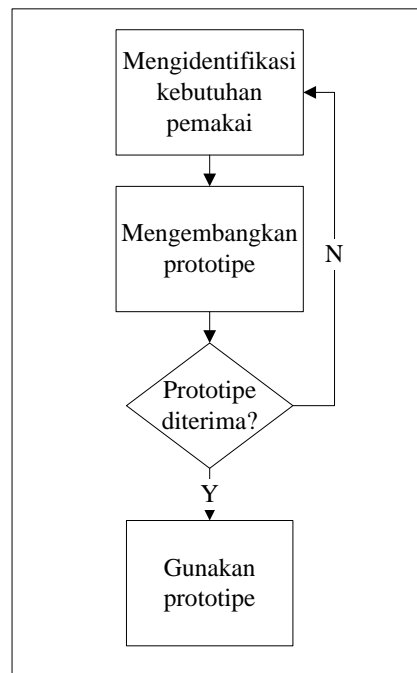
Ada 2 jenis metode *prototype* yang dikembangkan (Mc.Leod, 2007), yaitu:

- 1) *Evolutionary Prototype*, yaitu *prototype* yang terus menerus disempurnakan sampai memiliki seluruh fungsionalitas yang dibutuhkan pengguna dari sistem yang baru.
- 2) *Prototype persyaratan*, yaitu *prototype* yang dikembangkan sebagai salah satu cara untuk mendefinisikan persyaratan-persyaratan fungsional dari sistem baru ketika pengguna tidak mampu mengungkapkan dengan jelas apa yang mereka inginkan.

### b. Langkah-Langkah *Prototyping*

Terdapat beberapa langkah dalam metode pengembangan *prototype evolutioner* (Mc.Leod, 2007), yaitu:

- 1) Mengidentifikasi kebutuhan pengguna. Pengembang mewawancarai pengguna untuk mendapatkan ide mengenai apa yang diminta dari sistem.
- 2) Membuat satu *prototype*. Pengembang mempergunakan satu alat *prototyping* atau lebih untuk membuat *prototype*. Contoh dari alat-alat *prototyping* tersebut adalah generator aplikasi terintegrasi dan *toolkit prototyping*. Generator aplikasi terintegrasi adalah sistem piranti lunak yang siap pakai yang mampu membuat seluruh fitur yang diinginkan dari sistem baru. Sedangkan *toolkit prototyping* meliputi sistem-sistem piranti lunak terpisah, seperti *spreadsheet* elektronik atau sistem manajemen basis data, yang masing-masing mampu membuat sebagian dari fitur-fitur sistem yang diinginkan.
- 3) Menentukan apakah *prototype* dapat diterima. Pengembang mendemonstrasikan *prototype* kepada pengguna untuk mengetahui apakah telah memberikan hasil yang memuaskan. Jika iya, akan dilakukan langkah selanjutnya, dan jika tidak, *prototype* akan direvisi dari tahap awal dengan pemahaman yang lebih baik mengenai kebutuhan pengguna.
- 4) Menggunakan *prototype*. Menjadikan *prototype* sebagai sistem produksi.



Gambar II.5 Langkah-Langkah Pengembangan Prototipe Evolusioner  
 Sumber: Mc.Leod (2007)

### 3. Model *Rapid Application Development* (RAD)

RAD adalah model proses pengembangan perangkat lunak yang bersifat inkremental terutama untuk waktu pengerjaan yang pendek. Model RAD adalah adaptasi dari model air terjun versi kecepatan tinggi dengan menggunakan model air terjun untuk pengembangan setiap komponen perangkat lunak (Rosa dan Shalahuddin, 2011).

## 2.6 Produk

Produk adalah segala sesuatu yang ditawarkan kepada suatu pasar untuk memenuhi keinginan atau kebutuhan. Produk menurut Kotler dan Amstrong(1996) dan Assauri(1999) adalah: “*A product as anything that can be offered to a market for attention, acquisition, use or consumption and that might satisfy a want or need*”. Artinya produk adalah segala sesuatu yang ditawarkan ke

pasar untuk mendapatkan perhatian, dibeli, dipergunakan dan yang dapat memuaskan keinginan atau kebutuhan konsumen.

Menurut Stanton(1996) DALAM Assauri(1999) *“A product is asset of tangible and intangible attributes, including packaging, color, price quality and brand plus the services and reputation of the seller”*. Artinya suatu produk adalah kumpulan dari atribut-atribut yang nyata maupun tidak nyata, termasuk di dalamnya kemasan, warna, harga, kualitas, dan merk di tambah dengan jasa dan reputasi penjualannya.

Menurut Siswanto Sutojo dalam Yudha(2008), ada beberapa faktor penting yang wajib diperhatikan perusahaan dalam penyusunan strategi produk mereka. Faktor pertama adalah strategi pemilihan segmen pasar yang pernah mereka tentukan sebelumnya. Adapun faktor kedua adalah pengertian tentang hakekat produk dimata pembeli. Faktor ketiga adalah strategi produk pada tingkat kombinasi produk secara individual, pada tingkat seri produk dan pada tingkat kombinasi produk secara keseluruhan. Adapun faktor keempat adalah titik berat strategi pemasaran pada tiap tahap siklus kehidupan produk.

## **2.7 Produk Gagal**

Produk gagal adalah produk yang tidak memenuhi standar mutu yang telah ditentukan, tetapi dengan mengeluarkan biaya pengerjaan kembali untuk memperbaikinya, produk tersebut secara ekonomis dapat disempurnakan lagi menjadi produk jadi yang baik (Mulyadi, 2005).

Kegagalan produk adalah faktor-faktor yang terdapat dalam suatu barang atau hasil yang menyebabkan barang atau hasil tersebut tidak sesuai dengan tujuan untuk apa barang atau hasil itu dimaksudkan atau dibutuhkan (Assauri, 1993).

Assauri (1993) mengemukakan dalam bukunya *“Manajemen Produksi dan Operasi”* faktor-faktor yang terdapat dalam kegagalan produk adalah:

- a. Barang-barang yang salah atau cacat
- b. Barang-barang yang tidak mencapai standar mutu yang telah ditetapkan

- c. Bahan-bahan atau komponen yang ternyata tidak dapat dipergunakan
- d. Kondisi produksi ataupun kondisi-kondisi pengolahan (*processing*) yang ternyata tidak terdapat menghasilkan barang-barang yang memenuhi standar mutu yang telah ditetapkan.

Produk gagal dapat mengakibatkan ketidakpuasan karena tidak sesuai dengan standar perusahaan dan *customer*. Ketidaksesuaian spesifikasi dan standar mutu yang telah ditetapkan akan berdampak pada biaya yang dikeluarkan dan waktu untuk melakukan pengerjaan ulang, pemborosan bahan baku, melayani keluhan *customer* dan lain sebagainya. Produk gagal hasil produksi dalam perusahaan disebabkan oleh faktor yang dapat dilacak, misalnya kegagalan memenuhi standar kerja, kerusakan peralatan atau mesin produksi, faktor manusia dan faktor lainnya.

## 2.8 Pengendalian

Pengendalian berasal dari kata “kendali” yang berarti kekang. Menurut KBBI (2008) “pengendalian adalah proses, cara, perbuatan mengendalikan, atau pengawasan atas kemajuan (tugas) dengan membandingkan hasil dan sasaran secara teratur serta menyesuaikan usaha (kegiatan) dengan hasil pengawasan”.

Pengendalian menurut Glenn A. Welsch, Hilton, dan Gordon yang diterjemahkan oleh Purwatiningsih dan Warouw (2000) adalah “suatu proses untuk menjamin terciptanya kinerja yang efisien yang memungkinkan tercapainya tujuan perusahaan”. Menurut Gasperz (2005) “pengendalian dapat diartikan sebagai kegiatan yang dilakukan untuk memantau aktivitas dan memastikan kinerja sebenarnya yang dilakukan telah sesuai dengan yang direncanakan”.

Berdasarkan pengertian di atas dapat ditarik kesimpulan bahwa pengendalian adalah usaha untuk membandingkan prestasi kerja dengan rencana, pemeriksaan, evaluasi dan pemantauan atas pekerjaan yang telah dijalankan sesuai dengan rencana atau tujuan yang telah ditetapkan sebelumnya, dilakukan secara terus menerus agar dapat berfungsi sebaik mungkin untuk mencapai tujuan

perusahaan.

## **2.9 Laporan**

Laporan merupakan suatu bentuk pertanggungjawaban atas suatu tindakan atau kegiatan yang dilakukan. Berikut ini merupakan pengertian laporan yang disampaikan oleh beberapa ahli.

1. Menurut Keraf(2001), Laporan adalah suatu cara komunikasi di mana penulis menyampaikan informasi kepada seseorang atau suatu badan karena tanggung jawab yang dibebankan kepadanya.
2. Laporan berisi informasi yang didukung oleh data yang lengkap sesuai dengan fakta yang ditemukan. Data disusun sedemikian rupa sehingga akurasi informasi yang kita berikan dapat dipercaya dan mudah dipahami(Soegito dalam Wardani 2008).

Berdasarkan pendapat tersebut, dapat disimpulkan bahwa laporan adalah suatu bentuk penyampaian informasi yang didukung oleh data yang lengkap sesuai dengan fakta sehingga informasi yang diberikan dapat dipercaya serta mudah dipahami. Dalam penyampaiannya, laporan dapat bersifat lisan maupun tertulis.

### **2.9.1 Jenis Laporan**

Jenis laporan dibedakan berdasarkan isi, bentuk, sifat, penyampaian, dan waktu priodesasinya.

1. Laporan berdasarkan Isi dan Maksudnya  
Berdasarkan isi dan maksudnya, laporan terdiri atas:
  - a. Laporan informatif, yaitu laporan sebagaimana adanya atau sesuai dengan kenyataan
  - b. Laporan analisis, yaitu laporan yang berupa sumbangan pikiran, pendapat dan saran yang isinya matang dan mendalam

- c. Laporan rekomendasi, yaitu laporan yang berisi penilaian sekilas tanpa analisis mendalam
  - d. Laporan pertanggungjawaban, yaitu laporan yang berupa pertanggungjawaban yang dapat dikerjakan dengan berpedoman pada 2 hal:
    - Apabila proyeknya sudah selesai
    - Jika pekerjaan bertahap/laporan kemajuan
  - e. Laporan kelayakan, yaitu laporan yang bertujuan untuk menentukan mana yang terbaik setelah menganalisa suatu masalah secara mendalam untuk menuju penilaian yang bersifat pilihan layak atau tidak.
2. Jenis laporan Menurut Bentuk
- Menurut bentuknya, laporan dapat dibedakan menjadi:
- a. Laporan berbentuk memo, yaitu laporan yang biasa digunakan untuk keperluan intern organisasi/kantor dan hanya memuat pokok-pokoknya saja (isi laporan).
  - b. Laporan berbentuk surat, yaitu laporan yang dibuat dalam bentuk surat biasa, isinya kira-kira satu sampai tiga, paling banyak 5 halaman.
  - c. Laporan berbentuk naskah, yaitu laporan bentuk naskah dapat pendek atau panjang, penyampiannya memerlukan memo atau surat pengantar.
3. Jenis Laporan Menurut Sifat
- a. Laporan biasa
- Laporan biasa adalah laporan yang isinya bersifat biasa dan tidak rahasia, sehingga jika laporan terbaca orang lain tidak menimbulkan dampak negatif.

- b. Laporan penting  
Laporan penting adalah laporan yang isinya bersifat penting dan rahasia, sehingga hanya orang tertentu saja yang boleh mengetahuinya.
- 4. Jenis Laporan menurut Penyampaian
  - a. Laporan lisan  
Laporan lisan adalah laporan yang disampaikan secara langsung.
  - b. Laporan tertulis  
Laporan tertulis adalah laporan yang disampaikan melalui sebuah tulisan. Contoh: surat, naskah dan memo
  - c. Laporan visual  
Laporan visual adalah laporan yang disampaikan melalui penglihatan. Contoh: disampaikan melalui media presentasi(*power point*)
- 5. Jenis Laporan Menurut Waktu dan Periode Sasi
  - a. Laporan berkala adalah laporan yang dibuat secara periodik atau rutin dalam jangka waktu tertentu (laporan harian, mingguan, bulanan, atau tahunan). Contoh : laporan kehadiran karyawan setiap bulan.
  - b. Laporan insidental adalah laporan yang dibuat apabila diperlukan

## 2.10 Kualitas

Menurut Kamus Besar Bahasa Indonesia (2008), kualitas diartikan sebagai (1) tingkat baik buruknya sesuatu atau (2) derajat atau taraf mutu. Pengertian kualitas menurut beberapa pakar (Yamit, 2013):

1. Kualitas merupakan suatu kondisi dinamis yang berhubungan dengan produk, jasa, manusia, proses dan lingkungan yang memenuhi atau melebihi harapan (Goestch dan David, 1994).

2. Kualitas adalah kelayakan atau kecocokan dalam penggunaan (Juran, 1974).
3. Kualitas adalah kesesuaian dari permintaan spesifikasi (*Conformance to requirement*) (Crosby, 1979).
4. Kualitas sebagai jumlah atribut atau sifat-sifat sebagaimana dideskripsikan dalam produk dan jasa yang bersangkutan, sehingga dengan demikian termasuk dalam kualitas ini adalah daya tahan, kenyamanan pemakaian, daya guna dan sebagainya (Achyari, 1983).
5. Kualitas adalah men-*translate* untuk mengubah kebutuhan yang akan datang dari penggunaan ke dalam suatu karakteristik yang diperlukan agar sebuah produk dapat didesain dan dibuat untuk memberikan kepuasan dengan harga yang dibayar oleh pengguna (Deming, 1986).

### **2.11 *Quality Control***

*Quality control* merupakan suatu kegiatan untuk memastikan apakah kebijakan dalam hal mutu atau ukuran seberapa dekat sebuah barang memiliki kesesuaian dengan standar-standar yang dicantumkan. Dapat tercermin dalam hasil akhir atau pengendalian kualitas dapat dikatakan juga sebagai usaha untuk mempertahankan mutu dan kualitas dari barang yang dihasilkan agar sesuai dengan spesifikasi produk yang telah ditetapkan berdasarkan kebijakan-kebijakan perusahaan. (Nasrullah, 1997)

### **2.12 *Flowchart***



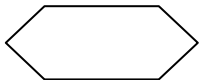

Bagan alir adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi (Jogiyanto, 2005).

Pada waktu akan menggambar suatu bagan alir, analisis sistem atau pemrograman dapat mengikuti pedoman-pedoman sebagai berikut (Jerry (1981) dalam Jogiyanto):

- a. Bagan alir sebaiknya digambar dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
- b. Kegiatan di dalam bagan alir harus ditunjukkan dengan jelas.
- c. Harus ditunjukkan dari mana kegiatan akan dimulai dan dimana akan berakhirnya.
- d. Masing-masing kegiatan di dalam bagan alir sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan, misal:
  - "Persiapkan" dokumen
  - "Hitung" gaji
- e. Masing-masing kegiatan di dalam bagan alir harus di dalam urutan yang semestinya.
- f. Kegiatan yang terpotong dan akan disambung di tempat lain harus ditunjukkan dengan jelas menggunakan simbol penghubung.
- g. Gunakan simbol-simbol bagan alir yang standar.

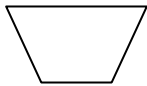


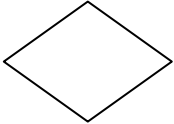
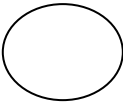
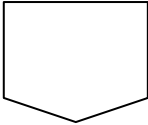

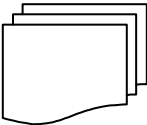
Suatu *flowchart* memberi gambaran dua dimensi berupa simbol-simbol grafis, masing-masing simbol memiliki fungsi dan arti tersendiri. Seperti yang dapat dilihat pada Tabel II.1 berikut ini:

Tabel II.1 Simbol-Simbol Diagram Alur

Simbol	Nama	Fungsi
	<i>Terminator</i>	Awal dan akhir dari suatu proses.
	Garis Akhir ( <i>Front Line</i> )	Arus dari suatu proses
	<i>Preparation</i>	Proses inisialisasi awal
	Proses	Proses pengolahan data

Sumber: Jogiyanto (2005)

Tabel II.1 Simbol-Simbol Diagram Alur(Lanjutan)

Simbol	Nama	Fungsi
	Proses	Proses manual
	<i>Input/Output Data</i>	Mewakili data masukan atau keluaran.
	<i>Predefined Process (Sub Proses)</i>	Permulaan sub proses
	<i>Decision</i>	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	<i>On Page Connector</i>	Penghubung bagian-bagian <i>flowchart</i> yang berada pada satu halaman
	<i>Off Page Connector</i>	Penghubung bagian-bagian <i>flowchart</i> yang berada pada halaman berbeda
	Dokumen	Menggambarkan dokumen asli
	Dokumen Rangkap	Menggambarkan dokumen asli dan tembusannya

Sumber: Jogiyanto (2005)

### 2.12.1 Jenis-Jenis *Flowchart*

Terdapat beberapa jenis bagan alir (*flowchart*), yaitu sebagai berikut (Jogiyanto, 2005):

1. Bagan alir sistem (*systems flowchart*)  
*Systems flowchart* merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem.
2. Bagan alir dokumen (*document flowchart*)  
Bagan alir dokumen (*document flowchart*) atau disebut juga bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusan. Bagan alir dokumen menggunakan simbol-simbol yang sama dengan yang digunakan di dalam bagan alir sistem (*systems flowchart*).
3. Bagan alir skematik (*schematic flowchart*)  
Bagan alir skematik (*schematic flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem. Perbedaan bagan alir skematik dari bagan alir sistem adalah selain menggunakan simbol-simbol bagan alir sistem digunakan pula gambar-gambar komputer dan peralatan lain. Maksud dari penggunaan gambar tersebut adalah untuk memudahkan komunikasi kepada orang yang kurang paham dengan simbol-simbol bagan alir. Penggunaan gambar mudah untuk dipahami, tetapi sulit serta lama pada pengerjaan.
4. Bagan alir program (*program flowchart*)  
Bagan alir program (*program flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dibuat dari derivikasi bagan alir sistem.
5. Bagan alir proses (*process flowchart*)  
Bagan alir proses (*process flowchart*) merupakan bagan alir yang banyak digunakan di teknik industri. Bagan alir ini juga berguna

bagi analisis sistem untuk menggambarkan proses dalam suatu prosedur.

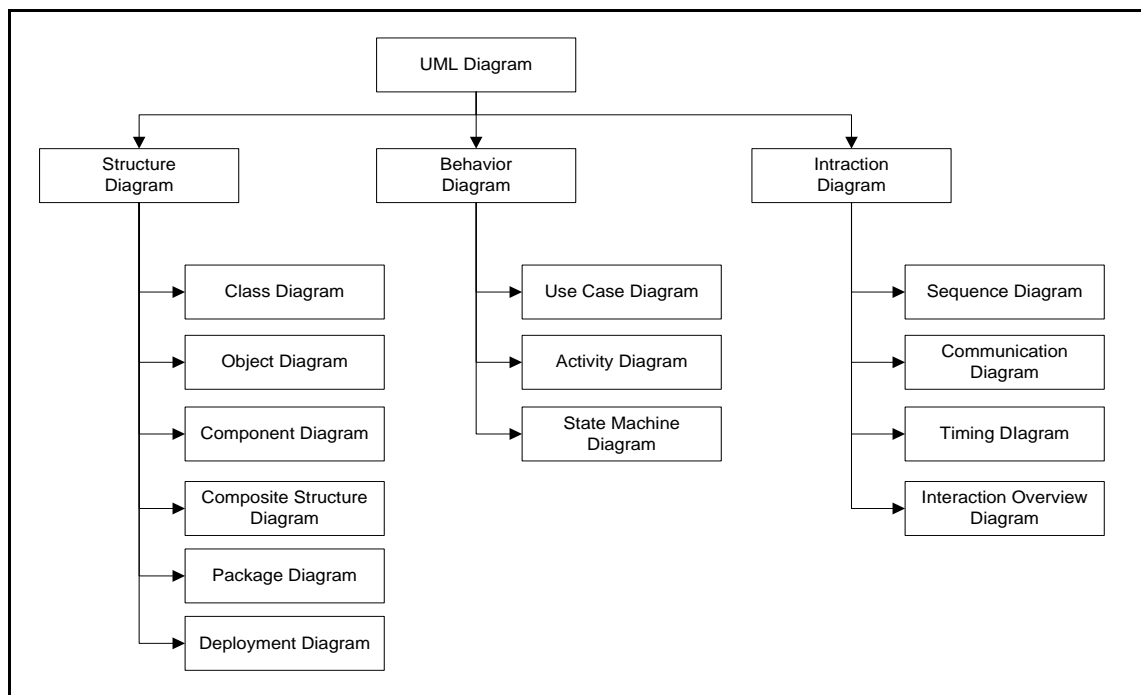
### **2.13 Unified Modeling Language (UML)**

Menurut Munawar (2005), UML (*Unified Modeling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini dikarenakan UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi dan mengkomunikasikan rancangan mereka dengan yang lain. UML merupakan kesatuan dari bahasa permodelan yang dikembangkan oleh Booch, *Object Modeling Technique* (OMT) dan *Object Oriented Software Engineering* (OOSE). Metode *Design Object Oriented* merupakan metode dari Grady Booch. Metode ini menjadikan proses analisis dan desain ke dalam empat tahapan *iterative*, yaitu identifikasi kelas-kelas dan objek-objek, identifikasi *semantic* dari hubungan objek dan kelas tersebut, perincian *interface* dan implementasi.

Sedangkan menurut Henderi (2007), *Unified Modeling Language* adalah sebuah bahasa pemodelan yang telah menjadi standar dalam industri *software* untuk visualisasi, merancang, dan mendokumentasikan sistem.

#### **2.13.1 Jenis-Jenis Diagram pada UML**

Diagram pada UML berbentuk grafik yang menunjukkan simbol elemen model disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu *view* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu. Terdapat beberapa macam diagram dalam *unified modeling language* (UML) seperti pada Gambar II.6.



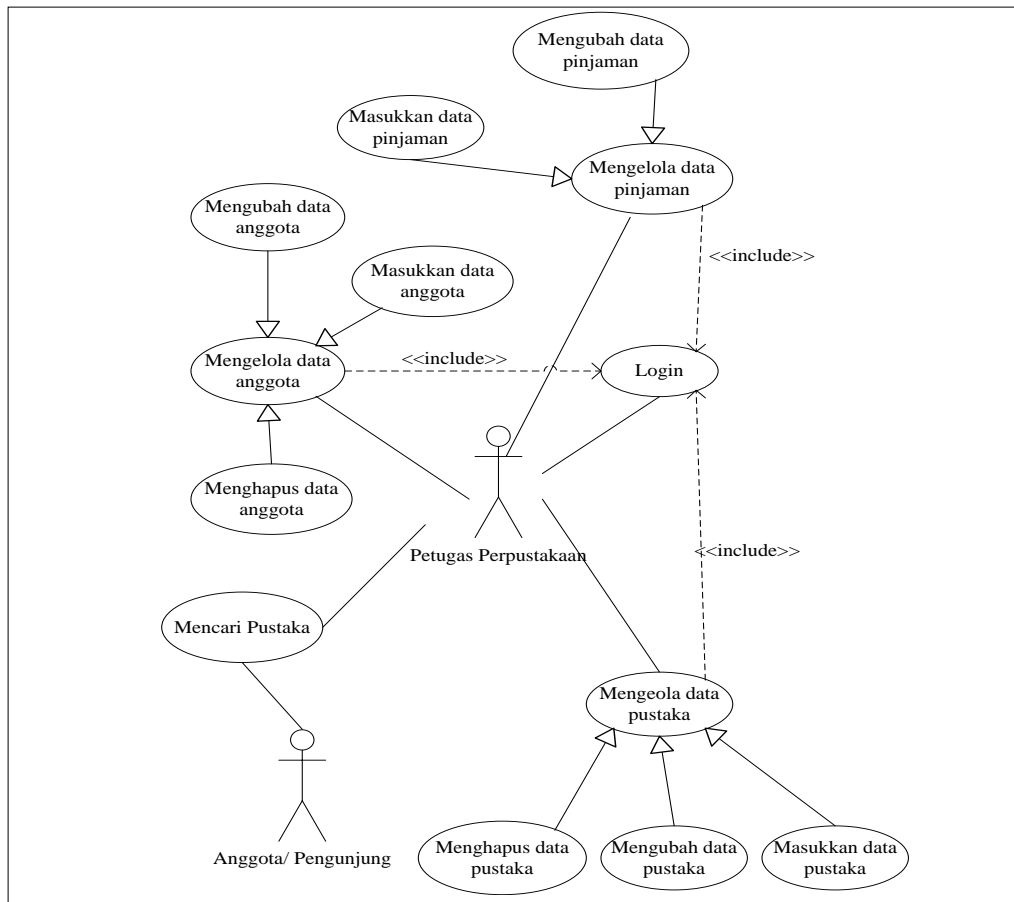
Gambar II.6 UML Diagram  
(Sumber: Rosa & Shalahuddin, 2013)

Berikut ini penjelasan dari gambar II.6 dan akan digunakan dalam laporan:

1. *Use Case Diagram*

*Use Case* adalah deskripsi dari sebuah sistem dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antar *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai (Munawar, 2005).

*Use Case Diagram* merepresentasikan sebuah interaksi antara aktor dengan sistem. Aktor sendiri adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan tertentu (Huda, 2010). Contoh *Use Case Diagram*, seperti Gambar II.7:




Gambar II.7 Contoh Use Case Diagram

Sumber: Rosa & Salahudin (2013)

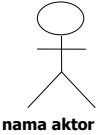

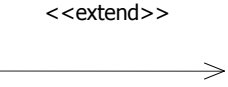

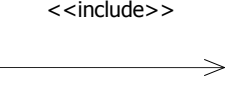
Berikut ini adalah simbol-simbol yang ada pada diagram use case (Rosa dan Shalahuddin, 2013):

Tabel II.2 Simbol-simbol Use Case Diagram

No	Simbol	Deskripsi
1	<p>Use Case</p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case.</p>

Sumber: Rosa dan Shalahuddin (2013)

Tabel II.2 Simbol-simbol *Use Case Diagram*(Lanjutan)

2	<p>Aktor/<i>actor</i></p> 	<p>Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, tapi biasanya dinyatakan menggunakan kata benda atau <i>frase</i>, nama aktor.</p>
3	<p>Asosiasi/<i>association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
4	<p>Ekstensi/<i>Extend</i></p> 	<p>Relasi <i>use case</i> tambahan ke <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu.</p>
5	<p>Generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesifikasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.</p>
6	<p><i>Include</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p>

Sumber: Rosa dan Shalahuddin (2013)

## 2. *Class Diagram*

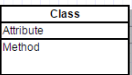
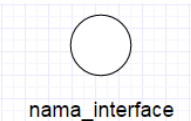
*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan keadaan (atribut atau properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda atau fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain (Munawar, 2005).

*Class* memiliki tiga area pokok:

1. Nama (dan *stereotype*), untuk mengetahui fungsi *class* tertentu.
2. Atribut, menggambarkan keadaan *class diagram* suatu sistem.
3. Metode, menawarkan layanan untuk memanipulasi keadaan suatu sistem.



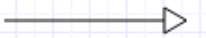


*Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem (Rosa dan Shalahuddin, 2013). Berikut ini adalah simbol-simbol yang ada pada *Class diagram* (Rosa dan Shalahuddin, 2013):

Tabel II.3 Simbol-Simbol *Class Diagram*

No	Simbol	Keterangan
1	Kelas 	Kelas pada struktur sistem.
2	Antarmuka 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.

Sumber: Rosa dan Shalahuddin (2013)

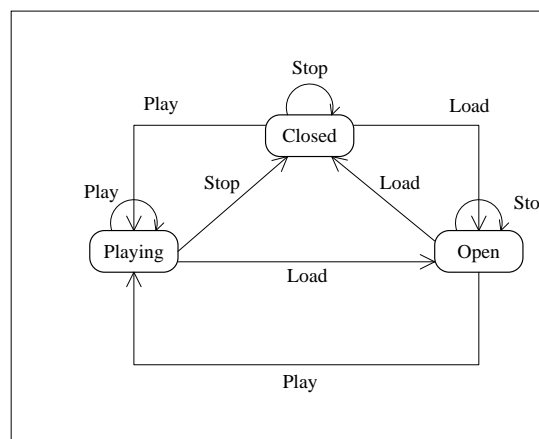
Tabel II.3 Simbol-Simbol *Class Diagram*(Lanjutan)

3	Asosiasi 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4	Asosiasi berarah 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i> .
5	Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus).
6	Kebergantungan 	Relasi antarkelas dengan makna kebergantungan antar kelas.
7	Agregasi 	Relasi antarkelas dengan makna semua-bagian ( <i>whole-part</i> ).

Sumber: Rosa dan Shalahuddin (2013)

### 3. *Statechart Diagram*

*Statechart Diagram* menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*). Dalam UML, *state* digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring. Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah (Munawar, 2005).

Gambar II.8 Contoh *Statechart Diagram*

Sumber: Munawar (2005)





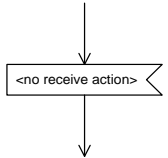

#### 4. *Activity Diagram*

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, dimana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan

titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu (Munawar, 2005).

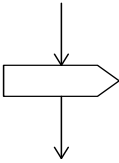
*Activity diagram* menurut Munawar (2005) terdapat beberapa simbol-simbol yang sering digunakan pada saat pembuatan *activity diagram* yaitu:

Tabel II.4 Simbol yang Sering Digunakan pada *Activity Diagram*

Simbol	Keterangan
	Titik Awal ( <i>Initial State</i> ) digunakan untuk memulai suatu aktivitas
	Titik Akhir ( <i>Final State</i> ) digunakan untuk mengakhiri suatu aktivitas
	<i>Activity</i> adalah aktivitas yang dilakukan oleh system
	Pilihan untuk mengambil keputusan ( <i>Decision</i> )
	Tanda Penerimaan ( <i>Signal Receipt</i> )
	<i>Fork</i> digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.

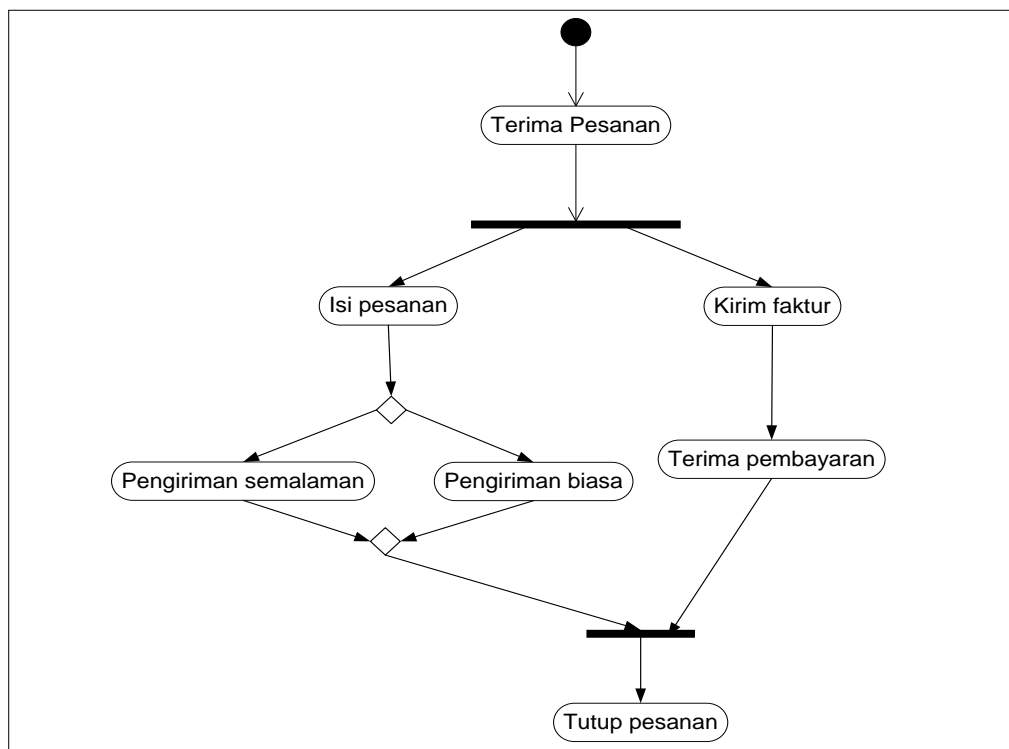
Sumber: Munawar (2005)

Tabel II.4 Simbol yang Sering Digunakan pada *Activity Diagram*(Lanjutan)

Simbol	Keterangan
	Tanda Pengiriman ( <i>Signal Send</i> )

Sumber: Munawar (2005)

Contoh *Activity Diagram* tanpa *swimlane* pada Gambar II.9 berikut:

Gambar II.9 Contoh *Activity Diagram*

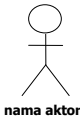
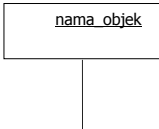

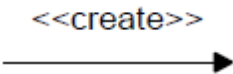
Sumber: Munawar (2005)

### 5. *Sequence Diagram*

*Sequence diagram* menggambarkan interaksi antar objek di dalam maupun di luar sistem berupa *message* yang digambarkan terhadap waktu. Terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Bisa digunakan untuk menggambarkan



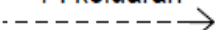
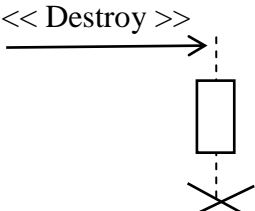
skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* (Huda, 2010). *Sequence diagram* menggambarkan perbuatan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu (Rosa dan Shalahuddin, 2013). Berikut ini adalah simbol-simbol yang ada pada diagram sekuen (Rosa dan Shalahuddin, 2013):

Tabel II.5 Simbol-simbol *Sequence Diagram*

No	Simbol	Deskripsi
1	Aktor 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi itu sendiri.
2	Objek 	Menyatakan objek yang berinteraksi pesan.
3	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
4	Pesan tipe <i>create</i> 	Menyatakan suatu objek membuat objek lain, arah panah mengarah kepada objek yang dibuat.

Sumber: Rosa dan Shalahuddin (2013)

Tabel II.5 Simbol-simbol *Sequence Diagram*(Lanjutan)

No	Simbol	Deskripsi
5	Pesan tipe <i>call</i> 1 : nama_metode() 	Menyatakan suatu objek memanggil operasi atau metode yang ada pada objek lain atau dirinya sendiri.
6	Pesan tipe <i>send</i> 1 : masukan 	Menyatakan bahwa suatu objek mengirimkan data atau masukan atau informasi ke objek lainnya, arah panah, mengarah kepada objek yang dikirim.
7	Pesan tipe <i>return</i> 1 : keluaran 	Menyatakan bahwa suatu objek yang telah dijalankan suatu operasi dan menghasilkan kembalian dari objek tertentu, arah panah mengarah kepada objek yang menerima kembalian.
8	Pesan tipe <i>destroy</i> << Destroy >> 	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah kepada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

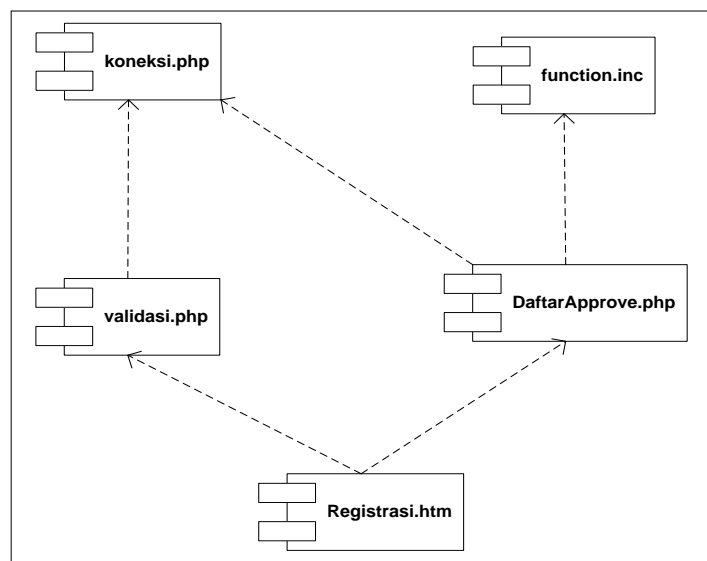
Sumber: Rosa dan Shalahuddin (2013)

#### 6. *Collaboration diagram*

*Collaboration diagram* menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, di mana *message* dari level tertinggi memiliki nomor 1 (Huda, 2010).

### 7. *Component Diagram*

*Component diagram* menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) diantaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time* (Huda, 2010). Gambar II.10 merupakan contoh dari *component diagram*.

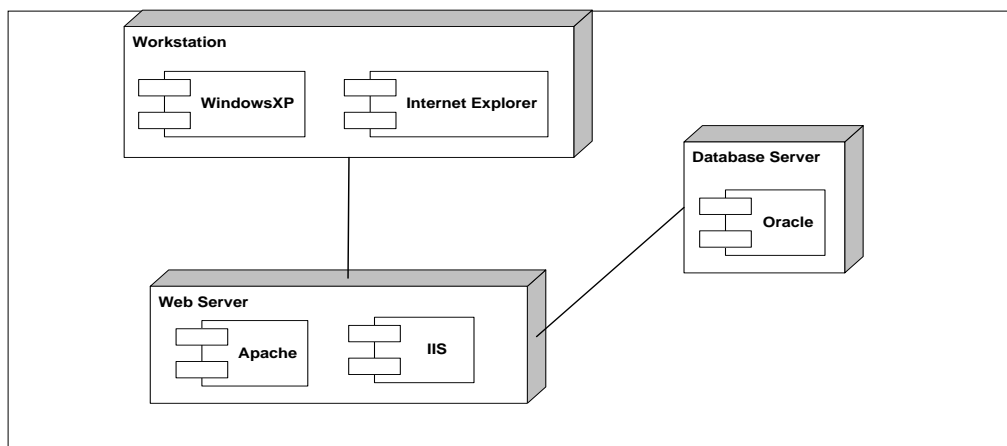


Gambar II.10 Contoh *Component Diagram*

Sumber: Munawar, 2005

### 8. *Deployment diagram*

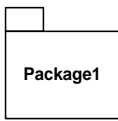


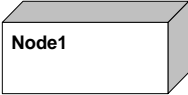
*Deployment diagram* menggambarkan detail bagaimana komponen dibentuk dan didistribusikan (*deploy*) dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server dan hal-hal yang bersifat fisik (Huda, 2010). Contoh *deployment diagram* dapat dilihat pada Gambar II.11.

Gambar II.11 Contoh *Deployment Diagram*

Sumber: Munawar, 2005

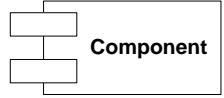
Berikut ini adalah simbol-simbol yang ada pada diagram *deployment* (Munawar, 2005):

Tabel II.6 Simbol-Simbol *Deployment Diagram*

No	Simbol	Keterangan
1	<i>Package</i> 	Merupakan sebuah kumpulan dari satu atau lebih <i>node</i> .
2	<i>Link</i> 	Relasi antar <i>node</i> .
3	<i>Dependency</i> 	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai.
4	<i>Node</i> 	Mengacu pada perangkat keras ( <i>hardware</i> ), perangkat lunak ( <i>software</i> ) yang tidak dibuat sendiri, jika didalam <i>node</i> disertakan

Sumber: Munawar (2005)

Tabel II.6 Simbol-Simbol *Deployment Diagram*(Lanjutan)

No	Simbol	Keterangan
5	<p><i>Component</i></p> 	Suatu komponen dari suatu infrastruktur sistem.

Sumber: Munawar (2005)

### 2.14 *Multiplicity*

*Multiplicity* pada UML memungkinkan untuk menentukan kardinalitas yaitu jumlah elemen dari beberapa kumpulan elemen. *Multiplicity* adalah definisi kardinalitas yaitu jumlah elemen dari beberapa kumpulan elemen dengan memberikan interval termasuk bilangan bulat (*non-negative*) untuk menentukan jumlah contoh elemen dijelaskan. banyaknya interval telah terikat beberapa *lower bound* dan (mungkin tak terbatas) *upper bound* (uml-diagrams, 2015):

*Multiplicity-range* ::= [lower-bound " .."] upper-bound

*Lower-bound* ::= natural-value-specification

*Upper-bound* ::= natural-value-specification | '\*'

*Lower* dan *upper bounds* bisa konstanta alam atau ekspresi konstan dievaluasi untuk angka alam (*non negative*). *upper bound* bisa juga ditetapkan sebagai tanda '\*' yang menunjukkan jumlah elemen yang tidak terbatas. *Upper bound* harus lebih besar dari atau sama dengan *lower bound*. Berikut ini pada Tabel II.7 menjelaskan beberapa contoh tipe *multiplicity* (uml-diagrams, 2015):

Tabel II.7 *Multiplicity*

<i>Multiplicity</i>	<i>Option</i>	<i>Cardinality</i>
0..0	0	Nol
0..1		Antara nol sampai satu
1..1	1	Tepat satu
0..*	*	Antara nol sampai banyak
1..*		Antara satu sampai banyak

Sumber: uml-diagrams (2015)

Tabel II.7 *Multiplicity* (Lanjutan)

<i>Multiplicity</i>	<i>Option</i>	<i>Cardinality</i>
5..5	5	Tepat Lima
m..n		Sedikitnya m tapi tidak boleh lebih dari n

Sumber: uml-diagrams (2015)

### 2.15 Kamus Data

Menurut Jogiyanto (2005), kamus data (*data dictionary*) adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan kamus data diharapkan, analisis sistem dapat mendefinisikan data yang mengalir dalam sistem dengan lengkap. Kamus data dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem.

Kamus data dapat berfungsi membantu pelaku sistem untuk mengartikan aplikasi secara detail dan mengorganisasi semua elemen data yang digunakan di dalam sistem secara persis sehingga pemakai dan penganalisis sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses.

Berikut ini adalah keuntungan penggunaan kamus data (Sommerville, 2003):

1. Kamus data merupakan mekanisme untuk manajemen nama. Banyak orang yang harus menciptakan nama untuk entitas dan relasi ketika mengembangkan model sistem yang besar. Nama-nama ini harus dipakai secara konsisten dan tidak boleh bentrok. Kamus data dapat memeriksa keunikan nama dan memberitahu analisis persyaratan sekiranya terjadi duplikasi nama.
2. Kamus data sebagai tempat penyimpanan informasi yang dapat menghubungkan analisis, desain, implementasi dan evolusi. Sementara sistem dikembangkan, informasi diambil untuk memberitahu perkembangan informasi baru ditambahkan pada sistem. Semua informasi mengenai entitas berada pada satu tempat.

## 2.16 Definisi *Hierarchy plus Input- Process-Output* (HIPO)

Menurut Jogiyanto (2005), *Hierarchy plus Input-Process-Output* (HIPO) merupakan metodologi yang dikembangkan dan didukung oleh IBM. HIPO sebenarnya adalah alat dokumentasi program, akan tetapi sekarang HIPO banyak digunakan sebagai alat desain dan teknik dokumentasi dalam siklus pengembangan sistem. HIPO berbasis pada fungsi, yaitu setiap modul didalam sistem digambarkan oleh fungsi utamanya.

HIPO (*Hierarchy plus Input-Process-Output*) mempunyai sasaran utama sebagai berikut:

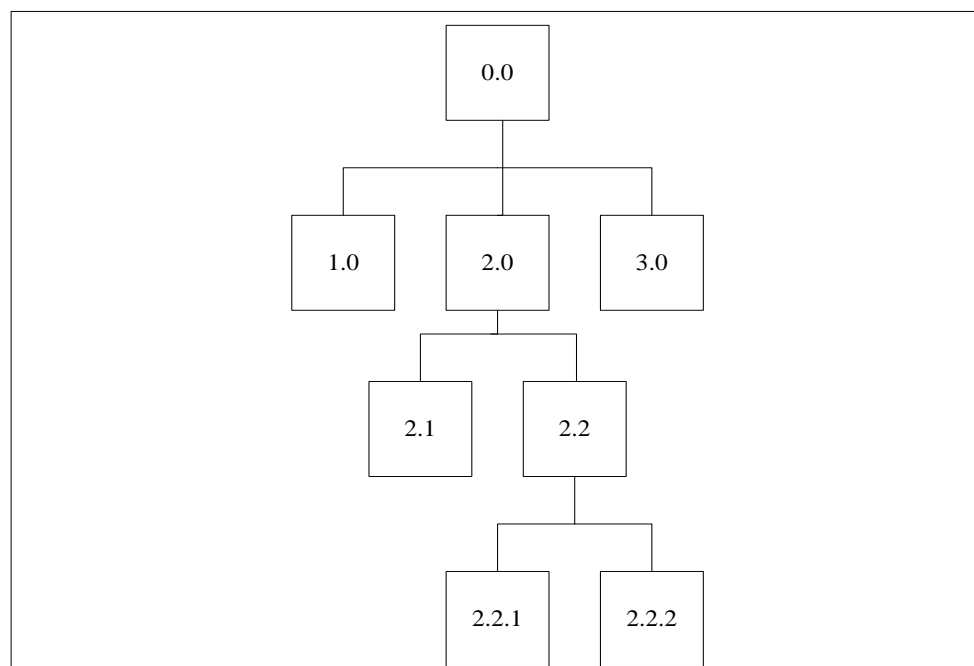
1. Untuk menyediakan suatu struktur guna memahami fungsi-fungsi dari sistem.
2. Untuk lebih menekankan fungsi-fungsi yang harus diselesaikan oleh program, bukannya menunjukkan statemen-statementen program yang digunakan untuk melaksanakan fungsi tersebut.
3. Untuk menyediakan penjelasan yang jelas dari *Input* yang harus digunakan dan *Output* yang harus dihasilkan oleh masing-masing fungsi pada tiap-tiap tingkatan dari diagram-diagram HIPO.
4. Untuk menyediakan *output* yang tepat dan sesuai dengan kebutuhan-kebutuhan pemakai.

### 2.15.1 Diagram Dalam Paket HIPO

Menurut Jogiyanto (2005), HIPO dapat digunakan sebagai alat pengembangan sistem dan teknik dokumentasi program, fungsi-fungsi dari sistem digambarkan oleh HIPO dalam tiga tingkatan. Untuk masing-masing tingkatan digambarkan dalam bentuk diagram tersendiri, dengan demikian HIPO menggunakan tiga macam diagram untuk masing-masing tingkatannya, yaitu sebagai berikut:

1. *Visual Table Of Contents* (VTOC)  
*Visual Table Of Contents* menggambarkan hubungan fungsi-fungsi di sistem secara berjenjang, *visual table of contents*

menggambarkan seluruh program HIPO baik rinci maupun ringkasan yang terstruktur. Pada diagram ini nama dan nomor dari program HIPO diidentifikasi. Struktur paket diagram dan hubungan fungsi juga diidentifikasi dalam bentuk hirarki. *Visual table of contents* dapat dilihat pada Gambar II.12.



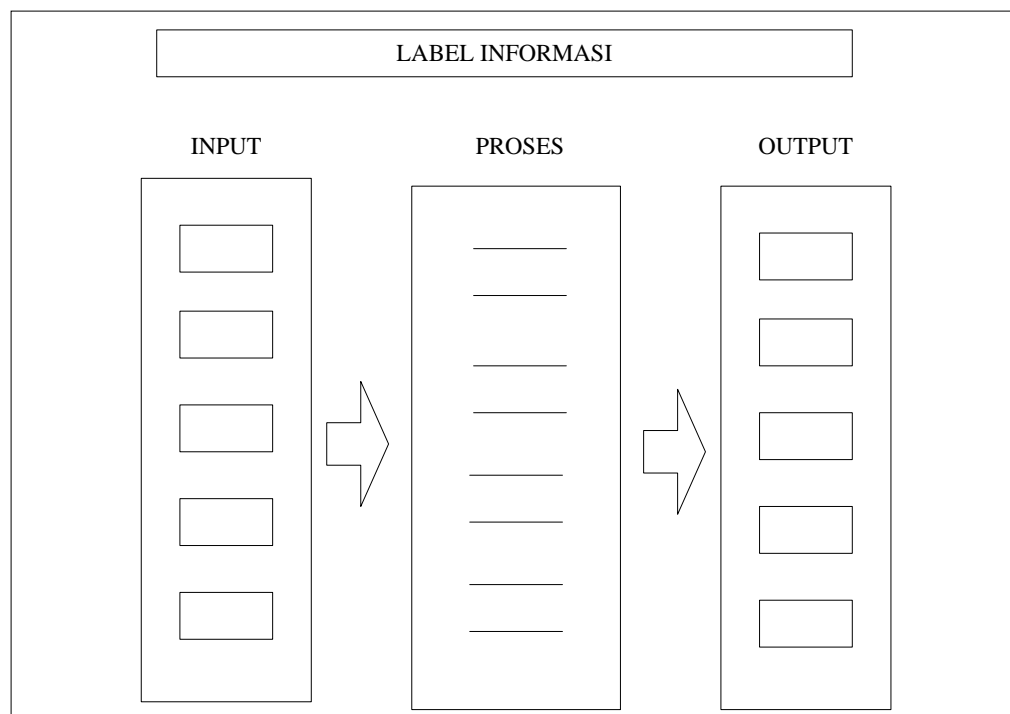
Gambar II.12 Visual Table Of Contents  
(Sumber: Jogiyanto, 2005)

Pada Gambar II.12 menunjukkan terdapat 7 buah fungsi di dalam sistem, berikut penjelasannya:

- Fungsi dengan nomor 1.0, 2.0, 3.0 merupakan tingkatan yang tertinggi
- Fungsi 2.1 dan 2.2 merupakan fungsi dibawah fungsi 2.0
- Fungsi 2.2.1 dan 2.2.2 merupakan fungsi dibawah fungsi 2.2

## 2. *Overview Diagram*

*Overview Diagram* menunjukkan secara garis besar hubungan dari *Input*, proses dan *Output*. Bagian *Input* menunjukkan item-item data yang akan digunakan oleh bagian proses. Bagian proses berisi sejumlah langkah-langkah yang menggambarkan kerja dari fungsi. Bagian *Output* berisi dengan item-item data yang dihasilkan atau dimodifikasi oleh langkah-langkah proses. *Overview Diagram* ini dapat dilihat pada Gambar II.13.



Gambar II.13 Overview Diagram

(Sumber: Jogiyanto, 2005)

## 3. *Detail Diagram*

*Detail Diagram* merupakan diagram tingkatan yang paling rendah didiagram HIPO. Diagram ini berisi elemen-elemen dasar dari paket yang menggambarkan secara rinci kerja dari fungsi.

## 2.17 *My Structure Query Language (MySQL)*

Menurut Kasiman (2006), MySQL adalah suatu *relational database management system* (RDBMS) yang mendukung *database* yang terdiri dari sekumpulan relasi atau tabel, *database* digunakan untuk menyimpan data. Data akan dipanggil pada MySQL melalui PHP, kemudian hasilnya dikirim ke komputer klien untuk ditampilkan pada *browser*. Data pada MySQL dapat dipanggil, dihapus, atau ditambah melalui *query*. MySQL adalah *database* yang menghubungkan *script* PHP menggunakan perintah *query* dan *escape character* yang sama dengan PHP.

Menurut Anhar (2010), MySQL adalah sebuah perangkat lunak sistem manajemen basis data (DBMS) dari sekian banyak DBMS seperti Oracle, MS SQL, Postagre SQL, dan lain-lain. MySQL merupakan DBMS yang *multithread*, *multi-user* yang bersifat gratis dibawah GNU *General Public Licence* (GPL).

### 2.17.1 Kelebihan MySQL

Menurut Anhar (2010), MySQL memiliki beberapa kelebihan antara lain:

1. *Portabilitas*  
MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. *Open Source*  
MySQL didistribusikan secara *open source*, dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.
3. *MultiUser*  
MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *Performance tuning*  
MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. Jenis Kolom

MySQL memiliki tipe kolom yang sangat kompleks, seperti *signed* atau *unsigned integer*, *float*, *double*, *char*, *text*, *date*, *timestamp*, dan lain-lain.

6. Perintah dan Fungsi

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *select* dan *where* dalam perintah (*query*).

7. Keamanan

MySQL memiliki beberapa lapisan sekuritas seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.

8. Konektivitas

MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix *socket* (UNIX), atau *Named Pipes* (NT).

9. Lokalisasi

MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk di dalamnya.

10. Antar Muka

MySQL memiliki *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

11. Klien dan Peralatan

MySQL dilengkapi dengan berbagai peralatan (*tools*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.

### 2.17.2 Tipe data pada MySQL

Beberapa jenis tipe data dalam MySQL memiliki tipe tersendiri dalam tiap *field* pada *table databasenya* (Syaukani, 2005). MySQL mengenal beberapa tipe data field, yaitu:

1. Tipe data Numerik

Tipe data numerik dapat dibedakan menjadi dua kelompok, yaitu tipe data *integer* dan tipe data *floating point*. Tipe data *integer* untuk data bilangan bulat sedangkan tipe data *floating point* digunakan untuk bilangan desimal seperti pada Tabel II.8.

Tabel II.8 Tipe Data Numerik

Tipe data	Kisaran Nilai
Tinyint	0-225
Smallint	0-65535
Mediumint	0-16777215
Int	0-4294967295
Bigint	0-18446744073709551615
Float (x)	(-3.402823466E+38)-(-1.175494351E-38), 0 dan 1.175494351E-38-3.402823466E+38
Float	(-3.402823466E+38)-(-1.175494351E-38), 0 dan 1.175494351E-38-3.402823466E+38
Double	(-1.7976E+308)-(-2.22E-308), 0 dan (2.22E-308)-(1.7976E+308)

(Sumber: Syaukani, 2005)

Besar kebutuhan memori penyimpanan untuk masing-masing tipe data di atas dapat dilihat pada Tabel II.9.

Tabel II.9 Besar Kebutuhan Memori Penyimpanan Tipe Data

<b>Tipe Kolom</b>	<b>Kebutuhan memori penyimpanan</b>
Tinyint	1 byte
Smallint	2 bytes
Mediumint	3 bytes
Int	4 bytes
Integer	4 bytes
Bigint	8 bytes
Float (4)	4 bytes
Float (8)	8 bytes
Float	4 bytes
Double	8 bytes
Double Precision	8 bytes
Real	8 bytes
Decimal (M, D)	M bytes (D+2, if M < D)
Numerical (M, D)	M bytes (D+2, if M < D)

(Sumber: Syaukani, 2005)

## 2. Tipe data String

Yang termasuk dalam tipe data string dapat dilihat pada Tabel II.10 berikut.

Tabel II.10 Tipe Data String

<b>Tipe Kolom</b>	<b>Kebutuhan memori penyimpanan</b>
Char (M)	M bytes, $1 \leq M \leq 225$
Varchar (M)	L+1 bytes, where $L \leq M$ and $1 \leq M \leq 255$
Tinyblob, Tinytext	L+1 bytes, where $L < 2^8$
Blob, Text	L+2 bytes, where $L < 2^{16}$
Mediumblob, Mediumtext	L+3 bytes, where $L < 2^{24}$
Longblob, Longtext	L+4 bytes, where $L < 2^{32}$
ENUM ('value1', 'value2', ...)	1 or 2 bytes, tergantung pada nomor enumeration values (65535 value maximum)

(Sumber: Syaukani, 2005)

Tabel II.10 Tipe Data String(Lanjutan)

Tipe Kolom	Kebutuhan memori penyimpanan
SET ('value1', 'value2', ...)	1, 2, 3, 4 or 8 bytes, tergantung pada nomor set values (64 members maximum)

(Sumber: Syaukani, 2005)

### 3. Tipe data waktu

Yang termasuk dalam tipe data tanggal dan waktu dapat dilihat pada Tabel II.11 berikut.

Tabel II.11 Tipe Data Waktu

Tipe Kolom	Kebutuhan memori penyimpanan
Datetime	8 bytes
Date	3 bytes
Timestamp	4 bytes
Time	3 bytes
Year	1 byte

(Sumber: Syaukani, 2005)

## 2.18 *Hypertext Preprocessor (PHP)*

Menurut Anhar (2010), PHP singkatan dari *Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada *pada server (server side HTML embedded scripting)*. PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru atau *up to date*. Semua *script* PHP dieksekusi pada *server* dimana *script* tersebut dijalankan.

PHP merupakan sebuah bahasa pemrograman *web* yang memiliki sintak atau aturan dalam menuliskan *script* atau kode-kodenya. Salah satu contoh penulisan kode PHP dapat dilihat sebagai berikut ini:

```
<?php
    echo ("Tes Pakai PHP");
```

?>

Kode-kode PHP memiliki tata aturan, yaitu diawali dengan tanda <?php dan diakhiri dengan tanda ?>. Tiap akhir baris harus selalu diber tanda titik koma (;). PHP bersifat *Case Sensitive*, artinya penulisan huruf besar dan kecil pada kode PHP sangat berpengaruh.

### 2.18.1 Kelebihan PHP

Menurut Anhar (2010), beberapa kelebihan PHP dari bahasa pemrograman lain adalah sebagai berikut:

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *Web server* yang mendukung PHP dapat ditemukan dimana-mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan *developer* yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin seperti Linux, Unix, Macintosh, dan Windows serta dapat dijalankan secara *runtime* melalui *console* dan dapat menjalankan perintah-perintah sistem.

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Metodologi Penelitian**

Metodologi penelitian adalah sekumpulan peraturan, kegiatan dan prosedur dalam melakukan penyelidikan yang sistematis untuk meningkatkan sejumlah pengetahuan, juga merupakan suatu usaha yang sistematis dan terorganisasi untuk menyelidiki masalah tertentu yang memerlukan jawaban oleh pelaku dari suatu disiplin ilmu (Nur dan Bambang, 1999).

Barang dapat dikatakan berkualitas apabila layak pada penggunaan produknya, sesuai usulan bentuk dan warna, dan baik atau tidak baik hasil yang dicapainya.

#### **3.2 Metode Pengumpulan Data**

Metode pengumpulan data merupakan cara yang dilakukan oleh penulis dalam mengumpulkan data-data yang diperlukan untuk kegiatan penelitian. Berikut metode pengumpulan data yang dilakukan penulis:

1. Penelitian Lapangan

- a. Observasi

Merupakan kegiatan pengumpulan data dengan cara melakukan peninjauan langsung terhadap sistem yang sedang berjalan yang nantinya akan di analisis untuk mengetahui permasalahan yang ada. Observasi dilakukan di PT Surya Sunjaya Industri pada Departemen Produksi.

- b. Wawancara

Merupakan kegiatan mengumpulkan data dengan cara tanya jawab secara langsung dengan pihak-pihak yang berkaitan dengan sistem produksi.

## 2. Studi Kepustakaan

Dilakukan dengan cara mengumpulkan teori-teori yang berhubungan dengan permasalahan yang akan dibahas.

### 3.3 Metode Pengembangan Sistem

Sistem akan dikembangkan dengan metode pengembangan sistem analisis dan desain berorientasi objek(OOAD). Metode ini menggunakan pemodelan dengan menggunakan diagram-diagram yang terdapat dalam *Unified Modeling Language(UML)*. Sedangkan metode pengembangan perangkat lunak yang digunakan yaitu dengan metode *prototype*.

*Prototype* adalah versi sistem informasi atau bagian dari sistem yang sudah dapat berfungsi, tetapi dimaksudkan hanya sebagai model awal saja. Setelah beroperasi, *prototype* akan lebih jauh diperhalus hingga cocok sekali dengan kebutuhan penggunanya. Ketika rancangan telah difinalisasi, *prototype* dapat dikonversi menjadi sistem produksi yang jauh lebih baik.

Pembuatan *prototype* meliputi pengembangan sistem uji coba yang cepat dan murah untuk dievaluasi oleh pengguna akhir. Lewat interaksi dengan *prototype*, para pengguna dapat memperoleh gagasan yang lebih baik mengenai kebutuhan informasi mereka. *Prototype* yang telah disetujui oleh pengguna dapat digunakan sebagai patokan untuk membuat sistem versi lainnya.

*Prototype* terdapat dua jenis, yaitu evolusioner dan persyaratan. Untuk penulisan tugas akhir ini, penulis menggunakan prototipe evolusioner (*evolutionary prototype*). Prototipe evolusioner terus menerus disempurnakan sampai memiliki seluruh fungsionalitas yang dibutuhkan pengguna dari sistem yang baru. Prototipe ini kemudian dilanjutkan produksi. Jadi, satu *prototype* evolusioner akan menjadi sistem aktual.

Empat langkah dalam pembuatan suatu prototipe evolusioner pada Gambar III.1, yaitu:

1. Mengidentifikasi kebutuhan pengguna. Pengembangan mewawancarai pengguna untuk mendapatkan ide mengenai apa yang diminta dari sistem.
2. Membuat satu prototipe. Pengembang menggunakan salah satu *Low Prototyping* atau lebih untuk membuat prototipe. Contoh dari alat-alat *prototyping* adalah generator aplikasi terintegrasi dan *toolkit prototyping*. Generator aplikasi terintegrasi (*Integrated Application Generator*) adalah sistem piranti lunak siap pakai yang mampu membuat seluruh fitur yang diinginkan dari sistem baru menu, laporan, tampilan, basis data, dan seterusnya. *Toolkit prototyping* meliputi sistem-sistem piranti lunak terpisah, seperti *spread sheet* elektronik atau sistem manajemen basis data, yang masing-masing mampu membuat sebagian dari fitur-fitur yang diinginkan.
3. Menentukan apakah prototipe dapat diterima. Pengembang mendemonstrasikan prototipe kepada para pengguna untuk mengetahui apakah telah memberikan hasil yang memuaskan. Jika ya, langkah 4 akan diambil. Jika tidak, prototipe direvisi dengan mengulang langkah 1, 2, dan 3 dengan penambahan yang lebih baik mengenai kebutuhan pengguna.
4. Menggunakan prototipe. Prototipe menjadi sistem produksi.

### **3.4 Kerangka Pemecah Masalah**

Dalam pemecahan masalah yang dibahas dalam Tugas Akhir ini, penulis menggunakan langkah-langkah atau harapan dalam pemecahan masalah yang ada pada Gambar III.1 sesuai dengan metodologi penelitian yang penulis gunakan dalam pemecahan masalah tersebut. Penjelasan langkah-langkah tersebut adalah sebagai berikut:

1. Pendahuluan

Pendahuluan merupakan awal yang digunakan untuk memulai penelitian. Dengan melakukan studi pustaka, observasi, dan wawancara.

2. Identifikasi Masalah

Pada tahap ini bertujuan untuk mengetahui sistem yang sedang berjalan. Serta mengetahui masalah yang ada pada sistem yang tersebut.

3. Rumusan Masalah

Tahap ini bertujuan untuk mencari jalan keluar dari masalah yang ada pada sistem.

4. Tujuan Penelitian

Tujuan penelitian dimaksudkan agar dapat mencapai tujuan yang hendak dicapai dari penelitian.

5. Batasan Masalah

Pada tahap ini bertujuan untuk membatasi masalah agar penelitian lebih terarah sehingga dapat menunjukkan gambaran yang lebih spesifik mengenai arah pemecahannya.

6. Penerapan Metode *Evolutionary Prototype*

a. Mengidentifikasi kebutuhan pelanggan

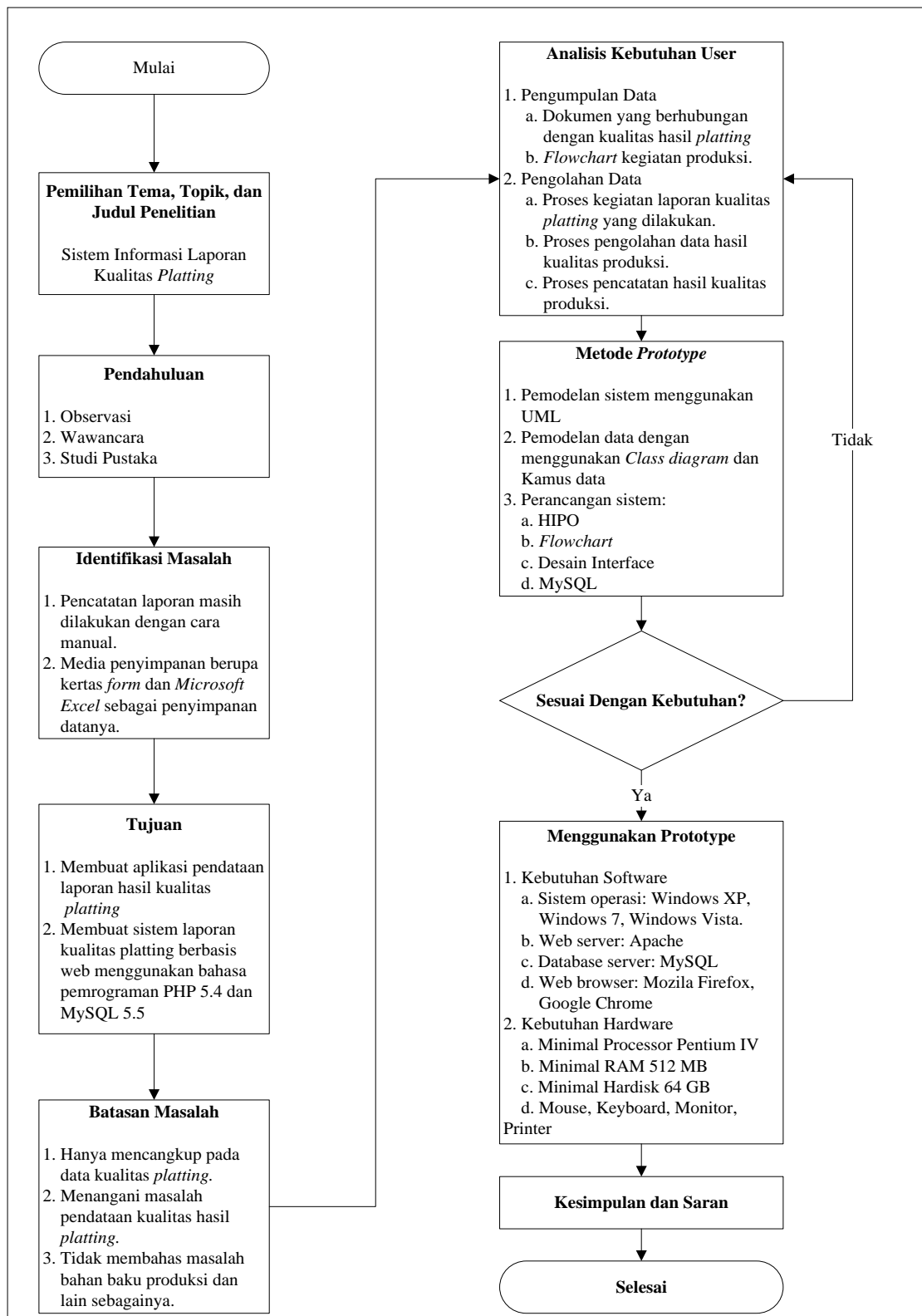
Menganalisis kebutuhan *user* terhadap program yang akan dibuat. Dengan melakukan metode pengumpulan data dan pengolahan data.

b. Membuat *prototype*

Membuat sebuah *prototype* sesuai dengan analisis yang dilakukan. Pada tahap ini akan dirancang aplikasi tentang aplikasi pelaporan hasil produksi. Perancangan sistem nantinya akan membuat tentang:

- 1) Pemodelan sistem menggunakan *Unified Modeling Language*(UML).

- 2) Pemodelan data menggunakan *Class diagram* dan Kamus data.
  - 3) Perancangan sistem menggunakan HIPO, *Flowchart*, Desain *interface*, dan MySQL.
- c. Menggunakan *prototype*
- Memutuskan untuk menggunakan prototipe yang telah di validasi menggunakan metode prototipe evolusioner. Apabila *prototype* tidak diterima maka akan kembali ke tahap identifikasi.
7. Kesimpulan dan Saran
- Kesimpulan digunakan untuk membandingkan hasil penelitian atau pengembangan sistem dengan sistem sebelumnya. Sedangkan saran berisi masukan untuk pengembang sistem selanjutnya.



Gambar III. 1 Kerangka Penelitian  
Sumber: Hasil Analisis (2016)

## **BAB IV**

### **PENGUMPULAN DAN PENGOLAHAN DATA**

#### **4.1 Sejarah Singkat Perusahaan**

PT. Surya Sunjaya Industri dahulunya bernama PT Mekindo Indonesia yang berdiri pada tahun 2005. Namun dengan adanya perpindahan kepemilikan pada tahun 2011, maka perubahan tersebut berubah nama menjadi PT Surya Sunjaya Industri. PT Surya Sunjaya Industri adalah perusahaan yang bergerak di bidang pembuatan *component manufacturing*, yang memegang komitmen secara profesional guna kepuasan pelanggan, dan menggunakan standard management sesuai dengan permintaan customer serta memiliki konsep ***Kaizen*** (*Continuous Improvement* / perbaikan secara terus menerus). Organisasi yang merupakan tim kerja yang berdedikasi tinggi dan memiliki pengetahuan serta keahlian untuk memenuhi berbagai kebutuhan pelanggan / *customer*. Merupakan solusi sederhana hingga solusi terpadu yang rumit, serta Solusi inovatif kami akan membantu anda menjadi perusahaan berskala internasional.

**Prioritas** utama kami dalam berkerjasama adalah membuat Produk dan memberikan Pelayanan yang bermutu tinggi sesuai dengan persyaratan Pelanggan dengan memperhatikan unsur-unsur QCDSM (*Quality, Cost, Delivery, safety* dan *Moral*) dan kelestarian lingkungan secara seimbang (*Green Company*). Juga membangun budaya dan etos kerja yang produktif dengan Melaksanakan kegiatan 5R dan K3, serta mengembangkan tanggung jawab sosial secara menyeluruh, sebagai bentuk CSR (*Company Social Responsibility*) kami.

#### **4.2 Metode Pengumpulan Data**

PT Surya Sunjaya Industri merupakan perusahaan yang bergerak di bidang pembuatan *component manufacturing*, mempunyai misi yaitu Terpercaya,

Terbaik, Beretika. Sedangkan visi PT Surya Sunjaya Industri adalah Tahun, Evaluasi, dan Konsolidasi.

### 4.3 Produk

Produk yang dihasilkan oleh PT Surya Sunjaya Industri adalah *Protector Muffler*, *Protector Siliencer*, dan *Handle Comp*. Jika dilihat berdasarkan fungsi dari produk yang dibedakan menjadi tiga level, maka *Protector Muffler*, *Protector Siliencer*, dan *Handle Comp* termasuk kedalam fungsi produk level dua (*Actual Product*) karena produk ini merupakan alasan utama konsumen untuk membeli dan menggunakannya sebagai aksesoris kendaraan bermotor.

*Protector Muffler*, *Protector Siliencer*, dan *Handle Comp* juga merupakan produk yang berdasarkan aspek daya tahannya ke dalam jenis barang yang tahan lama (*durable goods*). Material yang digunakan dirancang sedemikian rupa agar umur ekonomis, dan kekuatannya tahan lama. Sedangkan dibuat berdasarkan tujuan konsumen membeli barang, produk ini masuk ke dalam *business product/ industrial product*, yaitu produk yang dibeli untuk kemudian dijual lagi.

Produk yang dibuat merupakan produk dengan sistem *Make To Order*. Hasil produksi tersebut diproduksi berdasarkan permintaan *vendor*. Berdasarkan pesanan yang diterima oleh bagian *purchasing* ini akan dibuat suatu perencanaan kebutuhan bahan baku dan penentuan penjadwalan produksinya oleh bagian *production planning*.



Gambar IV.1 Produk yang dihasilkan  
Sumber: Hasil Analisis(2016)

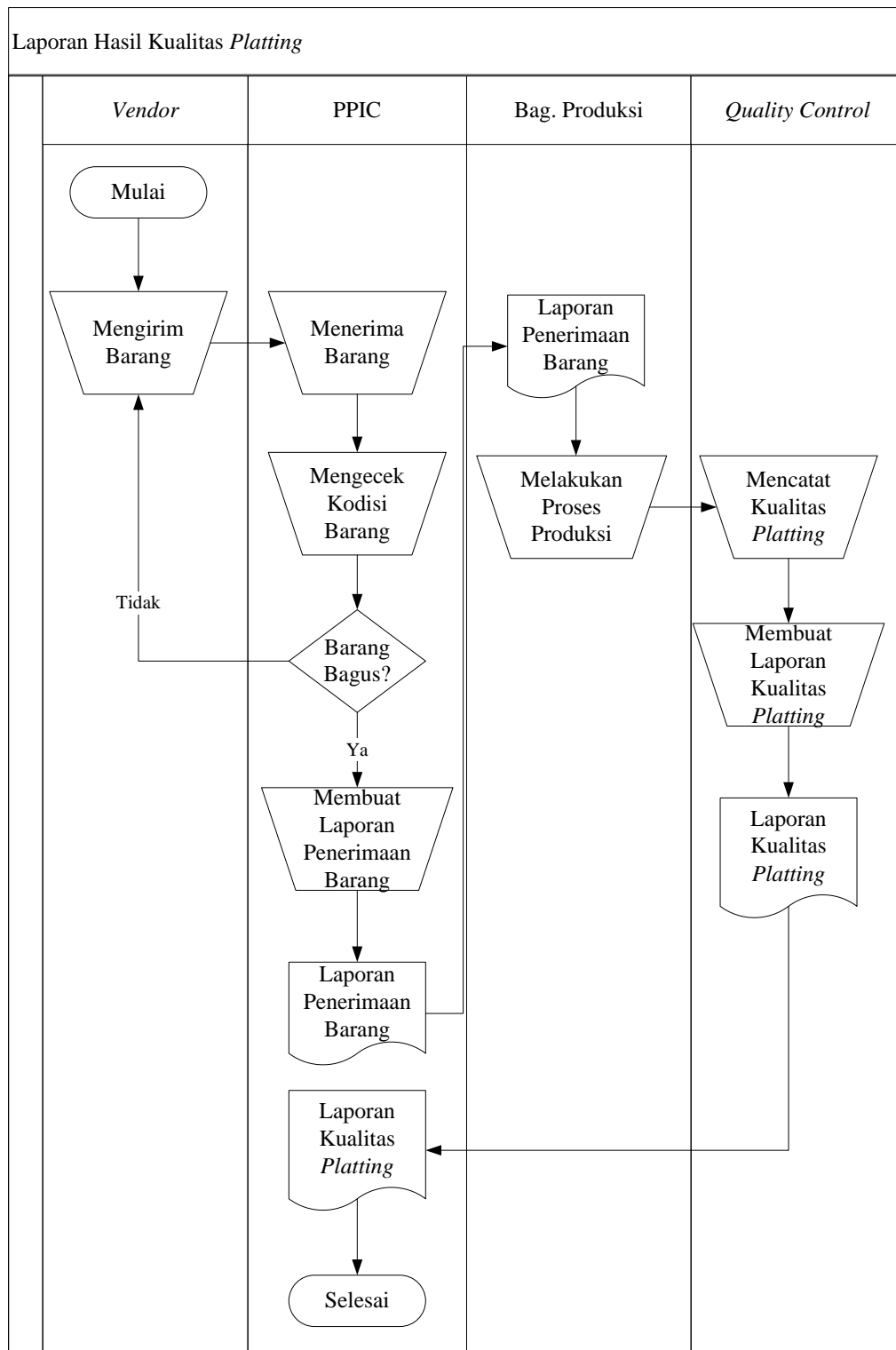
#### 4.4 Proses Produksi

*Plating* merupakan proses yang digunakan untuk memanipulasi sifat suatu substrat dengan cara melapisinya dengan logam lain. Proses *plating* yang dilakukan PT Surya Sunjaya Industri dalam pembuatan produk yang dihasilkan adalah sebagai berikut:

1. *Buffing*  
*Buffing* adalah proses penghalusan permukaan dengan menggunakan mesin *grinding* manual.
2. Inspeksi *loading*  
Inspeksi *loading* adalah proses pengecekan bahan baku sebelum produksi.
3. *Loading*  
*Loading* adalah proses mempersiapkan dan memasukkan bahan baku untuk produksi.
4. Masuk tank atau tangki air  
Dalam proses *plating*, bahan baku yang ada dimasukkan kedalam tangki air atau bak yang dicampur zat kimia dan bahan pendukung lainnya dalam kegiatan produksi.
5. *Unloading* dan *Inspeksi*  
Merupakan kegiatan mengeluarkan dan mengecek hasil produksi.
6. *Baking oven*  
Merupakan proses memasukkan hasil produksi ke dalam oven untuk dipanaskan.
7. Inspeksi akhir  
Melakukan pengecekan akhir terhadap barang hasil produksi.
8. Penyimpanan material siap kirim  
Menyimpan barang jadi untuk siap dikirim.

#### **4.5 Proses Laporan Hasil Kualitas *Plating***

Proses laporan hasil kualitas produksi dalam suatu perusahaan perlu dicatat agar terdata banyaknya barang yang dihasilkan dalam proses produksi. Hal itu dilakukan agar memudahkan pemimpin dalam mengambil keputusan. Berikut gambar *flowmap* proses laporan hasil kualitas *plating* pada PT Surya Sunjaya Industri.



Gambar IV.2 *Flowmap* Laporan Kualitas Hasil *Plating*  
 Sumber: Hasil Analisis(2016)

#### 4.6 Perancangan Aplikasi Pengecekan Kualitas Barang Hasil Produksi Menggunakan *Unified Modeling Language* (UML)

UML(*Unified Modeling Language*) adalah salah satu alat bantu yang sangat hadal di dunia pengembangan sistem yang berorientasi objek. Hal ini dikarenakan UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti, serta dilengkapi dengan mekanisme yang efektif untuk berbagi dan mengkomunikasikan rancangan mereka dengan yang lain.(Munawar, 2005)

##### 4.6.1 *Use Case Diagram* yang Berjalan pada Sistem Informasi Laporan Kualitas *Platting*

*Use Case* adalah deskripsi dari sebuah sistem dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antar *user*(pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai

*Use case diagram* menggambarkan siapa saja *actor* yang melakukan prosedur dalam sistem serta fungsi-fungsi (proses) yang terlibat dalam transformasi pada sistem tersebut. Pada dasarnya *actor* bukanlah bagian dari *use case diagram*, namun untuk terciptanya suatu *use case diagram* diperlukan beberapa *actor*, dimana *actor* tersebut menggambarkan seseorang atau sesuatu seperti perangkat, atau sistem lain yang berinteraksi dengan sistem. *Actor* adalah seseorang atau sistem lain yang berinteraksi mengirim atau menerima pesan dari dan ke dalam sistem dengan sistem.

Tabel IV.1 Definisi *Actor*

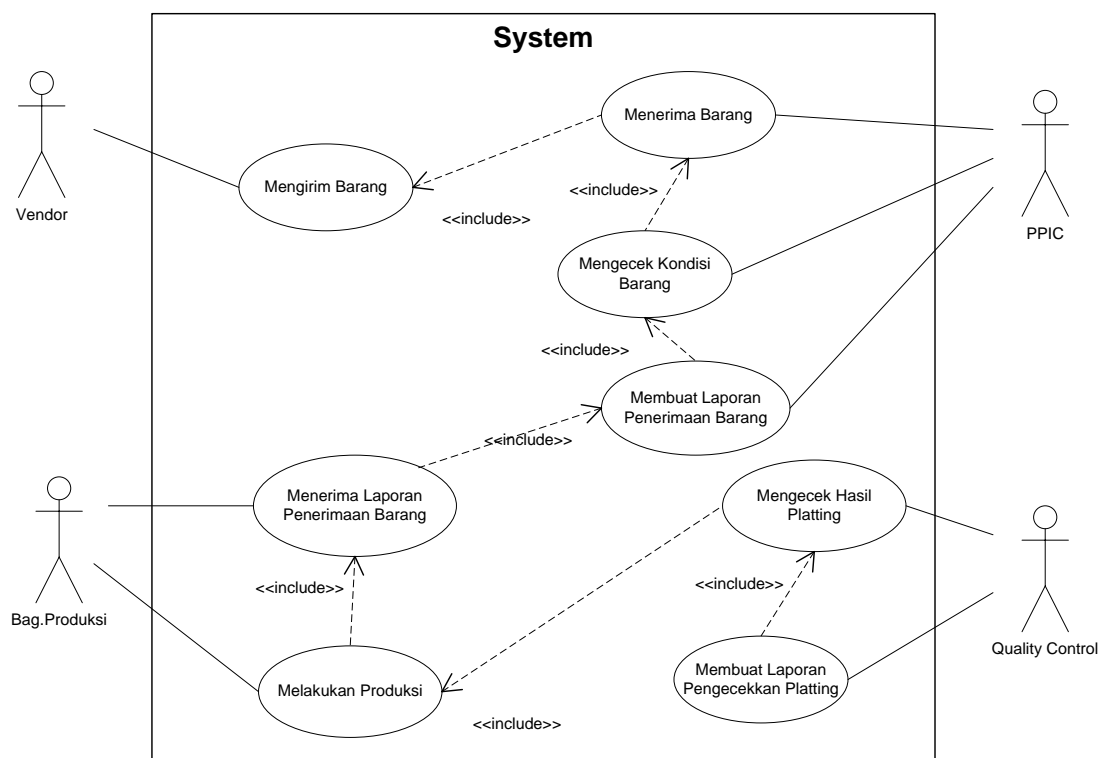
No	<i>Actor</i>	Keterangan
1.	Vendor	Perusahaan yang bekerja sama untuk menyediakan bahan baku untuk produksi
2.	PPIC	Bagian dalam perusahaan yang bertugas untuk memesan dan menerima bahan baku yang

Sumber: Hasil Analisis(2016)

Tabel IV.1 Definisi Actor(Lanjutan)

		digunakan untuk produksi
3.	Bag. Produksi	Bagian dalam perusahaan yang bertugas melakukan kegiatan produksi
4.	<i>Quality Control</i>	Bagian dalam perusahaan yang bertugas melakukan pengecekan kualitas barang hasil produksi

Sumber: Hasil Analisis(2016)



Gambar IV.3 Use Case Diagram Penerimaan dan Pengecekan Kualitas *Platting*

Sumber: Hasil Analisis(2016)

Berdasarkan sistem yang berjalan, terdapat beberapa permasalahan yang sering dihadapi oleh perusahaan. Permasalahan-permasalahan tersebut antara lain:

1. Masih kurang efektif dan efisiennya proses pengolahan data laporan hasil *plating*.
2. Media Penyimpanan dan sistem pencatatan laporan kualitas *plating* masing menggunakan cara manual, maka pencarian data akan membutuhkan waktu yang cukup lama.

3. Tidak adanya informasi pengolahan laporan kualitas *plating* yang terkomputerisasi.


#### 4.7 Dokumen-Dokumen Yang Digunakan

##### 1. Laporan Penerimaan Barang

 <b>SSI</b> <small>PT SURYA SUNJAYA INDUSTRI</small>	<b>LAPORAN PENERIMAAN BARANG</b>		<u>No.Dok</u>	015/LPB/PPIC/00
			<u>Tanggal</u>	
			<u>Revisi</u>	
Vendor :				
<u>Tanggal</u> :				
<u>NO</u>	<u>Part Name</u>	<u>Qty</u>	<u>Keterangan</u>	
1.				
2.				
3.				
4.				
5.				
6.				
Total				
<u>Diketahui,</u>		<u>Diperiksa,</u>	<u>Dibuat,</u>	
<u>PPIC Dept</u>		<u>Administrasi</u>	<u>Opr. WH</u>	

Gambar IV.4 Laporan Penerimaan Barang Pada PT Surya Sunjaya Industri  
Sumber: Hasil Analisis(2016)

2. Laporan Kualitas *Platting*

	<b>LAPORAN KUALITAS PLATTING</b>				No.Dok	015/LPB/QC/00
					Tanggal	
					Revisi	
Hari/Tanggal :						
No	Item Problem					Total
1.						
2.						
3.						
4.						
5.						
6.						
7.						
8.						
9.						
10.						
11.						
12.						
Total QC						
Total NG						
Total OK						
<u>Keterangan:</u>		Diketahui.	Diperiksa.	Dibuat.		
		SPV QC	Leader	Operator		

Gambar IV.5 Laporan Kualitas *Platting* Pada PT Surya Sunjaya Industri  
 Sumber: Hasil Analisis(2016)

## BAB V

### ANALISA DAN PEMBAHASAN

#### 5.1 Analisis Kebutuhan *User*

Berdasarkan hasil pengolahan data dan analisis sistem, dibutuhkan sistem informasi laporan kualitas *plating* yang sudah terprogram agar mempermudah mengolah dan memproses data. Sistem informasi kualitas *plating* hanya menampilkan data barang hasil *plating* yang siap dikirim. Untuk mendukung sistem informasi laporan kualitas *plating* yang terprogram dibutuhkan kemampuan *user* dalam mengoperasikan aplikasi tersebut, untuk mewujudkan hal tersebut maka harus membuat aplikasi yang *user friendly* agar lebih mudah mengoperasikannya bagi pengguna dalam menggunakan aplikasi.

Kebutuhan program aplikasi dapat dijelaskan melalui proses-proses yang dibutuhkan pemakai dan kebutuhan sistem berdasarkan usulan solusi yang dibuat. Identifikasi kebutuhan sistem komputerisasi menjelaskan proses-proses yang dilakukan oleh aplikasi, dimana proses-proses tersebut dibutuhkan oleh pemakai. Pada tabel V.I dijelaskan identifikasi kebutuhan *user* pada PT Surya Sunjaya Industri.

Tabel V.I Identifikasi Kebutuhan *User*

No	Identifikasi Kebutuhan	Uraian
1.	Proses Sistem Berjalan	Melakukan proses <i>plating</i> Mengecek hasil <i>plating</i> Membuat laporan hasil <i>plating</i>
2.	Data Masukan( <i>input</i> )	Data barang Data kualitas <i>plating</i>
3.	Data Keluaran( <i>output</i> )	Laporan barang Laporan kualitas <i>plating</i>

Sumber: Hasil Analisis Data(2016)

No	Identifikasi Kebutuhan	Uraian
4.	Model sistem informasi	<p><i>Input</i> : Data barang</p> <p><i>Proses</i> : Menginput data barang</p> <p><i>Output</i> : Data barang baru</p> <p><i>Input</i> : Data kualitas <i>plating</i></p> <p><i>Proses</i> : Menginput data kualitas <i>plating</i></p> <p><i>Output</i> : Data kualitas <i>plating</i> Baru</p>

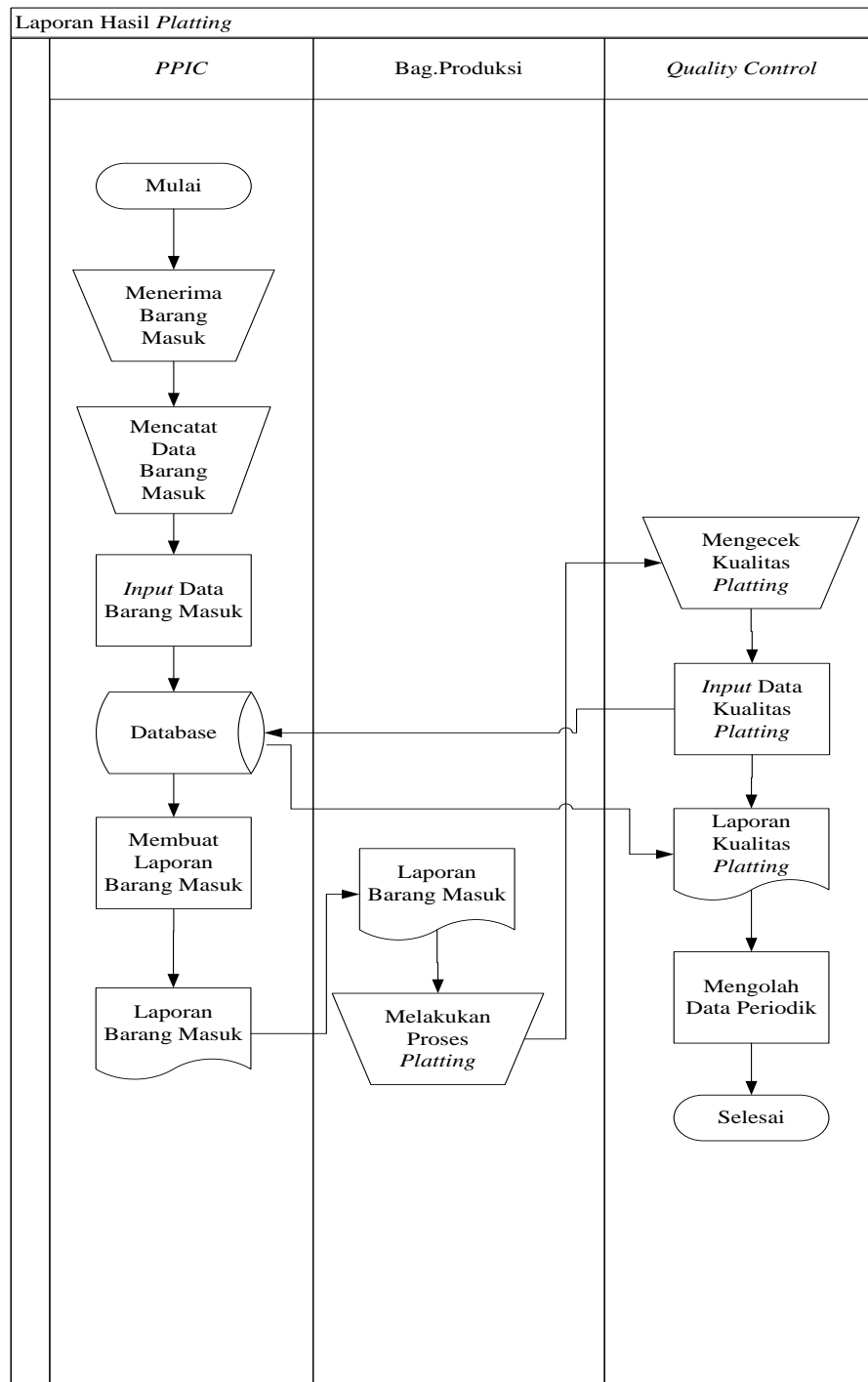
Sumber: Hasil Analisis Data(2016)

## 5.2 Sistem Informasi Laporan Kualitas *Plating* yang Diusulkan

Sistem informasi yang diusulkan bertujuan untuk mengembangkan prosedur atau proses suatu data agar tujuan dari suatu organisasi dapat tercapai. Perancangan dibuat untuk meminimalkan kekurangan dari sistem lama yang kurang efektif dan efisien ke dalam sistem baru yang lebih terprogram dan cepat dalam pengerjaan dalam perusahaan. Berikut adalah rancangan sistem informasi laporan kualitas *plating* usulan:

1. *PPIC* menginput data barang masuk.
2. Data barang masuk disimpan ke dalam database.
3. Data barang masuk yang disimpan ke dalam database, diolah menjadi laporan barang masuk.
4. Laporan barang masuk di kirimkan ke Bag.Produksi untuk di lakukan kegiatan Produksi.
5. Bag.Produksi melakukan kegiatan Produksi.
6. Barang hasil produksi di kirimkan ke bagian *quality control* untuk di cek.
7. Data dari mengecek barang hasil produki di *input* ke dalam sistem.

8. Data hasil *input*-an mengecek tersebut diolah menjadi laporan hasil kualitas *plating*.



Gambar V.I *Flowmap* Pendataan Kualitas *Plating* Yang Diusulkan  
 Sumber: Hasil Analisis Data(2016)

### 5.3 Analisis Sistem

Analisis sistem merupakan penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan. Pada tahap ini dilakukan upaya memperbaiki sistem atau membangun sistem dengan memanfaatkan teknologi baru untuk mengurangi dan mengatasi berbagai permasalahan yang telah terjadi pada sistem yang lama sehingga dapat meningkatkan efektifitas dan efisiensi. Aplikasi yang dirancang diharapkan memberi solusi alternatif baru yang memberi kemudahan terhadap proses pembuatan laporan kualitas *plating* pada PT Surya Sunjaya Industri.

Tujuan analisis sistem adalah untuk memberikan penjelasan dan mendefinisikan kepada *user* yang akan menggunakan program. Pembuatan program ini diharapkan dapat mengatasi kekurangan yang ada dan membantu *user* dalam menyelesaikan pekerjaannya. Adapun perancangan yang diusulkan merupakan langkah untuk lebih mengefektifkan dan mengefisienkan sistem yang sudah terkomputerisasi dengan menggunakan sistem yang terprogram.

#### 5.3.1 Use Case Diagram Yang Diusulkan

*Use case* merupakan pemodelan untuk sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Alasan memilih diagram *use case* untuk menggambarkan sistem yang diusulkan, agar mempermudah dalam menggambarkan dan membaca sistem informasi yang akan diusulkan.

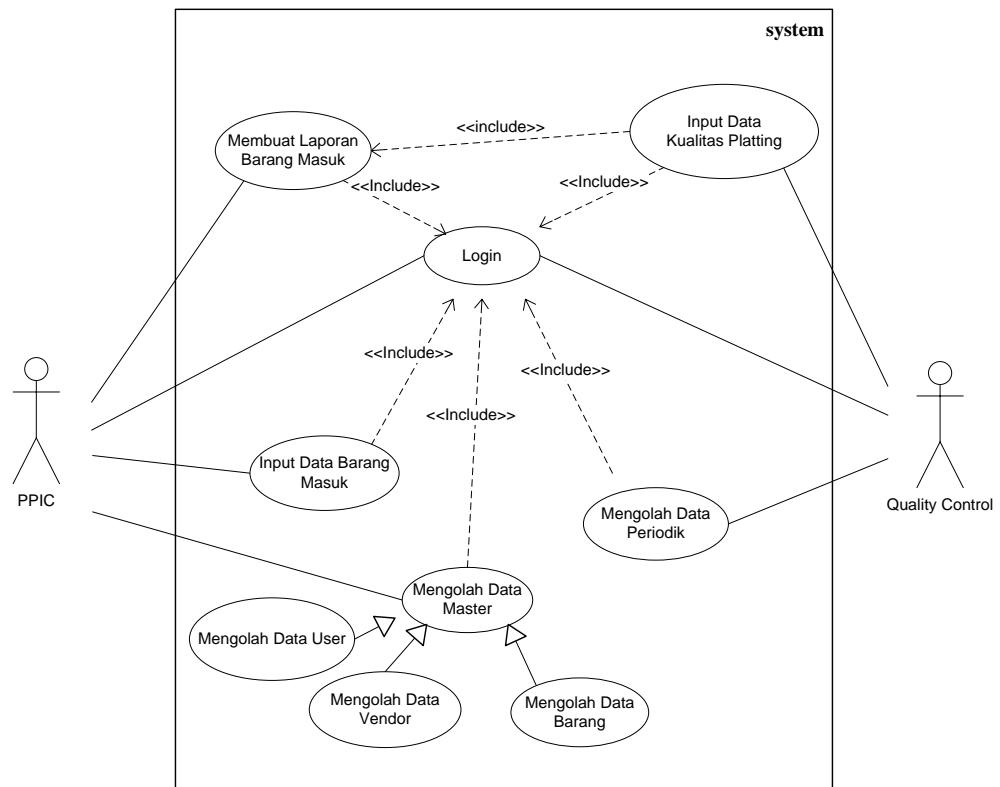
Berikut adalah definisi *use case* dan definisi aktor dari *use case* diagram proses laporan kualitas *plating* yang diusulkan:

Tabel V.2 Definisi Actor

No	Aktor	Deskripsi
1.	PPIC	Aktor yang melakukan pendataan data barang masuk dan data master
2.	<i>Quality Control</i>	Aktor yang melakukan pengecekan kualitas <i>plating</i> dan pendataan hasil produksi

Sumber: Hasil Analisis Data(2016)

Berdasarkan analisis yang telah dilakukan, maka *use case diagram* yang diusulkan untuk sistem informasi laporan kualitas *plating* pada PT Surya Sunjaya Industri dapat dilihat pada gambar V.2 berikut ini:



Gambar V.2 Use Case Diagram Proses Laporan Kualitas *Plating* Yang Diusulkan  
Sumber: Hasil Analisis Data(2016)

Berikut adalah skenario *use case diagram* proses laporan kualitas *plating* yang diusulkan:

1. Skenario *Use Case Login*Tabel V.3 Skenario *Use Case Login*

<i>Use case name</i>	<i>Login</i>
Actor	<u>PPIC dan <i>Quality Control</i></u>
<i>Description</i>	Melakukan identifikasi pada sistem
<i>Normal Flow</i>	Memasukan <i>username</i> Memasukan <i>password</i> Melakukan identifikasi
<i>Alternate Flow</i>	Tidak dapat memasuki sistem apabila <i>username</i> dan <i>password</i> tidak terdaftar ataupun melakukan salah pengetikan.

Sumber: Hasil Analisis Data(2016)

2. Skenario *Use Case Mengolah Data Master*Tabel V.4 Skenario *Use Case Mengolah Data Master*

<i>Use case name</i>	Mengolah data master
Actor	PPIC
<i>Description</i>	Mengelola data master pada aplikasi
<i>Normal Flow</i>	Mengolah data <i>user</i> . Mengolah data barang. Mengolah data vendor.

Sumber: Hasil Analisis Data(2016)

3. Skenario *Use Case Mengolah Data Barang Masuk*Tabel V.5 Skenario *Use Case Mengolah Data Barang Masuk*

<i>Use case name</i>	Mengolah data barang masuk
Actor	PPIC
<i>Description</i>	Mengelola data barang masuk yang telah diterima
<i>Normal Flow</i>	Menerima barang dari vendor. Menghitung barang yang diterima.

Sumber: Hasil Analisis Data(2016)

Tabel V.5 Skenario *Use Case* Mengolah Data Barang Masuk(Lanjutan)

	Menghitung barang yang diterima dan siap proses. Mengimput data barang masuk pada sistem
--	---

Sumber: Hasil Analisis Data(2016)

4. Skenario *Use Case* Membuat Laporan Barang MasukTabel V.6 Skenario *Use Case* Membuat Laporan Penerimaan Barang

<i>Use case name</i>	Membuat laporan penerimaan barang
Actor	PPIC
<i>Description</i>	Membuat laporan penerimaan barang yang diterima
<i>Normal Flow</i>	Mengecek data barang masuk pada sistem Memilih waktu barang masuk Mencetak data barang masuk

Sumber: Hasil Analisis Data(2016)

5. Skenario *Use Case* Mengolah Data Kualitas Hasil ProduksiTabel V.7 Skenario *Use Case* Mengolah Kualitas Hasil Produksi

<i>Use case name</i>	Mengolah data kualitas hasil produksi
Actor	<i>Quality Control</i>
<i>Description</i>	Mengelola data kualitas barang hasil produksi
<i>Normal Flow</i>	Menerima barang hasil produksi Mengecek barang hasil produksi Mencatat data barang hasil produksi ke aplikasi

Sumber: Hasil Analisis Data(2016)

6. Skenario *Use Case* Mengolah Laporan PeriodikTabel V.8 Skenario *Use Case* Mengolah Laporan periodik

<i>Use case name</i>	Mengolah laporan periodik
Actor	<i>Quality Control</i>

Sumber: Hasil Analisis Data(2016)

Tabel V.8 Skenario *Use Case* Mengolah Laporan periodik(Lanjutan)

<i>Description</i>	Mengelola laporan periodik kualitas barang
<i>Normal Flow</i>	Melihat kembali data kualitas produksi yang lama. Mencetak data kualitas produksi Melihat perbandingan hasil kualitas produksi dari beberapa hari.

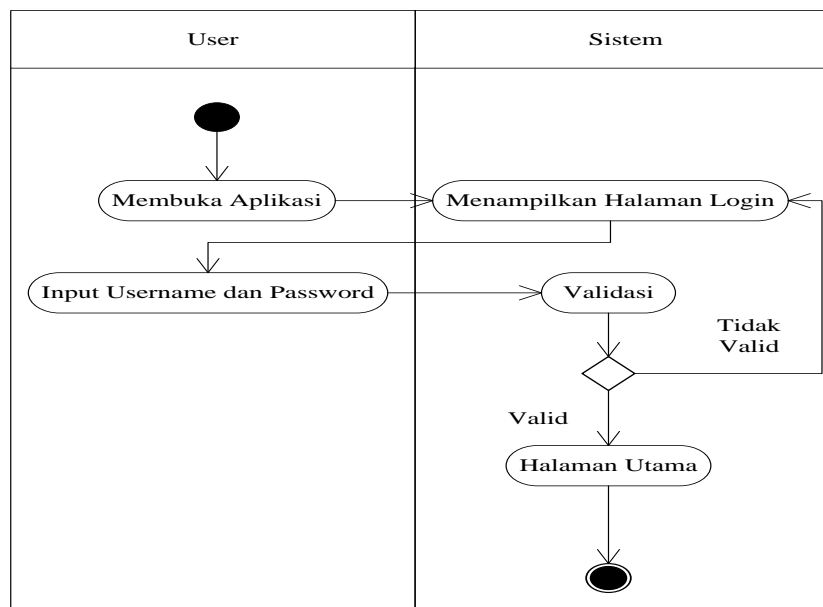
Sumber: Hasil Analisis Data(2016)

### 5.3.2 Activity Diagram Yang Diusulkan

*Activity diagram* digunakan untuk menggambarkan kegiatan-kegiatan yang ada didalam sistem. Agar lebih memahami sitem yang akan dibuat, maka perlu dibuatkan *activity diagram* tentang sistem informasi laporan kualitas *plating*.

#### 1. Activity Diagram Proses Login

Diagram berikut merupakan diagram aktivitas yang menjelaskan kegiatan *login* terhadap sistem, seperti yang dapat dilihat pada gambar V.3 berikut ini:



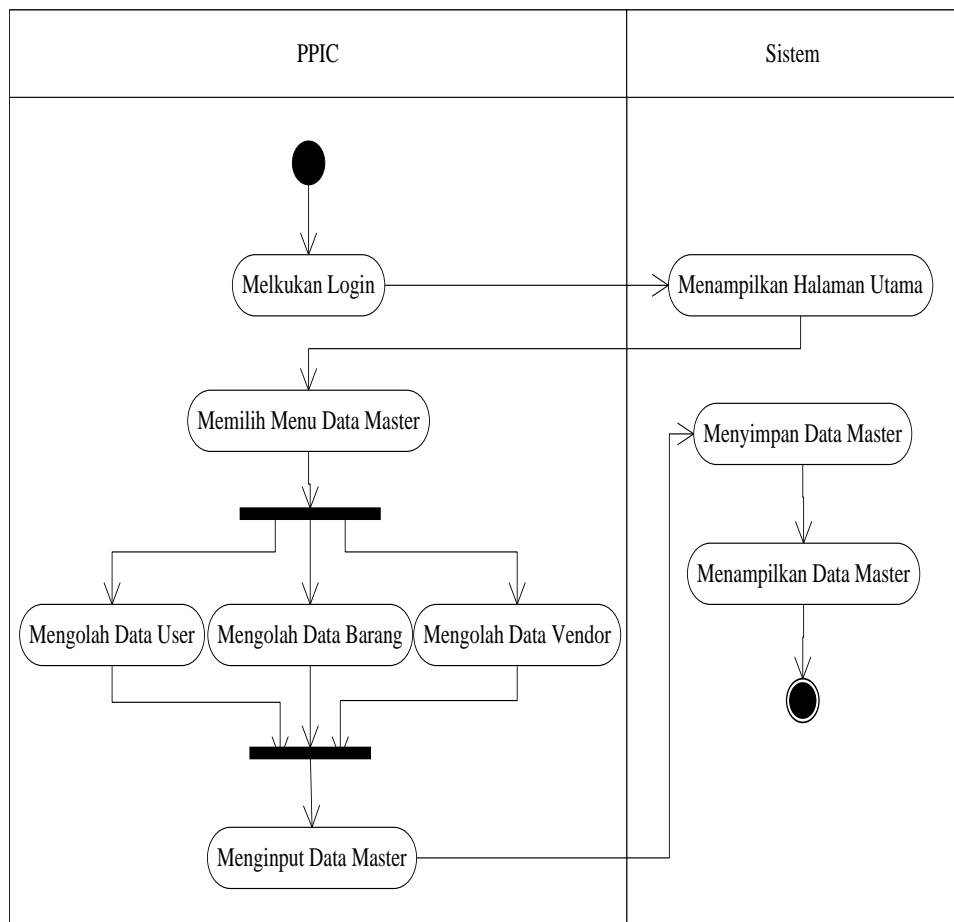
Gambar V.3 Activity Diagram Proses Login Yang Diusulkan

Sumber: Hasil Analisis Data(2016)

Pada gambar V.3 dapat dilihat sebelumnya, kita membuka aplikasi terlebih dahulu. Setelah kita membuka aplikasi, munculah halaman *login*. Kita memasukkan *username* dan *password* yang kita punya. *Username* dan *password* itu divalidasi. Jika *username* dan *password* terdaftar, maka muncul halaman utama. Jika gagal, maka akan kembali ke halaman *login*.

## 2. *Activity Diagram* Proses Mengolah Data Master

Diagram berikut merupakan diagram aktivitas yang menjelaskan tentang kegiatan mengolah data master yang dapat dilihat pada gambar V.4 sebagai berikut:

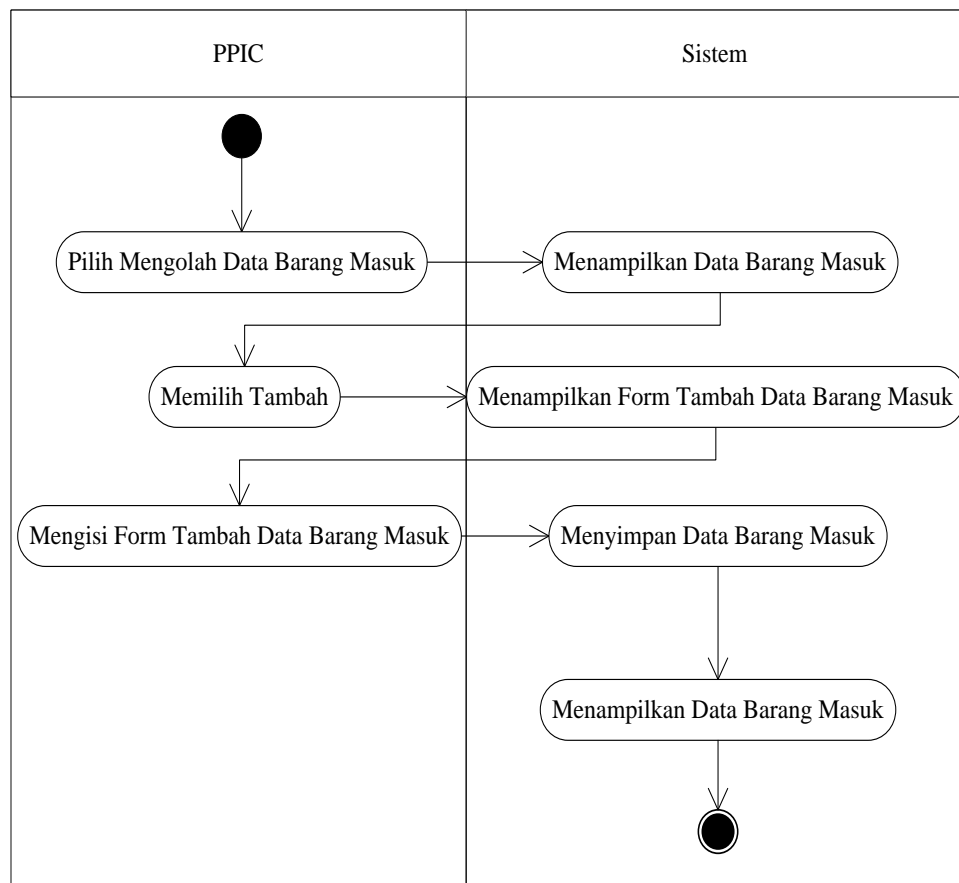


Gambar V.4 *Activity Diagram* Proses Mengolah Data Master Yang Diusulkan  
Sumber: Hasil Analisis Data(2016)

Pada gambar V.4 dapat dilihat setelah kita berhasil *login*, maka muncul halaman utama. Untuk mendata data master, kita memilih menu data master. Memilih tambah untuk menambah data master. Data master yang ditambah, disimpan untuk melihat data master yang ada pada sistem yang dibuat.

### 3. *Activity Diagram* Proses Mengolah Data Barang Masuk

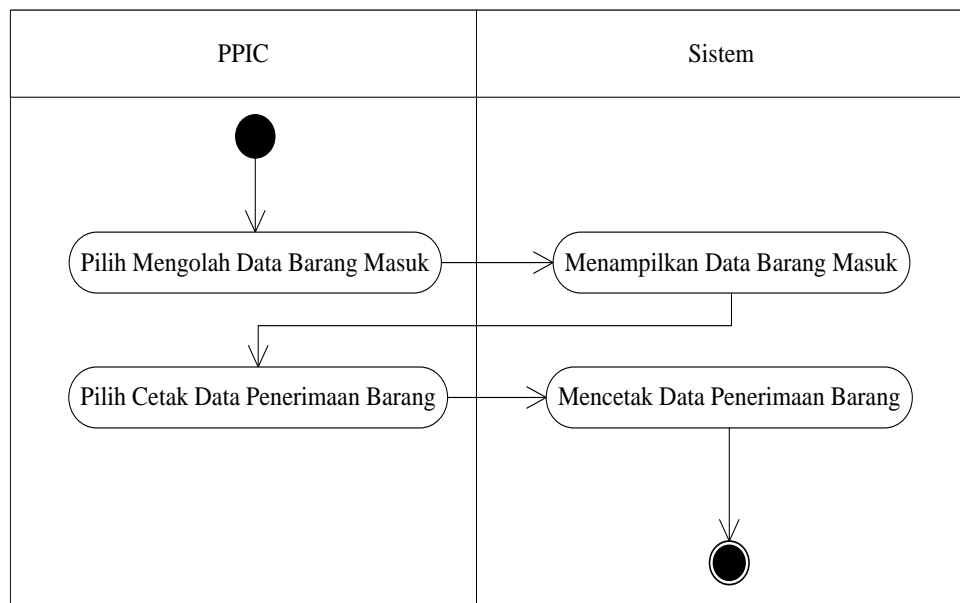
Diagram berikut merupakan diagram aktivitas yang menjelaskan tentang kegiatan mengolah data barang masuk yang dapat dilihat pada gambar V.5 sebagai berikut:



Gambar V.5 *Activity Diagram* Proses Mengolah Data Barang Masuk Yang Diusulkan  
Sumber: Hasil Analisis Data(2016)

Pada gambar V.5 dapat dilihat setelah kita melakukan *login*. Setelah berhasil *login*, maka muncul halaman menu utama. Untuk mendata barang masuk, kita memilih menu data barang masuk. Memilih tambah untuk menambah data barang masuk. Data barang masuk yang ditambah, disimpan untuk melihat data barang masuk yang ada pada sistem.

4. *Activity Diagram* Proses Membuat Laporan Penerimaan Barang  
 Diagram berikut merupakan diagram aktivitas yang menjelaskan tentang kegiatan membuat laporan penerimaan barang yang dapat dilihat pada gambar V.6 sebagai berikut:

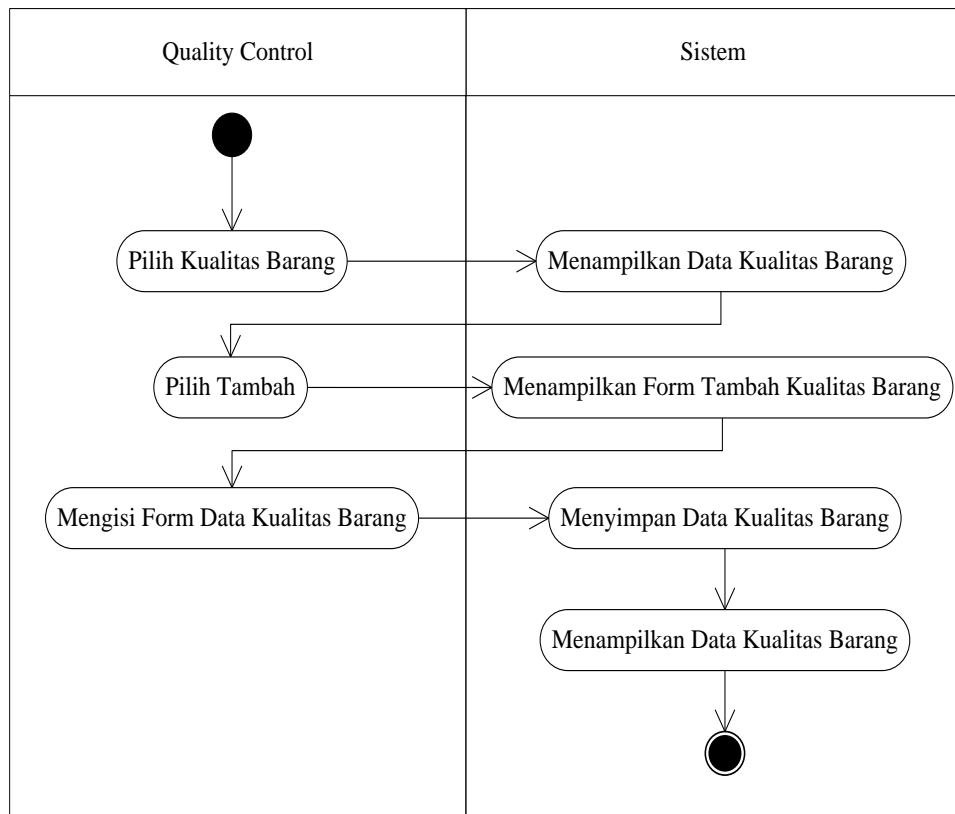


Gambar V.6 *Activity Diagram* Membuat Laporan Penerimaan Barang Yang Diusulkan  
 Sumber: Hasil Analisis Data(2016)

Pada gambar V.6 dapat dilihat setelah kita melakukan *login*. Setelah berhasil *login*, maka muncul halaman menu utama. Untuk membuat laporan penerimaan barang, kita memilih menu data barang masuk. Memilih data barang masuk. Memilih data barang masuk yang akan dicetak. Pilih cetak data barang masuk.

5. *Activity Diagram* Proses Mengolah Data Kualitas Hasil Produksi

Diagram berikut merupakan diagram aktivitas yang menjelaskan tentang kegiatan mengolah data kualitas hasil produksi yang dapat dilihat pada gambar V.7 sebagai berikut:

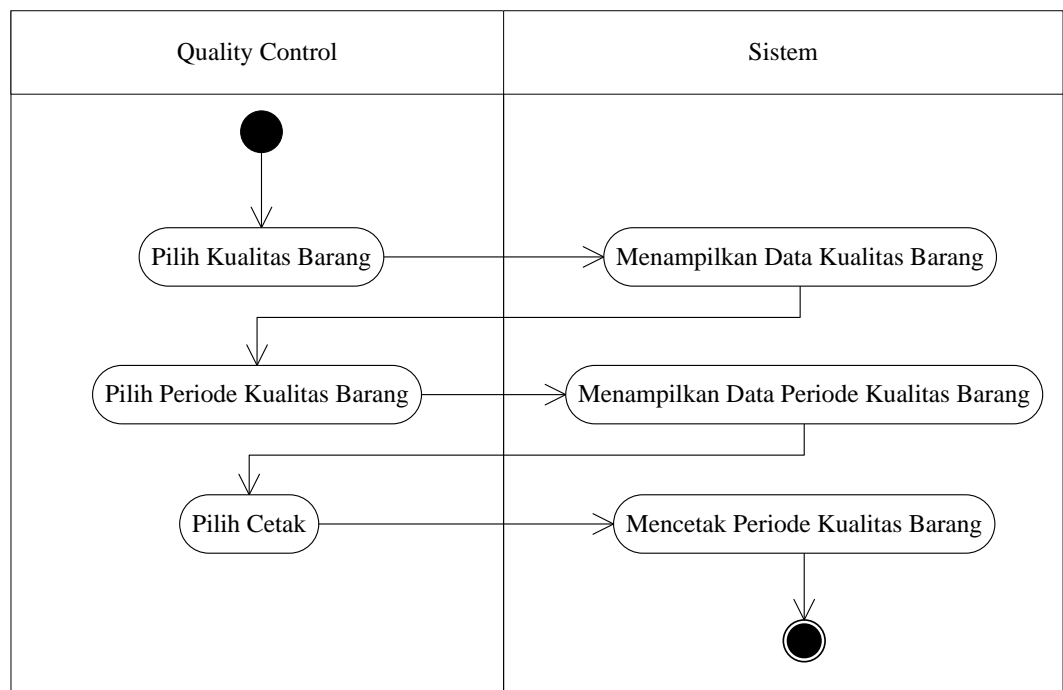


Gambar V.7 *Activity Diagram* Mengolah Data Kualitas Hasil Produksi Yang Diusulkan  
Sumber: Hasil Analisis Data(2016)

Pada gambar V.7 dapat dilihat setelah kita melakukan *login*. Setelah berhasil *login*, maka muncul halaman menu utama. Untuk mengolah data kualitas hasil produksi, kita memilih menu data kualitas hasil produksi. Memilih tambah, untuk menambah data kualitas barang hasil produksi. Mengisi *form* data kualitas barang hasil produksi dan menyimpannya.

#### 6. *Activity Diagram* Proses Mengolah Data Periodik

Diagram berikut merupakan diagram aktivitas yang menjelaskan tentang kegiatan mengolah data periodik yang dapat dilihat pada gambar V.8 sebagai berikut:



Gambar V.8 *Activity Diagram* Mengolah Data Periodik Yang Diusulkan  
Sumber: Hasil Analisis Data(2016)

Pada gambar V.8 dapat dilihat setelah kita melakukan *login*. Setelah berhasil *login*, maka muncul halaman menu utama. Untuk mengolah data hasil periodik, kita memilih menu mengolah kualitas hasil produksi. Memilih data kualitas hasil produksi yang akan di cetak. Pilih cetak pada sistem, untuk mencetak data kualitas hasil produksi.

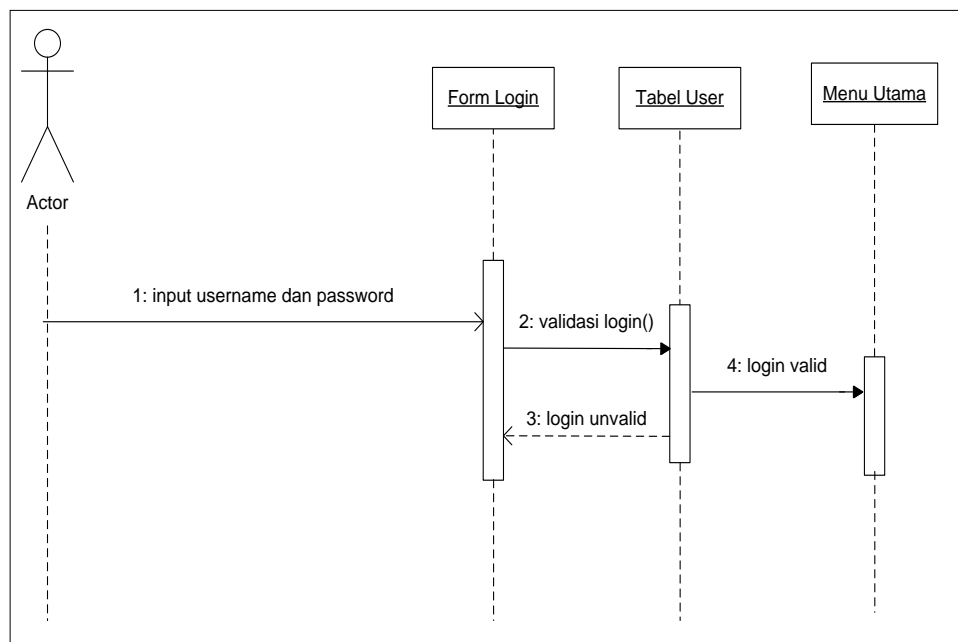
#### 5.3.3 *Sequence Diagram* Yang Diusulkan

*Sequence diagram* menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan *use case*. *Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu.

*Sequence diagram* terdiri atas dimensi *vertical*(waktu) dan dimensi *horizonlat*(*object-object* yang terkait). *Sequence diagram* digunakan untuk memodelkan pengiriman *message* antar *object*.

#### 1. *Sequence Diagram* Proses Login

*Sequence diagram login* menggambarkan proses yang terjadi dalam *login*. Prosesnya dimulai dengan masuk kedalam *form login* terlebih dahulu, kemudian selanjutnya *actor* dapat melakukan *login* dengan memasukkan *username* dan *password* untuk dapat masuk kedalam sistem. Proses *login* itu sendiri diikuti juga dengan validasi yang dilakukan oleh sistem, dimana terdapat proses verifikasi *username* dan *password*. Apabila validasi berhasil maka *actor* tersebut dapat masuk dan menggunakan sistem, seperti yang dapat dilihat pada gambar V.9 sebagai berikut:

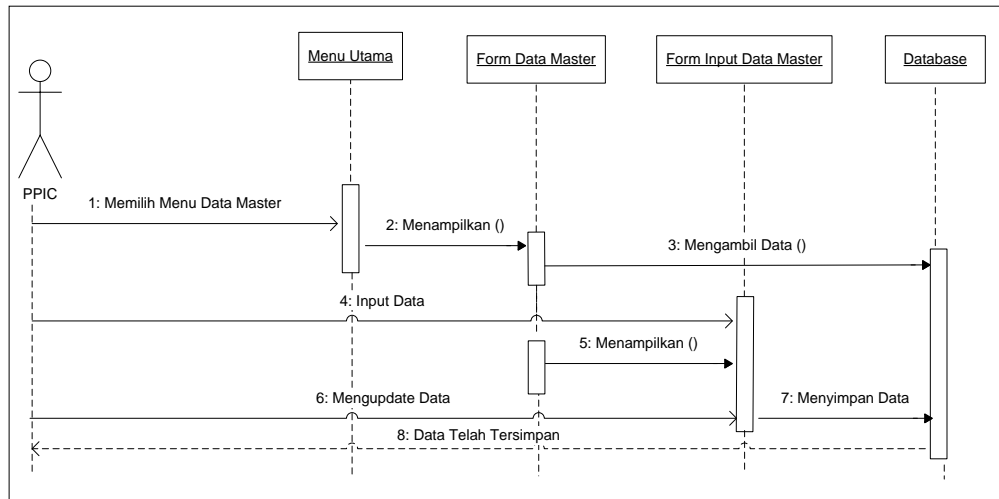


Gambar V.9 *Sequence Diagram Login* yang Diusulkan  
Sumber: Hasil Analisis Data(2016)

#### 2. *Sequence Diagram* Pendataan Data Master

*Sequence diagram* berikut yang merupakan proses pendataan data master. Diasumsikan bahwa *admin* sudah melakukan proses *login*, setelah itu *admin* akan masuk ke halaman utama dan memilih menu data master untuk

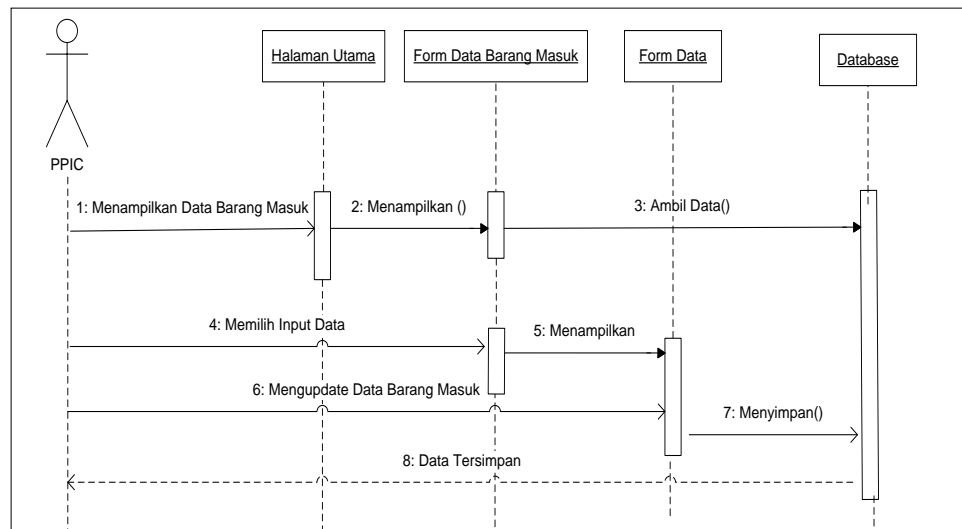
menginput data master. Jika tidak lengkap maka sistem akan menolak proses penyimpanan dan meminta input ulang, seperti yang dapat dilihat pada Gambar V.10 sebagai berikut:



Gambar V.10 *Sequence Diagram* Pendataan Data Master yang Diusulkan  
Sumber: Hasil Analisis Data(2016)

### 3. *Sequence Diagram* Pendataan Barang Masuk

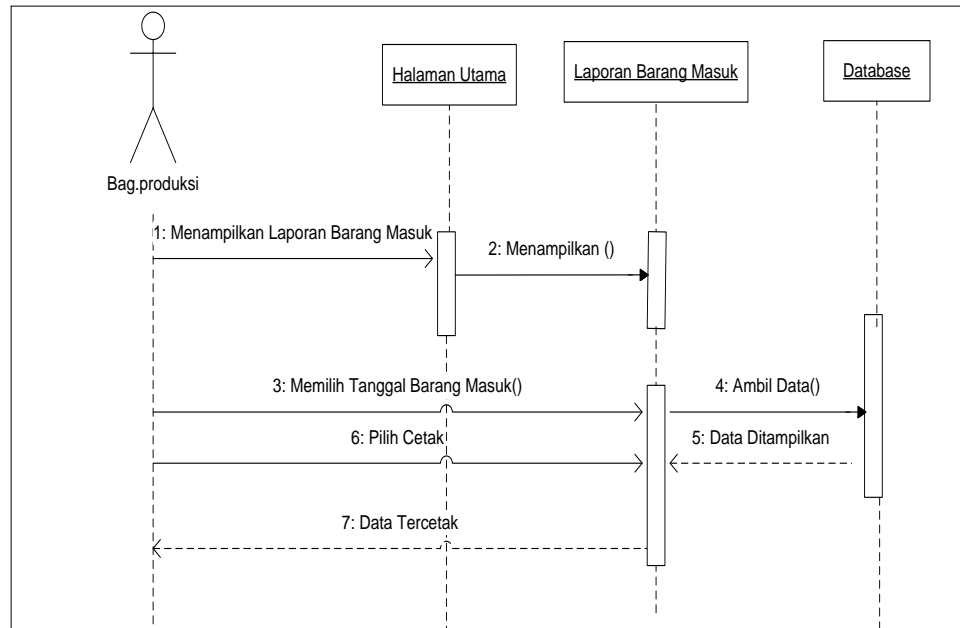
*Sequence diagram* berikut yang merupakan proses memasukkan data barang masuk. Diasumsikan bahwa PPIC sudah melakukan proses *login*, setelah itu PPIC akan masuk ke halaman utama dan memilih menu data barang masuk untuk menginput data barang masuk. Jika data tidak lengkap maka sistem akan menolak proses penyimpanan dan meminta input ulang, seperti yang dapat dilihat pada Gambar V.11 sebagai berikut:



Gambar V.11 *Sequence Diagram* Pendataan Barang Masuk Yang Diusulkan  
 Sumber: Hasil Analisis Data(2016)

#### 4. *Sequence diagram* Laporan Penerimaan Barang

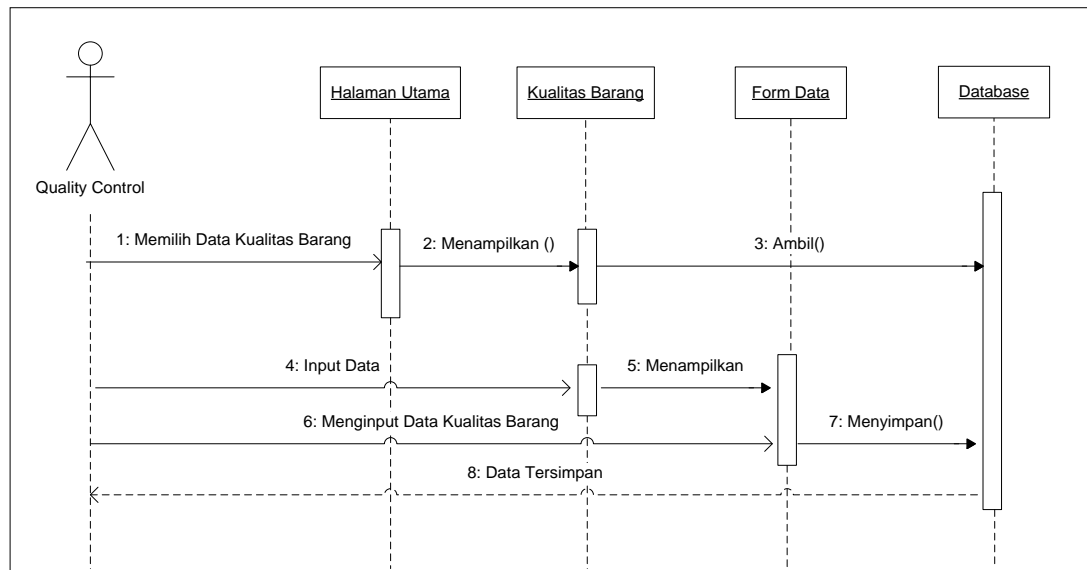
*Sequence diagram* berikut yang merupakan proses membuat laporan penerimaan barang. Diasumsikan bahwa PPIC sudah melakukan proses *login*, setelah itu PPIC akan masuk ke halaman utama dan memilih menu laporan penerimaan barang. PPIC akan memilih waktu penerimaan barang yang ingin dilihat. Memilih cetak, bila ingin mencetak data penerimaan barang. Seperti yang dapat dilihat pada Gambar V.12 sebagai berikut:



Gambar V.12 *Sequence Diagram* Laporan Penerimaan Barang Yang Diusulkan  
Sumber: Hasil Analisis Data(2016)

#### 5. *Sequence diagram* Mengolah Data Kualitas Hasil Produksi

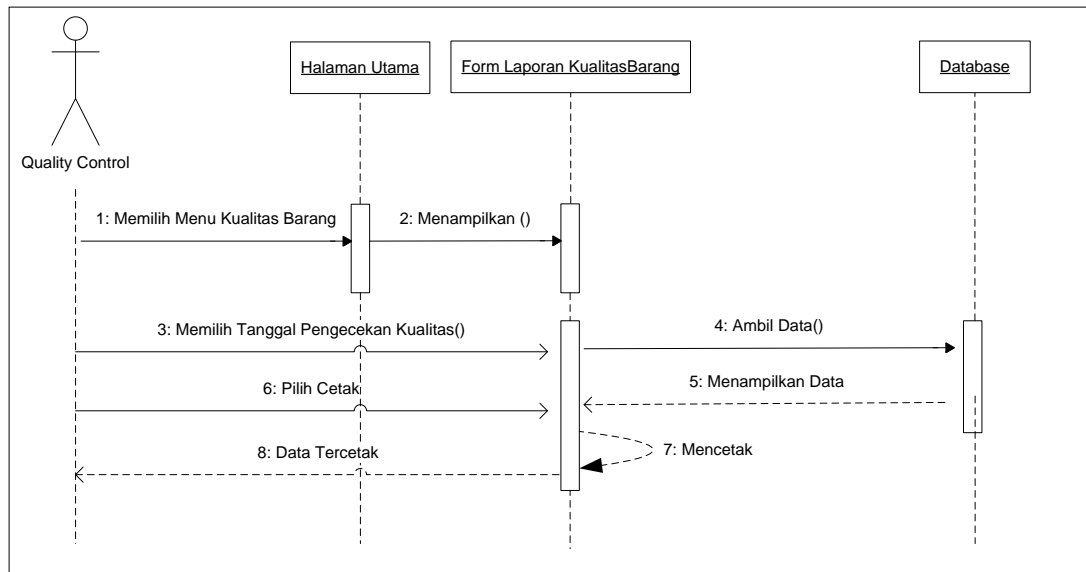
*Sequence diagram* berikut yang merupakan proses pengolahan data kualitas hasil produksi. Diasumsikan bahwa *quality control* sudah melakukan proses *login*, setelah itu *quality control* akan masuk ke halaman utama dan memilih menu kualitas barang. *Quality control* akan menginput kode barang dan memilih jumlah barang yang di proses. Memilih tambah item untuk memberi keterangan pada barang hasil produksi. Seperti yang dapat dilihat pada Gambar V.13 sebagai berikut:



Gambar V.13 *Sequence Diagram* Mengolah Data Kualitas Hasil Produksi Yang Diusulkan  
Sumber: Hasil Analisis Data(2016)

#### 6. *Sequence diagram* Mengolah Data Periodik

*Sequence diagram* berikut yang merupakan proses pengolahan data periodik. Diasumsikan bahwa *quality control* sudah melakukan proses *login*, setelah itu *quality control* akan masuk ke halaman utama dan memilih menu laporan kualitas barang. *Quality control* akan menginput tanggal awal dan akhir dari kualitas barang yang ingin dilihat. Memilih cetak untuk mencetak data kualitas barang yang diinginkan. Seperti yang dapat dilihat pada Gambar V.14 sebagai berikut:



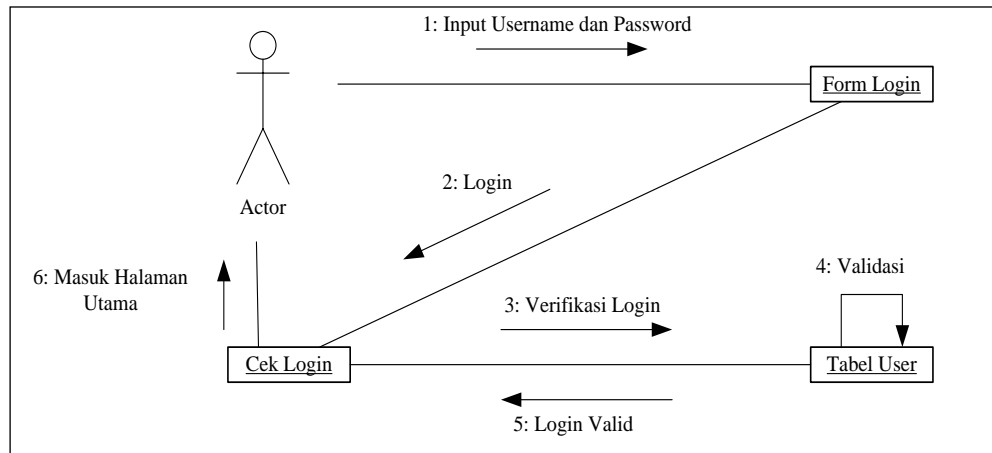
Gambar V.14 *Sequence Diagram* Mengolah Laporan Periodik Yang Diusulkan  
Sumber: Hasil Analisis Data(2016)

### 5.3.4 *Colaboration Diagram* yang Diusulkan

*Colaboration diagram* menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, dimana *message* dari level tertinggi memiliki nomor satu.

#### 1. *Colaboration Diagram* Proses Login

*Colaboration diagram login* ini menunjukkan interaksi antara *actor* dengan sistem dalam proses *login*. Proses *login*, admin harus meng-*input* *username* dan *password* terlebih dahulu untuk masuk kedalam sistem, seperti yang dapat dilihat pada Gambar V.15 sebagai berikut:

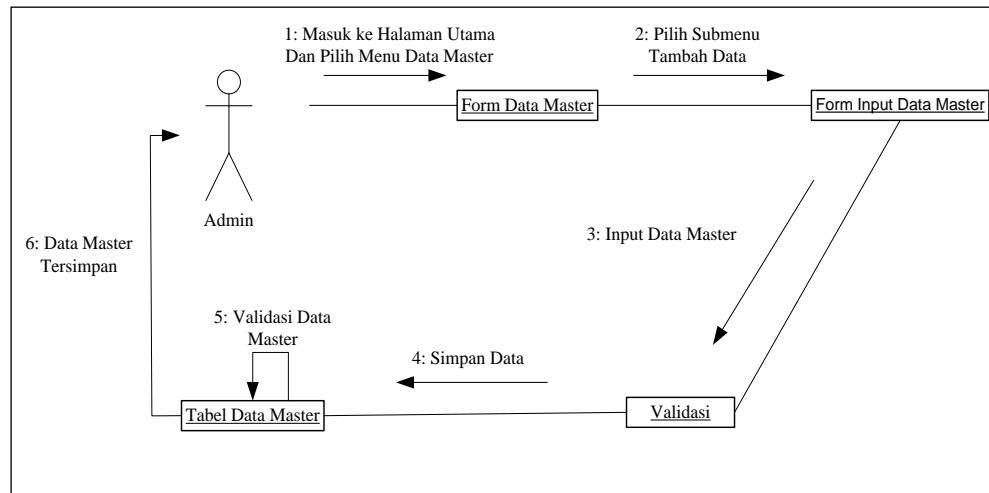


Gambar V.15 *Collaboration Diagram* Proses Login Yang Diusulkan  
Sumber: Hasil Analisis Data(2016)

*Collaboration diagram* proses login diatas menggambarkan actor melakukan login ke dalam sistem. Sistem meminta untuk *input username* dan *password*. Apabila *username* dan *password* benar, maka sistem memvalidasi proses login tersebut dan dapat masuk ke halaman menu utama. Apabila salah satu actor salah memasukkan *username* dan *passwords*, maka sistem akan memberikan pesan *error* dan meminta untuk memasukkan ulang *username* dan *password*.

## 2. *Colaboration Diagram* Pendataan Data Master

*Colaboration diagram* pendataan data master ini menunjukkan interaksi antara sistem dengan admin dalam proses pendataan data master, seperti yang dapat dilihat pada Gambar V.16 sebagai berikut:

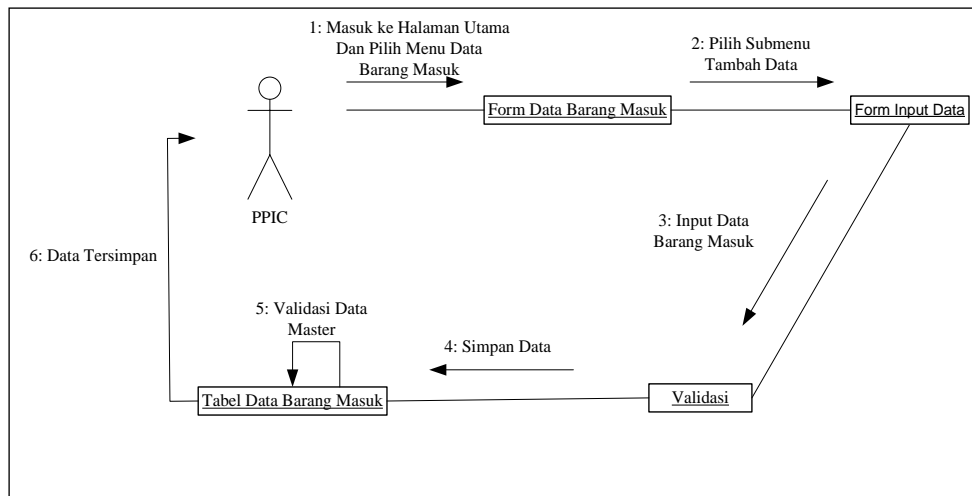


Gambar V.16 *Collaboration Diagram* Pendataan Data Master Yang Diusulkan  
 Sumber: Hasil Analisis Data(2016)

*Collaboration diagram* pendataan data master ini menggambarkan *admin* melakukan *input* di dalam sistem. Sebelum melakukan peng-*input-an*, *admin* diwajibkan untuk melakukan *login* terlebih dahulu. Setelah melakukan *login*, maka dapat memilih menu data master. Lalu memilih submenu tambah data master dan melakukan proses tambah data. Apabila data yang ditambah sesuai dengan kriteria yang sudah ditentukan oleh sistem, maka sistem akan menyimpan data barang tersebut dan menampilkan data yang sudah disimpan didalam basis data.

### 3. *Collaboration Diagram* Mengolah Data Barang Masuk

*Collaboration diagram* mengolah data barang masuk ini menunjukkan interaksi antara sistem dengan PPIC dalam proses pengolahan data barang masuk, seperti yang dapat dilihat pada Gambar V.17 sebagai berikut:

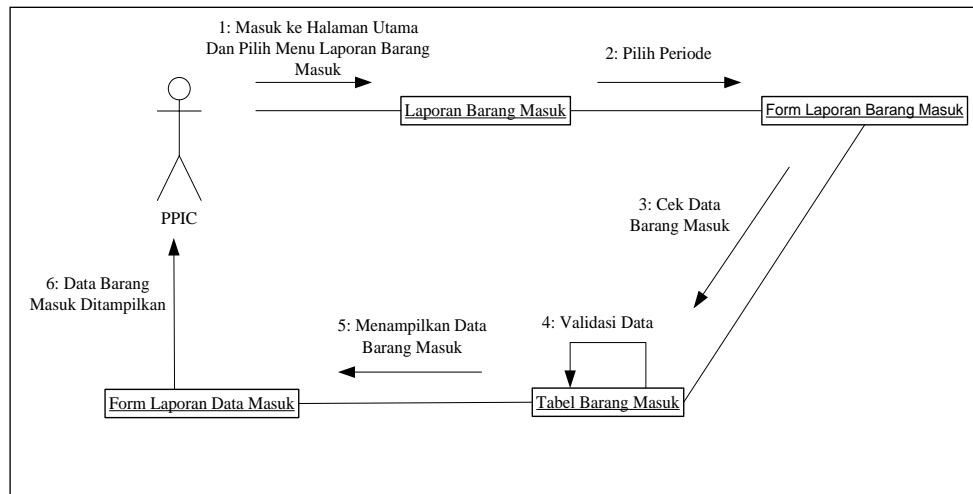


Gambar V.17 *Collaboration Diagram* Mengolah Data Barang Masuk Yang Diusulkan  
Sumber: Hasil Analisis Data(2016)

*Collaboration diagram* mengolah data barang masuk ini menggambarkan PPIC melakukan *input* didalam sistem. Sebelum melakukan peng-*input-an*, PPIC diwajibkan untuk melakukan *login* terlebih dahulu. Setelah proses *login* berhasil, maka dapat memilih menu data barang masuk. Lalu memilih submenu tambah untuk melakukan proses tambah data. Apabila data yang ditambah sesuai dengan kriteria yang sudah ditentukan oleh sistem, maka sistem akan menyimpan data tersebut dan menampilkan data yang sudah disimpan didalam basis data.

#### 4. *Collaboration Diagram* Laporan Penerimaan Barang

*Collaboration diagram* laporan penerimaan barang ini menunjukkan interaksi antara sistem dengan PPIC dalam proses laporan penerimaan barang, seperti yang dapat dilihat pada Gambar V.18 sebagai berikut:

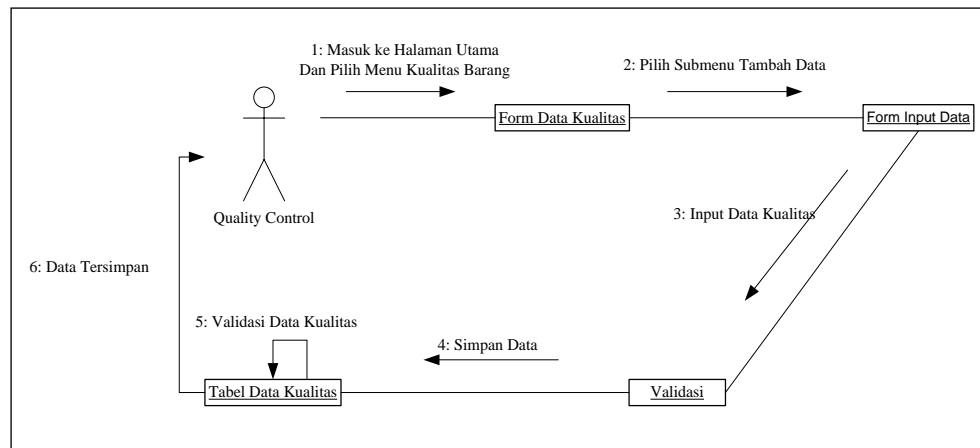


Gambar V.18 *Collaboration Diagram* Laporan Barang Masuk Yang Diusulkan  
Sumber: Hasil Analisis Data(2016)

*Collaboration diagram* laporan barang masuk ini menggambarkan PPIC melakukan *input* didalam sistem. Sebelum melakukan *peng-input-an*, PPIC diwajibkan untuk melakukan *login* terlebih dahulu. Setelah proses *login* berhasil, maka dapat memilih menu laporan barang masuk. Lalu memilih periode untuk melihat data barang masuk. Setelah memilih periode, maka sistem akan menampilkan data barang masuk dari periode yang dipilih.

##### 5. *Colaboration Diagram* Mengolah Hasil Produksi

*Colaboration diagram* mengolah hasil produksi ini menunjukkan interaksi antara sistem dengan *quality control* dalam proses laporan penerimaan barang, seperti yang dapat dilihat pada Gambar V.19 sebagai berikut:

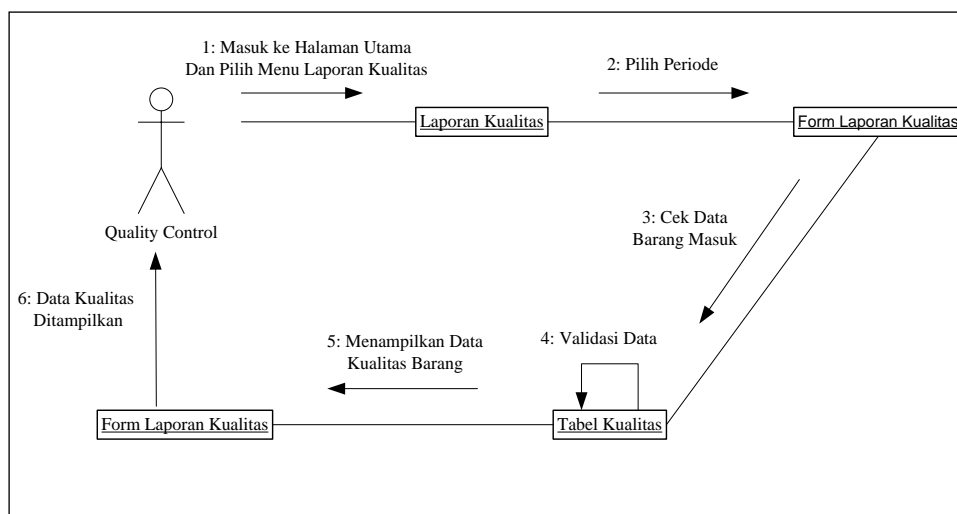


Gambar V.19 *Collaboration Diagram* Mengolah Hasil Produksi Yang Diusulkan  
Sumber: Hasil Analisis Data(2016)

*Collaboration diagram* mengolah hasil produksi ini menggambarkan *quality control* melakukan *input* didalam sistem. Sebelum melakukan peng-*input*-an, *quality* diwajibkan untuk melakukan *login* terlebih dahulu. Setelah proses *login* berhasil, maka dapat memilih menu kualitas. Lalu memilih submenu tambah untuk melakukan proses tambah data. Apabila data yang ditambah sesuai dengan kriteria yang sudah ditentukan oleh sistem, maka sistem akan menyimpan data tersebut dan menampilkan data yang sudah disimpan didalam basis data.

#### 6. *Colaboration Diagram* Mengolah Data Periodik

*Colaboration diagram* mengolah mengolah data periodik ini menunjukkan interaksi antara sistem dengan *quality control* dalam proses pengolahan data periodik, seperti yang dapat dilihat pada Gambar V.20 sebagai berikut:



Gambar V.20 Collaboration Diagram Mengolah Data Periodik Yang Diusulkan  
Sumber: Hasil Analisis Data(2016)

*Collaboration diagram* mengolah data periodik ini menggambarkan *quality control* melakukan *input* didalam sistem. Sebelum melakukan *peng-input-an*, *quality control* diwajibkan untuk melakukan *login* terlebih dahulu. Setelah proses *login* berhasil, maka dapat memilih menu laporan kualitas. Lalu memilih periode untuk melihat data kualitas yang diinginkan. Setelah memilih periode, maka sistem akan menampilkan data kualitas dari periode yang dipilih.

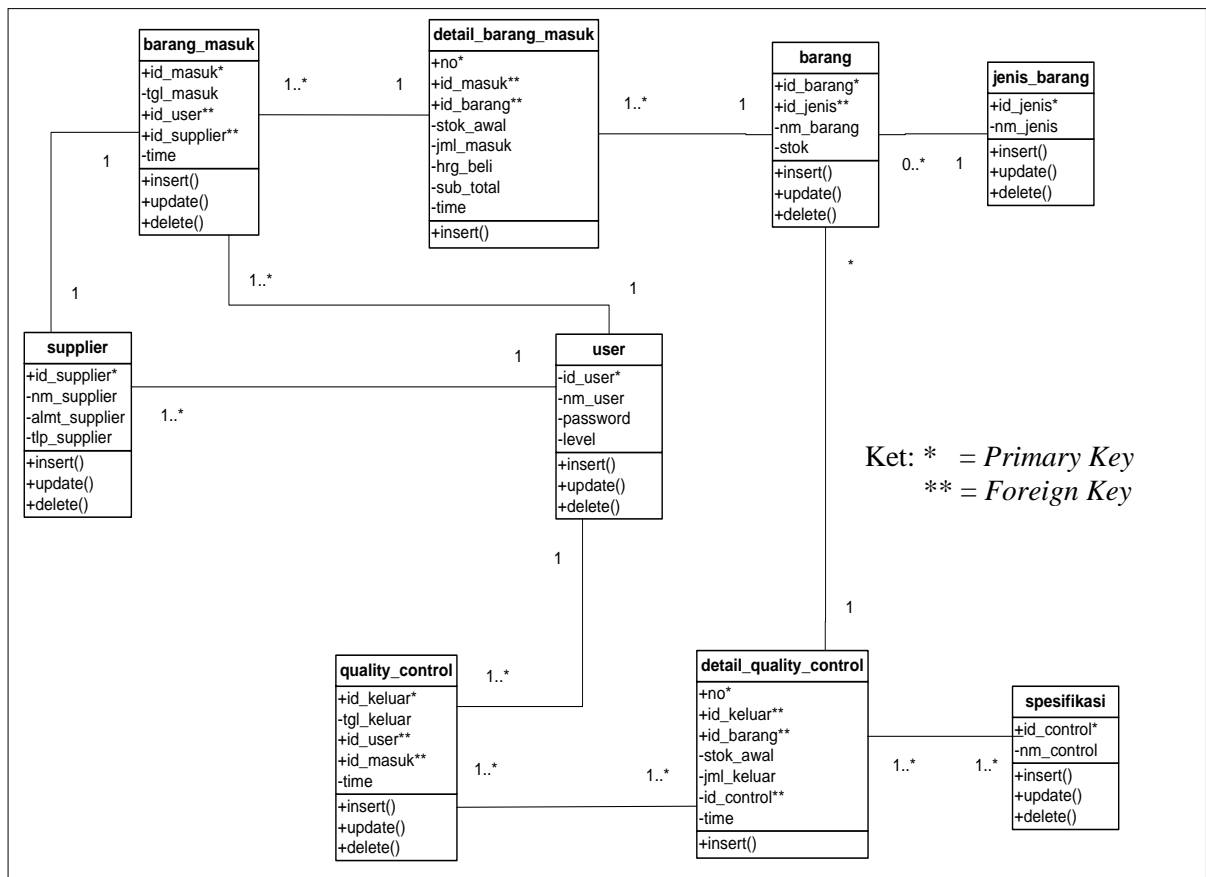
## 5.4 Perancangan Data

Pemodelan data adalah metode yang digunakan untuk menentukan dan menganalisis persyaratan data yang diperlukan untuk mendukung proses bisnis suatu organisasi. Data yang dibutuhkan adalah dicatat sebagai data model konseptual dengan definisi data yang terkait.

### 5.4.1 Class Diagram yang Diusulkan

*Class diagram* membantu dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem. Selama proses analisis, *class diagram*

memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama tahap desain, *class diagram* berperan dalam menggambarkan struktur dari semua kelas yang membentuk arsitektur yang dibuat, seperti yang dapat dilihat pada Gambar V.21 sebagai berikut:



Gambar V.21 *Class Diagram* Yang Diusulkan  
Sumber: Hasil Analisis(2016)

#### 5.4.2 Kamus Data

Kamus data adalah suatu daftar data elemen yang terorganisir dengan definisi yang tetap dan sesuai dengan sistem, sehingga *user* dan analis sistem mempunyai pengertian yang sama tentang *input*, *output*, dan komponen *data store*. Pada tahap perancangan sistem, kamus data digunakan untuk merancang *input*, laporan dan basis data. Berikut adalah kamus data sistem informasi laporan kualitas *plating*:

## 1. Spesifikasi Tabel Master Barang

Nama Tabel : barang

Fungsi : Untuk menyimpan data barang

Tipe : File Data Master

Tabel V.9 Spesifikasi Tabel Barang

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	ID Barang	id_barang	Varchar	16	<i>Primary key</i>
2	ID Jenis	id_jenis	Varchar	10	
3	Nama Barang	nm_barang	Varchar	30	
4	Stok	Stok	Int	11	

Sumber: Hasil Analisis(2016)

## 2. Spesifikasi Tabel Transaksi Barang Masuk

Nama Tabel : barang\_masuk

Fungsi : Untuk menyimpan data baarang masuk

Tipe : File Data Transaksi

Tabel V.10 Spesifikasi Tabel Barang Masuk

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	ID Barang Masuk	id_masuk	Varchar	16	<i>Primary key</i>
2	Tanggal Masuk Barang	tgl_masuk	Date		
3	ID User	id_user	Varchar	16	
4	ID Supplier	id_supplier	Varchar	16	<i>Foreign Key</i>
5	Waktu	Time	timestamp		

Sumber: Hasil Analisis(2016)

3. Spesifikasi Tabel Detail *Quality Control*

Nama Tabel : detail\_quality\_control

Fungsi : Untuk menyimpan data detail *quality control*

Tipe : File Data Transaksi

Tabel V.11 Spesifikasi Tabel Detail *Quality Control*

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	Nomor Detail	No	Int	11	
2	ID Keluar	id_keluar	Varchar	16	<i>Primary key</i>
3	ID Barang	id_barang	Varchar	16	
4	Stok Awal Barang	stok_awal	Int	11	
5	Jumlah Barang Keluar	jml_keluar	Int	11	
6	ID Control	id_control	Int	3	
7	Waktu	Time	timestamp		

Sumber: Hasil Analisis(2016)

#### 4. Spesifikasi Tabel Jenis Barang

Nama Tabel : jenis\_barang

Fungsi : Untuk menyimpan data jenis barang

Tipe : File Data Transaksi

Tabel V.12 Spesifikasi Tabel Jenis Barang

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	ID Jenis	id_jenis	Varchar	10	<i>Primary key</i>
2	Nama Jenis	nm_jenis	Varchar	30	

Sumber: Hasil Analisis(2016)

#### 5. Spesifikasi Tabel *Quality Control*

Nama Tabel : quality\_control

Fungsi : Untuk menyimpan data *quality control*

Tipe : File Data Transaksi

Tabel V.13 Spesifikasi Tabel Detail *Quality Control*

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	ID Keluar	id_keluar	Varchar	30	<i>Primary key</i>
2	Tanggal Keluar	tgl_keluar	Date		
3	ID User	id_user	Varchar	16	
4	ID Masuk	id_masuk	Varchar	30	
5	Waktu	Time	timestamp		

Sumber: Hasil Analisis(2016)

## 6. Spesifikasi Tabel Spesifikasi

Nama Tabel : spesifikasi

Fungsi : Untuk menyimpan data spesifikasi barang *quality control*

Tipe : File Data Master

Tabel V.14 Spesifikasi Tabel Spesifikasi

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	ID Control	id_control	Varchar	5	<i>Primary key</i>
2	Nama Control	nm_control	Varchar	30	

Sumber: Hasil Analisis(2016)

## 7. Spesifikasi Tabel Supplier

Nama Tabel : supplier

Fungsi : Untuk menyimpan data supplier

Tipe : File Data Master

Tabel V.15 Spesifikasi Tabel Supplier

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	ID Supplier	id_supplier	varchar	16	<i>Primary key</i>
2	Nama Supplier	nm_supplier	varchar	30	
3	Alamat Supplier	almt_supplier	varchar	100	
4	Nomor Telepon Supplier	telp_supplier	varchar	12	

Sumber: Hasil Analisis(2016)

## 8. Spesifikasi Tabel User

Nama Tabel : user

Fungsi : Untuk menyimpan data user

Tipe : File Data Master

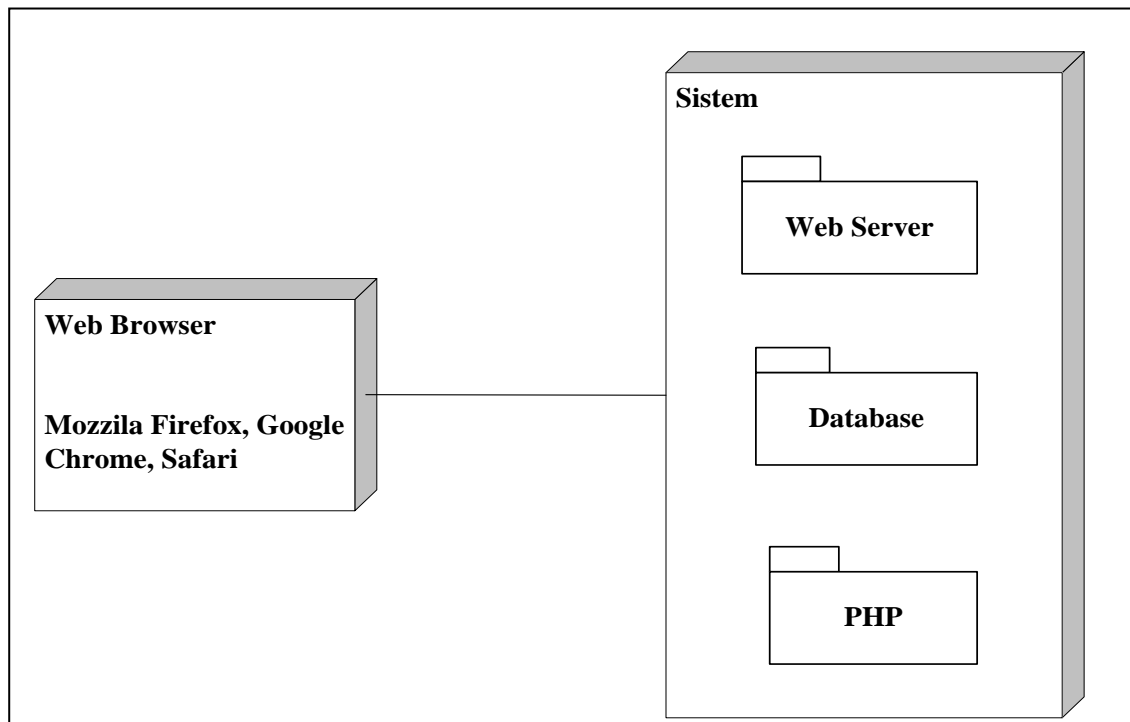
Tabel V.16 Spesifikasi Tabel User

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	ID User	id_user	Varchar	16	<i>Primary key</i>
2	Nama User	nm_user	Varchar	25	
3	Password	password	Varchar	15	
4	Level	Level	Varchar	15	

Sumber: Hasil Analisis(2016)

### 5.4.3 *Deployment Diagram* Yang Diusulkan

*Deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Seperti yang dapat dilihat pada gambar V.22 sebagai berikut:



Gambar V.22 *Deployment Diagram* Yang Diusulkan  
Sumber: Hasil Analisis Data (2016)

*Deployment diagram* di atas menggambarkan pembuatan aplikasi manajemen aset berbasis *web* yang memanfaatkan PHP sebagai desain sistemnya. Berikut adalah penjelasan *deployment diagram* aplikasi manajemen aset berbasis *web*:

1. PHP

PHP adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah aplikasi berbasis web dan bisa digunakan bersamaan dengan HTML.

2. *Database*

*Database* adalah program komputer yang menyediakan layanan data lainnya ke komputer atau program komputer, seperti yang ditetapkan oleh

model *client-server*. Istilah ini juga merujuk kepada sebuah komputer yang didedikasikan untuk menjalankan program *server database*.

3. *Web Server*

Suatu program sekaligus mesin yang menjalankan program komputer yang mengerti protokol HTTP dan dapat menanggapi permintaan-permintaan dari *web browser*.

4. *Web Browser*

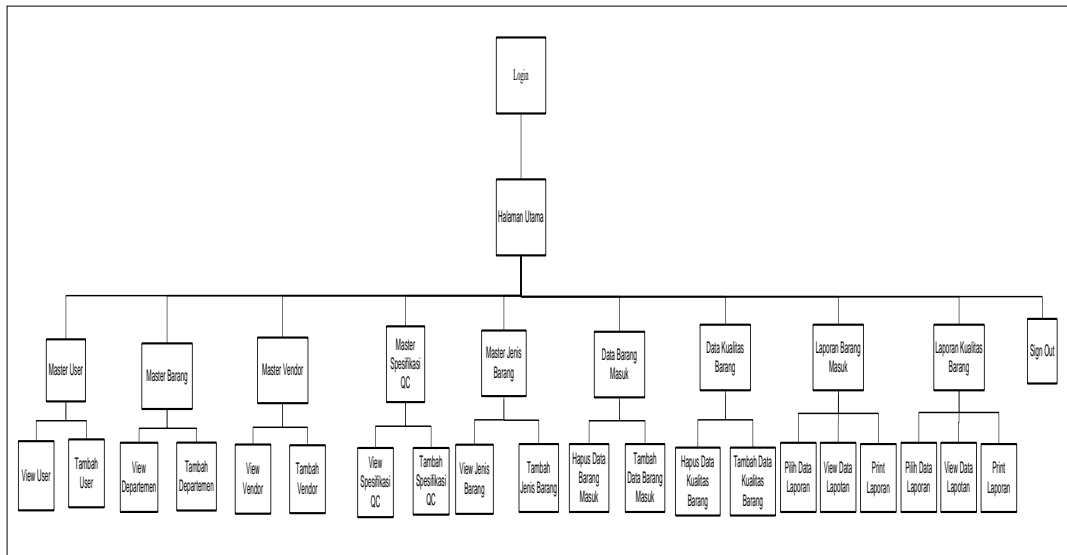
Dalam Bahasa Indonesia *web browser* memiliki arti sebagai penjelajah *web*. Fungsi *web browser* itu sendiri adalah untuk menampilkan dan melakukan interaksi dengan dokumen-dokumen yang disediakan oleh *web server*.

## 5.5 Perancangan Sistem

Perancangan sistem merupakan suatu kegiatan pengembangan serta perbaikan terhadap sebuah sistem yang berjalan. Pada tahap ini dilakukan upaya memperbaiki sistem atau membangun sistem dengan memanfaatkan teknologi baru untuk mengurangi dan mengatasi berbagai permasalahan yang telah terjadi pada sistem yang lama sehingga dapat meningkatkan efektifitas dan efisiensi.

### 5.5.1 *Hierarchy plus Input-Process-Output (HIPO)*

Pembuatan aplikasi dilakukan menggunakan bahasa pemrograman PHP dan MySQL sebagai perangkat lunak basis data. Berikut adalah struktur menu hirarki program yang digambarkan dalam *Hierarchy plus Input- Process-Output (HIPO)*:



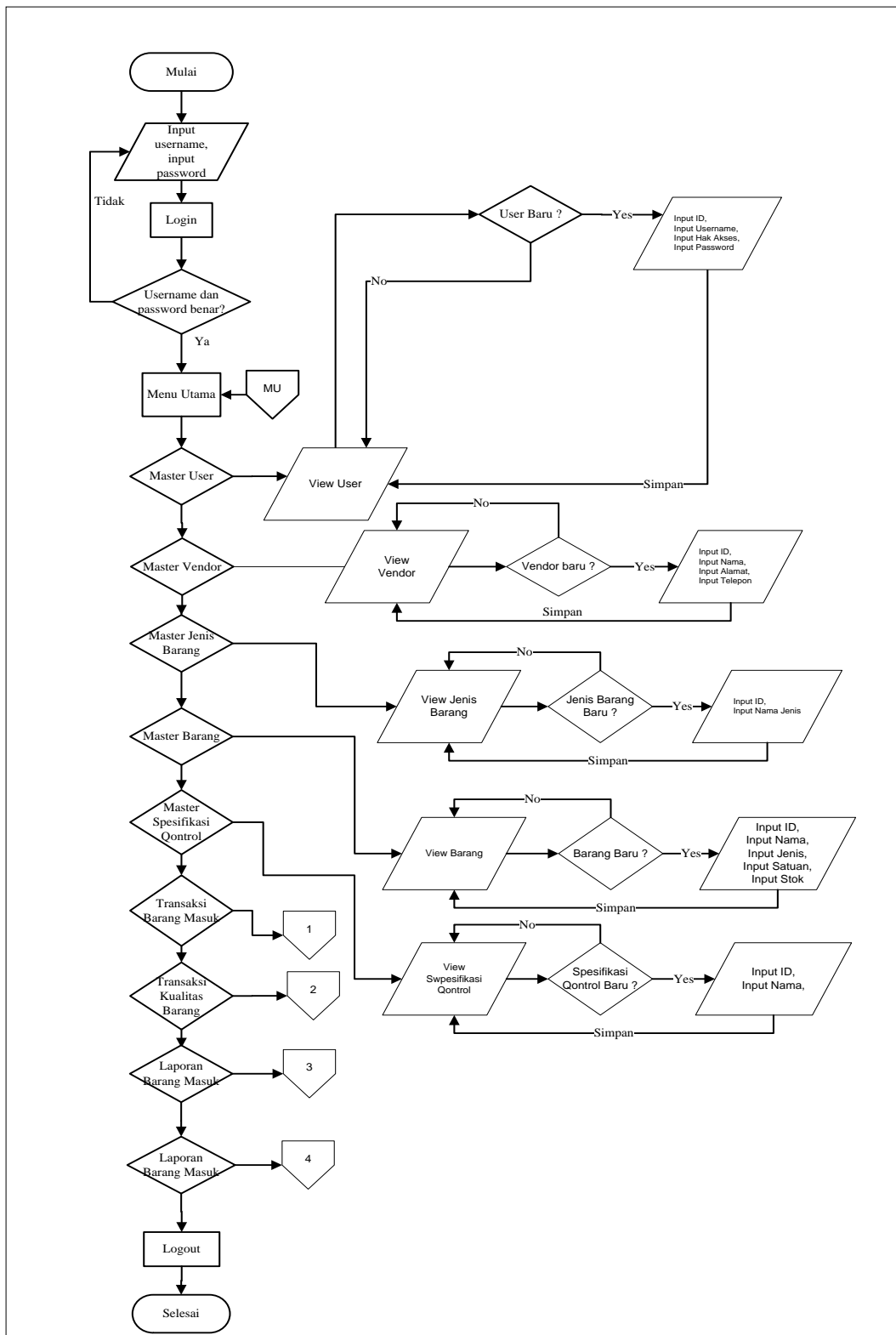
Gambar V.23 *Hirarchy plus Input Proses Output(HIPO)*

Sumber: Hasil Analisis Data (2016)

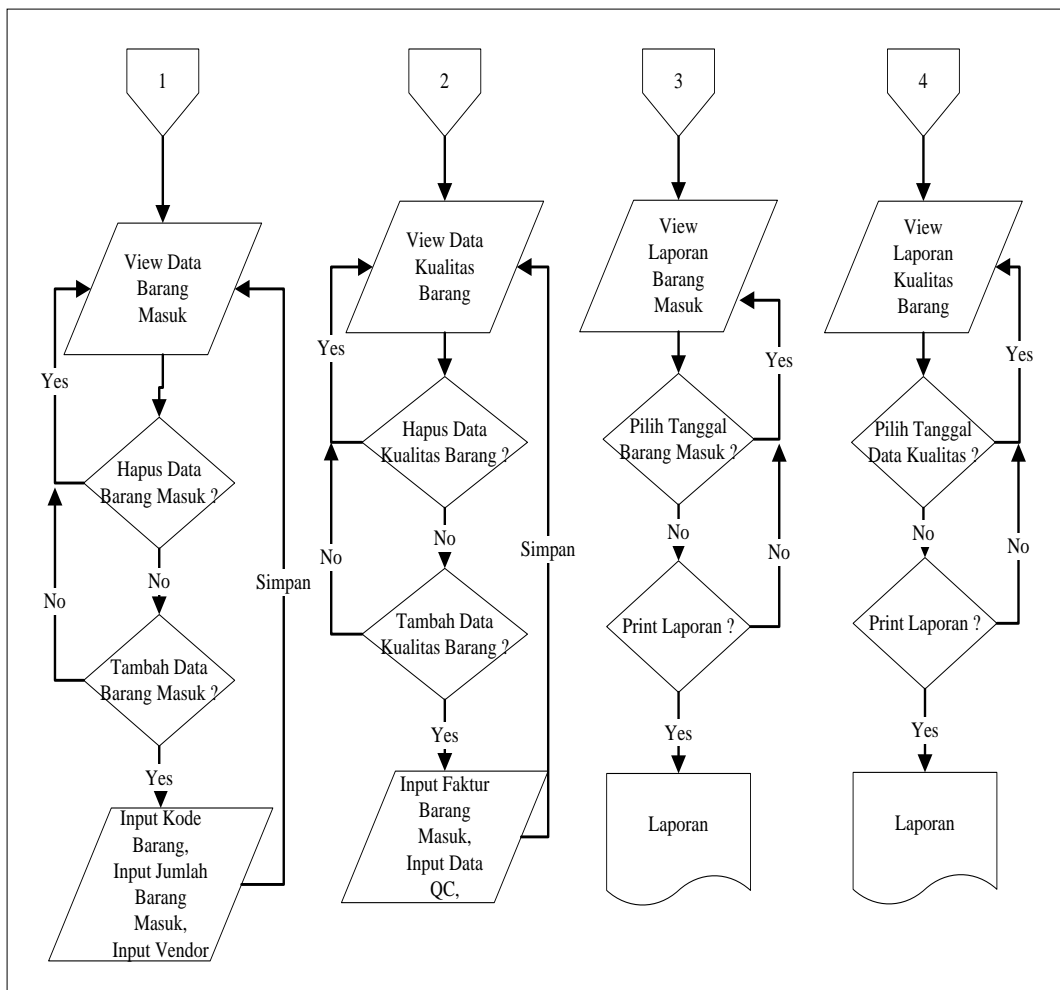
### 5.5.2 *Flowchart Program*

*Flowchart* program menggambarkan logika ataupun alur dari suatu program dengan segala percabangan, keadaan dan kemungkinannya. Selain itu *Flowchart* program menggambarkan proses dari suatu program dari awal hingga akhir dengan menggunakan konsep logika peta arus.

Dalam aplikasi program yang diusulkan ini keseluruhan menu dapat diakses oleh *admin* di PT Surya Sunjaya Industri. Berikut adalah *flowchart* program sistem informasi laporan kualitas *plating* secara keseluruhan:



Gambar V.24 Flowchart Program Yang Diusulkan  
Sumber: Hasil Analisis Data (2016)



Gambar V.25 Flowchart Program Yang Diusulkan(Lanjutan)

Sumber: Hasil Analisis Data (2016)

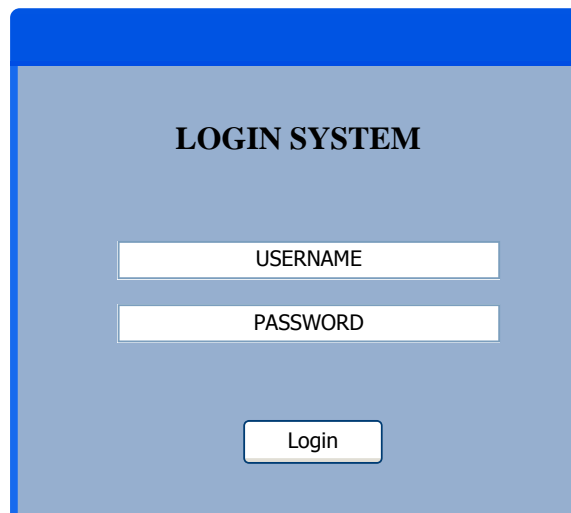
Proses 1 merupakan proses yang dilakukan untuk melihat data barang masuk, menambahkan data barang masuk, dan untuk melakukan penghapusan data pada data barang masuk. Proses 2 merupakan proses yang dilakukan pada data kualitas barang bertujuan untuk melihat data barang masuk, menghapus data barang masuk, dan menambahkan data barang masuk. Proses 3 dapat dilakukan untuk membuat laporan barang masuk. Sedangkan proses 4 untuk memuat laporan kualitas barang yang terdapat pada perusahaan.

### 5.5.3 Perancangan *Interface* Sistem Informasi Laporan Kualitas *Plating*

Perancangan *interface* merupakan tahapan untuk membuat tampilan atau desain dari sistem informasi laporan kualitas *plating*. Rancangan tampilan yang dibuat meliputi rancangan *input* dan rancangan *output* dari sistem yang akan dibuat. Berikut ini merupakan *interface* dari aplikasi laporan kualitas *plating*:

1. Rancangan *Form Login*

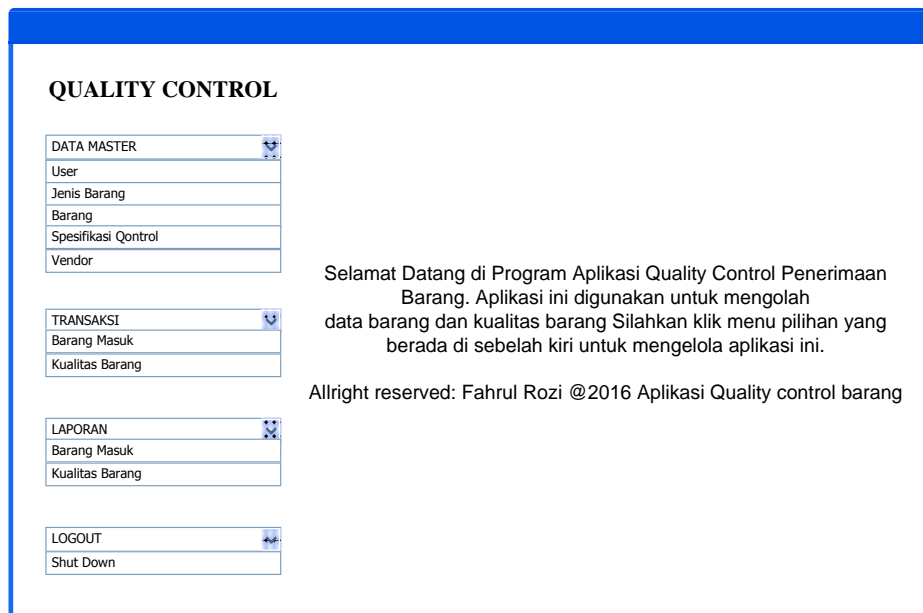
*Form Login* digunakan untuk membedakan hak akses pengguna. Melalui *Form Login* ini pengguna harus memasukkan *username* dan *password* yang nantinya dicek kedalam sistem. Pada Gambar V.26 adalah rancangan tampilan form login.

The image shows a simple login form with a blue border. At the top center, the text 'LOGIN SYSTEM' is displayed in bold. Below this, there are two white rectangular input fields. The first field is labeled 'USERNAME' and the second is labeled 'PASSWORD'. At the bottom center of the form, there is a white button with the text 'Login'.

Gambar V.26 Rancangan *Form Login*  
Sumber: Hasil Analisis Data (2016)

2. Rancangan Halaman Utama

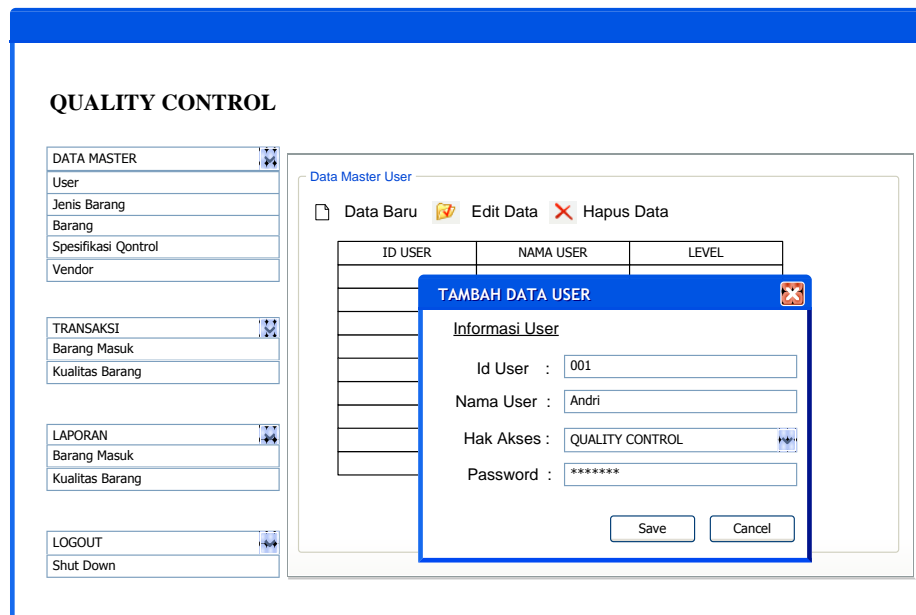
Menu utama ini dirancang untuk memudahkan *user* dalam mengakses aplikasi dengan pilihan menu-menu yang sudah disediakan sesuai kebutuhan.



Gambar V.27 Rancangan Menu Utama  
Sumber: Hasil Analisis Data (2016)

### 3. Menu Data User

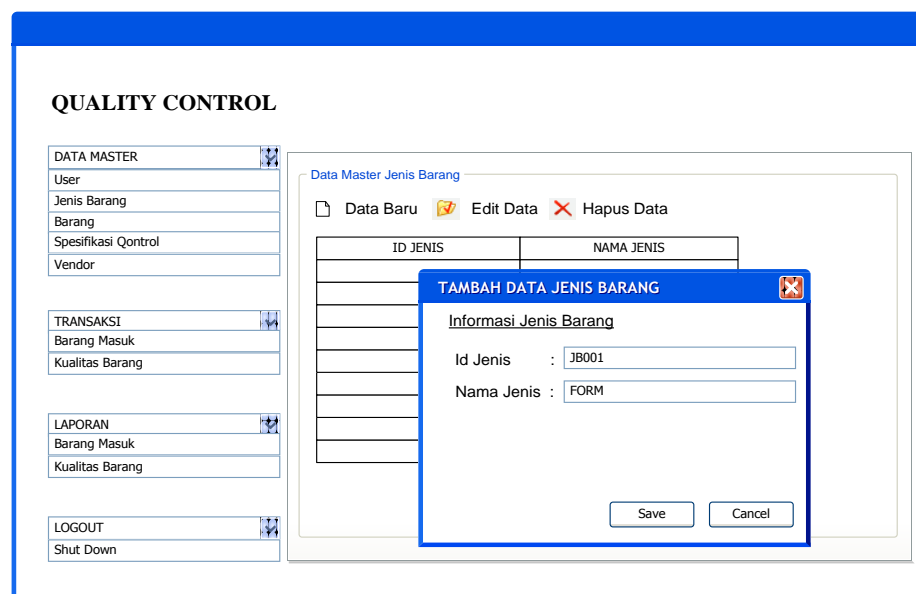
Menu data user ini merupakan salah satu sub menu dari menu data master, berfungsi untuk meng-*edit*, serta meng-*update* siapa saja user yang bisa mendapatkan hak akses untuk mengelola data laporan kualitas *plating* sesuai dengan perannya masing – masing .



Gambar V.28 Rancangan Data Master User  
Sumber: Hasil Analisis Data (2016)

#### 4. Menu Data Jenis Barang

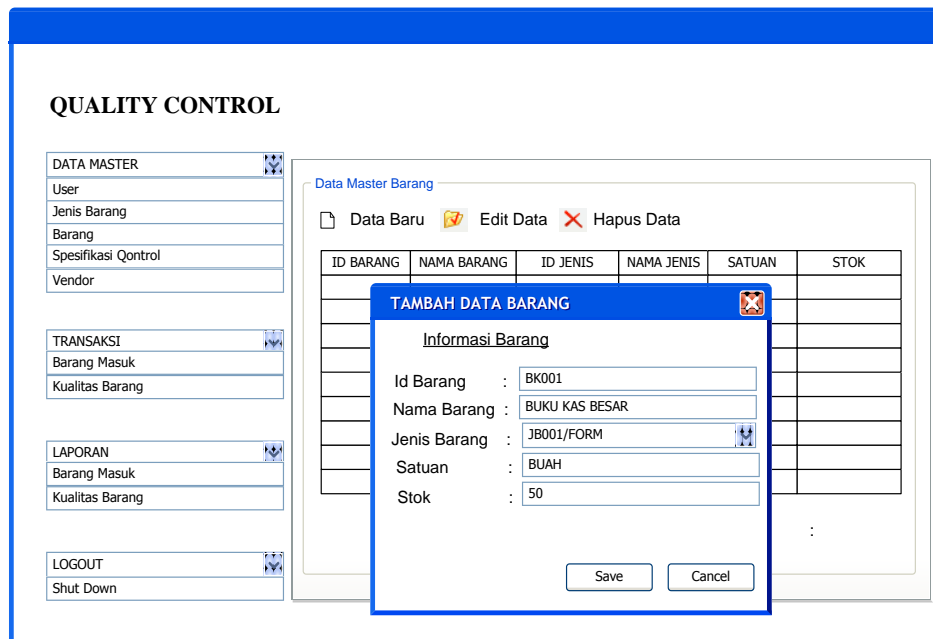
Menu data jenis barang ini merupakan salah satu sub menu dari menu data master, berfungsi untuk menambah, meng-*edit*, serta menghapus data jenis barang produksi pada sistem laporan kualitas *plating*.



Gambar V.29 Rancangan Data Jenis Barang  
Sumber: Hasil Analisis Data (2016)

## 5. Menu Data Barang

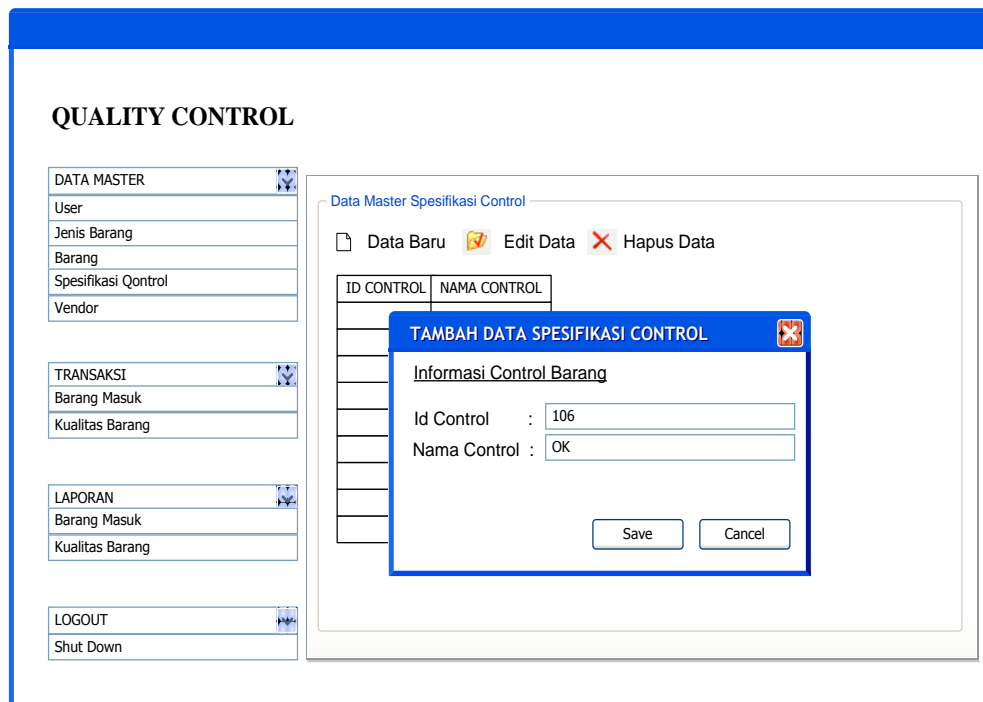
Menu data barang ini merupakan salah satu sub menu dari menu data master, berfungsi untuk menambah, meng-*edit*, serta menghapus data barang pada sistem laporan kualitas *plating*.



Gambar V.30 Rancangan Data Barang  
Sumber: Hasil Analisis Data (2016)

## 6. Menu Data Spesifikasi *Control*

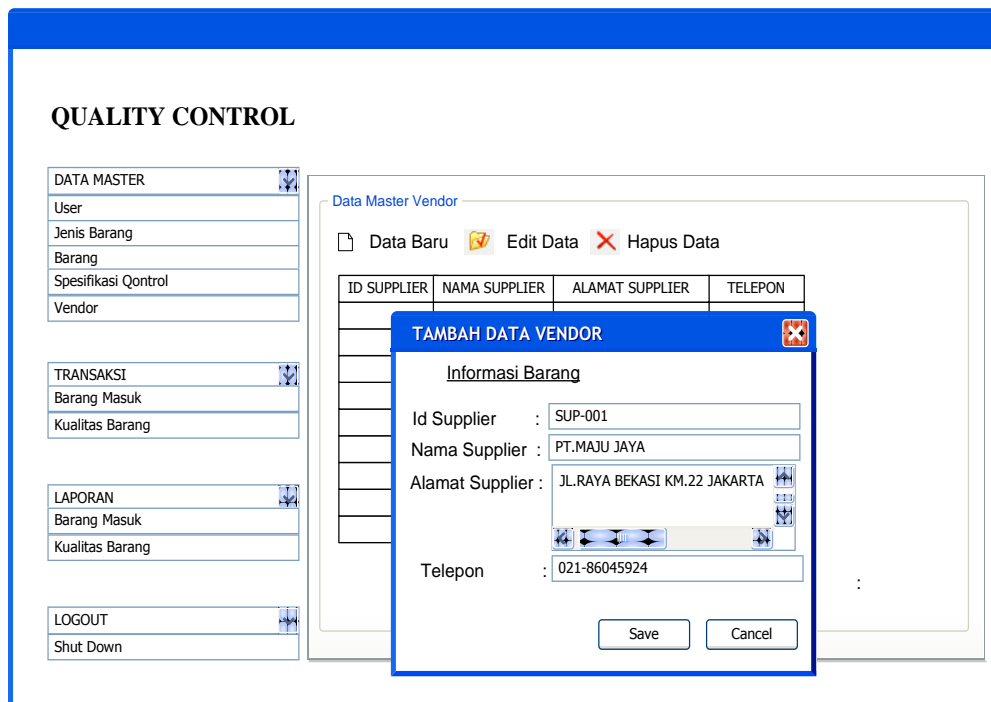
Menu data vendor ini merupakan salah satu sub menu dari menu data master, berfungsi untuk menambah, meng-*edit*, serta menghapus data spesifikasi *control* pada sistem laporan kualitas *plating*.



Gambar V.31 Rancangan Data Spesifikasi *Control*  
 Sumber: Hasil Analisis Data (2016)

#### 7. Menu Data Vendor

Menu data vendor ini merupakan salah satu sub menu dari menu data master, berfungsi untuk menambah, meng-*edit*, serta menghapus data vendor pada sistem laporan kualitas *plating*.



Gambar V.32 Rancangan Data Vendor  
Sumber: Hasil Analisis Data (2016)

#### 8. Menu Data Transaksi Barang Masuk

Menu data transaksi barang masuk ini merupakan salah satu sub menu dari menu data transaksi, berfungsi untuk menambah data barang masuk pada sistem laporan kualitas *plating*.

**QUALITY CONTROL**

**DATA MASTER**

User  
Jenis Barang  
Barang  
Spesifikasi Qontrol  
Vendor

**TRANSAKSI**

Barang Masuk  
Kualitas Barang

**LAPORAN**

Barang Masuk  
Kualitas Barang

**LOGOUT**

Shut Down

**Data Transaksi Barang Masuk**

Ketik Kode Barang :

ID BARANG	NAMABARANG	ID JENIS	JENIS BARANG	TOK	JUMLAH MASUK	PILIH
						Add Item

ID BARANG	NAMABARANG	ID JENIS	NAMA JENIS	JUMLAH MASUK

Barang Masuk Lengkap

Id Transaksi :  Id Supplier :

Gambar V.33 Rancangan Data Transaksi Barang Masuk  
Sumber: Hasil Analisis Data (2016)

9. Menu Data Transaksi Kualitas Barang

Menu data transaksi kualitas barang ini merupakan salah satu sub menu dari menu data transaksi, berfungsi untuk menambah data barang kualitas barang pada sistem laporan kualitas *plating*.

**QUALITY CONTROL**

**DATA MASTER**

User

Jenis Barang

Barang

Spesifikasi Qontrol

Vendor

---

**TRANSAKSI**

Barang Masuk

Kualitas Barang

---

**LAPORAN**

Barang Masuk

Kualitas Barang

---

**LOGOUT**

Shut Down

Input Masuk Faktur Barang :

ID MASUK	TGL MASUK	ID SUPPLIER	ID USER	ACTION
				Quality Control

Tambah Item Quality Control

ID BARANG	NAMA BARANG	ID JENIS	JUMLAH MASUK

Barang Masuk Lengkap

Id Transaksi :  Id Supplier :

---

**Data Transaksi Kualitas Barang**

Edit Data  Hapus Data

ID BARANG	NAMA BARANG	ID JENIS	JUMLAH MASUK

Selesai Menambahkan

Id Quality Control :  Id Quality Control :

User Entry :

Gambar V.34 Rancangan Data Transaksi Kualitas Barang  
Sumber: Hasil Analisis Data (2016)

10. Menu Data Laporan Barang Masuk

Menu data laporan barang masuk ini merupakan salah satu sub menu dari menu data laporan, berfungsi untuk mencari data barang masuk pada sistem laporan kualitas *plating*.

**QUALITY CONTROL**

DATA MASTER

User

Jenis Barang

Barang

Spesifikasi Qontrol

Vendor

TRANSAKSI

Barang Masuk

Kualitas Barang

LAPORAN

Barang Masuk

Kualitas Barang

LOGOUT

Shut Down

**LAPORAN BARANG MASUK**

Tanggal Awal : 02-08-2016

Tanggal Akhir : 31-08-2016

ID TRANSAKSI	TANGGAL	SUPPLIER	ID BARANG	JML MASUK

Gambar V.35 Rancangan Data Laporan Barang Masuk  
Sumber: Hasil Analisis Data (2016)

#### 11. Menu Data Laporan Kualitas Barang

Menu data laporan kualitas barang masuk ini merupakan salah satu sub menu dari menu data laporan, berfungsi untuk mencari data kualitas barang hasil produksi pada sistem laporan kualitas *plating*.

**QUALITY CONTROL**

**DATA MASTER**

User

Jenis Barang

Barang

Spesifikasi Qontrol

Vendor

**TRANSAKSI**

Barang Masuk

Kualitas Barang

**LAPORAN**

Barang Masuk

Kualitas Barang

**LOGOUT**

Shut Down

**LAPORAN QUALITY CONTROL**

Tanggal Awal : 02-08-2016

Tanggal Akhir : 31-08-2016

ID TRANSAKSI	TANGGAL	NAMA BARANG	ID CONTROL	JML KELUAR

Gambar V.36 Rancangan Data Laporan Kualitas Barang  
Sumber: Hasil Analisis Data (2016)

## 5.5 Testing Dan Implementasi Sistem

Setelah tahapan analisis dan desain selesai, selanjutnya adalah tahap testing dan implementasi. Pada tahap ini dilakukan pengujian sistem dengan memasukkan beberapa contoh kode program dan beberapa data untuk menunjang keberlangsungan dan pengujian sistem ini. Sedangkan untuk implementasi akan dikemukakan beberapa spesifikasi yang dibutuhkan oleh sistem usulan, baik dari perangkat lunak (*software*) maupun perangkat keras (*hardware*).

### 5.6.1 Implementasi Perangkat Keras dan Lunak

Perangkat lunak yang digunakan adalah:

1. *Microsoft Windows XP* atau *Microsoft Windows 7*

Sistem operasi yang digunakan pada komputer.

2. XAMPP 1.77

Aplikasi petugas gudang yang digunakan untuk mengolah basis data.

3. Google Chrome

Browser untuk menjalankan program XAMPP.

Perangkat keras yang digunakan dalam pembuatan aplikasi ini adalah:

1. Processor: IntelCore i3 CPU @2.40 GHz
2. Monitor: 10.2 WSVGA 1024 x 600
3. HDD: 500GB SATA
4. Memori RAM 2 GB DDR II
5. Printer untuk mencetak dokumen

## **BAB VI**

### **KESIMPULAN DAN SARAN**

#### **6.1 Kesimpulan**

Berdasarkan hasil penelitian dengan mengumpulkan data, mengolah data, dan melakukan perancangan, maka dapat disimpulkan bahwa:

1. Program sistem informasi laporan kualitas *plating* ini, maka proses pelaporan kualitas *plating* oleh bagian *quality control* menjadi lebih efektif dan efisien. Sebelumnya bagian *quality control* masih menggunakan form. Dengan terprogram bagian *quality control* hanya perlu membuka aplikasi laporan kualitas *plating*
2. Dengan adanya aplikasi ini bagian *quality control* dapat dengan mudah mencari data kualitas hasil produksi di *database* tanpa harus membuka map satu per satu.

#### **6.2 Saran**

Beberapa saran yang diharapkan agar program sistem informasi laporan kualitas *plating* ini menjadi lebih baik adalah sebagai berikut:

1. Diperlukan pelatihan khusus terhadap pengguna aplikasi sistem informasi laporan kualitas *plating* ini
2. Adanya pemeliharaan terhadap sistem yang telah dibuat agar sistem tetap terjaga dengan baik, dengan cara melakukan perbaikan pada sistem apabila terjadi kesalahan atau *error* pada aplikasi tersebut.
3. Untuk pengembangan sistem selanjutnya diharapkan dapat membahas rancangan *interface* yang lebih menarik agar *user* dapat lebih mudah menggunakannya.

## DAFTAR PUSTAKA

- Abdul Kadir, 2003, Pengenalan Sistem Informasi, Andi, Yogyakarta.
- Aman Hakim & Yudha Prasetyawan. 2008. Perencanaan dan Pengendalian Produksi. Yogyakarta: Graha Ilmu
- Anhar. (2010). *Panduan menguasai PHP & Mysql*. Jakarta.
- Arsip PT Surya Sunjaya Industri Bagian Produksi – Cikarang.
- A.S., Rosa dan Shalahuddin, M. 2013. Rekayasa Perangkat Lunak Terstruktur Dan Berorientasi Objek Informatika. Bandung.
- Assauri, Sofyan. 1993. Manajemen Produksi. Edisi Revisi. Lembaga Penerbit Fakultas Ekonomi Universitas Indonesia, Jakarta.
- Assauri, Sofyan S.E, MBA. 1999. *Manajemen Produksi dan Operasi*. Edisi Revisi. Lembaga Penerbit Fakultas Ekonomi Universitas Indonesia, Jakarta.
- Gaspersz, Vincent. 2005. Total Quality Management. Jakarta: Gramedia Pustaka Utama
- Henderi. 2007. *Analysis and Design System with Unified Modeling Language (UML)*. STMIK Raharja, Tangerang.

- Huda, Miftakhul. 2010. *Membuat Aplikasi Database dengan Java, My SQL, dan Netbeans*. Jakarta: Elex Media Komputindo.
- Jogiyanto, HM. 1999. *Pengenalan Komputer: Dasar Ilmu computer, Pemrograman, Sistem Informasi Dan Itelegensi buatan*. Edisi Ketiga. Andi Offset, Yogyakarta.
- Jogiyanto. 2005. *Analisis & Desain Sistem Informasi : pendekatan terstruktur teori dan praktek aplikasi bisnis*. Andi, Yogyakarta.
- Kadir, Abdul. 2003. *Pengenalan Sistem Infromasi*. Yogyakarta: Andi.
- Kamus Besar Bahasa Indonesia. 2008. Pusat Bahasa Departement Pendidikan Nasional, Jakarta.
- Keraf. (2001). *Ilmu Pengetahuan*, Jakarta : Kanisius
- Kotler, Philip dan Gary Amstrong, 1996, *Dasar-Dasar Pemasaran*, Edisi V, jilid 2, Intermedia, Jakarta.
- McLeod, Raymond, 2007, *Sistem Informasi Manajemen*. PT Indeks, Jakarta.
- Mulyadi. 2005. *Akuntansi Biaya*, Edisi Kelima, Cetakan Ketujuh, Yogyakarta: Akademi Manajemen Perusahaan YKPN.
- Munawar. 2005. *Pemodelan Visual dengan UML Edisi Pertama*. Graha Ilmu, Yogyakarta.

- Nasrullah. 1997. Pengantar Teknik Industri. Jakarta: Gunadarma Peranginangin,
- Kasiman. 2006. Aplikasi Web dengan PHP dan MySQL. Yogyakarta: Andi
- Nur Indriantoro dan Bambang Soepomo. 1999. Metodologi Penelitian Bisnis Untuk Akuntansi Dan Manajemen. Yogyakarta: BPEE
- Rosa, A.S. & Shalahuddin, M. 2011. *Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Cetakan Pertama. Bandung: Modula.
- Sommerville, I. 2003, *Software Engineering* (Rekayasa Perangkat Lunak) edisi 6 jilid 1, Erlangga: Jakarta.
- Sutedjo, Budi Dharma Oetomo. (2006). Perencanaan dan Pembangunan Sistem Informasi. Yogyakarta.
- Syaukani, M. (2005). Mengolah data pada MySQL Foxpro 8 +cd. Jakarta: Elex Media Komputindo.
- Wardani, Diah Kusuma, 2008. “Pengaruh Corporate Governance Terhadap Kinerja Perusahaan di Indonesia”. Fakultas Ekonomi UII: Yogyakarta
- Welsch, Glen A. 2003. Perencanaan Dan Pengendalian Laba. Edisi Pertama. Jakarta: Salemba Empat
- Yamit, Zulian. 2013. Manajemen Kualitas Produk Dan Jasa. Cetakan Keenam. Yogyakarta: Ekonisia

# LAMPIRAN

## 1. Koneksi Basis Data

```
<?php
$user_db="root";
$pass_db="";
$server_db="localhost";
$nama_db="quality";
$koneksi=mysql_connect($server_db,$user_db,$pass_db) or die(mysql_error());
$pilih_db=mysql_select_db($nama_db);
?>
```

## 2. Login

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Login Sistem Stok</title>
<link rel="stylesheet" type="text/css" href="mycss/login.css" />
<script type="text/javascript" src="jquery_easyui/jquery.min.js"></script>
<script type="text/javascript" src="libs_js/login.js"></script>
</head>

<body>
<div id="header"><marquee>Selamat Datang di Aplikasi Quality Control Barang
2016 @Fahrul Rozi Kampus STMI Jakarta </marquee>
</div>
<div class="lg-container">
<h1>LOGIN SYSTEM </h1>
<form action="login_akses.php" id="lg-form" name="lg-form" method="post">
```

```

<div>
    <label for="username">Username:</label>
    <input type="text" name="username" id="username"
placeholder="username"/>
</div>

    <div>
        <label for="password">Password:</label>
        <input type="password" name="password"
id="password" placeholder="password" />
    </div>

    <div>
        <button type="submit" id="login">
Login</button>
    </div>

</form>
<div id="message"></div>
</div>
</body>
</html>

```

### 3. Menu Utama

```

<?php
session_start();
$user_name=$_SESSION['nm_user'];

```

```

if(empty($_SESSION['id_user'])&&empty($_SESSION['nm_level'])){
header("location:login.php");
}
else{
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Sistem Quality Control</title>
<link rel="stylesheet" type="text/css" href="mycss/index.css">
<link
                rel="stylesheet"
                type="text/css"
href="jquery_easyui/themes/default/easyui.css">
<link rel="stylesheet" type="text/css" href="jquery_easyui/themes/icon.css">
<link rel="stylesheet" type="text/css" href="jquery_easyui/themes/style.css">
<link rel="stylesheet" type="text/css" href="jquery_easyui/themes/panel.css" />

<script type="text/javascript" src="jquery_easyui/jquery.min.js"></script>
<script type="text/javascript" src="jquery_easyui/jquery.easyui.min.js"></script>
<script type="text/javascript" src="jquery_easyui/jquery.form.js"></script>
<script>
function addTab(title, url){
                if ($('#tt').tabs('exists', title)){
                        $('#tt').tabs('select', title);
                } else {
                        var content = '<iframe scrolling="auto"
frameborder="0" src="'+url+'" style="width:100%;height:100%;"></iframe>';
                        $('#tt').tabs('add',{

```

```

        title:title,
        content:content,
        closable:true
    });
    }
}
</script>
</head>

<body>
<div id="header"><a href="index.php?module=beranda"></a> Quality Control
<div id="user-login"> Selamat Datang :  <font
color="#BB0000"><?php echo $user_name ?></font></div>
</div>
<div id="navigasi">
<div style="width:200px;height:auto;padding:5px;float:left; margin-right:10px;">
<div class="easyui-panel" title="DATA MASTER" collapsible="true"
style="width:200px;height:auto;padding:10px; ">
<a href="?module=pengguna" class="easyui-linkbutton" iconCls="icon-user"
plain="true" >User</a><br />
<a href="?module=jenis" class="easyui-linkbutton" iconCls="icon-box"
plain="true" >Jenis Barang</a><br />
<a href="?module=barang" class="easyui-linkbutton" iconCls="icon-box-fill"
plain="true">Barang</a><br />
<a href="?module=spesifikasicontrol" class="easyui-linkbutton" iconCls="icon-
shop" plain="true">Spesifikasi Qontrol</a><br />

```

```
<a href="?module=vendor" class="easyui-linkbutton" iconCls="icon-supplier"
plain="true">Vendor</a><br />
</div>
<br/>
<div class="easyui-panel" title="TRANSAKSI" collapsible="true"
style="width:200px;height:auto;padding:10px;
background:background:url('panel/status.png');">
<a href="?module=barangmasuk" class="easyui-linkbutton" iconCls="icon-trolley"
plain="true" onClick="addTab('Barang Masuk','barang_masuk.php')">Barang
Masuk</a><br />
<a href="?module=kualitasbarang" class="easyui-linkbutton" iconCls="icon-
truck" plain="true" onClick="addTab('Barang
Keluar','retur_keluar.php')">Kualitas Barang</a><br />
</div><br />
<div class="easyui-panel" title="LAPORAN" collapsed="true" collapsible="true"
style="width:200px;height:auto;padding:10px;">
<a href="?module=lapbarangmasuk" class="easyui-linkbutton" iconCls="icon-
report2" plain="true" onClick="addTab('Laporan Barang
Masuk','laporan/barang_masuk/pilih_lap.php')">Barang Masuk</a><br />
<a href="?module=lapquality" class="easyui-linkbutton" iconCls="icon-report3"
plain="true">Kualitas Barang</a><br />
<a href="?module=stokbarang" class="easyui-linkbutton" iconCls="icon-report4"
plain="true" onClick="addTab('Laporan Stok
Barang','laporan/stok_barang/pilih_lap.php')">Stok Barang</a>
</div><br />
<div class="easyui-panel" title="LOGOUT" collapsible="true"
style="width:200px;height:auto;padding:10px;">
<a href="logout.php" class="easyui-linkbutton" iconCls="icon-logout"
plain="true">Shut Down</a>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div id="tt" style="height:500px;">
```

```
<?php include "content.php"; ?>
```

```
</div>
```

```
<div id="footer" align="right">
```

```
</div>
```

```
</body>
```

```
</html>
```

```
<?php } ?>
```

#### 4. Barang

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<title>Untitled Document</title>
```

```
<script type="text/javascript" src="libs_js/barang.js"></script>
```

```
</head>
```

```
<body>
```

```
<div style="margin-bottom:10px">
```

```
<div class="tip icon-tip">&nbsp;   </div>
```

```
<div>Klik tombol pada datagrid toolbar untuk melakukan perubahan data.</div>
```

```
</div>
```

```
<table id="dg" title=".: MASTER DATA BARANG :." class="easyui-datagrid" style="height:390px"
```

```
url="data_master/barang/get_barang.php"
```

```
toolbar="#toolbar" pagination="true"
```

```
rownumbers="true" fitColumns="true" singleSelect="true">
```

```
<thead>
```

```
<tr>
```

```
<th field="id_barang" width="30">ID BARANG</th>
```

```
<th field="nm_barang" width="80">NAMA BARANG</th>
```

```
<th field="id_jenis" width="25">ID JENIS</th>
```

```
<th field="nm_jenis" width="50">NAMA JENIS</th>
```

```
<th field="satuan" width="40">SATUAN</th>
```

```
<th field="stok" width="40">STOK</th>
```

```
</tr>
```

```
</thead>
```

```
</table>
```

```
<div id="toolbar">
```

```
<a href="#" class="easyui-linkbutton" iconCls="icon-add" plain="true" onclick="newData()">Data Baru</a>
```

```
<a href="#" class="easyui-linkbutton" iconCls="icon-edit" plain="true" onclick="editData()">Edit Data</a>
```

```
                <a href="#" class="easyui-linkbutton" iconCls="icon-remove"
plain="true" onclick="removeData()">Hapus Data</a>
            </div>
```

```
        <div id="dlg" class="easyui-dialog"
style="width:400px;height:340px;padding:10px 20px"
        closed="true" buttons="#dlg-buttons">
            <div class="ftitle">Informasi Barang</div>
            <form id="fm" method="post" novalidate>
                <div class="fitem">
                    <label>Id Barang:</label>
                    <input name="id_barang" id="id_barang"
class="easyui-validatebox" required="true">
                </div>
                <div class="fitem">
                    <label>Nama Barang:</label>
                    <input name="nm_barang" id="nm_barang"
class="easyui-validatebox" required="true">
                </div>
                <div class="fitem">
                    <label>Jenis Barang:</label>
                    <select name="id_jenis" id="id_jenis" >
                        <option></option>
                    </select>
                </div>
                <div class="fitem">
                    <label>Satuan:</label>
                    <input name="satuan" class="easyui-validatebox"
required="true" >
```

```

        </div>
    <div class="fitem">
        <label>Stok:</label>
        <input      name="stok"      class="easyui-numberbox"
required="true" >
        </div>
    </form>
</div>
<div id="dlg-buttons">
    <a href="#" class="easyui-linkbutton" iconCls="icon-ok"
onclick="saveData()">Save</a>
    <a href="#" class="easyui-linkbutton" iconCls="icon-cancel"
onclick="javascript:$('#dlg').dialog('close')">Cancel</a>
</div>
</body>
</html>

```

## 5. Barang Masuk

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
<link href="mycss/request.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="libs_js/barang_masuk.js"></script>
</head>

<body>

```

```

<div id="cari">
<form action="transaksi/search_barang.php" method="post" id="form_cari">
  Ketik Kode Barang :
  <input name="cari_id_brg" type="text" id="cari_id_brg" />
  <input name="cari" type="image" id="cari2" value="Cari"
src="jquery_easyui/themes/icons/search.png" />
</form>
</div>
<br />
<div id="data">

  <table width="76%" border="1" align="center" cellpadding="0" cellspacing="0"
id="tabel_item">
  <tr>
    <td width="8%" align="center" bgcolor="#EEEEEE">ID BARANG</td>
    <td width="24%" align="center" bgcolor="#EEEEEE">NAMA
BARANG</td>
    <td width="7%" align="center" bgcolor="#EEEEEE">ID JENIS</td>
    <td width="19%" align="center" bgcolor="#EEEEEE">JENIS
BARANG</td>
    <td width="9%" align="center" bgcolor="#EEEEEE"> STOK</td>
    <td width="19%" align="center" bgcolor="#EEEEEE">JUMLAH
MASUK</td>
    <td width="14%" align="center" bgcolor="#EEEEEE">PILIH</td>
  </tr>

  <form action="transaksi/barang_masuk/save_item.php" method="post"
name="form_item" id="form_item">
  <tr>

```

```
        <td height="23"><input name="id_barang" type="text" disabled="disabled"
id="id_barang" size="15" readonly="readonly" /></td>
        <td><input name="nm_barang" type="text" disabled="disabled"
id="nm_barang" size="50" readonly="readonly" /></td>
        <td><label for="id_jenis"></label>
        <input name="id_jenis" type="text" disabled="disabled" id="id_jenis"
size="15" readonly="readonly" /></td>
        <td><label for="nm_jenis"></label>
        <input name="nm_jenis" type="text" disabled="disabled" id="nm_jenis"
size="40" readonly="readonly" /></td>
        <td><input name="stok" type="text" disabled="disabled" id="stok" size="8"
readonly="readonly" /></td>
        <td><input name="jumlah" type="text" disabled="disabled" id="jumlah"
size="12" /></td>
        <td><input type="submit" name="tambah_item" id="tambah_item"
value="Add Item" /></td>
    </tr>
</form>
```

```
</table>
</div>
<div id="detail_request">
<table id="dg" title="DATA BARANG MASUK" class="easyui-datagrid"
style="height:250px"
        url="transaksi/barang_masuk/get_item_masuk.php"
        toolbar="#toolbar" pagination="true"
        rownumbers="true" fitColumns="true" singleSelect="true">
    <thead>
        <tr>
```

```

        <th field="id_barang" width="50">ID BARANG</th>
                <th field="nm_barang" width="50">NAMA
BARANG</th>
                <th field="id_jenis" width="50">ID JENIS</th>
                <th field="nm_jenis" width="50">NAMA JENIS</th>
        <th field="jml" width="50">JUMLAH</th>

        </tr>
    </thead>
</table>

    <div id="toolbar">
        <a href="#" class="easyui-linkbutton" iconCls="icon-edit"
plain="true" onclick="editData()">Edit Data</a>
        <a href="#" class="easyui-linkbutton" iconCls="icon-remove"
plain="true" onclick="removeData()">Hapus Data</a>
    </div>

    <div id="dlg" class="easyui-dialog"
style="width:400px;height:280px;padding:10px 20px"
        closed="true" buttons="#dlg-buttons">
        <div class="ftitle">Edit Jumlah Masuk</div>
        <form id="fm" method="post" novalidate>
            <div class="fitem">
                <label>Id Barang:</label>
                <input name="id_barang" disabled="disabled"
class="easyui-validatebox" >
            </div>
            <div class="fitem">
                <label>Nama Barang:</label>

```

```

                <input    name="nm_barang"    disabled="disabled"
class="easyui-validatebox" >
            </div>
        <div class="fitem">
            <label>Jenis Barang:</label>
            <input    name="nm_jenis"    disabled="disabled"
class="easyui-validatebox">
        </div>
        <div class="fitem">
            <label>Jml Masuk:</label>
            <input    name="jml"        class="easyui-numberbox"
required="true" >
        </div>
    </form>
</div>
<div id="dlg-buttons">
    <a    href="#"    class="easyui-linkbutton"    iconCls="icon-ok"
onclick="saveData()">Save</a>
    <a    href="#"    class="easyui-linkbutton"    iconCls="icon-cancel"
onclick="javascript:$('#dlg').dialog('close')">Cancel</a>
</div>

<div id="proses">
    <input    type='checkbox'    name='term'    onClick="Javascript:disab(this,
1);"/>Barang Masuk Lengkap<br /><br />

<form    action="transaksi/barang_masuk/proses.php"    method="post"
id="form_proses" style="display:none">
<label>Id Transaksi:</label>

```

```

<input name="id_trx" type="text" id="id_trx" readonly="readonly" />
<label>Id Supplier:</label>
<select name="id_supplier" id="id_supplier" >

</select>
<input type="submit" name="proses_request" id="proses_request" value="Proses
Transaksi " disabled="disabled" />
</form>
</div>
</div>
</body>
</html>

```

## 6. Jenis Barang

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
<script type="text/javascript" src="libs_js/jenis_barang.js"></script>
</head>

<body>

<div style="margin-bottom:10px">
    <div class="tip icon-tip">&nbsp;</div>
    <div>Klik tombol pada datagrid toolbar untuk melakukan perubahan
data.</div>

```

```
</div>
```

```
<table id="dg" title=".: MASTER DATA JENIS BARANG :." class="easyui-datagrid" style="height:390px"
```

```
url="data_master/jenis_barang/get_jenis.php"
```

```
toolbar="#toolbar" pagination="true"
```

```
rownumbers="true" fitColumns="true" singleSelect="true">
```

```
<thead>
```

```
<tr>
```

```
<th field="id_jenis" width="50">ID JENIS</th>
```

```
<th field="nm_jenis" width="50">NAMA JENIS</th>
```

```
</tr>
```

```
</thead>
```

```
</table>
```

```
<div id="toolbar">
```

```
<a href="#" class="easyui-linkbutton" iconCls="icon-add" plain="true" onclick="newData()">Data Baru</a>
```

```
<a href="#" class="easyui-linkbutton" iconCls="icon-edit" plain="true" onclick="editData()">Edit Data</a>
```

```
<a href="#" class="easyui-linkbutton" iconCls="icon-remove" plain="true" onclick="removeData()">Hapus Data</a>
```

```
</div>
```

```
<div id="dlg" class="easyui-dialog" style="width:400px;height:280px;padding:10px 20px"
```

```
closed="true" buttons="#dlg-buttons">
```

```
<div class="fitem">Informasi Jenis Barang</div>
```

```
<form id="fm" method="post" novalidate>
```

```
<div class="fitem">
```

```

        <label>Id Jenis:</label>
        <input name="id_jenis" id="id_jenis" class="easyui-
validatebox" required="true">
    </div>
    <div class="fitem">
        <label>Nama Jenis:</label>
        <input name="nm_jenis" class="easyui-validatebox"
required="true">
    </div>
</form>
</div>
<div id="dlg-buttons">
    <a href="#" class="easyui-linkbutton" iconCls="icon-ok"
onclick="saveData()">Save</a>
    <a href="#" class="easyui-linkbutton" iconCls="icon-cancel"
onclick="javascript:$('#dlg').dialog('close')">Cancel</a>
</div>
</body>
</html>

```

## 7. Quality Control

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
<link rel="stylesheet" type="text/css" href="mycss/retur.css">
<script type="text/javascript" src="libs_js/retur_keluar.js"></script>

```

```
</head>
```

```
<body>
```

```
<div id="cari">
```

```
<form          action="transaksi/retur_barang/search_faktur_barang_keluar.php"
method="post" id="form_cari">
```

```
  Input Faktur Masuk Barang :
```

```
    <input name="cari_id_faktur" type="text" id="cari_id_faktur" />
```

```
    <input          type="image"          name="button_cari"          id="button_cari"
src="jquery_easyui/themes/icons/search.png" />
```

```
</form>
```

```
</div>
```

```
<div id="hasil_cari">
```

```
<table width="400" border="1" align="center" cellpadding="0" cellspacing="0"
id="tabel_item">
```

```
  <tr>
```

```
    <td align="center">ID MASUK</td>
```

```
    <td align="center">TGL MASUK</td>
```

```
    <td align="center">ID SUPPLIER</td>
```

```
    <td align="center">ID USER</td>
```

```
    <td align="center">ACTION</td>
```

```
  </tr>
```

```
  <form action="?module=kualitasbarang" method="post" name="form_item"
id="form_item">
```

```
    <tr>
```

```
      <td><input name="id_masuk" type="text" disabled="disabled" id="id_masuk"
size="16" readonly="readonly" /></td>
```

```

        <td><input      name="tgl_masuk"      type="text"      disabled="disabled"
id="tgl_masuk" size="15" readonly="readonly" /></td>
        <td><label for="id_supplier"></label>
        <input name="id_supplier" type="text" disabled="disabled" id="id_supplier"
size="16" readonly="readonly" /></td>
        <td align="center"><label for="id_user"></label>
        <input name="id_user" type="text" disabled="disabled" id="id_user"
size="15" readonly="readonly" /></td>
        <td><input type="submit" name="retur" id="retur" value="Quality Control"
/></td>
    </tr>
</form>

```

```

</table>
</div>
<div id="data">
<table id="dgKeluar" class="easyui-datagrid" style="height:200px"
toolbar="#toolbarKeluar" pagination="true"
url="transaksi/retur_barang/get_item_barang.php?id_trx=<?php      echo
$_POST['id_masuk'];      ?>&tr=out" rownumbers="true" fitColumns="true"
singleSelect="true">
    <thead>
        <tr>
            <th field="id_barang" width="50">ID BARANG</th>
            <th field="nm_barang" width="70">NAMA
BARANG</th>
            <th field="id_jenis" width="30">ID JENIS</th>
            <th field="jml_masuk" width="30">JUMLAH MASUK</th>

```

```

        </tr>
    </thead>
</table>
<div id="toolbarKeluar"><a href="#" class="easyui-linkbutton" iconCls="icon-add" plain="true" onclick="addRetur()">Tambah Item Quality Control</a> </div>

<div id="dlgKeluar" class="easyui-dialog"
style="width:400px;height:340px;padding:10px 20px" closed="true"
buttons="#dlg-buttons">
    <div class="ftitle">Quality Control</div>
    <form id="fmKeluar" method="post" novalidate>
        <div class="fitem">
            <label>Id Barang:</label>
            <input name="id_barang" id="id_barang"
class="easyui-validatebox" required="true">
        </div>
        <div class="fitem">
            <label>Nama Barang:</label>
            <input name="nm_barang" id="nm_barang"
class="easyui-validatebox" required="true" readonly="readonly">
        </div>
        <div class="fitem">
            <label>Jml Masuk:</label>
            <input name="jml_masuk" class="easyui-numberbox"
required="true" readonly="readonly" >
        </div>
        <div class="fitem">
            <label>Quality Control:</label>

```

```

                <select name="quality_control" class="easyui-combobox"
required="true">
                <?php
                    include"koneksi/koneksi.php";
                    $cari=mysql_query("select * from spesifikasi");
                    while($data=mysql_fetch_array($cari)){
                        ?>
                <option value="<?php echo "$data[id_control]"; ?>"><?php echo
"$data[id_control]"; ?>|<?php echo "$data[nm_control]"; ?></option>
                </select>
                </div>
                <div class="fitem">
                    <label>Jumlah Quality:</label>
                    <input name="jml_retur" class="easyui-numberbox"
required="true" >
                </div>
                </form>
            </div>
            <div id="dlg-buttons">
                <a href="#" class="easyui-linkbutton" iconCls="icon-ok"
onclick="saveRetur()">Save</a>
                <a href="#" class="easyui-linkbutton" iconCls="icon-cancel"
onclick="javascript:$('#dlgKeluar').dialog('close')">Cancel</a>
            </div>
        </div>

        <div id="detail_request">
            <table id="dgRetur" title="DATA QUALITY CONTROL" class="easyui-
datagrid" style="height:200px"toolbar="#toolbarRetur" pagination="true"

```

```
url="transaksi/retur_barang/get_item_retur.php?id_trx=<?php  
$_POST['id_masuk']; ?>" echo
```

```
        rownumbers="true" fitColumns="true" singleSelect="true">  
<thead>  
    <tr>  
        <th field="id_barang" width="50">ID BARANG</th>  
        <th field="nm_barang" width="50">NAMA  
BARANG</th>  
        <th field="hrg" width="50">ID CONTROL</th>  
        <th field="jml" width="50">JUMLAH BARANG</th>  
    </tr>  
</thead>  
</table>  
    <div id="toolbarRetur">  
        <a href="#" class="easyui-linkbutton" iconCls="icon-edit"  
plain="true" onclick="editRetur()">Edit Data</a>  
        <a href="#" class="easyui-linkbutton" iconCls="icon-remove"  
plain="true" onclick="removeRetur()">Hapus Data</a>  
    </div>  
  
    <div id="dlgRetur" class="easyui-dialog"  
style="width:400px;height:250px;padding:10px 20px"  
        closed="true" buttons="#dlg-buttons">  
        <div class="ftitle">Edit Item Keluar Barang</div>  
        <form id="fmRetur" method="post" novalidate>  
            <div class="fitem">  
                <label>Id Barang:</label>
```

```

                <input      name="id_barang"      id="id_barang"
class="easyui-validatebox" required="true">
            </div>
            <div class="fitem">
                <label>Nama Barang:</label>
                <input      name="nm_barang"      id="nm_barang"
class="easyui-validatebox" required="true" readonly="readonly">
            </div>
            <div class="fitem">
                <label>Harga Beli:</label>
                <input name="hrg_beli" id="hrg_beli" class="easyui-
validatebox" required="true">
            </div>
            <div class="fitem">
                <label>Jml Retur:</label>
                <input  name="jml"  id="jml"  class="easyui-numberbox"
required="true" >
            </div>
        </form>
    </div>
    <div id="dlg-buttons">
        <a href="#" class="easyui-linkbutton" iconCls="icon-ok"
onclick="saveEditRetur()">Save</a>
        <a href="#" class="easyui-linkbutton" iconCls="icon-cancel"
onclick="javascript:$('#dlgRetur').dialog('close')">Cancel</a>
    </div>

    <div id="proses">

```

```
<input type='checkbox' name='term' onClick="Javascript:disab(this,
1);"/>Selesai Menambahkan<br /><br />
```

```
<form action="transaksi/retur_barang/proses_retur_barang_keluar.php"
method="post" id="form_proses" style="display:none">
<label>Id Quality Control:</label>
<input name="id_trx" type="text" id="id_trx" readonly="readonly" />
<label>Id Barang Masuk:</label>
<input name="id_masuk" type="text" id="id_masuk" value="<?php echo
$_POST['id_masuk'] ?>" readonly="readonly" />
<label>Id User Entry:</label>
<input name="id_user" type="text" id="id_user" value="<?php echo
$_POST['id_user'] ?>" readonly="readonly" />
<input type="submit" name="proses_request" id="proses_request" value="Proses
Transaksi " disabled="disabled" />
</form>
</div>
</div>
</body>
</html>
```

## 8. Supplier

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
<script type="text/javascript" src="libs_js/spesifikasi_control.js"></script>
```

```
</head>
```

```
<body>
```

```
<div style="margin-bottom:10px">
```

```
    <div class="tip icon-tip">&nbsp;  </div>
```

```
    <div>Klik tombol pada datagrid toolbar untuk melakukan perubahan  
data.</div>
```

```
  </div>
```

```
    <table id="dg" title=".: MASTER DATA SPESIFIKASI CONTROL :."  
class="easyui-datagrid" style="height:390px"
```

```
    url="data_master/spesifikasi/get_control.php"
```

```
    toolbar="#toolbar" pagination="true"
```

```
    rownumbers="true" fitColumns="true" singleSelect="true">
```

```
    <thead>
```

```
      <tr>
```

```
        <th          field="id_control"          width="50">ID
```

```
CONTROL</th>
```

```
        <th          field="nm_control"          width="50">NAMA
```

```
CONTROL</th>
```

```
      </tr>
```

```
    </thead>
```

```
  </table>
```

```
  <div id="toolbar">
```

```
    <a href="#" class="easyui-linkbutton" iconCls="icon-add"  
plain="true" onclick="newData()">Data Baru</a>
```

```
    <a href="#" class="easyui-linkbutton" iconCls="icon-edit"  
plain="true" onclick="editData()">Edit Data</a>
```

```

        <a href="#" class="easyui-linkbutton" iconCls="icon-remove"
plain="true" onclick="removeData()">Hapus Data</a>
    </div>

    <div id="dlg" class="easyui-dialog"
style="width:400px;height:280px;padding:10px 20px"
        closed="true" buttons="#dlg-buttons">
        <div class="ftitle">Informasi control Barang</div>
        <form id="fm" method="post" novalidate>
            <div class="fitem">
                <label>Id control:</label>
                <input name="id_control" id="id_control" class="easyui-
validatebox" required="true">
            </div>
            <div class="fitem">
                <label>Nama control:</label>
                <input name="nm_control" class="easyui-validatebox"
required="true">
            </div>
        </form>
    </div>
    <div id="dlg-buttons">
        <a href="#" class="easyui-linkbutton" iconCls="icon-ok"
onclick="saveData()">Save</a>
        <a href="#" class="easyui-linkbutton" iconCls="icon-cancel"
onclick="javascript:$('#dlg').dialog('close')">Cancel</a>
    </div>
</body>
</html>

```

## 9. User

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
<script type="text/javascript" src="libs_js/user.js"></script>
</head>

<body>

<div style="margin-bottom:10px">
    <div class="tip icon-tip">&nbsp;</div>
    <div>Klik tombol pada datagrid toolbar untuk melakukan perubahan
data.</div>
</div>

    <table id="dg" title=".: MASTER DATA USER :." class="easyui-datagrid"
style="height:390px"
        url="data_master/user/get_user.php"
        toolbar="#toolbar" pagination="true"
        rownumbers="true" fitColumns="true" singleSelect="true">
    <thead>
    <tr>
        <th field="id_user" width="50">ID USER</th>
        <th field="nm_user" width="50">NAMA USER</th>
        <th field="level" width="50">LEVEL</th>
    </tr>
```

```

        </thead>
    </table>
    <div id="toolbar">
        <a href="#" class="easyui-linkbutton" iconCls="icon-add"
plain="true" onclick="newData()">Data Baru</a>
        <a href="#" class="easyui-linkbutton" iconCls="icon-edit"
plain="true" onclick="editData()">Edit Data</a>
        <a href="#" class="easyui-linkbutton" iconCls="icon-remove"
plain="true" onclick="removeData()">Hapus Data</a>
    </div>
    <div id="dlg" class="easyui-dialog"
style="width:400px;height:280px;padding:10px 20px"
        closed="true" buttons="#dlg-buttons">
        <div class="ftitle">Informasi User</div>
        <form id="fm" method="post" novalidate>
            <div class="fitem">
                <label>Id user:</label>
                <input name="id_user" id="id_user" class="easyui-
validatebox" required="true">
            </div>
            <div class="fitem">
                <label>Nama User:</label>
                <input name="nm_user" class="easyui-validatebox"
required="true">
            </div>
            <div class="fitem">
                <label>Hak Akses:</label>
                <select class="easyui-combobox" name="level" style="width:120px;"
required="true">

```

```

        <option value="admin">admin</option>
    <option value="super admin">super admin</option>
</select>

</div>
    <div class="fitem">
        <label>Password:</label>
        <input name="password" type="password" class="easyui-
validatebox" required="true">
    </div>
</form>
</div>
    <div id="dlg-buttons">
        <a href="#" class="easyui-linkbutton" iconCls="icon-ok"
onclick="saveData()">Save</a>
        <a href="#" class="easyui-linkbutton" iconCls="icon-cancel"
onclick="javascript:$('#dlg').dialog('close')">Cancel</a>
    </div>
</body>
</html>

```

## 10. Spesifikasi Control

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
<script type="text/javascript" src="libs_js/spesifikasi_control.js"></script>

```

```
</head>
```

```
<body>
```

```
<div style="margin-bottom:10px">
```

```
    <div class="tip icon-tip">&nbsp;  </div>
```

```
    <div>Klik tombol pada datagrid toolbar untuk melakukan perubahan  
data.</div>
```

```
  </div>
```

```
    <table id="dg" title=".: MASTER DATA SPESIFIKASI CONTROL :."  
class="easyui-datagrid" style="height:390px"
```

```
    url="data_master/spesifikasi/get_control.php"
```

```
    toolbar="#toolbar" pagination="true"
```

```
    rownumbers="true" fitColumns="true" singleSelect="true">
```

```
    <thead>
```

```
      <tr>
```

```
        <th          field="id_control"          width="50">ID
```

```
CONTROL</th>
```

```
        <th          field="nm_control"          width="50">NAMA
```

```
CONTROL</th>
```

```
      </tr>
```

```
    </thead>
```

```
  </table>
```

```
  <div id="toolbar">
```

```
    <a href="#" class="easyui-linkbutton" iconCls="icon-add"  
plain="true" onclick="newData()">Data Baru</a>
```

```
    <a href="#" class="easyui-linkbutton" iconCls="icon-edit"  
plain="true" onclick="editData()">Edit Data</a>
```

```

        <a href="#" class="easyui-linkbutton" iconCls="icon-remove"
plain="true" onclick="removeData()">Hapus Data</a>
    </div>

    <div id="dlg" class="easyui-dialog"
style="width:400px;height:280px;padding:10px 20px"
        closed="true" buttons="#dlg-buttons">
        <div class="ftitle">Informasi control Barang</div>
        <form id="fm" method="post" novalidate>
            <div class="fitem">
                <label>Id control:</label>
                <input name="id_control" id="id_control" class="easyui-
validatebox" required="true">
            </div>
            <div class="fitem">
                <label>Nama control:</label>
                <input name="nm_control" class="easyui-validatebox"
required="true">
            </div>
        </form>
    </div>
    <div id="dlg-buttons">
        <a href="#" class="easyui-linkbutton" iconCls="icon-ok"
onclick="saveData()">Save</a>
        <a href="#" class="easyui-linkbutton" iconCls="icon-cancel"
onclick="javascript:$('#dlg').dialog('close')">Cancel</a>
    </div>
</body>
</html>

```