

**RANCANG BANGUN APLIKASI  
PENERIMAAN DAN PENGELUARAN MATERIAL GUDANG  
KE PRODUKSI MENGGUNAKAN PHP 5.4.7 DAN MYSQL  
5.5.27 PADA PT TETRA PAK STAINLESS EQUIPMENT  
JAKARTA**

**TUGAS AKHIR**

Diajukan Untuk Memenuhi Sebagian Syarat Penyelesaian Jenjang Diploma Empat (D-4) Program Studi Sistem Informasi pada Politeknik STMI Jakarta

**DISUSUN OLEH**

**M. OKA AFRIANDY**

**1309031**



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

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

**LEMBAR PENGESAHAN**

**JUDUL TUGAS AKHIR : RANCANG BANGUN APLIKASI  
PENERIMAAN DAN PENGELUARAN MATERIAL GUDANG KE  
PRODUKSI MENGGUNAKAN PHP 5.4.7 DAN MYSQL 5.5.27  
PADA PT TETRA PAK STAINLESS EQUIPMENT JAKARTA**

**DISUSUN OLEH :**

Nama : M. Oka Afriandy

NIM : 1309031

Program Studi : Sistem Informasi

Telah diuji oleh Tim Penguji Sidang Tugas Akhir Diploma Empat  
(D-4) Program Studi Sistem Informasi pada Politeknik STMI Jakarta pada hari  
Senin tanggal 17 Oktober 2016.

Menyetujui,

Dosen Pembimbing

Ketua Penguji

Drs. Jacob Saragih, MM  
NIP: 19404281986031002

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

Anggota Penguji

Anggota Penguji

Ulil Hamida, ST, MT  
NIP: 198103272005022001

Dedy Trisanto, S.Kom, MMSI  
NIP: 197805052005021002

## **POLITEKNIK STMI JAKARTA**

### **TANDA PERSETUJUAN DOSEN PEMBIMBING**

Judul Tugas Akhir : **Rancang Bangun Aplikasi Penerimaan dan Pengeluaran Material Gudang ke Produksi Menggunakan PHP 5.4.7 Dan MYSQL 5.5.27 Pada PT Tetra Pak Stainless Equipment Jakarta**

Disusun Oleh :

Nama Mahasiswa : M. Oka Afriandy  
NIM : 1309031  
Program Studi : Sistem Informasi  
Tanggal Seminar : Rabu, 7 September 2016  
Tanggal Sidang : Senin, 17 Oktober 2016  
Tanggal Lulus : Senin, 17 Oktober 2016

Telah diperiksa dan disetujui untuk diajukan dalam sidang Tugas Akhir Politeknik STMI Jakarta.

Menyetujui,

Jakarta, 28 September 2016

Dosen Pembimbing

Drs. Jacob Saragih, M.M.  
NIP. 195404281986031002

## **POLITEKNIK STMI JAKARTA**

### **TANDA PERSETUJUAN ASISTEN DOSEN PEMBIMBING**

Judul Tugas Akhir : **Rancang Bangun Aplikasi Penerimaan dan Pengeluaran Material Gudang ke Produksi Menggunakan PHP 5.4.7 Dan MYSQL 5.5.27 Pada PT Tetra Pak Stainless Equipment Jakarta**

Disusun Oleh :

Nama Mahasiswa : M. Oka Afriandy  
NIM : 1309031  
Program Studi : Sistem Informasi  
Tanggal Seminar : Rabu, 7 September 2016  
Tanggal Sidang : Senin, 17 Oktober 2016  
Tanggal Lulus : Senin, 17 Oktober 2016

Telah diperiksa dan disetujui untuk diajukan dalam sidang Tugas Akhir Politeknik STMI Jakarta.

Menyetujui,

Jakarta, 28 September 2016  
Asisten Dosen Pembimbing

Ahlan Ismono, S.Kom  
NIP. 19790107 200604 1002



**LEMBAR BIMBINGAN TUGAS AKHIR**

Nama : M. Oka Afriandy  
NIM : 1309031  
Judul Tugas Akhir : Rancang Bangun Aplikasi Penerimaan dan Pengeluaran Material Gudang Ke Produksi Menggunakan PHP 5.4.7 dan My SQL 5.5.27 Pada PT Tetra Pak Stainless Equipment Jakarta  
Pembimbing : Drs. Jacob Saragih, M.M.

Tanggal	Keterangan	Paraf
11 Juni 2015	Proposal Tugas Akhir	
7 Juli 2015	Ganti Judul	
28 Juli 2015	BAB I - III	
05 Agustus 2015	Revisi BAB I – III dan Pengajuan BAB IV	
09 Agustus 2015	Revisi BAB IV	
19 September 2015	Revisi BAB IV dan Pengajuan BAB V	
20 oktober 2015	Revisi BAB V dan Program	
22 Maret 2016	Pengajuan BAB 5 revisi Program	
18 July 2016	Pembahasan Full BAB dan Program	
29 July 2016	Revisi Pembahasan Full BAB dan Program	

Mengetahui,  
Ketua Prodi  
Sistem Informasi Industri Otomotif

Dosen Pembimbing

(Drs. Jacob Saragih, MM.)  
NIP: 195404281986031002

(Drs. Jacob Saragih, MM.)  
NIP: 197906052006041002





**LEMBAR BIMBINGAN TUGAS AKHIR**

Nama : M. Oka Afriandy  
NIM : 1309031  
Judul Tugas Akhir : Rancang Bangun Aplikasi Penerimaan dan Pengeluaran Material Gudang Ke Produksi Menggunakan PHP 5.4.7 dan My SQL 5.5.27 Pada PT Tetra Pak Stainless Equipment Jakarta  
Pembimbing : Ahlan Ismono, S.Kom

Tanggal	Keterangan	Paraf
12 Juni 2015	Proposal Tugas Akhir	
20 January 2016	BAB I - III	
26 January 2016	Revisi BAB 1- III	
17 Februari 2016	Revisi BAB I – III dan Pengajuan BAB IV	
09 Agustus 2015	Revisi BAB IV	
15 Maret 2016	Revisi BAB IV dan Pengajuan BAB V	
19 April 2016	Revisi BAB IV dan Program	
22 Maret 2016	Pembahasan BAB V revisi Program	
5 April 2016	Revisi BAB V	
8 Juni 2016	Pembahasan Full BAB dan Program	

Mengetahui,  
Ketua Prodi  
Sistem Informasi Industri Otomotif

Asisten Dosen Pembimbing

(Drs. Jacob Saragih, MM.)  
NIP: 195404281986031002

(Ahlan Ismono, S.Kom)  
NIP: 19790107 200604 1002



## LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : M. Oka Afriandy

NIM : 1309031

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

**“RANCANG BANGUN APLIKASI PENERIMAAN DAN PENGELUARAN MATERIAL GUDANG KE PRODUKSI MENGGUNAKAN PHP 5.4.7 DAN MYSQL 5.5.27 PADA PT TETRA PAK STAINLESS EQUIPMENT JAKARTA”.**

**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, 28 September 2016

Yang Membuat Pernyataan,

M.Oka Afriandy

## KATA PENGANTAR

*Alhamdulillah* segala puji syukur penulis panjatkan kehadirat Allah SWT yang telah melimpahkan nikmat-Nya, terutama nikmat iman dan sehat sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul

**“RANCANG BANGUN APLIKASI PENERIMAAN DAN PENGELUARAN MATERIAL GUDANG KE PRODUKSI MENGGUNAKAN PHP 5.4.7 DAN MYSQL 5.5.27 PADA PT TETRA PAK STAINLESS EQUIPMENT JAKARTA”**.

Penyusunan Tugas Akhir ini dilakukan guna memenuhi salah satu persyaratan akademis untuk menyelesaikan Program Diploma Empat (D-4) Program Studi Sistem Informasi pada Politeknik STMI Jakarta.

Kelancaran dan kesuksesan dalam penyelesaian Tugas Akhir ini tak luput dari bantuan yang telah diberikan oleh beberapa pihak baik berupa bimbingan, saran, dan do'a sehingga dalam kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Allah SWT atas rahmat serta kemudahan yang diberikan.
2. Orang tua yang selalu memberikan dukungan, kasih sayang serta do'a restu, baik moral maupun material membantu dalam penyelesaian Tugas Akhir ini.
3. Bapak Dr. Mustofa, ST, MT selaku Direktur Politeknik STMI Jakarta.
4. Bapak Drs. Jacob Saragih, M.M. selaku ketua program studi sistem informasi. Dan juga selaku dosen pembimbing yang telah bersedia membantu memberikan bimbingan serta pengarahan kepada penulis untuk menyelesaikan Tugas Akhir ini.
5. Bapak Ahlan Ismono, S.Kom selaku asisten dosen pembimbing yang telah memberikan arahan, bimbingan serta penjelasan teknis dalam penyusunan Tugas Akhir.
6. Seluruh dosen Politeknik STMI Jakarta khususnya Program Studi Sistem Informasi yang telah memberikan ilmunya guna menambah pengetahuan dan pengalaman yang berguna bagi penulis.

7. Ibu Lia dan Ibu NurJannah, yang telah memberikan kesempatan dan bantuannya untuk bisa melakukan Praktik Kerja Lapangan di PT Tetra Pak Stainless Equipment.
8. Bapak Rudi selaku pembimbing selama Praktik Kerja Lapangan di PT Tetra Pak Stainless Equipment yang telah memberikan petunjuk serta informasi sehingga Tugas Akhir ini dapat diselesaikan.

Dengan segala kemampuan dan keterbatasan, penulis menyadari segala kekurangan yang ada di dalam penyusunan Tugas Akhir ini, karena itu penulis sangat mengharapkan segala kritik atau saran yang dapat membangun dari semua pihak.

**Wassalaamu'alaikum wr.wb.**

Jakarta, September 2016

Penulis

## ABSTRAK

PT Tetra Pak Stainless Equipment Jakarta merupakan perusahaan multinasional industri manufaktur di Indonesia yang bergerak di bidang pembuatan alat proses, pembungkus makanan, minuman, kosmetik dan obat-obatan. Adapun produk yang dibuat adalah seperti *tank, vessel, conveyor, tube heat exchanges*. Kemudian dalam memproduksi tentunya harus tepat waktu maka dari itu saat pengeluaran material dari gudang untuk produksi harus tersistem dan tidak lama agar dapat meminimalisir waktu kegiatan produksi. Untuk menunjang proses penerimaan dan pengeluaran material gudang ke produksi seperti melakukan permintaan material dari produksi ke gudang membutuhkan waktu lama karena permintaan material tersebut masih dilakukan dengan cara manual yang ditulis di formulir kertas dan bagian produksi mengantarkan ke gudang lalu membawa sendiri ke bagian produksi, dan barulah bagian gudang menginput data ke dalam *excel* hal tersebut dapat menyebabkan hal hal yang tidak diperlukan seperti waktu pengambilan menjadi lama karena adanya penyimpangan dalam pengambilan material, terjadinya kertas *form* permintaan material rusak dan hilang, dan juga ada bagian produksi yang letaknya berjauhan dengan gudang. Untuk perancangan sistem informasi yang diusulkan, metode pengembangan sistem yang digunakan ialah *evolutionary prototype* dengan melalui tahap-tahap yang digunakan mulai dari pemodelan sistem menggunakan *Unified Modeling Language* (UML), perancangan data menggunakan *class diagram*, perancangan HIPO, perancangan *flowchart* program yang diusulkan, perancangan *interface* program, dan pembuatan *prototype* menggunakan bahasa pemrograman PHP 5.4.7 serta *MySQL 5.5.27* sebagai basis datanya. Oleh karena itu dengan adanya aplikasi ini sistem penerimaan dan pengeluaran material gudang ke produksi pada PT Tetrapak Stainless Jakarta tidak lagi menggunakan *Microsoft Excel* dan di ganti dengan yang lebih baik yaitu menggunakan *My SQL*. Waktu yang dibutuhkan dalam proses pengeluaran material berdasarkan permintaan dari bagian produksi lebih cepat dari sebelumnya. Dengan menggunakan aplikasi ini mengurangi penggunaan kertas yang dapat terselip atau hilang. Proses pembuatan permintaan dan pengeluaran material juga laporan pengeluaran material secara langsung dapat diisi ke *database*.

**Kata Kunci** : Sistem Informasi, Penerimaan dan Pengeluaran Material Gudang ke Produksi, *Prototype Evolutioner*, *Unified Modeling Language* (UML), PHP5.4.7, *MySQL 5.5.27*

# DAFTAR ISI

Halaman

**HALAMAN JUDUL**

**LEMBAR PENGESAHAN**

**TANDA PERSETUJUAN DOSEN PEMBIMBING**

**TANDA PERSETUJUAN ASISTEN DOSEN PEMBIMBING**

**LEMBAR BIMBINGAN TUGAS AKHIR**

**LEMBAR PERNYATAAN KEASLIAN**

**ABSTRAK**

Kata Pengantar .....	i
Daftar Isi .....	iii
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 .....	3
1.4 Batasan Masalah.....	3
1.5 Manfaat Penelitian .....	3
1.6 Sistematika Penulisan .....	4
<b>BAB II LANDASAN TEORI .....</b>	<b>6</b>
2.1 Rancang Bangun .....	6
2.2 Aplikasi .....	6
2.2.1 Klasifikasi Aplikasi.....	7
2.3 Konsep Dasar Sistem .....	7

2.3.1	Karakteristik Sistem.....	8
2.3.2	Klasifikasi Sistem.....	11
2.4	Pengembangan Sistem .....	12
2.4.1	Pengertian <i>Prototype</i> .....	13
2.4.2	<i>Evolution Prototype</i> .....	15
2.5	Pengertian Informasi .....	16
2.5.1	Siklus Informasi .....	16
2.6	Pengertian Sistem Informasi .....	18
2.6.1	Komponen Sitem Informasi .....	18
2.7	Material .....	19
2.8	Penerimaan Material .....	20
2.9	Pengeluaran Material .....	22
2.10	Gudang ( <i>Warehouse</i> ) .....	24
2.10.1	Tipe Dan Jenis Gudang.....	24
2.10.2	Klasifikasi Persediaan Dalam Gudang.....	25
2.11	Diagram Alir .....	30
2.12	Unified Modeling Language.....	32
2.12.1	<i>Use Case Diagram</i> .....	33
2.12.2	<i>Activity Diagram</i> .....	34
2.12.3	<i>Sequence Diagram</i> .....	35
2.12.4	<i>Class Diagram</i> .....	37
2.12.5	<i>Component Diagram</i> .....	38
2.12.6	<i>Deployment Diagram</i> .....	38
2.13	Hierarchy plus Input-Process-output.....	39
2.14	<i>Hypertext Preprocessor</i> (PHP) .....	40
2.14.1	KemampuanPHP .....	42
2.15	MySQL.....	42

2.16	XAMPP.....	43
2.17	<i>Macromedia Dreamweaver</i> .....	43
<b>BAB III</b>	<b>METODOLOGI PENELITIAN</b> .....	<b>45</b>
3.1	Pengumpulan Data .....	45
3.1.1	Jenis dan Sumber Data.....	45
3.2	Metode Pengumpulan Data.....	46
3.3	Identifikasi Masalah.....	46
3.4	Tujuan Penelitian .....	47
3.5	Batasan Masalah.....	47
3.6	Metode Pengembangan Sistem .....	48
3.7	Kesimpulan Dan Saran.....	48
3.8	Kerangka Penelitian .....	48
<b>BAB IV</b>	<b>PENGUMPULAN DAN PENGOLAHAN DATA</b> .....	<b>53</b>
4.1	Sejarah Singkat Perusahaan .....	53
4.1.1	Profil Perusahaan.....	55
4.1.2	Visi dan Misi Perusahaan.....	56
4.1.3	Struktur Organisasi Perusahaan.....	56
4.2	Material Pada gudang Tetra Pak Stainless Equipment .....	60
4.2.1	<i>Stainless Steel 304</i> .....	61
4.2.2	<i>Stainless Steel 316</i> .....	62
4.3	Sistem Yang Berjalan.....	63
4.4	Data Terkait.....	66
4.5	Penggambaran Sistem Pengeluaran Bahan Baku Produksi ke Gudang.....	67
<b>BAB V</b>	<b>ANALISIS DAN PEMBAHASAN</b> .....	<b>73</b>
5.1	Analisis Kebutuhan Sistem.....	73
5.2	Perancangan <i>Flowmap</i> Sitem Usulan.....	74

5.3	Analisis Sistem Usulan .....	75
5.3.1	<i>Use Case Diagram</i> Usulan.....	76
5.3.2	<i>Activity Diagram</i> .....	83
5.3.3	<i>Sequence Diagram</i> .....	91
5.3.4	<i>Class Diagram yang Diusulkan</i> .....	101
5.3.5	Kamus Data.....	102
5.3.6	<i>Deployment diagram yang diusulkan</i> .....	106
5.4	Perancangan Sistem .....	107
5.4.1	<i>Hierarchy plus Input-Process-Output (HIPO)</i> . Aplikasi.....	108
5.5	Flowchart Aplikasi Usulan .....	108
5.6	Perancangan <i>Interface</i> Aplikasi Usulan.....	109
5.7	Analisis <i>Software</i> dan <i>Hardware</i> .....	131
<b>BAB VI</b>	<b>KESIMPULAN DAN SARAN</b> .....	133
6.1	Kesimpulan .....	133
6.2	Saran.....	133
	<b>DAFTAR PUSTAKA</b> .....	134
	<b>LAMPIRAN</b> .....	138

## DAFTAR GAMBAR

	Halaman
Gambar II.1 Karakteristik Sistem .....	10
Gambar II.2 Mekanisme Pengembangan Sistem Dengan <i>Prototype</i> .....	14
Gambar II.3 <i>Evolution Prototype Model</i> .....	16
Gambar II.4 Siklus Informasi.....	18
Gambar II.5 Diagram UML .....	32
Gambar II.6 <i>Visual Table Of contents</i> .....	40
Gambar III.1 Kerangka Penelitian .....	49
Gambar III.2 Kerangka Penelitian (lanjutan).....	50
Gambar IV.1 Struktur Organisasi PT Tetra Pak Stainless Equipment .....	57
Gambar IV.2 <i>Stainless Steel 304</i> .....	62
Gambar IV.3 <i>Stainless Steel 316</i> .....	63
Gambar IV.4 <i>FlowMap</i> Sistem Yang Berjalan .....	64
Gambar IV.5 <i>Delivery Order</i> .....	66
Gambar IV.6 Surat Permintaan Material Produksi ke Gudang.....	67
Gambar IV.7 <i>Use Case</i> Sistem Yang Berjalan .....	67
Gambar V.1 <i>Flowmap</i> Proses Penerimaan dan Pengeluaran Material Gudang ke Produksi usulan .....	75
Gambar V.2 <i>Use Case</i> Diagram Proses Penerimaan dan Pengeluaran MaterialGudang ke Produksi usulan.....	77
Gambar V.3 <i>Activity Diagram</i> Proses Login .....	83
Gambar V.4 <i>Activity Diagram</i> Mengelola Data Master <i>Supplier</i> .....	84
Gambar V.5 <i>Activity Diagram</i> Mengelola Data Master Material .....	85
Gambar V.6 <i>Activity Diagram</i> Mengelola Data <i>Receipt</i> Material .....	86
Gambar V.7 <i>Activity Diagram</i> Melakukan Permintaan Material.....	86
Gambar V.8 <i>Activity Diagram</i> Menerima <i>Request</i> Material.....	87
Gambar V.9 <i>Activity Diagram</i> Menerima Material .....	88

Gambar V.10	<i>Activity Diagram</i> Melakukan Konfirmasi Material .....	89
Gambar V.11	<i>Activity Diagram</i> Membuat <i>Retur</i> Material.....	90
Gambar V.12	<i>Activity Diagram</i> Melakukan <i>Retur</i> Material.....	90
Gambar V.12	<i>Activity Diagram</i> Mencetak Sura <i>Retur</i> .....	91
Gambar V.14	<i>Sequence Diagram</i> Login User .....	92
Gambar V.15	<i>Sequence Diagram</i> Mengelola Data Master Material .....	93
Gambar V.16	<i>Sequence Diagram</i> Mengelola Data Master <i>Supplier</i> .....	94
Gambar V.17	<i>Sequence Diagram</i> Mengelola Data <i>Receipt</i> Material .....	95
Gambar V.18	<i>Sequence Diagram</i> Membuat <i>Retur</i> Material.....	96
Gambar V.19	<i>Sequence Diagram</i> Mencetak <i>Retur</i> Material .....	97
Gambar V.20	<i>Sequence Diagram</i> Menerima <i>Request</i> Material.....	98
Gambar V.21	<i>Sequence Diagram</i> Melakukan <i>Approval</i> Material .....	99
Gambar V.22	<i>Sequence Diagram</i> Melakukan <i>Request</i> Material .....	100
Gambar V.23	<i>Sequence Diagram</i> Melakukan Konfirmasi Material.....	100
Gambar V.24	<i>Sequence Diagram</i> Melihat dan Mengecek Stock Report .....	101
Gambar V.25	<i>Class Diagram</i> yang Diusulkan.....	102
Gambar V.26	<i>Deployment Diagram</i> yang Diusulkan .....	107
Gambar V.27	HIPO Sistem Usulan .....	108
Gambar V.28	HIPO Flowchart Aplikasi Usulan.....	109
Gambar V.29	Rancangan <i>Form Login</i> .....	110
Gambar V.30	Rancangan <i>Form Menu</i> .....	111
Gambar V.31	Rancangan <i>Form Warehouse</i> .....	112
Gambar V.32	Rancangan <i>Form Receipt</i> Material.....	113
Gambar V.33	Rancangan <i>Add Reception</i> .....	114
Gambar V.35	Rancangan <i>Form Retur</i> Material .....	115
Gambar V.36	Rancangan <i>Form Add Retur</i> Material .....	116
Gambar V.37	Rancangan <i>Form Master</i> Material .....	117
Gambar V.38	Rancangan <i>Form Add</i> Material .....	118
Gambar V.39	Rancangan <i>Form Request Approval</i> .....	119
Gambar V.40	Rancangan <i>Request Procces</i> .....	120

Gambar V.41 Rancangan <i>Stock Report</i> .....	121
Gambar V.42 Rancangan <i>Form Stock</i> .....	122
Gambar V.43 Rancangan Output Report Materials .....	123
Gambar V.44 Rancangan <i>Request Material</i> .....	125
Gambar V.45 Rancangan <i>Form Add Request</i> .....	126
Gambar V.46 Rancangan <i>Form Stock Report Purchasing</i> .....	127
Gambar V.47 Rancangan <i>Form Supplier List</i> .....	128
Gambar V.48 Rancangan <i>Form Add Suppliers</i> .....	129
Gambar V.49 Rancangan <i>Form Manage User</i> .....	130
Gambar V.50 Rancangan <i>Form Add User</i> .....	131

## DAFTAR TABEL

	Halaman
Tabel II.1 Simbol Diagram Alir ( <i>flowchart</i> ) .....	30
Tabel II.2 Simbol-Simbol <i>Use Case Diagram</i> .....	33
Tabel II.3 Simbol-simbol <i>Activity Diagram</i> .....	34
Tabel II.4 Simbol-Simbol <i>Sequence Diagram</i> .....	35
Tabel II.5 Simbol-Simbol <i>Class Diagram</i> .....	37
Tabel II.6 Simbol-Simbol <i>Component Diagram</i> .....	38
Tabel II.7 Simbol-Simbol <i>Deployment Diagram</i> .....	38
Tabel IV.1 Definisi Aktor <i>Use Case</i> Sistem yang Berjalan.....	78
Tabel IV.2 Definisi <i>Use Case</i> Sistem yang Berjalan .....	71
Tabel V.1 Identifikasi Kebutuhan Sistem.....	73
Tabel V.1 Identifikasi Kebutuhan Sistem (lanjutan) .....	74
Tabel V.2 Definisi Aktor <i>Use Case Diagram</i> Sitem Usulan.....	78
Tabel V.3 Definisi Aktor <i>Use Case Diagram</i> Login User .....	78
Tabel V.4 <i>Use Case Description</i> Mengelola Data <i>Master Supplier</i> .....	78
Tabel V.5 <i>Use Case Description</i> Mengelola Data Master Material .....	79
Tabel V.6 <i>Use Case Description</i> Mengelola <i>Receipt</i> Material.....	79
Tabel V.7 <i>Use Case Description</i> Melakukan <i>Approval</i> Material.....	80
Tabel V.8 <i>Use Case Description</i> Membuat <i>Retur</i> Material.....	81
Tabel V.9 <i>Use Case Description</i> Mencetak <i>Retur</i> Material .....	81
Tabel V.10 <i>Use Case Description</i> Melakukan <i>Request</i> Material .....	81
Tabel V.11 <i>Use Case Description</i> Menerima Material.....	82
Tabel V.12 <i>Use Case Description</i> Melihat dan Mengecek <i>Stock Report</i> .....	82
Tabel V.13 Spesifikasi Tabel <i>Materials</i> .....	103
Tabel V.14 Spesifikasi Tabel <i>Receipt</i> .....	103
Tabel V.15 Spesifikasi Tabel <i>Receipt detail</i> .....	103
Tabel V.16 Spesifikasi Tabel <i>Request</i> .....	104

Tabel V.17 Spesifikasi Tabel Request_detail .....	104
Tabel V.18 Spesifikasi Tabel_return.....	105
Tabel V.19 Spesifikasi Tabel Return_detail .....	105
Tabel V.20 Spesifikasi Tabel Stocks .....	105
Tabel V.21 Spesifikasi Tabel Suppliers .....	106

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Teknologi informasi yang semakin maju dan berkembang di dunia bisnis dan industri sudah menjadi suatu kebutuhan yang sangat penting bagi dunia bisnis dan industri karena akan fungsi dari teknologi tersebut dapat mempermudah dan mempercepat dalam mengakses, mengelola informasi secara terkomputerisasi. Hal ini karena pengolahan data secara terkomputerisasi dapat memberikan kontribusi yang besar bagi kinerja suatu perusahaan untuk meningkatkan kualitas dan kuantitas produk yang dihasilkan. Pengolahan data secara terkomputerisasi lebih efektif dan efisien dibandingkan dengan manual. Pengolahan data secara terkomputerisasi merupakan serangkaian kegiatan untuk menyediakan informasi yang mencakup pengumpulan, pemrosesan, penyimpanan, dan pencarian data sehingga menghasilkan informasi yang memenuhi persyaratan kelengkapan, kemutakhiran yang tangguh yang dapat digunakan sebagai alat pendukung pengambilan keputusan secara efektif.

Pergudangan di dalam suatu perusahaan, merupakan suatu hal atau kegiatan yang cukup penting. Bahkan di dalam berbagai macam pembicaraan, dikatakan bahwa di setiap perusahaan dibutuhkan gudang atau pergudangan untuk menyimpan material demi kelancaran suatu proses produksi di pabrik maupun kegiatan di suatu perusahaan.

PT Tetra Pak Stainless Equipment adalah perusahaan multinasional yang menawarkan solusi terpadu pemrosesan dan pengemasan makanan dan minuman. Perusahaan ini membuat kemasan serta mesin pengisi dan proses untuk produk susu, minuman, keju, es krim, dan juga membuat peralatan seperti akumulator, konveyor, *tank*, *vessel* dan mesin pelapis plastik. Dalam menghasikan produknya, PT Tetra Pak Stainless Equipment membutuhkan material yang dipesan langsung dari *supplier*. Dan material utama tersebut adalah berupa plat baja kualitas terbaik.

Pada saat ini dalam proses penerimaan dan pengeluaran material gudang ke produksi pada PT Tetra Pak Stainless Equipment masih secara manual sehingga kinerjanya kurang efisien, karena untuk pengolahan data penerimaan material masih menggunakan *Microsoft Excel* dan belum menggunakan perangkat lunak sistem manajemen basis data SQL dan pengeluaran material berdasarkan permintaan dari bagian produksi masih menggunakan *form* kertas yang membutuhkan waktu lebih dalam prosesnya. Untuk mengatasi masalah tersebut dibutuhkan sistem informasi yang dapat membantu perusahaan dalam mengatasi permasalahan tersebut agar menjadi lebih efektif dari sebelumnya.

Berdasarkan permasalahan di atas, solusi yang dapat diusulkan yaitu merancang bangun aplikasi penerimaan dan pengeluaran material gudang ke produksi menggunakan PHP 5.4.7 dan MySQL 5.5.27 Pada PT Tetra Pak Stainless Equipment untuk membantu perusahaan agar kinerja produksi dan gudang menjadi efektif dan efisien.

## **1.2 Pokok Permasalahan**

Permasalahan yang terjadi pada PT Tetra Pak Stainless Equipment adalah sebagai berikut:

1. Proses pengolahan data penerimaan dan pengeluaran material di PT Tetra Pak Stainless Equipment Jakarta masih menggunakan *Microsoft Excel*, sehingga masih menimbulkan redundansi data.
2. Proses pengeluaran material berdasarkan permintaan dari bagian produksi untuk kebutuhan produksi masih menggunakan *form* permintaan material yang ditulis kemudian diantarkan ke bagian gudang dan harus menunggu material dikeluarkan untuk dibawa lagi ke bagian produksi. Karena itu sering terjadi *form* atau kertas permintaan material hilang, rusak atau terselip.
3. Proses permintaan material dari produksi ke gudang masih belum terintegrasi sehingga membutuhkan banyak waktu.

### 1.3 Tujuan Penelitian

Tujuan dari penelitian Tugas Akhir ini adalah sebagai berikut:

1. Menerapkan penggunaan database MySQL pada PT Tetra Pak Stainless Equipment Jakarta untuk menghilangkan redundansi.
2. Membuat *form* secara terkomputerisasi sehingga dapat mengurangi penggunaan kertas.
3. Membuat aplikasi penerimaan dan pengeluaran material gudang ke produksi menggunakan PHP 5.4.7 dan MySQL 5.5.27 Pada PT Tetra Pak Stainless Equipment untuk membantu sistem pergudangan dalam penerimaan dan pengeluaran material gudang ke produksi untuk kebutuhan produksi.

### 1.4 Batasan Masalah

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

1. Tempat pengamatan adalah PT Tetra Pak Stainless Equipment, Pulo Gebang Jakarta Timur.
2. Pengamatan dilakukan di bagian gudang penyimpanan material pada PT Tetra Pak Stainless Equipment.
3. Pembatasan masalah hanya mengamati proses penerimaan material dari *supplier* berdasarkan data *delivery order* dan pengeluaran material gudang ke produksi di PT Tetra Pak Stainless Equipment Jakarta.

### 1.5 Manfaat Tugas Akhir

Adapun manfaat yang dapat diperoleh dari penulisan Tugas Akhir ini sebagai berikut:

1. Bagi Diri Sendiri  
Dapat memberikan pengalaman kepada penulis dalam menganalisa, mengamati serta dapat menggambarkan permasalahan yang terjadi di dalam suatu sistem dan diharapkan dapat memberikan solusi permasalahannya dengan melakukan analisis, merancang dan membangun sistem informasi.
2. Bagi Perusahaan

Dapat mengimplementasikan sistem di perusahaan untuk membantu kinerja perusahaan karena dengan adanya sistem informasi produksi yang terintegrasi.

### 3. Bagi Pihak Lain

Manfaat yang dapat diberikan adalah sebagai referensi kepada penulis lain yang melakukan penelitian serupa.

## 1.6 Sistematika Penulisan

Agar dapat mempermudah perumusan dan pemecahan masalah yang akan dibahas pada Tugas Akhir ini maka penulis menguraikan tahapan-tahapan dalam penyusunan laporan ini sebagai berikut:

### **BAB I PENDAHULUAN**

Bab ini memuat latar belakang masalah, pokok permasalahan tujuan penelitian, batasan masalah, manfaat Tugas Akhir dan sistematika penulisan.

### **BAB II LANDASAN TEORI**

Bab ini membahas tentang definisi sistem, sistem informasi, proses/sistem produksi, perencanaan kebutuhan material, *flowchart*, *Unified Modeling Language (UML)*, *Hypertext Preprocessor (PHP)*, MySQL, XAMPP dan Macromedia Dreamweaver 8 sebagai alat bantu pembuatan *interface*.

### **BAB III METODOLOGI PENELITIAN**

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

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

Bab ini menguraikan tentang hasil pengamatan selama penelitian yang telah dilakukan seperti informasi mengenai perusahaan, produk, sistem produksi, analisis sistem yang berjalan dan bentuk dokumen yang terlibat dalam proses bisnis produksi.

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

Bab ini berisi tentang analisis data dan permasalahan yang ada di lapangan serta pembahasannya dengan menggunakan *Flowchart*, *Unified Modelling Process* (UML), *Hypertext Preprocessor* (PHP) dan MySQL.

## **BABVI PENUTUP**

Bab ini berisikan tentang kesimpulan dari hasil penelitian yang dilakukan serta mengemukakan 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 Aplikasi**

Ada banyak pengertian mengenai aplikasi oleh para ahli yaitu:

1. Aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas (Buyens, 2001).
2. Aplikasi adalah sistem lengkap yang mengerjakan tugas spesifik (Post, 1999).
3. Menurut Dhanta (2009), aplikasi (*application*) adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Microsoft Word*, *Microsoft Excel*.

Dari pengertian diatas, dapat disimpulkan bahwa aplikasi merupakan *software* yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data. Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau *suite* aplikasi (*application suite*). Contohnya adalah *Microsoft Office* yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna

untuk mempelajari dan menggunakan tiap aplikasi. Sering kali mereka memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

### **2.2.1 Klasifikasi Aplikasi**

Aplikasi dapat digolongkan menjadi beberapa kelas, antara lain:

- a. Perangkat lunak perusahaan
- b. Perangkat lunak infrastruktur perusahaan
- c. Perangkat lunak informasi kerja
- d. Perangkat lunak media dan hiburan
- e. Perangkat lunak pendidikan
- f. Perangkat lunak pengembangan media
- g. Perangkat lunak rekayasa produk

Pada pengertian umumnya, aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya.

## **2.3 Konsep Dasar Sistem**

Kata *system* sendiri berasal dari bahasa Latin “*systema*” dan bahasa Yunani “*sustēma*” yang artinya adalah himpunan bagian atau unsur yang saling berhubungan secara teratur untuk mencapai tujuan bersama. Untuk beberapa sistem yang diintisarikan dari beberapa sumber referensi, mendefinisikan bahwa sistem:

- a. Menurut Ludwig (1991) “Sistem adalah seperangkat unsur yang saling berhubungan dan saling mempengaruhi dalam satu lingkungan tertentu”.
- b. Menurut Sutanta (2003) “Sistem adalah sekumpulan hal atau kegiatan atau elemen atau subsistem yang saling bekerja sama atau yang dihubungkan dengan cara-cara tertentu sehingga membentuk satu kesatuan untuk melaksanakan suatu fungsi guna mencapai suatu tujuan”.
- c. Menurut Jogiyanto (2005) “Sistem adalah kumpulan dari elemen – elemen yang berinteraksi untuk mencapai suatu tujuan tertentu ”Suatu sistem

dapat terdiri dari sistem-sistem bagian (*subsystem*). Sebagai contoh, sistem komputer dapat terdiri dari subsistem perangkat keras (*hardware*) dan perangkat lunak (*software*). Masing-masing subsistem dapat terdiri dari subsistem-subsistem yang lebih kecil atau terdiri dari komponen-komponen. Subsistem perangkat keras (*hardware*) dapat terdiri dari alat masukan, alat pemroses, alat keluaran dan simpanan luar. Subsistem-subsistem saling berinteraksi dan saling berhubungan membentuk satu kesatuan sehingga tujuan atau sasaran sistem tersebut dapat tercapai. Interaksi dari subsistem-subsistem sedemikian rupa sehingga dicapai suatu kesatuan yang terpadu atau terintegrasi. (Jogiyanto, 2005).

Berdasarkan pengertian diatas, maka dapat disimpulkan bahwa sistem adalah suatu kesatuan yang kompleks yang terdiri dari kumpulan elemen-elemen yang berbeda yang saling berinteraksi untuk mencapai suatu tujuan dan sasaran tertentu. Suatu elemen dikatakan saling membangun jika elemen yang berbeda-beda tersebut saling bekerja sama, dan jika elemen-elemen tersebut masih bekerja secara individu tanpa mempengaruhi aktivitas lain, maka kumpulan elemen tersebut masih belum dapat dikatakan sebagai suatu sistem.

### **2.3.1 Karakteristik Sistem**

Menurut Jogiyanto (2005) Karakteristik sistem adalah sistem yang mempunyai komponen (*components*), batas sistem (*boundary*), lingkungan sistem (*environments*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolah (*process*), dan sasaran (*objectives*) atau tujuan (*goal*). Berikut ini merupakan karakteristik sebuah sistem:

#### **1. Komponen Sistem**

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerjasama 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 untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

## 2. Batas Sistem

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

## 3. Lingkungan Luar Sistem

Lingkungan luar (*environment*) dari suatu sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat 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 (*interface*) 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 (*output*) dari satu subsistem akan menjadi masukan (*input*) untuk subsistem yang lainnya dengan melalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

## 5. Masukan Sistem

Masukan (*input*) 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 mendapatkan keluaran.

## 6. Keluaran Sistem

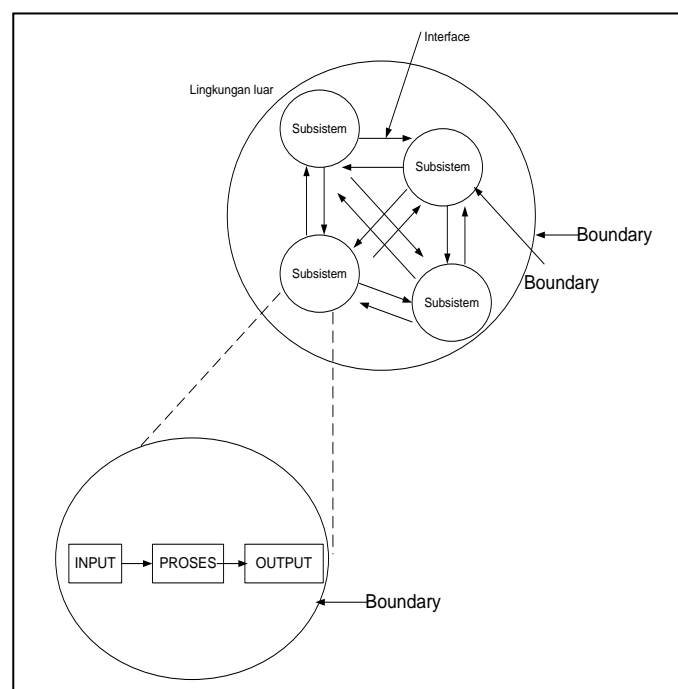
Keluaran (*output*) adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain.

## 7. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolahan yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa *material* dan bahan-bahan lain menjadi keluaran berupa material jadi.

## 8. Sasaran Sistem

Suatu sistem pasti memiliki 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. (Jogiyanto, 2005)



Gambar II.1 Karakteristik Sistem

Sumber: jogiyanto 2005

### 2.3.2 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang :

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 dan 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, misalnya sistem perputaran bumi. Sistem buatan manusia adalah sistem yang dirancang oleh manusia dan melibatkan interaksi antara manusia dengan mesin, misalnya sistem informasi akuntansi.
3. Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*). Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti sehingga keluarannya dapat diramalkan. Sistem komputer adalah contoh dari sistem tertentu yang tingkah lakunya dapat dipastikan melalui program yang dijelankannya. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.
4. Sistem diklasifikasikan sebagai sistem tertutup (*closed system*) dan sistem terbuka (*open system*). Sistem tertutup adalah sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak diluarnya.

Pada kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system*. 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, sehingga harus memiliki sistem pengendalian yang baik (Jogiyanto, 2005) .

#### **2.4 Pengembangan Sistem**

Menurut Rosa dan Shalahuddin (2011), proses perancangan atau pengembangan sistem informasi dimulai dari pemilihan konsep sampai dengan implementasinya yang biasa disebut dengan istilah *Software Development Life Cycle* atau dapat disebut juga *System Development Life Cycle (SDLC)*. 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-tahapan yang ada pada *System Development Life Cycle* secara umum adalah sebagai berikut:

1. Inisiasi (*initiation*)  
Tahap ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak.
2. Pengembangan konsep sistem (*system concept development*)  
Mendefinisikan lingkup konsep termasuk dokumen lingkup sistem analisis manfaat biaya, manajemen rencana dan pembelajaran kemudahan sistem.
3. Perencanaan (*planning*)  
Mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya. Menyediakan dasar untuk mendapatkan sumber daya (*resources*) yang dibutuhkan untuk memperoleh solusi.
4. Analisis kebutuhan (*requirements analysis*)  
Menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *users*sertamembuat dokumen-dokumen kebutuhan fungsional.

5. Desain (*design*)

Mentransformasikan kebutuhan *detail* menjadi kebutuhan yang sudah lengkap, dokumen desain sistem fokus pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan.

6. Pengembangan (*development*)

Mengkonversi desain ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan, membuat basis data dan mempersiapkan prosedur kasus pengujian, mempersiapkan berkas atau *file* pengujian, pengkodean, pengkompilasian, memperbaiki dan membersihkan program serta peninjauan pengujian.

7. Integrasi dan pengujian (*integration and test*)

Mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional. Dengan diarahkan oleh staf penjamin kualitas (*quality assurance*) dan *user* sehingga menghasilkan laporan analisis pengujian.

8. Implementasi (*implementation*)

Termasuk pada persiapan implementasi, implementasi perangkat lunak pada lingkungan produksi (lingkungan luar *user*) dan menjalankan resolusi dari permasalahan yang teridentifikasi dari fase integrasi dan pengujian.

9. Operasi dan pemeliharaan (*operations and maintenance*)

Mendeskripsikan pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan produksi (lingkungan pada *user*) termasuk implementasi akhir dan masuk pada proses peninjauan.

10. Disposisi (*disposition*)

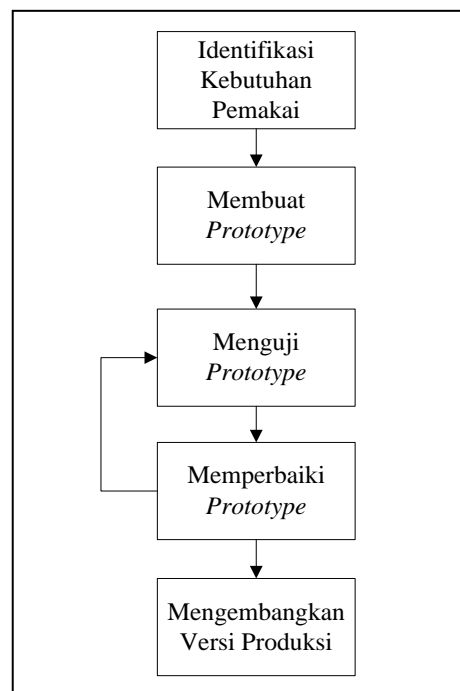
Mendeskripsikan aktivitas akhir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktifitas *user*.

#### 2.4.1 Pengertian *Prototype*

Menurut McLeod & Schell (2008), *prototype* adalah suatu versi dari sebuah sistem potensial yang memberikan ide bagi para pengembang dan calon

pengguna, bagaimana sistem akan berfungsi dalam bentuk yang telah selesai. Dasar pemikirannya adalah membuat *prototype* secepat mungkin, bahkan dalam waktu semalam lalu memperoleh umpan balik dari pengguna yang akan memungkinkan *prototype* tersebut diperbaiki kembali dengan sangat cepat.

Mekanisme pengembangan sistem dengan *prototype* dapat dilihat pada Gambar II.2.



Gambar II.2 Mekanisme Pengembangan Sistem Dengan *Prototype*  
(Sumber:McLeod & Schell, 2008)

Keterangan:

1. Identifikasi kebutuhan pemakai  
Tahap ini merupakan tahap dimana pengembang dan pemakai bertemu. Pada tahap ini juga pemakai akan menjelaskan kebutuhan sistemnya.
2. Membuat *prototype*  
Pada tahap ini, pengembang akan mulai membuat *prototypesesuai* dengan kebutuhan pemakai.
3. Menguji *prototype*  
Pada tahap ini, pemakai akan menguji *prototypedan* memberikan kritikan atau saran.

#### 4. Memperbaiki *prototype*

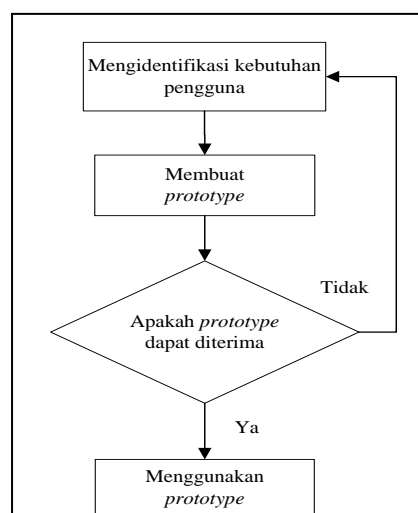
Pada tahap ini, pengembang akan melakukan modifikasi sesuai dengan masukan pemakai. Setelah modifikasi selesai dilakukan, maka akan dilanjutkan ke tahap menguji *prototype*. Tahap ini akan terus berulang sampai *prototype* yang dibuat sesuai dengan kebutuhan pemakai.

#### 5. Mengembangkan versi produksi

Tahap ini merupakan tahap terakhir dalam pembuatan *prototype* dimana pengembang merampungkan sistem sesuai dengan masukan terakhir dari pemakai.

### 2.4.2 *Evolutionary Prototype*

Menurut Mc Leod dan Schell (2008), *Evolutionary prototype* adalah *prototype* yang secara terus menerus dikembangkan hingga *prototype* tersebut memenuhi fungsi dan prosedur yang dibutuhkan oleh sistem. Pada pendekatan evolusioner, suatu *prototype* dibangun berdasarkan pada kebutuhan dan pemahaman secara umum. *Prototype* kemudian diubah dan dievolusikan daripada dibuang. *Prototype* yang dibuang biasanya digunakan dengan aspek sistem yang dimengerti secara luas dan dibangun atas kekuatan tahapan *evolutionary prototype*. Langkah-langkah dari *evolutionary prototype* dapat dilihat pada Gambar II.3.



Gambar II.3 *Evolutionary Prototype Model*  
(Sumber: McLeod dan Schell, 2008)

Keterangan:

1. Mengidentifikasi kebutuhan pengguna  
Pengembang mewawancarai pengguna untuk mendapatkan ide mengenai apa yang diminta dari sistem.
2. Membuat *prototype*  
Pengembang membuat *prototype* dari sistem yang telah dijelaskan oleh pengguna atau pemilik sistem.
3. Menentukan apakah *prototype* dapat diterima  
Pengembang mendemonstrasikan *prototype* kepada pengguna atau pemilik sistem tentang *prototype* yang sudah dibuat untuk mengetahui apakah sesuai atau tidak dengan kebutuhan sistem.
4. Menggunakan *prototype*  
Sistem mulai dikembangkan dengan *prototype* yang sudah dibuat.

## 2.5 Pengertian Informasi

Menurut Jogiyanto (1995) informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Menurut Bodnar dan Hopwood (2000) informasi adalah data yang berguna dan telah diolah sehingga dapat dijadikan dasar untuk mengambil keputusan yang tepat. Menurut Wilkinson (1993) informasi adalah data yang telah diproses sehingga berubah bentuknya dan mempunyai nilai semakin tinggi. Jadi informasi adalah data yang telah diolah, sehingga lebih berguna, berarti, dan dapat dijadikan dasar untuk mengambil keputusan yang tepat oleh orang yang menerimanya.

### 2.5.1 Siklus Informasi

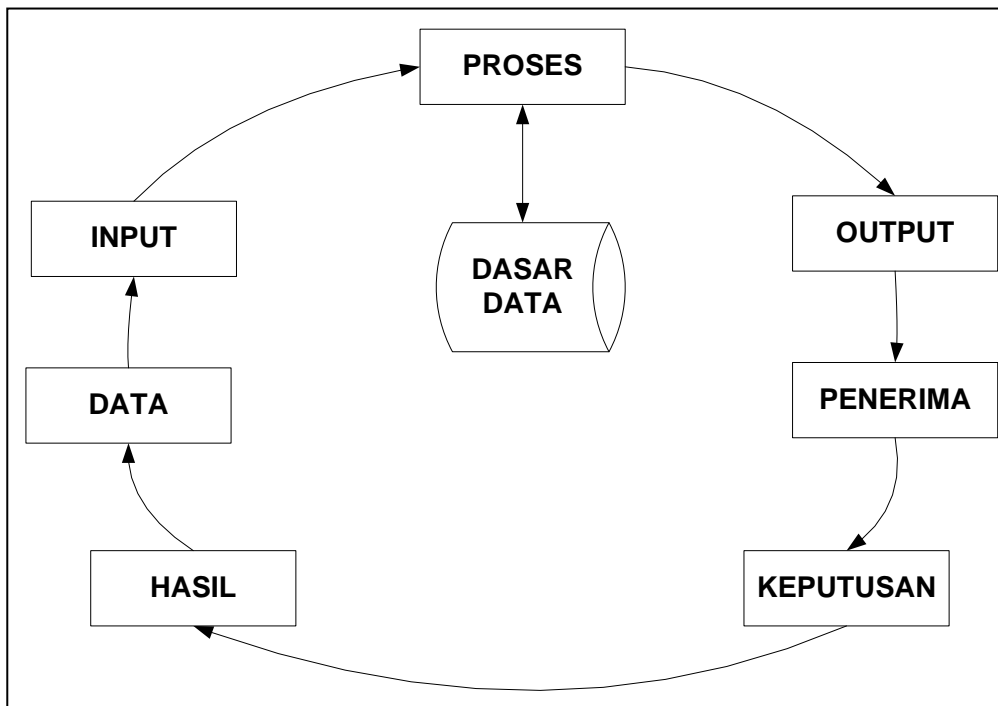
Menurut Sutabri (2004) Secara umum informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerima yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan.

Menurut Sutabri (2004) Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata.

Kejadian-kejadian adalah sesuatu yang terjadi pada saat tertentu. Di dalam dunia bisnis, kejadian-kejadian yang sering terjadi adalah transaksi perubahan dari suatu nilai yang disebut transaksi. Kesatuan nyata adalah berupa suatu objek nyata seperti tempat, benda dan orang yang betul-betul ada dan terjadi.

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, sinyal, gambar, dsb.

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. Siklus informasi ini dapat digambarkan sebagai berikut :



Gambar II.4 Siklus Informasi  
Sumber: Jogiyanto (2005)

## 2.6 Pengertian Sistem Informasi

Menurut Sutabri (2004) sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.

Menurut Davis (1991) sistem informasi adalah adanya informasi didapat dari suatu sistem dimana sistem informasi adalah suatu sistem dalam suatu sistem organisasi yang mempertemukan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategis dari satu organisasi dan menyediakan pihak luar tertentu dengan laporan yang diperlukan.

Menurut Kadir (2003) sistem informasi adalah kombinasi antara prosedur kerja, informasi, orang dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.

Menurut Leitch dan Davis (1983) sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

Oleh karena itu sistem informasi dapat didefinisikan sebagai suatu sistem didalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, media, prosedur-prosedur dan pengendalian yang ditunjukkan untuk memperoleh data agar menghasilkan suatu desain informasi untuk menunjang pengambilan keputusan oleh manajemen atau yang lainnya.

### 2.6.1 Komponen Sistem Informasi

Sistem informasi terdiri dari beberapa komponen antara lain (Sutabri, 2004):

#### 1. Blok Masukan (*Input Block*)

Input mewakili data yang masuk ke dalam sistem informasi. Yang dimaksud dengan input disini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

## 2. Blok Model (*Model Block*)

Blok ini terdiri dari kombinasi prosedur, logika dan model matematika yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara tertentu untuk menghasilkan keluaran yang diinginkan.

## 3. Blok Keluaran (*Output Block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

## 4. Blok Kendali (*Control Block*)

Banyak hal yang dapat merusak sistem informasi, seperti bencana alam, api, temperatur, air, debu, kecurangan-kecurangan, kegagalan pada sistem itu sendiri, ketidak-efisienan dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk menyakinkan bahwa hal-hal yang dapat merusak sistem dicegah dan bila terlanjur terjadi maka kesalahan-kesalahan dapat dengan cepat diatasi.

## 2.7 Material

Material merupakan salah satu unsur yang paling aktif didalam perusahaan yang secara terus-menerus diperoleh, diubah yang kemudian dijual kembali. Sebagian besar dari sumber-sumber perusahaan-perusahaan juga sering dikaitkan dalam persediaan material yang akan digunakan dalam operasi perusahaan pabrik.

Material adalah suatu benda yang diolah menjadi produk bahan jadi dan pemakaian dapat diidentifikasi secara langsung atau diikuti jejaknya atau merupakan integral dari produk tertentu. Kelompok-kelompok Persediaan:

### 1. Material

menurut Smith (1992). Merupakan barang-barang yang diperoleh untuk digunakan dalam proses produksi. Beberapa material diperoleh secara langsung dari sumber-sumber alam. Namun demikian, lebih sering lagi bahwa material diperoleh dari perusahaan lain dan ini merupakan produksi akhir dari para pemasok. Sebagai contoh, kertas cetak

merupakan produk akhir dari pabrik kertas, akan tetapi merupakan material bagi perusahaan percetakan

## 2. Barang-barang dalam Proses

Barang-barang dalam proses (*Good In Process*), dapat juga disebut pekerjaan dalam proses (*Work in process*), terdiri dari barang-barang baru sebagian diproses dan perlu dikerjakan lebih lanjut sebelum dijual.

Persediaan ini meliputi tiga unsur biaya yaitu :

- a. Biaya langsung
- b. Upah langsung
- c. Biaya tak langsung (*Factory overhead*) atau biaya tak langsung / biaya *overhead* produksi (*Manufacturing overhead*).

## 3. Barang-barang Jadi

Barang-barang Selesai (*Finished goods*) merupakan produk yang telah diproduksi dan menunggu dijual. Pada saat produk ini diselesaikan, biaya yang diakumulasikan dalam proses produksi ditransfer dari barang dalam proses ke perkiraan persediaan selesai.

## 2.8 Penerimaan Material

Pengertian Penerimaan material adalah menerima fisik material dari pabrik, *supplier*, distributor yang disesuaikan dengan dokumen pemesanan dan pengiriman dan dalam kondisi yang sesuai dengan persyaratan penanganan barangnya. Didalam aktifitas penerimaan material ini terdapat tiga point penting yang tidak dapat dipisahkan satu dengan lainnya. Manurung (2012):

### 1. Fisik Material Yang Diterima

Adalah bentuk fisik barang yang harus dapat diraba, diraba atau dilihat langsung. Penerimaan yang bukan berupa fisik barang dapat menyebabkan perbedaan proses dan hasil yang akan dicapai. Pada umumnya hasilnya adalah negatif. Jika ada penerimaan tanpa harus menangani fisik barangnya, maka perlu dilakukan proses tambahan untuk memastikan keabsahan proses tsb.

- a. Prinsip penerimaan material adalah menerima fisik material secara langsung. Bukan hanya dokumennya saja.
- b. Secara fisik, barang dapat dilihat, diraba atau dirasa dan dapat dibandingkan dengan dokumen pengantaran.
- c. Pengecekan acak atau keseluruhan kondisi isi kemasan.
- d. Tanggal Kadaluarsa barang.
- e. Kuantitas barang VS dokumen.

## 2. Dokumen

Dokumen pemesanan barang diterima berdasarkan adanya dokumen yang mendasari berupa barang yang harus diterima, jenis barangnya apa dan untuk memastikan bahwa barang yang diterima adalah sama dengan barang yang dikirimkan.

- a. Dokumen adalah pendamping barang yang secara fisik dapat dibaca dan dicocokkan dengan barang yang dikirimkan.
- b. Dokumen yang diperlukan minimal dokumen pengiriman (DN (*Delivery Note*), DO (*Delivery Order*), *Packing List* atau Surat Jalan).
- c. Akan lebih baik jika dokumen Pemesanan (*PO-Purchase Order*) dilampirkan juga.

## 3. Cara Penanganan Barang

Persyaratan penanganan kondisi khusus yang harus disiapkan pada saat barang tsb diterima. Apakah perlu ditangani pada suhu/temperatur khusus atau perlu dilakukan penanganan khusus dikarenakan faktor beratnya, tingkat kesulitannya atau masalah lainnya.

- a. Tangani material sesuai dengan siklus hidupnya
  1. Suhu
  2. Kadaluarsa
  3. Maksimal tumpukan

- b. Gunakan peralatan yang sesuai
  - 1. *Pallet*
  - 2. *Drum*
  - 3. *Forklift*
- c. Pahami aturan keselamatannya
  - 1. Kimia
  - 2. Racun
  - 3. Meledak

Secara umum dapat dinyatakan bahwa penerimaan material merupakan aktifitas operasional gudang yang sangat penting karena merupakan awal dari penanganan barang. Logika umum mengatakan bahwa penerimaan material yang baik saja masih memungkinkan terjadinya kerusakan/kesalahan material didalam gudang, terlebih jika pada saat penerimaan material ditangani dengan cara yang tidak benar, dijamin kerusakan/kesalahan tersebut pasti terjadi.

Proses Penerimaan barang di *warehouse* meliputi beberapa kegiatan yaitu:

- 1. Pembongkaran.
- 2. Pembukaan, pengepakan.
- 3. Pemeriksaan isi.
- 4. Mengecek kualitas dan kuantitas.
- 5. Keputusan penerimaan.
- 6. Penyimpanan ke gudang

## **2.9 Pengeluaran Material**

Pengeluaran material merupakan pemakaian material oleh bagian produksi dalam rangka proses produksi. Atau Pengeluaran dapat didefinisikan sebagai berikut : Pengeluaran adalah barang yang dikeluarkan oleh bagian gudang berdasarkan permintaan atau kebutuhan. Bagian yang terkait dalam transaksi pengeluaran barang dilaksanakan oleh bagian-bagian yang terpisah, yaitu: bagian gudang dan bagian produksi. Secara umum pelaksanaan transaksi pengeluaran barang dengan masing-masing bagian melaksanakan pekerjaan sesuai dengan fungsinya.

Siklus Pengeluaran merupakan serangkaian aktivitas bisnis dan kegiatan pengolahan data atas barang dan jasa yang dibeli. Dalam siklus ini, pihak eksternal yang terlibat adalah supplier, sedangkan pihak internal yang terkait adalah siklus produksi, siklus pendapatan, dan siklus buku besar dan pelaporan. Bentuk interaksi antara siklus pengeluaran dan siklus lainnya adalah siklus pengeluaran menerima pemberitahuan dari siklus pendapatan dan sistem produksi tentang kebutuhan material dan juga memberitahu kapan material tersebut harus diterima. Siklus pengeluaran dapat diselenggarakan secara manual dan berbasis komputer. Perbedaan pokok kedua cara tersebut adalah pada mekanisme pengolahan datanya, sedangkan input dan output yang dihasilkan relatif sama. Tujuan utama dari siklus pengeluaran ini adalah untuk mempermudah pengolahan data dengan para supplier untuk material yang dibutuhkan dimana tujuan khusus yang terkandung didalamnya meliputi :

- a. Memastikan bahwa seluruh material yang dipesan sesuai keperluan.
- b. Menerima seluruh barang yang dipesan dan menverifikasi bahwa barang tersebut adalah valid dan benar.
- c. Menjaga barang tersebut sampai dibutuhkan.
- d. Memastikan bahwa faktur yang berhubungan dengan material adalah valid dan benar.
- e. Mencatat dan mengklasifikasi pengeluaran secara cepat dan tepat.
- f. Menyiapkan seluruh dokumen dan laporan manajerial yang diperlukan yang berhubungan dengan material yang diperoleh.

Berikut ini adalah prosedur pengeluaran material.

1. Bagian produksi akan mengisi formulir permintaan material, yang selanjutnya diberikan ke bagian gudang. Formulir permintaan material berisi antara lain :
  - a. Nomor pesanan dan bagian yang meminta material (bagian pemakai).
  - b. Nama material yang diminta.
  - c. Jumlah dan keterangan bahan baku yang diperlukan.
  - d. Kolom paraf .

2. Bagian gudang akan memberikan material sesuai dengan informasi yang tercantum dalam formulir permintaan material tersebut.
3. Bagian gudang akan mencatat pengeluaran material ini .

## **2.10 Gudang (*Warehouse*)**

Pergudangan memegang peran yang sangat penting dalam kehidupan sebuah perusahaan. Peningkatan produktivitas dan pelayanan pergudangan akan sangat berpengaruh pada performansi perusahaan secara keseluruhan. Gudang merupakan sarana pendukung kegiatan produksi industri manufaktur yang berfungsi untuk menyimpan material, bahan penolong dan bahan habis pakai yang belum didistribusikan. Selain untuk penyimpanan, gudang juga berfungsi untuk melindungi material, bahan penolong dan bahan habis pakai. Fungsi gudang lainnya agar terhindar dari pengaruh luar, binatang pengerat dan serangga serta melindungi material dari kerusakan. Agar dapat menjalankan fungsi tersebut, maka harus dilakukan pengelolaan pergudangan secara benar.

Pergudangan adalah segala upaya pengelolaan gudang yang meliputi penerimaan, penyimpanan, pemeliharaan, pendistribusian, pengendalian dan pemusnahan agar kualitas dan kuantitas tetap terjaga.

### **2.10.1 Tipe dan Jenis Gudang**

Gudang adalah jenis yang paling umum dari tempat penyimpanan meskipun memang ada bentuk-bentuk lain (misalnya, tangki penyimpanan).

Beberapa gudang memiliki bangunan besar dan luas sehingga memungkinkan kegiatan pembongkaran material dari truk *supplier* dan kegiatan memuat material ke pelanggan secara bersamaan. Dibawah ini adalah tipe-tipe gudang menurut Warman (2004) yaitu:

#### **1. Gudang Pribadi/Swasta**

Jenis gudang ini dimiliki dan dioperasikan oleh *supplier* dan *reseller* untuk digunakan dalam kegiatan distribusi mereka sendiri. Sebagai contoh, jaringan ritel besar menyediakan gudang untuk toko mereka atau grosir

mengoperasikan sebuah gudang di mana ia menerima dan mendistribusikan produk.

## 2. Gudang Publik/Umum

Gudang umum pada dasarnya adalah ruang yang dapat disewakan untuk mengatasi kebutuhan distribusi dalam jangka pendek. Pengecer yang memiliki gudang sendiri terkadang mereka mencari ruang penyimpanan tambahan jika kapasitas gudang mereka tidak mencukupi atau jika mereka melakukan pembelian produk dalam jumlah besar dengan alasan tertentu. Sebagai contoh, pengecer bisa memesan tambahan material untuk memaksimalkan penjualan di toko atau ketika ada harga promosi dari pemasok jika membeli dalam jumlah besar.

### 2.10.2 Klasifikasi Persediaan dalam Gudang

Persediaan atau *inventory* menurut Herjanto (2007), merupakan bahan atau barang yang disimpan dan akan digunakan untuk memenuhi tujuan tertentu, misalnya untuk proses produksi atau perakitan untuk dijual kembali dan untuk suku cadang dari suatu peralatan. Persediaan dapat berupa bahan mentah, bahan pembantu, barang dalam proses, barang jadi, ataupun suku cadang.

Gudang seperti kegunaannya secara umum merupakan suatu tempat untuk menyimpan benda, benda yang disimpan di dalam gudang ini dapat pula disebut sebagai persediaan atau *inventory*.

Menurut Warman (2004), secara umum persediaan dapat diklasifikasikan berdasarkan dua hal yaitu, klasifikasi persediaan berdasarkan fungsi dari barang dalam gudang dan klasifikasi persediaan berdasarkan kecepatan arus aliran barang.

#### 1. Klasifikasi Persediaan berdasarkan Fungsi Barang

Dalam industri, persediaan yang disimpan dalam gudang dapat bermacam-macam fungsinya. Dalam klasifikasi ini gudang akan dibagi-bagi sesuai dengan barang apa saja yang disimpan dalam gudang tersebut.

Secara umum berdasarkan fungsi fisiknya, persediaan dapat dibagi menjadi 4 fungsi utama, keempat fungsi itu adalah:

a. Sebagai *Raw Material*

*Raw material* merupakan barang yang akan di proses dan diberi nilai tambah untuk kemudian dapat di jual dengan nilai yang lebih tinggi. *Raw material* dapat berbeda-beda untuk setiap perusahaan, tergantung jenis usaha dan tujuan usahanya. Barang yang menjadi *raw material* di suatu perusahaan belum tentu menjadi *raw material* pula di perusahaan lain. Bisa saja *raw material* di suatu perusahaan menjadi *finished good* di perusahaan lain, misalnya, dalam perusahaan roti, barang yang menjadi *raw material* di perusahaan itu adalah tepung, akan tetapi bagi sebuah pabrik tepung, tepung adalah *finished good* yang dihasilkan dari proses-proses rumit yang mengubah biji gandum menjadi tepung.

b. Sebagai *Work In Process (WIP)*

Barang *work in process* dalam bahasa sehari-hari dikenal dengan nama barang setengah jadi. Barang *work in process* ini adalah *raw material* yang dikenai proses untuk menjadi suatu produk, hanya saja belum selesai, atau dapat dikatakan masih setengah jalan.

c. Sebagai *Finished Good*

*Finished goods* merupakan barang yang siap untuk di sajikan atau siap untuk dipasarkan kepada konsumen. *Finished good* ini merupakan barang yang diperoleh dari bahan dasar berupa *raw material* yang telah diproses dan diberi nilai tambah.

d. Sebagai *Sparepart*

Peralatan atau *sparepart* adalah barang yang tidak memberikan nilai tambah kepada suatu *raw material* untuk menjadi *finished good*, akan tetapi *sparepart* akan sangat berguna sekali untuk mendukung kelancaran proses pemberian nilai tambah kepada *raw material* untuk menghasilkan *finished good*. Contohnya pada perusahaan kertas, peralatan yang digunakan untuk membentuk kertas A4 adalah sebuah pisau potong, dalam hal ini pisau potong merupakan peralatan yang digunakan untuk memberikan nilai tambah dari kertas yang berupa glondongan untuk kemudian di proses menjadi berukuran A4. Dalam hal ini peralatan dapat pula seluruh barang

yang digunakan untuk keperluan *maintenance* mesin atau peralatan untuk proses *finished good*.

## 2. Klasifikasi Persediaan jika dipandang dari Aliran Arus Barang

Dalam gudang, baik gudang yang merupakan gudang *raw material*, gudang *WIP*, gudang *finished good* ataupun gudang *sparepart* pasti akan terdapat perbedaan arus aliran barang-barang yang ada di dalamnya.

Dalam suatu gudang misalnya gudang *finished good* ada terdapat bermacam-macam *finished good* yang disimpan dalam gudang tersebut yang berbeda jenisnya, dengan adanya perbedaan jenis tersebut maka aliran setiap barang tidak akan sama.

Dalam klasifikasi ini persediaan akan dipandang berdasarkan aliran barang tersebut, apakah barang tersebut merupakan golongan *fast moving*, *medium moving* dan *slow moving*.

### a. Barang *Fast Moving*

Barang-barang yang disebut sebagai *fast moving* adalah barang dengan aliran yang sangat cepat, atau dengan kata lain barang *fast moving* ini akan berada di gudang dalam waktu yang sangat singkat.

### b. Barang *Medium Moving*

Barang *medium moving* adalah barang-barang yang aliran barangnya sedang-sedang saja, tidak terlalu cepat atau terlalu lambat. Biasanya barang ini akan berada di gudang dalam waktu yang relatif lebih lama jika dibanding dengan barang-barang *fast moving*.

### c. Barang *Slow Moving*

Barang-barang *slow moving* merupakan barang dengan arus aliran barang yang sangat lambat, sehingga biasanya barang-barang yang *slow moving* ini akan tersedia di gudang dalam jangka waktu yang cukup lama. Aliran barang ini harus sangat diperhatikan dalam menjalankan manajemen pergudangan secara efektif atau belum. Dengan memperhatikan kecepatan aliran barang tersebut diharapkan aliran barang yang ada di gudang menjadi lancar.

Untuk barang *fast moving* dijaga agar stok di gudang tidak kehabisan sehingga tidak mengecewakan konsumen, sedangkan untuk barang yang *slow moving* dijaga supaya tidak terjadi penumpukan barang yang tidak perlu digunakan sehingga kapasitas gudang dapat digunakan sebaik dan seefektif mungkin.

### 3. Penyimpanan Barang

Dalam penyimpanan barang di gudang terdapat beberapa teknik yang terdiri dari tata letak barang dan *racking system*. Tata letak barang dalam gudang biasanya disebut dengan *layout* barang. *Layout* barang merupakan suatu metode peletakan barang dalam gudang untuk mempermudah, mempercepat dan meningkatkan efisiensi dari gudang tersebut dalam menampung barang maupun mengalirkan permintaan barang kepada pihak yang melakukan permintaan.

Pihak yang melakukan permintaan ini dapat dibagi menjadi *internal customer* atau *external customer*. *Internal customer* adalah pelaku *demand* yang berada dalam suatu perusahaan yaitu departemen lain dalam perusahaan. *External customer* adalah konsumen dalam pengertian secara umum yaitu pihak pelaku *demand* yang berasal dari luar perusahaan.

### 4. Tata Letak Barang

Dalam melakukan pengaturan tata letak barang di gudang terdapat beberapa hal yang harus diperhatikan. Menurut Warman (2004), hal yang harus diperhatikan dalam melakukan pengaturan tata letak gudang adalah sistem pengukuran kecepatan yang baik dan sistem pengendalian yang baik.

Sistem pengukuran kecepatan akan melihat barang berdasarkan klasifikasi kecepatan arus aliran barang dimana barang akan dibagi menjadi tiga macam yaitu *slow moving*, *medium moving*, dan *fast moving*. Dengan melihat ketiga macam barang tersebut maka akan dapat dilakukan pengendalian barang dengan baik.

Untuk barang-barang *slow moving* hendaknya diletakkan di bagian gudang yang paling sulit untuk dijangkau, dengan alasan karena barang ini sangat jarang mengalami perpindahan barang.

Untuk barang-barang *fast moving* biasanya diletakkan di bagian yang cukup terbuka sehingga dapat memudahkan dalam melakukan pengambilan barang. Dengan melakukan peletakan barang seperti di atas maka pengendalian dalam melakukan pengambilan barang akan lebih mudah, sehingga efisiensi gudang akan menjadi tinggi. Dalam kenyataannya, tidak akan ada gudang yang sama persis untuk satu perusahaan dengan perusahaan lain, ini dipengaruhi oleh jumlah *item* yang diproduksi oleh perusahaan, *item* yang menjadi *slow moving* atau *fast moving* dan kapasitas gudang dari masing-masing perusahaan

#### 5. *Racking System*

*Racking System* adalah suatu cara untuk meningkatkan kapasitas tanpa melakukan pelebaran gudang. Selain itu juga dapat digunakan untuk melakukan pengelompokan barang sehingga gudang terlihat lebih teratur tanpa membutuhkan tempat yang terlalu luas.

Tujuan utama adalah untuk meningkatkan kapasitas gudang tanpa harus melakukan pelebaran gudang. Hal ini disebabkan karena dengan menggunakan sistem rak, kita akan melakukan penyusunan barang dengan konsep bertingkat, dengan kata lain kita melakukan pemanfaatan ketinggian untuk memperbanyak kapasitas dari gudang. Proses Penerimaan bahan baku di warehouse meliputi beberapa kegiatan yaitu:

1. Pembongkaran.
2. Pembukaan, pengepakan.
3. Pemeriksaan isi vs *packing listing*.
4. Inspeksi kualitas.
5. Laporan penerimaan dan inspeksi.
6. Keputusan penerimaan.
7. Penyerahan ke gudang.
8. Prosedur Penerimaan barang.

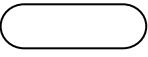
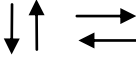


## 2.11 Diagram Alir

Bagan alir (*flowchart*) 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). Berikut ini pedoman-pedoman dalam pembuatan bagan alir (Jogiyanto, 2005):

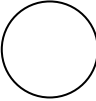
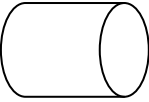
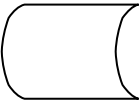


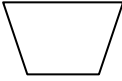


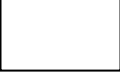

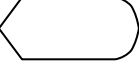

1. Bagan alir sebaiknya digambar dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
2. Kegiatan di dalam bagan alir harus ditunjukkan dengan jelas.
3. Harus ditunjukkan dari mana kegiatan akan dimulai dan dimana akan berakhirnya.
4. Masing-masing kegiatan di dalam bagan alir sebaiknya menggunakan suatu kata yang mewakili suatu pekerjaan. Misalnya, Persiapkan dokumen, dan Hitung gaji.
5. Masing-masing kegiatan di dalam bagan alir harus di dalam urutan yang semestinya.
6. Kegiatan yang terpotong dan akan disambung di tempat lain harus ditunjukkan dengan jelas menggunakan simbol penghubung.
7. Gunakanlah simbol-simbol bagan alir yang standar.

Simbol-simbol diagram alir dapat dilihat pada tabel di bawah ini.

Tabel II.1 Simbol-simbol *Flowchart*

Simbol	Nama	Keterangan
	Titik terminal	Menunjukkan awal dan akhir dari suatu proses.
	Arus/ <i>Flow</i>	Menunjukkan jalannya suatu arus proses.
	Dokumen	Menunjukkan dokumen <i>input</i> dan <i>output</i> baik untuk proses manual, mekanik, atau komputer.
	Dokumen rangkap	Menggambarkan dokumen/formulir asli dan tembusannya.

Tabel II.1 Simbol-simbol *Flowchart* (lanjutan)

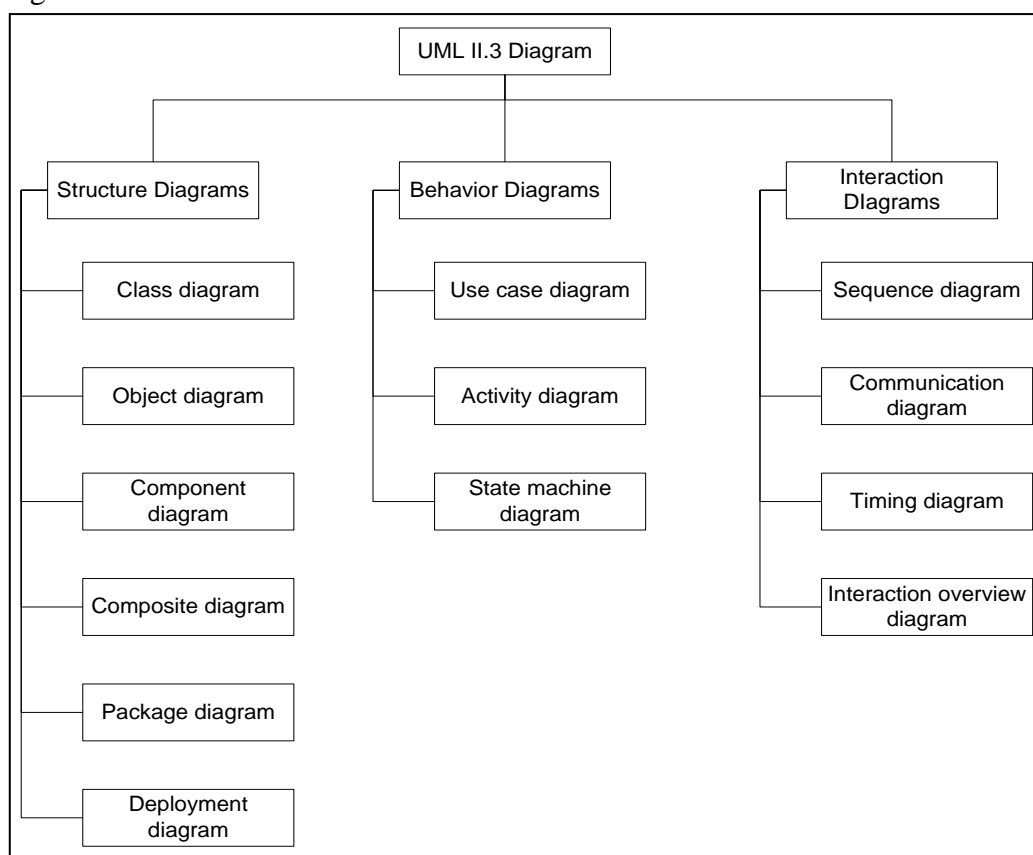
Simbol	Nama	Keterangan
	<i>Online connector</i>	Simbol penghubung yang memungkinkan aliran dokumen berhenti di suatu lokasi pada halaman tertentu dan kembali berjalan pada halaman yang sama.
	Simbol Data Base	Simbol database yang system nya sudah menggunakan DBMS dan dapat di akses langsung
	Simbol Penyimpanan Secara Manual	Menunjukkan media penyimpanan data / informasi secara manual
	<i>Offline connector</i>	Untuk menggambarkan bagan alir dokumen suatu sistem diperlukan lebih dari satu halaman.
	Proses terdefinisi	Menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain.
	Operasi/ kegiatan manual	Menggambarkan kegiatan manual seperti menerima, mengisi formulir, membandingkan, dan lain-lain.
	Arsip sementara	Menunjukkan tempat penyimpanan dokumen.
	Arsip permanen	Menunjukkan tempat penyimpanan dokumen secara permanen yang tidak akan diproses lagi.
	Proses	Simbol proses yang digunakan untuk mewakili suatu proses dengan komputer.
	Keputusan	Menggambarkan keputusan yang harus dibuat dalam proses pengolahan data.
	Tampilan <i>Output</i>	Menunjukkan <i>output</i> yang ditampilkan pada <i>monitor</i> .
	<i>Input/Output</i>	Menggambarkan data <i>input</i> atau <i>output</i> dari proses.

(Sumber: Jogiyanto, 2005)

## 2.12 *Unified Modeling Language (UML)*

*Unified modeling language (UML)* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek (Rosa dan Shalahuddin, 2011)

Pada UML II.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar di bawah ini.



Gambar II.5 Diagram UML  
(Sumber: Rosa dan Shalahuddin, 2011)

Berikut ini penjelasan singkat dari pembagian kategori diagram UML pada Gambar II.3 (Rosa dan Shalahuddin, 2011):



1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

2. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu system.



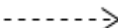
### 2.12.1 Use case Diagram

Menurut Rosa dan Shalahuddin (2011), *use case diagram* merupakan pemodelan untuk perilaku (*behavior*) sistem informasi yang akan dibuat. *Use case* menggambarkan interaksi antara sistem dengan eksternal sistem dan pengguna. Dengan kata lain, secara grafis menggambarkan siapa yang akan menggunakan sistem dan dengan cara apa pengguna mengharapkan untuk berinteraksi dengan sistem. Simbol-simbol pada *use case diagram* dapat dilihat pada Tabel II.2.

Tabel II.2 Simbol-simbol *Use Case Diagram*

Simbol	Nama	Keterangan
	Aktor/ <i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambaran orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
	<i>Use Case</i>	Fungsionalitas yang disediakan oleh sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .

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







Simbol	Nama	Keterangan
	Asosiasi/ <i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
	<i>Include</i>	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.
	Generalisasi/ <i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).
	Ekstensi/ <i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek.

(Sumber: Rosa dan Shalahudin, 2011)

### 2.12.2 *Activity Diagram*

Menurut Rosa dan Shalahuddin (2011), *activity diagram* menggambarkan aliran kerja atau aktivitas dari sebuah sistem yang ada pada perangkat lunak. Adapun yang perlu diperhatikan disini adalah bahwa *activity diagram* menggambarkan aktivitas sistem bukan apa yang dilakukan aktor. Adapun simbol-simbol pada *activity diagram* dapat dilihat pada Tabel II.3.

Tabel II.3 Simbol-simbol *Activity Diagram*

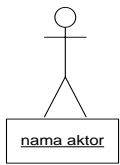

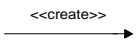
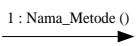
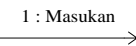
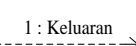
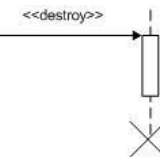
Simbol	Nama	Keterangan
	Status Awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	Percabangan/ <i>decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
	Penggabungan <i>/ join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas yang digabungkan menjadi satu.
	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggungjawab terhadap aktivitas yang terjadi.

(Sumber: Rosa dan Shalahudin, 2011)

### 2.12.3 *Sequence Diagram*

Menurut Rosa dan Shalahuddin (2011), *sequence diagram* yaitu diagram yang menggambarkan perilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek (*life line*) dan *message* yang dikirimkan dan diterima antar objek. Untuk menggambar *sequencediagram* terlebih dahulu harus mengetahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Simbol-simbol pada *sequencediagram* dapat dilihat pada Tabel II.4.

Tabel II.4 Simbol-simbol *Sequence Diagram*

Simbol	Nama	Keterangan
	Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi, jadi simbol dari aktor ini adalah gambaran orang.
	Garis Hidup ( <i>lifeline</i> )	Menyatakan kehidupan suatu objek.
	Waktu Aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
	Pesan tipe <i>Create</i>	Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat.
	Pesan tipe <i>Call</i>	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi/metode.
	Pesan tipe <i>Send</i>	Menyatakan bahwa suatu objek mengirimkan data/masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
	Pesan tipe <i>Return</i>	Menyatakan suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
	Pesan tipe <i>Destroy</i>	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

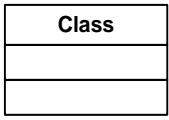


(Sumber: Rosa A. S. dan Shalahudin, 2011)

#### 2.12.4 Class Diagram

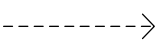

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas (Rosa dan Shalahuddin, 2011).

Semua benda dari suatu kelas yang identik dalam struktur dan perilaku, tetapi mengandung data yang berbeda dalam atributnya masing-masing. Pada umumnya kelas dibagi menjadi dua jenis kelas yang menarik selama tahap analisis, yaitu konkret dan abstrak. Kelas konkret adalah sesuatu yang nyata-dunia yang diwakilinya. Ada kelas aplikasi domain, kelas *user interface*, kelas struktur data, struktur *file* kelas, kelas lingkungan operasi, kelas dokumen, dan berbagai jenis kelas *multimedia*. Kelas abstrak hanya merupakan abstraksi yang bermanfaat. Misalnya, dari kelas karyawan dan kelas pelanggan dapat diidentifikasi generalisasi dari dua kelas dan nama kelas abstrak orang. Dennis dan Wixom (2005). Berikut contoh simbol *class diagram*.

Tabel II.5 Simbol-Simbol *Class Diagram*.

Simbol	Nama	Keterangan
	<i>Class</i>	Kelas adalah <i>template</i> umum yang digunakan untuk membuat kasus tertentu atau benda dalam domain aplikasi.
	<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).

Tabel II.5 Simbol-Simbol *Class Diagram*. (lanjutan)

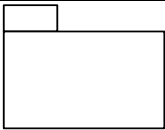
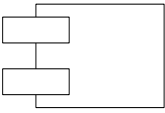
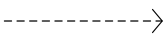
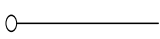
Simbol	Nama	Keterangan
	<i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
	<i>Aggregation</i>	Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> ).

Sumber: Rosa dan Shalahuddin (2011)

### 2.12.5 *Component Diagram*

*Component diagram* adalah diagram yang dibuat untuk menunjukkan organisasi dan ketergantungan di antara kumpulan komponen dalam sebuah sistem (Rosa dan Shalahuddin, 2011). Simbol *component diagram* dapat dilihat pada tabel II.6.

Tabel II.6 Simbol-Simbol *Component Diagram*

Simbol	Nama	Keterangan
	<i>Package</i>	<i>Package</i> merupakan sebuah bungkusan dari satu atau lebih komponen.
	<i>Component</i>	Komponen sistem.
	Kebergantungan/ <i>dependency</i>	Kebergantungan antar komponen arah panah mengarah pada komponen yang dipakai.
	Kebergantungan/ <i>dependency</i>	Sebagai antarmuka komponen agar tidak mengakses langsung komponen.

Sumber: Rosa dan Shalahuddin (2011)




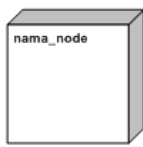
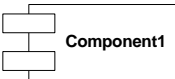
### 2.12.6 *Deployment Diagram*

Diagram *deployment* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi (Rosa dan Shalahuddin, 2011). Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal berikut:

1. Sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node*, dan *hardware*.
2. Sistem *client/server*.

Simbol *deployment diagram* dapat dilihat pada tabel II.7.

Tabel II.7 *Deployment Diagram*

Simbol	Nama	Keterangan
	<i>Link</i>	Relasi antar komponen.
	<i>Package</i>	Merupakan sebuah bungkus dari satu atau lebih <i>node</i>
	Kebergantungan/ <i>dependency</i>	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai.
	<i>Node</i>	Biasanya mengacu pada perangkat keras dan perangkat lunak yang tidak dibuat sendiri, jika di dalam <i>node</i> disertakan komponen untuk konsistensi rancangan maka komponen yang diikutsertakan sesuai dengan komponen yang telah didefinisikan pada diagram komponen.
	Komponen	Komponen sistem

Sumber: Rosa dan Shalahuddin (2011)

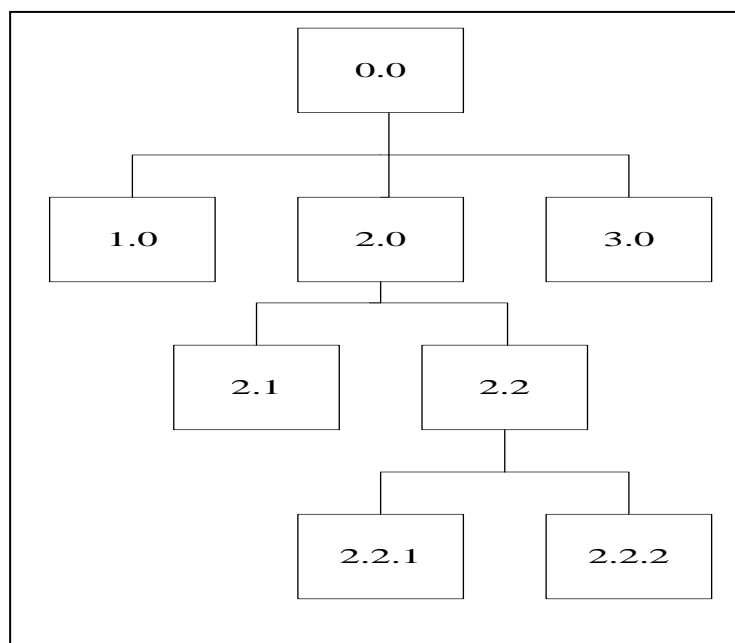
### 2.13 *Hierarchy plus Input-Process-Output (HIPO)*

*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 di dalam sistem digambarkan oleh fungsi utamanya (Jogiyanto, 2005). HIPO (*Hierarchy plus Input-Process-Output*) mempunyai sasaran utama sebagai berikut (Jogiyanto, 2005):

1. Untuk menyediakan suatu struktur guna memahami fungsi-fungsi dari sistem.
2. Untuk lebih menekankan fungsi-fungsi yang harus diselesaikan oleh program, bukannya menunjukkan *statement* 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.

Salah satu jenis dari HIPO, yaitu *visual table of contents* (VTOC). VTOC 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. Keterangan masing-masing fungsi diberikan pada bagian penjelasan yang diikutsertakan dalam diagram ini. *Visual table of contents* ini dapat digambarkan sebagai berikut:



Gambar II.6 *Visual Table Of Contents*  
 Sumber: Jogyanto, (2005)

#### 2.14 *Hypertext Preprocessor (PHP)*

*Hypertext Preprocessor* (PHP) adalah bahasa *scripting* yang menyatu dengan HTML dan dijalankan dan di eksekusi pada *server web*. Dokumen HTML yang dihasilkan dari suatu aplikasi yang bukan dokumen HTML yang dibuat

dengan menggunakan *editor teks* atau *editor HTML* dikenal sebagai bahasa pemrograman *server side*.

PHP pertama kali dibuat pada musim gugur tahun 1994 oleh Rasmus Lerdoff ([rasmus@php.net](mailto:rasmus@php.net)) awalnya digunakan pada *web* sitenya biodatanya. Versi pertama dirilis tersediapada awal tahun 1995, dikenal sebagai *Tool Personal Home Page*, yang terdiri atas makro khusus dan sejumlah utilitas yang sering digunakan pada halaman – halaman web, seperti buku tamu , *counter* pengunjung dan lainnya. Parser diprogram ulang pada pertengahan 1995 dan diberi nama PHP/FI versi 2.0. FI berasal dari paket Rasmus lainnya yang dikombinasikan dengan *Tool Personal Home Page* dan ditambahkan dukungan untuk *database mSQL* (mini SQL).

Tahun 1995 ini dianggap sebagai tahun kelahiran dari PHP/FI yang kemudian membuat pertumbuhan aplikasi *web* yang pesat, dan banyak orang kemudian berkontribusi mengembangkan PHP/FI. Sulit untuk mendapatkan statistik yang tepat untuk memperkirakan penggunaan PHP/FI, tetapi diperkirakan pada akhir 1996 telah digunakan oleh sedikitnya 15000 *website* di seluruh dunia. Dan pertengahan 1997 mencapai 50000 situs.

Pada Pertengahan tahun 1997 terjadi perubahan pengembangan PHP. Pengembangan pengembangan dilakukan oleh tim yang terorganisasi bukan oleh Rasmus Sendiri lagi. Parser dikembangkan oleh Zeev Suraski dan Andi Gutschans yang kemudian menjadi dasar untuk versi 3, dan banyak utilitas tambahan yang diprogram untuk menambah kemampuan dari versi 2. PHP versi 4 menggunakan *Engine Script Zend* untuk lebih meningkatkan kinerja dan mempunyai dukungan yang banyak berupa ekstensi dan fungsi dari berbagai *library* pihak ketiga (*third party*), dan berjalan seolah modul asli dari berbagai *server web* populer.

Sejak Januari 2001 PHP3 dan PHP4 disertakan pada sejumlah produk *server web* komersial seperti *server web StrongHold RedHat*. Perkiraan konservatif yang didapat dari angka yang diberikan Netcraft (<http://netcraft.com>) yang diekstrapolasi, pengguna PHP sekitar 5.100.000 sedikit lebih banyak dari *server web* yang menggunakan *Microsoft IIS* (5.03juta) di Internet.

PHP versi 4.0 dirilis pada tanggal 22 April 2002. Perbaikan pada bug (kesalahan-kesalahan) terutama pada *upload file* melalui *browser* telah dibetulkan dan banyak penambahan fungsi yang lebih memudahkan lagi pengembang aplikasi untuk membuat program yang lebih baik. Jumlah fungsi yang ada sampai dengan versi ini mencapai ribuan fungsi dan dikelompokkan menjadi 109 fungsi, banyak bertambah dari versi sebelumnya. (Sidik, 2014)

Saat itu pengembangan PHP telah direncanakan untuk mulai memasuki versi 6 dan PHP 5 telah mencapai versi 5.3 yang stabil dan telah dikeluarkan juga versi 5.4 yang baru dilepaskan sebagai kandidat ke-4, PHP versi 5.4.0 RC4.

### 2.14.1 Kemampuan PHP

PHP secara mendasar dapat mengerjakan semua yang dapat dikerjakan oleh program, seperti mendapatkann data dari *form*, menghasilkan isi halaman web yang dinamik, dan menerima *cookies*. PHP telah dikembangkan juga menjadi bahasa pemrograman script yang dapat dijalankan di atas *flatfom* sistem operasi secara langsung dan dapat juga digunakan untuk program *desktop*. (Sidik, 2014)

### 2.15 MySQL

*MySQL* adalah sebuah perangkat lunak sistem manajemen basis data *SQL database management system* atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia (Solichin, 2010). *MySQL* memiliki cukup banyak tipe data untuk kolom (*field*) tabel. Tipe *field* ini menentukan besar kecilnya ukuran suatu tabel. Berikut ini tipe-tipe data *field* yang ada dalam *MySQL* (Solichin, 2010):

#### 1. Numerik

Tipe data numerik digunakan untuk menyimpan data *numeric* (angka). Ciri utama data *numeric* adalah suatu data yang memungkinkan untuk dikenai operasi aritmatika seperti penambahan, pengurangan, perkalian dan pembagian.

#### 2. *Date and Time*

Tipe data *date* dan *time* digunakan untuk menyimpan data tanggal dan waktu.

### 3. *String (Text)*

Tipe data *string* digunakan untuk menyimpan data *string (text)*. Ciri utama data *string* adalah suatu data yang memungkinkan untuk dikenai operasi aritmatika seperti penambahan, pengurangan, perkalian dan pembagian.

### 4. BLOB (*Binary*)

Tipe data BLOB digunakan untuk menyimpan data biner. Tipe ini biasanya digunakan untuk menyimpan kode-kode biner dari suatu *file* atau *object*. BLOB merupakan singkatan dari *Binary Large Object*.

### 5. *Enum*

Tipe data *enum* merupakan kumpulan (enumerasi) data yang memiliki ukuran data sampai dengan 65535 *string*.

### 6. *Set*

Tipe data *set* merupakan himpunan (*combination*) data yang memiliki ukuran data sampai dengan 255 *string* anggota.

## 2.16 XAMPP

*Web Server* merupakan sebuah perangkat lunak dalam *server* yang berfungsi menerima permintaan (*request*) berupa halaman *web* melalui HTTP atau HTTPS dari klien yang dikenal dengan *browser web* dan mengirimkan kembali (*response*) hasilnya dalam bentuk halaman-halaman *web* yang umumnya berbentuk dokumen HTML. XAMPP adalah sebuah *software web apache* yang di dalamnya sudah tersedia *database server mysql* dan *support php programming*. XAMPP merupakan *tool* yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dengan meng-*install* XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi *web server Apache PHP* dan *MySQL* secara manual. XAMPP akan menginstalasi dan konfigurasinya secara otomatis untuk *user* atau auto konfigurasi (Solichin, 2010).

## 2.17 Macromedia Dreamweaver

*Dreamweaver* merupakan produk *software* Adobe yang digunakan sebagai HTML *editor* profesional untuk mendesain *web* secara visual dan dapat juga

digunakan untuk mengelola situs atau halaman *web*. Selain itu, *dreamweaver* memberikan keleluasaan kepada pengguna untuk menggunakannya sebagai media penulisan bahasa pemrograman *web* (MADCOMS, 2009).

Ruang kerja (*workspace*) pada *dreamweaver* berisi *toolbar*, *property inspector*, dan *panel* yang dapat digunakan untuk mendesain halaman *web*. Berikut ini penjelasan *workspace* yang terdapat dalam *dreamweaver* (MADCOMS, 2009):

1. *Toolbar*

Baris yang berisi tombol-tombol yang digunakan untuk mengubah tampilan jendela dokumen, seperti tampilan desain atau tampilan kode. Selain itu, baris ini juga dapat digunakan untuk operasi-operasi umum. Misalnya untuk melihat hasil sementara halaman *web* pada jendela *browser*.

2. *Property Inspector*

Komponen ini digunakan untuk melihat dan mengubah berbagai properti objek atau teks pada jendela desain. Properti untuk satu objek dengan objek yang lainnya selalu berbeda-beda. Jendela ini tidak dapat diuraikan pada tampilan jendela kode.

3. *Panel*

*Panel* dibagi menjadi dua, yaitu:

- a. *Panel Insert*

*Panel* ini berisi tombol-tombol untuk menyisipkan berbagai jenis objek, seperti gambar, tabel atau objek media ke dalam jendela dokumen. Masing-masing objek adalah suatu potongan kode HTML, *user* dapat mengatur properti dari objek tersebut dengan bebas.

- b. *Panel Files*

*Panel* ini digunakan untuk mengatur semua *file* dan *folder* yang membentuk situs *web user*, seperti menyalin, memindahkan atau mengganti nama *file*. Selain itu, *panel* ini juga dapat digunakan untuk melakukan *uploading file* ke *server* baik *server intranet* maupun *internet*.

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Metodologi Penelitian**

Metodologi penelitian adalah tata cara bagaimana suatu penelitian dilaksanakan atau cara yang ditempuh sehubungan dengan penelitian yang dilakukan, yang memiliki langkah-langkah sistematis (Hasan, 2002).

Dengan adanya metodologi penelitian, kegiatan penelitian yang akan dikerjakan akan lebih terarah sesuai dengan prosedur yang telah ditetapkan namun tidak menutup kemungkinan akan terjadi variasi urutan penelitian yang tentunya disebabkan oleh permasalahan yang sedang diteliti.

##### **3.1.1 Jenis dan Sumber Data**

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

1. Data primer

Data yang diperoleh melalui pengamatan langsung pada PT Tetra Pak Stainless Equipment Jakarta di departemen pergudangan terhadap sistem yang sedang berjalan dan wawancara dengan pegawai sebagai sumber informasinya. Dalam penelitian ini data tersebut berupa data pengeluaran material.

2. Data Sekunder

Data yang diperoleh dari data yang tersedia dan telah terlebih dahulu dikumpulkan dan dilaporkan oleh orang lain, buku-buku dan kajian ilmiah dari berbagai sumber yang berkaitan dengan penelitian. Data tersebut adalah data umum perusahaan, profil perusahaan dan struktur organisasi perusahaan.

### 3.2 Metode Pengumpulan Data

Metode yang digunakan dalam pengumpulan data, diantaranya:

#### 1. Studi Lapangan

##### a. Metode Observasi

Metode yang dilakukan dengan cara mengamati proses-proses yang berlangsung selama produksi di Departemen Pergudangan PT Tetra Pak Stainless Equipment Jakarta dan kemudian mencatat informasi tentang objek yang dibutuhkan dalam penelitian ini.

##### b. Metode Wawancara

Metode wawancara ini dilakukan dengan cara melakukan tanya jawab kepada karyawan dan petugas di dalam Departemen Pergudangan PT Tetra Pak Stainless Equipment Jakarta untuk mendapatkan informasi sistem pengeluaran material yang ada pada perusahaan, sistem pemesanan material dari departement Produksi ke Pergudangan.

##### c. Metode Analisis Dokumen

Metode ini dilakukan dengan cara melakukan analisis terhadap dokumen-dokumen yang berkaitan dalam pemenuhan kebutuhan material bagi produksi dan pergudangan di PT Tetra Pak Stainless Equipment Jakarta.

#### 2. Studi Pustaka

Metode pengumpulan data dilakukan dengan cara membaca berbagai buku atau literatur yang berhubungan dengan sistem informasi, sistem produksi, perencanaan kebutuhan material, analisis dan desain sistem informasi, pengetahuan mengenai PHP dan MySQL, pengembangan perangkat lunak, dan berbagai contoh kode pemrograman dari media *internet*.

### 3.3 Identifikasi Masalah

Proses ini mencakup pemilihan, penentuan dan penetapan permasalahan yang dihadapi. Dalam hal ini permasalahan yang dihadapi oleh PT Tetra Pak Stainless Equipment Jakarta:

1. Proses pengolahan data penerimaan dan pengeluaran material di PT Tetra Pak Stainless Equipment Jakarta masih menggunakan *Microsoft Excel*, sehingga masih menimbulkan redundansi data.
2. Proses pengeluaran material berdasarkan permintaan dari bagian produksi untuk kebutuhan produksi masih menggunakan *form* permintaan material yang ditulis kemudian diantarkan ke bagian gudang dan harus menunggu material dikeluarkan untuk dibawa lagi ke bagian produksi. Karena itu sering terjadi *form* atau kertas permintaan material hilang, rusak atau terselip.
3. Proses permintaan material dari produksi ke gudang masih belum terintegrasi sehingga membutuhkan banyak waktu.

### **3.4 Tujuan Penelitian**

Tujuan penelitian merupakan penentuan tujuan akhir dari penelitian yang telah dilakukan dan diharapkan dapat menjawab permasalahan yang dihadapi oleh organisasi. Tujuan dari penelitian ini adalah merancang dan membangun sebuah sistem terintegrasi yang berfungsi sebagai berikut:

1. Menerapkan penggunaan database MySQL pada PT Tetra Pak Stainless Equipment Jakarta untuk menghilangkan redundansi.
2. Membuat *form* secara terkomputerisasi sehingga dapat mengurangi penggunaan kertas.
3. Membuat aplikasi penerimaan dan pengeluaran material gudang ke produksi menggunakan PHP 5.4.7 dan MySQL 5.5.27 Pada PT Tetra Pak Stainless Equipment untuk membantu sistem pergudangan dalam penerimaan dan pengeluaran material gudang ke produksi untuk kebutuhan produksi.

### **3.5 Batasan Masalah**

Batasan masalah merupakan hal-hal yang membatasi permasalahan dalam sebuah penelitian agar lebih fokus dan terarah. Batasan masalah pada penelitian ini sebagai berikut:

1. Tempat pengamatan adalah PT Tetra Pak Stainless Equipment, Pulo Gebang Jakarta Timur.

2. Pengamatan dilakukan di bagian gudang penyimpanan material pada PT Tetra Pak Stainless Equipment Jakarta.
3. Pembatasan masalah hanya mengamati proses penerimaan dari *supplier* berdasarkan data *delivery order* dan pengeluaran material gudang ke produksi di PT Tetra Pak Stainless Equipment Jakarta.

### **3.6 Metode Pengembangan Sistem**

Metode pengembangan sistem informasi penerimaan dan pengeluaran material gudang ke produksi berbasis *website* ini menggunakan model pengembangan sistem *prototyping*. Metode pengembangan sistem ini dipilih karena cocok digunakan untuk menjabarkan kebutuhan *user* yang *detail*.

Tahapan-tahapan pada model *prototype*, sebagai berikut (Rosa dan Shalahuddin, 2011):

1. Mengumpulkan kebutuhan pelanggan atau *user* terhadap perangkat lunak yang akan dibuat.
2. Membuat *prototype* agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program *prototype* biasanya merupakan program yang belum jadi.
3. Program *prototype* selanjutnya dievaluasi oleh pelanggan atau *user* sampai ditemukan spesifikasi yang sesuai dengan keinginan pelanggan atau *user*.

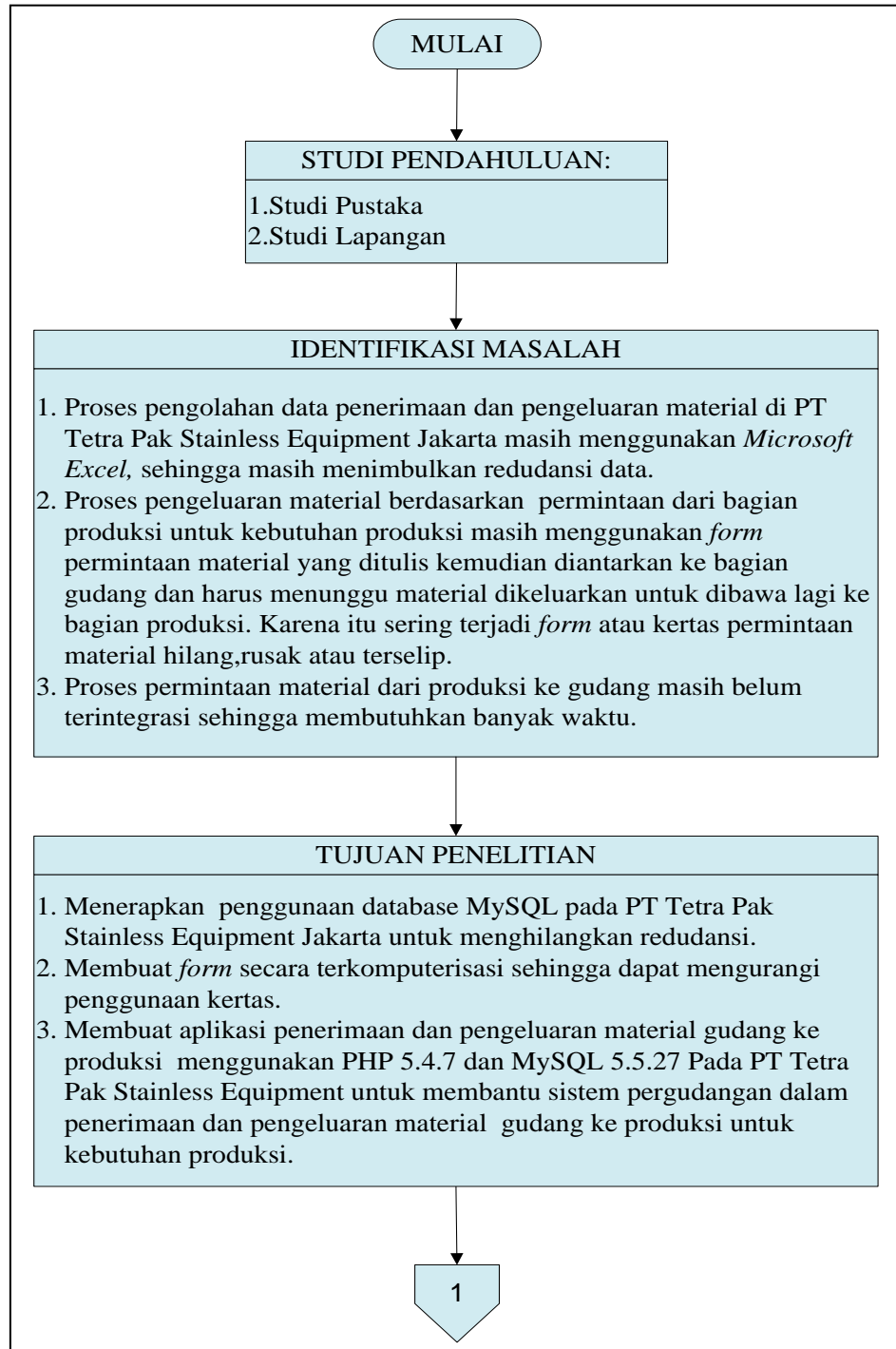
### **3.7 Kesimpulan dan Saran**

Setelah seluruh tahap penelitian dilakukan, langkah selanjutnya adalah membuat kesimpulan terhadap penelitian yang telah dilakukan. Kesimpulan yang dibuat harus berkaitan dengan tujuan awal penelitian sistem informasi produksi yang dilakukan serta memberikan usulan yang dapat diberikan pada penelitian selanjutnya.

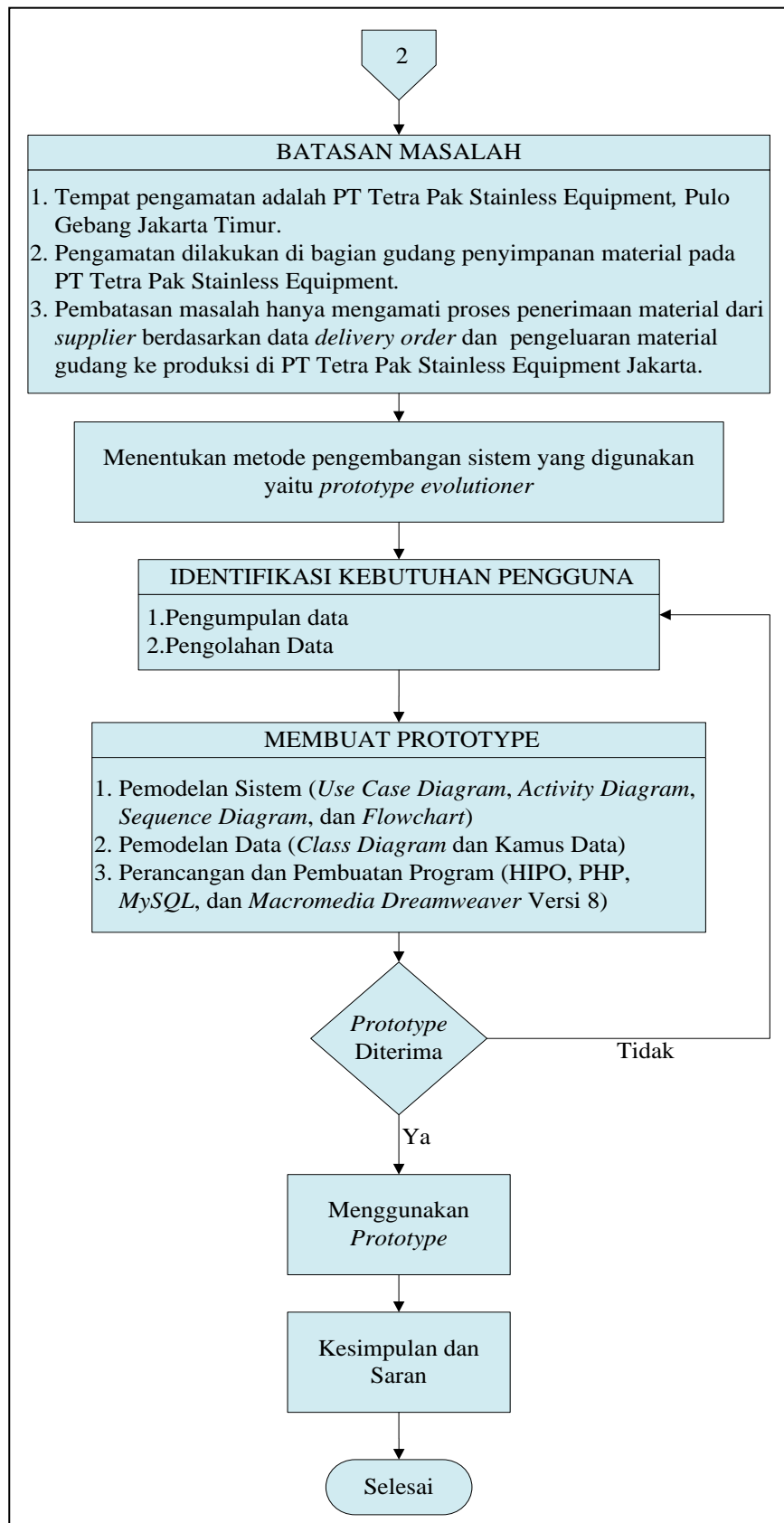
### **3.8 Kerangka Penelitian**

Dalam pemecahan masalah yang dibahas dalam Tugas Akhir ini, dilakukan langkah-langkah atau tahapan dalam pemecahan masalah yang ada pada

Gambar III.1 sesuai dengan metodologi penelitian yang penulis gunakan dalam pemecahan masalah tersebut. Penjelasan langkah-langkah atau tahapan dalam pemecahan masalah tersebut adalah sebagai berikut:



Gambar III.1 Kerangka Penelitian



Gambar III.2 Kerangka Penelitian (lanjutan)

Agar penelitian fokus dan terarah, berikut ini tahapan-tahapan yang dilakukan dalam melakukan penelitian:

1. Tahap awal yang dilakukan yaitu studi pendahuluan.

Tahap yang dilakukan dalam penelitian ini dengan studi lapangan dan studi pustaka untuk memahami pengetahuan umum tentang pergudangan.

2. Tahap selanjutnya yaitu identifikasi masalah.

Pada tahap ini permasalahan yang ditemukan selama penelitian yaitu pengolahan data penerimaan dan pengeluaran *Microsoft excel*, proses pengeluaran material berdasarkan permintaan dari bagian produksi untuk kebutuhan produksi masih menggunakan surat kertas permintaan material yang ditulis kemudian di antarkan ke bagian gudang dan harus menunggu material dikeluarkan untuk dibawa lagi ke bagian produksi, terjadinya Kertas atau form permintaan material hilang, terselip, rusak. Membutuhkan banyak waktu dalam melakukan permintaan material gudang ke produksi, laporan atau pengecekan laporan, dan berpengaruh pada waktu produksi. Penelitian yang dilakukan bertujuan untuk merancang dan membangun aplikasi pengeluaran material sehingga memudahkan dan membantu kinerja departemen produksi dan departemen pergudangan dalam memaksimalkan waktu proses produksi dan pemesanan material yang efektif dan efisien.

3. Menentukan batasan masalah.

Batasan masalah dalam penelitian ini terlebih ke pengamatan proses penerimaan dan pengeluaran material di bagian gudang material PT Tetra Pak Stainless Equipment Jakarta.

4. Menentukan metode pengembangan sistem.

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode pengembangan sistem *prototyping evolusioner*.

5. Identifikasi kebutuhan pengguna (*user*).

Tahap awal metode pengembangan sistem *prototyping* yaitu menjabarkan segala sesuatu yang dibutuhkan dan diinginkan pengguna pada sistem yang akan dibuat.

## 6. Pengembangan sistem menggunakan *Evolutionary Prototype*

Penggunaan *evolutionary prototype* dimulai dengan empat tahapan yang diantaranya adalah sebagai berikut:

### a. Identifikasi kebutuhan pengguna

Pada tahap ini dimulai dari analisis kebutuhan pengguna untuk mengetahui sistem yang akan dibangun

### b. Membuat *prototype*

Membuat *prototype* terdiri dari enam macam antara lain:

- 1) Perancangan model sistem yang diusulkan menggunakan UML
- 2) Perancangan data menggunakan *class diagram* dan kamus data
- 3) Perancangan *flowchart* program yang diusulkan
- 4) Perancangan *interface* sistem informasi yang diusulkan
- 5) Pembuatan *prototype* dengan menggunakan bahasa pemrograman PHP 5.4.7 dan MySQL 5.5.27 sebagai basis data
- 6) Spesifikasi *software* dan *hardware*

### c. Menentukan apakah *prototype* dapat diterima

Pada tahap ini pula akan dilihat apakah *prototype* yang dibuat dapat diterima oleh pengguna atau tidak. Jika tidak, akan dilakukan kembali mengidentifikasi kebutuhan pengguna, tetapi jika ya, tahap selanjutnya adalah menggunakan *prototype* tersebut.

### d. Menggunakan *prototype*

Pada tahap ini merupakan hasil dari keputusan yang diambil berdasarkan tahap evaluasi yang berarti bahwa sistem yang telah dirancang sudah sesuai dengan kebutuhan pengguna dan akan digunakan.

## 7. Kesimpulan dan Saran

Mengambil kesimpulan dari perbandingan hasil analisis sistem yang berjalan dengan sistem yang diusulkan, serta dapat memberikan saran kepada PT Tetra Pak Stainless Equipment Jakarta.

## **BAB IV**

### **PENGUMPULAN DAN PENGOLAHAN DATA**

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

PT Tetra Pak Stainless Equipment Jakarta memulai usahanya pada awal tahun 1950-an sebagai salah satu dari perusahaan multinasional pertama di bidang pemrosesan dan pengemasan. Sejak itu PT Tetra Pak Stainless Equipment Jakarta menjadi salah satu pemasok terbesar di dunia untuk sistem pembungkus susu, jus buah serta minuman, dan berbagai produk lainnya. Pada tahun 1991, PT Tetra Pak Stainless Equipment Jakarta melebarkan usahanya ke peralatan pengolahan makanan cair, teknik instalasi, dan peralatan produksi keju. PT Tetra Pak Stainless Equipment Jakarta adalah satu-satunya perusahaan internasional di dunia yang mampu menyediakan solusi pengolahan, pengemasan dan jalur distribusi serta instalasi terintegrasi untuk produksi makanan. Ini berarti para pelanggan PT Tetra Pak Stainless Equipment Jakarta menikmati manfaatnya karena mereka bisa mendapatkan solusi untuk berbagai produk dari satu sumber saja, dengan peralatan yang sesuai di setiap tahap.

Sejak didirikan, perusahaan telah beberapa kali mengalami perubahan nama. Pada awalnya PT TPSE bernama PT Indo Lava, dan pada tahun 1997 berganti nama menjadi PT Tetra Pak Indonesia melalui akta notaris No. 20 tanggal 09 Juni 1997 oleh Notaris Ny. R. Arie Soetardjo, SH. Kemudian ditahun 2001 kembali bernama PT Indo Lava, karena nama Tetra Pak Indonesia telah dipakai oleh Brikindo Jaya yang dilikuidasi. Namun pada tanggal 03 Juni 2004 berganti nama menjadi PT Tetra Pak Stainless Equipment Jakarta. Perubahan tersebut bertujuan untuk mendekatkan kepada *brand* dan *image* Tetra Pak, yang diharapkan mampu meningkatkan *sales* dan *order receive*, keuntungan perusahaan dan kemajuan yang lebih baik.

Dan pada tanggal 15 Maret 2005 PT TPSE menjadi kawasan berikat yang merangkap dalam perusahaan dalam kawasan berikat (PDKB) dengan SK Menkeu No.15/KMK.04/2005 tanggal 13 Januari 2005.

PT TPSE juga menyediakan bahan pembungkus dari jaringan fasilitas-fasilitas produksi di seluruh dunia. Pabrik-pabrik ini menyediakan kiriman rutin ke 9.115 mesin pengemas PT TPSE yang saat ini sudah dipasang di instalasi- instalasi pelanggan di seluruh dunia. PT TPSE menyediakan solusi pemrosesan dalam lima kategori makanan yaitu susu, keju, es krim, minuman dan makanan jadi. PT TPSE menyediakan instalasi yang dijamin performanya dan memberikan bantuan instalasi tersebut. Termasuk di dalamnya adalah sistem otomasi pabrik yang terintegrasi, lengkap dengan alat pelacak produk serta layanan teknik instalasi yang lengkap untuk melindungi keamanan makanan. PT TPSE bisa memberikan solusi jalur yang lengkap dari penerimaan produk sampai penanganan terakhir dan pengemasannya. PT TPSE menawarkan jalur lengkap untuk pasturisasi dan modul UHT, *homogenisers*, separator (alat pemisah), unit standardisasi, sistem CIP, *heat exchanger* (alat penukar panas), *flow equipment* (alat pengalir), dan komponen lainnya. Untuk industri keju pada PT Tetra Pak Stainless Equipment Jakarta dapat menawarkan peralatan dan jalur dari penerimaan susu mentah sampai barang jadi (termasuk pengemasan untuk keju berisi cairan). Tetra Pak Tebel dalam Tetra Pak *Processing System* berkonsentrasi pada lima segmen yaitu keju cedar, keju keras, keju semi keras, keju lembut, keju segar (berisi cairan). Tetra Pak *Processing System* juga termasuk Tetra Pak *Hoyer*, pemasok terbesar dengan layanan penuh untuk sistem, peralatan, barang habis pakai dan bantuan lengkap untuk pemrosesan dan pengemasan es krim.

Sistem pengemasan Tetra Rex menawarkan solusi alternatif baru untuk mengisinya dengan makanan kering. Bentuk karton *gable-top* (segitiga di bagian atas), penutup dan modelnya sedang dimodifikasi agar dapat memenuhi kebutuhan para pelanggan akan kemasan makanan kering dengan cara yang lebih baik.

#### 4.1.1 Profil Perusahaan

PT Tetra Pak Stainless Equipment Jakarta mempunyai 41 perusahaan pemasaran dan 74 kantor penjualan di seluruh dunia, 43 pabrik untuk kemasan dan penutup dan 10 pabrik perakitan mesin kemasan. PT TPSE mempunyai 21,640 karyawan dan pada tahun 2008. Produk-produk Tetra Pak dijual pada lebih 150 pasar. PT TPSE merupakan perusahaan gabungan antara perusahaan swasta Swedia Alfa Laval (Tetra Laval Group) dengan *Alpha Austenite* perusahaan swasta Indonesia. PT TPSE bergerak dalam bidang manufaktur, dan berbadan hukum Perseroan Terbatas (PT), terletak di Jl. Raya Pulogebang KM 3, Cakung Jakarta Timur 13950 dengan luas area 13.000 m<sup>2</sup> dan luas bangunan 5.000 m<sup>2</sup>. Diaktakan oleh Notaris Kartini Muljadi, SH, dengan akta notaris No.7 pada tanggal 06 Januari 1984.

Tetra Pak dalam Angka Erat bekerja sama dengan para pelanggan dan pemasok, PT TPSE menyediakan produk yang aman, inovatif, dan ramah lingkungan setiap harinya untuk memenuhi kebutuhan ratusan juta orang di lebih dari 170 negara seluruh dunia, seperti:

- a. Mesin kemasan yang beroperasi pada tahun 2012 sebanyak 8,688
- b. Mesin kemasan yang dikirim pada tahun 2011 sebanyak 520
- c. Unit pemrosesan yang beroperasi pada tahun 2012 sebanyak 66,003
- d. Unit pemrosesan yang dikirim pada tahun 2011 sebanyak 2,250
- e. Peralatan distribusi yang beroperasi pada tahun 2012 sebanyak 16,794
- f. Peralatan distribusi yang dikirim pada tahun 2011 sebanyak 1,659
- g. Pabrik perakitan mesin sebanyak 9 pabrik
- h. Pabrik produksi kemasan dan penutup sebanyak 42 pabrik
- i. Negara kemasan Tetra Pak tersedia lebih dari 170 negara
- j. Perusahaan Pemasaran sebanyak 38
- k. Kantor Penjualan sebanyak 79
- l. Jumlah karyawan saat ini sebanyak 22,896 orang
- m. Pusat Pelatihan Teknis sebanyak 11

#### **4.1.2 Visi dan Misi Perusahaan**

##### **Visi TPSE:**

Komitmen PT Tetra Pak Stainless Equipment Jakarta adalah untuk menyediakan makanan yang aman dimanapun juga.

##### **Misi TPSE:**

1. PT Tetra Pak Stainless Equipment Jakarta bekerja untuk dan bersama para pelanggan untuk memberi solusi yang tepat di bidang pemrosesan dan pengemasan pangan.
2. PT Tetra Pak Stainless Equipment Jakarta bekerjasama secara erat dengan para pelanggan dan pemasok untuk menyediakan produk yang aman, inovatif dan ramah lingkungan yang setiap harinya memenuhi kebutuhan ratusan juta orang di seluruh dunia.
3. PT Tetra Pak Stainless Equipment Jakarta percaya akan kepemimpinan industri yang bertanggung jawab dalam menciptakan pertumbuhan yang menguntungkan, seiringdengan keinginan untuk menjadi warga korporat yang baik dan pendekatan bisnis yang berkelanjutan.

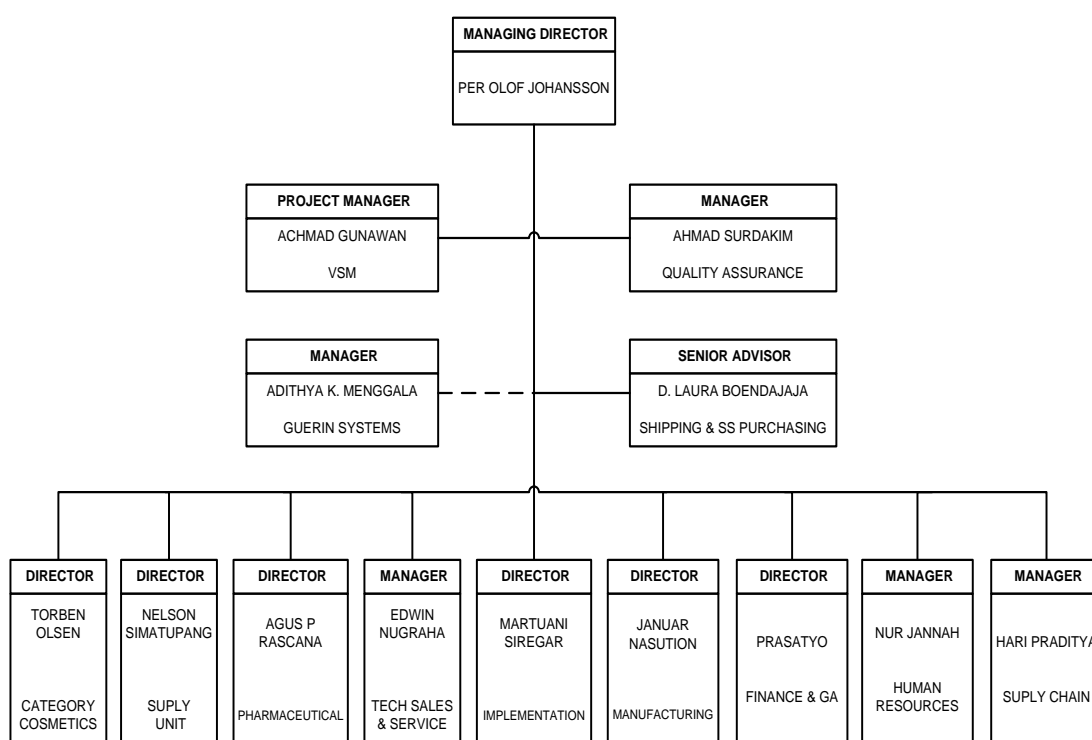
#### **4.1.3 Struktur Organisasi Perusahaan**

Organisasi merupakan sebuah perkumpulan atau wadah bagi sekelompok orang untuk bekerjasama, terkendali dan dipimpin untuk tujuan tertentu. Agar suatu organisasi bekerja dengan baik maka diperlukan struktur organisasi. Struktur organisasi adalah suatu susunan dan hubungan antara tiap bagian serta posisi yang ada pada suatu organisasi atau perusahaan dalam menjalankan kegiatan operasional untuk mencapai tujuan.

Struktur organisasi didalam suatu perusahaan memegang peranan yang sangat pentingn karena efektifitas dan kinerja dari suatu perusahaan sangat ditentukan oleh struktur organisasi diperusahaan tersebut. Dengan adanya struktur organisasi dapat memudahkan komunikasi dan koordinasi tiap departemen sehingga tercipta manajemen yang baik.

Struktur organisasi yang digunakan PT Tetra Pak Stainless Equipment Jakarta adalah struktur lini yakni bersifat fleksibel dengan adanya spesialisasi dalam bekerja, koordinasi dan *job description* serta tanggung jawab yang jelas.

Dalam struktur organisasi PT Tetra Pak Stainless Equipment Jakarta dipimpin oleh seorang *Manager Director* yang membawahi beberapa *director* yang berjumlah 9 divisi yang masing-masing dikepalai oleh seorang direksi. Departemen secara keseluruhan berjumlah 17 di mana dikepalai oleh seorang *manager*.



Gambar IV.1 Struktur Organisasi PT Tetra Pak Stainless Equipment Jakarta  
(Sumber : PT Tetra Pak Stainless Equipment Jakarta, 2016)

Berikut ini merupakan *job description* dari jabatan-jabatan yang ada pada PT Tetra Pak Stainless Equipment Jakarta:

1. *Managing Director*

*Managing Director* bertugas untuk mengawasi kegiatan dari management team (Direktur dan Manajer Divisi) dan melaporkan hasil perkembangan perusahaan ke *Head Quarter* (Kantor Pusat) di Lund, Sweden.

2. *Business Unit Director*

Business Unit Director dibagi menjadi tiga bagian :

a. *Category Cosmetic*

*Category Cosmetic Director* bertugas untuk mencari dan mendapatkan pasar di *industry cosmetic* baik yang berada di area lokal (dalam negeri) maupun pasar global (dunia). Melaporkan kegiatan dan perkembangan bisnisnya pada *Managing Director*.

b. *Supply Unit*

*Supply Unit Director* bertugas untuk mencari dan mendapatkan pasar di *industry food and beverage* baik yang berada di area local (dalam negeri) maupun di pasar global (dunia). Melaporkan kegiatan dan perkembangan bisnisnya pada *Managing Director*.

c. *Pharmaceutical*

*Pharmaceutical Director* bertugas untuk mencari dan mendapatkan pasar *industry pharmacy* baik yang berada di area local (dalam negeri) dan pada saat ini sedang mencoba mengembangkan bisnisnya di India dan Cina. Melaporkan kegiatan dan perkembangan bisnisnya pada *Managing Director*.

3. *Pharma Business Development Director*

*Pharma Business Development Director* bertugas untuk mengembangkan dan mencari pasar baru di industri farmasi khusus untuk pasar global. Pada saat ini sedang mencoba untuk mendapatkan pasar India dan Cina. Melaporkan kegiatan dan perkembangan pasar pada *Managing Director*.

4. *Product Safety Manager*

*Product Safety Manager* bertugas untuk mengumpulkan data regulasi standar pembuatan bejana bertekanan (*pressure vessel*) untuk berbagai Negara, mendapatkan sertifikasi pembuatan *pressure vessel* berdasarkan code yang ditetapkan oleh Negara tempat dimana dilakukan instalasi, serta memastikan bahwa *pressure vessel* yang akan dikirim sudah sesuai dengan aturan yang ditetapkan oleh Negara tujuan. Melaporkan kegiatan kepada *Managing Director*.

5. *Process Office Manager*

*Process Office Manager* bertugas untuk memberikan saran mengenai proses manufaktur sehingga mendapatkan hasil yang baik dengan tingkat efisiensi yang tinggi, melaporkan kegiatan pada *Managing Director*.

6. *Technical Sales & Service Manager*

*Technical Sales & Service Manager* bertugas untuk menerima kalim dari pelanggan mengenai kerusakan yang terjadi pada instalasi peralatan, mengembangkan bisnis penjualan suku cadang (spare parts), melaporkan kegiatan kepada *Managing Director*.

7. *Implementation Director*

*Implementation Director* bertugas sebagai tim eksekusi proyek, membawahi *project manager* dan *engineering manager*, memastikan bahwa proyek berjalan dengan baik sesuai dengan waktu dan biaya yang sudah ditentukan, melaporkan perkembangan proyek kepada *Managing Director*.

8. *Manufacturing Director*

*Manufacturing Director* bertugas untuk memastikan bahwa proses *manufactur* berjalan dengan baik dan sesuai dengan waktu pengiriman yang telah ditetapkan, membawahi *Production Manager*, *PPIC Manager*, *Warehouse Manager*, *Quality Control Manager*, *Maintenance Manager* dan *Order Owner*, melaporkan perkembangan manufaktur kepada *Managing Director*.

9. *Finance & GA Director*

*Finance & GA Director* bertugas untuk menangani masalah keuangan di perusahaan, memastikan kondisi keuangan dalam keadaan sehat dan pembayaran juga penerimaan kas berjalan dengan baik pula, selain itu juga bertugas untuk mengurus perizinan legal dan kebutuhan karyawan, melaporkan perkembangan dan keadaan perusahaan kepada *Managing Director*.

10. *Human Resources Manager*

*Human Resources Manager* bertugas untuk pengelolaan dan pengembangan sumber daya manusia yaitu dalam hal perencanaan, pelaksanaan dan pengawasan kegiatan sumber daya manusia, termasuk pengembangan kualitas dengan pedoman kebijaksanaan dan prosedur yang berlaku di perusahaan, melaporkan perkembangan dan kegiatan karyawan kepada *Managing Director*.

11. *Strategic Purchasing Manager*

*Strategic Purchasing Manager* bertugas untuk mengatur pembelian material dan peralatan yang diperlukan oleh proyek, memastikan bahwa harga material dan barang yang dibeli kompetitif, mengatur *vendor* yang boleh bertransaksi dengan perusahaan, mengatur jumlah pembelian Material yang diperlukan untuk proses manufaktur, melaporkan perkembangan harga dan stok material kepada *Managing Director*.

#### **4.2 Material Pada Gudang Tetra Pak Stainless Equipment Jakarta**

Peran material dalam suatu produksi sangatlah penting karena material ini adalah satu dari beberapa tahapan dalam proses produksi. Dari material inilah akan diproses menjadi barang jadi atau barang setengah jadi. PT Tetra Pak Stainless Equipment Jakarta sangat mengutamakan kualitas dari material yang nantinya akan diolah dalam proses produksi pembuatan *Tetra Alsafe*.

Material utama yang digunakan pada PT Tetra Pak Stainless Equipment Jakarta adalah Plat Besi *Stainless* dengan tipe 304 dan 316. Dalam *metalurgi, stainless steel*, juga dikenal sebagai *inox* baja atau *inox* dari bahasa Perancis "*inoxydable*", adalah baja paduan dengan minimal 10,5% kromium konten oleh massa.

*Stainless steel* tidak mudah menimbulkan korosi, karat atau noda dengan air sebagai baja biasa tidak, tapi meskipun nama itu tidak sepenuhnya noda, terutama di bawah oksigen rendah, salinitas tinggi, atau lingkungan sirkulasi yang buruk. Ada nilai yang berbeda dan selesai permukaan *stainless steel* untuk menyesuaikan dengan lingkungan paduan harus bertahan.

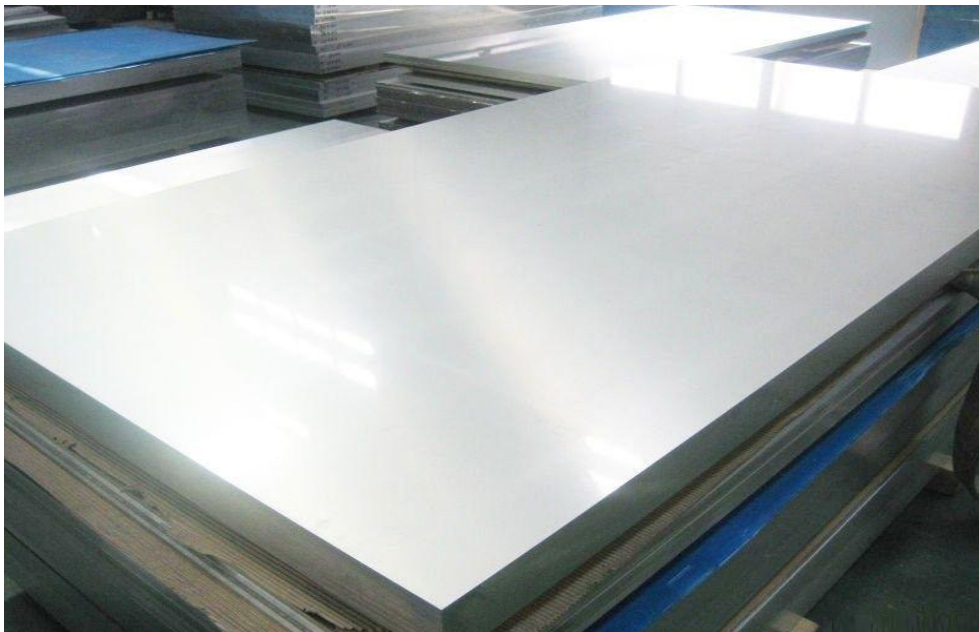
*Stainless steel* digunakan di mana kedua sifat baja dan ketahanan terhadap korosi yang diperlukan. *Stainless steel* berbeda dari baja karbon dengan jumlah kromium. *Unprotected* baja karbon karat mudah bila terkena udara dan kelembaban. Ini oksida besi (karat) aktif dan mempercepat korosi dengan membentuk oksida besi yang lebih, dan karena *volume* yang lebih besar dari oksida besi ini cenderung mengelupas.

Baja *stainless* mengandung cukup kromium untuk membentuk pasif film kromium oksida, yang mencegah korosi permukaan lanjut dengan memblokir difusi oksigen ke permukaan baja dan blok korosi dari menyebar ke struktur internal logam, dan karena ukuran yang sama dari ion besi dan oksida mereka ikatan sangat kuat dan tetap melekat ke permukaan.

#### **4.2.1 *Stainless Steel 304***

*Stainless Steel* Tipe 304 merupakan variasi dari dasar Tipe 302, dengan tinggi kromium dan kandungan karbon yang lebih rendah. Rendah karbon meminimalkan kromium mengulur curah hujan karena pengelasan dan yang kerentanan terhadap korosi. Sedangkan tipe 302 untuk mempertahankan ketahanan korosi yang memadai. Tipe 304L adalah tambahan karbon rendah variasi Tipe 304 dengan maksimum 0,03% kandungan karbon yang menghilangkan karbida curah hujan akibat pengelasan. Akibatnya, paduan ini dapat digunakan dalam kondisi, bahkan dalam berat korosi kondisi baik. Ini sering menghilangkan perlunya *weldments* kecuali aplikasi menentukan menghilangkan stres. Itu memiliki sifat mekanik sedikit lebih rendah tipe dari 304.

*Stainless Steel* Tipe 304 adalah yang paling fleksibel dan *stainless steel* yang paling banyak digunakan, tersedia dalam berbagai jenis produk, bentuk dan selesai dari yang lain. *Stainless Steel Tipe 304* ini memiliki karakteristik pembentukan dan pengelasan baik. Struktur *austenitic* seimbang kelas 304 memungkinkan untuk sangat mendalam ditarik tanpa perantara yang telah membuat kelas ini dominan dalam pembuatan bagian-bagian *stainless* ditarik. Kelas 304 juga memiliki karakteristik pengelasan yang luar biasa.



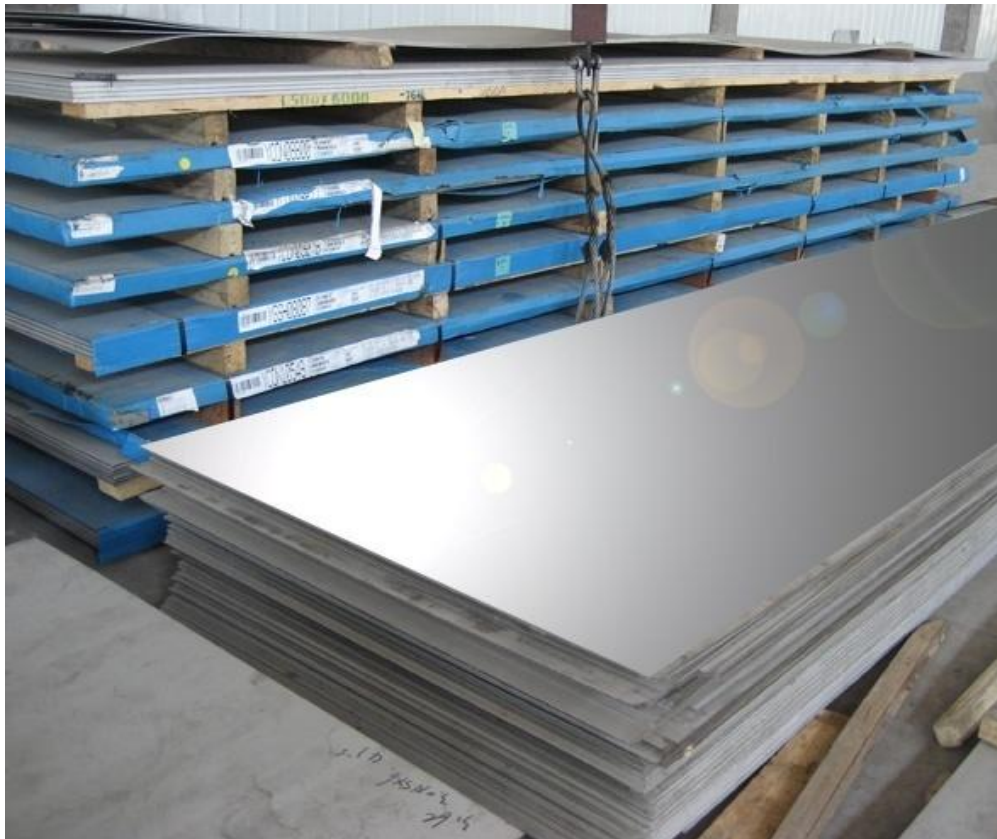
Gambar IV.2 *Stainless Steel 304*  
(Sumber: PT Tetra Pak Stainless Equipment Jakarta, 2016)

#### 4.2.2 *Stainless Steel 316*

Tipe 316 adalah kromium *stainless steel* nikel yang melawan panas dengan ketahanan korosi unggul dibandingkan dengan baja krom nikel bila terkena berbagai jenis yang dapat menyebabkan korosi kimia seperti air laut, solusi air garam, dan sejenisnya. Ketahanan korosi ditingkatkan, partikulasi terhadap sulfat, klorida, asetat, format dan tartrat asam, asam sulfat dan klorida basah. Tipe 316L adalah karbon ekstra rendah Tipe 316 yang meminimalkan karbida akibat pengelasan. Tipe 316 dan 316L Baja *Stainless exhibit* ketahanan korosi yang lebih baik daripada Tipe 304. PT Tetra Pak Stainless Equipment Jakarta menyediakan *pitting* baik resistensi dan ketahanan yang baik terhadap sebagian bahan kimia.

Kelas 316 adalah kelas *molybdenum* standar, kedua pentingnya untuk 304 antara baja tahan karat *austenit*. *Molibdenum* memberikan Tipe 316 sifat yang lebih baik secara keseluruhan tahan korosi daripada Tipe 304, resistensi terutama tinggi terhadap korosi *pitting* dan celah dalam lingkungan klorida. Ini memiliki karakteristik pembentukan dan pengelasan baik. Hal ini mudah rem atau memutar dibentuk menjadi berbagai suku cadang untuk aplikasi dalam industri, arsitektur,

dan bidang transportasi. Kelas 316 juga memiliki karakteristik pengelasan yang luar biasa. Struktur *austenitic* juga memberikan nilai ketangguhan ini sangat baik, bahkan sampai kesuhu *cryogenic*.

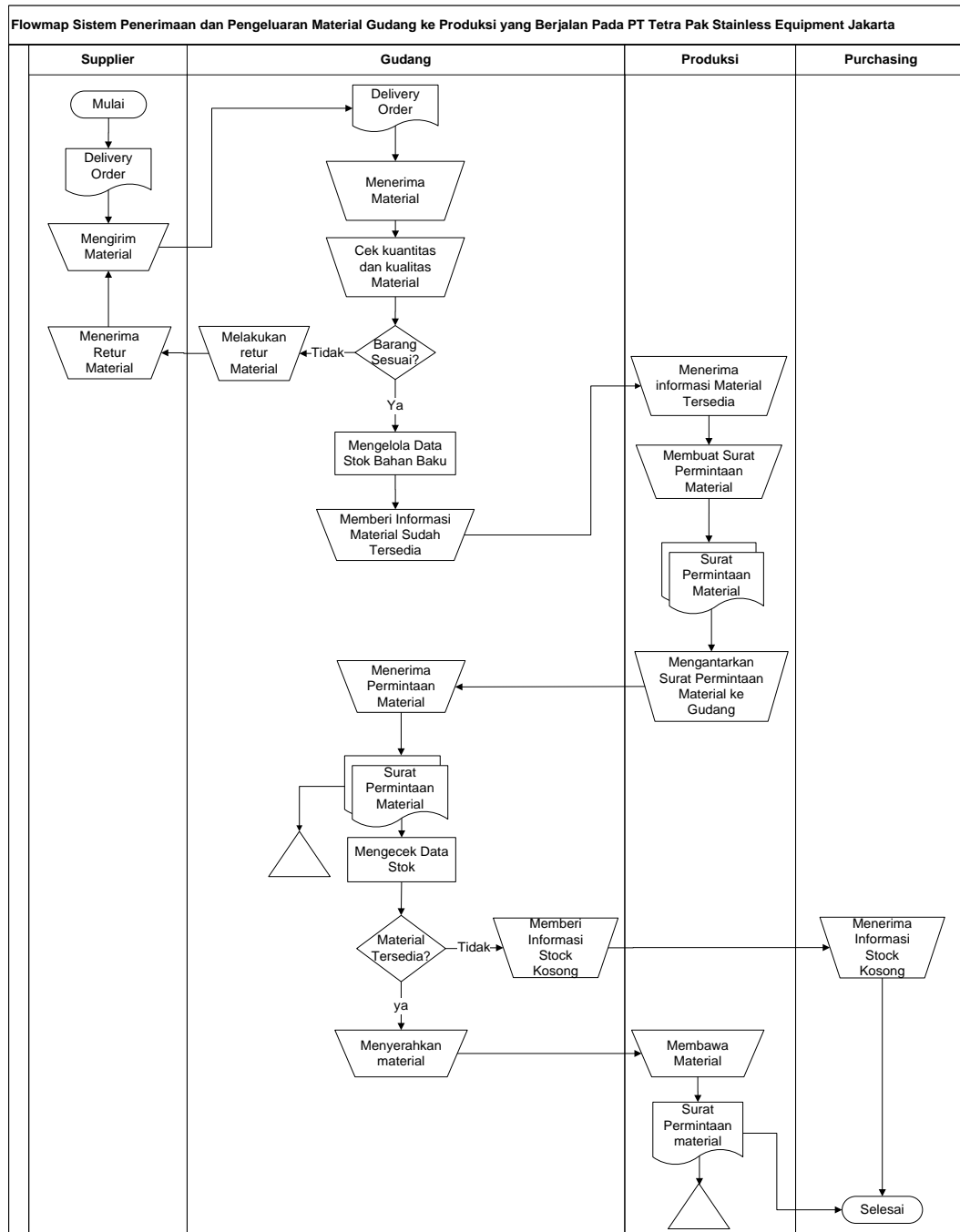


Gambar IV.3 *Stainless Steel 316*  
(Sumber: PT Tetra Pak Stainless Equipment Jakarta, 2016)

### 4.3 Sistem Yang Berjalan

Untuk mengetahui sistem yang berjalan dilakukan pengamatan bagaimana cara kerja sistem yang digunakan di PT Tetra Pak Stainless Equipment Jakarta khususnya mengenai rancang bangun aplikasi penerimaan dan pengeluaran material gudang ke produksi. Tujuan melakukan pengamatan sistem tersebut adalah agar dapat mengidentifikasi masalah-masalah yang ada pada sistem informasi penerimaan dan pengeluaran material gudang ke produksi sehingga dapat melakukan saran-saran perbaikan. Sistem yang diamati pada saat sedang berjalan dan bertujuan untuk mengetahui lebih jelas bagaimana cara kerja sistem tersebut serta masalah apa yang sedang dihadapi oleh sistem sehingga lebih

mudah dalam merancang dan membangun sebuah perangkat lunak yang akan diusulkan. Berikut *flowmap* sistem yang sedang berjalan di PT Tetra Pak Stainless Equipment Jakarta.



Gambar IV.4 *Flowmap* Sistem Penerimaan dan Pengeluaran Material Gudang ke Produksi yang Berjalan di PT Tetra Pak Stainless Equipment Jakarta  
(Sumber: PT Tetra Pak Stainless Equipment Jakarta, 2016)

Berikut adalah penjelasan Gambar di atas tentang proses penerimaan dan pengeluaran material di PT Tetra Pak Stainless Equipment Jakarta adalah:

1. Bagian *Supplier* mengirimkan *delivery order* .
2. Bagian *Supplier* mengantarkan material pesanan ke bagian gudang.
3. Bagian gudang menerima *delivery order* dari *supplier*.
4. Bagian gudang menerima material dari *supplier*.
5. Bagian gudang melakukan pengecekan kuantitas dan kualitas material.
6. Setelah dicek oleh bagian gudang untuk barang yang tidak sesuai akan dikembalikan ke *supplier* dan diretur.
7. Jika sesuai maka material yang telah di uji kualitas dan kuantitas akan disimpan ke dalam gudang.
8. Bagian gudang melakukan pengolahan data material.
9. Bagian gudang member informasi kepada bagian produksi bahwa material sudah tersedia dan siap untuk dilakukan permintaan.
10. Bagian produksi menerima Informasi material sudah tersedia.
11. Bagian produksi membuat surat permintaan material.
12. Bagian produksi mengantarkan surat permintaan material ke gudang.
13. Bagian gudang menerima surat permintaan material dari bagian produksi.
14. Bagian gudang mengecek stok material di daftar stok material, apabila material tersedia maka bagian gudang akan melakukan cek fisik dan mengambil bahan baku yang diminta oleh bagian produksi.
15. Bagian gudang menyerahkan material kepada bagian produksi yang meminta.
16. Bagian gudang menyimpan rangkap dua surat permintaan material dan menyerahkan rangkap satu kepada bagian produksi.
17. Bagian produksi membawa material dan rangkap satu surat permintaan material.
18. Apabila material kosong maka bagian gudang melaporkan kepada bagian *purchasing*.
19. Bagian *purchasing* menerima laporan kekurangan stok dari bagian gudang.

#### 4.4 Data Terkait

Dokumen-dokumen yang terlibat dalam rancang bangun pengeluaran bahan baku yang berjalan pada PT Tetra Pak Stainless Equipment Jakarta adalah sebagai berikut:

##### 1. *Delivery order* dari Supplier

Product Description	Capacity	UOM	Quantity
ALCOHOL 10100, 10101, 10102, 10103, 10104, 10105, 10106, 10107, 10108	7.10	TUBE	10
ALCOHOL 10109 10110, 10111, 10112, 10113	7.10	TUBE	5
<b>Total Cylinder</b>			<b>15</b>

Gambar IV.5: *Delivery order*  
(Sumber : PT Tetra Pak Stainless Equipment Jakarta, 2016)

*Delivery Order* adalah surat yang berfungsi sebagai surat perintah pengantaran barang yang tercantum didalamnya kepada penerima berdasarkan yang telah dipesan oleh bagian *purchasing* dan akan diterima oleh bagian *incoming* gudang beserta materialnya dan akan diperiksa kuantitas dan kualitas.

Berikut adalah keterangan gambar IV.5 surat *delivery order*:

- To* : berisi alamat perusahaan penerima atau pembeli.
- IGGI no* : berisi nomor *delivery order*.
- Date* : berisi tanggal pengiriman.
- Product description* : berisi nama-nama barang atau material yang dikirim berdasarkan pemesanan atau pembelian.

- e. *Capacity* : berisikan kapasitas dari barang yang di dipesan.  
 f. *UOM* : berisikan satuan barang yang dipesan.  
 g. *Quantity* : berisikan jumlah barang yang telah dipesan.

## 2. Form permintaan material.

**FORM PERMINTAAN MATERIAL**

No : \_\_\_\_\_  
 Tanggal : \_\_\_\_\_

No	Nama Material	Qty	Keterangan

Yang Mengetahui Yang Mengajukan Permintaan

.....  
 Nama Jelas .....  
 Nama Jelas

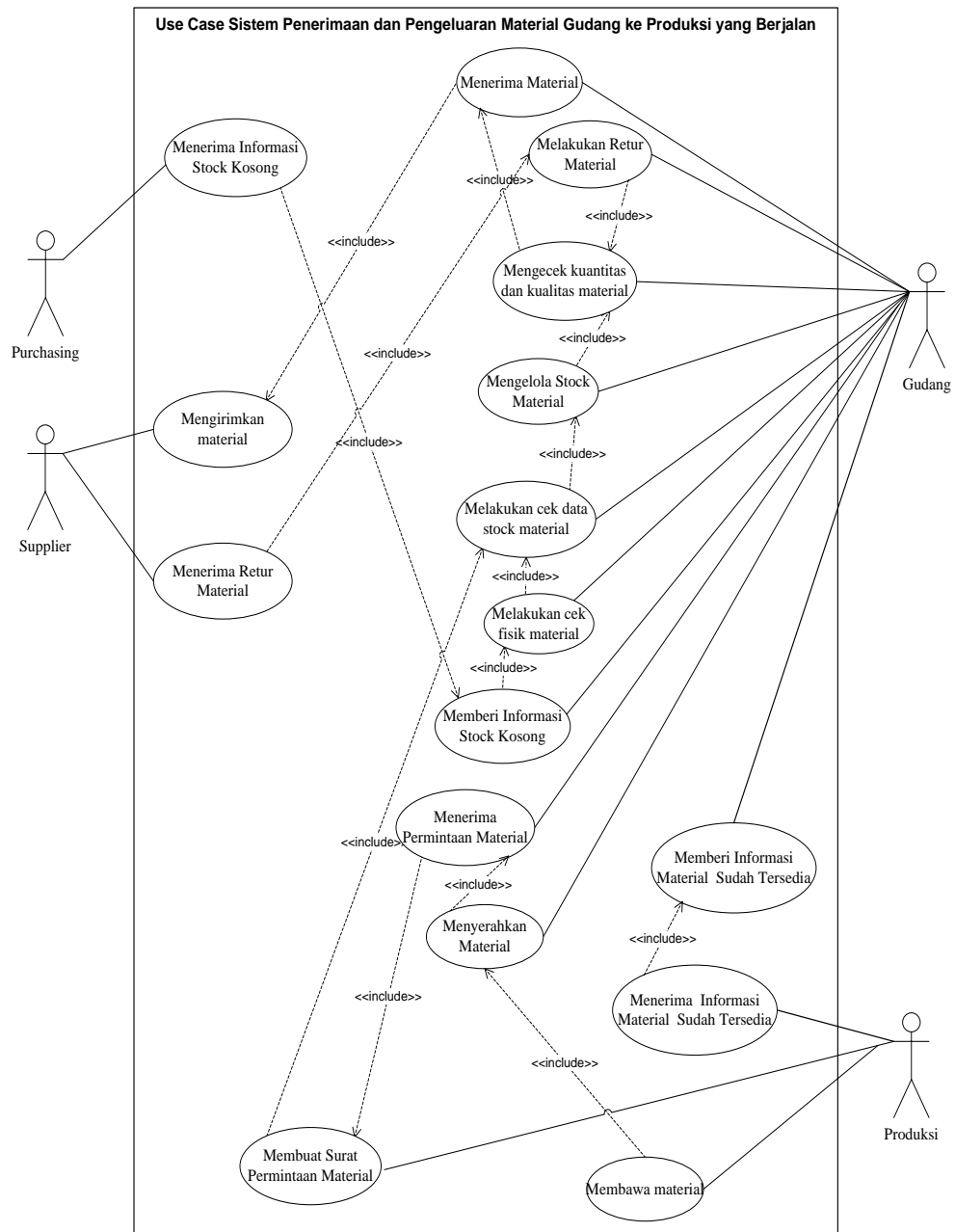
Gambar IV.6: *Form* Permintaan Material Poduksi ke gudang  
 (Sumber : PT Tetra Pak Stainless Equipment Jakarta, 2016)

*Form* atau surat Permintaan material ini adalah *form* rangkap dua yang dibutuhkan untuk proses pengeluaran material gudang ke produksi pada PT Tetra Pak Stainless Equipment Jakarta, tanpa adanya *form* ini proses pengeluaran material gudang ke produksi berdasarkan permintaan bagian produksi tidak dapat dilakukan. *Form* permintaan material ini diisi oleh bagian produksi untuk melakukan permintaan material ke bagian gudang untuk kebutuhan produksi. Surat ini akan di serahkan ke bagian gudang dan diberikan kepada karyawan gudang agar material yang diminta dapat dikeluarkan dan di bawa oleh bagian produksi sebagai dokumen untuk bagian produksi.

#### 4.5 Penggambaran Sistem Use Case yang Sedang Berjalan.

*Use case diagram* adalah deskripsi fungsi yang menggambarkan siapa saja aktor yang melakukan prosedur dalam sistem serta proses yang terlibat dalam transformasi pada sistem tersebut. Urutan langkah-langkah yang menerangkan antara pengguna dan system disebut skenario. Setiap skenario mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, system yang lain, perangkat keras atau urutan waktu. Penggambaran sistem penerimaan dan pengeluaran material gudang ke produksi pada PT Tetra Pak Stainless Equipment Jakarta dengan menggunakan *use case diagram* bertujuan untuk menggambarkan fungsionalitas dari sebuah sistem merepresentasikan sebuah interaksi antara aktor dengan sistem itu sendiri. Dengan menggunakan *use case diagram*, aktivitas yang dilakukan oleh sebuah aktor, dan keterkaitan antara satu aktor dengan aktor yang lain juga. Use case adalah alat bantu terbaik guna menstimulasi pengguna potensial untuk mengatakan tentang suatu system dari sudut pandangnya. Tidak selalu mudah bagi pengguna untuk menyatakan bagaimana mereka bermaksud untuk menggunakan sistem. Karena sistem pengembangan tradisional sering ceroboh dalam melakukan analisis, akibatnya pengguna seringkali susah menjawabnya tatkala dimintai masukan tentang sesuatu

Dalam pembicaraan tentang *use case*, pengguna biasanya disebut dengan actor. Aktor adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan system. Berikut adalah *use case diagram* sistem informasi penerimaan dan pengeluaran material gudang ke produksi pada PT Tetra Pak Stainless Equipment Jakarta yang sedang berjalan:



Gambar: IV.7 Use Case Sistem Penerimaan dan Pengeluaran Material Gudang ke Produksi yang Berjalan di PT Tetra Pak Stainless Equipment Jakarta  
(Sumber : PT Tetra Pak Stainless Equipment Jakarta, 2016)

Penjelasan *use case diagram* system penerimaan dan pengeluaran material gudang ke produksi pada PT Tetra Pak Stainless Equipment Jakarta di atas, dapat dijelaskan sebagai berikut:

## 1. Definisi Aktor

Pendefinisian aktor pada *use case* system penerimaan dan pengeluaran material gudang ke produksi pada PT Tetra pak Stainless equipment Jakarta. dapat dilihat pada tabel IV.1 berikut:

Tabel IV.1 Definisi Aktor *Use Case* Sistem yang berjalan

1.	Gudang	Bagian yang menerima material, melakukan cek kuantitas dan kualitas material, melakukan retur material, melakukan pengolahan data material, melakukan pengecekan stock material, melakukan pengecekan fisik material, menerima permintaan material, mengeluarkan material, menyerahkan material, dan sekaligus membuat laporan stok kosong.
2.	Produksi	Bagian yang melakukan permintaan material kepada bagian gudang untuk kebutuhan produksi. Serta membawa material dari bagian gudang ke produksi.
3.	<i>Purchasing</i>	Bagian yang menerima laporan stok kosong dari bagian gudang.
4.	<i>Supplier</i>	Penyedia material atau pihak yang menjual material dan mengirimkan material ke bagian gudang.

(Sumber: Hasil Pengolahan Data, 2016)

## 2. Definisi *Use Case*

Pendefinisian *use case* pada *use case diagram* sistem Sistem Pengeluaran bahan baku pada PT Tetra Pak Stainless Equipment Jakarta dapat dilihat pada tabel IV.2 berikut:

Tabel IV.2 Definisi *Use Case* Sistem yang Berjalan

No.	<i>Use Case</i>	Deskripsi
1.	Mengirimkan material	Bagian <i>Supplier</i> mengirimkan material kepada bagian gudang.
2.	Menerima material	Bagian gudang menerima material beserta <i>delivery order</i> dari <i>Supplier</i> .
3.	Cek kuantitas dan kualitas bahan baku	Bagian gudang mengecek kuantitas dan kualitas material yang baru diterima dari <i>supplier</i> .
4.	Melakukan retur material	Bagian gudang melakukan retur material jika material yang diterima tidak sesuai atau rusak.
5.	Menerima retur material	Bagian <i>Supplier</i> menerima retur material dari gudang karena material tidak sesuai atau ada yang rusak .
6.	Mengelola stok material	Bagian gudang mengelola stok material dengan menginput dan memperbaharui stok material.
7.	Memberi Informasi Bahwa material sudah tersedia.	Bagian gudang memberi informasi ke bagian produksi bahwa material sudah tersedia dan siap untuk dilakukan proses permintaan material.
8.	Menerima Informasi Material sudah tersedia	Bagian produksi menerima informasi dari bagian gudang bahwa material sudah tersedia.
9.	Membuat surat permintaan material	Bagian produksi membuat surat permintaan material untuk mendapatkan material yang diinginkan.
10.	Menerima permintaan material	Bagian gudang menerima permintaan material dari bagian produksi berdasarkan surat permintaan material.

Tabel IV.2 Definisi *Use Case* Sistem yang Berjalan (lanjutan)

No	<i>Use Case</i>	<b>Deskripsi</b>
11.	Melakukan cek data stock material	Bagian gudang melakukan pengecekan data stock material setelah menerima permintaan material dari bagian produksi.
12	Mengecek fisik material	Bagian gudang akan melakukan cek fisik material setelah melakukan cek data stock material.
13.	Menyerahkan material	Bagian gudang melakukan penyerahan material kepada orang produksi yang meminta material.
14.	Membawa material	Bagian produksi membawa material yang diserahkan oleh bagian gudang.
15.	Membuat laporan stok kosong	Bagian gudang membuat laporan ini setelah melakukan pengecekan stok dan ternyata stok habis sehingga perlunya pembelian kembali.
16.	Menerima laporan stok kosong	Bagian purchasing menerima laporan stok kosong dari bagian gudang.

(Sumber: Hasil Pengolahan Data, 2016)

## BAB V

### ANALISIS DAN PEMBAHASAN

#### 5.1 Analisis Kebutuhan Sistem

Berdasarkan hasil pengolahan data, dibutuhkan perancangan aplikasi untuk menunjang kegiatan Penerimaan dan Pengeluaran material gudang ke produksi pada PT Tetra Pak Stainless Equipment Jakarta. Perancangan aplikasi yang memudahkan *user* dalam menggunakan aplikasi ini.

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 program aplikasi, dimana proses-proses tersebut dibutuhkan oleh pemakai. Berikut adalah penjelasan identifikasi kebutuhan pemakai aplikasi penerimaan dan pengeluaran material gudang ke produksi:

Tabel V.1 Identifikasi Kebutuhan Sistem

<b>Kebutuhan Sistem</b>	
Nama Proyek	Rancang bangun aplikasi penerimaan dan pengeluaran material gudang ke produksi.
Proses Sistem Berjalan	1. Terwujudnya sebuah sistem yang dapat membantu proses penerimaan dan pengeluaran material gudang ke produksi.
Data Masukan ( <i>input</i> )	1. Data Material. 2. Data <i>Supplier</i> . 3. Data Receipt Material. 4. Data Request Material.
Data Keluaran ( <i>output</i> )	1. Data Output Report Material. 2. Data Empty Stock Material. 3. Data Retur Material

(Sumber: Hasil Analisis Data, 2016)

Tabel V.1 Identifikasi Kebutuhan Sistem (Lanjutan)

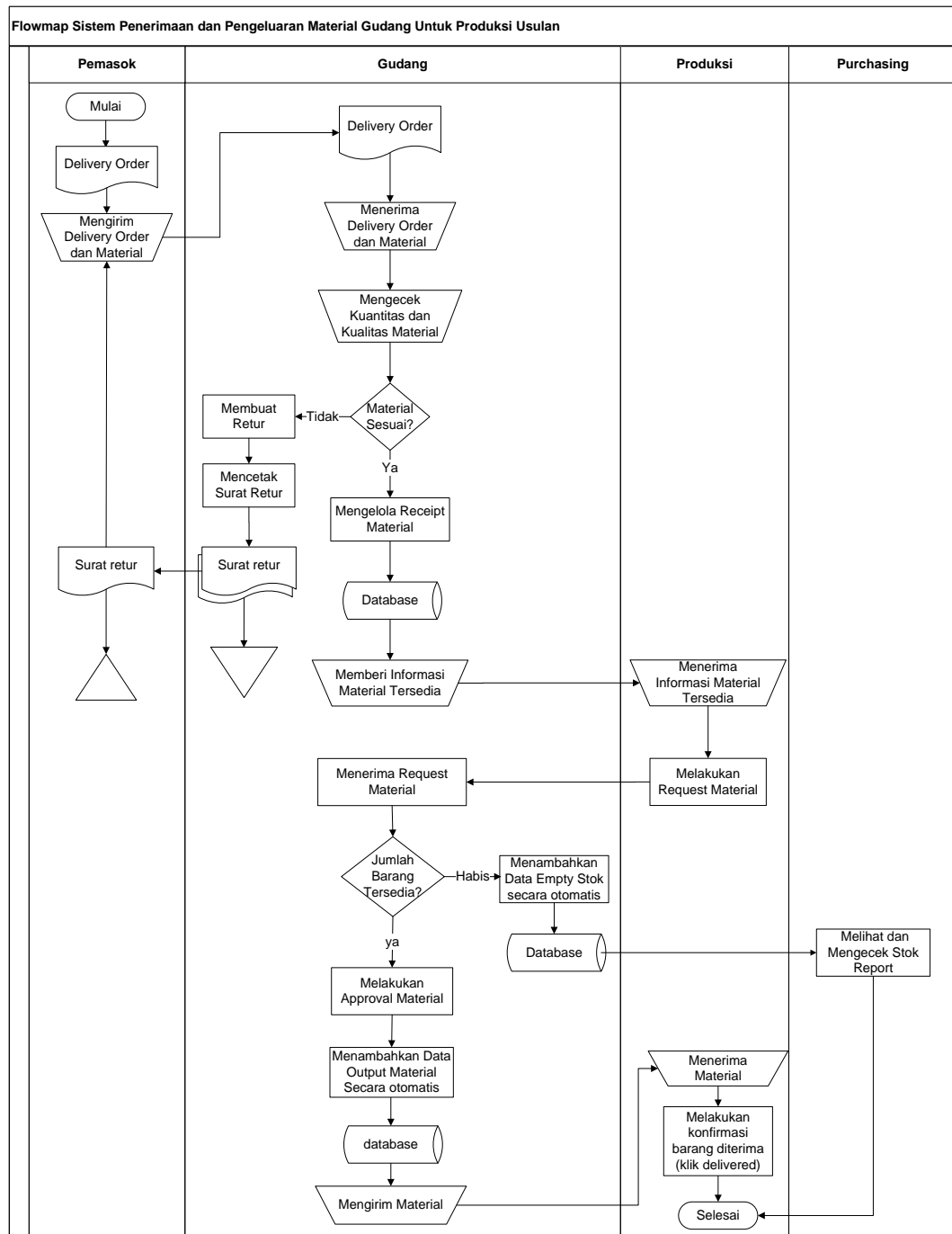
<b>Kebutuhan Sistem</b>	
Model Sistem Informasi	1. <i>Input</i> : <i>Receipt Material</i> . Proses : Pengolahan data <i>Output</i> : <i>Output Report Material</i> .

(Sumber: Hasil Analisis Data, 2016)

## 5.2 Perancangan *Flowmap* Sistem Usulan

Proses penerimaan dan pengeluaran material gudang ke produksi yang diusulkan dengan bagian produksi yang melakukan permintaan material ke bagian gudang. Permintaan Material ke bagian gudang dilakukan oleh bagian produksi dengan memesan Material kebutuhan produksi tersebut melalui sistem komputer dimana pihak gudang akan menerima permintaan dan mengolah dan memproses permintaan tersebut serta mengantarkannya dan dengan hasil *output report* material dan konfirmasi material oleh bagian produksi.

Seperti yang dapat dilihat pada gambar V.1 adalah *flowmap* proses penerimaan dan pengeluaran material gudang ke produksi yang diusulan sebagai berikut:



Gambar V.1 *Flowmap* Proses Penerimaan dan Pengeluaran Material Gudang ke Produksi yang di Usulkan Pada PT Tetra Pak Stainless Equipment Jakarta (Sumber: Hasil Analisis Data, 2016)

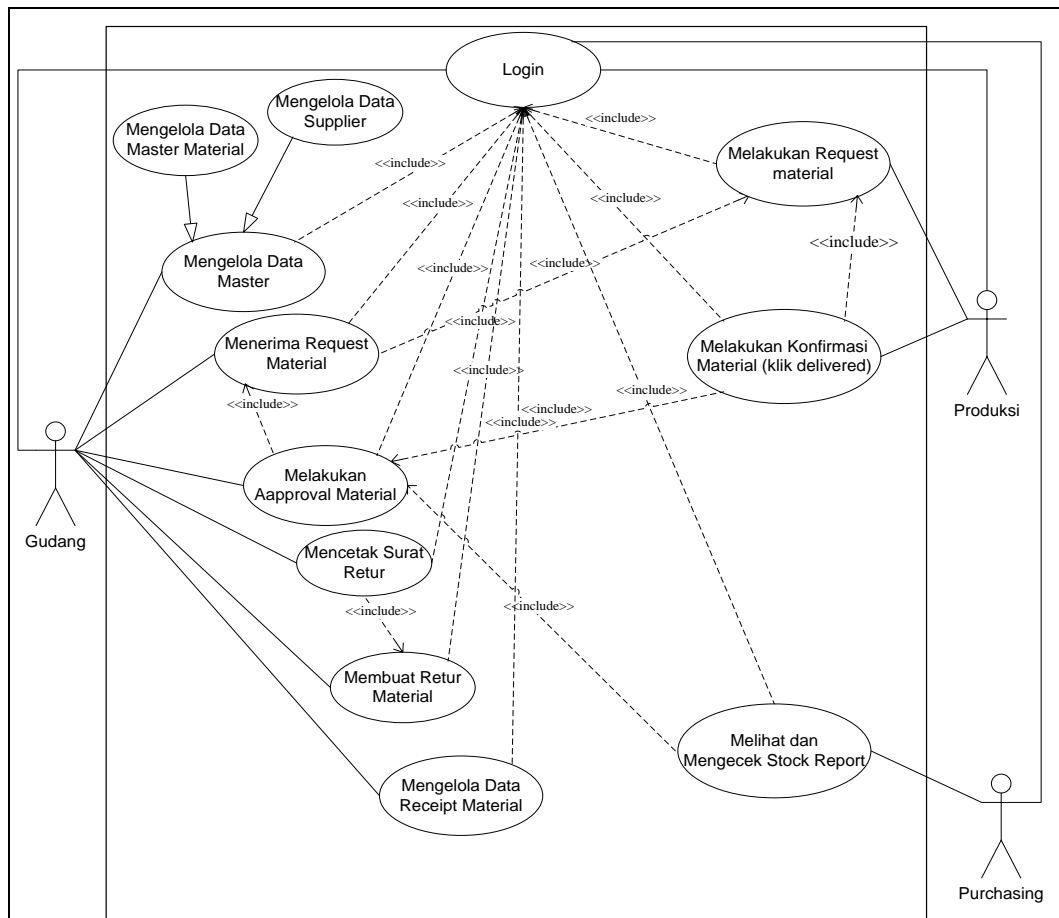
### 5.3 Analisis Sistem Usulan

Perancangan aplikasi yang diusulkan menggunakan model pengembangan perangkat lunak *prototype evolutioner*. Tahap ini adalah tahap kedua pada

metodologi *prototipe evolusioner* yaitu membuat *prototipe* untuk model sistem setelah sebelumnya melakukan tahap mengidentifikasi kebutuhan. Analisis proses sistem informasi penerimaan dan pengeluaran material gudang ke produksi menggunakan *tools* pemodelan sistem UML (*unified modeling language*) yang meliputi pembuatan *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram*, *deployment diagram* dan juga pembuatan kamus data. Tahapan analisis ini akan memberikan gambaran mengenai aliran informasi dan data pada sistem informasi yang akan dibangun. Selanjutnya akan dilakukan perancangan sistem informasi penerimaan dan pengeluaran material gudang ke produksi berdasarkan hasil analisis.

### **5.3.1 Use Case Diagram Usulan**

*Use case diagram* digunakan untuk menjelaskan interaksi antara *actor* dengan sistem informasi yang akan dibuat. Berdasarkan analisis yang telah dilakukan, maka *use case diagram* sistem informasi penerimaan dan pengeluaran material gudang ke produksi yang diusulkan dapat dilihat pada gambar V.2 sebagai berikut ini:



Gambar V.2 Use Case Diagram Proses Penerimaan dan Pengeluaran Material Gudang ke Produksi yang di Usulkan Pada PT Tetr Pak Stainless Equipment Jakarta.  
(Sumber: Hasil Analisis Data, 2016)

Penjelasan *use case diagram* penerimaan dan pengeluaran material gudang ke produksi usulan di atas, dapat dijelaskan sebagai berikut:

#### 1. Definisi Aktor

Pendefinisian aktor pada *use case* perancangan sistem informasi penerimaan dan pengeluaran material gudang ke produksi usulan dapat dilihat pada tabel V.2 berikut:

Tabel V.2 Definisi Aktor *Use Case Diagram* Sistem Usulan

Aktor	Deskripsi
Gudang	Bagian yang menerima dan mengeluarkan material, mengelola data master, mengelola receipt material, menerima request material dari bagian produksi, membuat retur, mencetak retur serta yang melakukan approval material.
Produksi	Bagian yang melakukan permintaan material untuk kebutuhan produksi.
<i>Purchasing</i>	Bagian yang melihat laporan <i>stock</i> kosong.

(Sumber: Hasil Analisis Data, 2016)

2. Definisi dari *use case diagram* usulan Pengeluaran material gudang ke produksi adalah sebagai berikut:

Tabel V.3 Definisi Aktor *Use Case Diagram Login User*

Nama <i>Use Case</i>	Mengelola Data <i>Login User</i>
Deskripsi <i>Use Case</i>	Sistem ini memungkinkan aktor mengakses sistem informasi penerimaan dan pengeluaran material gudang untuk produksi.
Aktor	Staf Gudang, Staf Produksi, Staf <i>Purchasing</i> , Staf <i>Supplier</i>
<i>Normal Flow Events:</i>	1. Memasukan <i>username</i> 2. Memasukan <i>password</i> 3. Memasukan Identifikasi

(Sumber: Hasil Analisis Data, 2016)

Tabel V.4 *Use Case Description* Mengelola Data Master *Supplier*

Nama <i>Use Case</i>	Mengelola Data master <i>Supplier</i>
Deskripsi <i>Use Case</i>	1. <i>Use case</i> ini menggambarkan proses pengolahan <i>supplier</i>

Tabel V.4 *Use Case Description* Mengelola Data Master *Supplier* (Lanjutan)

Aktor	Staf Gudang
<i>Normal Flow Events:</i>	<ol style="list-style-type: none"> <li>2. Bagian Gudang masuk ke tampilan menu utama sistem.</li> <li>3. Bagian Gudang memilih menu Add Supplier.</li> <li>4. Aplikasi menampilkan form add supplier.</li> <li>5. Bagian gudang mengisi semua data yang ada pada form yang ditampilkan oleh sistem dan kemudian submit maka data supplier akan bertambah pada form supplier list.</li> </ol>

(Sumber: Hasil Analisis Data, 2016)

Tabel V.5 *Use Case Description* Mengelola Data Master Material.

Nama <i>Use Case</i>	Mengelola Data Master Material
Deskripsi <i>Use Case</i>	<i>Use case</i> ini menggambarkan proses pengolahan data master material.
Aktor	Staf Gudang
<i>Normal Flow Events</i>	<ol style="list-style-type: none"> <li>1. Staf Gudang masuk ke tampilan menu utama sistem.</li> <li>2. Staf Gudang memilih menu <i>warehouse dan master material</i>.</li> <li>3. Aplikasi menampilkan <i>form master material</i>.</li> <li>4. Pilih <i>add new</i> untuk menambahkan data material.</li> <li>5. Pilih <i>delete</i> untuk menghapus data material.</li> <li>6. Pilih <i>edit</i> untuk mengubah data material.</li> </ol>

(Sumber: Hasil Analisis Data, 2016)

Tabel V.6 Definisi Aktor *Use Case* Mengelola *Receipt* Material

Nama <i>Use Case</i>	Mengelola <i>Receipt</i> Material
Deskripsi <i>Use Case</i>	Sistem ini menggambarkan proses dari pengolahan <i>receipt</i> material.
Aktor	Staf Gudang

Tabel V.6 Definisi Aktor *Use Case* Mengelola *Receipt Material* (Lanjutan)

<p><i>Normal Flow</i></p> <p><i>Events:</i></p>	<ol style="list-style-type: none"> <li>1. Staf Gudang masuk ke tampilan menu utama sistem.</li> <li>2. Staf gudang memilih <i>form receipt</i> material</li> <li>3. System menampilkan <i>form receipt</i> material.</li> <li>4. Pilih <i>add new</i> untuk menambahkan data <i>receipt</i> material dan data stok bertambah secara otomatis oleh sistem.</li> <li>5. Pilih <i>delete</i> untuk menghapus data <i>receipt</i> material.</li> <li>6. Pilih <i>edit</i> untuk mengubah data <i>receipt</i> material.</li> <li>7. Pilih <i>detail</i> untuk menampilkan data detail.</li> <li>8. Pilih <i>print</i> untuk mencetak retur material.</li> </ol>
---	--

Tabel V.7 *Use Case Description* Melakukan *Approval Material*.

Nama <i>Use Case</i>	Melakukan <i>Approvals</i> Material
Deskripsi <i>Use Case</i>	<i>Use case</i> ini menggambarkan proses <i>approval</i> material.
Aktor	Staf Gudang
<i>Normal Flow Events:</i>	<ol style="list-style-type: none"> <li>1. Staf gudang masuk ke tampilan menu utama sistem.</li> <li>2. Staf gudang memilih menu <i>request approval</i> material .</li> <li>3. Sistem menampilkan <i>form request approvals</i>.</li> <li>4. Staf gudang mengisi data yang sesuai berdasarkan request dari bagian produksi</li> <li>5. Jika material tersedia maka data <i>output report</i> material bertambah dan stok berkurang secara otomatis oleh sistem.</li> <li>6. Jika material habis maka data <i>empty stock</i> bertambah secara otomatis oleh sistem.</li> </ol>

(Sumber: Hasil Analisis Data, 2016)

Tabel V.8 *Use Case Description* Membuat Retur material

Nama <i>Use Case</i>	Membuat Retur material
Deskripsi <i>Use Case</i>	<i>Use case</i> ini menggambarkan proses retur material.
Aktor	Staf Gudang
<i>Normal Flow Events:</i>	<ol style="list-style-type: none"> <li>1. Staf gudang masuk ke tampilan menu utama sistem.</li> <li>2. Staf gudang memilih form retur material.</li> <li>3. Sistem menampilkan form retur material.</li> <li>4. Pilih add new untuk menambahkan retur material.</li> </ol>

(Sumber: Hasil Analisis Data, 2016)

Tabel V.9 *Use Case Description* Mencetak Surat Retur

Nama <i>Use Case</i>	Mencetak Surat Retur
Deskripsi <i>Use Case</i>	<i>Use case</i> ini menggambarkan proses mencetak surat retur.
Aktor	Staf Gudang
<i>Normal Flow Events:</i>	<ol style="list-style-type: none"> <li>1. Staf gudang masuk ke tampilan menu utama</li> <li>2. Pilih <i>submenu</i> retur material.</li> <li>3. Sistem menampilkan data retur material</li> <li>4. Staf gudang pilih detail dan kemudian klik print.</li> </ol>

(Sumber: Hasil Analisis Data, 2016)

Tabel V.10 *Use Case Description* Melakukan *Request* Material.

Nama <i>Use Case</i>	Melakukan <i>Request</i> Material.
Deskripsi <i>Use Case</i>	<i>Use case</i> ini menggambarkan proses <i>Request</i> Material.
Aktor	Staf Produksi

Tabel V.10 *Use Case Description* Melakukan *Request Material* (Lanjutan)

<i>Normal Flow Events:</i>	<ol style="list-style-type: none"> <li>1. Staf produksi masuk ke tampilan menu utama sistem.</li> <li>2. Staf produksi memilih form <i>request material</i>.</li> <li>3. Sistem menampilkan form <i>request material</i>.</li> <li>4. Pilih add new untuk melakukan dan menambahkan proses <i>request material</i>.</li> <li>5. Pilih detail untuk data <i>request material</i> secara detail.</li> </ol>
----------------------------	---

(Sumber: Hasil Analisis Data, 2016)

Tabel V.11 *Use Case Description* Menerima *Material*

Deskripsi Use Case	Menerima Material
Aktor	Staf Produksi
<i>Normal Flow Event:</i>	<ol style="list-style-type: none"> <li>1. Staf produksi masuk ke tampilan menu utama.</li> <li>2. Staf produksi memilih <i>menu request material</i>.</li> <li>3. Staf produksi memilih tombol <i>delivered</i></li> </ol>

(Sumber: Hasil Analisis Data, 2016)

Tabel V.12 *Use Case Description* Melihat Dan Mengecek *Stock Report*

Nama Use Case	Melihat Dan Mengecek <i>Stock Report</i>
Deskripsi Use Case	<i>Use case</i> ini menggambarkan proses mengecek laporan <i>stock</i> kosong.
Aktor	Staf <i>Purchasing</i>
<i>Normal Flow Events:</i>	<ol style="list-style-type: none"> <li>1. Staf <i>Purchasing</i> masuk ke tampilan menu utama sistem.</li> <li>2. Staf <i>Purchasing</i> memilih menu <i>stock report</i>.</li> <li>3. Staf <i>Purchasing</i> mengecek laporan yang sudah ditampilkan oleh aplikasi.</li> </ol>

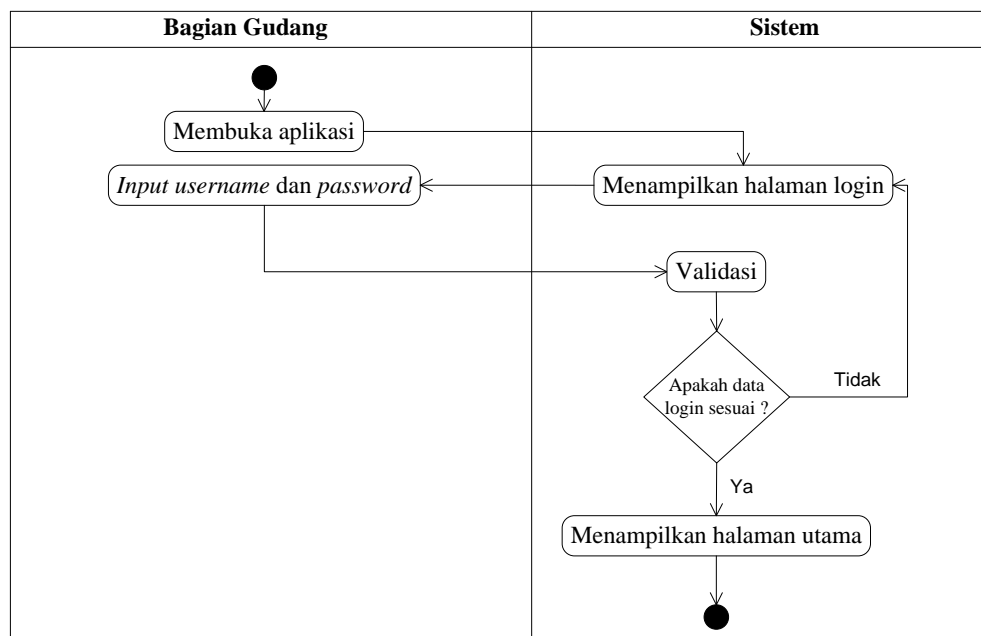
(Sumber: Hasil Analisis Data, 2016)

### 5.3.2 Activity Diagram

*Activity diagram* digunakan untuk menggambarkan aliran kerja tiap *use case* pada sistem informasi proses pengeluaran *material*. Berikut adalah *activity diagram* tiap *use case*:

#### 1. Activity Diagram Proses Login

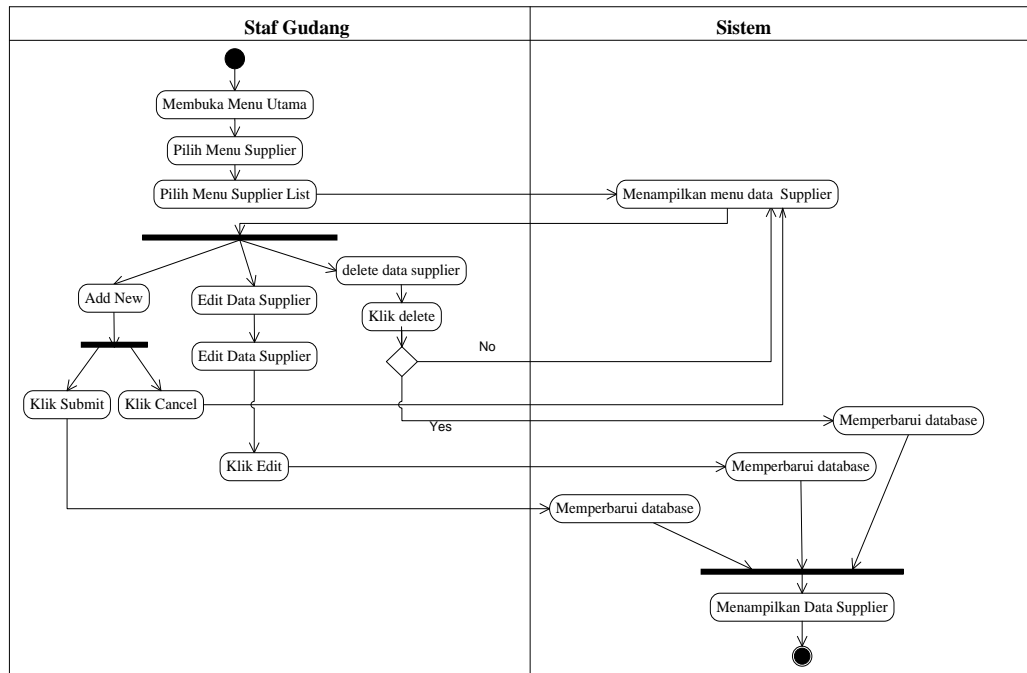
*Activity diagram* berikut ini menjelaskan aktifitas yang dilakukan ketika melakukan proses *login*. Dimana *user* memasukkan *username* dan *password* untuk dapat masuk ke dalam sistem. Jika *username* dan *password* tidak sesuai maka tidak dapat masuk ke dalam sistem. Berikut adalah gambar V.3 *activity diagram* proses *login*:



Gambar V.3 Activity Diagram Proses Login  
(Sumber: Hasil Analisis Data, 2016)

#### 2. Activity Diagram Mengelola Data Master Supplier.

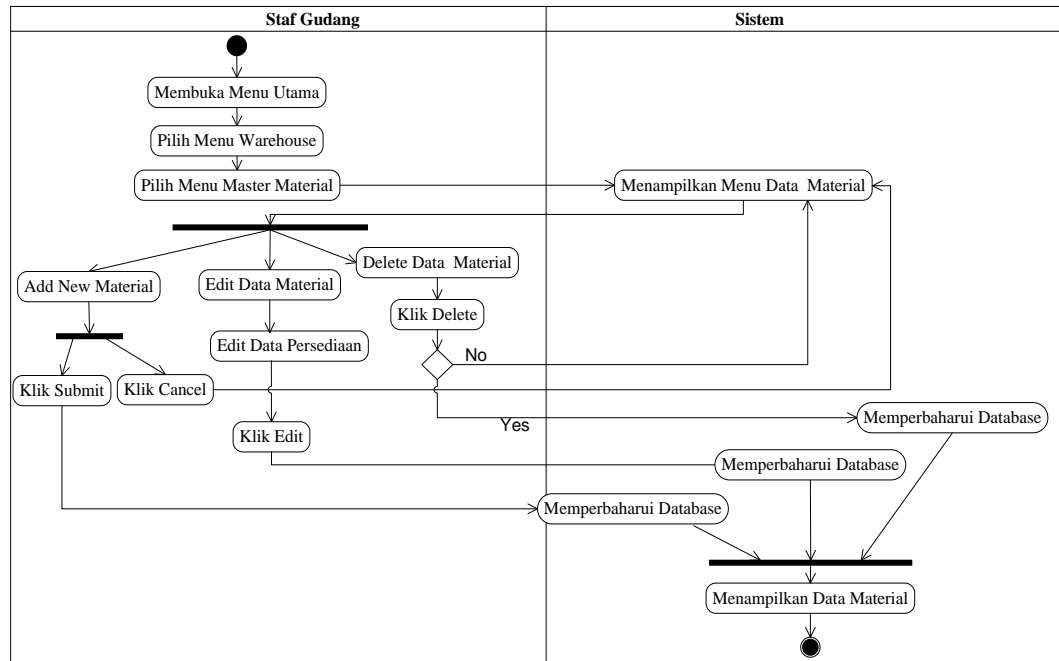
*Activity diagram* mengelola data *master supplier* menjelaskan alur aktifitas yang dilakukan ketika memilih menu *warehouse* dengan *submenu add supplier* untuk menambahkan data *supplier*, menghapus serta mengubah data *supplier*. Berikut adalah gambar V.4 *activity diagram* Mengelola data *master supplier* :



Gambar V.4 *Activity Diagram* Mengelola Data Master Supplier  
(Sumber: Hasil Analisis Data, 2016)

### 3. *Activity Diagram* Mengelola Data Master Material

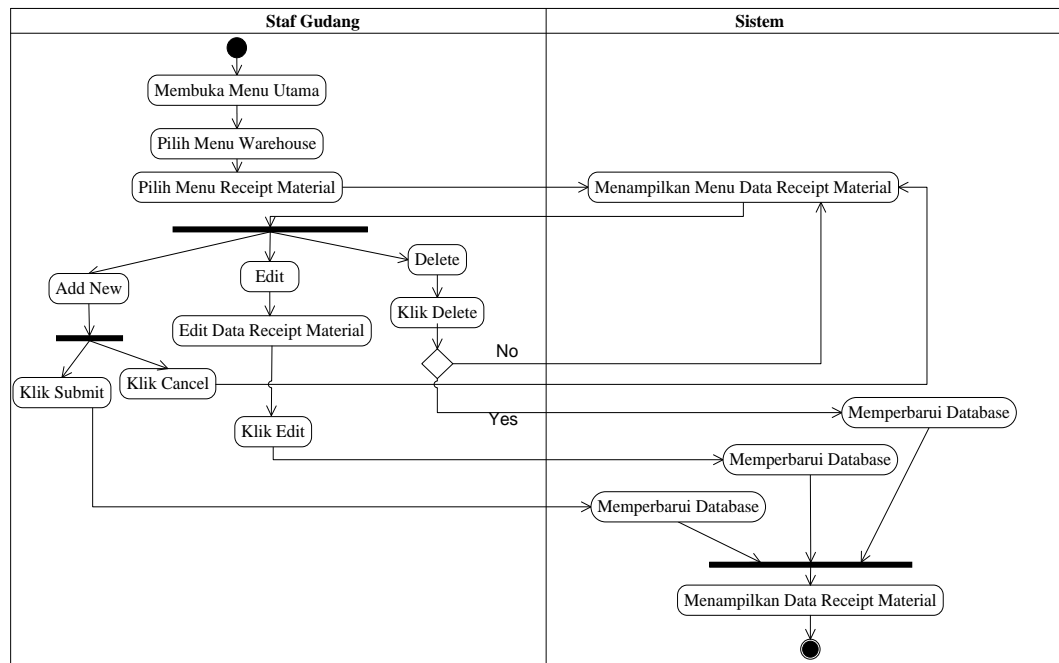
*Activity diagram* berikut ini menjelaskan aktifitas yang dilakukan ketika memilih menu *warehouse* untuk melakukan pengelolaan data *master* material. Data *master* material ditambahkan masuk ke *database* yang dikelola oleh setiap staf gudang. Sistem menampilkan hasil *input* data material pada gudang. Berikut adalah gambar V.5 *activity diagram* mengelola data *master* material:



Gambar V.5 Activity Diagram Mengelola Data Master Material  
(Sumber: Hasil Analisis Data, 2016)

#### 4. Activity Diagram Mengelola Data Receipt Material

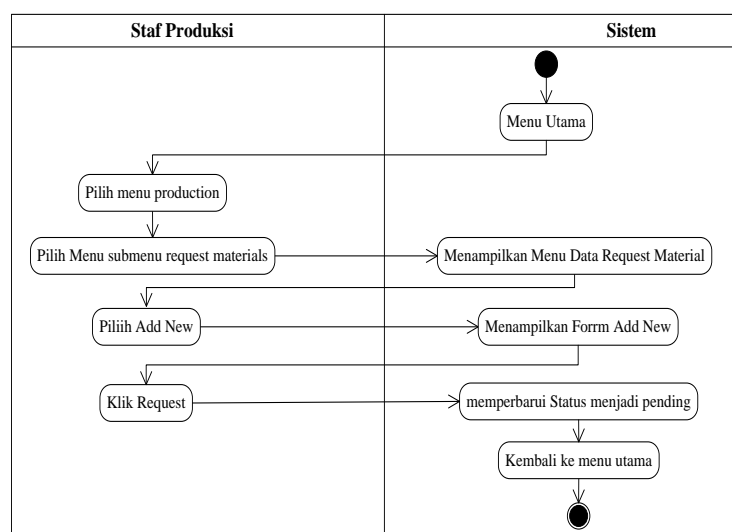
Activity Diagram berikut ini menjelaskan tentang aktifitas yang dilakukan ketika memilih menu *warehouse* untuk melakukan pengolahan data *receipt* material. Proses ini terjadi ketika adanya penerimaan material dari *supplier*. Berikut adalah gambar V.6 activity diagram proses mengelola *receipt* material:



Gambar V.6 Activity Diagram Mengelola Data Receipt Material  
(Sumber: Hasil Analisis Data, 2016)

## 5. Activity Diagram Melakukan Request Material

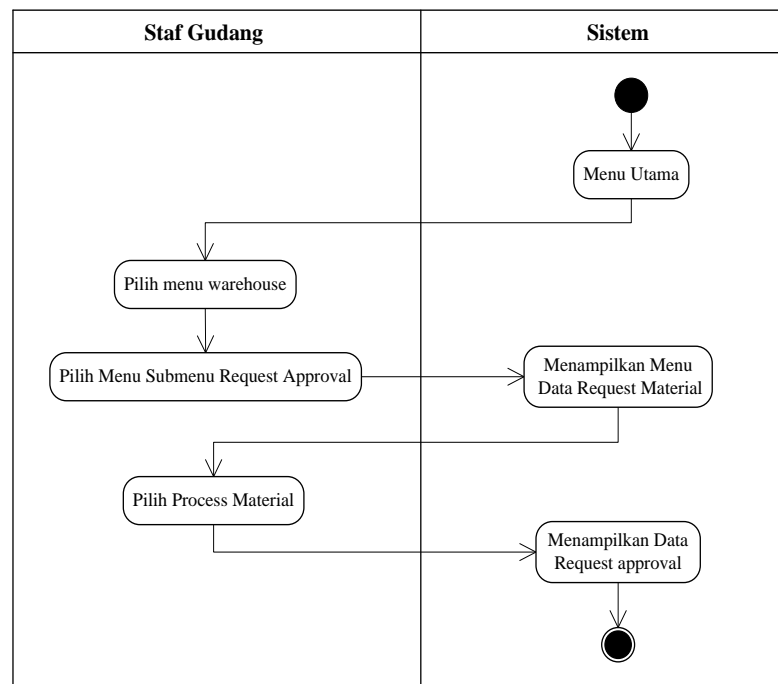
Activity diagram berikut ini menjelaskan aktifitas yang dilakukan ketika memilih menu *Production* untuk melakukan *request* material. *request* material dibuat oleh staf produksi untuk kebutuhan produksi. Sistem menampilkan hasil dari melakukan *request* material. Berikut adalah gambar V.7 activity diagram proses melakukan *request* material:



Gambar V.7 Activity Diagram Melakukan Request Material  
(Sumber: Hasil Analisis Data, 2016)

#### 6. Activity Diagram Menerima Request Material.

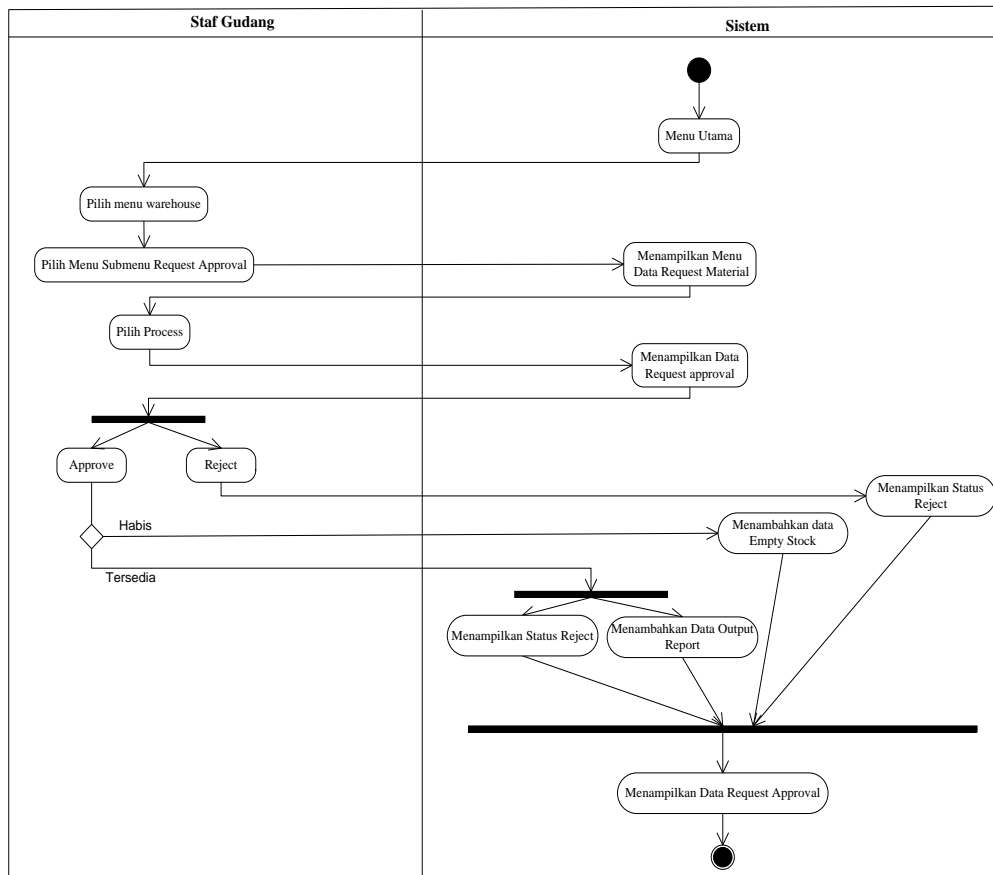
Activity diagram berikut ini menjelaskan aktifitas yang dilakukan ketika memilih menu Gudang untuk menerima *request* material dari bagian produksi. Sistem menampilkan data dari request material. Berikut adalah gambar V.8 activity diagram proses menerima request material:



Gambar V.8 Activity Diagram Menerima Request Material  
(Sumber: Hasil Analisis Data, 2016)

#### 7. Activity Diagram Melakukan Approval Material

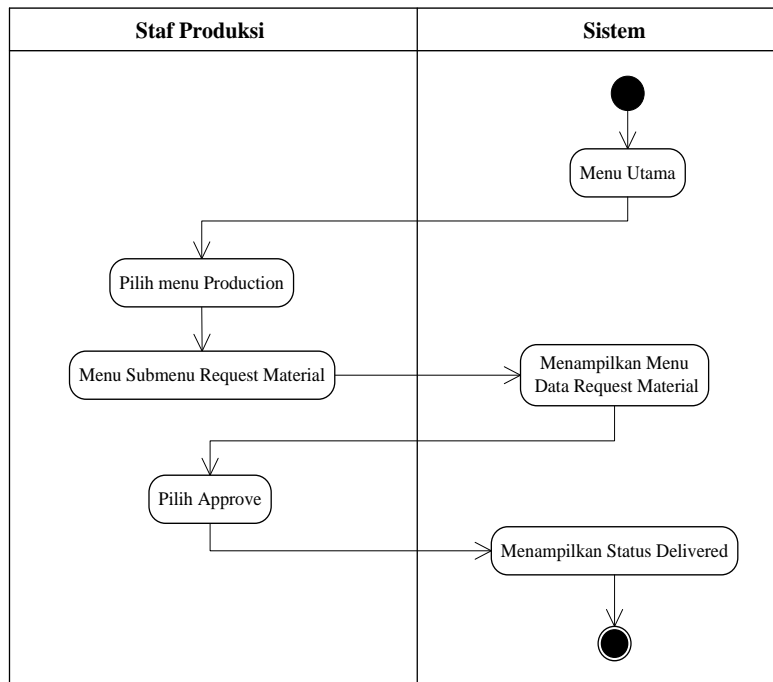
Activity diagram melakukan *approval* material ini menjelaskan aktifitas yang dilakukan ketika memilih menu *warehouse* untuk proses lanjutan dari proses penerimaan *request* material. Berikut adalah gambar V.9 activity diagram melakukan *approval* material:



Gambar V.9 *Activity Diagram Approval Material*  
(Sumber: Hasil Analisis Data, 2016)

#### 8. *Activity Diagram* Melakukan Konfirmasi Material

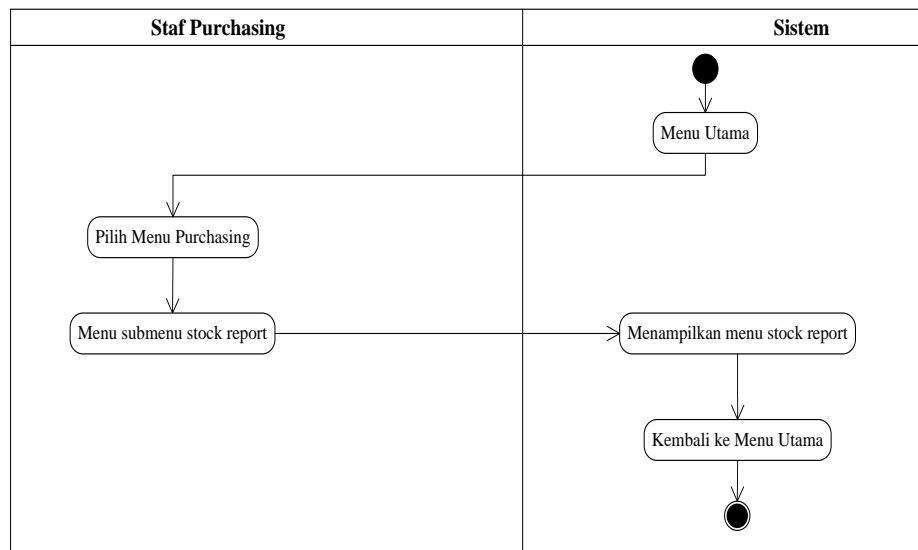
*Activity Diagram* berikut ini menjelaskan aktifitas yang dilakukan ketika bagian produksi telah menerima material dari bagian gudang. Aktifitas ini dilakukan untuk kebutuhan dari proses pengeluaran material gudang ke produksi bahwa material yang telah keluar mendapat tanda atau berstatus sudah diterima oleh bagian produksi. Sistem menampilkan hasil Konfirmasi material. Berikut adalah gambar V.10 *activity diagram* melakukan konfirmasi material:



Gambar V.10 *Activity Digram* Melakukan Konfirmasi Material  
(Sumber: Hasil Analisis Data, 2016)

### 9. *Activity Diagram* Melihat Dan Mengecek *Stock Report*

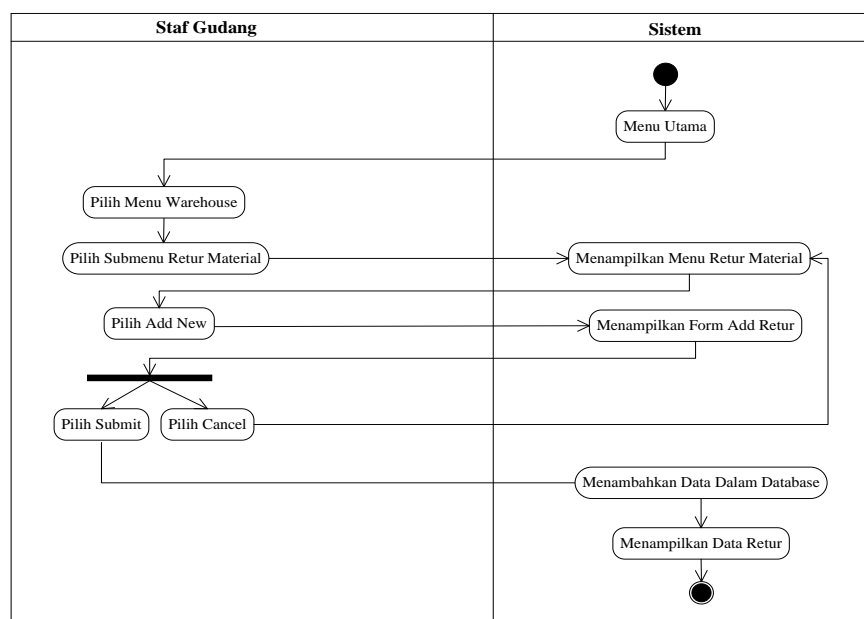
*Activity diagram* berikut ini menjelaskan aktifitas yang dilakukan ketika memilih menu *Purchasing* untuk melihat dan mengecek *Stock Report*. Pengecekan *stock report* dilakukan oleh staf *purchasing*. Sistem menampilkan hasil *Stock Report*. Berikut adalah gambar V.11 *activity diagram* proses mengecek *stock report*:



Gambar V.11 *Activity Diagram* Melihat Dan Mengecek *stock report*  
(Sumber: Hasil Analisis Data, 2016)

#### 10. *Activity Diagram* Membuat Retur Material

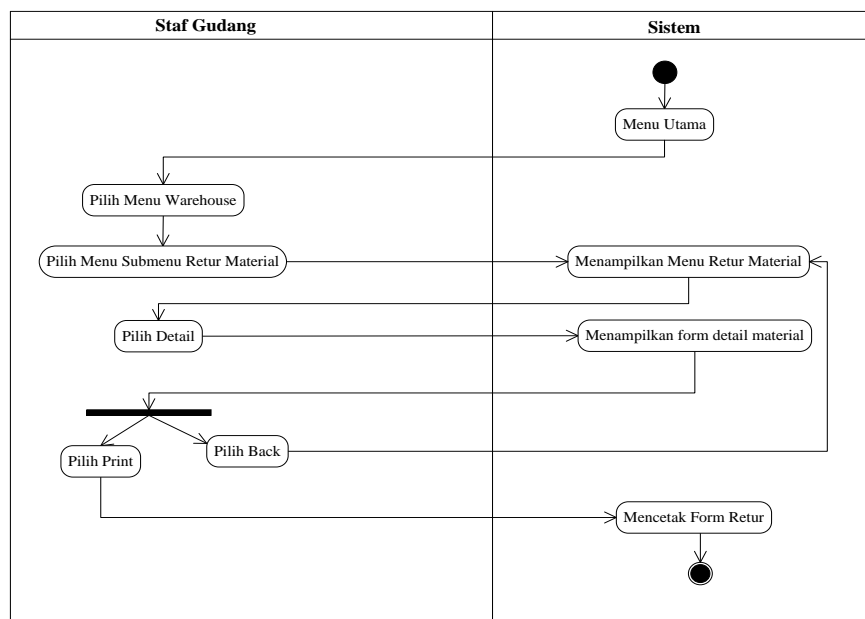
*Activity diagram* berikut ini menjelaskan aktifitas yang dilakukan ketika memilih menu *warehouse* kemudian pilih retur material untuk meretur material ketika ada penerimaan material yang tidak sesuai dari *supplier*. Retur material ini dilakukan oleh bagian gudang. Berikut adalah gambar V.12 *activity diagram* Melakukan retur:



Gambar V.12 *Activity Diagram* Melakukan Retur Material  
(Sumber: Hasil Analisis Data, 2016)

### 11. Activity Diagram Mencetak Surat Retur

Activity diagram berikut ini menjelaskan aktifitas yang dilakukan ketika ingin mencetak surat retur yaitu dengan memilih menu *warehouse* kemudian submenu retur material pilih detail untuk menampilkan form yang ingin di cetak. retur material ini dilakukan oleh bagian gudang. Berikut adalah gambar V.13 activity diagram mencetak retur.



Gambar V.13 Activity Diagram Mencetak Retur Material

(Sumber: Hasil Analisis Data, 2016)

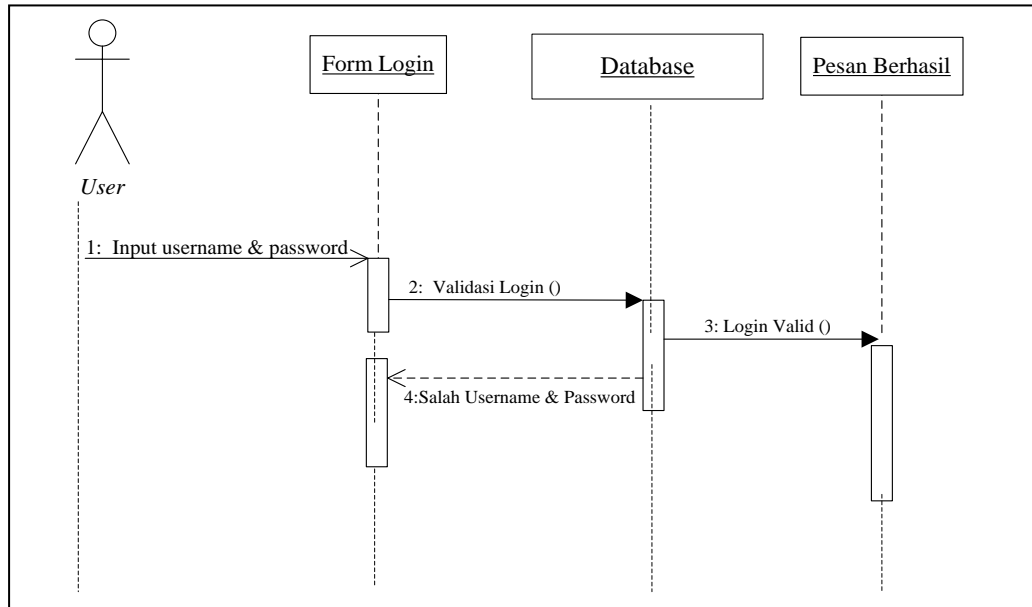
### 5.3.3 Sequence Diagram

Sequence diagram menggambarkan interaksi yang terjadi pada suatu objek *use case diagram* ketika melakukan suatu proses tertentu, di mana urutan proses ketika melakukan suatu proses tertentu dapat diketahui dengan melihat gambaran pada diagram. Hubungan yang ada pada gambar di bawah ini adalah proses yang dilakukan oleh sistem ketika melakukan proses yang sesuai dengan suatu objek *use case diagram*, berikut adalah *sequence diagram* pada proses pengeluaran material.

#### 1. Sequence Diagram pada Use Case Login User

Sequence diagram login user menjelaskan sebuah *sequence diagram* dalam proses login. Proses ini dilakukan oleh user sebelum masuk ke sistem.

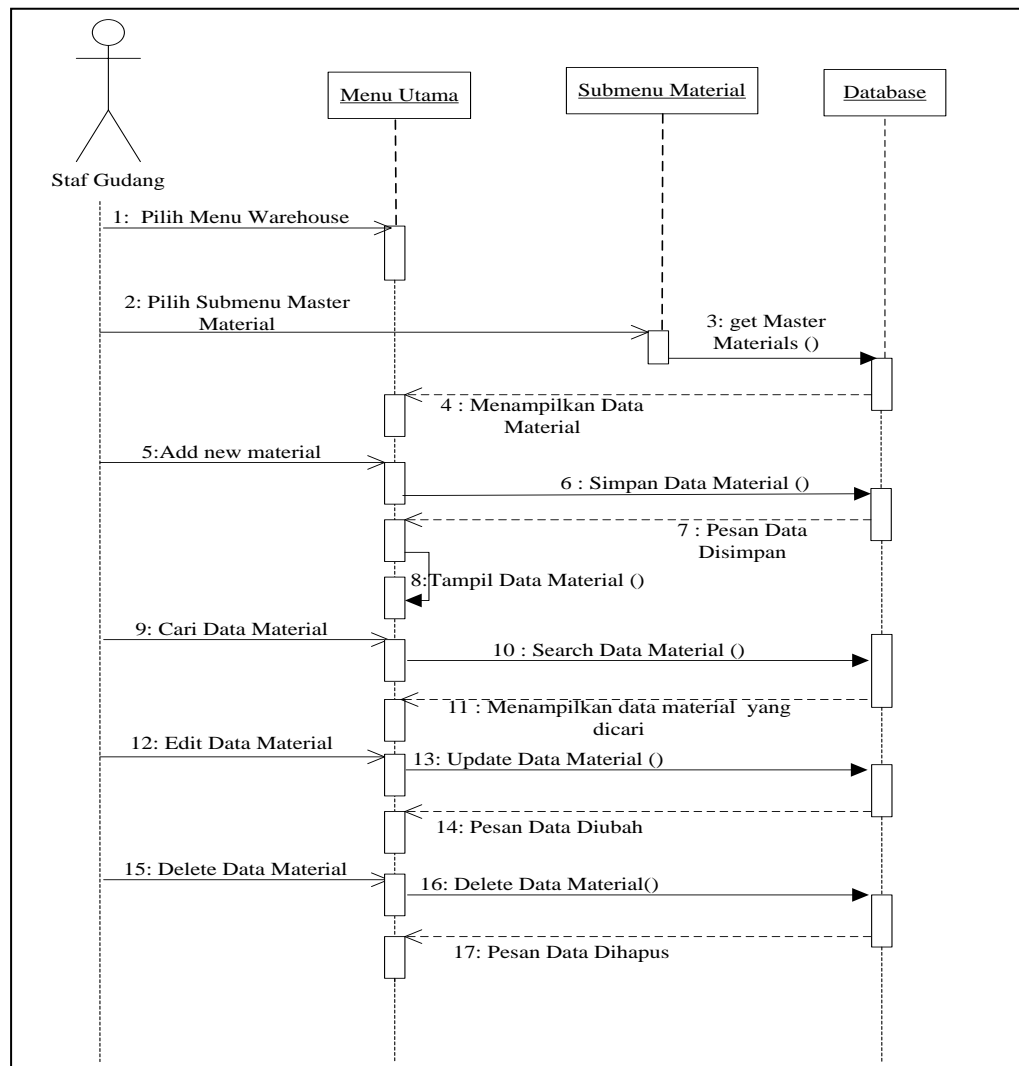
Adapun *sequence diagram* dari *use case login* dapat dilihat pada gambar V.14 di bawah ini.



Gambar V.14 *Sequence Diagram Login User*  
(Sumber: Hasil Analisis Data, 2016)

## 2. *Sequence Diagram* pada *Use Case* Mengelola Data Master Material

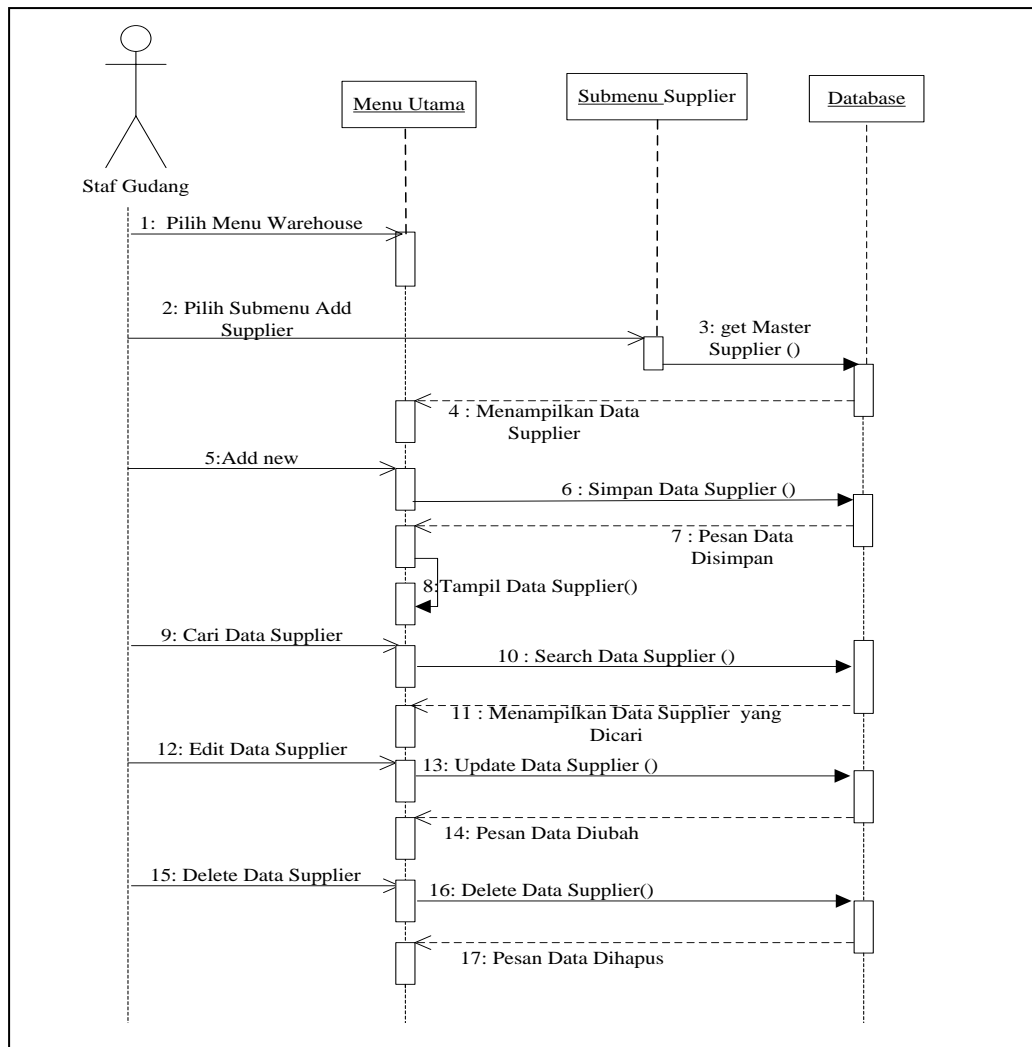
*Sequence diagram* mengelola data *master material* menjelaskan sebuah *sequence diagram* dalam proses *add new*, *search*, *edit* dan *delete* data *master material* yang dilakukan oleh staf gudang. Adapun *sequence diagram* dari *use case* mengelola data *master material* dapat dilihat pada gambar V.15 sebagai berikut



Gambar V.15 Sequence Diagram Mengelola Data Master Material  
(Sumber: Hasil Analisis Data, 2016)

### 3. Sequence Diagram pada Use Case Mengelola Data Master Supplier

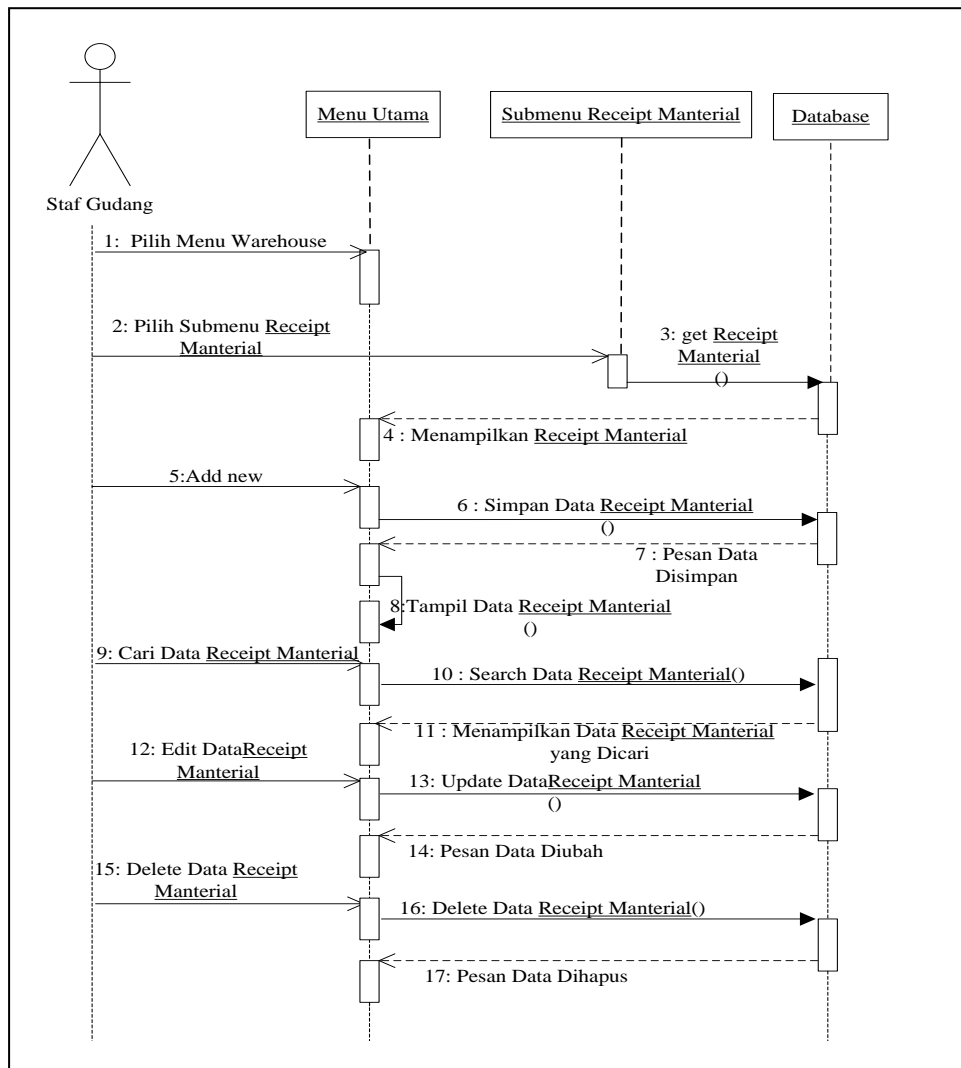
Sequence diagram mengelola Data Master Supplier menjelaskan sebuah sequence diagram dalam proses add new, search, edit dan delete data master supplier yang dilakukan oleh staf gudang. Adapun sequence diagram dari use case mengelola data dapat dilihat pada gambar V.16 sebagai berikut



Gambar V.16 *Sequence Diagram* Mengelola Data *Master Supplier*  
(Sumber: Hasil Analisis Data, 2016)

#### 4. *Sequence Diagram* pada *Use Case* Mengelola Data *Receipt Material*

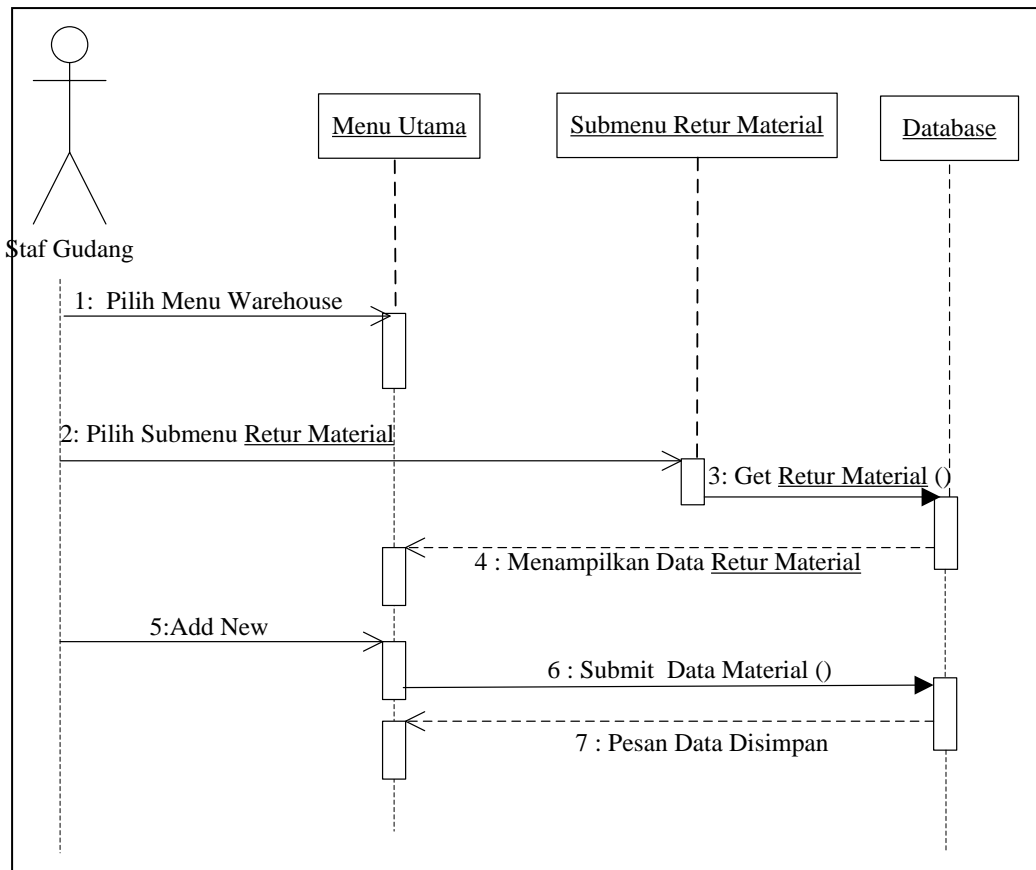
*Sequence diagram* mengelola data *receipt material* menjelaskan sebuah *sequence diagram* yang dilakukan oleh bagian gudang untuk proses pengolahan data *receipt material* ketika adanya penerimaan material dari yang dilakukan oleh staf gudang. Adapun *sequence diagram* dari *use case* mengelola data *receipt material* dapat dilihat pada gambar V.17 sebagai berikut:



Gambar V.17 *Sequence Diagram* Mengelola Data *Receipt Material*  
(Sumber: Hasil Analisis Data, 2016)

##### 5. *Sequence Diagram* pada *Use Case* Membuat Retur Material

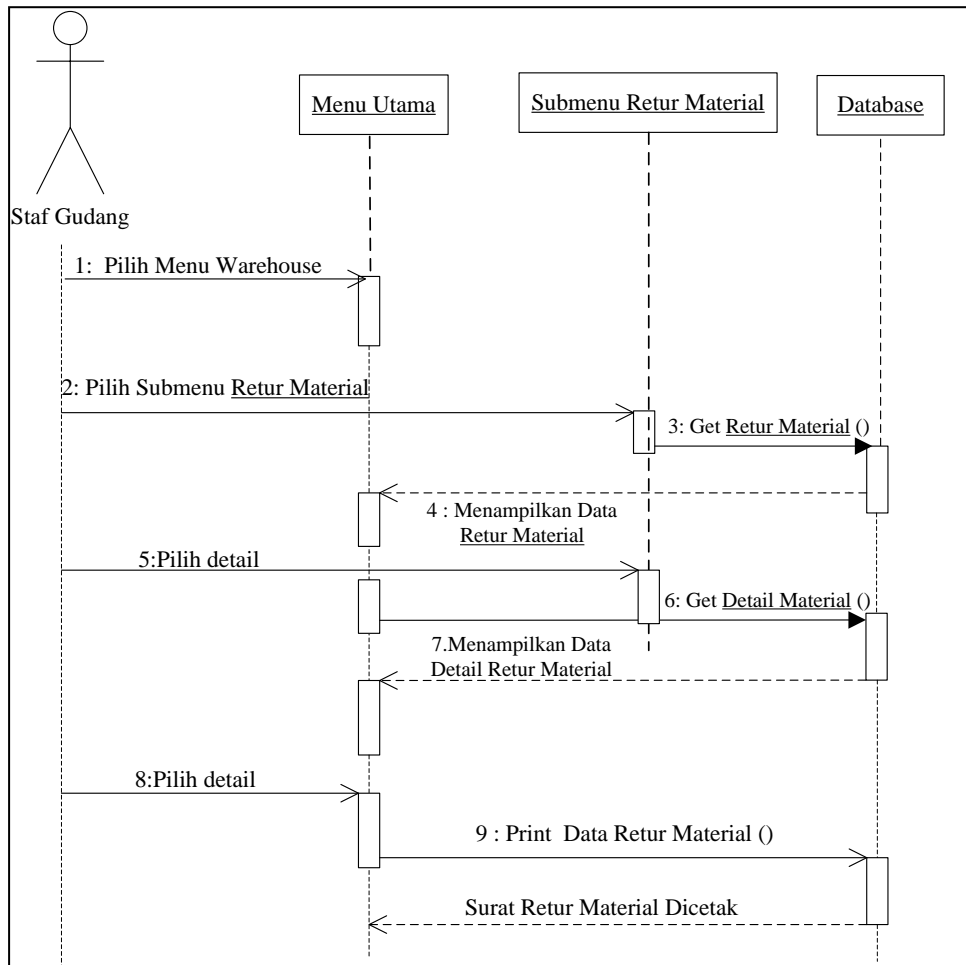
*Sequence diagram* membuat retur material menjelaskan sebuah *sequence diagram* meretur material karena pada penerimaan material terdapat material yang tidak sesuai atau rusak. Proses ini dilakukan oleh staf gudang. Adapun *sequence diagram* dari *use case* melakukan retur material dapat dilihat pada gambar V.18 sebagai berikut:



Gambar V.18 *Sequence Diagram* Membuat Retur Material  
(Sumber: Hasil Analisis Data, 2016)

#### 6. *Sequence Diagram* pada *Use Case* Mencetak Retur Material

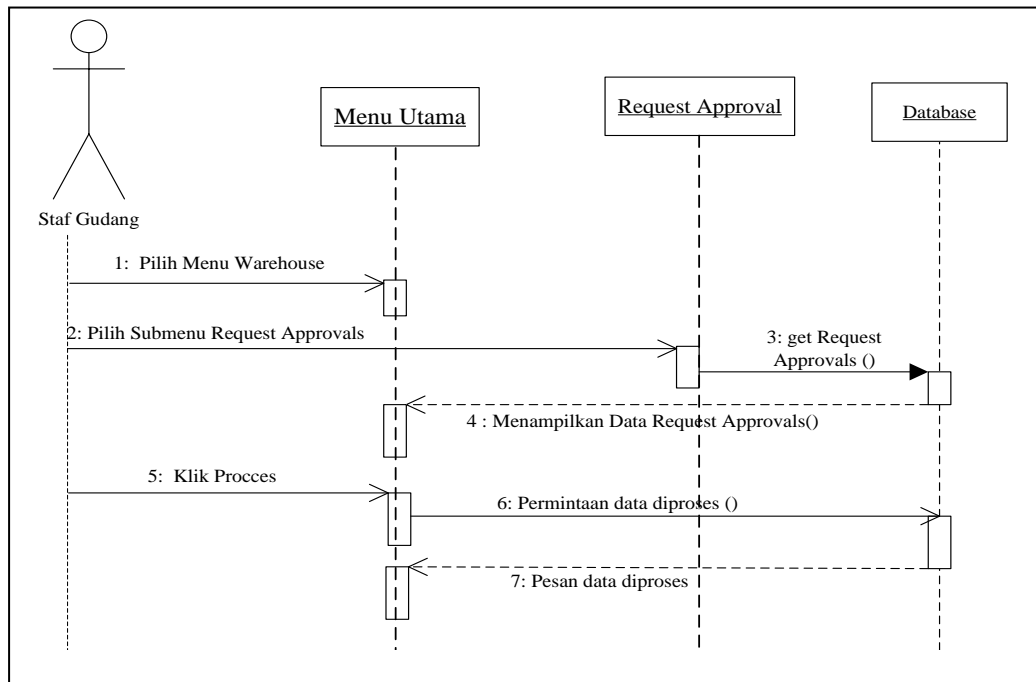
*Sequence diagram* mencetak retur material menjelaskan sebuah *sequence diagram* mencetak surat retur material. Proses ini dilakukan oleh staf gudang. Adapun *sequence diagram* dari *use case* mencetak retur material dapat dilihat pada gambar V.19 sebagai berikut:



Gambar V.19 *Sequence Diagram* Mencetak Retur Material  
(Sumber: Hasil Analisis Data, 2016)

#### 7. *Sequence Diagram* pada *Use Case* Menerima *Request* Material

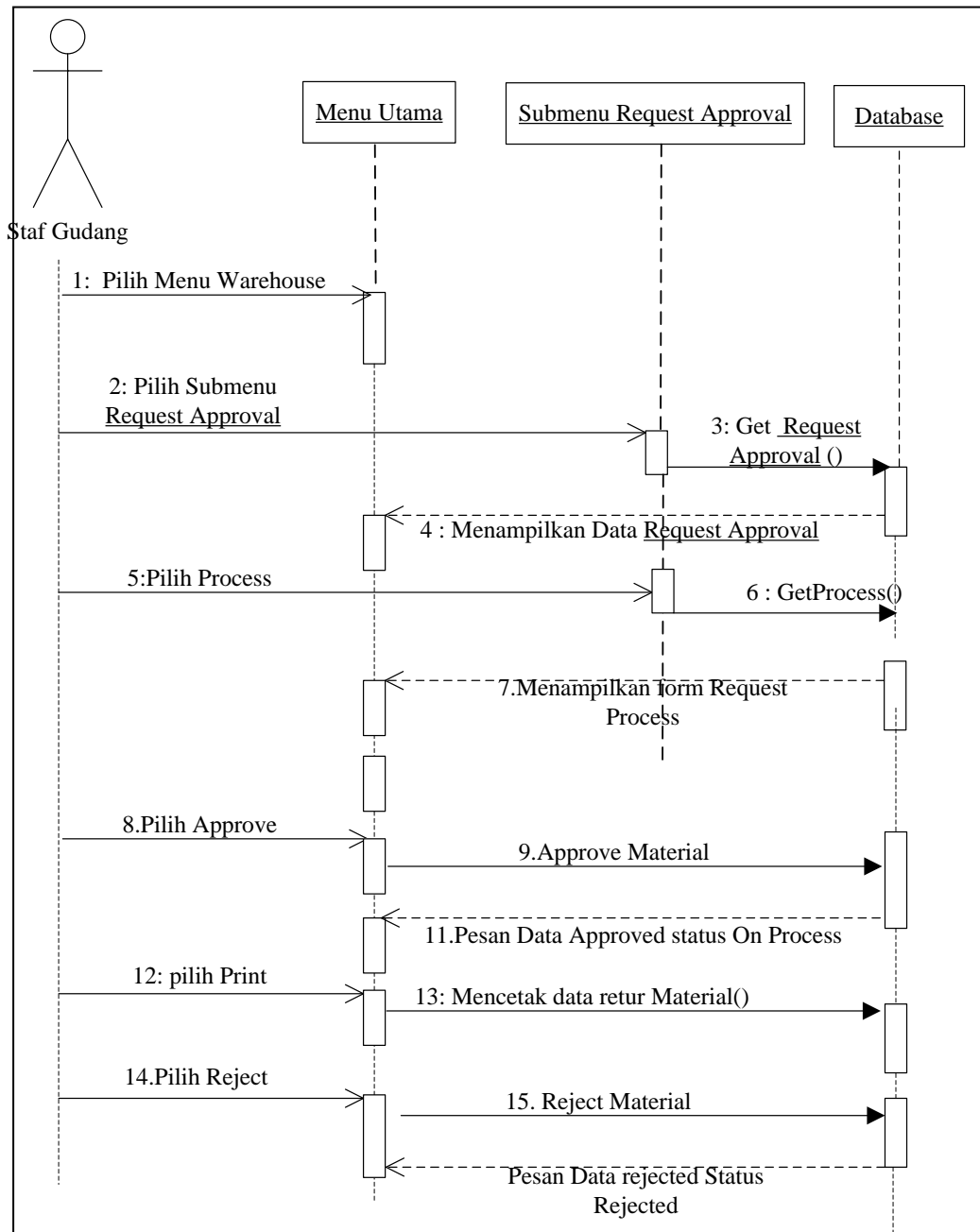
*Sequence diagram* menerima *request* material menjelaskan sebuah *sequence diagram* dalam proses keterangan *request* material dari bagian produksi yang dilakukan oleh staf gudang. Adapun *sequence diagram* dari *use case* menerima *request* material dapat dilihat pada gambar V.20 sebagai berikut:



Gambar V.20 *Sequence Diagram* menerima *request material*  
(Sumber: Hasil Analisis Data, 2016)

#### 8. *Sequence Diagram* pada *Use Case* Melakukan *Approval Material*

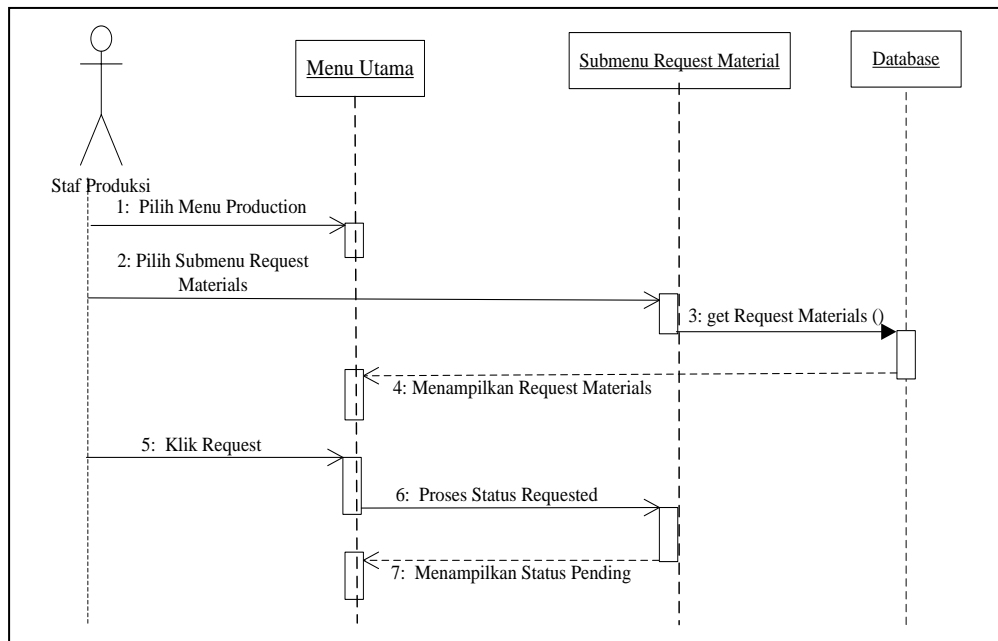
*Sequence diagram* melakukan approval material menjelaskan sebuah *sequence diagram* sesudah proses penerimaan *request material* dari bagian produksi. Adapun *sequence diagram* Melakukan *approval material* dapat dilihat pada gambar V.21 sebagai berikut:



Gambar V.21 *Sequence Diagram* Melakukan Approval Material  
(Sumber: Hasil Analisis Data, 2016)

#### 9. *Sequence Diagram* pada Use Case Melakukan Request Material

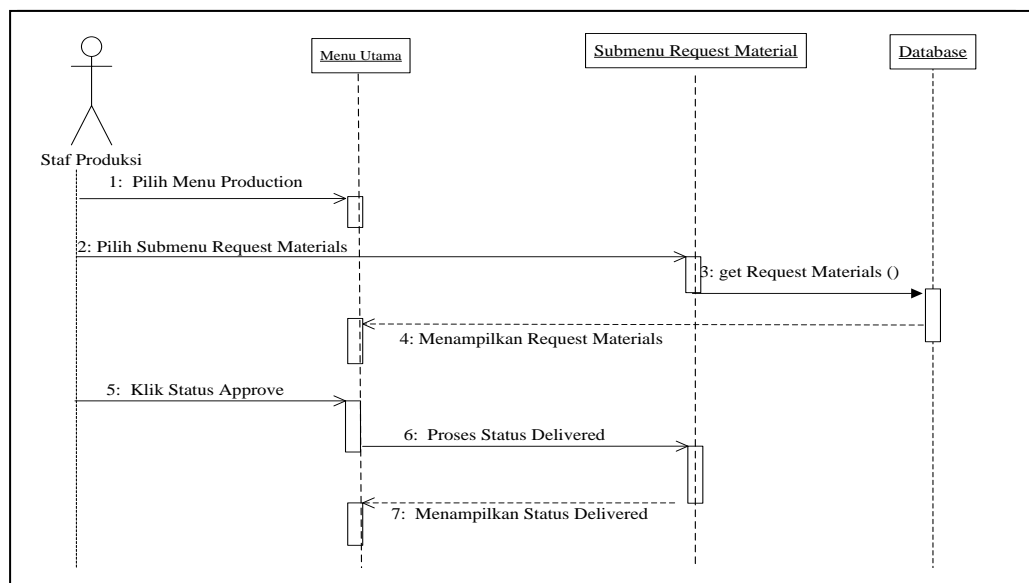
*Sequence diagram* melakukan *request* material menjelaskan proses *sequence Diagram* melakukan *request* material yang dilakukan oleh staf produksi. Adapun *sequence diagram* dari *use case* melakukan *request* material dapat dilihat pada gambar V.22 sebagai berikut:



Gambar V.22 *Sequence Diagram* Melakukan *Request Material*  
(Sumber: Hasil Analisis Data, 2016)

#### 10. *Sequence Diagram* pada *Use Case* Melakukan konfirmasi Material

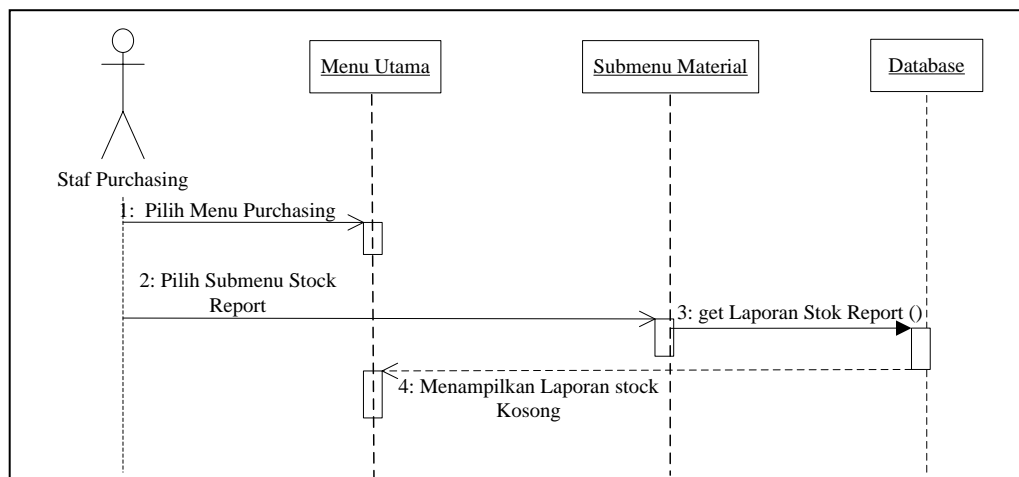
*Sequence diagram* melakukan konfirmasi material menjelaskan sebuah *sequence diagram* yang dilakukan staf produksi untuk mengkonfirmasi bahwa material yang di *request* sudah diterima. Adapun *sequence diagram* dari *use case request material Material* dapat dilihat pad gambar V.23 sebagai berikut:



Gambar V.23 *Sequence Diagram* Melakukan Konfirmasi Material.  
(Sumber: Hasil Analisis Data, 2016)

### 11. Sequence Diagram Melihat dan Mengecek Stock Report

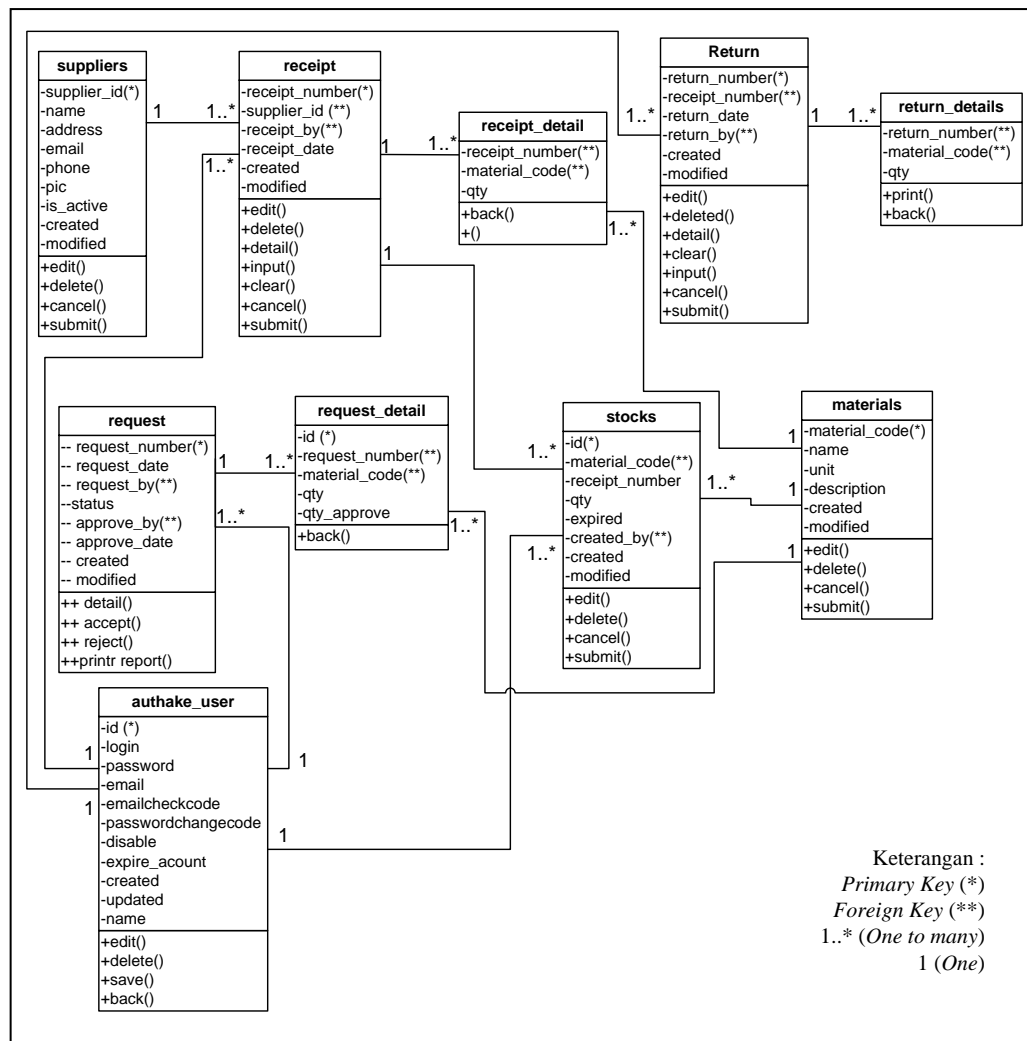
*Sequence diagram* melihat dan mengecek *stock report* menjelaskan sebuah *sequence diagram* dalam proses melakukan pengecekan stok kosong yang dilakukan oleh staf *purchasing*. Adapun *sequence diagram* dari *use case* melakukan pengecekan *stock* kosong dapat dilihat pada gambar V.24 sebagai berikut:



Gambar V.24 *Sequence Diagram* Melihat dan Mengecek *Stock Report*  
(Sumber: Hasil Analisis Data, 2016)

#### 5.3.4 Class Diagram yang Diusulkan.

*Class diagram* pada sistem informasi Pengeluaran *material* gudang ke produksi. *Class diagram* juga membantu dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe *diagram* yang paling banyak digunakan. Rancangan *class diagram* yang diusulkan pada sistem informasi pemeliharaan mesin produksi dapat dilihat pada Gambar V.25.



Gambar V.25 Class Diagram  
 (Sumber: Hasil Analisis Data, 2016)

### 5.3.5 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 kamus data pada aplikasi sistem informasi penerimaan dan pengeluaran material gudang ke produksi:

- Spesifikasi tabel *materials*
  - Nama Tabel : *materials*
  - Fungsi : Untuk menambahkan material
  - Type : File data master

Tabel V.13 Spesifikasi Tabel *Materials*

Elemen	Tipe data	Panjang	Keterangan
Material_code	Varchar	15	<i>Primary key</i>
Name	Varchar	30	
Unit	Varchar	10	
Description	Text		
Create	Datetime		
Modified	Timestamp		

(Sumber: Hasil Analisis, 2016)

2. Spesifikasi tabel *receipt*

Nama Tabel : *receipt*  
 Fungsi : menyimpan data penerimaan material  
 Type : file data transaksi

Tabel V.14 Spesifikasi Tabel *Receipt*

Elemen	Tipe data	Panjang	Keterangan
Receipt_number	Varchar	5	<i>Primary key</i>
Supplier_id	Bigint	15	<i>Foreign key</i>
Receipt_by	Varchar	30	
Receipt_date	Datetime		
Created	Datetime		
Modified	Datetime		

(Sumber: Hasil Analisis, 2016)

3. Spesifikasi tabel *receipt\_details*

Nama Tabel : *receipt\_details*  
 Fungsi : menampilkan rincian dari transaksi *receipt*  
 (penerimaan)  
 Type : *file* data transaksi

Tabel V.15 Spesifikasi Tabel *Receipt Detail*

Elemen	Tipe data	Panjang	Keterangan
Receipt_number	Varchar	5	
Material_code	Varchar	15	
Capacity	Varchar	20	
Qty	Int	11	

(Sumber: Hasil Analisis, 2016)

4. Spesifikasi tabel *request*

Nama Tabel : *request*  
 Fungsi : untuk memproses permintaan material  
 Type : *File* data transaksi

Tabel V.16 Spesifikasi Tabel *request*

Elemen	Tipe data	Panjang	Keterangan
Request_number	Varchar	11	<i>Primary key</i>
Request_date	Datetime		
Request_by	Int	11	
Status	Int	11	
Approve_by	Int	11	
Approve_date	Date		
Created	Datetime		
Modified	Datetime		

(Sumber: Hasil Analisis, 2016)

5. Spesifikasi tabel *request\_details*

Nama Tabel : *request\_details*  
 Fungsi : menampilkan rincian dari transaksi *request*  
 (permintaan)  
 Type : file data transaksi

Tabel V.17 Spesifikasi Tabel *request\_detail*

Elemen	Tipe data	Panjang	Keterangan
ID	Int	11	
Request_number	Varchar	15	
Material_code	Varchar	15	
Qty	Int	11	
Qty_Approve	Int	11	

(Sumber: Hasil Analisis, 2016)

6. Spesifikasi tabel *return*

Nama Tabel : *return*  
 Fungsi : memproses pengembalian barang yang rusak atau cacat saat penerimaan  
 Type : File data transaksi

Tabel V.18 Spesifikasi Tabel *Return*

Elemen	Tipe data	Panjang	Keterangan
Return_number	Varchar	11	<i>Primary key</i>
Receipt_number	Varchar	11	<i>Foreign key</i>
Return_date	Date		
Return_by	Varchar	35	
Created	Datetime		
Modified	Datetime		

(Sumber: Hasil Analisis, 2016)

7. Spesifikasi tabel *return\_details*Nama Tabel : *return\_details*Fungsi : menampilkan rincian dari transaksi *return*  
(*pengembalian barang*)

Type : File data transaksi.

Tabel V.19 Spesifikasi Tabel *return\_detail*

Elemen	Tipe data	Panjang	Keterangan
Return_number	Varchar	15	<i>Primary key</i>
Material_id	Varchar	15	<i>Foreign key</i>
Qty	Int	11	

(Sumber: Hasil Analisis, 2016)

8. Spesifikasi tabel *stocks*Nama Tabel : *stocks*

Fungsi : menampilkan dan menambahkan stock baru

Type : *file* data masterTabel V.20 Spesifikasi Tabel *Stocks*

Elemen	Tipe data	Panjang	Keterangan
Id	Int	11	<i>Primary key</i>
Material_code	Varchar	15	<i>Foreign key</i>
Receipt_number	Varchar	15	
Qty	Int	11	
Expired	Date		
Created_by	Varchar	35	
Created	Datetime		
Modified	Datetime		

(Sumber: Hasil Analisis, 2016)

### 9. Spesifikasi tabel *suppliers*

Nama Tabel	: <i>suppliers</i>
Fungsi	: daftar supplier dan menambahkan <i>supplier</i> baru
Type	: <i>file</i> data master

Tabel V.21 Spesifikasi Tabel *Suppliers*

Elemen	Tipe data	Panjang	Keterangan
Supplier_id	Int	11	<i>Primary key</i>
Name	Varchar	30	
Address	Text		
Email	Varchar	30	
Phone	Varchar	20	
Pic	Varchar	30	
Is_active	Int	11	
Created	Datetime		
Modified	Timestamp		

(Sumber: Hasil Analisis, 2016)

### 5.3.6 *Deployment Diagram* yang Diusulkan

*Deployment diagram* ini digunakan untuk membuat gambaran susunan fisik sebuah sistem, dan gambaran bagian perangkat lunak mana yang berjalan pada perangkat keras yang mana. Berikut ini adalah penjelasan *deployment diagram* dari sistem informasi Pengeluaran *material* dari gudang ke produksi:

#### 1. PHP

PHP adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah aplikasi berbasis web dan bias 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*

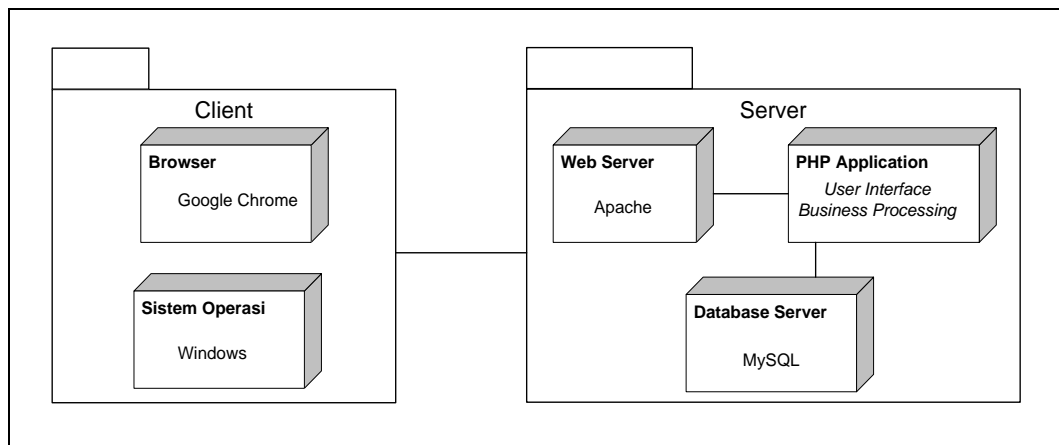
Suatu program sekaligus mesin yang menjalankan program komputer yang mengerti protokol HTTP dan dapat menanggapi permintaan-permintaan dari *web browser*.

### 4. *Web Browser*

*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. Sistem Operasi

Sistem operasi adalah perangkat lunak komputer atau *software* yang bertugas untuk melakukan control dan manajemen perangkat keras dan juga operasi-operasi dasar sistem, termasuk menjalankan *software* aplikasi seperti program-program pengolah data.



Gambar V.26 *Deployment Diagram* yang Diusulan

(Sumber: Hasil Analisis Data, 2016)

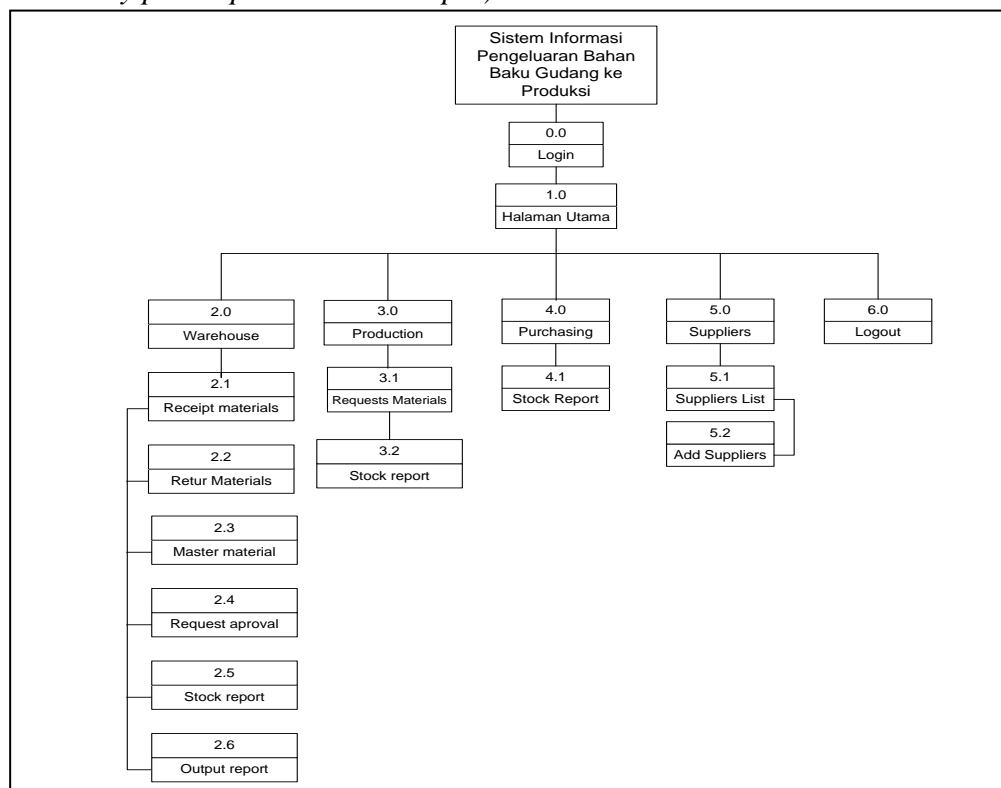
## 5.4 Perancangan Sistem

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

#### 5.4.1 *Hierarchy plus Input-Process-Output (HIPO) Aplikasi*

*HIPO* merupakan alat desain dan teknik dokumentasi dalam siklus pengembangan sistem. Dalam hal ini pembuatan sistem informasi absensi karyawan *outsourcing* dibuat menggunakan perangkat lunak PHP untuk bahas pemrograman, serta *MySQL My Structured Query Language*) yang berfungsi sebagai perangkat lunak basis data.

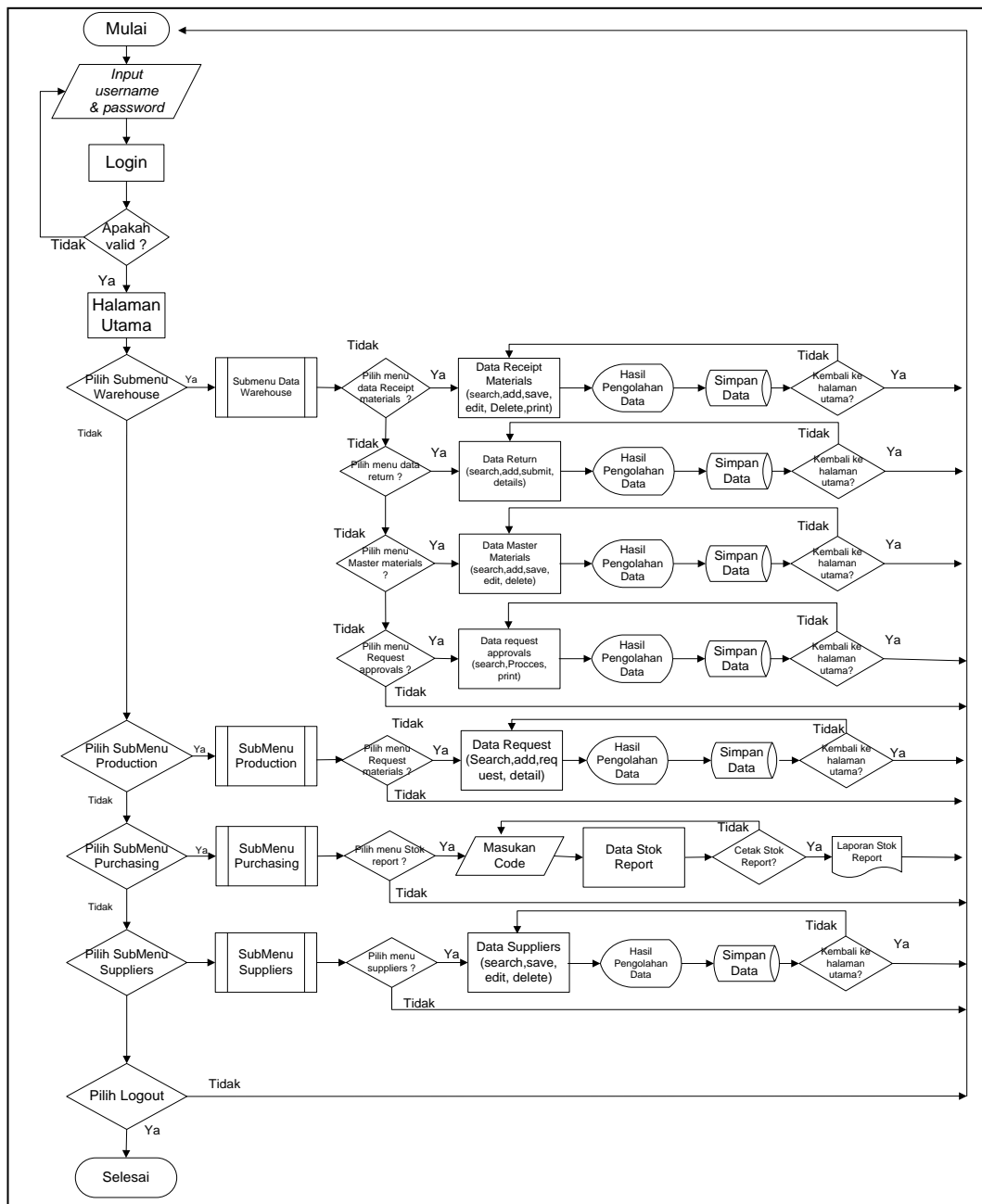
Berikut adalah struktur menu hirarki program yang digambarkan dalam *HIPO* (*Hierarchy plus Input-Process-Output*):



Gambar V.27 HIPO Sistem Usulan  
(Sumber: Hasil Analisis Data, 2016)

#### 5.5 *Flowchart Aplikasi Usulan*

*Flowchart* aplikasi sistem informasi penerimaan dan pengeluaran material gudang ke produksi dapat menggambarkan alur logika yang sebenarnya. Bagian ini juga memperjelas urutan prosedur sistem dan spesifikasi proses. Berikut adalah *flowchart* aplikasi sistem informasi penerimaan dan pengeluaran material gudang ke produksi yang diusulkan terdapat pada gambar V.28:



Gambar V.28 Flowchart Aplikasi Usulan  
(Sumber: Hasil Analisis Data, 2016)

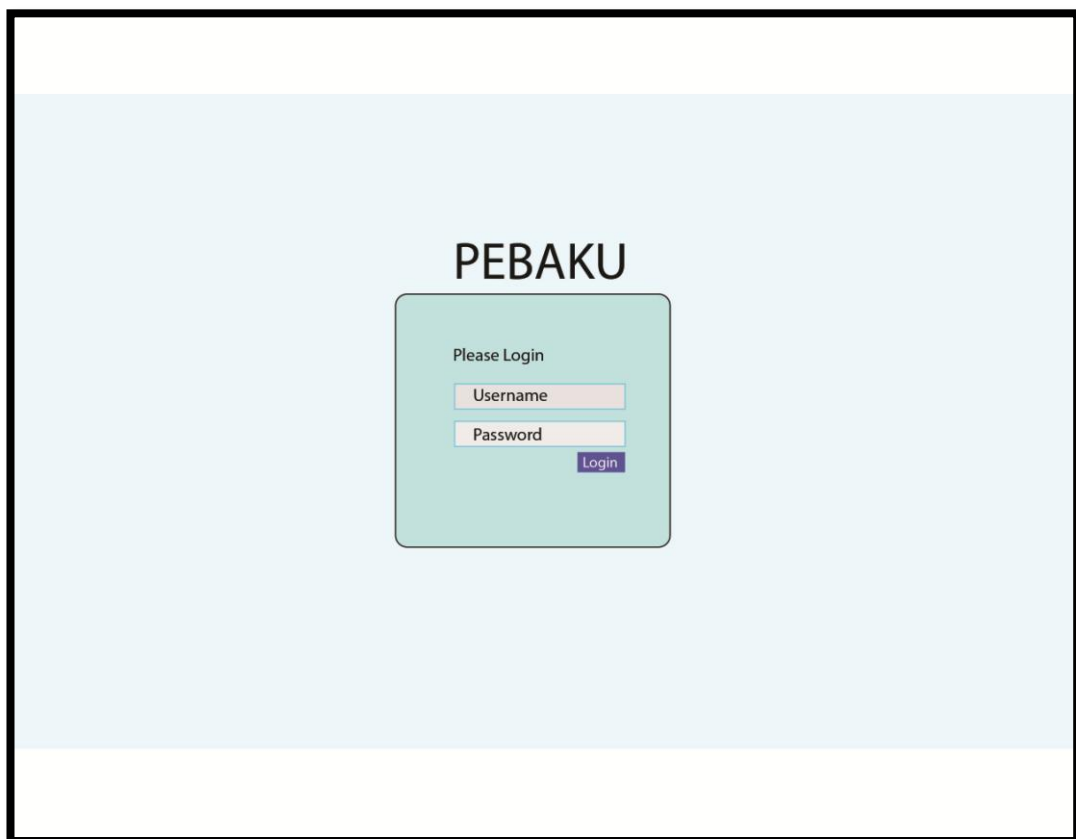
## 5.6 Perancangan Interface Aplikasi Usulan

Interface aplikasi yang terdapat pada penerimaan dan pengeluaran material gudang ke produksi ini berupa rancangan tampilan yang akan dibuat, mulai dari *form login* hingga tampilan laporan yang akan dihasilkan. Perancangan *interface* aplikasi usulan adalah sebagai berikut:

## 1. Tampilan Log in

*Form log in* adalah *form* yang digunakan untuk membedakan hak akses pengguna. *Form login* ini pengguna yang boleh masuk sistem adalah pengguna yang mengetahui *username* dan *password* atau pengguna yang memiliki wewenang untuk menggunakan sistem. Berikut adalah gambar perancangan *form log in*:

- Tombol *log in* : Untuk masuk ke sistem.
- Username : Berisikan nama pengguna.
- Password : Berisikan kata sandi pengguna.

The image shows a login form titled "PEBAKU". The form is centered on a light blue background. It contains the text "Please Login" above two input fields: "Username" and "Password". Below the "Password" field is a purple "Login" button. The entire form is enclosed in a rounded rectangular border.

Gambar V.29 Rancangan *Form Login*  
(Sumber: Hasil Analisis Data, 2016)

## 2. Tampilan Menu / Home

*Form Menu* adalah salah satu cara untuk menyajikan petunjuk yang tersedia kepada *user* dan *Form* menu utama ini dirancang untuk memudahkan *user* dalam mengakses aplikasi dengan pilihan menu yang sudah disediakan sesuai

dengan kebutuhan. Rancangan *interface* dari *form* menu utama dapat dilihat pada gambar berikut. Menu yang terdapat pada menu utama dalam aplikasi yaitu:

Gambar V.30 Rancangan *Form Menu*  
(Sumber: Hasil Analisis Data, 2016)

Menu yang terdapat pada menu utama yaitu:

a. *Home*

Merupakan tampilan dasar utama dan awal, ketika kita *login* program atau suatu aplikasi dan terdapat *list of request for warehouse* (daftar permintaan untuk gudang) dan *list of request from production* (daftar permintaan dari produksi).

3. Tampilan *Warehouse*

Tampilan dari *Warehouse* di *home* terdapat *submenu* yang terdiri dari

- *receipt of material.*
- *Return.*
- *Master of material.*
- *Request approvals.*
- *stock.*
- *output report.*

Berikut gambarnya:

The screenshot shows the PEBAKU software interface. On the left is a sidebar menu with the following items: Home, Warehouse, Production, Purchasing, Suppliers, and Master User. The main content area is titled 'HOME' and contains a search bar and a table titled 'LIST OF REQUEST ON PRODUCTION'. The table has the following columns: No, Request Number, Request Date, Request by, and Status. Below the table is a 'Show' button and another search bar.

Gambar V.31 Rancangan *Form Warehouse*  
(Sumber: Hasil Analisis Data, 2016)

Submenu dari *warehouse* adalah sebagai berikut

a. Tampilan *Receipt Materials*

*Form Receipt materials* atau penerimaan *material* menampilkan data-data material yang telah diinput melalui *add reception*. Rancangan *interface* dari *form receipt materials* dapat dilihat pada gambar V.32. Berikut penjelasan penggunaan pada *form* ini:

- Add new : Untuk menambahkan atau input penerimaan baru.
- Show : Pengaturan berapa baris data yang akan ditampilkan.
- Search : Kolom untuk mencari data material.
- Receipt number : Kolom berisikan nomor penerimaan.
- Supplier : Kolom berisikan nama *Supplier*.
- PIC Supplier : Kolom berisikan nama PIC *supplier*.
- Received by : Kolom berisikan nama penerima.

- Received date : Kolom berisikan tanggal penerimaan.
- Status : Kolom berisikan tombol edit, delete, detail.
- Edit : Tombol untuk merubah data penerimaan.
- Delete : Tombol untuk menghapus data penerimaan.
- Detail : Rincian data penerimaan.

The screenshot displays the 'Receipt Material' form within the PEBAKU system. On the left is a vertical sidebar menu with the following items: Home, Warehouse, Production, Purchasing, Suppliers, and Master User. The main content area features a light blue header with the text 'Receipt Material'. Below the header, there is a blue 'Add New' button. To the right of the button are two input fields: 'Show' and 'Search:'. Below these elements is a table with the following structure:

No	Receipt Number	Supplier	PIC Supplier	Received by	Receipt date	Status
						edit delete detail

Gambar V.32 Rancangan *Form Receipt Material*  
(Sumber: Hasil Analisis Data, 2016)

Berikut *Form Add Reception* adalah *form* untuk menambahkan *reception* atau penerimaan material yang baru dan merupakan *subform* dari *form receipt material*. Rancangan *interface* dari *form Add reception* dapat dilihat pada gambar V.33. Berikut penjelasan penggunaan pada *form* ini:

- Receipt number : Berisikan nomor penerimaan.
- Supplier : Berisikan nama *supplier* barang yang diterima.
- Receipt date : Berisikan tanggal transaksi penerimaan.
- Material Code : Berisikan *kode* material.
- Add New Material : Tombol untuk menambahkan *material* baru.

- Qty : Berisikan jumlah material.
- Clear : Tombol membersihkan kolom detail material.
- Add : Tombol untuk menambahkan data sementara.
- Cancel : Tombol untuk membatalkan.
- Save : Tombol untuk menambah dan menyimpan data yang telah diisi.

**PEBAKU**

Home

Warehouse

Production

Purchasing

Suppliers

Master User

## Add Reception

**FORM ADD RECEPTION**

Reception Number

Supplier

Reception date

**Detail Material**

Material Code

Qty

**List Of Materials**

Code	Name	Qty

Gambar V.33 Rancangan *Form Add Reception*  
(Sumber: Hasil Analisis Data, 2016)

b. Tampilan *Return Materials*

*Form Return Materials* adalah form yang menampilkan data-data pengembalian material yang tersimpan dan terdaftar. Rancangan *interface* dari *form Return materials* dapat dilihat pada gambar V.34. Berikut penjelasan penggunaan pada *form* ini:

- Add New : Tombol untuk menambahkan data retur baru.
- Show : Pengaturan berapa baris data yang akan ditampilkan.
- Search : Kolom untuk mencari data.
- No : Kolom berisikan no urut data.
- Return Number : Berisikan nomor pengembalian.
- Reception Number : Berisikan nomor penerimaan.
- Return date : Berisikan tanggal pengembalian.
- Detail : Rincian data penerimaan.

The screenshot shows a web application interface for 'Return Material'. On the left is a vertical sidebar menu with the following items: Home, Warehouse, Production, Purchasing, Suppliers, and Master User. The main content area has a header 'Return Material' and an 'Add New' button. Below this are two input fields: 'Show' and 'Search:'. A table is displayed with the following structure:

No	Return Number	Reception Number	Return Date	Receipt date	Action
					detail

Gambar V. 35 Rancangan *Form Retur Material*  
(Sumber: Hasil Analisis Data, 2016)

Berikut *Form Add Retur Material* adalah *form* untuk mengembalikan atau menukar material baku yang cacat, tidak sesuai pesanan, rusak, pada saat pengecekan material dipesan kepada *supplier*.

Rancangan *interface* dari *form add retur* dapat dilihat pada gambar V.36  
Berikut penjelasan penggunaan pada *form* ini:

- Return number : Berisikan nomor pengembalian.
- Receipt number : Berisikan nomor penerimaan.
- Date of retur : Berisikan tanggal retur.
- Material Code : Berisikan kode material.
- Qty : Berisikan jumlah material yang akan diretur.
- Tombol clear : Untuk mengosongkan kolom-kolom detail material.
- Tombol Submit : Untuk menginput atau menyimpan data yang telah diisi.

The screenshot shows a web application interface for adding return materials. On the left is a dark sidebar with menu items: Home, Warehouse, Production, Purchasing, Suppliers, and Master User. The main area has a light blue header 'Add Retur Material' and a sub-header 'FORM ADD RETUR'. The form includes three input fields for 'Return Number', 'Reception Number', and 'Date of Return'. Below these is a 'Detail Material' section with a 'Material Code' input field, an 'Add New Material' button, and a 'Qty' input field. At the bottom of this section are 'Clear' and 'Add' buttons. Below the detail section is a table titled 'List Of Temporary Materials' with columns for 'Code', 'Name', and 'Qty'. At the very bottom right are 'Cancel' and 'Submit' buttons.

Gambar V.36 Rancangan *Form Add Retur Material*  
(Sumber: Hasil Analisis Data, 2016)

c. *Master Material*

*Form Master Material* adalah *form* yang menampilkan data-data material yang dipakai di perusahaan. untuk menambahkan material master supaya databasanya muncul ketika bagian produksi memesan material ke bagian gudang. Rancangan *interface* dari *form Master Materials* dapat dilihat pada gambar V.37. Berikut penjelasan penggunaan pada *form* ini:

- Add New : Tombol untuk menambahkan data retur baru.
- Show : Pengaturan brapa baris data yang akan ditampilkan.
- Search : Kolom untuk mencari data.
- Edit : Tombol untuk merubah data penerimaan.
- Delete : Tombol untuk menghapus data penerimaan.
- No : Berisikan no urut data.
- Material Code : Berisikan kode material.
- Material Name : Berisikan nama material.
- Description : Berisikan keterangan material.
- Unit : Berisikan satuan dari material.
- Action : Berisikan tombol edit dan *delete*

No	Material code	Material name	Description	Unit	Status
					Edit Delete
					Edit Delete

Gambar V.37 Rancangan *Form Master Material*

(Sumber: Hasil Analisis Data, 2016)

Berikut *Form add new material* adalah *form* untuk menambahkan data material awal atau data material baru. Ketika ada material baru yang belum ada sebelumnya maka di *add materials* inilah proses input data tersebut dapat dilakukan. Rancangan *interface* dari *form add materials* dapat dilihat pada gambar V.38 Berikut penjelasan penggunaan pada *form* ini:

- Material Code : Berisikan kode material .
- Material Name : Berisikan nama material.
- Unit : Berisikan satuan material.
- Description : Berisikan keterangan material.
- Cancel : Tombol untuk membatalkan dan kembali ke halaman sebelumnya.
- Save : Untuk menginput atau menyimpan data yang telah diisi dan proses akhir dari *form* ini.

The image shows a web application interface for 'PEBAKU'. On the left is a dark sidebar with white text for navigation: Home, Warehouse, Production, Purchasing, Suppliers, and Master User. The main content area has a light gray header with the title 'Add New Material'. Below the header, the form is titled 'FORM ADD NEW MATERIAL' in teal. It contains four input fields: 'Material Code' (a single-line text box), 'Material Name' (a single-line text box), 'Unit' (a single-line text box), and 'Description' (a larger multi-line text area). At the bottom right of the form are two buttons: a gray 'Cancel' button and a teal 'Save' button.

Gambar V.38 Rancangan *Form Add New Material*  
(Sumber: Hasil Analisis Data, 2016)

d. *Request Approval*

*Form Request Approval* adalah *form* untuk menyetujui proses permintaan yang dilakukan oleh bagian produksi. Rancangan *interface* dari

*form Request Approval* dapat dilihat pada gambar V.39 Berikut penjelasan penggunaan pada *form* ini:

- Show : Pengaturan berapa baris data yang akan ditampilkan.
- Search : Kolom untuk mencari data.
- No : Berisikan no urut dari data permintaan produksi.
- Request Number : berisikan nomor permintaan.
- Request Date : Berisikan tanggal permintaan.
- Request by : Berisikan nama yang melakukan permintaan.
- Status : Berisikan keterangan apakah pending, diproses dan diterima.
- Action : Berisikan tombol untuk memproses / menyetujui permintaan.

The screenshot shows a web application interface for 'PEBAKU'. On the left is a dark sidebar menu with the following items: Home, Warehouse, Production, Purchasing, Suppliers, and Master User. The main content area has a light blue header 'Request Approval'. Below the header is a white box containing an 'Add New' button, a 'Show' input field, and a 'Search:' input field. Below this is a table titled 'LIST STOCK OF MATERIALS'. The table has the following structure:

No	Request Number	Request Date	Request by	Status	Action
				Delivered	Print
				Delivered	Print

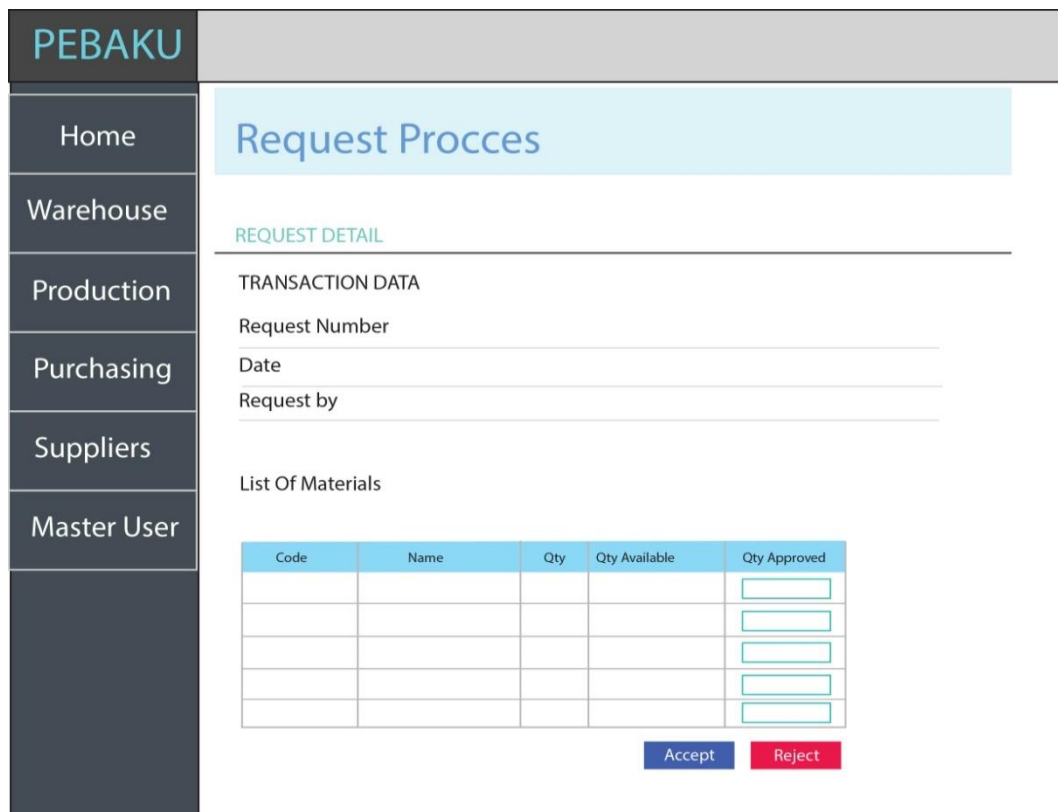
Gambar V.39 Rancangan *Form Request Approval*  
(Sumber: Hasil Analisis Data, 2016)

e. *Request Procces*

*Form Request Approvals* adalah *form* yang, muncul ketika *user* bagian gudang mengklik tombol *procces*, *form* ini untuk menyetujui

material apa saja yang disetujui berdasarkan stok material yang tersedia. Rancangan *interface* dari *form Request Procces* dapat dilihat pada gambar V.40 Berikut penjelasan penggunaan pada *form* ini:

- Request Number : Berisikan keterangan nomor permintaan.
- Date : Berisikan keterangan tanggal.
- Request by : Berisikan keterangan yang melakukan Permintaan.
- Code : Berisikan kode material.
- Name : Berisikan nama material.
- Qty : Berisikan jumlah material.
- Qty Approval : Berisikan jumlah material yang disetujui.
- Accept : Tombol terima permintaan.
- Reject : Tombol penolakan permintaan.



PEBAKU

Home

Warehouse

Production

Purchasing

Suppliers

Master User

## Request Procces

REQUEST DETAIL

TRANSACTION DATA

Request Number

Date

Request by

List Of Materials

Code	Name	Qty	Qty Available	Qty Approved
				<input type="text"/>
				<input type="text"/>
				<input type="text"/>
				<input type="text"/>
				<input type="text"/>

Accept Reject

Gambar V.40 Rancangan *Request Procces*  
(Sumber: Hasil Analisis Data, 2015)

f. Tampilan *Stock*

*Form Stock* adalah untuk menambahkan *stock material* berdasarkan penerimaan. dapat dilihat pada gambar V.41 Berikut penjelasan penggunaan pada *form* ini:

- Tombol *add new* : Untuk menambahkan *stock material*.
- Tombol *edit* : Untuk merubah data yang ada di tabel .
- Kolom *show* : Untuk membatasi tampilan daftar tabel terdiri dari 5, 10, dan *show all* (tampilkan semua).
- Kolom Search : Untuk mencari data yang ada pada tabel.

Gambar V.41 Rancangan *Form Stock*  
(Sumber: Hasil Analisis Data, 2016)

g. *Stock report*

*Form stock report* adalah *form* yang menampilkan *stock material* dan daftar material yang sudah kosong. Rancangan *interface* dari *form Stock report* dapat dilihat pada gambar V.42 Berikut penjelasan penggunaan pada *form* ini:

- Empty stock : Keterangan bahwa *table empty stock* / bagian tabel stok kosong.
- Entries : Pengaturan berapa baris data yang akan ditampilkan.
- Search : Kolom untuk mencari data.
- Code : Berisikan kode material yang kosong.
- Name : Berisikan nama material yang kosong.
- List of Materials : Keterangan untuk tabel daftar material.
- Code : Berisikan kode material.
- Name : Berisikan nama material.
- Expired : Berisikan jumlah material yang *expired*.
- Not Expired : Berisikan jumlah material yang belum *expired*.

Gambar V.42 Rancangan *Form Stock Report*  
(Sumber: Hasil Analisis Data, 2016)

#### *h. Output Report Material*

*Form output report material* adalah *form* yang menampilkan laporan seluruh pengeluaran material baku yang ada di gudang. Rancangan

*interface* dari *form output report material* dapat dilihat pada gambar V.43

Berikut penjelasan penggunaan pada *form* ini:

- Show : Pengaturan berapa baris data yang akan ditampilkan.
- Search : Kolom untuk mencari data.
- Request code : Berisikan kode permintaan material.
- Requestor : Berisikan nama *user* yang melakukan Permintaan.
- Requestment date : Berisikan tanggal permintaan.
- Approver : Berisikan nama *user* bagian gudang yang menyetujui permintaan.
- Approvement date : Berisikan tanggal persetujuan oleh bagian gudang.
- Material : Berisikan data jumlah material yang diminta disetujui dan tidak disetujui.

The screenshot displays the PEBAKU web application interface. On the left is a dark sidebar menu with the following items: Home, Warehouse, Production, Purchasing, Suppliers, and Master User. The main content area has a light blue header with the text 'OUTPUT REPORT MATERIALS'. Below the header, there are several buttons: Print, Copy, PDF, Excel, CSV, and Columes. There are also two input fields: 'Show' and 'Search:'. Below these is a table titled 'LIST STOCK OF MATERIALS' with the following columns: Request code, Requestor, Requestment Date, Approver, Approvement Date, and Material Name. The table currently contains two empty rows.

Gambar V.43 Rancangan *Output Report Materials*  
(Sumber: Hasil Analisis Data, 2016)

#### 4. Tampilan *Production*

Tampilan dari *production* di *home* terdapat *submenu* yang terdiri dari *request*. *Form* ini khusus untuk bagian produksi agar supaya mereka bisa melakukan permintaan *material*.

##### a. *Request Material*

*Form Request material* adalah *form* permintaan material untuk bagian produksi. Rancangan *interface* dari *form Request Material* dapat dilihat pada gambar V.44 Berikut penjelasan penggunaan pada *form* ini:

- Add New : Tombol untuk menambahkan *request* material baru.
- Show : Pengaturan berapa baris data yang akan ditampilkan.
- Search : Kolom untuk mencari data.
- No : Berisikan no urut data.
- Request Number : Berisikan kode material.
- Nama : Berisikan nama material.
- Description : Berisikan keterangan material.
- Action : Berisikan tombol rincian data permintaan.

The screenshot shows a web application interface for 'Request Material'. On the left is a dark sidebar with the logo 'PEBAKU' at the top and navigation links: Home, Warehouse, Production, Purchasing, Suppliers, and Master User. The main content area has a light blue header 'Request Material'. Below the header, there is an 'Add New' button in a blue box, a 'Show' dropdown menu, and a 'Search:' input field. The main content area contains a table titled 'LIST STOCK OF MATERIALS' with the following columns: No, Request Number, Request Date, Request by, Status, and Action. The table has two empty rows.

Gambar V.44 Rancangan *Request Material*  
(Sumber: Hasil Analisis Data, 2016)

Berikut *Form add Request* adalah *form* untuk menambahkan data *request* atau permintaan material ke gudang yang dilakukan oleh bagian produksi. Rancangan *interface* dari *form add request* dapat dilihat pada gambar V.45 Berikut penjelasan penggunaan pada *form* ini:

- Request Number : Berisikan kode material.
- Date : Berisikan tanggal permintaan.
- Code : Berisikan kode material.
- Qty : Berisikan jumlah material.
- Clear : Untuk membersihkan kolom-kolom *detail* material.
- Cancel : Untuk membatalkan dan kembali ke halaman Sebelumnya.
- Submit : Untuk menginput atau menyimpan data yang telah diisi dan proses akhir dari *form* ini.

**PEBAKU**

Home | Add Request

Warehouse | **FORM ADD REQUEST**

Production | Request Number

Purchasing | Date

Suppliers | **Detail Material**

Master User | Material Name

Qty

List Of Temporary Materials

Code	Name	Qty

Gambar V.45 Rancangan *Form Add Request*  
(Sumber: Hasil Analisis Data, 2016)

## 5. Tampilan *Purchasing*

Tampilan dari *purchasing* di *home* terdapat *submenu* yang terdiri dari *stock report*. Form ini khusus untuk *purchasing* agar dapat memantau laporan stock dan *stock kosong*.

### a. *Stock Report Purchasing*

*Form stock report* adalah *form* yang menampilkan *stock* dari material dan daftar material yang sudah kosong. Rancangan *interface* dari *form stock report* dapat dilihat pada gambar V.46 Berikut penjelasan penggunaan pada *form* ini:

- Empty stock : Keterangan bahwa *table empty stock* / bagian tabel stok kosong.
- Entries : Pengaturan berapa baris data yang akan ditampilkan.
- Search : Kolom untuk mencari data.
- Material Code : Berisikan code material yang kosong.
- Material Name : Berisikan nama material yang kosong.
- List of Materials : Keterangan untuk tabel daftar material.
- Expired : Berisikan jumlah material yang kadaluarsa.
- Not Expired : Berisikan jumlah material yang belum *expired*.

Gambar V.46 Rancangan *Form Stock Report Purchasing*  
(Sumber: Hasil Analisis Data, 2016)

## 6. Tampilan *Suppliers*

Tampilan dari *Suppliers* di *home* terdapat *submenu* yang terdiri dari *Supplier list* dan *add Suppliers*.

### a. *Supllier list*

Berikut *Form Supplier list* adalah *form* yang hanya menampilkan *data supplier* yang bekerja sama dengan perusahaan. Rancangan *interface*

dari *form add request* dapat dilihat pada gambar V.47 Berikut penjelasan penggunaan pada *form* ini:

- Add new : Untuk menambahkan atau input penerimaan baru.
- Show : Pengaturan brapa baris data yang akan ditampilkan.
- Search : Kolom untuk mencari material.
- No : Berisikan no urut data.
- Company : Berisikan nama perusahaan *supplier*.
- Address : Berisikan alamat *supplier*.
- PIC : Berisikan nama PIC dr *supplier*.
- Email : Berisikan email *supplier*
- Contact : Berisikan no *handphone* atau telepon *supplier*.
- Active : Berisikan status *supplier* apakah aktif atau tidak aktif.
- Action : Berisikan tombol pilihan untuk *edit* (ubah) dan *delete* (hapus).

The screenshot shows the PEBAKU system interface for managing suppliers. The sidebar on the left contains the following menu items: Home, Warehouse, Production, Purchasing, Suppliers, and Master User. The main content area is titled 'Suppliers' and contains a 'MANAGE TABLE' section. This section includes an 'Add New' button, a 'Show' dropdown menu, and a 'Search' input field. Below these controls is a table with the following columns: No, Company, Address, PIC, Email, Contact, Active, and Action. The table displays two rows of data, each with an 'Active' status and 'Edit' and 'Delete' action buttons.

No	Company	Address	PIC	Email	Contact	Active	Action
						Active	Edit Delete
						Active	Edit Delete

Gambar V.47 Rancangan *Form Supplier List*  
(Sumber: Hasil Analisis Data, 2016)

b. *Add Suppliers*

Berikut *Form Add Supplier* adalah *form* untuk menambahkan *supplier-supplier* baru. Rancangan *interface* dari *form add Supplier* dapat dilihat pada gambar V.48 Berikut penjelasan penggunaan pada *form* ini:

- **Company** : Berisikan nama *supplier* baru.
- **Address** : Berisikan alamat *supplier* baru.
- **PIC** : Berisikan nama PIC *supplier* baru.
- **Email** : Berisikan email *supplier* baru.
- **Contact** : Berisikan kontak ( nomor *handphone*) *supplier* baru.
- **Active** : Check box untuk menyatakan bahwa *supplier* aktif
- **Cancel** : Untuk membatalkan dan kembali ke halaman Sebelumnya.
- **Submit** : Untuk menginput atau menyimpan data yang telah diisi dan proses akhir dari *form* ini.

The image shows a web application interface for 'PEBAKU'. On the left is a vertical navigation menu with the following items: Home, Warehouse, Production, Purchasing, Suppliers, and Master User. The main content area is titled 'Add Suppliers' and contains a form labeled 'FORM ADD SUPPLIER'. The form has the following fields and controls:

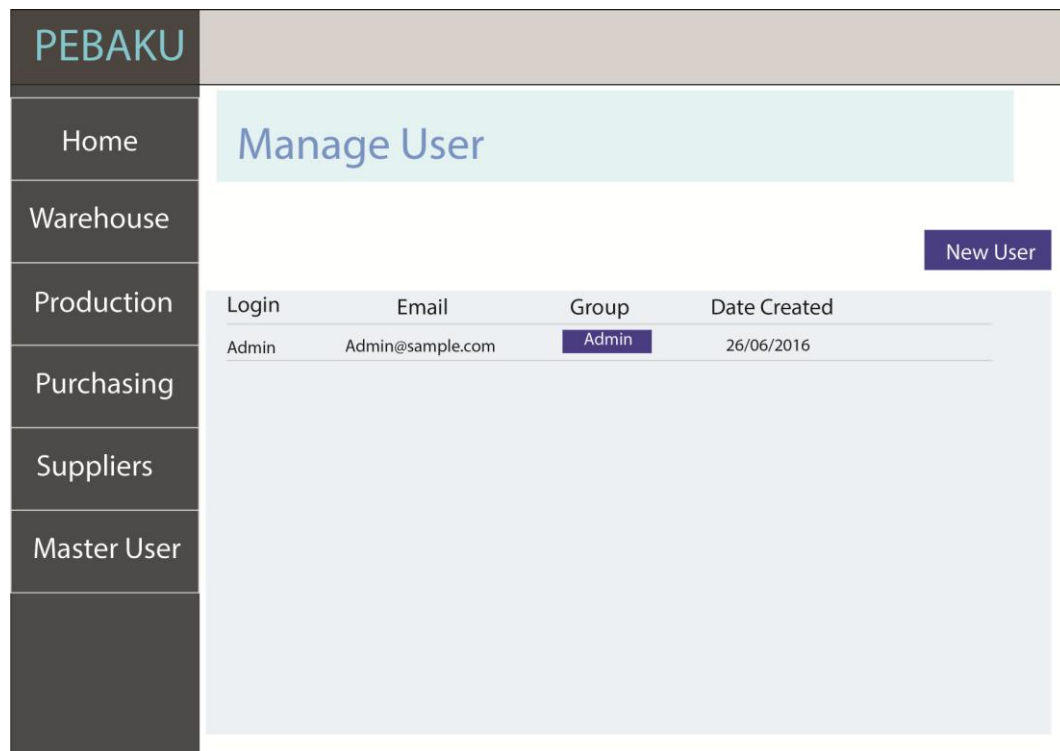
- Company**: A single-line text input field.
- Address**: A multi-line text input field.
- PIC**: A single-line text input field.
- Email**: A single-line text input field.
- Contact**: A single-line text input field.
- Active?**: A checkbox.
- Buttons**: A grey 'Cancel' button and a teal 'Submit' button.

Gambar V.48 Rancangan *Add Suppliers*  
(Sumber: Hasil Analisis Data, 2016)

## 7. Tampilan *Master User*

Tampilan dari *Master User* di *home* terdapat *submenu* yang terdiri dari *Manage Users* dan *add Users*. Rancangan *interface* dari *form Manage Supplier* dapat dilihat pada gambar V.49 Berikut penjelasan penggunaan pada *form* ini:

- **New User** : Tombol untuk membuat *user* baru.
- **Log in** : Berisikan nama *user* untuk *log in*.
- **Email** : Berisikan *email user*.
- **Group** : Berisikan untuk hak akses bagi para *user*.
- **Date Created** : Berisikan tanggal *user* baru dibuat.



Gambar V.49 Rancangan *form Manage User*  
(Sumber: Hasil Analisis Data, 2016)

### a. *New User*

Berikut *Form New User* adalah *form* untuk menambahkan *user* baru. Rancangan *interface* dari *form New User* dapat dilihat pada gambar V.50. Berikut penjelasan penggunaan pada *form* ini:

- Log in : Berisikan nama pengguna untuk *log in*.
- Name : Berisikan nama asli pengguna.
- Password : Berisikan kata sandi untuk *log in*.
- Email : Berisikan *email* pengguna.
- Group : Kolom pilihan untuk hak akses pengguna Baru.
- Save : Tombol untuk menyimpan data.
- Back : Tombol untuk kembali ke halaman sebelumnya.

The image shows a web application interface for 'PEBAKU'. On the left is a dark sidebar with menu items: Home, Warehouse, Production, Purchasing, Suppliers, and Master User. The 'Home' menu item is selected, and the main content area displays the 'Add User' form. The form has a light blue header with the title 'Add User'. Below the header are five input fields: 'Login', 'Name', 'Password', 'Email', and 'Group'. At the bottom of the form are two buttons: 'Save' (blue) and 'Back' (white with blue border).

Gambar V.50 Rancangan *Form Add User*  
(Sumber: Hasil Analisis Data, 2015)

## 5.7 Analisis Software dan Hardware

Untuk dapat menggunakan aplikasi, tentunya diperlukan suatu spesifikasi perangkat keras (*hardware*) ataupun perangkat lunak (*software*) yang mendukung agar aplikasi dapat berjalan dengan baik. Adapun spesifikasinya sebagai berikut:

1. Kebutuhan *Hardware*
  - a. *Processor* : Intel Core i3.
  - b. *Main Memory* : 1 GB
  - c. *Hard Disk* : 250 GB
  - d. *Graphic Adapter (VGA)* : 500 MB

## 2. *Kebutuhan Software*

- a. *Sistem Operasi* : *Microsoft Windows 7*
- b. *Web Server* : *Apache*
- c. *Database Server* : *MySQL 5.5.27*
- d. *Design Interface* : *PHP 5.4.7*
- e. *Web Browser* : *Mozilla firefox, Google Chrome.*

## **BAB VI**

### **KESIMPULAN DAN SARAN**

#### **6.1 Kesimpulan**

Berdasarkan hasil pengumpulan, pengolahan dan analisis data yang telah dilakukan dalam penelitian ini dapat disimpulkan bahwa:

1. Dengan adanya aplikasi ini sistem penerimaan dan pengeluaran material gudang ke produksi pada PT Tetrapak Stainless Jakarta tidak lagi menggunakan *Microsoft Excel* dan di ganti dengan yang lebih baik yaitu menggunakan *My SQL*.
2. Waktu yang dibutuhkan dalam proses pengeluaran material berdasarkan permintaan dari bagian produksi lebih cepat dari sebelumnya.
3. Dengan menggunakan aplikasi ini mengurangi penggunaan kertas yang dapat terselip atau hilang.
4. Proses pembuatan permintaan dan pengeluaran material juga laporan pengeluaran material secara langsung dapat diisi ke *database*.

#### **6.2 Saran**

Saran yang dapat diberikan untuk pengembangan sistem informasi penerimaan dan pengeluaran material gudang ke produksi ini selanjutnya adalah sebagai berikut:

1. Pengalihan sistem lama ke sistem yang baru sebaiknya dilakukan secara bertahap dan harus disosialisasikan terlebih dahulu kepada pengguna yang akan menggunakan sistem baru ini.
2. Diharapkan adanya pengembangan lebih lanjut dari sistem informasi yang sudah dirancang. menjadi sistem yang lebih baik dan berkualitas dengan memperbaharui fitur fitur dan melakukan penambahan dan pengurangan.
3. Sebaiknya dilakukan pemeliharaan aplikasi secara berkala, sehingga aplikasi dapat berjalan dengan baik.

## Daftar Pustaka

- Bodnar, George H. dan William S. Hoopwood. 2006, *Sistem Informasi Akuntansi, Edisi 9*. ANDI, Yogyakarta.
- Gordon, B Davis. 1991, *Sistem Informasi Manajemen*. PT. Gramedia, Jakarta.
- Herjanto, Eddi. *Manajemen Operasi Edisi ke*. PT Gramedia Widiasarana Indonesia, Jakarta.
- Kadir, Abdul. 2003, *Pengenalan Sistem Infromasi*. ANDI, Yogyakarta.
- MADCOMS. 2009, *Membangun Sistem Jaringan Komputer*. ANDI, Yogyakarta.
- McLeod, R. J., Scholl, G. 2008, *Sistem Informasi Manajemen Edisi 10*. Salemba Empat, Jakarta.
- Mulyadi. 2001, *Pengertian Sistem Informasi Persediaan*. Salemba Empat, Jakarta.
- Munawar. 2005, *Pemodelan dengan UML*. Graha Ilmu, Yogyakarta.
- O'Brien, James A. 2010, *Introduction to Information Systems*. McGraw-Hill, New York.
- Paryati, Murya Yosef. 2008, *Sistem Informasi*. Ardana Media, Yogyakarta.
- Rizky Dhanta. 2009, *Pengantar Ilmu Komputer*. INDAH, Surabaya.
- Robert K. Leitch dan K. Roscoe Davis, 1983, *Accounting Information Systems*. Prentice-Hall, New Jersey.

- Rosa, S., Shalahuddin, M. 2011, *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Modula, Bandung.
- Sidik, Betha. 2014, *Pemrograman Web Dengan PHP*. Informatika, Bandung.
- Siswoutomo, W. 2005, *PHP Undercover Mengungkap Rahasia Pemograman PHP*. PT Elex Media Komputindo, Jakarta.
- Smith, Jay M. 1992, *Akuntansi Intermediate Volume Komprehensif Edisi Kedelapan*. Erlangga, Jakarta.
- Sutabri, Tata. 2004, *Analisa Sistem Informasi*. Andi Offset, Yogyakarta.
- Sutanta, Edhy. 2003, *Sistem Informasi Manajemen*. Graha Ilmu, Yogyakarta.
- Von Bertalanffy, Ludwig, 1991, *General System Theory*. George Braziller. New York.
- Warman, John. 2004, *Manajemen Pergudangan*. Pustaka Sinar Harapan, Jakarta.
- Widigdo, Anon K. 2003, *Dasar Pemrograman PHP dan MySQL*. Ilmu Komputer.Com, Jakarta.
- \_\_\_\_\_. 2011. *Kamus Besar Bahasa Indonesia*. Jakarta, Balai Pustaka.
- \_\_\_\_\_. 2012. *Siklus Permintaan dan Pengeluaran Barang Gudang*.
- \_\_\_\_\_. 2013, *Arsip PT Tetra Pak Stainless Equipment*. PT Tetra Pak Stainless Equipment, Pulogebang.

[http://ferryantooo.blogspot.com/2011/02/pengertian-penerimaan-jenis\\_jenis.html](http://ferryantooo.blogspot.com/2011/02/pengertian-penerimaan-jenis_jenis.html)  
(tanggal akses 22 February 2016)

[http://www.aksteel.com/pdf/markets\\_products/stainless/austenitic/304\\_304L](http://www.aksteel.com/pdf/markets_products/stainless/austenitic/304_304L)

<http://jimmy-manurung.blogspot.co.id/2012/07/penerimaan-barang.html>  
(tanggal akses 29 september 2016)

[http://www.aksteel.com/pdf/markets\\_products/stainless/austenitic/316\\_316L](http://www.aksteel.com/pdf/markets_products/stainless/austenitic/316_316L)  
(tanggal akses 22 February 2016)

<http://www.nemaenclosures.com/blog/304-and-316-stainless-steel>  
(tanggal akses 5 Maret 2016)

<http://www.tetrapak.com> (tanggal akses 3 January 2016)

## **LAMPIRAN**

## LAMPIRAN A TAMPILAN PROGRAM

### 1. Form *Login*



## 2. Form Menu Utama

PEBAKU
Oka

You Login As admin
✕

### Home

Welcome to Pebaku.

#### LIST REQUEST FOR WAREHOUSE

Show:  Search:

No	Request Number	Request Date	Request By	Status
1	REQ00001	2016-08-23 13:25:21	Staff Produksi	Delivered
2	REQ00002	2016-08-23 14:30:55	Oka	Delivered
3	REQ00003	2016-08-23 12:25:58	Staff Produksi	Delivered
4	REQ00004	2016-08-25 10:50:53	Oka	Delivered
5	REQ00005	2016-08-25 12:45:48	Oka	Delivered
6	REQ00006	2016-08-25 10:50:13	Oka	on process

Showing 1 to 6 of 6 records < 1 >

#### LIST REQUEST FROM PRODUCTION

Show:  Search:

No	Request Number	Request Date	Request By	Status
1	REQ00001	2016-08-23 13:25:21	Staff Produksi	Delivered
2	REQ00002	2016-08-23 14:30:55	Oka	Delivered
3	REQ00003	2016-08-23 12:25:58	Staff Produksi	Delivered
4	REQ00004	2016-08-25 10:50:53	Oka	Delivered
5	REQ00005	2016-08-25 12:45:48	Oka	Delivered

Showing 1 to 5 of 6 records Page < 1 > of 2

### 3. Form *Receipt Materials*

PEBAKU

Home

Warehouse

Production

Purchasing

Suppliers

Master User

Receipt Materials

Home > Reception

MANAGE TABLE

Add New +

Show 5 Search:

No	Receipt Number	Supplier	PIC Supplier	received by	Receipt date	Action
1	D00001	PT. Distrindo Perkasa	Mochamad Fadillah Hadyanto	gudang	2016-08-23 09:45:08	Edit Delete Detail
2	D00002	PT. Indotrans	Rath Endah Pratiwi	gudang	2016-08-23 17:20:04	Edit Delete Detail
3	D00003	PT. Iwatani Industrial Gas Ind	Marco	gudang	2016-08-23 12:30:32	Edit Delete Detail

Showing 1 to 3 of 3 records

### 4. Form Add Receipt Materials

PEBAKU

Home

Warehouse

Production

Purchasing

Suppliers

Master User

Add Reception

Home > Reception > Add Reception

FORM ADD RECEPTION

Transaction Data

Reception Number

Supplier

Date

Detail Material

Code

Add New Material

Expired

Qty

Clear Add

## 5. Form return materials

The screenshot shows the 'Return Materials' management interface. On the left is a dark sidebar with navigation icons for Home, Warehouse, Production, Purchasing, Suppliers, and Master User. The main content area is titled 'Return Materials' and includes a breadcrumb 'Home > Return'. Below the title is a 'MANAGE TABLE' section with an 'Add new +' button, a 'Show' dropdown set to '5', and a search field. A table displays one record with the following data:

No	Return Number	Reception Number	Return date	Action
1	R00001	D00003	2016-08-23	<a href="#">Detail</a>

Below the table, it indicates 'Showing 1 to 1 of 1 records' and includes pagination controls.

## 6. Form add Return Materialsz

The screenshot shows the 'Add Return' form. The sidebar is identical to the previous screenshot. The main content area is titled 'Add Return' with a breadcrumb 'Home > Return > Add Return'. The form is divided into two main sections:

**Return Data**

- Return Number:
- Receipt Number:
- Date of return:

**Material Detail**

- Material Code:
- Qty:
- Buttons:

Below the form is a section titled 'List of temporary Materials' with a dropdown arrow. It contains a table with the following columns:

Code	Name	Qty

At the bottom right of the form are 'Cancel' and 'Submit' buttons.

## 7. Form Master Materials

PEBAKU Oka ▾

Home

Warehouse

Production

Purchasing

Suppliers

Master User

### Master Materials

Home > Materials

**MANAGE TABLE**

[Add new +](#)

Show  Search:

No	Material Code	Material Name	Description	Unit	Action
1	AG0001	Argon Gas High Purity	99,995%	Bottle	<a href="#">Edit</a> <a href="#">Delete</a>
2	AG0002	Argon gas Purity 99,98%	99,98%	bottle	<a href="#">Edit</a> <a href="#">Delete</a>
3	PB304	plat baja 304	T(5MM) LxP(1,524m x 6,096m)	sheet	<a href="#">Edit</a> <a href="#">Delete</a>
4	PB304a	plat baja 304a	T(10mm) LxP(1,524m x 6,096m)	sheet	<a href="#">Edit</a> <a href="#">Delete</a>
5	PB316	Plat Baja 316	T(5mm) LxP(1,524m x 6,096m)	sheet	<a href="#">Edit</a> <a href="#">Delete</a>

## 8. Form Add Materials

PEBAKU Oka ▾

Home

Warehouse

Production

Purchasing

Suppliers

Master User

### Add Materials

Home > Master Materials > Add Materials

**FORM ADD MATERIALS**

Material Code

Material Name

Unit

Description

## 9. Form Request Approvals

**Request Approvals**

Home > Approval

APPROVAL

Show 5 Search:

No	Request Number	Request Date	Request by	Status	Action
1	REQ00001	2016-08-23 13:25:21	Staff Produksi	Delivered	Print
2	REQ00002	2016-08-23 14:30:55	Oka	Delivered	Print
3	REQ00003	2016-08-23 12:25:58	Staff Produksi	Delivered	Print
4	REQ00004	2016-08-25 10:50:53	Oka	Delivered	Print
5	REQ00005	2016-08-25 12:45:48	Oka	Delivered	Print

Showing 1 to 5 of 6 records

## 10. Form Stocks

**Stock & Stock Report**

Home > Stock Report

EMPTY STOCK

10 entries Search:

Code	Name
PB304	plat baja 304

Showing 1 to 1 of 1 entries

LIST STOCK MATERIALS

10 entries Search:

Code	Name	Expired	Not Expired
AG0001	Argon Gas High Purity	0	1
AG0002	Argon gas Purity 99.99%	0	0
PB304	plat baja 304	0	0
PB304a	plat baja 304a	0	0
PB316	Plat Baja 316	0	0
PB316L	Plat Baja 316	0	0
PB310300	pipa besi _10300	0	10
PB315300	Pipa besi 15300	0	0
PB36300	pipa besi 6300	0	0

Showing 1 to 9 of 9 entries

## 11. Form Output Report

PEBAKU Oka ▾

Home

Warehouse

Production

Purchasing

Suppliers

Master User

### Output Report Materials

Home > Output Report

OUTPUT REPORT MATERIALS Print Copy PDF Excel CSV Columns

10 entries Search:

Request Code	Requestor	Request date	Approver	Approval Date	Material Name
REQ00001	Staff Produksi	2016-08-23 13:25:21	Staff Gudang	2016-08-22	1. PB304 plat baja 304 Requested : 5 Approved : 5
REQ00002	Oka	2016-08-23 14:30:55	Staff Gudang	2016-08-23	1. PB304 plat baja 304 Requested : 2 Approved : 2
REQ00003	Staff Produksi	2016-08-23 12:25:58	Staff Gudang	2016-08-23	1. PB304 plat baja 304 Requested : 5 Approved : 3
REQ00004	Oka	2016-08-25 10:50:53	Oka	2016-08-25	1. AG0001 Argon Gas High Purity Requested : 2 Approved : 2
REQ00005	Oka	2016-08-25 12:45:48	Oka	2016-08-25	1. AG0001 Argon Gas High Purity Requested : 1 Approved : 1
REQ00006	Oka	2016-08-25 10:50:13	Oka	2016-08-28	1. AG0001 Argon Gas High Purity Requested : 1 Approved : 1

Showing 1 to 6 of 6 entries < 1 >

## 12. Form Request

PEBAKU Oka ▾

Home

Warehouse

Production

Purchasing

Suppliers

Master User

### Request Materials

Production > Request

MANAGE TABLE

[Add New +](#)

Show 5 Search:

No	Request Number	Request Date	Request by	Status	Action
1	REQ00001	2016-08-23 13:25:21	Staff Produksi	Delivered	<a href="#">Detail</a>
2	REQ00002	2016-08-23 14:30:55	Oka	Delivered	<a href="#">Detail</a>
3	REQ00003	2016-08-23 12:25:58	Staff Produksi	Delivered	<a href="#">Detail</a>
4	REQ00004	2016-08-25 10:50:53	Oka	Delivered	<a href="#">Detail</a>
5	REQ00005	2016-08-25 12:45:48	Oka	Delivered	<a href="#">Detail</a>

### 13. Form Add Request

**PEBAKU** Oka ▾

**Add Request**

Production > Request > Add Request

**FORM ADD REQUEST**

**Data Request**

Request Number: REQ00007

Date:

**Detail Material**

Code:

Qty:

Clear Add

list of temporary materials

Code	Name	Qty

Cancel Request

### 14. Form Supplier

**PEBAKU** Oka ▾

**Suppliers**

Home > Suppliers

**MANAGE TABLE**

Add New +

Show: 5 Search:

No	Company	Address	PIC	Email	Contact	Active	Action
1	PT. Distrindo Perkasa	Jl. Pedalaman Luar	Mochamad Fadillah Hadiyanto	fadillah.hadiyanto@live.com	089634648411	Active	Edit Delete
2	PT. Indotrans	Jl. Rancamaya	Ratih Endah Pratiwi	eggi.pratiwi@gmail.com	089634648410	Active	Edit Delete
3	PT. Iwatani Industrial Gas Ind	Kawasan Industrial KIIC Lot A - 12 Teluk Jembe karawang	Marco	marco.iwatani1234@yahoo.com	082112778851	Active	Edit Delete
4	PT. JOSINTO TEHNIK PERKASA	JLN WARAKAS RAYA V.I. NO 102B. GANG 20. TANJUNG PRIOK JAKARTA UTARA JAKARTA 14340, Jakarta Indonesia	Albert	Albert.josinto@gmail.com	082112345655	Active	Edit Delete

Showing 1 to 4 of 4 records

« < 1 > »

## 15. Form Add Supplier

The screenshot shows the 'Add Supplier' form in the PEBAKU system. The form is titled 'FORM ADD SUPPLIER' and is located in the 'Master Suppliers > Add Supplier' section. The form contains the following fields:

- Name: A text input field.
- Address: A larger text input field.
- PIC: A text input field.
- Email: A text input field with a placeholder 'Alamat Email'.
- Contact: A text input field.
- Active?: A checkbox labeled 'Active'.

At the bottom of the form, there are two buttons: 'Cancel' and 'Submit'.

## 16. Form User

The screenshot shows the 'Manage Users' page in the PEBAKU system. The page is titled 'Manage Users' and is located in the 'Users / Manage Users' section. The page displays a table of users with the following columns:

- Login
- Email
- Grup
- Date Created

A 'New User' button is located in the top right corner of the table. The table contains the following data:

Login	Email	Grup	Date Created
admin	root	Admin	12/02/2008
afriandy	okaafriandy@gmail.com	Admin	19/05/2016
gudang		Gudang	
produksi		Produksi	
purchasing		Purchasing	
rath	eggi.pratiwi@gmail.com	Gudang	17/08/2016

At the bottom of the table, there are navigation arrows: '<' and '>'.

### 17. Form New User

The screenshot shows a web application interface for adding a new user. On the left is a dark sidebar with the logo 'PEBAKU' and a menu with icons for Home, Warehouse, Production, Purchasing, Suppliers, and Master User. The main content area is titled 'Add User' and contains a breadcrumb trail 'Users / Add'. The form fields are: Login (text input), Name (text input), Password (text input), Email (text input), and Group (dropdown menu with 'Admin' selected). At the bottom of the form are two buttons: a blue 'Save' button and a white 'Back' button with a left-pointing arrow.

## LAMPIRAN B

### CODING PROGRAM

#### Localhost\pebaku\_cake\app\Form Index

```

<?php
/**
 * @var $this View
 */
$this->start('opsicss');
echo $this->Html->css('/assets/global/plugins/datatables/datatables.min');
echo $this->Html-
>css('/assets/global/plugins/datatables/plugins/bootstrap/datatables.bootstrap');
$this->end();
$this->start('opsijs');
echo $this->Html->script('/assets/global/scripts/datatable');
echo $this->Html->script('/assets/global/plugins/datatables/datatables.min');
echo $this->Html-
>script('/assets/global/plugins/datatables/plugins/bootstrap/datatables.bootstrap');
echo $this->Html->script('/assets/pages/scripts/table-datatables-managed.min');
$this->end();
?>

<h3 class="page-title">Receipt Materials</h3>
<div class="page-bar">
  <ul class="page-breadcrumb">
    <li>
      <i class="icon-home"></i>
      <?php echo $this->Html->link('Home', array('controller' => 'pages', 'action' =>
'display', 'dashboard')) ?>

```



```

        <th> received by</th>
        <th> Date</th>
        <th> Action</th>
    </tr>
</thead>
<tbody>
    <?php
    $no = 1;
    foreach ($receipt as $row):
        ?>
        <tr class="odd gradeX">
            <td><?php echo $no ?></td>
            <td><?php echo $row['Receipt']['receipt_number'] ?></td>
            <td><?php echo $row['Supplier']['name'] ?></td>
            <td><?php echo $row['Supplier']['pic'] ?></td>
            <td class="center"><?php echo $row['Receiver']['login'] ?></td>
            <td><?php echo $row['Receipt']['receipt_date'] ?></td>
            <td>
                <?php
                echo $this->Html->link('Edit <i class="fa fa-edit"></i>', array('action'
=> 'edit', $row['Receipt']['receipt_number']), array('escape' => false, 'class' => 'btn btn-sm
yellow'));
                echo $this->Html->link('Delete <i class="fa fa-trash"></i>',
array('action' => 'delete', $row['Receipt']['receipt_number']), array('escape' => false,
'class' => 'btn btn-sm red'), 'Are you sure ?');
                echo $this->Html->link('Detail <i class="fa fa-list"></i>', array('action'
=> 'detail', $row['Receipt']['receipt_number']), array('escape' => false, 'class' => 'btn btn-
sm btn-primary'));
            ?>
        </td>
    </tr>
    <?php
    $no++;
endforeach;

```

```

        ?>
    </tbody>
</table>
</div>
</div>
Form Login
<!DOCTYPE html>
<!--[if IE 8]> <html lang="en" class="ie8 no-js"> <![endif]-->
<!--[if IE 9]> <html lang="en" class="ie9 no-js"> <![endif]-->
<!--[if !IE]><!-->
<html lang="en">
<!--<![endif]-->
<!-- BEGIN HEAD -->
<head>
    <meta charset="utf-8"/>
    <title>Pebaku | Login</title>
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta content="width=device-width, initial-scale=1.0" name="viewport"/>
    <meta http-equiv="Content-type" content="text/html; charset=utf-8">
    <meta content="" name="description"/>
    <meta content="" name="author"/>
    <!-- BEGIN GLOBAL MANDATORY STYLES -->
<!-- <link
href="http://fonts.googleapis.com/css?family=Open+Sans:400,300,600,700&subset=all"
rel="stylesheet" type="text/css"/>-->
<?php
echo $this->Html->css(array(
    '/assets/global/plugins/font-awesome/css/font-awesome.min',
    '/assets/global/plugins/simple-line-icons/simple-line-icons.min',
    '/assets/global/plugins/bootstrap/css/bootstrap.min',
    '/assets/global/plugins/uniform/css/uniform.default',
    '/assets/global/plugins/bootstrap-switch/css/bootstrap-switch.min',
    '/assets/global/plugins/select2/css/select2.min',
    '/assets/global/plugins/select2/css/select2-bootstrap.min',

```

```

        '/assets/global/css/components.min',
        '/assets/global/css/plugins.min',
        '/assets/pages/css/login-4.min',
    ));
?>
<!-- END THEME STYLES -->
<?php
echo $this->Html->meta('icon', 'favicon.png');
?>
</head>
<!-- END HEAD -->
<!-- BEGIN BODY -->
<body class="login">
<!-- BEGIN LOGO -->
<div class="logo">
    <?php
    echo $this->Html->image("/assets/layouts/layout2/img/logo-big.png", array(
        "alt" => "logo",
        'class' => "logo-default",
        'url' => array('plugin'=>false,'controller' => 'pages', 'action' => 'display', 'dashboard')
    ));
    ?>
</div>
<!-- END LOGO -->
<!-- BEGIN SIDEBAR TOGGLER BUTTON -->
<div class="menu-toggler sidebar-toggler">
</div>
<!-- END SIDEBAR TOGGLER BUTTON -->
<!-- BEGIN LOGIN -->
<div class="content">
    <!-- BEGIN LOGIN FORM -->
    <?php echo $this->Session->flash() ?>
    <?php echo $this->fetch('content'); ?>
    <!-- END LOGIN FORM -->

```

```

//<!-- BEGIN FORGOT PASSWORD FORM -->
<form class="forget-form" action="index.html" method="post">
  <h3>Forget Password ?</h3>
  <p>
    Enter your e-mail address below to reset your password.
  </p>
  <div class="form-group">
    <div class="input-icon">
      <i class="fa fa-envelope"></i>
      <input class="form-control placeholder-no-fix" type="text" autocomplete="off"
placeholder="Email" name="email"/>
    </div>
  </div>
  <div class="form-actions">
    <button type="button" id="back-btn" class="btn">
      <i class="m-icon-swapleft"></i> Back </button>
    <button type="submit" class="btn blue pull-right">
      Submit <i class="m-icon-swapright m-icon-white"></i>
    </button>
  </div>
</form>
<!-- END FORGOT PASSWORD FORM -->
</div>
<!-- END LOGIN -->
<!-- BEGIN COPYRIGHT -->
<div class="copyright">
  2016 &copy; Oka Afriandy - Politeknik STMI
</div>
<!-- END COPYRIGHT -->
<!-- BEGIN JAVASCRIPTS(Load javascripts at bottom, this will reduce page load time)
-->
<!-- BEGIN CORE PLUGINS -->
<!--[if lt IE 9]>
<?php

```



```

        required: !0
    },
    remember: {
        required: !1
    }
},
messages: {
    username: {
        required: "Username is required."
    },
    password: {
        required: "Password is required."
    }
},
invalidHandler: function(e, r) {
    $(".alert-danger", $(".login-form")).show()
},
highlight: function(e) {
    $(e).closest(".form-group").addClass("has-error")
},
success: function(e) {
    e.closest(".form-group").removeClass("has-error"), e.remove()
},
errorPlacement: function(e, r) {
    e.insertAfter(r.closest(".input-icon"))
},
submitHandler: function(e) {
    e.submit()
}
}), $(".login-form input").keypress(function(e) {
    return 13 == e.which ? $(".login-form").validate().form() && $(".login-
form").submit(), !1) : void 0
})
},

```

```

r = function() {
  $(".forget-form").validate({
    errorElement: "span",
    errorClass: "help-block",
    focusInvalid: !1,
    ignore: "",
    rules: {
      email: {
        required: !0,
        email: !0
      }
    },
    messages: {
      email: {
        required: "Email is required."
      }
    },
    invalidHandler: function(e, r) {},
    highlight: function(e) {
      $(e).closest(".form-group").addClass("has-error")
    },
    success: function(e) {
      e.closest(".form-group").removeClass("has-error"), e.remove()
    },
    errorPlacement: function(e, r) {
      e.insertAfter(r.closest(".input-icon"))
    },
    submitHandler: function(e) {
      e.submit()
    }
  }), $(".forget-form input").keypress(function(e) {
    return 13 == e.which ? ($.validate($(".forget-form").validate().form()) && $(".forget-
form").submit(), !1) : void 0
  }), jQuery("#forget-password").click(function() {

```

```

        jQuery(".login-form").hide(), jQuery(".forget-form").show()
    }, jQuery("#back-btn").click(function() {
        jQuery(".login-form").show(), jQuery(".forget-form").hide()
    })
},
s = function() {
    function e(e) {
        if (!e.id) return e.text;
        var r = $('<span>' + e.text + "</span>");
        return r
    }
    jQuery().select2 && $("#country_list").size() > 0 &&
($("#country_list").select2({
        placeholder: '<i class="fa fa-map-marker"></i>&nbsp;Select a Country',
        templateResult: e,
        templateSelection: e,
        width: "auto",
        escapeMarkup: function(e) {
            return e
        }
    })), $("#country_list").change(function() {
        $(".register-form").validate().element($(this))
    })), $(".register-form").validate({
        errorElement: "span",
        errorClass: "help-block",
        focusInvalid: !1,
        ignore: "",
        rules: {
            fullname: {
                required: !0
            },
        },
        email: {
            required: !0,

```

```
    email: !0
  },
  address: {
    required: !0
  },
  city: {
    required: !0
  },
  country: {
    required: !0
  },
  username: {
    required: !0
  },
  password: {
    required: !0
  },
  rpassword: {
    equalTo: "#register_password"
  },
  tnc: {
    required: !0
  }
},
messages: {
  tnc: {
    required: "Please accept TNC first."
  }
},
invalidHandler: function(e, r) {},
highlight: function(e) {
  $(e).closest(".form-group").addClass("has-error")
},
success: function(e) {
```

```

        e.closest(".form-group").removeClass("has-error"), e.remove()
    },
    errorPlacement: function(e, r) {
        "tnc" == r.attr("name") ? e.insertAfter($("#register_tnc_error")) : 1 ===
r.closest(".input-icon").size() ? e.insertAfter(r.closest(".input-icon")) : e.insertAfter(r
    },
    submitHandler: function(e) {
        e.submit()
    }
}), $(".register-form input").keypress(function(e) {
    return 13 == e.which ? ($("#register-form").validate().form() && $(".register-
form").submit(), !1) : void 0
}), jQuery("#register-btn").click(function() {
    jQuery(".login-form").hide(), jQuery(".register-form").show()
}), jQuery("#register-back-btn").click(function() {
    jQuery(".login-form").show(), jQuery(".register-form").hide()
})
});
return {
    init: function() {
        e(), r(), s(), $.backstretch(["<?=Router::url('/',
true);?>/assets/pages/media/bg/1.jpg", "<?=Router::url('/',
true);?>/assets/pages/media/bg/2.jpg", "<?=Router::url('/',
true);?>/assets/pages/media/bg/3.jpg", "<?=Router::url('/',
true);?>/assets/pages/media/bg/4.jpg"], {
            fade: 1e3,
            duration: 8e3
        })
    }
}
}
jQuery(document).ready(function() {
    Login.init()
});

```

```

</script>
<!-- END JAVASCRIPTS -->
</body>
<!-- END BODY -->
</html>

```

## Localhost\pebaku\_cake\app\Login

```

<?php echo $this->Form->create(null, array('url' => array('controller' => 'user', 'action' =>
'login'), 'class' => 'login-form')); ?>
<h3 class="form-title">Please Login</h3>
<div class="alert alert-danger display-hide">
    <button class="close" data-close="alert"></button>
    <span>
        Enter the another user name and password. </span>
</div>
<div class="form-group">
    <!--ie8, ie9 does not support html5 placeholder, so we just show field title for that-->
    <label class="control-label visible-ie8 visible-ie9">Username</label>
    <div class="input-icon">
        <i class="fa fa-user"></i>
        <?php echo $this->Form->input('login', array('div' => false, 'label' => false,
'placeholder' => 'Username', 'class' => 'form-control placeholder-no-fix', 'autocomplete'
=> 'off')); ?>
    </div>
</div>
<div class="form-group">
    <label class="control-label visible-ie8 visible-ie9">Password</label>
    <div class="input-icon">
        <i class="fa fa-lock"></i>
        <?php echo $this->Form->input('password', array('div' => false, 'label' => false,
'placeholder' => 'Password', 'class' => 'form-control placeholder-no-fix', 'autocomplete'
=> 'off', 'value' => '')); ?>
    </div>

```

```

</div>
<div class="form-actions">
  <!--<label class="checkbox">
  <input type="checkbox" name="remember" value="1"/> Remember me </label-->
  <?php echo $this->Form->button("Login <i class='m-icon-swapright m-icon-
white'></i>", array('type' => 'submit', 'div' => false, 'class' => 'btn blue pull-right',
'escape' => false)); ?>
</div>
<?<div class="forget-password">
  <h4></h4>
  <p>
    <a href="mailto:okaafriandy@gmail.com" id="forgot-password">
      </a>
    .
  </p>
</div?>
<?php echo $this->Form->end(); ?>

```

### Localhost\pebaku\_cake\app\Form Receipt

```

<?php
App::uses('AppModel', 'Model');
class Request extends AppModel {
  public $actAs = array('Containable');
  public $useTable = 'request';
  public $primaryKey = 'request_number';
  public $hasMany = array(
    'RequestDetail' => array(
      'className' => 'RequestDetail',
      'foreignKey' => 'request_number',
    )
  );
  public $belongsTo = array(
    'Requestor' => array(
      'className' => 'Authake.User',

```

```

        'foreignKey' => 'request_by'
    ),
    'Approver' => array(
        'className' => 'Authake.User',
        'foreignKey' => 'approve_by'
    ),
);

}

```

### Localhost\pebaku\_cake\receipts\add

```

<?php
/**
 * @var $this View
 */
$this->start('opsicss');
echo $this->Html->css(
    array(
        '/assets/global/plugins/select2/css/select2.min',
        '/assets/global/plugins/select2/css/select2-bootstrap.min',
        '/assets/global/plugins/bootstrap-datepicker/css/bootstrap-datepicker3.min',
        '/assets/global/plugins/bootstrap-timepicker/css/bootstrap-timepicker.min',
        '/assets/global/plugins/bootstrap-datetimepicker/css/bootstrap-datetimepicker.min'
    ));
$this->end();
$this->start('opsijs');
echo $this->Html->script(
    array(
        '/assets/global/plugins/select2/js/select2.full.min',
        '/assets/pages/scripts/components-select2.min',
        '/assets/global/plugins/moment.min',
        '/assets/global/plugins/bootstrap-datetimepicker/js/bootstrap-datetimepicker.min',
        '/assets/pages/scripts/components-date-time-pickers.min'
    ));

```

```

?>
<script>
function addMaterial() {
    var dataDari = [];
    dataDari['materialCode'] = $("#ReceiptMaterialCode").val();
    dataDari['expired'] = $("#ReceiptExpired").val();
    dataDari['capacity'] = $("#ReceiptCapacity").val();
    dataDari['qty'] = $("#ReceiptQty").val();
    var dataFinal = {
        material_code: dataDari['materialCode'],
        expired: dataDari['expired'],
        capacity: dataDari['capacity'],
        qty: dataDari['qty']
    };
    $.ajax({
        type: 'POST',
        url: '<?php echo Router::url(array('controller' => 'receipts', 'action' =>
'addMaterial')); ?>',
        data: dataFinal,
        success: function (data) {
            addrowTemp(data);
            $("#ReceiptMaterialCode").select2("val", "");
            $("#ReceiptExpired").val("");
            $("#ReceiptCapacity").val("");
            $("#ReceiptQty").val("");
        },
        error: function (error) {
            console.log(error);
        }
    });
}

function addrowTemp(data) {
    var tempdata = JSON.parse(data);
    var tempMat = "";

```

```

tempMat += "<tr id=\" + tempdata['Material']['material_code'] + \">";
tempMat += " <td class=\"highlight\">";
tempMat += " <div class=\"success\"></div>";
tempMat += " <a id=\"mcode\" href=\"javascript:;\">\" +
tempdata['Material']['material_code'] + "</a>";
tempMat += " </td>";
tempMat += " <td id=\"mname\" class=\"hidden-xs\">\" +
tempdata['Material']['name'] + "</td>";
tempMat += " <td id=\"mcapacity\" class=\"\">\" + tempdata['expired'] + "</td>";
tempMat += " <td id=\"mcapacity\" class=\"hidden\">\" + tempdata['capacity'] +
"</td>";
tempMat += " <td id=\"mqty\" >\" + tempdata['qty'] + "</td>";
tempMat += " <td>";
tempMat += " <a class=\"btn btn-outline btn-circle dark btn-sm black\"
href=\"javascript:delMat(\" + tempdata['Material']['material_code'] + \");\">";
tempMat += " <i class=\"fa fa-trash-o\"></i> Delete </a>";
tempMat += " </td>";
tempMat += "</tr>";
$('#material-table').append(tempMat);
}
function delMat(idx) {
$.ajax({
type: 'POST',
url: '<?php echo Router::url(array('controller' => 'receipts', 'action' =>
'remMaterial')); ?>',
data: {index: idx},
success: function (data) {
$('#tr#' + idx).remove()
},
error: function (error) {
console.log(error);
}
});
}
}

```

```

</script>
<?php
$this->end();
?>
<h3 class="page-title">Add Reception</h3>
<div class="page-bar">
  <ul class="page-breadcrumb">
    <li>
      <i class="icon-home"></i>
      <?php echo $this->Html->link('Home', array('controller' => 'pages', 'action' =>
'display', 'dashboard')) ?>
      <i class="fa fa-angle-right"></i>
    </li>
    <li>
      <?php echo $this->Html->link('Reception', array('action' => 'index')) ?>
      <i class="fa fa-angle-right"></i>
    </li>
    <li>
      <span>Add Reception</span>
    </li>
  </ul>
</div>
<div class="portlet light">
  <div class="portlet-title">
    <div class="caption">
      <span class="caption-subject font-green-haze bold uppercase">Form Add
Reception </span>
    </div>
    <div class="tools">
      <a href="javascript:;" class="collapse"> </a>
    </div>
  </div>
  <div class="portlet-body form">
    <?php echo $this->Form->create('Receipt', array('class' => 'form-horizontal')) ?>

```

```

<div class="form-body">
  <h3 style="margin-top:0;">Transaction Data</h3>
  <div class="form-group">
    <label class="col-md-3 control-label">Reception Number</label>
    <div class="col-md-4">
      <?php echo $this->Form->input('receipt_number', array('type' => 'text',
'class' => 'form-control', 'placeholder' => 'Enter Text', 'label' => false)) ?>
    </div>
  </div>
  <div class="form-group">
    <label for="single" class="col-md-3 control-label">Supplier</label>
    <div class="col-md-4">
      <?php echo $this->Form->input('supplier_id', array('options' => $supplier,
'empty' => array(0 => ''), 'class' => 'form-control select2', 'label' => false)) ?>
    </div>
  </div>
  <div class="form-group">
    <label class="col-md-3 control-label">Date</label>
    <div class="col-md-4">
      <div class="input-group date form_datetime">
        <?php echo $this->Form->input('receipt_date', array('type' => 'text', 'class'
=> 'form-control', 'readonly' => 'readonly', 'label' => false)) ?>
        <span class="input-group-btn">
          <button class="btn default date-set" type="button">
            <i class="fa fa-calendar"></i>
          </button>
        </span>
      </div>
    </div>
  </div>
  <br>
  <h3>Detail Material</h3>
  <div class="form-group">
    <label for="single" class="col-md-3 control-label">Code</label>

```

```

<div class="col-md-4">
    <?php echo $this->Form->input('material_code', array('options' =>
$material, 'empty' => array(0 => ""), 'class' => 'form-control select2', 'label' => false)) ?>
    <?php echo $this->Html->link('Add New Material', array('controller' =>
'materials', 'action' => 'add'), array('class' => 'btn btn-info btn-sm')); ?>
</div>
</div>
<div class="form-group">
    <label class="col-md-3 control-label">Expired</label>
    <div class="col-md-4">
        <div class="input-group date form_datetime">
            <?php echo $this->Form->input('expired', array('type' => 'text', 'class' =>
'form-control', 'readonly' => 'readonly', 'label' => false)) ?>
            <span class="input-group-btn">
                <button class="btn default date-set" type="button">
                    <i class="fa fa-calendar"></i>
                </button>
            </span>
        </div>
    </div>
</div>
<div class="form-group hidden">
    <label class="col-md-3 control-label">Capacity</label>
    <div class="col-md-4">
        <?php echo $this->Form->input('capacity', array('class' => 'form-control',
'placeholder' => 'Enter Text', 'label' => false)) ?>
    </div>
</div>
<div class="form-group">
    <label class="col-md-3 control-label">Qty</label>
    <div class="col-md-4">
        <?php echo $this->Form->input('qty', array('class' => 'form-control',
'placeholder' => 'Enter Text', 'label' => false)) ?>
    </div>

```

```

</div>
<div class="col-md-offset-3 col-md-3">
  <button type="reset" class="btn default">Clear</button>
  <button type="button" onclick="addMaterial()" class="btn
green">Add</button>
</div>
<div class="row" style="padding-top:80px;">
  <div class="col-md-offset-1 col-md-8">
    <!-- BEGIN SAMPLE TABLE PORTLET-->
    <div class="portlet">
      <div class="portlet-title">
        <div class="caption">
          <i class="fa fa-shopping-cart"></i>List of temporary Materials
        </div>
        <div class="tools">
          <a href="javascript:;" class="collapse"> </a>
        </div>
      </div>
      <div class="portlet-body">
        <div class="table-scrollable">
          <table id="material-table"
            class="table table-striped table-bordered table-advance table-
hover">
            <thead>
              <tr>
                <th>Code</th>
                <th class="hidden-xs">Name</th>
                <th class="">Expired</th>
                <th class="hidden">Capacity</th>
                <th>Qty</th>
                <th></th>
              </tr>
            </thead>
            <tbody>

```

```

<?php foreach ($viewTemp as $sesMat): ?>
    <tr id="<?php echo $sesMat['Material']['material_code'] ?>">
        <td class="highlight">
            <div class="success"></div>
            <a href="javascript:;" id="mcode"> <?php echo
$sesMat['Material']['material_code'] ?> </a>
        </td>
        <td id="mname" class="hidden-xs"> <?php echo
$sesMat['Material']['name'] ?></td>
        <td id="mname" class="hidden-xs"> <?php echo
$sesMat['expired'] ?></td>
        <td id="mcapacity" class="hidden"> <?php echo
$sesMat['capacity'] ?></td>
        <td id="mqty"> <?php echo $sesMat['qty'] ?></td>
        <td>
            <a href="javascript:delMat('<?php echo
$sesMat['Material']['material_code'] ?>');"
                class="btn btn-outline btn-circle dark btn-sm black">
                <i class="fa fa-trash-o"></i> Hapus </a>
        </td>
    </tr>
</tbody>
</table>
</div>
</div>
</div>
<!-- END SAMPLE TABLE PORTLET-->
</div>
</div>
<div class="form-actions">
    <div class="row">
        <div class="col-md-offset-9 col-md-3">

```

```

        <button type="button" class="btn default"
onclick=self.history.back()>Cancel</button>
        <button type="submit" class="btn green">Save</button>
    </div>
</div>
</div>
<?php echo $this->Form->end(); ?>
</div>
</div>

```

### Localhost/pebaku\_cake/receipts/detail

```

<?php
$data = $this->request->data;
?>
<h3 class="page-title">Reception Detail</h3>
<div class="page-bar">
    <ul class="page-breadcrumb">
        <li>
            <i class="icon-home"></i>
            <?php echo $this->Html->link('Home', array('controller' => 'pages', 'action' =>
'display', 'dashboard')) ?>
            <i class="fa fa-angle-right"></i>
        </li>
        <li>
            <?php echo $this->Html->link('Reception', array('action' => 'index')) ?>
            <i class="fa fa-angle-right"></i>
        </li>
        <li>
            <span>Reception Detail</span>
        </li>
    </ul>
</div>
<div class="portlet light">
    <div class="portlet-title">

```

```

<div class="caption">
  <span class="caption-subject font-green-haze bold uppercase">Reception
Detail</span>
</div>
<div class="tools">
  <a href="javascript:;" class="collapse"> </a>
</div>
</div>
<div class="portlet-body form">
  <div class="form-body">
    <h3 style="margin-top:0;">Transaction Data</h3>
    <table class="table">
      <tr>
        <td>Reception Number</td>
        <td>:</td>
        <td><?php echo $data['Receipt']['receipt_number'] ?></td>
      </tr>
      <tr>
        <td>Supplier</td>
        <td>:</td>
        <td><?php echo $data['Supplier']['name'] ?></td>
      </tr>
      <tr>
        <td>Date</td>
        <td>:</td>
        <td><?php echo $data['Receipt']['receipt_date'] ?></td>
      </tr>
      <tr>
        <td>Received by</td>
        <td>:</td>
        <td><?php echo $data['Receiver']['name'] ?></td>
      </tr>
    </table>
    <br>

```

```

<!-- BEGIN SAMPLE TABLE PORTLET-->
<div class="portlet">
  <div class="portlet-title">
    <div class="caption">
      <h3>List of Material</h3>
    </div>
  </div>
  <div class="portlet-body">
    <div class="table-scrollable">
      <table id="material-table"
        class="table table-striped table-bordered table-advance table-hover">
        <thead>
          <tr>
            <th>Code</th>
            <th class="hidden-xs">Name</th>
            <th class="hidden-xs">Expired</th>
            <th>Qty</th>
          </tr>
        </thead>
        <tbody>
          <?php foreach ($data['ReceiptDetail'] as $key => $sesMat): ?>
            <tr id="<?php echo $sesMat['receipt_number'] ?>">
              <td class="highlight">
                <div class="success"></div>
                <a href="javascript:;"
                  id="mcode"> <?php echo
$sesMat['Material']['material_code'] ?> </a>
              </td>
              <td id="mname"
                class="hidden-xs"> <?php echo $sesMat['Material']['name']
?></td>
              <td id="mexpired"
                class="hidden-xs"> <?php echo
$data['Stock'][$key]['expired'] ?></td>

```

```

        <td id="mcapacity" class="hidden"> <?php echo $sesMat['capacity'] ?></td>
            <td id="mqty"> <?php echo $sesMat['qty'] ?></td>
        </tr>
    <?php endforeach; ?>
</tbody>
</table>
</div>
</div>
</div>
<!-- END SAMPLE TABLE PORTLET-->
</div>
<!-- END SAMPLE TABLE PORTLET-->
<div class="col-md-offset-11 col-md-3">
    <button type="button" class="btn
green"onclick=self.history.back()>Back</button>
</div>
</div>
</div>
</div>

```

### **localhost/pebaku\_cake/returns**

```

<?php
/**
 * @var $this View
 */
$this->start('opsicss');
echo $this->Html->css('/assets/global/plugins/datatables/datatables.min');
echo $this->Html-
>css('/assets/global/plugins/datatables/plugins/bootstrap/datatables.bootstrap');
$this->end();
$this->start('opsijs');
echo $this->Html->script('/assets/global/scripts/datatable');
echo $this->Html->script('/assets/global/plugins/datatables/datatables.min');

```

```

echo $this->Html-
>script('/assets/global/plugins/datatables/plugins/bootstrap/datatables.bootstrap');
echo $this->Html->script('/assets/pages/scripts/table-datatables-managed.min');
$this->end();
?>
<h3 class="page-title">Return Materials</h3>
<div class="page-bar">
  <ul class="page-breadcrumb">
    <li>
      <i class="icon-home"></i>
      <?php echo $this->Html->link('Home', array('controller' => 'pages', 'action' =>
'display', 'dashboard')) ?>
      <i class="fa fa-angle-right"></i>
    </li>
    <li>
      <span>Return</span>
    </li>
  </ul>
</div>
<div class="portlet light ">
  <div class="portlet-title">
    <div class="caption font-dark">
      <span class="caption-subject bold uppercase"> Manage table</span>
    </div>
  </div>
  <div class="portlet-body">
    <div class="table-toolbar">
      <div class="row">
        <div class="col-md-6">
          <div class="btn-group">
            <?php
              echo $this->Html->link("Add new <i class='fa fa-plus'></i>",
array('action' => 'add'), array('escape' => false, 'class' => 'btn sbold green'));
            ?>

```

```

        </div>
    </div>
</div>
</div>
<table class="table table-striped table-bordered table-hover table-checkable order-
column" id="sample_1">
    <thead>
        <tr>
            <th>No</th>
            <th> Return Number</th>
            <th> Reception Number</th>
            <th> Return date</th>
            <th> Action</th>
        </tr>
    </thead>
    <tbody>
        <?php
        $no = 1;
        foreach ($return as $row):
            ?>
            <tr class="odd gradeX">
                <td><?php echo $no ?></td>
                <td><?php echo $row['Returns']['return_number'] ?></td>
                <td><?php echo $row['Receipt']['receipt_number'] ?></td>
                <td><?php echo $row['Returns']['return_date'] ?></td>
                <td>
                    <?php
//            echo $this->Html->link('Edit <i class="fa fa-edit"></i>', array('action'
=> 'edit', $row['Returns']['return_number']), array('escape' => false, 'class' => 'btn btn-sm
yellow'));
//            echo $this->Html->link('Delete <i class="fa fa-trash"></i>',
array('action' => 'delete', $row['Returns']['return_number']), array('escape' => false, 'class'
=> 'btn btn-sm red'), 'Are you Sure ?');

```

```

        echo $this->Html->link('Detail <i class="fa fa-list"></i>', array('action'
=> 'detail', $row['Returns']['return_number']), array('escape' => false, 'class' => 'btn btn-
sm btn-primary'));
        ?>
    </td>
</tr>
<?php
    $no++;
endforeach;
?>
</tbody>
</table>
</div>
</div>

```

### **localhost/pebaku\_cake/returns/add**

```

<?php
/**
 * @var $this View
 */
$this->start('opsicss');
echo $this->Html->css(
    array(
        '/assets/global/plugins/select2/css/select2.min',
        '/assets/global/plugins/select2/css/select2-bootstrap.min',
        '/assets/global/plugins/bootstrap-datepicker/css/bootstrap-datepicker3.min',
        '/assets/global/plugins/bootstrap-timepicker/css/bootstrap-timepicker.min',
        '/assets/global/plugins/bootstrap-datetimepicker/css/bootstrap-datetimepicker.min'
    ));
$this->end();
$this->start('opsijs');
echo $this->Html->script(
    array(
        '/assets/global/plugins/select2/js/select2.full.min',

```

```

'/assets/pages/scripts/components-select2.min',
'/assets/global/plugins/moment.min',
'/assets/global/plugins/bootstrap-datetimepicker/js/bootstrap-datetimepicker.min',
'/assets/pages/scripts/components-date-time-pickers.min'
));
?>
<script>
function addMaterial() {
    var dataDari = [];
    dataDari['materialCode'] = $("#ReturnsMaterialCode").val();
    dataDari['qty'] = $("#ReturnsQty").val();
    var dataFinal = {
        material_code: dataDari['materialCode'],
        qty: dataDari['qty']
    };
    $.ajax({
        type: 'POST',
        url: '<?php echo Router::url(array('controller' => 'returns', 'action' =>
'addMaterial')); ?>',
        data: dataFinal,
        success: function (data) {
            addrowTemp(data);
            $("#ReturnsMaterialId").select2("val", "");
            $("#ReturnsQty").val("");
        },
        error: function (error) {
            console.log(error);
        }
    });
}
function addrowTemp(data) {
    var tempdata = JSON.parse(data);
    var tempMat = "";
    tempMat += "<tr id=\" + tempdata['Material']['material_code'] + \"\>";

```

```

tempMat += " <td class=\"highlight\">";
tempMat += " <div class=\"success\"></div>";
tempMat += " <a id=\"mcode\" href=\"javascript:;\">\" +
tempdata['Material']['material_code'] + "</a>";
tempMat += " </td>";
tempMat += " <td id=\"mname\" class=\"hidden-xs\">\" +
tempdata['Material']['name'] + "</td>";
tempMat += " <td id=\"mqty\" >\" + tempdata['qty'] + "</td>";
tempMat += " <td>";
tempMat += " <a class=\"btn btn-outline btn-circle dark btn-sm black\"
href=\"javascript:delMat('\" + tempdata['Material']['material_code'] + '\");\">";
tempMat += " <i class=\"fa fa-trash-o\"></i> Delete </a>";
tempMat += " </td>";
tempMat += "</tr>";
$('#material-table').append(tempMat);
}
function delMat(idx) {
$.ajax({
type: 'POST',
url: '<?php echo Router::url(array('controller' => 'returns', 'action' =>
'remMaterial')); ?>',
data: {index: idx},
success: function (data) {
$('#tr#' + idx).remove()
},
error: function (error) {
console.log(error);
}
});
}

function cekMaterial() {
}
</script>

```

```

<?php
$this->end();
?>
<h3 class="page-title">Add Return</h3>
<div class="page-bar">
  <ul class="page-breadcrumb">
    <li>
      <i class="icon-home"></i>
      <?php echo $this->Html->link('Home', array('controller' => 'pages', 'action' =>
'display', 'dashboard')) ?>
      <i class="fa fa-angle-right"></i>
    </li>
    <li>
      <?php echo $this->Html->link('Return', array('action' => 'index')) ?>
      <i class="fa fa-angle-right"></i>
    </li>
    <li>
      <span>Add Return</span>
    </li>
  </ul>
</div>
<div class="portlet light">
  <div class="portlet-title">
    <div class="caption">
      <span class="caption-subject font-green-haze bold uppercase">Form Add
Return</span>
    </div>
    <div class="tools">
      <a href="javascript:;" class="collapse"> </a>
    </div>
  </div>
  <div class="portlet-body form">
    <?php echo $this->Form->create('Returns', array('class' => 'form-horizontal')) ?>
    <div class="form-body">

```

```

<h3 style="margin-top:0;">Return Data</h3>
<div class="form-group">
  <label class="col-md-3 control-label">Return Number</label>
  <div class="col-md-4">
    <?php echo $this->Form->input('return_number', array('type' => 'text', 'class'
=> 'form-control', 'placeholder' => 'Enter Text', 'label' => false)) ?>
  </div>
</div>
<div class="form-group">
  <label for="single" class="col-md-3 control-label">Receipt Number</label>
  <div class="col-md-4">
    <?php echo $this->Form->input('receipt_number', array('onchange' =>
'cekMaterial', 'options' => $receipt, 'empty' => array(0 => ''), 'class' => 'form-control
select2', 'label' => false)) ?>
  </div>
</div>
<div class="form-group">
  <label class="col-md-3 control-label">Date of return</label>
  <div class="col-md-4">
    <div class="input-group date form_datetime">
      <?php echo $this->Form->input('return_date', array('type' => 'text', 'class'
=> 'form-control', 'readonly' => 'readonly', 'label' => false)) ?>
      <span class="input-group-btn">
        <button class="btn default date-set" type="button">
          <i class="fa fa-calendar"></i>
        </button>
      </span>
    </div>
  </div>
</div>
<br>
<h3>Material Detail</h3>
<div class="form-group">
  <label for="single" class="col-md-3 control-label">Material Code</label>

```

```

        <div class="col-md-4">
            <?php echo $this->Form->input('material_code', array('options' =>
$material, 'empty' => array(0 => ""), 'class' => 'form-control select2', 'label' => false)) ?>
        </div>
    </div>
    <div class="form-group">
        <label class="col-md-3 control-label">Qty</label>
        <div class="col-md-4">
            <?php echo $this->Form->input('qty', array('class' => 'form-control',
'placeholder' => 'Enter Text', 'label' => false)) ?>
        </div>
    </div>
    <div class="col-md-offset-3 col-md-3">
        <button type="reset" class="btn default">Clear</button>
        <button type="button" onclick="addMaterial()" class="btn
green">Add</button>
    </div>
    <div class="row" style="padding-top:80px;">
        <div class="col-md-offset-1 col-md-8">
            <!-- BEGIN SAMPLE TABLE PORTLET-->
            <div class="portlet">
                <div class="portlet-title">
                    <div class="caption">
                        <i class="fa fa-shopping-cart"></i>List of temporary Materials
                    </div>
                    <div class="tools">
                        <a href="javascript:;" class="collapse"> </a>
                    </div>
                </div>
                <div class="portlet-body">
                    <div class="table-scrollable">
                        <table id="material-table"
                            class="table table-striped table-bordered table-advance table-
hover">

```

```

<thead>
  <tr>
    <th>Code</th>
    <th class="hidden-xs">Name</th>
    <th>Qty</th>
    <th></th>
  </tr>
</thead>
<tbody>
  <?php foreach ($viewTemp as $sesMat): ?>
    <tr id="<?php echo $sesMat['Material']['material_code'] ?>">
      <td class="highlight">
        <div class="success"></div>
        <a href="javascript:;" id="mcode"> <?php echo
$sesMat['Material']['material_code'] ?> </a>
      </td>
      <td id="mname" class="hidden-xs"> <?php echo
$sesMat['Material']['name'] ?></td>
      <td id="mqty"> <?php echo $sesMat['qty'] ?></td>
      <td>
        <a href="javascript:delMat('<?php echo
$sesMat['Material']['material_code'] ?>');"
          class="btn btn-outline btn-circle dark btn-sm black">
          <i class="fa fa-trash-o"></i> Delete </a>
      </td>
    </tr>
  <?php endforeach; ?>
</tbody>
</table>
</div>
</div>
</div>
<!-- END SAMPLE TABLE PORTLET-->
</div>

```

```

        </div>
    </div>
    <div class="form-actions">
        <div class="row">
            <div class="col-md-offset-9 col-md-3">
                <button type="button" class="btn default"
onclick=self.history.back()>Cancel</button>
                <button type="submit" class="btn green">Submit</button>
            </div>
        </div>
    </div>
    <?php echo $this->Form->end(); ?>
</div>
</div>

```

### localhost/pebaku\_cake/returns/detail

```

<?php
$data = $this->request->data;
?>
<h3 class="page-title">Retur Detail </h3>
<div class="page-bar">
    <ul class="page-breadcrumb">
        <li>
            <i class="icon-home"></i>
            <?php echo $this->Html->link('Home', array('controller' => 'pages', 'action' =>
'display', 'dashboard')) ?>
            <i class="fa fa-angle-right"></i>
        </li>
        <li>
            <?php echo $this->Html->link('Retur', array('action' => 'index')) ?>
            <i class="fa fa-angle-right"></i>
        </li>
        <li>
            <span>Retur Details</span>

```

```

    </li>
  </ul>
</div>
<div class="portlet light">
  <div class="portlet-title">
    <div class="caption">
      <span class="caption-subject font-green-haze bold uppercase">Retur
Detail</span>
    </div>
    <div class="tools">
      <a href="javascript:;" class="collapse"> </a>
    </div>
  </div>
  <div class="portlet-body form">
    <div class="form-body">
      <h3 style="margin-top:0;">Data Transaction</h3>
      <table class="table">
        <tr>
          <td>Return Number</td>
          <td>:</td>
          <td><?php echo $data['Returns']['return_number'] ?></td>
        </tr>
        <tr>
          <td>Receipt Number</td>
          <td>:</td>
          <td><?php echo $data['Receipt']['receipt_number'] ?></td>
        </tr>
        <tr>
          <td>Date</td>
          <td>:</td>
          <td><?php echo $data['Returns']['return_date'] ?></td>
        </tr>
      </table>
      <br>

```

```

<!-- BEGIN SAMPLE TABLE PORTLET-->
<div class="portlet">
  <div class="portlet-title">
    <div class="caption">
      <h3>List of Material</h3>
    </div>
  </div>
  <div class="portlet-body">
    <div class="table-scrollable">
      <table id="material-table"
        class="table table-striped table-bordered table-advance table-hover">
        <thead>
          <tr>
            <th>Code</th>
            <th class="hidden-xs">Name</th>
            <th>Qty</th>
          </tr>
        </thead>
        <tbody>
          <?php foreach ($data['ReturnDetail'] as $sesMat): ?>
            <tr id="<?php echo $sesMat['return_number'] ?>">
              <td class="highlight">
                <div class="success"></div>
                <a href="javascript:;"
                  id="mcode"> <?php echo
$sesMat['Material']['material_code'] ?> </a>
              </td>
              <td id="mname"
                class="hidden-xs"> <?php echo $sesMat['Material']['name']
?></td>
              <td id="mqty"> <?php echo $sesMat['qty'] ?></td>
            </tr>
          <?php endforeach; ?>
        </tbody>
      </table>
    </div>
  </div>
</div>

```

```

        </table>
    </div>
</div>
</div>
<!-- END SAMPLE TABLE PORTLET-->
</div>
</div>
</div>
<div class="row">
    <div class="col-sm-12">
        <button type="button" class="btn
green"onclick=self.history.back()>Back</button>z    <?php echo $this->Html-
>link('Print <i class="fa fa-check-square-o"></i>', array('action' => 'print_document',
$data['Returns']['return_number']), array('escape' => false, 'class' => 'btn btn-lg blue pull-
right', 'target' => '_blank')); ?>
        <div class="col-md-offset-11 col-md-3">
            <br>

        </div>
    </div>
</div>

```

### Localhost/pebaku\_cake/request

```

<?php
/**
 * @var $this View
 */
$this->start('opsicss');
echo $this->Html->css('/assets/global/plugins/datatables/datatables.min');
echo $this->Html-
>css('/assets/global/plugins/datatables/plugins/bootstrap/datatables.bootstrap');
$this->end();

$this->start('opsijs');
echo $this->Html->script('/assets/global/scripts/datatable');

```

```

echo $this->Html->script('/assets/global/plugins/datatables/datatables.min');
echo $this->Html-
>script('/assets/global/plugins/datatables/plugins/bootstrap/datatables.bootstrap');
echo $this->Html->script('/assets/pages/scripts/table-datatables-managed.min');
$this->end();
?>
<h3 class="page-title">Request Materials</h3>
<div class="page-bar">
  <ul class="page-breadcrumb">
    <li>
      <i class="icon-home"></i>
      <?php echo $this->Html->link('Production', array('controller' => 'pages', 'action'
=> 'display', 'dashboard')) ?>
      <i class="fa fa-angle-right"></i>
    </li>
    <li>
      <span>Request</span>
    </li>
  </ul>
</div>
<div class="portlet light ">
  <div class="portlet-title">
    <div class="caption font-dark">
      <span class="caption-subject bold uppercase"> Manage Table</span>
    </div>
  </div>
  <div class="portlet-body">
    <div class="table-toolbar">
      <div class="row">
        <div class="col-md-6">
          <div class="btn-group">
            <?php
              echo $this->Html->link("Add New <i class='fa fa-plus'></i>",
array('action' => 'add'), array('escape' => false, 'class' => 'btn sbold green'));

```

```

        ?>
    </div>
</div>
</div>
</div>
<table class="table table-striped table-bordered table-hover table-checkable order-
column" id="sample_1">
    <thead>
        <tr>
            <th>No</th>
            <th> Request Number</th>
            <th> Request Date</th>
            <th> Request by</th>
            <th> Status</th>
            <th> Action</th>
        </tr>
    </thead>
    <tbody>
        <?php
        $no = 1;
        foreach ($request as $row):
            $textStatus = array(-1 => 'Reject', 0 => 'Pending', 1 => 'On proces', 2 =>
'Delivered');
            $labelStatus = array(-1 => 'label-danger', 0 => 'label-default', 1 => 'label-
warning', 2 => 'label-primary');
            ?>
            <tr class="odd gradeX">
                <td><?php echo $no ?></td>
                <td><?php echo $row['Request']['request_number'] ?></td>
                <td><?php echo $row['Request']['request_date'] ?></td>
                <td><?php echo $row['Requestor']['name'] ?></td>
                <td><label
                    class="label <?php echo $labelStatus[$row['Request']['status']]
?>"><?php echo $textStatus[$row['Request']['status']] ?></label>

```

```

        </td>
        <td>
            <?php
                //          echo $this->Html->link('Ubah <i class="fa fa-
edit"></i>', array('action' => 'edit', $row['Request']['id']), array('escape' => false, 'class'
=> 'btn btn-sm yellow'));
                //          echo $this->Html->link('Hapus <i class="fa fa-
trash"></i>', array('action' => 'delete', $row['Request']['id']), array('escape' => false, 'class'
=> 'btn btn-sm red'), 'Apa Anda yakin?');
                if ($row['Request']['status'] == 0) {
                    echo $this->Html->link('Cancel <i class="fa fa-trash"></i>',
array('action' => 'cancel', $row['Request']['request_number']), array('escape' => false,
'class' => 'btn btn-sm red'), 'Apa Anda yakin?');
                } elseif ($row['Request']['status'] == 1) {
                    echo $this->Html->link('Approve <i class="fa fa-check"></i>',
array('action' => 'approve', $row['Request']['request_number']), array('escape' => false,
'class' => 'btn btn-sm green'), 'Are you sure?');
                }
                echo $this->Html->link('Detail <i class="fa fa-list"></i>', array('action'
=> 'detail', $row['Request']['request_number']), array('escape' => false, 'class' => 'btn btn-
sm btn-primary'));
            ?>
        </td>
    </tr>
    <?php
        $no++;
    endforeach;
    ?>
</tbody>
</table>
</div>
</div>

```

**Localhost/pebaku\_cake/request/add**

```

<?php
/**
 * @var $this View
 */
$this->start('opsicss');
echo $this->Html->css(
    array(
        '/assets/global/plugins/select2/css/select2.min',
        '/assets/global/plugins/select2/css/select2-bootstrap.min',
        '/assets/global/plugins/bootstrap-datepicker/css/bootstrap-datepicker3.min',
        '/assets/global/plugins/bootstrap-timepicker/css/bootstrap-timepicker.min',
        '/assets/global/plugins/bootstrap-datetimepicker/css/bootstrap-datetimepicker.min'
    ));
$this->end();
$this->start('opsijs');
echo $this->Html->script(
    array(
        '/assets/global/plugins/select2/js/select2.full.min',
        '/assets/pages/scripts/components-select2.min',
        '/assets/global/plugins/moment.min',
        '/assets/global/plugins/bootstrap-datetimepicker/js/bootstrap-datetimepicker.min',
        '/assets/pages/scripts/components-date-time-pickers.min'
    ));
?>
<script>
function addMaterial() {
    var dataDari = [];
    dataDari['materialCode'] = $("#RequestMaterialId").val();
    dataDari['qty'] = $("#RequestQty").val();

    var dataFinal = {
        material_code: dataDari['materialCode'],
        qty: dataDari['qty']
    }
}

```

```

};
if (dataDari['materialCode'] == 0) {
    alert('cannot be empty');
} else {
    $.ajax({
        type: 'POST',
        url: '<?php echo Router::url(array('controller' => 'requests', 'action' =>
'addMaterial')); ?>',
        data: dataFinal,
        success: function (data) {
            addrowTemp(data);
            $('#RequestMaterialId').select2("val", "");
            $('#RequestQty').val("");
        },
        error: function (error) {
            console.log(error);
        }
    });
}
}

function addrowTemp(data) {
    var tempdata = JSON.parse(data);
    var tempMat = "";
    tempMat += "<tr id=\" + tempdata['Material']['material_code'] + \">";
    tempMat += " <td class=\"highlight\">";
    tempMat += " <div class=\"success\"></div>";
    tempMat += " <a id=\"mcode\" href=\"javascript:;\"> +
tempdata['Material']['material_code'] + "</a>";
    tempMat += " </td>";
    tempMat += " <td id=\"mname\" class=\"hidden-xs\"> +
tempdata['Material']['name'] + "</td>";
    tempMat += " <td id=\"mqty\" > " + tempdata['qty'] + "</td>";
    tempMat += " <td>";

```

```

    tempMat += " <a class=\"btn btn-outline btn-circle dark btn-sm black\"
href=\"javascript:delMat('\" + tempdata['Material']['material_code'] + '\");\">";
    tempMat += " <i class=\"fa fa-trash-o\"></i> Delete </a>";
    tempMat += " </td>";
    tempMat += "</tr>";
    $('#material-table').append(tempMat);
}
function delMat(idx) {
    $.ajax({
        type: 'POST',
        url: '<?php echo Router::url(array('controller' => 'requests', 'action' =>
'remMaterial')); ?>',
        data: {index: idx},
        success: function (data) {
            $('#tr#' + idx).remove()
        },
        error: function (error) {
            console.log(error);
        }
    });
}
</script>
<?php
$this->end();
?>
<h3 class="page-title">Add Request</h3>
<div class="page-bar">
    <ul class="page-breadcrumb">
        <li>
            <i class="icon-home"></i>
            <?php echo $this->Html->link('Production', array('controller' => 'pages', 'action'
=> 'display', 'dashboard')) ?>
            <i class="fa fa-angle-right"></i>
        </li>

```

```

</li>
    <?php echo $this->Html->link('Request', array('action' => 'index')) ?>
    <i class="fa fa-angle-right"></i>
</li>
</li>
    <span>Add Request</span>
</li>
</ul>
</div>
<div class="portlet light">
    <div class="portlet-title">
        <div class="caption">
            <span class="caption-subject font-green-haze bold uppercase">Form Add
Request</span>
        </div>
        <div class="tools">
            <a href="javascript:;" class="collapse"> </a>
        </div>
    </div>
    <div class="portlet-body form">
        <?php echo $this->Form->create('Request', array('class' => 'form-horizontal')) ?>
        <div class="form-body">
            <h3 style="margin-top:0;">Data Request</h3>
            <div class="form-group">
                <label class="col-md-3 control-label">Request Number</label>
                <div class="col-md-4">
                    <?php echo $this->Form->input('request_number', array('type' => 'text',
'class' => 'form-control', 'placeholder' => 'Enter Text', 'label' => false, 'value' => $code,
'readonly')) ?>
                </div>
            </div>
            <div class="form-group">
                <label class="col-md-3 control-label">Date</label>
                <div class="col-md-4">

```

```

<div class="input-group date form_datetime">
    <?php echo $this->Form->input('request_date', array('type' => 'text', 'class'
=> 'form-control', 'readonly' => 'readonly', 'label' => false)) ?>
    <span class="input-group-btn">
        <button class="btn default date-set" type="button">
            <i class="fa fa-calendar"></i>
        </button>
    </span>
</div>
</div>
</div>
<br>
<h3>Detail Material</h3>
<div class="form-group">
    <label for="single" class="col-md-3 control-label">Code </label>
    <div class="col-md-4">
        <?php echo $this->Form->input('material_id', array('options' => $material,
'empty' => array(0 => ''), 'class' => 'form-control select2', 'label' => false)) ?>
    </div>
</div>
<div class="form-group">
    <label class="col-md-3 control-label">Qty</label>
    <div class="col-md-4">
        <?php echo $this->Form->input('qty', array('class' => 'form-control',
'placeholder' => 'Enter Text', 'label' => false)) ?>
    </div>
</div>
<div class="col-md-offset-3 col-md-3">
    <button type="reset" class="btn default">Clear</button>
    <button type="button" onclick="addMaterial()" class="btn
green">Add</button>
</div>
<div class="row" style="padding-top:80px;">
    <div class="col-md-offset-1 col-md-8">

```

```

<!-- BEGIN SAMPLE TABLE PORTLET-->
<div class="portlet">
  <div class="portlet-title">
    <div class="caption">
      <i class="fa fa-shopping-cart"></i>list of temporary materials
    </div>
    <div class="tools">
      <a href="javascript:;" class="collapse"> </a>
    </div>
  </div>
  <div class="portlet-body">
    <div class="table-scrollable">
      <table id="material-table"
        class="table table-striped table-bordered table-advance table-
hover">
        <thead>
          <tr>
            <th>Code</th>
            <th class="hidden-xs">Name</th>
            <th>Qty</th>
            <th></th>
          </tr>
        </thead>
        <tbody>
          <?php foreach ($viewTemp as $sesMat): ?>
            <tr id="<?php echo $sesMat['Material']['material_code'] ?>">
              <td class="highlight">
                <div class="success"></div>
                <a href="javascript:;" id="mcode"> <?php echo
$sesMat['Material']['material_code'] ?> </a>
              </td>
              <td id="mname" class="hidden-xs"> <?php echo
$sesMat['Material']['name'] ?></td>
              <td id="mqty"> <?php echo $sesMat['qty'] ?></td>

```

```

        <td>
            <a href="javascript:delMat('<?php echo
$sesMat['Material']['material_code'] ?>');"
            class="btn btn-outline btn-circle dark btn-sm black">
                <i class="fa fa-trash-o"></i> Delete </a>
        </td>
    </tr>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>
</div>
</div>
</div>
<!-- END SAMPLE TABLE PORTLET-->
</div>
</div>
</div>
<div class="form-actions">
    <div class="row">
        <div class="col-md-offset-9 col-md-3">
            <button type="button" class="btn default"
onclick=self.history.back()>Cancel</button>
            <button type="submit" class="btn green">Request</button>
        </div>
    </div>
</div>
<?php echo $this->Form->end(); ?>
</div>
</div>

```

### localhost/pebaku\_cake/approvals

```

<?php
/**
 * @var $this View

```

```

*/
$this->start('opsicss');
echo $this->Html->css('/assets/global/plugins/datatables/datatables.min');
echo $this->Html-
>css('/assets/global/plugins/datatables/plugins/bootstrap/datatables.bootstrap');
$this->end();
$this->start('opsijs');
echo $this->Html->script('/assets/global/scripts/datatable');
echo $this->Html->script('/assets/global/plugins/datatables/datatables.min');
echo $this->Html-
>script('/assets/global/plugins/datatables/plugins/bootstrap/datatables.bootstrap');
echo $this->Html->script('/assets/pages/scripts/table-datatables-managed.min');
$this->end();
?>
<h3 class="page-title">Request Approvals</h3>
<div class="page-bar">
  <ul class="page-breadcrumb">
    <li>
      <i class="icon-home"></i>
      <?php echo $this->Html->link('Home', array('controller' => 'pages', 'action' =>
'display', 'dashboard')) ?>
      <i class="fa fa-angle-right"></i>
    </li>
    <li>
      <span>Approval</span>
    </li>
  </ul>
</div>
<div class="portlet light ">
  <div class="portlet-title">
    <div class="caption font-dark">
      <span class="caption-subject bold uppercase"> Approval</span>
    </div>
  </div>
</div>

```

```

<div class="portlet-body">
  <table class="table table-striped table-bordered table-hover table-checkable order-
column" id="sample_1">
    <thead>
      <tr>
        <th>No</th>
        <th> Request Number</th>
        <th> Request Date</th>
        <th> Request by</th>
        <th> Status</th>
        <th> Action</th>
      </tr>
    </thead>
    <tbody>
      <?php
      $no = 1;
      foreach ($request as $r):
        ?>
        <tr class="odd gradeX">
          <td><?php echo $no ?></td>
          <td><?php echo $r['Request']['request_number'] ?></td>
          <td><?php echo $r['Request']['request_date'] ?></td>
          <td><?php echo $r['Requestor']['name'] ?></td>
          <td>
            <?php
            if ($r['Request']['status'] == 0) {
              echo $this->Html->link('Pending', array('action' => 'proceed',
$r['Request']['request_number']), array('escape' => false, 'class' => 'btn btn-sm red'));
            } elseif ($r['Request']['status'] == 1) {
              echo $this->Html->link('On Process', '#', array('escape' => false,
'class' => 'btn btn-sm yellow'));
            } elseif ($r['Request']['status'] == 2) {
              echo $this->Html->link('Delivered', '#', array('escape' => false, 'class'
=> 'btn btn-sm green  '));
            }
          </td>
        </tr>
      </foreach>
    </tbody>
  </table>
</div>

```

```

        }
    ?>
</td>
<td>
    <?php
        if ($r['Request']['status'] == 0) {
            echo $this->Html->link(' Proses <i class="fa fa-check-square-
o"></i>', array('action' => 'proceed', $r['Request']['request_number']), array('escape' =>
false, 'class' => 'btn btn-sm blue'));
        } else {
            echo $this->Html->link('Print <i class="fa fa-check-square-o"></i>',
array('action' => 'print_document', $r['Request']['request_number']), array('escape' =>
false, 'class' => 'btn btn-sm blue', 'target' => '_blank'));
        }
    ?>
</td>
</tr>
<?php
    $no++;
endforeach;
?>
</tbody>
</table>
</div>
</div>

```

### Localhost/pebaku\_cake/stocks

```

<?php
/**
 * @var $this View
 */
$this->start('opsicss');
echo $this->Html->css('/assets/global/plugins/datatables/datatables.min');

```

```

echo $this->Html-
>css('/assets/global/plugins/datatables/plugins/bootstrap/datatables.bootstrap');
$this->end();

$this->start('opsijs');
echo $this->Html->script('/assets/global/scripts/datatable');
echo $this->Html->script('/assets/global/plugins/datatables/datatables.min');
echo $this->Html-
>script('/assets/global/plugins/datatables/plugins/bootstrap/datatables.bootstrap');
echo $this->Html->script('/assets/pages/scripts/table-datatables-managed.min');
$this->end();
?>
<h3 class="page-title">Stock</h3>
<div class="page-bar">
  <ul class="page-breadcrumb">
    <li>
      <i class="icon-home"></i>
      <?php echo $this->Html->link('Home', array('controller' => 'pages', 'action' =>
'display', 'dashboard')) ?>
      <i class="fa fa-angle-right"></i>
    </li>
    <li>
      <span>Stock</span>
    </li>
  </ul>
</div>
<div class="portlet light ">
  <div class="portlet-title">
    <div class="caption font-dark">
      <span class="caption-subject bold uppercase"> Manage Table</span>
    </div>
  </div>
  <div class="portlet-body">
    <div class="table-toolbar">

```

```

<div class="row">
  <div class="col-md-6">
    <div class="btn-group">
      <?php
        echo $this->Html->link("Add New <i class='fa fa-plus'></i>",
array('action' => 'add'), array('escape' => false, 'class' => 'btn sbold green'));
      ?>
    </div>
  </div>
</div>
</div>
</div>
<table class="table table-striped table-bordered table-hover table-checkable order-
column" id="sample_1">
  <thead>
    <tr>
      <th>No</th>
      <th> Code</th>
      <th> Name</th>
      <th> Qty</th>
      <th> Expired</th>
      <th> Receipt Number</th>
      <th> Add by</th>
      <th> Action</th>
    </tr>
  </thead>
  <tbody>
    <?php
      $no = 1;
      foreach ($Stock as $row):
        ?>
        <tr class="odd gradeX">
          <td><?php echo $no ?></td>
          <td><?php echo $row['Material']['material_code'] ?></td>
          <td><?php echo $row['Material']['name'] ?></td>

```

```

<td><?php echo $row['Stock']['qty']?></td>
<td><?php echo $row['Stock']['expired'] ?></td>
<td><?php echo $row['Receipt']['receipt_number'] ?></td>
<td><?php echo ucfirst($row['AuthakeUsers']['login']) ?></td>
<td>
    <?php
        echo $this->Html->link('Edit <i class="fa fa-
edit"></i>',array('action'=>'edit',$row['Stock']['id']),array('escape'=>false, 'class'=>'btn
btn-sm yellow'));
        echo $this->Html->link('Delete <i class="fa fa-
trash"></i>',array('action'=>'delete',$row['Stock']['id']),array('escape'=>false, 'class'=>'btn
btn-sm red'),'Are you sure ?');
    ?>
</td>
</tr>
<?php
    $no++;
endforeach;
?>
</tbody>
</table>
</div>
</div>

```

### Localhost/pebaku\_cake/suppliers

```

<?php
/**
 * @var $this View
 */
$this->start('opsicss');
echo $this->Html->css('/assets/global/plugins/datatables/datatables.min');
echo $this->Html-
>css('/assets/global/plugins/datatables/plugins/bootstrap/datatables.bootstrap');
$this->end();

```

```

$this->start('opsijs');
echo $this->Html->script('/assets/global/scripts/datatable');
echo $this->Html->script('/assets/global/plugins/datatables/datatables.min');
echo $this->Html-
>script('/assets/global/plugins/datatables/plugins/bootstrap/datatables.bootstrap');
echo $this->Html->script('/assets/pages/scripts/table-datatables-managed.min');
$this->end();
?>
<h3 class="page-title">Suppliers</h3>
<div class="page-bar">
  <ul class="page-breadcrumb">
    <li>
      <i class="icon-home"></i>
      <?php echo $this->Html->link('Home', array('controller' => 'pages', 'action' =>
'display', 'dashboard')) ?>
      <i class="fa fa-angle-right"></i>
    </li>
    <li>
      <span>Suppliers</span>
    </li>
  </ul>
</div>
<div class="portlet light ">
  <div class="portlet-title">
    <div class="caption font-dark">
      <span class="caption-subject bold uppercase"> Manage Table</span>
    </div>
  </div>
  <div class="portlet-body">
    <div class="table-toolbar">
      <div class="row">
        <div class="col-md-6">
          <div class="btn-group">

```

```

        <?php
            echo $this->Html->link("Add New <i class='fa fa-plus'></i>",
array('action' => 'add'), array('escape' => false, 'class' => 'btn sbold green'));
        ?>
    </div>
</div>
</div>
</div>
<table class="table table-striped table-bordered table-hover table-checkable order-
column" id="sample_1">
    <thead>
        <tr>
            <th> No</th>
            <th> Company</th>
            <th> Address</th>
            <th> PIC</th>
            <th> Email</th>
            <th> Contact</th>
            <th> Active</th>
            <th> Action</th>
        </tr>
    </thead>
    <tbody>
        <?php
            $no = 1;
            foreach ($supplier as $s):
                ?>
                <tr class="odd gradeX">
                    <td><?php echo $no ?></td>
                    <td><?php echo $s['Supplier']['name'] ?></td>
                    <td><?php echo $s['Supplier']['address'] ?></td>
                    <td><?php echo $s['Supplier']['pic'] ?></td>
                    <td>

```

```

        <a href="mailto:<?php echo $$['Supplier']['email'] ?>"><?php echo
$$['Supplier']['email'] ?></a>
    </td>
    <td class="center"><?php echo $$['Supplier']['phone'] ?></td>
    <td>
        <?php if($$['Supplier']['is_active'] == 1){ ?>
        <span class="label label-sm label-success"> Active </span>
        <?php }else{ ?>
            <span class="label label-sm label-danger"> Not Active </span>
        <?php }?>
    </td>
    <td>
        <?php
        echo $this->Html->link('Edit <i class="fa fa-
edit"></i>',array('action'=>'edit',$$['Supplier']['supplier_id']),array('escape'=>false,
'class'=>'btn btn-sm yellow'));
        echo $this->Html->link('Delete <i class="fa fa-
trash"></i>',array('action'=>'delete',$$['Supplier']['supplier_id']),array('escape'=>false,
'class'=>'btn btn-sm red'),'Are you sure ?');
        ?>
    </td>
</tr>
<?php
$no++;
endforeach;
?>
</tbody>
</table>
</div>
</div>

```

### Localhost/pebaku\_cake/user

```

<div class="row">
    <div class="col-lg-12">

```

```

<h1 class="page-header">
    Manage Users
</h1>
<ol class="breadcrumb">
    <li><a href="#"><i class="fa fa-home"></i></a></li>
    <li><?php echo $this->Html->link('Users', array('controller' => 'users', 'action' =>
'Manage Users', 'plugin' => 'authake')) ?></li>
    <li><a href="#">Manage Users</a></li>
</ol>
<div class="pull-right">
    <?php echo $this->Html->link('New User', array('controller' => 'users', 'action' =>
'add'), array('class' => 'btn btn-primary pull-right')); ?>
</div>
</div>
<div class="row">
    <div class="col-lg-12">
        <div class="boxed no-padding">
            <div class="inner">
                <div class="table-holder">
                    <!-- Table data -->
                    <div class="table-holder">
                        <table cellpadding="0" cellspacing="0" border="0" class="table">
                            <thead>
                                <tr>
                                    <th><?php echo $this->Paginator->sort('login', 'Login',
array('escape' => false)); ?></th>
                                    <th><?php echo $this->Paginator->sort('email', 'Email',
array('escape' => false, 'style' => 'width: 300px')); ?></th>
                                    <th>Grup</th>
                                    <th><?php echo $this->Paginator->sort('created', 'Date Created',
array('escape' => false)); ?></th>
                                    <!-- <th>Status</th> -->
                                <th></th>
                            </thead>

```

```

</tr>
</thead>
<tbody>
<?php
$i = 0;
foreach ($users as $user):
    $class = "";
    if ($i++ % 2 == 0) {
        $class = 'altrow';
    }
    // check if user account enables
    $exp = $user['User']['expire_account'];

    if ($user['User']['disable'] or ( $exp != '0000-00-00' and $this-
>Time->fromString($exp) < time()))
        $class = " class=\"{$class} disabled\"";
    else
        $class = " class=\"{$class}\"";
    ?>
<tr>
<td>
    <?php echo $user['User']['login'] ?>
</td>
<td>
    <?php
    $email = $user['User']['email'];
    echo "<a href=\"mailto:{$email}\">{$email}</a>";
    ?>
</td>
<td>
    <div class="muted">
    <?php
    //pr($user['Group']);

```

```

        $gr = (count($user['Group'])) ? array() : array(__('Guest'));
// Specify Guest group if lonely group
        foreach ($user['Group'] as $k => $group)
            $gr[] = $this->Html->link(__($group['name']), '#',
array('class' => 'label label-info'));
            echo implode(' ', $gr);
            ?>
        </div>
    </td>
    <td>
        <?php echo $this->Time->format('d/m/Y',
$user['User']['created']); ?>&nbsp;
    </td>
    <!-- <td>
        <?php
            if ($user['User']['disable']) {
                echo '<span class="label label-
important">Disabled</span>&nbsp;';
            } else {
                echo '<span class="label label-
success">Enabled</span>&nbsp;';
            }
        ?>
    </td> -->
    <td>
        <?php echo $this->Html->link('<i class="fa fa-pencil"></i>',
array('action' => 'edit', $user['User']['id'], $user['User']['id']), array('class' => 'edit text-
info', 'escape' => false)); ?>
        <?php echo $this->Html->link('<i class="fa fa-minus-
circle"></i>', array('action' => 'delete', $user['User']['id'], $user['User']['id'], '?' =>
array('token' => $this->Session->read('token'))), array('class' => 'remove text-danger',
'escape' => false, 'confirm' => 'Do you want to delete?')); ?>
    </td>
</tr>

```

```
        <?php endforeach; ?>
    </tbody>
</table>
    <?php echo $this->element('admin/paging'); ?>
</div>
</div>
</div>
</div>
</div>
</div>
```