

**RANCANG BANGUN SISTEM INFORMASI PERSEDIAAN
BAHAN BAKU DIVISI WAREHOUSE RAW MATERIAL
MENGUNAKAN PHP 5.4.7 DAN MYSQL 5.5.27 PADA
PT REKADAYA MULTI ADIPRIMA**

TUGAS AKHIR

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

**OLEH
ARIFIN SUPARDAN
1413010**



**POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA
JAKARTA
2018**

POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN RI

LEMBAR PERSETUJUAN DOSEN PEMBIMBING

JUDUL TUGAS AKHIR :

**RANCANG BANGUN SISTEM INFORMASI PERSEDIAAN BAHAN
BAKU DIVISI *WAREHOUSE RAW MATERIAL* MENGGUNAKAN
PHP 5.4.7 DAN MYSQL 5.5.27 PADA PT REKADAYA MULTI
ADIPRIMA**

DISUSUN OLEH :

NAMA : ARIFIN SUPARDAN
NIM : 1413010

PROGRAM STUDI : SISTEM INFORMASI

Telah diperiksa dan disetujui untuk memenuhi sebagian syarat penyelesaian Program
Studi Sistem Informasi Industri Otomotif Diploma Empat (D-IV) Sistem Informasi
pada Politeknik STMI Jakarta

Jakarta, 18 Oktober 2017

Dosen Pembimbing

Drs. Jacob Saragih, M.M

NIP : 195404281986031002

POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN RI

LEMBAR PERSETUJUAN ASISTEN DOSEN PEMBIMBING

JUDUL TUGAS AKHIR :

**RANCANG BANGUN SISTEM INFORMASI PERSEDIAAN BAHAN
BAKU DIVISI *WAREHOUSE RAW MATERIAL* MENGGUNAKAN
PHP 5.4.7 DAN MYSQL 5.5.27 PADA PT REKADAYA MULTI
ADIPRIMA**

DISUSUN OLEH :

NAMA : ARIFIN SUPARDAN
NIM : 1413010

PROGRAM STUDI : SISTEM INFORMASI

Telah diperiksa dan disetujui untuk memenuhi sebagian syarat penyelesaian Program
Studi Sistem Informasi Industri Otomotif Diploma Empat (D-IV) Sistem Informasi
pada Politeknik STMI Jakarta

Jakarta, 12 Oktober 2017

Asisten Dosen Pembimbing

Ahmad Juniar, S.Kom, MT
NIP : 197906052006041002

POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

LEMBAR PENGESAHAN

Judul Tugas Akhir : RANCANG BANGUN SISTEM INFORMASI
PERSEDIAAN BAHAN BAKU DIVISI *WAREHOUSE*
RAW MATERIAL MENGGUNAKAN PHP 5.4.7 DAN
MYSQL 5.5.27 PADA PT REKADAYA MULTI
ADIPRIMA

Disusun Oleh :
Nama : Arifin Supardan
Nim : 1413010
Program Studi : Sistem Informasi Industri Otomotif

Telah Diuji Oleh Tim Penguji Sidang Tugas Akhir Program Studi Sistem
Informasi Industri Otomotif Politeknik STMI Jakarta Kementerian Perindustrian
Republik Indonesia pada hari Jumat tanggal 16 Maret 2018

Jakarta, 16 Maret 2018

Dosen Pembimbing

Ketua Penguji

Drs. Jacob Saragih, M.M
NIP. 195404281986031002

Ulil Hamida, ST, MT
NIP. 198103272005022001

Dosen Penguji

Dosen Penguji

Lucky Heriyanto, ST, MTI
NIP. 197908202009011009

Fifi Lailasari Hadianastuti S.kom, M.kes
NIP. 197310162005022001

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : Arifin Supardan

Nim : 1413010

Program Studi : Sistem Informasi Industri Otomotif

Berstatus sebagai mahasiswa Program Studi Sistem Informasi di Politeknik STMI Jakarta Kementerian Perindustrian RI. Dengan ini menyatakan bahwa hasil karya Tugas Akhir yang saya buat dengan judul :

“RANCANG BANGUN SISTEM INFORMASI PERSEDIAAN BAHAN BAKU DIVISI WAREHOUSE RAW MATERIAL MENGGUNAKAN PHP 5.4.7 DAN MYSQL 5.5.27 PADA PT REKADAYA MULTI ADIPRIMA”.

- **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/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 tidak memenuhi apa yang telah saya nyatakan diatas, maka saya bersedia menerima sanksi atas apa yang telah saya lakukan sesuai peraturan yang berlaku.

Jakarta, 10 Oktober 2017

Yang Membuat Persyataan

Arifin Supardan

LEMBAR BIMBINGAN PENYUSUNAN TUGAS AKHIR

Nama : Arifin Supardan
NIM : 1413010
Judul Laporan TA : Rancang Bangun Sistem Informasi Persediaan Bahan Baku Divisi Warehouse Raw Material menggunakan PHP 5.4.7 dan MYSQL 5.5.27 pada PT Rekadaya Multi Adiprima
Dosen Pembimbing : Drs. Jacob Saragih, M.M

Tanggal	BAB	Keterangan	Paraf
25/07/2017	I	Usulan TA	
01/08/2017	I - III	BAB I – BAB III	
07/08/2017	IV	BAB IV	
14/08/2017	V	BAB V - Koreksi	
28/08/2017	V - I	BAB V - I	
04/09/2017	I - VI	BAB I - VI	
11/09/2017	I - VI	Koreksi BAB I - VI	
18/10/2017	I - VI	Presentasi	

Mengetahui,
Kepala Program Studi
Sistem Informasi Industri Otomotif

Dosen Pembimbing

Drs. Jacob Saragih, M.M
NIP : 195404281986031002

Drs. Jacob Saragih, M.M
NIP : 195404281986031002

LEMBAR BIMBINGAN PENYUSUNAN TUGAS AKHIR

Nama : Arifin Supardan
NIM : 1413010
Judul Laporan TA : Rancang Bangun Sistem Informasi Persediaan Bahan Baku Divisi Warehouse Raw Material menggunakan PHP 5.4.7 dan MYSQL 5.5.27 pada PT Rekadaya Multi Adiprima

Asisten Dosen Pembimbing: Ahmad Juniar, S.Kom, MT

Tanggal	BAB	Keterangan	Paraf
18/07/2017	I	BAB I Pendahuluan. Perbaiki Hal 2.	
27/07/2017	II	BAB II Landasan Teori. Perbaiki hal 36.	
28/07/2017	II	BAB II Landasan Teori. Perbaiki hal 53, 61.	
22/08/2017	III	BAB III Metodologi. Perbaiki hal 65, 66.	
27/09/2017	IV	BAB IV Pengumpulan dan Pengolahan Data.	
29/09/2017	V	BAB V Analisis dan Pembahasan. Perbaiki.	
05/10/2017	I - VI	BAB I – VI Kesimpulan. Perbaiki.	
12/10/2017	I - VI	Daftar Isi, Daftar Gambar, Daftar Tabel, Daftar Pustaka dll. Selesai.	

Mengetahui,
Kepala Program Studi
Sistem Informasi Industri Otomotif

Asisten Dosen Pembimbing

Drs. Jacob Saragih, M.M
NIP : 195404281986031002

Ahmad Juniar, S.Kom, MT
NIP : 197906052006041002

ABSTRAK

PT Rekadaya Multi Adiprima merupakan salah satu perusahaan yang bergerak dalam industri manufaktur komponen otomotif seperti *part* plastik *service hole*, *felt*, *insulator dush panel* serta komponen otomotif lainnya. Pencatatan bahan baku masuk dan keluar pada Divisi *Warehouse Raw Material* masih dilakukan secara manual dengan cara meng-*update* kartu stok, sehingga data yang ada masih belum terorganisir dengan baik. Belum tersedianya laporan persediaan bahan baku pada Divisi *Warehouse Raw Material* sehingga pengecekan stok bahan baku yang masuk dan keluar masih berdasarkan kartu stok. Sistem informasi persediaan bahan baku sangat diperlukan dalam perusahaan untuk melakukan proses pengolahan data bahan baku. Sistem informasi yang diusulkan akan mempermudah dan mengintegrasikan proses pengolahan data sehingga membantu bagian di dalam sistem untuk saling bertukar informasi dan mengambil keputusan dengan cepat. Dalam mengolah data diperlukan metode pengembangan sistem. Adapun metode pengembangan sistem yang digunakan pada rancangan sistem informasi persediaan bahan baku dengan menggunakan *prototype evolutioner*. Dalam pengolahan data pengembangan sistem informasi data barang menggunakan beberapa alat bantu seperti ERD, *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*, *Deployment Diagram*, Kamus Data, HIPO dan *Flowchart*. Aplikasi tersebut dirancang dengan menggunakan PHP 5.4.7 dan MySQL 5.5.27. Melihat hal tersebut penulis mengangkat permasalahan tersebut kedalam sebuah penelitian dengan judul “RANCANG BANGUN SISTEM INFORMASI PERSEDIAAN BAHAN BAKU DIVISI WAREHOUSE RAW MATERIAL MENGGUNAKAN PHP 5.4.7 DAN MYSQL 5.5.27 PADA PT REKADAYA MULTI ADIPRIMA.

Kata kunci: Sistem Informasi, Persediaan, Bahan Baku, *Use Case*, PHP dan MySQL.

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 ini dengan baik dan lancar.

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

Dalam penyusunan Tugas Akhir ini, penulis mengambil pokok bahasan dengan judul “**Rancang Bangun Sistem Informasi Persediaan Bahan Baku Divisi Warehouse Raw Material Menggunakan PHP 5.4.7 dan MySql 5.5.27 pada PT Rekadaya Multi Adiprima**”. Penulisan ini juga merupakan penerapan atau latihan untuk mengaplikasikan teori-teori yang pernah penulis dapatkan pada mata kuliah yang dipelajari dengan kenyataan di lapangan.

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. Kedua orang tua, Bapak Suparman dan Ibu Supinah serta keluarga tercinta yang telah memberikan doa, dukungan, pengorbanan, semangat dan kasih sayang hingga saat ini.
2. Bapak Dr. Mustofa, S.T., M.T., selaku Direktur Politeknik STMI Jakarta.
3. Bapak Drs. Jacob Saragih, M.M., selaku Ketua Program Studi Sistem Informasi Industri Otomotif dan selaku dosen pembimbing yang telah memberikan arahan dan masukan dalam penyusunan Tugas Akhir.
4. Bapak Ahmad Juniar, S.Kom, MT selaku asisten dosen pembimbing yang telah memberikan arahan, bimbingan serta penjelasan dalam penyusunan Tugas Akhir.
5. Ibu Rosalina Faried selaku Presiden Direktur PT Rekadaya Multi Adiprima.

6. Bapak Joko, Bapak August, Bapak Bowo, Bapak Ali, Bapak Eriza, Bapak Fahmi, Ibu Ika, Ibu Mery dan seluruh pegawai di PT Rekadaya Multi Adiprima yang telah membantu memberi arahan serta informasi untuk pengumpulan data.
7. Seluruh dosen Politeknik STMI Jakarta yang telah memberikan ilmunya guna menambah pengetahuan dan pengalaman yang sangat bermanfaat.
8. Teman-teman mahasiswa/i Politeknik STMI Jakarta Program Studi Sistem Informasi terutama SA21 2013 atas kebersamaan dan motivasinya selama ini.

Serta semua pihak yang baik langsung maupun tidak langsung memberikan kritik, saran dan bantuan dalam pembuatan laporan ini yang tidak dapat Penulis sebutkan satu persatu. 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.

Jakarta, 18 Oktober 2017

Penulis

DAFTAR ISI

	Halaman
Halaman Judul	i
Lembar Persetujuan Dosen Pembimbing	ii
Lembar Persetujuan Asisten Dosen Pembimbing	iii
Lembar Bimbingan Tugas Akhir	iv
Lembar Pernyataan Keaslian	vi
Abstrak	vii
Kata Pengantar	viii
Daftar Isi	x
Daftar Gambar	xiv
Daftar Tabel	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Pokok Permasalahan	2
1.3 Tujuan Penelitian	3
1.4 Batasan Masalah.....	3
1.5 Manfaat Tugas Akhir	3
1.6 Sistematika Penulisan	4
BAB II LANDASAN TEORI	6
2.1 Rancang Bangun	6
2.2 Konsep Dasar Sistem Informasi.....	6
2.2.1 Komponen Sistem Informasi	6
2.2.2 Tipe Sistem Informasi	8
2.3 Pengertian Persediaan	10
2.3.1 Fungsi Persediaan	11
2.3.2 Jenis-Jenis Persediaan	12
2.3.3 Penyebab Timbulnya Persediaan	12

2.3.4	Faktor-Faktor yang Menentukan Persediaan	13
2.3.5	<i>Reorder Point</i> (Titik Pemesanan Kembali).....	13
2.3.6	Metode dalam Gudang	18
2.4	Pengertian Bahan Baku	19
2.4.1	Sistem Bahan Baku	19
2.5	Bahasa Pemrograman.....	20
2.5.1	PHP (<i>Hypertext Preprocessor</i>)	20
2.5.2	XAMPP.....	21
2.6	Perancangan Database.....	21
2.6.1	Pengertian Database	21
2.6.2	Normalisasi	22
2.6.3	<i>Relational Database Management Systems</i> (RDBMS).....	22
2.6.4	Permodelan Data	22
2.6.5	MySQL.....	24
2.7	Analisis Sistem Informasi	27
2.8	Metode Pengembangan Sistem	28
2.8.1	<i>Model Prototype</i>	30
2.8.2	<i>Evolutionary Prototype</i>	32
2.9	<i>Flowmap</i> dan <i>Flowchart</i>	33
2.10	Jenis-Jenis <i>Flowchart</i>	34
2.10.1	<i>Systems Flowchart</i>	34
2.10.2	<i>Document Flowchart</i>	35
2.10.3	<i>Schematic Flowchart</i>	35
2.10.4	<i>Program Flowchart</i>	36
2.10.5	<i>Process Flowchart</i>	37
2.11	<i>Unified Modeling Language</i> (UML) 2.0	38
2.12	Diagram Permodelan.....	38
2.12.1	<i>Structure Diagrams</i>	39
2.12.2	<i>Behavior Diagram</i>	43

2.13	<i>Hierarchy plus Input-Proses-Output (HIPO)</i>	51
2.13.1	Diagram dalam Paket HIPO.....	51
2.14	Kamus Data	53
BAB III	METODOLOGI PENELITIAN	54
3.1	<u>Metodologi Penelitian</u>	54
3.2	Jenis dan Sumber Data	54
3.3	Metode Pengumpulan Data	55
3.4	Instrumen Pengumpulan Data	55
3.5	Metode Pengembangan Sistem	56
3.6	Kerangka Penelitian	57
BAB IV	PENGUMPULAN DAN PENGOLAHAN DATA	62
4.1	Sejarah Perusahaan.....	62
4.2	Profil Perusahaan	63
4.3	Visi, Misi dan Pilar Lingkungan Perusahaan	64
4.4	Struktur Organisasi Perusahaan	64
4.5	<i>Warehouse Raw Material</i>	66
4.5.1	Struktur Organisasi <i>Warehouse Raw Material</i>	66
4.6	Bahan Baku	70
4.7	Proses dalam Persediaan Bahan Baku	71
4.8	Sistem yang Sedang Berjalan.....	72
4.8.1	Persediaan Bahan Baku.....	75
4.8.2	Penerimaan dan Penyimpanan Bahan Baku.....	76
4.8.3	Pemakaian Bahan Baku.....	80
4.9	<i>Use Case Diagram</i> Sistem yang Berjalan.....	81
BAB V	ANALISIS DAN PEMBAHASAN	86
5.1	Tahap Analisis.....	86
5.2	Perancangan Sistem	90
5.2.1	Tujuan Sistem yang Diusulkan	90
5.2.2	Perancangan Sistem yang Diusulkan	90
5.2.3	<i>Use Case Diagram</i> yang Diusulkan.....	92
5.2.4	<i>Activity Diagram</i> yang Diusulkan	97

5.2.5	<i>Sequence Diagram</i> yang Diusulkan	106
5.2.6	<i>Deployment Diagram</i> yang Diusulkan	115
5.2.7	<i>Class Diagram</i> yang Diusulkan	115
5.2.8	<i>Entity Relationship Diagram</i> yang Diusulkan	116
5.2.9	Kamus Data yang Diusulkan	116
5.3	Analisis Desain Program.....	124
5.3.1	Struktur Menu Program.....	125
5.3.2	Perancangan <i>Interface</i> Program	126
5.3.3	<i>Flowchart</i> Program	135
5.4	Analisis <i>Software</i> dan <i>Hardware</i>	143
BAB VI	KESIMPULAN DAN SARAN	144
6.1	Kesimpulan	144
6.2	Saran	144
	DAFTAR PUSTAKA	145
	LAMPIRAN	

DAFTAR GAMBAR

	Halaman
Gambar II.1 Tipe Sistem Informasi	8
Gambar II.2 Kurva Titik Pemesanan Kembali	14
Gambar II.3 Hubungan antara ROP, <i>Safety Stock</i> , dan <i>Lead Time</i>	18
Gambar II.4 Ilustrasi Model <i>Prototype</i>	31
Gambar II.5 <i>Evolutionary Prototype Model</i>	32
Gambar II.6 <i>Visual Table Of Contents</i>	52
Gambar II.7 <i>Overview Diagram</i>	52
Gambar III.1 Kerangka Penelitian	60
Gambar IV.1 Struktur Organisasi PT Rekadaya Multi Adiprima	65
Gambar IV.2 Struktur Organisasi Divisi <i>Warehouse Raw Material</i>	66
Gambar IV.3 <i>Fiber</i>	71
Gambar IV.4 Proses Persediaan Bahan Baku	71
Gambar IV.5 Sistem Persediaan Bahan Baku yang Berjalan	73
Gambar IV.6 Form Bukti Order Barang (BOB)	75
Gambar IV.7 Surat Jalan Tanda Penerimaan Barang	76
Gambar IV.8 <i>Form Check Sheet Raw Material</i>	77
Gambar IV.9 <i>Form Check Sheet Incoming Inspection Raw Material</i> ..	78
Gambar IV.10 Kartu Stok	79
Gambar IV.11 Form Bukti Keluar Barang	80
Gambar IV.12 <i>Use Case Diagram</i> Sistem yang Berjalan	81
Gambar V.1 <i>Flowmap</i> Program Usulan Sistem Persediaan Bahan Baku .	91
Gambar V.2 <i>Use Case Diagram</i> Sistem Persediaan Bahan Baku yang Diusulkan	92
Gambar V.3 <i>Activity Diagram</i> Kelola Data Bahan Baku.....	98
Gambar V.4 <i>Activity Diagram</i> Input Data BB Masuk	99
Gambar V.5 <i>Activity Diagram</i> Input Data BB Keluar	100
Gambar V.6 <i>Activity Diagram</i> Lihat Laporan BB Masuk	101

Gambar V.7	<i>Activity Diagram</i> Lihat Laporan BB Keluar	102
Gambar V.8	<i>Activity Diagram</i> PO Request	103
Gambar V.9	<i>Activity Diagram</i> BB Request	104
Gambar V.10	<i>Activity Diagram</i> Buat PO.....	105
Gambar V.11	<i>Sequence Diagram</i> Proses Login.....	106
Gambar V.12	<i>Sequence Diagram</i> Pengelolaan Data Bahan Baku.....	107
Gambar V.13	<i>Sequence Diagram</i> Input BB Request	108
Gambar V.14	<i>Sequence Diagram</i> Input BB Masuk.....	109
Gambar V.15	<i>Sequence Diagram</i> Input PO Request	110
Gambar V.16	<i>Sequence Diagram</i> Input BB Keluar.....	111
Gambar V.17	<i>Sequence Diagram</i> Buat PO.....	112
Gambar V.18	<i>Sequence Diagram</i> Lihat Laporan BB Masuk	113
Gambar V.19	<i>Sequence Diagram</i> Lihat Laporan BB Keluar	114
Gambar V.20	<i>Deployment Diagram</i> Sistem Informasi Persediaan Bahan Baku	115
Gambar V.21	<i>Class Diagram</i> Sistem Informasi Persediaan Bahan Baku	116
Gambar V.22	<i>Entity Relationship Diagram</i> Sistem Persediaan Bahan Baku Usulan.....	117
Gambar V.23	Struktur Menu Program.....	125
Gambar V.24	Rancangan <i>Form Login</i>	126
Gambar V.25	Rancangan <i>Form</i> Menu Utama	127
Gambar V.26	Rancangan <i>Form</i> Tambah Bahan Baku	128
Gambar V.27	Rancangan <i>Form</i> Transaksi Penerimaan Bahan Baku	129
Gambar V.28	Rancangan <i>Form</i> Transaksi Pengeluaran Bahan Baku	130
Gambar V.29	Rancangan <i>Form</i> Transaksi Permintaan PO	131
Gambar V.30	Rancangan <i>Form</i> Permintaan Bahan Baku	132
Gambar V.31	Rancangan <i>Form</i> Laporan Bahan Baku Masuk	133
Gambar V.32	Rancangan <i>Form</i> Laporan Bahan Baku Keluar	134
Gambar V.33	<i>Flowchart</i> Menu Utama	135
Gambar V.34	<i>Flowchart</i> Menu Bahan Baku	136
Gambar V.35	<i>Flowchart</i> Menu BB Masuk.....	137

Gambar V.36	<i>Flowchart Menu BB Keluar</i>	138
Gambar V.37	<i>Flowchart Menu BB Request</i>	139
Gambar V.38	<i>Flowchart Menu PO Request</i>	140
Gambar V.39	<i>Flowchart Menu Laporan BB Masuk</i>	141
Gambar V.40	<i>Flowchart Menu Laporan BB Keluar</i>	142

DAFTAR TABEL

	Halaman
Tabel II.1	Elemem-Elemen <i>Entity Relationship Diagram</i> 23
Tabel II.2	Beberapa Tipe Data pada MySQL 26
Tabel II.3	Simbol Bagan Alir Sistem (<i>Systems Flowchart</i>)..... 34
Tabel II.4	Simbol Bagan Alir Program (<i>Program Flowchart</i>) 36
Tabel II.5	Simbol Bagan Alir Proses (<i>Process Flowchart</i>) 37
Tabel II.6	Elemen-Elemen <i>Class Diagram</i> 39
Tabel II.7	Elemen-Elemen <i>Package Diagram</i> 40
Tabel II.8	Elemen-Elemen <i>Deployment Diagram</i> 41
Tabel II.9	Elemen-Elemen <i>Activity Diagram</i> 43
Tabel II.10	Elemen-Elemen <i>Sequence Diagram</i> 45
Tabel II.11	Elemen-Elemen <i>Communication Diagram</i> 47
Tabel II.12	Elemen-Elemen <i>Behavior State Machines</i> 48
Tabel II.13	Elemen-Elemen <i>Use Case Diagram</i> 50
Tabel IV.1	Definisi Aktor dan <i>Use Case Diagram</i> 82
Tabel IV.2	Definisi <i>Use Case</i> Permintaan Bahan Baku 83
Tabel IV.3	Definisi <i>Use Case</i> Cek Persediaan Bahan Baku 83
Tabel IV.4	Definisi <i>Use Case</i> Pemesanan Bahan Baku 84
Tabel IV.5	Definisi <i>Use Case</i> Pengiriman Bahan Baku 84
Tabel IV.6	Definisi <i>Use Case</i> Pengecekan Bahan Baku 84
Tabel V.1	Identifikasi Kebutuhan <i>User</i> 86
Tabel V.2	Definisi Aktor Sistem Usulan 93
Tabel V.3	Definisi <i>Use Case Login</i> Sistem Usulan 93
Tabel V.4	Definisi <i>Use Case</i> Pengelolaan Data Bahan Baku Sistem Usulan 94
Tabel V.5	Definisi <i>Use Case Input BB Request</i> Sistem Usulan..... 94
Tabel V.6	Definisi <i>Use Case Input BB Masuk</i> Sistem Usulan..... 95
Tabel V.7	Definisi <i>Use Case Input PO Request</i> Sistem Usulan 95

Tabel V.8	Definisi <i>Use Case Input</i> BB Keluar Sistem Usulan.....	96
Tabel V.9	Definisi <i>Use Case</i> Buat PO Sistem Usulan.....	96
Tabel V.10	Definisi <i>Use Case</i> Melakukan <i>Approval</i> Sistem Usulan.....	96
Tabel V.11	Definisi <i>Use Case</i> Lihat Laporan Bahan Baku Masuk Sistem Usulan	97
Tabel V.12	Definisi <i>Use Case</i> Lihat Laporan Bahan Baku Keluar Sistem Usulan	97
Tabel V.13	Tabel <i>Master</i> Karyawan	118
Tabel V.14	Tabel <i>Master Supplier</i>	118
Tabel V.15	Tabel <i>Master</i> Kategori	119
Tabel V.16	Tabel <i>Master</i> Bahan Baku.....	119
Tabel V.17	Tabel Transaksi Bahan Baku Masuk	120
Tabel V.18	Tabel Transaksi Bahan Baku Masuk Detail.....	121
Tabel V.19	Tabel Transaksi Bahan Baku Keluar.....	121
Tabel V.20	Tabel Transaksi Bahan Baku Keluar Detail.....	121
Tabel V.21	Tabel Transaksi Permintaan Bahan Baku	122
Tabel V.22	Tabel Transaksi Permintaan Bahan Baku Detail	122
Tabel V.23	Tabel Transaksi Permintaan PO.....	123
Tabel V.24	Tabel Transaksi Permintaan PO Detail	123
Tabel V.25	Tabel PO.....	124
Tabel V.26	Tabel PO Detail.....	124

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era globalisasi ini, persaingan bidang industri semakin ketat seiring meningkatnya kebutuhan masyarakat terhadap sejumlah produk barang dan jasa. Hal ini mendorong suatu organisasi atau perusahaan untuk lebih kreatif, inovatif dan efisien dalam menghasilkan produk yang berkualitas tinggi. Untuk menciptakan produk yang berkualitas tinggi, harus memperhatikan proses produksi yang sedang berjalan mulai dari pengambilan bahan baku di gudang hingga menjadi produk yang mempunyai nilai jual dan sampai ke tangan konsumen.

PT Rekadaya Multi Adiprima merupakan perusahaan yang bergerak dibidang industri otomotif. Dalam melakukan aktifitas produksi, keberadaan *warehouse* pada PT Rekadaya Multi Adiprima terbagi ke dalam beberapa divisi dimana kelancaran proses di masing-masing divisi ini akan menentukan keberhasilan proses produksi. Salah satu Divisi *Warehouse* yang dimiliki PT Rekadaya Multi Adiprima adalah Divisi *Warehouse Raw Material*.

Persediaan bahan baku menjadi sangat penting untuk PT Rekadaya Multi Adiprima karena jika persediaan bahan baku habis maka proses produksi akan terhambat. Divisi yang bertanggung jawab terhadap persediaan bahan baku adalah Divisi *Warehouse Raw Material*. Bahan baku yang dipakai dalam memproduksi *felt* pada PT Rekadaya Multi Adiprima di *Plant 11* terbagi menjadi tiga jenis yaitu bahan baku utama, bahan baku tambahan dan bahan baku penolong. Bahan baku yang digunakan diantaranya *Fiber C* (murni), *Fiber A (melt)*, dan *Fiber B (Recycle)*.

Dalam menjalankan kegiatannya, pencatatan bahan baku masuk dan keluar pada Divisi *Warehouse Raw Material* masih dilakukan secara manual dengan cara meng-*update* kartu stok, sehingga data yang ada masih belum terorganisir dengan

baik. Hal tersebut menyebabkan terjadinya kerusakan, kehilangan dan ketidaksesuaian pada data bahan baku. Belum tersedianya laporan persediaan bahan baku pada Divisi *Warehouse Raw Material* sehingga sangat sulit dalam menentukan titik pesan kembali dan melihat ketersediaan bahan baku. Hal tersebut menyebabkan terjadinya kekurangan stok akibat melonjaknya permintaan.

Untuk mengatasi masalah yang terjadi pada PT Rekadaya Multi Adiprima, maka perlu menerapkan sistem secara komputerisasi untuk mengatasi hal tersebut. Informasi yang dihasilkan akan membantu Divisi *Warehouse Raw Material* dalam memutuskan jumlah bahan baku yang akan dibeli secara ekonomis. Berdasarkan uraian-uraian diatas, penulisan tugas akhir ini mengambil judul “RANCANG BANGUN SISTEM INFORMASI PERSEDIAAN BAHAN BAKU DIVISI *WAREHOUSE RAW MATERIAL* MENGGUNAKAN PHP 5.4.7 DAN MYSQL 5.5.27 PADA PT REKADAYA MULTI ADIPRIMA”

1.2 Pokok Permasalahan

Permasalahan yang terjadi pada PT Rekadaya Multi Adiprima adalah sebagai berikut:

1. Pencatatan bahan baku masuk dan keluar masih dilakukan dengan cara meng-*update* kartu stok sehingga data yang ada masih belum terorganisir dengan baik. Hal tersebut menyebabkan terjadinya kerusakan, kehilangan dan ketidaksesuaian pada data bahan baku.
2. Belum tersedianya laporan persediaan bahan baku pada Divisi *Warehouse Raw Material* sehingga sangat sulit dalam menentukan titik pesan kembali dan melihat ketersediaan bahan baku. Hal tersebut menyebabkan terjadinya kekurangan stok akibat melonjaknya permintaan.

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian Tugas Akhir ini adalah sebagai berikut:

1. Merancang bangun sistem informasi pada Divisi *Warehouse Raw Material* agar pencatatan bahan baku yang masuk dan keluar gudang dilakukan secara komputerisasi sehingga tidak akan terjadi lagi kerusakan, kehilangan dan ketidaksesuaian pada data bahan baku.
2. Merancang bangun sistem informasi pada Divisi *Warehouse Raw Material* agar penyediaan laporan persediaan bahan baku dapat dilakukan secara komputerisasi sehingga dapat mengetahui titik pesan kembali stok agar tidak terjadi lagi kekurangan persediaan pada bahan baku.

1.4 Batasan Masalah

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

1. Penelitian dilakukan pada Divisi *Warehouse Raw Material* di *Plant 11* pada PT Rekadaya Multi Adiprima.
2. Penelitian ini dilakukan dalam kurun waktu 1 (satu) bulan, sejak bulan Agustus s.d.September 2016.
3. Penelitian dilakukan hanya sebatas menangani masalah pengolahan data bahan baku, pencatatan data penerimaan bahan baku, pencatatan data pengeluaran bahan baku dan laporan ketersediaan bahan baku.

1.5 Manfaat Tugas Akhir

Adapun manfaat yang dapat diambil dari penelitian pada Tugas Akhir ini adalah sebagai berikut:

1. Bagi penulis
 - a. Untuk memberikan pengalaman dalam menganalisis suatu sistem dan diharapkan dapat memberikan suatu solusi permasalahan.
 - b. Untuk memberikan kemampuan dalam mengaplikasikan teori tentang pengembangan sistem informasi sistem secara jelas untuk memberikan solusi terhadap masalah yang diamati.
2. Bagi institusi pendidikan

- a. Menjalin kerja sama antara dunia pendidikan dan dunia kerja dalam rangka pengembangan ilmu.
 - b. Untuk menghasilkan lulusan yang memiliki keterampilan sesuai dengan kebutuhan dunia kerja.
3. Bagi perusahaan
 - a. Memberikan partisipasi dalam pengembangan di dunia pendidikan.
 - b. Mendapatkan saran yang diharapkan mampu menyelesaikan permasalahan yang terjadi pada perusahaan.

1.6 Sistematika Penulisan

Agar lebih mempermudah perumusan dan pemecahan masalah yang akan dibahas pada penelitian Tugas Akhir ini, maka dapat diuraikan tahapan-tahapan dalam penyusunan laporan ini sebagai berikut:

BAB I PENDAHULUAN

Bab ini memuat latar belakang, pokok permasalahan, tujuan penelitian, batasan masalah, manfaat tugas akhir, dan sistematika penulisan laporan.

BAB II LANDASAN TEORI

Bab ini membahas tentang pengertian rancang bangun, konsep dasar sistem informasi, pengertian persediaan, pengertian bahan baku, bahasa pemrograman, perancangan database, analisis sistem informasi, metode pengembangan sistem, *flowmap* dan *flowchart*, *Unified Modelling Language* (UML) 2.0, diagram permodelan, *Hierarchy plus Input-Proses-Output* (HIPO), dan kamus data sebagai alat bantu untuk membuat rancangan konseptual, dan teori-teori lain yang berhubungan dengan perancangan sistem.

BAB III METODOLOGI PENELITIAN

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

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Bab ini akan menguraikan pengumpulan data penelitian yang telah dilakukan, seperti fungsi sistem persediaan bahan baku, alur proses sistem persediaan bahan baku, dokumen-dokumen yang terlibat dalam persediaan bahan baku dan transaksi-transaksi yang terjadi di sistem persediaan bahan baku.

BAB V ANALISIS DAN PEMBAHASAN

Bab ini berisi analisis rinci dari pengolahan data, yakni mulai dari identifikasi kebutuhan sistem, analisis sistem yang meliputi diagram alir sistem yang berjalan, perancangan sistem dengan menggunakan diagram UML, perancangan basis data, perancangan tampilan layar, perancangan hierarki menu, pembuatan *flowchart* program, dan spesifikasi kebutuhan *software* dan *hardware* untuk mengimplementasi sistem yang dibangun.

BAB VI KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan secara keseluruhan yang diperoleh dari hasil analisis dan perancangan sistem serta saran-saran untuk penerapan dan pengembangan lebih lanjut pada sistem yang bersangkutan.

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 Konsep Dasar Sistem Informasi

Telah diketahui bahwa informasi merupakan hal yang sangat penting bagi manajemen dalam pengambilan keputusan. Informasi diperoleh dari sistem informasi (*information systems*) atau disebut juga *processing systems* atau *information processing systems* atau *information-generating systems*. 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 (Jogiyanto, 2005).

2.2.1 Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan (*building block*), yang terdiri dari komponen *input*, komponen *model*, komponen *output*, komponen teknologi, komponen *hardware*, komponen *software*, komponen basis data, dan komponen kontrol. Semua komponen tersebut saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran (Jogiyanto, 2005).

Berikut ini adalah penjelasan dari masing-masing komponen yang terlibat di dalamnya, yaitu (Jogiyanto, 2005):

1. Komponen *input*

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

2. Komponen model

Komponen ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Komponen *output*

Hasil dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua pemakai sistem.

4. Komponen teknologi

Teknologi merupakan *tool box* dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, dan membantu pengendalian dari sistem secara keseluruhan.

5. Komponen *hardware*

Hardware berperan penting sebagai suatu media penyimpanan vital bagi sistem informasi. Yang berfungsi sebagai tempat untuk menampung *database* atau lebih mudah dikatakan sebagai sumber data dan informasi untuk memperlancar dan mempermudah kerja dari sistem informasi.

6. Komponen *software*

Software berfungsi sebagai tempat untuk mengolah, menghitung dan memanipulasi data yang diambil dari *hardware* untuk menciptakan suatu informasi.

7. Komponen basis data

Merupakan blok yang berisi definisi basis data yang disediakan untuk menyimpan data-data yang akan disimpan dalam media penyimpanan. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi menggunakan perangkat lunak paket yang disebut DBMS (*Database Management System*).

8. Komponen control

Merupakan sebuah komponen yang bertugas mendefinisikan bagaimana kontrol terhadap sistem dilakukan sehingga sistem dapat berjalan dengan baik. Dalam blok kontrol ini misalnya didefinisikan bagaimana melindungi data yang ada di database agar selalu sama dengan kenyataan yang dicatat.

2.2.2 Tipe Sistem Informasi

Klasifikasi operasi dan manajemen sistem informasi menurut O'Brien (2005), ditunjukkan pada Gambar II.1 berikut:



Gambar II.1. Tipe Sistem Informasi

(Sumber: O'Brien, 2005)

Gambar II.1 menggambarkan klasifikasi konseptual aplikasi sistem informasi. Sistem informasi dikategorikan dalam cara ini untuk menekankan peran-peran utama yang dimainkan setiap sistem dalam operasi dan manajemen suatu bisnis. Beberapa contoh kategori sistem informasi (O'Brien, 2005):

1. Sistem Pendukung Operasi

Sistem informasi selalu dibutuhkan untuk memproses data yang dihasilkan oleh, dan digunakan dalam operasi bisnis. Sistem pendukung operasi semacam ini menghasilkan berbagai produk informasi yang paling dapat digunakan oleh para manajer.

a. Sistem Pemrosesan Transaksi

Memproses data yang dihasilkan dari transaksi bisnis, memperbaharui *database* operasional, dan menghasilkan dokumen bisnis.

b. Sistem Pengendalian Proses

Mengawasi dan mengendalikan berbagai proses industrial.

c. Sistem Kerja sama Perusahaan

Mendukung komunikasi dan kerja sama tim, kelompok kerja, dan perusahaan.

2. Sistem Pendukung Manajemen

Ketika aplikasi sistem informasi berfokus pada penyediaan informasi dan dukungan untuk pengambilan keputusan yang efektif oleh para manajer, aplikasi sistem tersebut akan disebut sebagai sistem pendukung manajemen.

a. Sistem Informasi Manajemen

Memberikan informasi dalam bentuk laporan yang telah ditentukan sebelumnya untuk mendukung pengambilan keputusan bisnis.

b. *Decision Support System*

Memberikan dukungan interaktif khusus untuk proses pengambilan keputusan para manajer dan praktisi bisnis lainnya.

c. *Executive Information System*

Memberikan informasi penting dari SIM, DSS, dan sumber lainnya yang dibentuk sesuai kebutuhan informasi para eksekutif.

3. Kategori lainnya sistem informasi

a. Sistem Pakar

Sistem berbasis pengetahuan yang menyediakan saran pakar dan bertindak sebagai konsultan pakar bagi para pemakai.

b. Sistem Manajemen Pengetahuan

Sistem berbasis pengetahuan yang mendukung pembuatan, pengaturan dan penyebaran pengetahuan bisnis dalam perusahaan.

c. Sistem Informasi Strategis

Mendukung operasi dan proses manajemen yang memberi perusahaan produk, layanan dan kemampuan strategis sebagai keunggulan kompetitif.

d. Sistem Bisnis Fungsional

Mendukung berbagai aplikasi operasional dan manajerial atas berbagai fungsi bisnis perusahaan.

2.3 Pengertian Persediaan

Persediaan adalah item atau material yang dipakai oleh suatu organisasi atau perusahaan untuk menjalankan bisnisnya. Jika perusahaan tersebut memproduksi suatu barang atau jasa maka material tersebut digunakan untuk mendukung atau menyediakan kebutuhan produksi.

Menurut Ristono (2009) persediaan adalah suatu teknik untuk manajemen material yang berkaitan dengan persediaan. Definisi persediaan secara umum adalah sebagai berikut:

1. Suatu item yang disimpan untuk memenuhi kebutuhan yang akan datang.
2. Suatu item yang harus ditentukan kapan harus dibutuhkan.
3. Suatu item yang harus ditentukan berapa banyak yang harus dibutuhkan.
4. Suatu item dengan seberapa jauh harus di-maintain.

Jenis material yang dipakai dalam bisnis perusahaan seperti *raw material*, *work in proses*, material yang digunakan untuk kepentingan produksi seperti *operating supplies*, *maintenance*, item yang bisa langsung dikonsumsi oleh pelanggan seperti *finished good*. Untuk itulah persediaan dalam industri manapun menjadi hal yang perlu diperhatikan. Persediaan yang ideal akan memenuhi syarat-syarat sebagai berikut:

1. Peningkatan layanan terhadap pelanggan, melalui pemberian layanan berupa penyediaan bahan baku atau barang yang dibutuhkan pelanggan (*service availability*).
2. Penekanan biaya persediaan tidak hanya sekedar menyediakan bahan atau barang sesuai kebutuhan saja, tetapi harus mempertimbangkan hal-hal lain seperti ketetapan waktu, ketetapan mutu, biaya yang ekonomis dan ketetapan jumlah.

2.3.1 Fungsi Persediaan

Persediaan dapat memiliki berbagai fungsi penting menambah fleksibilitas dari operasi suatu perusahaan. Fungsi dasar persediaan sebenarnya sangat sederhana, yaitu meningkatkan *profitability* perusahaan.

Bagi sebagian perusahaan kebijakan persediaan yang aman adalah memiliki persediaan dalam jumlah banyak, tetapi ternyata hal ini akan menyebabkan tingginya biaya untuk penyimpanan dan pembelian bahan atau barang yang bersangkutan, sedangkan kelebihan persediaan juga akan menyebabkan banyaknya dana yang terserap dalam persediaan sehingga tidak efisien. Sebaliknya, bila persediaan terlalu sedikit akan berisiko kekurangan bahan atau barang. Hal ini akan mengganggu kelancaran produksi, selain itu juga biaya pembelian dan biaya persediaan juga semakin besar

Selain fungsi dasar persediaan, ada beberapa fungsi persediaan yang lainnya, yakni fungsi wilayah, fungsi *decoupling*, dan fungsi penyeimbang dengan permintaan dan fungsi penyangga.

1. Fungsi pemisahan wilayah, merupakan spesialisasi ekonomis antara unit pembuatan (*manufacturing*) dan unit distribusi yang dibagikan dalam wilayah-wilayah yang ditangani.
2. Fungsi *decoupling*, merupakan fungsi suatu produk yang diproses dan di distribusikan dalam ukuran yang ekonomis.
3. Fungsi penyeimbang dengan permintaan merupakan fungsi untuk kebutuhan konsumsi agar dapat dipenuhi dengan lancar bagi proses produksi.

2.3.2 Jenis-Jenis Persediaan

Tersine (1994) mengemukakan bahwa persediaan terdiri dari *raw material*, *good in process* dan *finished good*. *Raw material* adalah item yang dibeli dari pemasok untuk digunakan sebagai input pada proses produksi. Mereka akan memodifikasi dan merubah bentuk sampai kepada barang jadi. Jenisnya seperti: kayu, zat warna, lem, sekrup, paku, dan cat. *Good in process* adalah bagian produk akhir yang akan diselesaikan dan masih dalam proses produksi. *Finished good* adalah produk akhir yang tersedia untuk penjualan, distribusi dan penyimpanan.

2.3.3 Penyebab Timbulnya Persediaan

Persediaan merupakan sesuatu yang sulit dihindari. Penyebab timbulnya persediaan adalah sebagai berikut (Baroto, 2002):

1. Mekanisme pemenuhan atas permintaan. Permintaan terhadap suatu barang tidak dapat dipenuhi seketika bila barang tersebut tidak tersedia sebelumnya. Untuk menyiapkan barang ini diperlukan waktu untuk pembuatan dan pengiriman, maka adanya persediaan merupakan hal yang membantu permintaan terhadap barang.
2. Keinginan untuk mengurangi ketidakpastian. Ketidakpastian terjadi akibat permintaan yang bervariasi dan tidak pasti dalam jumlah maupun waktu kedatangan, waktu pembuatan cenderung tidak konstan antara satu produk dengan produk berikutnya, waktu tenggang (*lead time*) yang cenderung tidak pasti karena banyak faktor yang tidak dapat dikendalikan. Ketidakpastian ini dapat dikurangi dengan persediaan. Keinginan melakukan spekulasi yang bertujuan mendapatkan keuntungan besar dari kenaikan harga di masa mendatang.

2.3.4 Faktor-Faktor yang Menentukan Persediaan

Yang menjadi masalah bagi perusahaan adalah bagaimana menentukan persediaan yang optimal, oleh karena itu perlu diketahui faktor-faktor yang

mempengaruhi besar kecilnya persediaan. Menurut Ristono (2009) faktor-faktor yang menentukan persediaan adalah sebagai berikut:

1. *Volume* atau jumlah yang dibutuhkan, yaitu yang dimaksudkan untuk menjaga kelangsungan (kontinuitas) proses produksi. Semakin banyak jumlah bahan baku yang dibutuhkan, maka akan semakin besar tingkat persediaan bahan baku.
2. Kontinuitas produksi tidak terhenti, diperlukan tingkat persediaan bahan baku yang tinggi dan sebaliknya.
3. Sifat bahan baku/penolong, apakah cepat rusak (*durable good*) atau tahan lama (*udurable good*).

2.3.5 *Reorder Point* (Titik Pemesanan Kembali)

Reorder point adalah titik atau tingkat persediaan, dimana pemesanan kembali harus dilakukan. Pemesanan harus dilakukan kembali sehingga kedatangan atau penerimaan bahan baku yang dipesan tepat pada tingkat *safety stock*. Model persediaan sederhana mengasumsikan bahwa penerimaan suatu pesanan bersifat seketika. Artinya, model persediaan mengasumsikan bahwa suatu perusahaan akan menunggu sampai tingkat persediaannya mencapai nol, sebelum perusahaan memesan kembali dan dengan seketika kiriman yang dipesan akan diterima. Akan tetapi, waktu antara dilakukannya pemesanan atau waktu pengiriman bisa cepat atau lambat, sehingga perlu ditetapkan kapan harus dilakukan pemesanan ulang. ROP harus dihitung secara cermat dan tepat, untuk menjaga keseimbangan persediaan serta perusahaan tidak kehabisan bahan jika sewaktu-waktu terdapat jumlah pesanan atau produk yang lebih besar.

Rumus *Reorder Point* adalah:

$$\text{ROP} = (\text{L} \times \text{d}) + \text{SS}$$

Keterangan:

ROP = *Reorder Point* atau titik pemesanan kembali

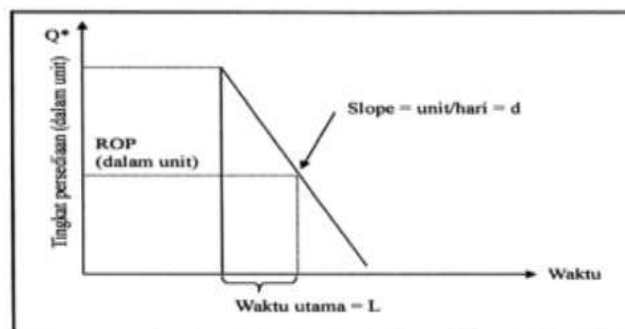
d = *Demand* (jumlah permintaan/penggunaan perhari)

L = *Lead Time*

SS = *Safety Stock*

Permintaan perhari d , dicari dengan membagi permintaan tahunan (D), dengan jumlah periode yang digunakan (bisa hari dalam setahun, bulan dalam setahun tergantung periode yang digunakan perusahaan)

$$d = \frac{D}{\text{Jumlah periode waktu per tahun}}$$



Gambar II.2 Kurva Titik Pemesanan Kembali
(Sumber: Siagian, 2005)

Dalam pelaksanaannya, ternyata perusahaan-perusahaan menetapkan beberapa kebijakan dalam menentukan titik pemesanan kembali

1. Menetapkan jumlah penggunaan selama *lead time*, yaitu waktu mulai barang dipesan sampai barang datang ditambah persentase tertentu sebagai persediaan pengaman.

Contoh:

Suatu perusahaan elektronik memerlukan bahan baku per hari sebanyak 500 unit dengan waktu tunggu 4 hari. Jika kebutuhan pengamannya ditetapkan sebesar 50% dari kebutuhan per hari, tentukan titik pemesanan kembali!

Jawaban:

Kebutuhan per hari = 500 unit

Lead Time = 4 hari

Safety stock = 50% dari kebutuhan

$ROP = (4 \times 500 \text{ unit}) + 50\% (4 \times 500 \text{ unit})$

$$= 2000 \text{ unit} + 1000 \text{ unit} = 3000 \text{ unit}$$

Jadi pemesanan akan dilakukan kembali jika persediaan mencapai 3000 unit

2. Menetapkan jumlah penggunaan selama *lead time* ditambah penggunaan selama periode tertentu sebagai *safety stock*

Contoh:

Suatu perusahaan elektronik memerlukan bahan baku per hari sebanyak 500 unit dengan waktu tunggu 4 hari. Jika kebutuhan pengamannya ditetapkan kebutuhan selama 3 hari, tentukan titik pemesanan kembali!

Jawaban:

Kebutuhan per hari = 500 unit

Lead Time = 4 hari

Safety stock = kebutuhan selama 3 hari

ROP = $(4 \times 500 \text{ unit}) + (3 \times 500 \text{ unit})$

$$= 2000 \text{ unit} + 1500 \text{ unit} = 3500 \text{ unit}$$

Jadi pemesanan akan dilakukan kembali jika persediaan mencapai 3500 unit.

Ada 2 faktor yang menentukan *Reorder Point*, yaitu:

1. **Lead Time (Waktu Tunggu)**

Lead time merupakan waktu yang dibutuhkan antara pemesanan dengan barang sampai di perusahaan, sehingga *lead time* berhubungan dengan *Re-Order Point* dan saat penerimaan barang. *Lead Time* muncul karena setiap pesanan membutuhkan waktu dan tidak semua pesanan bisa dipenuhi seketika, sehingga selalu ada jeda waktu. Dalam ROP, *lead time* diasumsikan konstan artinya dari waktu ke waktu selalu tetap misal *lead time* 5 hari, maka akan berulang dalam setiap periode. Akan tetapi dalam prakteknya *lead time* banyak berubah-ubah, untuk mengantisipasinya perusahaan sering menyediakan *safety stock*. Faktor waktu sangatlah penting dalam pengisian kembali persediaan karena terdapat perbedaan waktu yang kadang cukup lama saat mengadakan pesanan untuk menggantikan atau pengisian kembali persediaan.

2. **Safety Stock (Persediaan Pengaman)**

Safety stock (juga disebut *buffer stock*) adalah istilah yang digunakan oleh logistik untuk menggambarkan tingkat stok ekstra yang dipertahankan untuk mengurangi risiko *stock out* (kekurangan bahan baku atau kemasan) karena ketidakpastian pasokan dan permintaan. Keuntungan adanya *safety stock* adalah pada saat jumlah permintaan mengalami lonjakan, maka persediaan pengaman dapat digunakan untuk menutup permintaan tersebut. Pentingnya *safety stock* disebabkan oleh karena kerugian yang akan ditanggung oleh perusahaan karena proses terhenti.

Ada 3 komponen yang menjadi pertimbangan dalam menentukan *safety stock*:

a. Variasi permintaan.

Sangat jarang sekali kita menemukan kasus dimana permintaan itu stabil apalagi sama tiap bulannya. Selalu ada variasi permintaan. Logikanya semakin tinggi variasi permintaan dari waktu ke waktu, pasti peluang untuk terjadi *stock out* (kekurangan persediaan saat ada permintaan) akan semakin besar. Oleh karena itu, faktor variasi permintaan ini pun harus berbanding lurus dengan *safety stock* yang harus kita siapkan.

b. *Lead time*.

Ada berbagai macam *lead time* mulai dari *lead time* produksi, *leadtime* transportasi, *leadtime* inspeksi, dan atau *leadtime* yang lain bergantung terminologi tiap-tiap perusahaan. Yang jelas sejak suatu produk dipesan hingga diantar kepada yang memesan, waktu yang dibutuhkan juga bervariasi.

c. *Service level*.

Setiap perusahaan perlu menetapkan berapa *service level* yang diberikan kepada pelanggannya. Secara sederhana, kalau ada 100 permintaan, berapa banyak yang dapat kita tolerir untuk tidak terpenuhi? jika hanya 5, maka *service level* kita adalah 95%, idealnya

memang 100%, tetapi itu berarti kita harus menyediakan *safety stock* yang sangat besar. Karena *safety stock* adalah *inventory*, maka uang yang tertanam di situ harus diperhatikan.

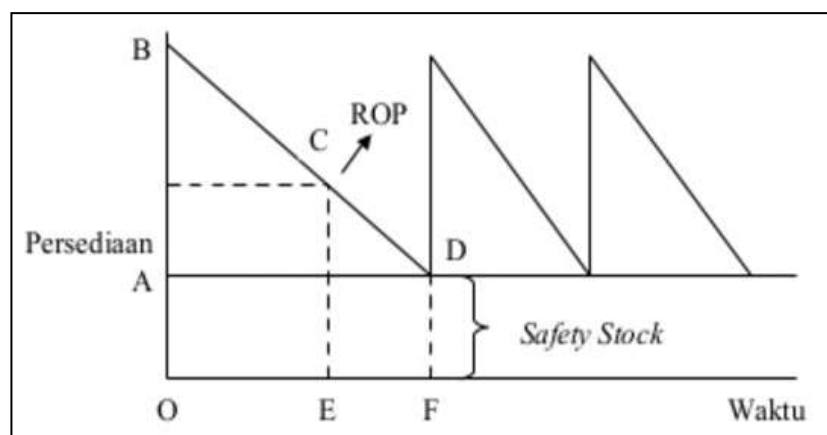
Rumus *safety stock* adalah:

(Pemakaian Maksimum – Pemakaian Rata-Rata) Lead Time

Metode ini dilakukan dengan menghitung selisih antara pemakaian maksimum dengan pemakaian rata-rata dalam jangka waktu tertentu (misalnya perminggu), kemudian selisih tersebut dikalikan dengan lead time, contoh:

PT. ABC memperkirakan pemakaian maksimum bahan-bahan perminggu sebesar 650 kg, sedangkan pemakaian rata-ratanya sebesar 500 kg dan lamanya *lead time* 2 minggu, maka data-data tersebut *safety stock* sebesar:

$$\begin{aligned}\text{Safety Stock} &= (650 - 500) 2 \\ &= 300 \text{ Kg}\end{aligned}$$



Gambar II.3 Hubungan antara ROP, *Safety Stock*, dan *Lead Time*

(Sumber: Siagian, 2005)

2.3.6 Metode dalam Gudang

Pengolahan barang di gudang memiliki metode tertentu guna meminimalisir biaya penyimpanan tapi juga seimbang dengan biaya pemesanan. Berikut merupakan metode yang digunakan (Yunarto dan Santika, 2005):

1. Metode FIFO (*First In First Out*), yaitu sistem pengeluaran barang yang terlebih dahulu masuk ke gudang, didahulukan keluar. Biasanya sistem seperti ini banyak dipakai di perusahaan manufaktur makanan karena untuk menghindari bahan baku busuk di dalam gudang. Namun tidak menutup kemungkinan digunakan juga untuk perusahaan manufaktur selain makanan, untuk menjaga kualitas bahan baku tidak rusak di dalam gudang. Singkatnya, sistem ini seperti sistem antrian (*queue*).
2. Metode LIFO (*Last In First Out*), yaitu sistem pengeluaran barang dari gudang dimana barang yang terakhir masuk adalah yang pertama dikeluarkan. Biasanya sistem seperti ini tidak banyak digunakan perusahaan manufaktur, namun banyak digunakan di perusahaan dagang yang mengutamakan kesegaran barang jual. Singkatnya sistem ini seperti sistem tumpukan (*stack*).
3. Metode FEFO (*First Expired Date First Out*), yaitu barang persediaan yang pertama kadaluarsa harus yang pertama keluar untuk didistribusikan. Dalam penyusunan persediaan yang masa kadaluarsanya lebih awal atau yang diterima lebih awal biasanya juga diproduksi lebih awal dan umurnya relatif lebih tahan masa kadaluarsanya mungkin lebih awal. Adapun kendala yang biasa dihadapi dalam pengelolaan persediaan di dalam gudang adalah akurasi pergerakan barang dan penjadwalan titik pemesanan barang. Sehingga pada pengaplikasiannya dibutuhkan kontrol aktifitas pergerakan barang dan dokumen untuk meningkatkan efisiensi penggunaan persediaan agar jumlah dan rentang waktu barang dipesan tepat waktu atau sesuai perencanaan dan dapat selalu memenuhi kebutuhan produksi.

2.4 Pengertian Bahan Baku

Bahan baku merupakan bahan langsung (*direct material*) yaitu bahan yang membentuk suatu kesatuan yang tidak terpisahkan dari produk jadi. Bahan baku adalah bahan utama atau bahan pokok dan merupakan komponen utama dari suatu produk. Bahan baku biasanya lebih mudah ditelusuri dalam suatu produk dan harganya relatif tinggi dibandingkan dengan bahan pembantu. Misalnya produk kursi rotan, seperti: paku, lem kayu, dan lain-lain. Bahan pembantu (*indirect material*) merupakan bahan pelengkap yang melekat pada suatu produk. Bahan pembantu biasanya tidak mudah ditelusuri dalam suatu produk dan harganya relatif rendah dibandingkan dengan bahan baku (Nafarin, 2007).

2.4.1 Sistem Bahan Baku

Pada proses penerimaan bahan baku dapat di *assign* tiga macam informasi *lot*, yaitu (Yunarto dan Santika, 2005):

1. *Receiving/ use lot*

Receiving/ use lot adalah *lot* internal perusahaan.

2. *Vendor lot*

Vendor lot adalah nomor *lot* internal *vendor/ supplier* digunakan sebagai referensi *lot* sehingga jika terjadi masalah kualitas masalah dapat dilacak sampai ke *supplier*.

3. *Purchase order lot*

Purchase order lot adalah nomor *lot* yang biasanya nilainya sama dengan nomor *purchase order*. Pada proses penerimaan barang ini dapat diisikan informasi *lot* dan juga dapat diisikan informasi *serial number*.

2.5 Bahasa Pemrograman

2.5.1 PHP (*Hypertext Preprocessor*)

PHP (*Hypertext Preprocessor*) adalah bahasa *server-side scripting* yang digunakan untuk aplikasi web yang dinamis dan interaktif. Sebuah halaman PHP

adalah sebuah halaman HTML yang memiliki *server-side scripts* yang ditempatkan dalam *server* dan diproses oleh *web server* sebelum dikirim ke *browser* pemakai (Welling dan Thomson, 2003).

Server-side scripts dijalankan ketika *browser* melakukan permintaan *file.php* dari *server*. PHP dipanggil oleh *web server*, dimana proses script perintah yang ada di suatu halaman dieksekusi mulai dari awal sampai akhir di dalam mesin PHP. Setelah *script* PHP tersebut diolah, hasilnya akan ditampilkan kepada *client* melalui *web browser* berupa tampilan HTML. Menurut Welling dan Thomson (2003), beberapa keunggulan PHP adalah:

1. *High Performance*

PHP sangat efisien. Dengan menggunakan *server* tunggal yang tidak mahal, user dapat melakukan banyak pekerjaan setiap harinya.

2. *Database Integration*

PHP mempunyai sambungan ke banyak sistem basis data, antara lain MySQL, PostgreSQL, Oracle, Informix, dan Sysbase Databases.

3. *Built-in-Libraries*

PHP dirancang khusus untuk web, dan mempunyai banyak *built-in-function* untuk menampilkan banyak fungsi di dalam web.

4. Harga yang murah

PHP adalah perangkat lunak gratis.

5. Mudah dalam pembelajaran dan penggunaan

Sintaks PHP berdasarkan bahasa pemrograman lainnya, terutama C dan Java.

6. *Portability*

PHP dapat digunakan di banyak sistem operasi yang berbeda.

7. Ketersediaan *Source Code*

Kode PHP dapat langsung diakses dan dimodifikasi secara bebas.

2.5.2 XAMPP

XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolahan data MySQL di komputer lokal. XAMPP berperan sebagai *web server* pada komputer. XAMPP juga dapat disebut sebagai sebuah *CPanel server virtual*, yang dapat membantu melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses dengan internet (Kadir, 2008).

2.6 Perancangan Database

2.6.1 Pengertian Database

Kumpulan file-file yang saling berhubungan atau berelasi sehingga membentuk suatu basis data. Basis data adalah sekumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, tanpa adanya suatu kerangkapan data, sehingga mudah untuk digunakan kembali, dapat digunakan oleh satu atau lebih program aplikasi secara optimal, data disimpan tanpa mengalami ketergantungan pada program yang akan menggunakannya, data disimpan sedemikian rupa sehingga apabila ada penambahan, pengambilan, dan modifikasi data dapat dilakukan dengan mudah dan terkontrol. Dari pengertian tersebut dapat disimpulkan bahwa basis data mempunyai beberapa kriteria yang penting yaitu :

1. Bersifat data oriented dan bukan program oriented.
2. Dapat digunakan oleh beberapa program aplikasi tanpa perlu mengubah basis datanya.
3. Dapat berkembang dengan mudah, baik volume maupun strukturnya.
4. Dapat memenuhi sistem-sistem baru secara mudah.
5. Dapat digunakan dengan cara yang berbeda-beda.

2.6.2 Normalisasi

Proses pengelompokan data ke dalam bentuk tabel atau relasi atau file untuk menyatakan entitas dan hubungan mereka sehingga terwujud satu bentuk database yang mudah untuk dimodifikasi. Pada proses normalisasi selalu diuji beberapa kondisi. Apakah ada kesulitan selama proses *delete*, *insert* atau *update*

dalam basis data yang disebut penyimpangan-penyimpangan atau dikenal dengan *Anomaly*.

2.6.3 Relational Database Management Systems (RDMS)

RDBMS (*Relationship Database Management System*) adalah sebuah DBMS yang mendukung adanya hubungan relationship antara tabel. MySQL adalah salah satu RDBMS yang paling populer digunakan di seluruh dunia karna selain gratis MySQL juga stabil dan memiliki fitur yang lumayan lengkap.

Suatu database terdiri dari banyak tabel dan tabel ini terdiri dari banyak field yang merupakan kolomnya. Isi tiap baris dari tabel inilah merupakan data. Sedangkan untuk membuat data saling terhubung dan terintegrasi, maka satu tabel dengan tabel yang lainnya harus memiliki hubungan antara satu dengan yang lainnya. Setiap tabel mempunyai sebuah primary key, lalu primary key ini dihubungkan dengan tabel kedua dan menjadi foreign key di tabel kedua ini.

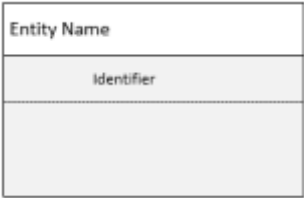
Dengan konsep seperti inilah data akan saling terhubung dan disebut relational, karna data dalam tabel yang satu terhubung dengan tabel yang lainnya.

2.6.4 Permodelan Data

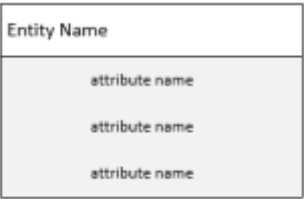
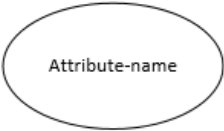
Model data adalah cara formal untuk merepresentasikan data yang digunakan dan diciptakan oleh sistem bisnis. Model data menggambarkan data yang mengalir melalui proses bisnis dalam suatu organisasi. Ini menggambarkan orang, tempat, atau hal-hal tentang informasi yang ditangkap dan bagaimana hubungan mereka satu sama lain. Model data digambar dengan proses iteratif dimana model menjadi lebih rinci dan kurang konseptual dari waktu ke waktu. Selama analisis, analis menggambar model data logis, yang menunjukkan kumpulan data logis tanpa menunjukkan bagaimana data disimpan, dibuat, atau dimanipulasi, karena model ini bebas dari implementasi atau rincian teknis, para analis dapat lebih fokus pada pencocokan diagram dengan kebutuhan bisnis yang sebenarnya dari sistem. Pada tahap perancangan, analis menggambar model data fisik untuk mencerminkan bagaimanadata secara fisik akan disimpan dalam database dan file. (Dennis et al, 2012).

Teknik pemodelan data yang paling umum digunakan di industri adalah *Entity Relationship Diagram*. *Entity Relationship Diagram* (ERD) adalah gambar yang menunjukkan informasi yang dibuat, disimpan, dan digunakan oleh sistem bisnis. Seorang analis dapat membaca ERD untuk menemukan kumpulan informasi individual dalam sebuah sistem dan bagaimana cara mereka mengatur dan berhubungan satu sama lain. Pada ERD, jenis informasi serupa tercantum bersama-sama dan ditempatkan di dalam kotak yang disebut entitas. Garis ditarik antar entitas untuk mewakili hubungan antar data, dan simbol khusus ditambahkan ke diagram untuk mengkomunikasikan aturan bisnis tingkat tinggi yang perlu didukung oleh sistem. Ada tiga elemen dasar dalam bahasa pemodelan data (entitas, atribut, dan hubungan), masing-masing diwakili oleh simbol grafis yang berbeda. Ada banyak set simbol yang berbeda yang dapat digunakan pada ERD. Elemen-elemen *Entity Relationship Diagram* terlihat pada Tabel II.1.

Tabel II.1 Elemen-Elemen *Entity Relationship Diagram*

N o	Elemen	Elemen	Fungsi Elemen
1.			ENTITY adalah seseorang, tempat, atau benda, memiliki nama tunggal yang tercetak di semua huruf besar, memiliki pengenalan, harus berisi lebih dari satu contoh data.

Tabel II.1 Elemen-Elemen *Entity Relationship Diagram* (Lanjutan)

N o	Elemen	Elemen	Fungsi Elemen
2.			ATTRIBUTE adalah milik suatu entitas. Digunakan setidaknya pada satu proses bisnis. Dipecah ke tingkat detail yang paling berguna.

3.	<u>Relationship-name</u>		RELATIONSHIP menunjukkan hubungan antara dua entitas. Memiliki entitas induk dan entitas anak. Dijelaskan dengan ungkapan kata kerja. Memiliki kardinalitas (1: 1, 1: N, atau M: N). Memiliki modalitas (null, bukan null). Tergantung atau mandiri.
----	--------------------------	---	--

(Sumber: Dennis et al, 2012)

2.6.5 MySQL

MySQL (*Structured Query Language*) adalah sebuah perangkat lunak sistem manajemen basis data (*database management system*) atau DBMS yang *multithread*, *multiuser*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public Licence* (GPL), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL (Solichin, 2010). Fitur-fitur MySQL antara lain (Solichin, 2010):

1. *Relational Database System*

Seperti halnya *software database* lain yang ada di pasaran, MySQL termasuk RDBMS (*Relational DataBase Management System*).

2. *Arsitektur Client-Server*

MySQL memiliki arsitektur *client-server* dimana *server database* MySQL terinstal di *server*. *Client* MySQL dapat berada di komputer yang sama dengan *server* dan dapat juga di komputer lain yang berkomunikasi dengan *server* melalui jaringan bahkan internet.

3. Mengetahui perintah SQL standar

SQL (*Structured Query Language*) merupakan suatu bahasa standar yang berlaku di hampir semua *software database*. MySQL mendukung SQL versi SQL:2003.

4. Mendukung *Sub Select*

Mulai versi 4.1 MySQL telah mendukung *select* dalam *select (sub select)*.

5. Mendukung Views

MySQL mendukung views sejak versi 5.0

6. Mendukung Stored Prosedured (SP)

MySQL mendukung SP sejak versi 5.0

7. Mendukung Triggers

MySQL mendukung trigger pada versi 5.0 namun masih terbatas. Pengembang MySQL berjanji akan meningkatkan kemampuan trigger pada versi 5.1.

8. Mendukung replication

9. Mendukung transaksi

10. Mendukung foreign key

11. Tersedia fungsi GIS

12. Free (bebas diunduh)

13. Stabil dan tangguh

14. Fleksibel dengan berbagai pemrograman

15. Security yang baik

16. Dukungan dari banyak komunitas

17. Perkembangan software yang cukup cepat

MySQL adalah suatu bahasa permintaan yang terstruktur dan telah distandarkan untuk semua program pengakses *database* seperti Oracle, PosgreSQL, SQL Server, dan lain-lain. (Kadir, 2008). Pada MySQL masing-masing tipe data memiliki kegunaan dan keterangan untuk digunakan dalam merancang tabel. Beberapa tipe data pada MySQL terlihat pada Tabel II.2.

Tabel II.2 Beberapa Tipe Data pada MySQL

Tipe Data	Ukuran	Keterangan
-----------	--------	------------

Tipe Data	Ukuran	Keterangan
CHAR	M	Menampung maksimal M karakter (kombinasi huruf, angka, dan simbol-simbol). Jumlah memori yang dibutuhkan selalu M byte. M terbesar adalah 255.
VARCHAR	M	Karakter yang disimpan maksimal M karakter. Jumlah memori yang dibutuhkan tergantung jumlah karakter. M bisa mencapai 65535.
DATE	8 byte	Menyatakan tanggal.
TIME	8 byte	Menyatakan waktu (jam:menit:detik).
TINYINT	1 byte	Bilangan antara -128 sampai dengan +127.
SMALLINT	2 byte	Bilangan antara -32768 sampai dengan +32768.
INT	4 byte	Bilangan antara -2147683648 sampai dengan +2147683647.
FLOAT	4 byte	Bilangan pecahan.
DOUBLE	8 byte	Bilangan pecahan dengan presisi tinggi.
BOOL	1 byte	Untuk menampung nilai <i>true</i> (benar) dan <i>false</i> (salah). Identik dengan TINYINT.
ENUM	-	Menyatakan suatu tipe yang nilainya tertentu (disebutkan dalam pendefinisian).
TEXT	-	Menyimpan teks yang ukurannya sangat panjang.
BLOB	-	Untuk menyimpan data biner (misalnya gambar atau suara).

(Sumber: Kadir, 2008)

2.7 Analisis Sistem Informasi

Analisis sistem adalah sebuah teknik pemecahan masalah yang menguraikan sebuah sistem menjadi bagian-bagian komponen dengan tujuan untuk mempelajari seberapa bagus bagian-bagian komponen tersebut dalam berinteraksi untuk meraih tujuan mereka.

1. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem terdiri dari dua kebutuhan, yaitu kebutuhan sistem fungsional dan non fungsional.

a) Kebutuhan Sistem Fungsional

- Menggambarkan fungsionalitas sistem atau layanan-layanan system
- Sangat bergantung dari jenis perangkat lunak, pengguna sistem, dan jenis sistem dimana perangkat lunak tersebut digunakan.
- Kebutuhan fungsional dapat berupa pernyataan-pernyataan tingkat tinggi, yaitu:
 - a. Apa yang sistem harus lakukan?
 - b. Harus dapat menggambarkan layanan-layanan yang dapat diberikan oleh sistem kepada pengguna secara mendetail.

b) Kebutuhan Sistem Non Fungsional

Kebutuhan non fungsional menempatkan batasan pada produk yang sedang dikembangkan, proses pengembangannya, dan menentukan batasan-batasan eksternal yang harus dipenuhi oleh produk tersebut. Beberapa kategori non fungsional, meliputi keselamatan (*safety*) dan keamanan (*security*), ketergunaan (*usability*), reliabilitas, dan performansi (Pinandito, 2012).

2.8 Analisis Sistem Informasi

Analisis sistem informasi merupakan langkah untuk menentukan seberapa jauh sistem yang berjalan telah mencapai sarannya dan mengetahui kelemahan-kelemahannya sehingga dapat diusulkan perbaikan.

Dalam buku *Analisa Sistem Informasi* oleh Sutabri (2004), berikut ini merupakan pengertian analisis sistem menurut para ahli, yaitu:

1. Menurut Mc Leod (2004), analisis sistem adalah penelitian atas sistem yang telah ada dengan tujuan merancang sistem yang baru.
2. Menurut Laudon (2007), analisis sistem adalah memeriksa sebuah masalah yang ada yang akan diselesaikan oleh perusahaan dengan menggunakan

sistem informasi. Analisis sistem mencakup beberapa langkah yang harus dilakukan, yaitu:

- a. Menentukan masalah.
 - b. Mengidentifikasi penyebab dari masalah tersebut.
 - c. Menentukan pemecahan masalahnya.
 - d. Mengidentifikasi kebutuhan informasi yang dibutuhkan untuk memecahkan masalah tersebut.
3. Menurut Bentley dan Whitten (2007), Analisis sistem adalah suatu teknik untuk menyelesaikan suatu masalah yang ada pada suatu sistem dengan cara membagi masalah tersebut ke beberapa bagian dengan maksud agar mudah dicari penyelesaiannya.

Langkah-langkah yang harus dilakukan dalam menganalisis sistem menurut Jogiyanto (2005), yaitu:

1. *Identify*

Mengidentifikasi masalah merupakan langkah pertama yang dilakukan dalam tahap analisis. Masalah dapat didefinisikan sebagai suatu pertanyaan yang diinginkan untuk dipecahkan. Masalah inilah yang menyebabkan sasaran dari sistem tidak dapat dicapai. Oleh karena itulah pada tahap analisis sistem, langkah yang pertama dilakukan adalah mengidentifikasi terlebih dahulu masalah-masalah yang terjadi.

2. *Understand*

Langkah kedua dari tahap analisis sistem adalah memahami sistem yang ada. Langkah ini dapat dilakukan dengan mempelajari operasi secara terperinci bagaimana sistem yang ada beroperasi. Untuk mempelajari operasi yang berjalan diperlukan data yang dapat diperoleh dengan cara melakukan penelitian.

3. *Analyze*

Langkah ini dilakukan berdasarkan data yang diperoleh dari hasil penelitian yang telah dilakukan.

4. *Report*

Setelah proses analisis sistem selesai dilakukan, langkah terakhir adalah membuat laporan hasil analisis sistem yang menjelaskan temuan-temuan yang dapat dijadikan acuan alternatif penyelesaian masalah.

2.8 Metode Pengembangan Sistem

Pengembangan sistem dapat berarti menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada. Sistem yang lama perlu diperbaiki oleh karena beberapa hal, yaitu (Sutabri, 2003):

1. Munculnya masalah pada sistem yang lama.
2. Untuk meraih kesempatan.
3. Adanya instruksi.

Proses perancangan atau pengembangan sistem informasi, mulai dari konsep sampai dengan implementasinya disebut dengan istilah *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* (SDLC) adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya berdasarkan *best practice* atau cara-cara yang sudah teruji baik (Rosa dan Shalahuddin, 2011).

Menurut Rosa dan Shalahuddin (2011), tahapan-tahapan yang ada pada SDLC secara global 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 *user*. Membuat 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*)
Mendeskrripsikan pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan produksi (lingkungan pada *user*), termasuk implementasi akhir dan masuk pada proses peninjauan.

10. Disposisi (*disposition*)

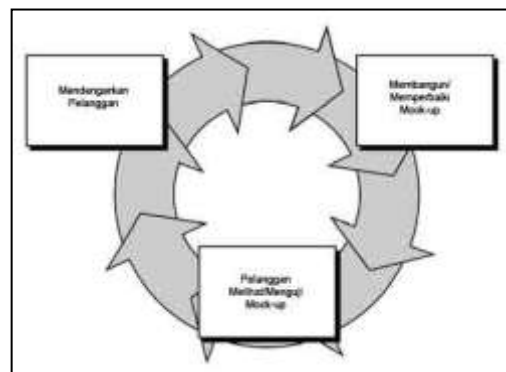
Mendesripsikan aktifitas akhir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktifitas *user*.

2.8.1 Model *Prototype*

Model *prototype* dapat digunakan untuk menyambungkan ketidakpahaman pelanggan mengenai hal teknis yang memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak (Rosa dan Shalahuddin, 2011).

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

1. Mengumpulkan kebutuhan pelanggan 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*.



Gambar II.4 Ilustrasi

(Sumber: Rosa dan Shalahuddin, 2011)

Model *Prototype*

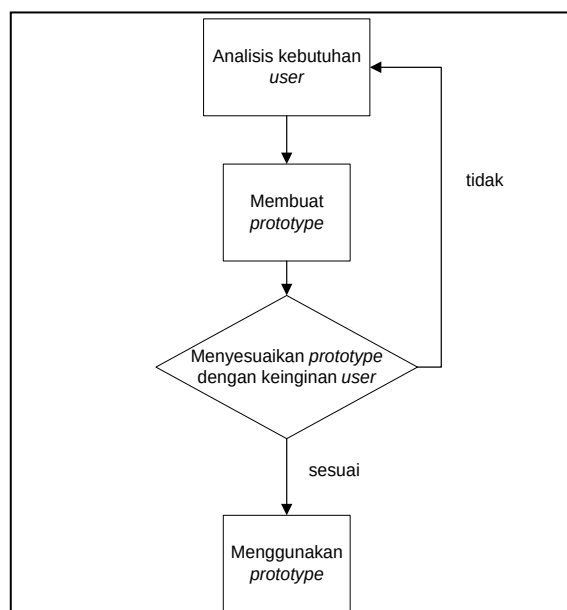
Mock-up adalah sesuatu yang digunakan sebagai model desain yang digunakan untuk mengajar, demonstrasi, evaluasi desain, promosi atau keperluan lain. Sebuah *mock-up* disebut sebagai *prototype* perangkat lunak jika menyediakan atau mampu mendemonstrasikan sebagian besar fungsi sistem

perangkat lunak dan memungkinkan pengujian desain sistem perangkat lunak. Iterasi terjadi pada pembuatan *prototype* sampai sesuai dengan keinginan pelanggan atau *user* (Rosa dan Shalahuddin, 2011). Kelemahan model *prototype* adalah sebagai berikut (Rosa dan Shalahuddin, 2011):

1. *User* dapat sering mengubah-ubah atau menambah spesifikasi kebutuhan karena menganggap aplikasi sudah dengan cepat dikembangkan, karena adanya iterasi ini dapat menyebabkan pengembang banyak mengalah dengan *user* karena perubahan atau penambahan spesifikasi kebutuhan perangkat lunak.
2. Pengembang lebih sering mengambil kompromi dengan pelanggan untuk mendapatkan *prototype* dengan waktu yang cepat sehingga pengembang lebih sering melakukan segala cara (tanpa idealis) guna menghasilkan *prototype* untuk didemonstrasikan. Hal ini dapat menyebabkan kualitas perangkat lunak yang kurang baik atau bahkan menyebabkan iteratif tanpa akhir.

2.8.2 *Evolutionary Prototype*

Evolutionary prototype yaitu, *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 dari pada dibuang. *Prototype* yang dibuang biasanya digunakan dengan aspek sistem yang dimengerti secara luas dan dibangun atas kekuatan tahapan *evolutionary prototype* (McLeod, 2008).



Gambar II.5 *Evolutionary Prototype Model*
(Sumber: McLeod, 2008)

1. Analisis kebutuhan *user*, pengembang dan *user* atau pemilik sistem melakukan diskusi dimana *user* atau pemilik sistem menjelaskan kepada pengembang tentang kebutuhan sistem yang mereka inginkan.
2. Membuat *prototype*, pengembang membuat *prototype* dari sistem yang telah dijelaskan oleh *user* atau pemilik sistem.
3. Menyesuaikan *prototype* dengan keinginan *user* atau pemilik sistem, pengembang menanyakan kepada *user* atau pemilik sistem tentang *prototype* yang sudah dibuat, apakah sesuai atau tidak dengan kebutuhan sistem.
4. Menggunakan *prototype*, sistem mulai dikembangkan dengan *prototype* yang sudah dibuat.

2.9 *Flowmap dan Flowchart*

Flowmap adalah diagram yang menunjukkan aliran data berupa formulir-formulir ataupun keterangan berupa dokumentasi yang mengalir atau beredar dalam suatu sistem. *Flowchart* adalah bagan yang menunjukkan alir di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. Notasi yang digunakan dalam suatu *flowmap* merupakan penggabungan notasi *flowchart* program.

Pada waktu akan menggambar suatu bagan alir, analis sistem atau pemrogram dapat mengikuti pedoman-pedoman sebagai berikut (Jerry, 2007):

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 digunakan suatu kata yang mewakili suatu pekerjaan, misal:
 - a. "Persiapkan" dokumen bb *request*
 - b. "Hitung" persediaan bahan baku
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. Gunakan simbol-simbol bagan alir yang standar.

2.10 Jenis-Jenis *Flowchart*

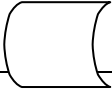
Terdapat beberapa jenis bagan alir (*flowchart*), yaitu sebagai berikut (Jogiyanto, 2005):

1. Bagan alir sistem (*systems flowchart*)
2. Bagan alir dokumen (*document flowchart*)
3. Bagan alir skematik (*schematic flowchart*)
4. Bagan alir program (*program flowchart*)
5. Bagan alir proses (*process flowchart*)

2.10.1 *Systems flowchart*

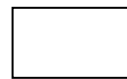
Systems flowchart merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem. *Systems flowchart* digambar dengan menggunakan simbol-simbol yang tampak pada Tabel II.3.

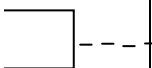
Tabel II.3 Simbol Bagan Alir Sistem (*Systems Flowchart*)

Simbol	Keterangan	Simbol	Keterangan
Dokumen 	Menunjukkan dokumen input dan output manual,	Diskette 	Menunjukkan input/output menggunakan

	mekanik atau komputer.		<i>diskette</i>
Kegiatan Manual 	Menunjukkan pekerjaan manual	Drum Magnetik 	Menunjukkan <i>input/output</i> menggunakan drum magnetic
Simpanan Offline N 	File <i>non</i> -komputer yang diarsip menurut angka (<i>Numerical</i>)	Pita Kertas Berlubang 	Menunjukkan <i>input/output</i> menggunakan pita kertas berlubang
Simpanan Offline A 	File <i>non</i> -komputer yang diarsip menurut huruf (<i>Alphabetical</i>)	Keyboard 	Menunjukkan <i>input</i> yang menggunakan <i>online keyboard</i>
Simpanan Offline C 	File <i>non</i> -komputer yang diarsip menurut Tanggal (<i>Cronological</i>)	Display 	Menunjukkan <i>output</i> yang ditampilkan di monitor

Tabel II.3 Simbol Bagan Alir Sistem (Systems Flowchart) Lanjutan

Simbol	Keterangan	Simbol	Keterangan
Kartu Plong 	<i>Input/output</i> yang menggunakan kartu plong (<i>punched card</i>)	Pita Kontrol 	Penggunaan pita kontrol dalam <i>batch control total</i> untuk pencocokan dalam <i>batch processing</i>
Proses 	Menunjukkan kegiatan proses dari operasi program komputer	Hubungan Komunikasi 	Menunjukkan proses transmisi data melalui <i>channel</i> komunikasi

Operasi Luar 	Menunjukkan operasi yang dilakukan di luar proses operasi computer	Garis Alir 	Menunjukkan arus dari proses
Pengurutan Offline 	Menunjukkan proses pengurutan data di luar proses komputer	Penjelasan 	Menunjukkan penjelasan dari suatu proses
Pita Magnetik 	Menunjukkan input/output menggunakan pita magnetic	Penghubung 	Menunjukkan penghubung ke halaman yang masih sama atau ke halaman lain
Hard Disk 	Menunjukkan input/output menggunakan hard disk		

(Sumber: Jogiyanto, 2005)

2.10.2 Document Flowchart

Bagan alir dokumen (*document flowchart*) atau disebut juga bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusan. Bagan alir dokumen menggunakan simbol-simbol yang sama dengan yang digunakan di dalam bagan alir sistem (*systems flowchart*).

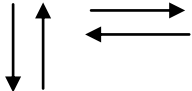
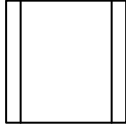
2.10.3 Schematic Flowchart

Bagan alir skematik (*schematic flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem. Perbedaan bagan alir skematik dari bagan alir sistem adalah selain menggunakan simbol-simbol bagan alir sistem digunakan pula gambar-gambar komputer dan peralatan lain. Maksud dari penggunaan gambar tersebut adalah untuk memudahkan komunikasi kepada orang yang kurang paham dengan simbol-simbol bagan alir. Penggunaan gambar mudah untuk dipahami, tetapi sulit serta lama pada pengerjaan.

2.10.4 Program Flowchart

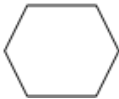
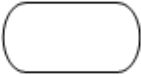
Bagan alir program (*program flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dibuat dari derivikasi bagan alir sistem. Bagan alir program dibuat dengan menggunakan simbol yang terdapat pada Tabel II.4.

Tabel II.4 Simbol Bagan Alir Program (*Program Flowchart*)

Simbol	Nama	Keterangan
	<i>Input/ output</i>	Simbol <i>input/ output</i> digunakan untuk mewakili data <i>input/ output</i>
	Proses	Simbol proses digunakan untuk mewakili suatu proses
	Garis Alir	Simbol garis alir digunakan untuk menunjukkan arus dari proses
	Penghubung	Simbol penghubung digunakan untuk menunjukkan sambungan dari bagan alir yang terputus dihalaman yang masih sama atau dihalaman lain
	Keputusan	Simbol keputusan digunakan untuk suatu penyeleksian kondisi di dalam program
	Proses terdefinisi	Simbol proses terdefinisi digunakan untuk menunjukkan suatu operasi yang rincian ditunjukkan di tempat lain

Tabel II.4 Simbol Bagan Alir Program (*Program Flowchart*) Lanjutan

Simbol	Nama	Keterangan
--------	------	------------

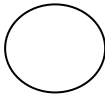
	Persiapan	Simbol persiapan digunakan untuk memberi nilai awal suatu sasaran
	Titik terminal	Simbol titik terminal digunakan untuk menunjukkan awal dan akhir dari suatu proses

(Sumber: Jogiyanto, 2005)

2.10.5 *Process Flowchart*

Bagan alir proses (*process flowchart*) merupakan bagan alir yang banyak digunakan di teknik industri. Bagan alir ini juga berguna bagi analisis sistem untuk menggambarkan proses dalam suatu prosedur. Bagan alir proses memiliki simbol yang terdapat pada Tabel II.5.

Tabel II.5 Simbol Bagan Alir Proses (*Process Flowchart*)

Simbol	Nama	Keterangan
	<i>Operation</i>	Menunjukkan suatu operasi
	<i>Movement</i>	Menunjukkan suatu pemindahan
	<i>Storage</i>	Menunjukkan suatu simpanan
	<i>Inspection</i>	Menunjukkan suatu inspeksi
	<i>Delay</i>	Menunjukkan suatu penundaan

(Sumber: Jogiyanto, 2005)

2.11 *Unified Modelling Language (UML) 2.0*

Sampai tahun 1995, konsep objek menjadi populer tetapi dilaksanakan dengan berbagai cara oleh pengembang yang berbeda. Setiap pengembang memiliki metodologi dan notasi sendiri (misalnya, Boach, Coad, Moses, OMT, OOSE, dan SOMA). Kemudian pada tahun 1995, *Rational Software* mengajak tiga pemimpin industri bersama-sama untuk menciptakan sebuah pendekatan tunggal untuk pengembangan sistem berorientasi objek (Dennis et al, 2012). Grady Boach, Ivar Jacobson dan James Rumbaugh bekerja dengan orang lain untuk menciptakan satu set standar teknik diagram yang dikenal sebagai *Unified Modelling Language* (UML). Tujuan dari UML adalah untuk menyediakan *vocabulary* (kosakata) umum untuk istilah berorientasi objek dan teknik diagram yang cukup kaya untuk model setiap proyek pengembangan sistem dari analisis melalui implementasi. Pada November 1997, *Object Management Group* (OMG) secara resmi menerima UML sebagai standar untuk semua pengembang objek. Selama bertahun-tahun sejak itu, UML telah melalui beberapa revisi kecil (Dennis et al, 2012).

2.12 Diagram Pemodelan

Versi 2.0 dari UML mendefinisikan satu set dari empat belas teknik diagram yang digunakan untuk memodelkan sistem. Dengan yang digunakan dipecah menjadi dua kelompok utama, kelompok pertama untuk pemodelan struktur dari sistem dan yang kedua untuk pemodelan sifat. Diagram pemodelan struktur termasuk *class diagram*, *object diagram*, *package diagram*, *deployment diagram*, *component diagram*, dan *composite structure diagram*. Diagram pemodelan sifat, meliputi *activity diagram*, *sequence diagram*, *communication diagram*, *interaction overview diagram*, *timing diagram*, *behavior state machine diagram*, *protocol state machine diagram*, dan *use case diagram*. Diagram bergerak dari mendokumentasikan *requirements* untuk menentukan desain. Secara keseluruhan, notasi yang konsisten, integrasi antara teknik diagram, dan penerapan diagram seluruh proses pengembangan, membuat UML menjadi bahasa yang kuat dan fleksibel untuk analisis dan pengembangan (Dennis et al, 2012).

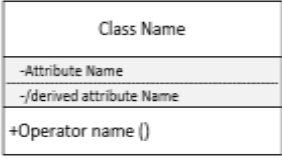
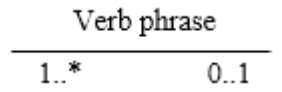
2.12.1 Structure Diagrams

Structure Diagram (diagram struktur) menyediakan cara untuk mewakili data dan hubungan statis yang berada dalam sistem informasi. Di bawah ini, akan digambarkan tujuan dasar masing-masing diagram struktur (Dennis et al, 2012).

1. *Class diagram*

Tujuan utama dari *class diagram* adalah untuk menciptakan sebuah *vocabulary* (kosakata) yang digunakan oleh analis dan pengguna. *Class diagram* biasanya merupakan hal-hal, ide-ide atau konsep yang terkandung dalam aplikasi. Misalnya, jika anda sedang membangun sebuah aplikasi penggajian, *class diagram* mungkin akan berisi kelas yang mewakili hal-hal seperti karyawan, cek, dan pendaftaran gaji. *Class diagram*, juga akan menggambarkan hubungan antara kelas (Dennis et al, 2012). Elemen-elemen dari *class diagram* terlihat pada Tabel II.6.

Tabel II.6 Elemen-Elemen *Class Diagram*

No.	Elemen	Nama Elemen	Fungsi Elemen
1.		<i>Class</i>	Untuk menggambarkan <i>class</i> atau kumpulan <i>object-object</i> yang mempunyai <i>attribute</i> , dan <i>operation</i> .
2.	Attribute name /derived attribute name	<i>Attribute</i>	Untuk menggambarkan atribut yang dimiliki suatu <i>class</i> .
3.	Operation name ()	<i>Operation</i>	Untuk menunjukkan suatu tindakan/fungsi yang dapat dilakukan oleh <i>class</i> .
4.		<i>Associattion</i>	Untuk menggambarkan hubungan antara dua <i>class</i> dan hubungan suatu <i>class</i> itu sendiri.

(Sumber: Dennis et al, 2012)

2. *Object diagram*

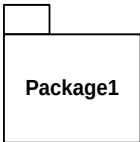
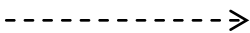
Object diagram sangat mirip dengan *class diagram*. Perbedaan utama adalah bahwa *object diagram* menggambarkan objek dan hubungan antar

objek. Tujuan utama dari *object diagram* adalah memungkinkan analisis untuk mengungkap rincian tambahan dari sebuah kelas. Dalam beberapa kasus, representatif variabel dari sebuah *class diagram* dapat membantu pengguna atau analis dalam menentukan atribut tambahan yang relevan, hubungan, dan atau operasi, atau mungkin menemukan beberapa atribut, hubungan, atau operasi yang salah tempat (Dennis et al, 2012).

3. *Package Diagram*

Package diagram digunakan untuk mengelompokkan elemen diagram UML yang berlainan secara bersama-sama ke dalam tingkat *construct* yang lebih tinggi yaitu berupa sebuah paket (Dennis et al, 2012). Elemen-elemen dari *package diagram* terlihat pada Tabel II.7.

Tabel II.7. Elemen-Elemen *Package Diagram*

No	Elemen	Nama Elemen	Fungsi Elemen
1.		<i>Package</i>	Untuk menyederhanakan diagram UML dengan mengelompokkan elemen-elemen terkait ke elemen satu tingkat lebih tinggi.
2.		<i>Dependency Relationship</i>	Untuk menggambarkan ketergantungan antara paket.

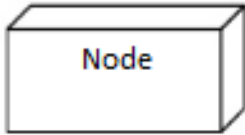
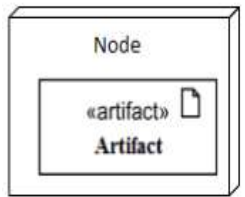
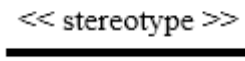
(Sumber: Dennis et al, 2012)

4. *Deployment Diagram*

Deployment diagram digunakan untuk mewakili hubungan antara komponen-komponen *hardware* yang digunakan dalam infrastruktur fisik sistem informasi. Misalnya, ketika merancang suatu sistem informasi terdistribusi yang akan menggunakan jaringan luas, *deployment diagram* dapat digunakan untuk menunjukkan hubungan komunikasi antara node yang berbeda dalam jaringan. *Deployment diagram* juga dapat digunakan untuk mewakili komponen-komponen *software* dan cara *software* ditempatkan pada arsitektur fisik atau infrastruktur sistem informasi. Dalam hal ini, *deployment diagram* mewakili lingkungan pembuatan

software (Dennis et al, 2012). Elemen-elemen dari *deployment diagram* terlihat pada Tabel II.8.

Tabel II.8 Elemen-Elemen *Deployment Diagram*

No	Elemen	Nama Elemen	Fungsi Elemen
1.		<i>Node</i>	Untuk menggambarkan sumber daya komputasi dalam sebuah sistem (misalnya, komputer klien, server, jaringan yang terpisah, atau individu perangkat jaringan).
2.		<i>Artifact</i>	Untuk menggambarkan spesifikasi dari <i>software</i> atau <i>database</i> , misalnya <i>file</i> sumber, tabel <i>database</i> , <i>file executable</i> .
3.		<i>Node with a Deployed Artifact</i>	Untuk menggambarkan <i>artifact</i> yang ditempatkan pada node fisik. Mendukung pemodelan distribusi perangkat lunak melalui jaringan.
4.		<i>Communication Path</i>	Untuk menggambarkan hubungan antara dua node. Memungkinkan node untuk bertukar pesan.

(Sumber: Dennis et al, 2012)

5. *Component Diagram*

Component diagram memungkinkan perancang untuk memodelkan hubungan fisik anantara modul-modul fiisik dari kode. Diagram ini bila dikombinasikan dengan *deployment diagram* dapat digunakan untuk menggambarkan distribusi fisik dari modul *software* melalui jaringan. Misalnya, ketika merancang sistem *client-server*, hal ini berguna untuk menunjukan kelas atau paket kelas mana yang akan berada pada node *client* dan mana yang akan berada di *server*. *Component diagram* juga

dapat berguna dalam merancang dan mengembangkan sistem berbasis komponen (Dennis et al, 2012).

6. *Composite Structure Diagram*

UML 2.0 menyediakan diagram baru, ketika struktur internal *class* bersifat kompleks maka disediakan *composite structure diagram*. *Composite structure diagram* ini digunakan untuk pemodelan hubungan antara bagian-bagian dari sebuah *class*. Sebagai contoh, ketika pemodelan pendaftaran penggajian, analis mungkin ingin *class* yang mewakili seluruh laporan serta *class* yang mewakili *header*, *footer*, dan garis detail laporan. Dalam sebuah *class diagram* standar, akan membutuhkan analis untuk memodelkan pendaftaran penggajian menjadi empat *class* terpisah yang memiliki hubungan, kemudian menghubungkan mereka bersama-sama. Sedangkan, jika menggunakan *composite structure diagram* maka akan ada tiga *subclass*: *header*, *footer*, dan garis detail. *Composite structure diagram* juga berguna ketika dilakukan pemodelan struktur internal komponen untuk sistem berbasis komponen. Seringkali, *composite structure diagram* merupakan mekanisme pemodelan yang kurang efektif karena pemodelan juga bisa dilakukan dengan mengkomunikasikan penggunaan paket dan diagram paket (Dennis et al, 2012).

2.12.2 *Behavior Diagram*

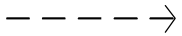
Behavior Diagram menyediakan analis dengan cara menggambarkan hubungan dinamis antara instansi atau benda yang mewakili sistem informasi bisnis. Diagram ini juga memungkinkan pemodelan perilaku dinamik dari objek individu sepanjang masa aktifnya. Diagram ini mendukung analis dalam memodelkan *requirements* fungsional agar sistem informasi berkembang (Dennis

et al, 2012). Berikut ini akan digambarkan tujuan dasar dari masing-masing *behavior diagram*

1. *Activity Diagram*

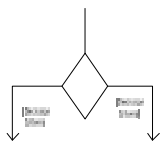
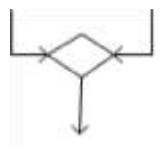
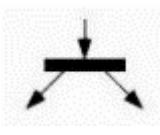
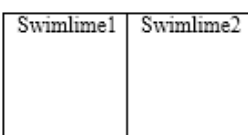
Activity diagram digunakan untuk memodelkan proses dalam suatu sistem informasi. Diagram ini dapat digunakan untuk memodelkan *lur kerja*, *use case* individual, atau logika keputusan yang terkandung dalam metode individual. Diagram ini juga menyediakan pendekatan untuk memodelkan proses parallel (Dennis et al, 2012). Elemen-elemen dari *activity diagram* terlihat pada Tabel II.9.

Tabel II.9 Elemen-Elemen *Activity Diagram*

No	Elemen	Nama Elemen	Fungsi Elemen
1.		<i>Action</i>	Untuk menggambarkan perilaku yang sederhana dan bersifat <i>non decomposable</i> .
2.		<i>Activity</i>	Untuk mewakili serangkaian aksi (<i>action</i>).
3.		<i>Object Node</i>	Untuk mewakili objek yang terhubung dengan serangkaian <i>object flow</i> .
4		<i>Control Flow</i>	Untuk mewakili serangkaian pelaksanaan.
5		<i>Object Flow</i>	Untuk menunjukkan aliran sebuah objek dari sebuah aktivitas (aksi), atau ke sebuah aktivitas (aksi).

Tabel II.9 Elemen-Elemen *Activity Diagram* (Lanjutan)

No	Elemen	Nama Elemen	Fungsi Elemen
6		<i>Initial Node</i>	Untuk menandakan awal dari serangkaian aksi atau aktivitas.

7		<i>Final-Activity Node</i>	Untuk menghentikan seluruh <i>control flow</i> atau <i>object flow</i> pada sebuah aktivitas (aksi).
8		<i>Final-Flow Node</i>	Untuk menghentikan <i>control flow</i> atau <i>object flow</i> tertentu
9		<i>Decision Node</i>	Untuk mewakili suatu kondisi pengujian, yang bertujuan untuk memastikan bahwa <i>control flow</i> atau <i>object flow</i> hanya menuju ke satu arah.
10		<i>Merge Node</i>	Untuk menyatukan kembali <i>decision path</i> yang dibuat dengan menggunakan <i>decision node</i> .
11		<i>Fork node</i>	Untuk memisahkan perilaku menjadi serangkaian aktivitas yang berjalan secara <i>parallel</i> atau bersamaan.
12		<i>Join Node</i>	Untuk menyatukan kembali serangkaian aktivitas yang berjalan secara <i>parallel</i> atau bersamaan.
13		<i>Swimlane</i>	Untuk membagi sebuah <i>activity diagram</i> menjadi beberapa kolom, dengan tujuan menempatkan aktivitas (aksi) tertentu pada individu (objek) yang bertanggung jawab atas terlaksananya aktivitas (aksi) tersebut.

(Sumber: Dennis et al, 2012)

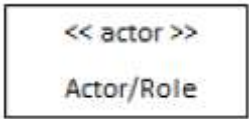
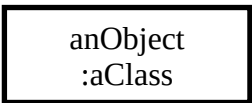
2. *Interaction diagram*

Interaction diagram menggambarkan interaksi antara objek-objek dari suatu sistem informasi berorientasi objek. UML 2.0 menyediakan empat *interaction diagram* yang berbeda-beda, yaitu *sequence diagram*,

communication diagram, *interaction overview diagram*, dan diagram waktu (Dennis et al, 2012).

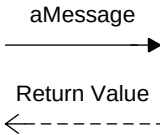
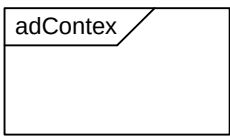
- a. *Sequence diagram* memungkinkan analisis untuk menggambarkan interaksi dinamis antara objek-objek dalam suatu sistem informasi. *Sequence diagram* sejauh ini merupakan jenis interaksi yang paling umum digunakan dalam pemodelan berorientasi objek. Diagram ini menekankan penyusunan berbasis waktu untuk kegiatan yang dilakukandengan sekumpulan objek yang berkolaborasi. Diagram ini sangat berguna dalam membantu analisis, memahami spesifikasi *real time* dan *use case* yang rumit. Diagram ini dapat digunakan untuk mendeskripsikan interaksi antara objek-objek, baik secara fisik maupun logis. Hal tersebut menyebabkan diagram ini berguna dalam kegiatan analisis dan juga desain (Dennis et al, 2012). Elemen-elemen dari *sequence diagram* terlihat pada Tabel II.10. di bawah ini.

Tabel II.10 Elemen-Elemen *Sequence Diagram*

No	Elemen	Nama Elemen	Fungsi Elemen
1.	 Actor 	<i>Actor</i>	Untuk menggambarkan <i>actor</i> (sistem), yang memperoleh keuntungan dan berada di luar sistem. Diletakkan di bagian atas diagram. Jika <i>actor</i> bukan manusia, maka menggunakan persegi panjang.
2.		<i>Object</i>	Untuk menggambarkan secara berurutan pengiriman data/atau penerimaan <i>message</i> . Diletakkan di bagian atas diagram.

Tabel II.10 Elemen-Elemen *Sequence Diagram* (Lanjutan)

No	Elemen	Nama Elemen	Fungsi Elemen
----	--------	-------------	---------------

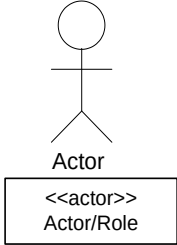
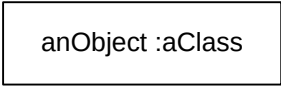
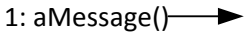
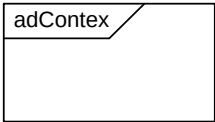
3		<i>Lifeline</i>	Untuk menggambarkan keberadaan <i>object</i> pada waktu tertentu.
4		<i>Execution Occurrence</i>	Untuk menggambarkan periode waktu selama <i>object</i> atau <i>actor</i> melakukan aktivitas dalam pengiriman atau penerimaan pesan.
5		<i>Message</i>	Untuk menyampaikan informasi dari satu objek ke objek lain. Panggilan operasi diberi label satu pesan dikirim dan panah padat sedangkan yang kembali diberi label suatu nilai yang dikembalikan dan ditampilkan sebagai sebuah panah putus-putus.
6		<i>Object Destruction</i>	Untuk menggambarkan penghancuran sebuah elemen header setelah selesainya sebuah operasi.
7		<i>Frame</i>	Untuk menunjukkan konteks dari <i>sequence diagram</i> .

(Sumber: Dennis et al, 2012)

- b. *Communication diagram* memberi pandangan alternatif dari interaksi diterima yang terjadi di antara objek-objek dalam suatu sistem informasi berorientasi objek. Jika *sequence diagram* menekankan penyusunan berbasis waktu dari suatu kegiatan, *communication diagram* lebih berfokus pada sekumpulan pesan yang disampaikan dalam sekumpulan objek yang berkolaborasi. Dengan kata lain, *communication diagram* menggambarkan bagaimana objek-objek dapat berkolaborasi untuk mendukung beberapa aspek dari fungsi yang diperlukan oleh sistem. Urutan atau penyusunan berdasarkan

waktu dari pesan ditunjukkan melalui skema urutan penomoran (Dennis et al, 2012). Elemen-elemen dari *communication diagram* terlihat pada Tabel II.11.

Tabel II.11 Elemen-Elemen *Communication Diagram*

No	Elemen	Nama Elemen	Fungsi Elemen
1.		<i>Actor</i>	Untuk menggambarkan <i>actor</i> (sistem), yang memperoleh keuntungan dan berada di luar sistem.
2.		<i>Object</i>	Untuk menggambarkan pengiriman dan/atau penerimaan <i>message</i> . Diletakkan dibagian atas diagram
3.		<i>Association</i>	Untuk menggambarkan hubungan antara aktor-aktor dan/atau objek-objek. Pesan dikirim melalui <i>association</i> .
4.		<i>Message</i>	Untuk menyampaikan informasi dari satu objek ke objek lainnya, arah ditunjukkan menggunakan sebuah mata panah. Urutan ditunjukkan oleh nomor urut
5.		<i>Frame</i>	Untuk menunjukkan konteks dari <i>communication diagram</i> .

(Sumber: Dennis et al, 2012)

- c. *Interaction overview diagram* membantu analisis memahami *use case* yang kompleks. Diagram ini memberikan gambaran kontrol aliran proses. Diagram ini memperluas *activity diagram* melalui penambahan fragmen-fragmen *sequence* dari *sequence diagram*. Akibatnya, fragmen-fragmen *sequence* diperlakukan seolah-olah fragmen merupakan *activity* dalam *activity diagram*. Keuntungan utama menggunakan diagram ini adalah bahwa anda dapat dengan mudah memodelkan aliran *sequence* alternatif walaupun dalam

prakteknya, hal ini bisa dicapai dengan menggunakan *activity diagram* disamping menggunakan *use case diagram* (Dennis et al, 2012).

- d. *Timing diagram* menggambarkan interaksi antara objek sepanjang suatu poros waktu. Tujuan utama dari *diagram* ini adalah untuk menunjukkan perubahan keadaan objek dalam menanggapi peristiwa-peristiwa yang terjadi dari waktu ke waktu. Diagram ini cenderung sangat berguna ketika mengembangkan sistem yang *real time* atau *embedded* (Dennis et al, 2012).

3. *State Machines*

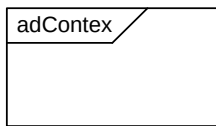
Dalam UML 2.0, ada dua state machines yang berad yaitu *behavior state machines* dan *protocol state machines*. *Behavior state machines* digunakan untuk menjelaskan perubahan yang dialami objek selama masa aktifnya. *Protocol state machines* menggambarkan spesifikasi sequence dari suatu peristiwa, dapat direspon objek (Dennis et al, 2012).

- a. *Behavior state machines* adalah metode untuk memodelkan keadaan atau sekumpulan nilai yang berbeda. Sebagai contoh, seorang pasien bisa berubah statusnya dari waktu ke waktu, misalnya seorang yang sebelumnya pasien baru bisa berubah menjadi pasien lama. Masing-masing kategori pasien ini benar-benar menunjukkan keadaan yang berbeda dari pasien yang lama. Keadaan yang berbeda tersebut dihubungkan dengan peristiwa yang menyebabkan pasien berubah dari suatu keadaan ke keadaan lainnya (Dennis et al, 2012). Elemen-elemen *behavior state machines* terlihat pada Tabel II.12.

Tabel II.12 Elemen-Elemen *Behavior State Machines*

No	Elemen	Nama Elemen	Fungsi Elemen
1.		<i>State</i>	Untuk mewakili keadaan dari suatu objek.
2.		<i>Initial State</i>	Untuk menggambarkan keadaan awal dari sebuah objek.

Tabel II.12 Elemen-Elemen *Behavior State Machines* (Lanjutan)

No	Elemen	Nama Elemen	Fungsi Elemen
3.		<i>Find State</i>	Untuk menggambarkan penyelesaian suatu keadaan atau aktivitas.
4.	anEvent :	<i>Event</i>	Untuk menunjukkan nama dari tansition, dan merupakan kejadian penting yang memicu perubahan suatu keadaan.
5.		<i>Transition</i>	Untuk menggambarkan perpindahan <i>flow event</i> dari <i>state</i> suatu ke <i>state</i> yang lain.
6.		<i>Frame</i>	Untuk menunjukkan konteks dari <i>behavior state machines</i> .

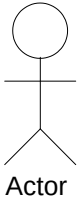
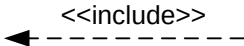
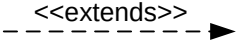
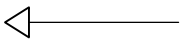
(Sumber: Dennis et al, 2012)

- b. *Protocol state machines* digunakan merancang hubungan antara unsur-unsur *interface* pada suatu kelas. Sebagai contoh, biasanya membuka *file* atau *database* sebelum membuat *query* atau memperbarui *query*. Tidak seperti *behavior state machines*, *protocol state machines* bisa dikaitkan dengan port pada komponen atau *interface* pada kelas (Dennis et al, 2012).

4. *Use Case*

Use case diagram digunakan untuk memodelkan interaksi antara sistem informasi dan lingkungannya. Lingkungan sitem informasi dapat mencakup *user* dan setiap sistem eksternal yang berinteraksi dengan sistem informasi. Penggunaan utama dari *use case diagram* adalah untuk menyediakan sarana dalam mendokumentasikan dan memahami *requirements* sistem informasi yang sedang berkembang. *Use cases* dan *use case diagram* adalah alat-alat yang paling penting untuk digunakan dalam analisis dan desain sistem berorientasi objek (Dennis et al, 2012). Elemen-elemen dari *use case diagram* terlihat pada Tabel II.13.

Tabel II.13 Elemen-Elemen *Use Case Diagram*

No	Elemen	Nama Elemen	Fungsi Elemen
1.	 Actor <<actor>> Actor/Role	<i>Actor</i>	Untuk menggambarkan <i>actor</i> (sistem), yang memperoleh keuntungan dan berada di luar sistem. <i>Actor</i> dapat berasosiasi dengan <i>actor</i> lainnya, dengan menggunakan <i>specialization</i> (<i>superclass association</i>). <i>Actor</i> ditempatkan di luar <i>subject boundary</i> .
2.		<i>Use case</i>	Untuk mewakili sebuah bagian dari fungsionalitas sistem dan ditempatkan dalam <i>system boundary</i> . Dapat <i>extend</i> dan <i>include use case</i> lain.
3.	Subject Boundary	<i>Subject Boundary</i>	Untuk menyatakan lingkup dari subjek.
4.		<i>Association Relationship</i>	Untuk menghubungkan <i>actor</i> dan <i>use case</i> agar dapat saling berinteraksi.
5.		<i>Include Relationship</i>	Untuk menunjukkan <i>inclusion</i> dari sebuah <i>use case</i> dengan <i>use case</i> lainnya. Arah panah dari <i>base use case</i> ke <i>include use case</i> .
6.		<i>Extand Relationship</i>	Untuk menunjukkan <i>extension</i> dari sebuah <i>use case</i> , untuk menambahkan <i>optional behavior</i> . Arah panah dari <i>extension use case</i> ke <i>base use case</i> .
7.		<i>Generalization Relationship</i>	Untuk menunjukkan generalisasi dari <i>use case</i> khusus ke yang lebih umum.

(Sumber: Dennis et al, 2012)

2.13 *Hierarchy plus Input-Proses-Output (HIPO)*

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

Hierarchy plus Input-Process-Output (HIPO) mempunyai sasaran utama sebagai berikut (Jogiyanto, 2005):

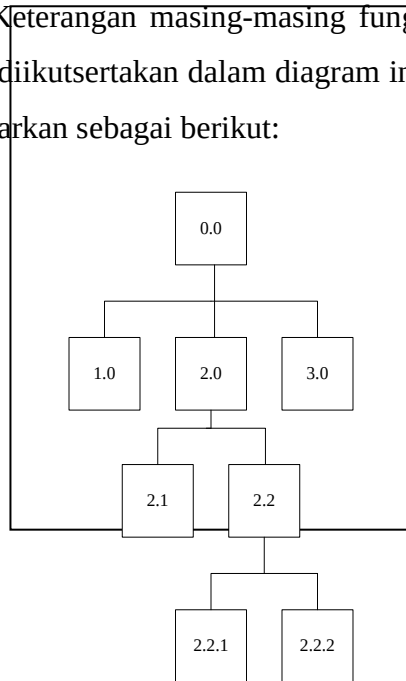
1. Untuk menyediakan suatu struktur guna memahami fungsi-fungsi dari sistem.
2. Untuk lebih menekankan fungsi-fungsi yang harus diselesaikan oleh program, bukannya menunjukkan statemen-statemen program yang digunakan untuk melaksanakan fungsi tersebut.
3. Untuk menyediakan penjelasan yang jelas dari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

2.13.1 Diagram dalam Paket HIPO

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

1. *Visual Table Of Contents (VTOC)*

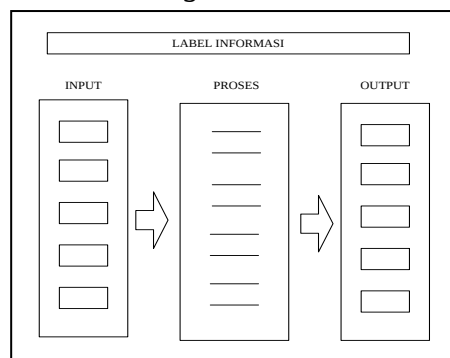
Visual table of contents menggambarkan hubungan fungsi-fungsi di sistem secara berjenjang. *Visual table of contents* menggambarkan seluruh program HIPO baik rinci maupun ringkasan yang terstruktur. Pada diagram ini namadan nomor dari program HIPO diidentifikasi. Struktur paket diagram dan hubungan fungsi 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: Jogiyanto, 2005)

2. *Overview diagram*

Overview diagram menunjukkan secara garis besar hubungan dari *Input*, proses dan *output*. Bagian *input* menunjukkan item-item data yang akan digunakan oleh bagian proses. Bagian proses berisi sejumlah langkah-langkah yang menggambarkan kerja dari fungsi. Bagian *output* berisi dengan item-item data yang dihasilkan atau dimodifikasi oleh langkah-langkah proses. *Overview diagram* ini :



Gambar II.7 Overview Diagram
(Sumber: Jogiyanto, 2005)

3. *Detail Diagram*

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

2.14 Kamus Data

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

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

BAB III

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

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

Metode penelitian adalah cara yang digunakan oleh peneliti dalam pengumpulan data penelitiannya. Berdasarkan pengertian tersebut dapat dikatakan bahwa metode penelitian adalah cara yang dipergunakan untuk mengumpulkan data yang diperlukan dalam penelitian (Arikunto, 2002). Metode penelitian adalah cara-cara ilmiah untuk mendapatkan data yang valid, dengan tujuan dapat ditemukan, dikembangkan dan dibuktikan, suatu pengetahuan tertentu sehingga pada gilirannya dapat digunakan untuk memahami, memecahkan, dan mengantisipasi masalah (Sugiyono, 2003). Berikut ini akan dijelaskan tentang metodologi penelitian yang akan digunakan pada penelitian ini.

3.2 Jenis dan Sumber Data

Data yang dibutuhkan dalam penelitian ini dikelompokkan menjadi dua jenis, yaitu sumber data primer dan data sekunder.

1. Data Primer

Data yang diperoleh melalui pengamatan langsung terhadap sistem yang sedang berjalan dan wawancara dengan pegawai *Warehouse Raw Material* sebagai sumber informasinya. Dalam penelitian ini data tersebut berupa proses bisnis, aliran data sistem informasi persediaan bahan baku, prosedur pembuatan laporan permintaan produksi, prosedur penerimaan bahan baku, prosedur penyimpanan bahan baku, prosedur pengeluaran bahan baku untuk produksi pada PT Rekadaya Multi Adiprima.

2. Data Sekunder

Data sekunder merupakan sumber data penelitian yang diperoleh secara tidak langsung melalui media perantara atau data yang diperoleh dan dicatat oleh pihak lain yang telah disusun dan dipublikasikan. Data sekunder yang digunakan dalam penelitian ini laporan-laporan, buku-buku dan kajian ilmiah dari berbagai sumber yang berkaitan dengan penelitian.

3.3 Metode Pengumpulan Data

Langkah pertama yang dilakukan sebagai dasar dari kerangka pemikiran dalam penelitian ini adalah dengan melakukan studi pendahuluan terhadap objek penelitian. Secara sederhana, pengumpulan data diartikan sebagai proses atau kegiatan yang dilakukan peneliti untuk mengungkap atau menjangkau berbagai fenomena, informasi atau kondisi lokasi penelitian sesuai dengan lingkup penelitian. Pengumpulan data sangat penting dalam proses penelitian, karena tanpa data lapangan, proses analisis data dan kesimpulan hasil penelitian, tidak dapat dilaksanakan.

3.4 Instrumen Pengumpulan Data

Metode yang digunakan dalam pengumpulan data, diantaranya:

1. Studi lapangan

Studi lapangan adalah usaha melakukan pengumpulan data secara langsung pada objek yang dilakukan dengan cara sebagai berikut:

- a. Pengamatan, tahap ini dilakukan secara langsung di bagian *Warehouse Raw Material* PT Rekadaya Multi Adiprima dengan mengamati sistem informasi persediaan bahan baku pada gudang *raw material*, mulai dari bahan baku datang, disimpan sampai proses bisnis persediaan bahan baku. Melalui teknik ini, data yang dibutuhkan diamati, dikumpulkan dan diolah sebagai bahan dalam penelitian.
- b. Wawancara, yaitu mencari data yang dibutuhkan secara langsung dengan memberikan pertanyaan spontan tentang proses bisnis

persediaan bahan baku yang diperlukan pada penyusunan tugas akhir ini. Wawancara ini dilakukan kepada pegawai dibagian *Warehouse Raw Material* yaitu Bapak Joko dan Bapak Tria.

2. Studi Literatur

Tahap ini merupakan tahap awal pengumpulan data dari sumber-sumber seperti buku yang berhubungan analisis dan desain sistem informasi, mengelola kebutuhan dan persediaan bahan baku, pemrograman web dengan PHP dan MySQL, sistem informasi manajemen, pengembangan perangkat lunak, dan juga mencari data tambahan yang diperlukan seperti konsep pemrograman PHP, beberapa contoh *coding* program melalui media *internet* dan video tutorial.

3.5 Metode Pengembangan Sistem

Metode pengembangan sistem adalah menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah berjalan (Arikunto, 2006). Pada penelitian ini untuk mengatasi masalah yang ada pada sistem, diputuskan untuk membuat pengembangan sistem. Dalam pengembangan sistem ini digunakan metode *Evolutionary Prototype*. Alasan menggunakan metode *Prototype* ini pada PT Rekadaya Multi Adiprima karena metode *Prototype* lebih menghemat waktu dalam pengembangan sistem, penerapan menjadi lebih mudah karena karyawan mengetahui apa yang diharapkannya, dan lebih efisien dibanding dengan metode *Waterfall* dan metode RAD (*Rapid Application Development*) yang memakan waktu lama dalam pengembangan sistem dan kurang efisien. Tahapan-tahapan pada model *Prototype* adalah sebagai berikut:

1. Identifikasi kebutuhan *user*

Melakukan diskusi dengan karyawan pengolahan data persediaan bahan baku di Divisi *Warehouse Raw Material* tentang kondisi sistem saat ini dan kebutuhan sistem yang diinginkan.

2. Membuat *Prototype*

Membuat *Prototype* dari sistem yang telah dijelaskan oleh karyawan pengolahan data persediaan bahan baku di Divisi *Warehouse Raw Material*.

3. Menyesuaikan dan evaluasi *Prototype* dengan keinginan *user*

Menanyakan kepada karyawan pengolahan data persediaan bahan baku di Divisi *Warehouse Raw Material* tentang *Prototype* yang sudah dibuat, apakah sesuai atau tidak dengan kebutuhan sistem.

4. Menggunakan *Prototype*

Sistem mulai dikembangkan dengan *Prototype* yang sudah dibuat oleh karyawan pengolahan data persediaan bahan baku di Divisi *Warehouse Raw Material*.

3.6 Kerangka Penelitian

Beberapa tahapan yang penulis lakukan dalam melakukan penelitian kualitatif ini adalah sebagai berikut:

1. Studi Pendahuluan

Studi pendahuluan yang penulis lakukan yaitu dengan membaca buku literatur, *browsing internet* serta sumber-sumber lain dalam lingkup perkuliahan maupun di luar lingkup perkuliahan yang berhubungan dengan judul dan permasalahan tugas akhir. Studi pendahuluan ini dimaksudkan untuk dapat mengetahui suatu gambaran yang jelas mengenai kondisi dan situasi pada Divisi *Warehouse Raw Material* yang masih manual dalam pengecekan persediaan bahan baku, serta untuk mengetahui masalah yang sedang dihadapi. Langkah-langkah yang dilakukan dalam studi pendahuluan bertujuan untuk mendapatkan informasi, yaitu dengan cara observasi dan melakukan wawancara terhadap pihak-pihak terkait yang berada di Divisi *Warehouse Raw Material*.

2. Identifikasi Masalah

Pokok dari permasalahan yang menjadi objek penelitian ini adalah mengenai persoalan sistem persediaan bahan baku di Divisi *Warehouse Raw Material* pada PT Rekadaya Multi Adiprima. Identifikasi dari permasalahan tersebut yaitu pencatatan bahan baku masuk dan keluar masih dilakukan dengan cara meng-*update* kartu stok sehingga data yang ada belum terorganisir dengan baik. Hal tersebut menyebabkan terjadinya kerusakan, kehilangan dan ketidaksesuaian data. Belum tersedianya laporan persediaan bahan baku pada Divisi *Warehouse Raw Material* sehingga sangat sulit dalam menentukan titik pesan kembali dan melihat ketersediaan bahan baku. Hal tersebut menyebabkan terjadinya kekurangan stok akibat melonjaknya permintaan.

3. Tujuan Penelitian

Tujuan penelitian dari tugas akhir ini adalah merancang bangun sistem informasi persediaan bahan baku berbasis web agar pencatatan bahan baku yang masuk dan keluar pada gudang dilakukan secara komputerisasi dari Divisi *Warehouse Raw Material* pada PT Rekadaya Multi Adiprima sehingga tidak akan terjadi lagi kerusakan, kehilangan dan ketidaksesuaian pada data bahan baku. Merancang bangun sistem informasi pada Divisi *Warehouse Raw Material* agar penyediaan laporan persediaan bahan baku dapat dilakukan secara komputerisasi sehingga dapat mengetahui titik pesan kembali stok agar tidak terjadi lagi kekurangan persediaan pada bahan baku.

4. Batasan Masalah

Pembatasan masalah dilakukan agar penelitian yang dilakukan di Divisi *Warehouse Raw Material* lebih fokus, terarah dan hanya sebatas menangani masalah pengolahan data bahan baku, pencatatan data penerimaan bahan baku, pencatatan data pengeluaran bahan baku dan laporan ketersediaan bahan baku.

5. Identifikasi Kebutuhan Pengguna

Identifikasi kebutuhan pengguna yaitu karyawan pada Divisi *Warehouse Raw Material* merupakan langkah lanjutan dari pengolahan data, dimana semua spesifikasi sistem dan kebutuhannya dituangkan ke dalam sebuah dokumen persyaratan perangkat lunak.

6. Membuat *Prototype*

Proses pembuatan *Prototype* melalui beberapa tahap antara lain, *Modelling system* dengan menggunakan *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Flowchart* dan HIPO lalu *Modelling Data* dengan menggunakan *Class Diagram* dan Kamus Data. *Tools* yang digunakan antara lain PHP 5.4.7, MySQL 5.5.27, Mozilla Firefox Browser, phpMyAdmin, StarUML dan Ms. Visio 2010. Alasan menggunakan PHP 5.4.7 dan MySQL 5.5.27 karena dapat digunakan untuk membuat aplikasi berbasis web dinamis, pemrograman *database*, *server webhosting* banyak yang *support* dengan PHP, dan penggunaan memori yang lebih sedikit dibanding dengan Java dan ASP.NET yang penggunaan memori yang cukup banyak dan mengharuskan pengguna meng-*compile* programnya sebelum dijalankan. Salah satu *Tools Database Management System* (DBMS) yang mendukung PHP adalah MySQL, karena performa tinggi, dapat menampung jumlah *database* yang sangat besar, dan tidak mengurangi kecepatan dalam hal akses ke *database*-nya.

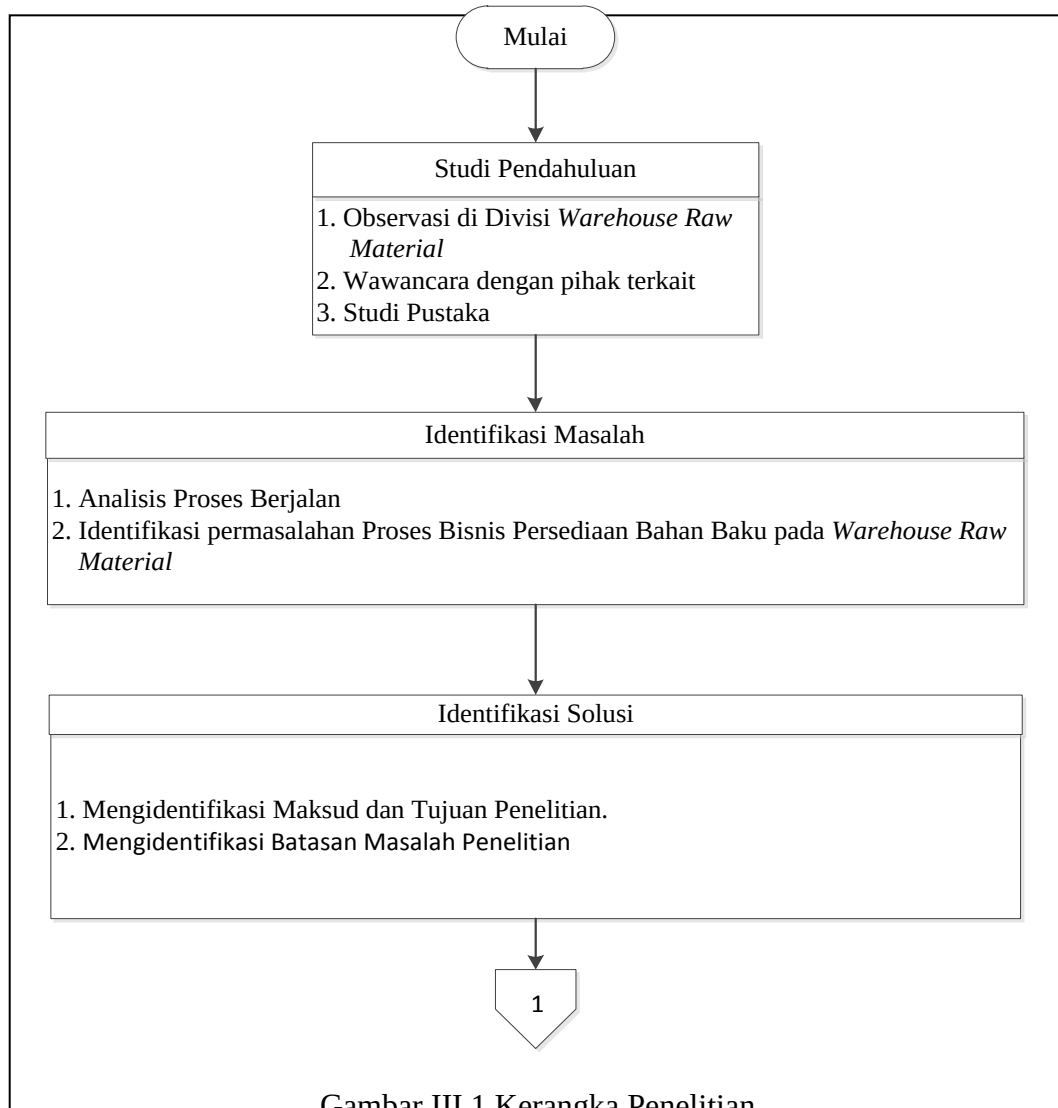
7. Menggunakan *Prototype*

Pada tahap ini, *Prototype* dari sistem yang penulis usulkan akan dilihat apakah sistem usulan tersebut sudah sesuai dengan keinginan karyawan pada Divisi *Warehouse Raw Material* atau belum. Jika belum, maka penulis akan melakukan pengumpulan data lagi dan menganalisis data kembali dan mencari tahu kebutuhan apa lagi yang dibutuhkan oleh pengguna sistem, serta melakukan perancangan sistem ulang. Sedangkan jika sudah sesuai keinginan karyawan pada Divisi *Warehouse Raw Material*, maka *Prototype* dapat digunakan.

8. Kesimpulan dan Saran

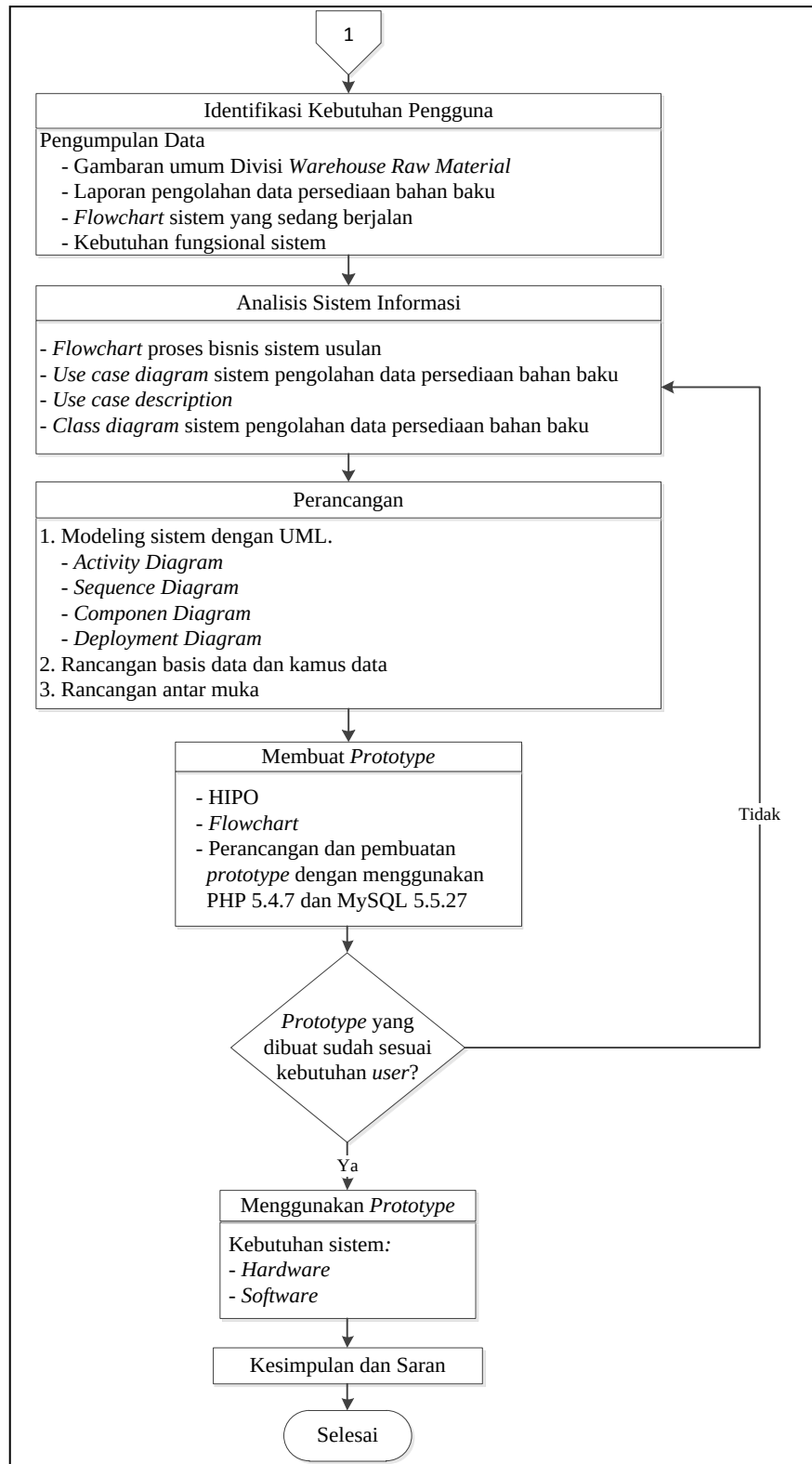
Mengambil kesimpulan dari hasil analisis sistem berjalan, sistem yang diusulkan dan aplikasi yang dirancang serta memberikan saran kepada Divisi *Warehouse Raw Material* dan kepada peneliti selanjutnya untuk pengembangan lanjutan.

Berikut adalah *flowchart* kerangka penelitian untuk menyelesaikan masalah dalam Tugas Akhir ini:



Gambar III.1 Kerangka Penelitian

(Sumber: Pengolahan Data, 2017)



Gambar III.1 Kerangka Penelitian (Lanjutan)
(Sumber: Pengolahan Data, 2017)

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

1.1 Sejarah Perusahaan

PT Rekadaya Multi Adiprima merupakan salah satu perusahaan yang bergerak dalam industri manufaktur. Didirikan sejak tahun 1994 telah memproduksi komponen otomotif baik untuk kendaraan roda empat maupun roda dua yang merupakan produk orisinil atau *Original Equipment Manufactured* (OEM). Kini PT Rekadaya Multi Adiprima telah memiliki lokasi usaha yang tersebar di Cikeas Nagrak-Kabupaten Bogor sebanyak 9 *plant*. 1 *plant* di Sumurkondang – Klari, Kabupaten Karawang, dan 1 *plant* di Cikarang – Kabupaten Bekasi.

Seiring dengan pertumbuhan bisnis otomotif di Indonesia juga terjadinya peningkatan permintaan di luar negeri (ekspor), berimbas kepada industri komponen otomotif tersebut, PT Rekadaya Multi Adiprima berfokus pada *stamping parts*, *insulator dush panel*, dan *furniture parts* baik untuk kendaraan roda empat maupun roda dua.

Di Indonesia khususnya daerah Jawa Barat merupakan tempat dimana tersebarnya divisi dari Rekadaya Group. Adapun anggota Rekadaya Group adalah sebagai berikut:

1. PT Rekadaya Multi Adiprima (PT RMA)
2. PT Rekadaya Kreasi Indonesia (PT RKI)
3. PT Rekadaya Prima Indonesia (PT RPI)
4. PT Rekadaya Global Logistik (PT RGL)
5. PT Rekadaya Nonwoven Indonesia (PT RNI)

1.2 Profil Perusahaan

Nama perusahaan : PT Rekadaya Multi Adiprima

Alamat kantor pusat : Alt. Cibubur-Cileungsi, Ciangsana Raya, No.55
Nagrak-Cikeas, Gunung Putri, Bogor-Jawa Barat

Telepon : 021 – 8232888

Fax : 021 – 8231774

Email : rma@rekadaya.com

Website : www.rekadaya.com

Produksi : *Part Plastik Service Hole, Felt, Insulator Dush Panel*

Status : Perseroan Terbatas

Luas Tanah : *Plant 1 1000m²*
Plant 2 1200m²
Plant 3 2000m²
Plant 4 4000m²
Plant 5 900m²
Plant 6 4000m²
Plant 7 2700m²
Plant 8 1300 m²
Plant 9 3000m²
Cikarang 30000m²
Karawang 15800m²

1.3 Visi, Misi, dan Pilar Lingkungan Perusahaan

PT Rekadaya Multi Adiprima perusahaan yang bergerak di bidang industri manufaktur komponen otomotif seperti *Part Plastik Service Hole, Felt, Insulator Dush Panel* serta komponen otomotif lainnya mempunyai visi dan misi sebagai berikut:

Visi:

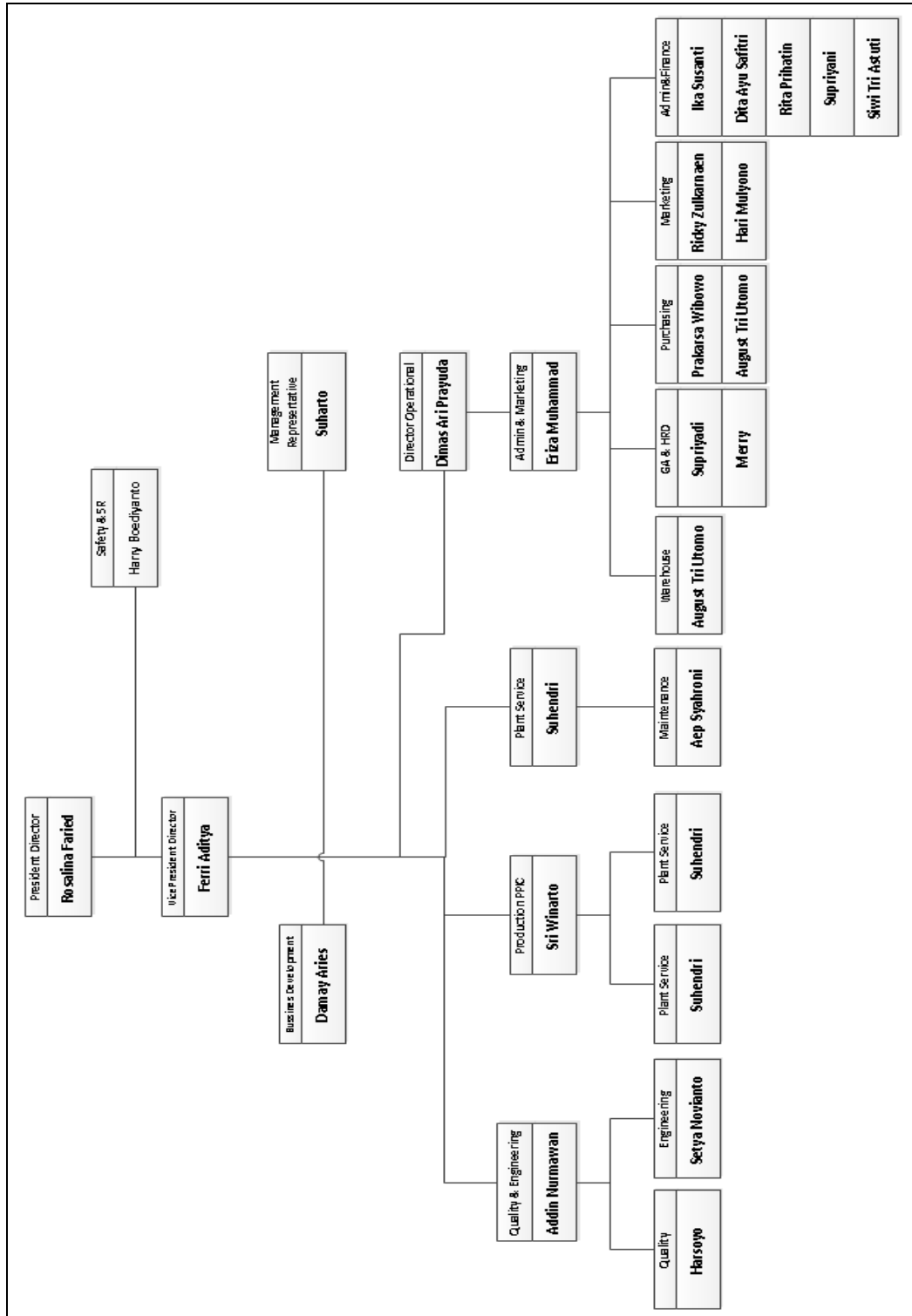
Menjadi perusahaan manufaktur komponen otomotif dan industri pendukung terbesar di Indonesia.

Misi:

1. Mewujudkan perusahaan yang maju berdasarkan Sistem Manajemen berstandar Internasional.
2. Mewujudkan produk perusahaan yang bermutu dan berwawasan lingkungan, keselamatan dan kesehatan kerja untuk memenuhi kebutuhan pelanggan dalam hal *Quality, Cost, Delivery, and Moral*.

1.4 Struktur Organisasi Perusahaan

Struktur organisasi yang tersusun dengan baik akan memudahkan koordinasi, integrasi, serta meningkatkan efektivitas dan efisiensi suatu perusahaan didalam mencapai tujuannya. Berikut merupakan struktur organisasi PT Rekadaya Multi Adiprima secara keseluruhan yang dapat dilihat pada Gambar IV.1.



Gambar IV.1 Struktur Organisasi PT Rekadaya Multi Adiprima
(Sumber: PT Rekadaya Multi Adiprima, 2016)

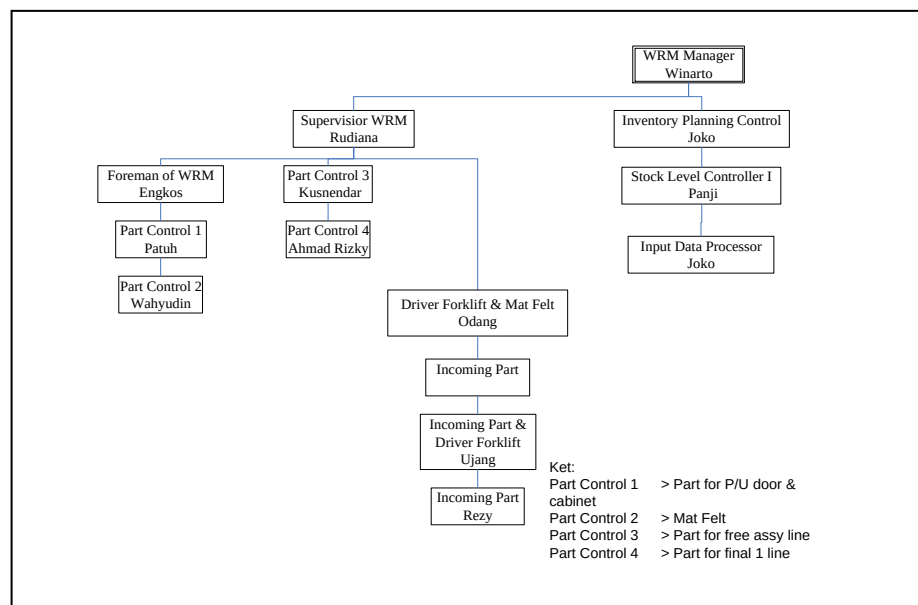
1.5 Warehouse Raw Material

Warehouse atau pergudangan berfungsi menyimpan barang untuk produksi atau hasil produksi dalam jumlah atau rentan waktu tertentu yang kemudian didistribusikan ke lokasi yang dituju.

Pada PT Rekadaya Multi Adiprima terdapat gudang di setiap *Plant* nya yang mempunyai fungsi sama yaitu untuk menyimpan barang jadi atau bahan baku. Untuk gudang di *Plant* 11 mempunyai fungsi menyimpan bahan baku untuk produksi setiap harinya.

Pada gudang *Plant* 11 ini terdapat pegawai gudang yang mempunyai tugas untuk mencatat setiap barang yang masuk dan barang yang keluar. Kemudian pegawai gudang juga harus memperhatikan kondisi gudang seperti merapihkan posisi barang yang berantakan dan membantu mengeluarkan barang yang ingin didistribusikan ke produksi. Serta memperhatikan kondisi barang apabila terjadi kerusakan pada kemasan maka harus diberikan kepada Divisi *packing* untuk dibenarkan plastik kemasannya.

Struktur Organisasi Warehouse Raw Material



Gambar IV.2 Struktur Organisasi Divisi *Warehouse Raw Material*
(Sumber: PT Rekadaya Multi Adiprima, 2016)

Untuk mempermudah dalam pemahaman struktur organisasi tentunya ada beberapa uraian tentang tugas dan wewenang yang tercantum pada struktur organisasi di atas:

1. *Manager Warehouse Raw Material (WRM)*

Tugas dan tanggung jawab:

- a. Mengkoordinasi kegiatan gudang mulai dari mengendalikan kedatangan *material* dan penyimpanannya pada tempat yang tepat.
- b. Meyakinkan semua *material* telah diterima sesuai dengan *purchase order* (PO) dan melakukan *approval*.
- c. Meyakinkan pengiriman *material* dari gudang kepada departemen produksi ataupun departemen lainnya sesuai dengan *material requisition*.
- d. Menjaga *inventory level* sesuai standar.
- e. Mengatur siklus *inventory* sesuai dengan metode *First In First Out*.
- f. Meyakinkan semua catatan *inventory material* sama dengan aktual.
- g. Melakukan analisis kemampuan vendor.

2. *Supervisor Warehouse Raw Material (WRM)*

Tugas dan tanggung jawab:

- a. Mengkoordinasikan anak buah untuk mencapai sistem gudang yang baik.
- b. Berkoordinasi dengan divisi operasional.
- c. Berkoordinasi dengan semua departemen untuk mencapai sistem gudang yang baik.
- d. Memantau stok *material* penting dan kritis, serta melaporkan kepada atasan.
- e. Menganalisa dan membandingkan stok *material* pada sistem dengan aktual.
- f. Melakukan *follow up material* impor.
- g. Memantau dan menganalisa *prioritas loading* dan *unloading material*.
- h. Bertanggung jawab dalam penerapan dan formtasi ISO.
- i. Meyakinkan seluruh data operasional akurat dan *up to date*.
- j. Memantau ketepatan data *trimming* dan *scrap*.

3. *Inventory Planning Control*

Tugas dan tanggung jawab:

- a. Mengkoordinasikan anak buah untuk mencapai sistem gudang yang baik.
- b. Berkoordinasi dengan divisi operasional.
- c. Berkoordinasi dengan semua departemen untuk mencapai sistem gudang yang baik.
- d. Memantau stok *material* penting dan kritis, serta melaporkannya kepada atasan.
- e. Menganalisa dan membandingkan stok *material* pada sistem dengan aktual.
- f. Bertanggung jawab dalam penerapan struktur organisasi (SO) dan Formtasi struktur organisasi (SO).
- g. Meyakinkan seluruh data administrasi akurat dan *up to date*.
- h. Membuat laporan *inventory* mingguan dan bulanan.

4. *Stock Level Controller*

Tugas dan tanggung jawab:

- a. Berkoordinasi dengan divisi operasional.
- b. Memantau stok *material* penting dan kritis, serta melaporkannya kepada atasan.
- c. Menganalisa dan membandingkan stok *material* pada sistem dengan aktual.
- d. Melakukan *follow up material local* untuk kebutuhan produksi dan non produksi.
- e. Memantau dan melaporkan vendor yang bermasalah.
- f. Meyakinkan seluruh data administrasi akurat dan *up to date*.

5. *Input Data Processor*

Tugas dan tanggung jawab:

- a. Berkoordinasi dengan divisi operasional.
- b. Memantau stok *material* penting dan kritis, serta melaporkannya kepada atasan.
- c. Menganalisa dan membandingkan stok *material* pada sistem dengan aktual.
- d. Melakukan input *material requisition* (MR) ke sistem database *excel* secara aktual dan *up to date*.

- e. Melakukan input *good received requisition* (GRR) ke sistem database *excel* secara aktual dan *up to date*.
- f. Membuat surat jalan
- g. Memeriksa dan membuat laporan data akhir bulan secara aktual.
- h. Meyakinkan seluruh data administrasi akurat dan *up to date*.

6. *Foreman Warehouse Raw Material* (WRM)

Tugas dan tanggung jawab:

- a. Berkoordinasi dengan *Supervisor Warehouse Raw Material* (WRM).
- b. Memantau stok *material* penting dan kritis, serta melaporkannya kepada atasan.
- c. Menganalisa dan membandingkan stok *material* pada sistem dengan aktual.
- d. Meyakinkan seluruh data stok *material* akurat dan *up to date*.
- e. Menyiapkan dan mengirimkan *material* untuk memenuhi kebutuhan departemen terkait sesuai dengan *material requisition*.
- f. Mengontrol pekerjaan yang dilakukan *Part Control/ Incoming Part* berdasarkan *stock* atau pengadaan *material*.

7. *Incoming Part/ Part Control*

Tugas dan tanggung jawab:

- a. Berkoordinasi dengan *Supervisor Warehouse Raw Material* (WRM).
- b. Memantau stok *material* penting dan kritis, serta melaporkannya kepada atasan.
- c. Membandingkan stok *material* akurat dan *up to date*
- d. Meyakinkan seluruh data stok *material* akurat dan *up to date*.
- e. Menyiapkan dan mengirimkan *material* untuk memenuhi kebutuhan Departemen terkait sesuai dengan *material requisition*.

1.6 Bahan Baku

Bahan baku/*raw material* merupakan salah satu unsur yang paling aktif di dalam perusahaan yang secara terus-menerus diperoleh, diubah yang kemudian dijual kembali. Tanpa bahan baku/*raw material* proses produksi tidak akan berjalan. Berikut merupakan bahan baku/*raw material* komponen pembuatan *felt* pada PT Rekadaya Multi Adiprima:

1. *Fiber*

Fiber atau yang sering disebut sebagai serat ini berasal dari limbah-limbah pakaian seperti limbah baju, celana jeans dan jenis pakaian lain. *Fiber* menjadi bahan baku pokok dalam pembuatan *felt*, ada 4 jenis *fiber* yang dipakai oleh PT. Rekadaya Multi Adiprima untuk memproduksi *felt*, yaitu, *fiber C* (murni), *fiber melt* atau *fiber A* dan *fiber recycle B*. *Fiber C* adalah *fiber* murni yang berasal dari limbah pakaian, sedangkan *fiber melt* adalah *fiber* pakaian yang didalamnya terdapat campuran bahan *polimer* yang berfungsi sebagai pengeras. Sedangkan *fiber recycle B* adalah *fiber* yang berasal dari limbah *felt* dan *dacron*.

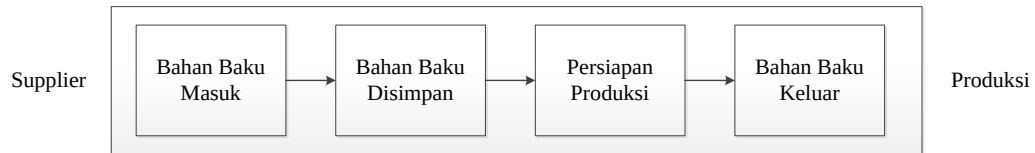


Gambar IV.3 *Fiber*
(Sumber: PT Rekadaya Multi Adiprima, 2016)

1.7 Proses dalam Persediaan Bahan Baku

Proses persediaan bahan baku yaitu suatu kumpulan aktivitas saling terkait yang dimulai dengan adanya pengiriman bahan baku dari *supplier* yang kemudian mengalami beberapa tahapan yaitu penerimaan, penyimpanan, persiapan untuk produksi dan pengeluaran. Setiap tahapan tersebut akan mengalami proses kontrol dan administrasi. Proses pembelian sampai kemudian bahan baku diproduksi dilakukan dibawah pengawasan divisi-divisi terkait perusahaan sehingga proses

persediaan bahan baku berjalan sesuai perencanaan. Gambar proses persediaan bahan baku pada PT Rekadaya Multi Adiprima, dapat dilihat pada Gambar IV.4.



Gambar IV.4 Proses Persediaan Bahan Baku.
(Sumber: PT Rekadaya Multi Adiprima, 2016)

Penjelasan proses persediaan bahan baku adalah sebagai berikut:

1. Bahan baku masuk

Kegiatan ini ada ketika bahan baku diterima perusahaan dari *supplier*, kemudian dilakukan pengecekan oleh petugas lapangan untuk memastikan kesesuaian antara jenis dan jumlah bahan baku yang ada di surat jalan dan PO (*purchase order*) dengan kondisi fisiknya.

2. Bahan baku disimpan

Penyimpanan bahan baku diletakkan pada gudang yang terdekat dengan mesin produksi sesuai spesifikasinya, penyimpanan dilakukan sebagai persediaan untuk kegiatan produksi.

3. Bahan baku keluar

Bahan baku keluar atau bahan baku yang diproduksi harus sesuai dengan form BOB yang sudah dibuat sebelumnya.

1.8 Sistem yang Sedang Berjalan

Untuk mengetahui sistem yang berjalan dilakukan pengamatan bagaimana cara kerja sistem yang digunakan di PT Rekadaya Multi Adiprima, khususnya mengenai sistem informasi persediaan bahan baku. Tujuan melakukan pengamatan sistem tersebut adalah agar dapat mengidentifikasi masalah-masalah

yang ada pada sistem informasi persediaan bahan baku sehingga dapat melakukan saran-saran perbaikan.

Pengamatan dilakukan berdasarkan urutan proses yang sekaligus menunjukkan pertukaran data seperti form yang terlihat, peran setiap divisi pada divisi *raw material* dan informasi proses pada sistem persediaan bahan baku. Proses pada sistem persediaan bahan baku terdiri dari prosedur penerimaan bahan baku, penyimpanan bahan baku, pemeriksaan persediaan bahan baku, penginputan, pencarian data bahan baku dan pembuatan laporan. Sistem Persediaan Bahan Baku yang berjalan dapat dilihat pada Gambar IV.5.

Penjelasan sistem informasi persediaan bahan baku adalah sebagai berikut:

1. Divisi Produksi membuat form BOB (Bukti Order Barang) sebagai permintaan dalam memenuhi kebutuhan bahan baku untuk diproduksi, kemudian diserahkan kepada Admin WRM.
2. Divisi *Warehouse Raw Material* mengecek data persediaan bahan baku pada Microsoft Excel apa saja yang diperlukan untuk produksi yang telah diajukan.
3. Jika persediaan bahan baku mencukupi maka Divisi *Warehouse Raw Material* akan meng-*input* data persediaan bahan baku, meng-*update* persediaan bahan baku, membuat Form BKB (Bukti Keluar Barang) untuk bukti pengeluaran bahan baku beserta barang untuk di produksi dan sebagai arsip di *Raw Material*.
4. Apabila persediaan bahan baku tidak mencukupi maka akan dibuatkan Form Permintaan PO
5. Form Permintaan PO atau permintaan pembelian bahan baku diserahkan ke Divisi *Purchasing* untuk melakukan pembelian bahan baku ke *Supplier*.
6. Divisi *Purchasing* mengecek Form Permintaan PO, sebagai acuan dalam pembuatan *purchase order*.
7. Kemudian *purchase order* akan dikirim ke *Supplier*.
8. *Supplier* menerima *purchase order*, lalu *Supplier* mengirim bahan baku bersama surat jalan ke *Quality Control*.
9. *Quality Control* melakukan pengecekan untuk memastikan kesesuaian antara surat jalan, PO dan kondisi fisik.
10. Apabila barang lolos inspeksi dan sudah sesuai *Quality Control* menandatangani surat jalan dan diberikan ke Divisi *Raw Material*. Sebaliknya apabila barang tidak lolos inspeksi *Quality Control* akan mengembalikan barang tersebut kepada *Supplier*.
11. Divisi *Raw Material* membuat Form laporan bahan baku masuk untuk dibuat sebagai arsip.

Persediaan Bahan Baku

Pembuatan form persediaan bahan baku dibuat berdasarkan pengamatan form yang telah dianalisis yang bertujuan untuk mengetahui prosedur yang digunakan PT Rekadaya Multi Adiprima.

1. Form Bukti Order Barang (BOB)

Pengamatan form dilakukan untuk mengetahui form apa saja yang ada dan digunakan dalam sistem informasi persediaan bahan baku pada PT Rekadaya Multi Adiprima. Salah satunya form Bukti Order Barang (BOB) atau laporan kebutuhan bahan baku produksi dapat dilihat pada Gambar IV.6.

PT REKADAYA MULTI ADIPRIMA
Tanggal: 01-01-2016

NO	NAMA PART	QUANTITY	KETERANGAN
1.	10. 5/16	10 pcs	✓
2.	10. 5/16	10 pcs	✓
3.	10. 5/16	10 pcs	✓
4.	10. 5/16	10 pcs	✓
5.	10. 5/16	10 pcs	✓
6.	10. 5/16	10 pcs	✓
7.	10. 5/16	10 pcs	✓
8.	10. 5/16	10 pcs	✓
9.	10. 5/16	10 pcs	✓
10.	10. 5/16	10 pcs	✓

Di Buat	Di Cek	Di Cek	Di Cek	Disetujui
Leader	Supervisor	Ka Produksi	PIC	Director

RMA/END/BOB/01/16

Gambar IV.6 Form Bukti Order Barang (BOB)
(Sumber: PT Rekadaya Multi Adiprima, 2016)

Pada Form Bukti Order Barang (BOB) terdapat nama *part*, *quantity*, dan keterangan. Form permintaan bahan baku ini dibuat setelah Divisi *Marketing* menerima pesanan dari Pembeli. Pesanan tersebut kemudian diberikan kepada Divisi Produksi untuk mengetahui berapa jumlah produksi yang akan dilakukan dan banyaknya jumlah bahan baku yang akan dipesan. Form ini dibuat untuk diteruskan ke Divisi *Warehouse Raw Material* jika persediaan di gudang tidak memenuhi maka akan dibuat Form *Material Kritis* (MK) yang akan dikirim ke Divisi *Purchasing*.

Penerimaan dan Penyimpanan Bahan Baku

Pembuatan laporan penerimaan dan penyimpanan bahan baku dibuat berdasarkan pengamatan form yang telah dianalisis yang bertujuan untuk mengetahui prosedur yang digunakan PT Rekadaya Multi Adiprima.

1. Form Surat Jalan Tanda Terima Barang (SJ)

Bentuk Form Surat Jalan Tanda Terima Barang pada PT Rekadaya Multi Adiprima dapat dilihat pada Gambar IV.7.

SURAT JALAN No. _____ Tujuan _____
 Lokasi _____

Kami kirimkan barang-barang tersebut dibawah ini dengan kendaraan No. _____

BANYAKNYA	NAMA BARANG

P.11 - KRW
 PARAF.

Hormat kami,



Gambar IV.7 Surat Jalan Tanda Penerimaan Barang
(Sumber: PT Rekadaya Multi Adiprima, 2016)

Pada Surat Jalan Tanda Terima Barang terdapat no. surat jalan, nama toko/tuan/supplier, nama kendaraan, no. kendaraan, no. PO, tanggal, banyaknya barang dan nama barang. Form ini dibuat ketika ada bahan baku masuk ke dalam gudang, sebelum dilakukan *input* harus cek terlebih dahulu kesamaan data yang ada di PO (*purchase order*) dan kondisi fisik.

2. Form Check Sheet Raw Material

Bentuk Form *Check Sheet Raw Material* dapat dilihat pada Gambar IV.8.

PT REKADAYA KREASI INDONESIA
NON WOVEN FABRIC

CHECK SHEET RAW MATERIAL

SUPPLIER : HARAPAN JAYA
JENIS MATERIAL FIBER

NO	BERAT DARI SUPPLIER (KG)	BERAT AKTUAL DI RUMAH (KG)	NO	BERAT DARI SUPPLIER (KG)	BERAT AKTUAL DI RUMAH (KG)
1	13.5	13.5	21	15.0	15.0
2	15.9	15.9	22	15.0	15.0
3	16.5	16.5	23	15.0	15.0
4	20.1	20.1	24	15.0	15.0
5	15.5	15.5	25	15.0	15.0
6	15.0	15.0	26	15.0	15.0
7	15.5	15.5	27	15.0	15.0
8	15.5	15.5	28	15.0	15.0
9	15.0	15.0	29	15.0	15.0
10	15.5	15.5	30	15.0	15.0
11	15.0	15.0	31	15.0	15.0
12	15.5	15.5	32	15.0	15.0
13	15.0	15.0	33		
14	15.0	15.0	34		
15	15.0	15.0	35		
16	15.0	15.0	36		
17	15.0	15.0	37		
18	15.0	15.0	38		
19	15.0	15.0	39		
20	15.0	15.0	40		

SUPPLIER : PT. RRI
Jumlah Ball : 32 Ball
Berat Total : 4.950 kg

PT. RRI
Jumlah Ball : 32 Ball
Berat Total : 4.950 kg

NOTE : SELUSIH

P.11 - KRW
PARAF :

Gambar IV.8 Form *Check Sheet Raw Material*
(Sumber: PT Rekadaya Multi Adiprima, 2016)

Form ini digunakan oleh QC untuk mencatat *actual* dari bahan baku atau *raw material* yang diterima oleh gudang. Form ini diperlukan untuk keperluan data produksi.

3. Form *Check Sheet Incoming Inspection Raw Material*

Bentuk Form Kartu Stok dapat dilihat pada Gambar IV.9.

PT REKADAYA KRAU INDONESIA
NON WOVEN FABRIC

CHECK SHEET INCOMING INSPECTION RAW MATERIAL

CUSTOMER: _____
MODEL: _____
PART NAME: _____
PART NUMBER: _____

NO	ITEM CHECK	STD	MODE	SAMPLE					AVERAGE	REMARK
				1	2	3	4	5		
1	PACKING	BAKUP	VISUAL							
2	WABE	ADA	VISUAL							
3	KONDISI									
1.1	TIDAK BAKU		VISUAL							
1.2	TIDAK BAKUP		VISUAL							
1.3	TIDAK ADA BAHAN METAL		VISUAL							

NOTE: SETAP KECANTANGAN MATERIAL, CHECK 5 BAKU MATERIAL SECARA ACHE

① = OK
X = NG

RESULT	
NG	OK
INSPECTOR	CHECKER

Gambar IV.9 Form *Check Sheet Incomming Inspection Raw Material*
(Sumber: PT Rekadaya Multi Adiprima, 2016)

Form ini digunakan oleh QC untuk mencatat kondisi dari *packing* bahan baku atau *raw material* yang diterima oleh gudang. Form ini diperlukan untuk keperluan data QC terkait kondisi barang yang diterima dalam kondisi baik (OK) atau tidak baik (NG). Form ini diisi saat team QC melakukan pengecekan sample barang yang dikirimkan supplier ke PT. RMA.

4. Form Kartu Stok

Bentuk Form Kartu Stok dapat dilihat pada Gambar IV.10.


PT REKADAYA KREASI INDONESIA
 NON WOVEN FABRIC

BULAN :

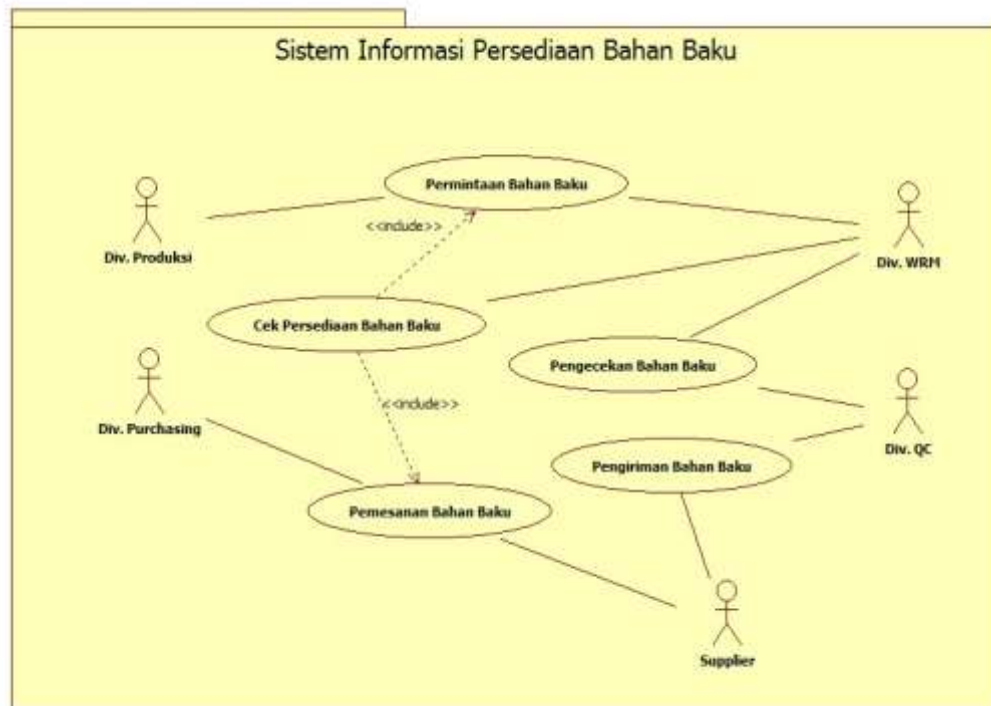
TGL	STOCK AWAL	MASUK	KELUAR	STOCK AKHIR	KETERANGAN

Gambar IV.10 Kartu Stok
(Sumber: PT Rekadaya Multi Adiprima, 2016)

Form kartu stok ini digunakan untuk mencatat berapa jumlah stock awal hasil produksi dan berapa jumlah barang yang dikeluarkan pada Shift 1 dan Shift 2 serta berapa jumlah stock akhir barang yang ada pada gudang. Kartu stock ini diisi oleh admin gudang berdasarkan *Form Check Sheet Incoming Inspection Raw Material* (CSIIRM) yang diisi oleh QC sebelumnya. Setelah diisi kartu stock akan diarsipkan oleh admin WRM dan disimpan di dinding rak yang berada di *Warehouse Raw Material*.

Pemakaian Bahan Baku

Use case diagram menggambarkan siapa saja aktor yang melakukan prosedur dalam sistem serta fungsi-fungsi atau proses yang terlibat dalam transformasi pada sistem tersebut. *Use case diagram* sistem persediaan bahan baku saat ini pada PT RMA, dapat dilihat pada Gambar IV.12.



Gambar IV.12 *Use Case Diagram* Sistem yang Berjalan
(Sumber: PT Rekadaya Multi Aduiprima, 2016)

Penjelasan *use case diagram* sistem saat ini, dapat dijelaskan sebagai berikut:

1. Definisi Aktor

Pendefinisian actor pada *use case* sistem saat ini dapat dilihat pada Tabel IV.1.

Tabel IV.1 Definisi Aktor dan Use Case Diagram

No.	Aktor	Deskripsi
1	Divisi Produksi	Orang yang bertugas untuk melakukan produksi dan memberikan informasi laporan mengenai kebutuhan bahan baku untuk diproduksi.
2	Divisi Raw Material	Orang yang bertugas memeriksa laporan kebutuhan bahan baku dan melakukan pengecekan di gudang. Jika mencukupi, maka bahan baku akan segera dipersiapkan. Jika tidak, jumlah yang tertera pada laporan permintaan bahan baku akan disesuaikan dengan persediaan bahan baku yang tersedia. Selain itu Divisi ini bertugas juga dalam penerimaan bahan baku yang sudah dilakukan pengecekan dan dimasukkan ke gudang.
3	Purchasing	Orang yang bertugas melakukan pemesanan bahan baku ke <i>Supplier</i> jika bahan baku di gudang sudah hampir habis.
4	Supplier	Orang yang memiliki bahan baku, jika ada pemesanan bahan baku maka akan dicek bahan baku apa saja yang dipesan dan langsung dikirim kepada pemesannya.
5	Quality Control	Orang yang bertugas melakukan pemeriksaan kualitas bahan baku yang datang dari <i>Supplier</i> , jika bahan baku sudah sesuai dengan standar maka bahan baku akan segera dikirim ke gudang, jika tidak sesuai dengan standar maka bahan baku akan dikembalikan kepada <i>Supplier</i> .

(Sumber: Hasil Analisis Data, 2017)

2. Definisi use case

Pendefinisian *use case* pada *use case* sistem saat ini, dapat dilihat pada Tabel IV.2.

Tabel IV.2 Definisi Use Case Permintaan Bahan Baku

Nama <i>Use Case</i>	Permintaan Bahan Baku
Deskripsi <i>Use Case</i>	<i>Use case</i> ini menggambarkan permintaan bahan baku ke Divisi <i>Warehouse Raw Material</i> .
Aktor	Staff Produksi, Staff Divisi <i>Warehouse Raw Material</i> .
<i>Relationship</i>	
<i>Normal Flow Events:</i>	1. Staf Produksi membuat Form Bukti Order Barang (BOB). 2. Staf Produksi menyerahkan Form Bukti Order Barang (BOB) ke Divisi <i>Warehouse Raw Material</i>. 3. Staff Divisi <i>Warehouse Raw Material</i> menerima Form Bukti Order Barang (BOB) dari Divisi Produksi dan melakukan pengecekan persediaan bahan baku.

(Sumber: Hasil Analisis Data, 2017)

Tabel IV.3 Definisi Use Case Cek Persediaan Bahan Baku

Nama <i>Use Case</i>	Cek Persediaan Bahan Baku
Deskripsi <i>Use Case</i>	<i>Use case</i> ini menggambarkan pengecekan persediaan bahan baku yang tersedia di gudang.
Aktor	Staff Divisi <i>Warehouse Raw Material</i> .
<i>Relationship</i>	
<i>Normal Flow Events:</i>	1. Divisi <i>Warehouse Raw Material</i> menerima Form Bukti Order Barang (BOB) dari Divisi Produksi. 2. Staff Divisi <i>Warehouse Raw Material</i> melakukan pengecekan persediaan bahan baku berdasarkan Form Bukti Order Barang (BOB)..

(Sumber: Hasil Analisis Data, 2017)

Tabel IV.4 Definisi Use Case Pemesanan Bahan Baku

Nama <i>Use Case</i>	Pemesanan Bahan Baku.
Deskripsi <i>Use Case</i>	<i>Use case</i> ini menggambarkan pemesanan bahan baku karena bahan baku tidak mencukupi.
Aktor	Staff Divisi <i>Warehouse Raw Material</i> , Staff <i>Purchasing</i> .
<i>Relationship</i>	
<i>Normal Flow Events:</i>	1. Staff Divisi <i>Warehouse Raw Material</i> membuat Form <i>Material Kritis</i> (MK). 2. Staff Divisi <i>Warehouse Raw Material</i> menyerahkan Form <i>Material Kritis</i> (MK) ke Divisi <i>Purchasing</i>. 3. Divisi <i>Purchasing</i> menerima Form <i>Material Kritis</i> (MK) membuat <i>purchase order</i> (PO) untuk melakukan pemesanan ke <i>Supplier</i>.

(Sumber: Hasil Analisis Data, 2017)

Tabel IV.5 Definisi Use Case Pengiriman Bahan Baku

Nama <i>Use Case</i>	Pengiriman Bahan Baku.
Deskripsi <i>Use Case</i>	<i>Use case</i> ini menggambarkan pengiriman bahan baku oleh <i>Supplier</i> .
Aktor	<i>Supplier</i> , Staff Divisi <i>Quality Control</i> .
<i>Relationship</i>	
<i>Normal Flow Events:</i>	1. <i>Supplier</i> mengirimkan bahan baku sesuai <i>purchase order</i> (PO). 2. Staff Divisi <i>Quality Control</i> mengecek kualitas bahan

	baku.
--	-------

(Sumber: Hasil Analisis Data, 2017)

Tabel IV.6 Definisi Use Case Pengecekan Bahan Baku

Nama <i>Use Case</i>	Pengecekan Bahan Baku.
Deskripsi <i>Use Case</i>	<i>Use case</i> ini menggambarkan pengecekan kualitas bahan baku oleh <i>Quality Control</i> .
Aktor	Staff Divisi <i>Quality Control</i> , Staff Divisi <i>Warehouse Raw Material</i>

(Sumber: Hasil Analisis Data, 2017)

Tabel IV.6 Definisi Use Case Pengecekan Bahan Baku (Lanjutan)

<i>Relationship</i>	
<i>Normal Flow Events:</i>	1. Staff Divisi <i>Quality Control</i> mengecek kualitas bahan baku yang diterima dari <i>Supplier</i>. Jika kualitas bahan baku tidak sesuai standar maka akan dilakukan retur pembelian. 2. Staff Divisi <i>Quality Control</i> mengirim bahan baku ke Divisi <i>Warehouse Raw Material</i>.

(Sumber: Hasil Analisis Data, 2017)

BAB V

ANALISIS DAN PEMBAHASAN

5.1 Tahap Analisis

Berdasarkan hasil pengolahan data dan analisis sistem, dibutuhkan sistem informasi persediaan bahan baku yang terintegrasi agar mempermudah dalam mengolah dan memproses pengecekan persediaan bahan baku. Sistem informasi persediaan bahan baku menampilkan data persediaan bahan baku, data bahan baku masuk dan data bahan baku keluar. Untuk mendukung sistem informasi persediaan bahan baku dibutuhkan kemampuan *user* dalam mengoperasikannya. Maka untuk mewujudkan hal tersebut dibuat sistem informasi yang *user friendly*, sehingga lebih mudah pengoperasiannya.

Kebutuhan program sistem informasi 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 sistem informasi, dimana proses-proses tersebut dibutuhkan oleh pemakai. Berikut adalah penjelasan identifikasi kebutuhan *user* sistem informasi persediaan bahan baku di PT Rekadaya Multi Adiprima.

Tabel V.1 Identifikasi Kebutuhan User

Masalah	Kebutuhan User	Kebutuhan Sistem	Uraian	
			<i>Functional Requirement</i>	<i>Non Functional Requirement</i>
	Dapat menambah, mengubah, menghapus, dan memilih	Pengelolaan data master karyawan.	Hanya dapat menampilkan <i>username</i> , <i>contact</i> , <i>address</i>	

	data master karyawan.		dan bagian.	
--	-----------------------	--	-------------	--

Tabel V.1 Identifikasi Kebutuhan User (Lanjutan)

Masalah	Kebutuhan User	Kebutuhan Sistem	Uraian	
			Functional Requirement	Non Functional Requirement
	Dapat menambah, mengubah dan menghapus data master supplier.	Pengelolaan data master supplier.	Hanya dapat menampilkan supplier name, contact, fax, address, dan description.	
	Dapat menambah, mengubah dan menghapus data master bahan baku.	Pengelolaan data master bahan baku.	Hanya dapat menampilkan BB numb, BB name, description, UOM, ukuran, stock, safety stock, kebutuhan /hari, ,lead time dan ROP.	
Permintaan bahan baku masih dilakukan dengan penulisan tangan sering kali terjadi	Dapat menginput, mencari, mengubah, menghapus dan menambah data BB Request.	Pengelolaan data transaksi BB Request	Hanya dapat menampilkan nomor transaksi, tanggal, keterangan, kode bahan baku, dan jumlah.	

kesalahan dalam penulisan data bahan baku.				
--	--	--	--	--

Tabel V.1 Identifikasi Kebutuhan User (Lanjutan)

Masalah	Kebutuhan User	Kebutuhan Sistem	Uraian	
			<i>Functional Requirement</i>	<i>Non Functional Requirement</i>
Informasi yang dihasilkan hanya ditampilkan dalam bentuk buku catatan.	Dapat menginput, mencari, mengubah, menghapus dan menambah data PO Request.	Pengelolaan data transaksi PO Request	Hanya dapat menampilkan nomor transaksi, tanggal, keterangan, kode bahan baku, dan jumlah.	
Kegiatan pembuatan PO memiliki tahapan yang cukup panjang dan memakan waktu yang lama.	Dapat memvalidasi data PO Request apabila terdapat data bahan baku yang akan dilakukan pemesanan dan mencari data transaksi PO Request.	Pengelolaan data transaksi Purchase Order.	Hanya dapat menampilkan nomor transaksi, tanggal, data PO Request dan keterangan.	

Informasi yang dihasilkan hanya ditampilkan dalam bentuk kartu stok.	Dapat menginput, mencari, dan menambah data BB Masuk	Pengelolaan data transaksi BB Masuk.	Hanya dapat Menampilkan nomor transaksi, tanggal masuk, DO, <i>supplier</i> , keterangan, kode bahan baku, dan jumlah.	
--	--	--------------------------------------	--	--

Tabel V.1 Identifikasi Kebutuhan User (Lanjutan)

Masalah	Kebutuhan User	Kebutuhan Sistem	Uraian	
			<i>Functional Requirement</i>	<i>Non Functional Requirement</i>
Informasi yang dihasilkan hanya ditampilkan dalam bentuk kartu stok.	Dapat menginput, mencari, dan menambah data BB Keluar	Pengelolaan data transaksi BB Keluar.	Hanya dapat menampilkan nomor transaksi, tanggal, data BB <i>Request</i> , dan keterangan.	
Pembuatan laporan bahan baku masuk masih dengan penulisan tangan	Dapat mencetak bukti penerimaan bahan baku apabila bahan baku masuk sesuai dengan SJ, lolos dari	Mencetak Laporan BB Masuk	Hanya dapat menampilkan tanggal, ID BB Masuk, DO, jumlah, dan <i>supplier</i> .	

	inspeksi dan SJ telah divalidasi berdasarkan PO.			
Pembuatan laporan bahan baku keluar masih dengan penulisan tangan	Dapat mencetak bukti pengeluaran bahan baku apabila bahan baku yang di <i>request</i> oleh produksi tersedia pada stok di gudang.	Mencetak Laporan BB Keluar	Hanya dapat menampilkan tanggal, BB ID, <i>request number</i> , jumlah, dan <i>out by</i> .	

(Sumber: Hasil Analisis Data, 2017)

5.2 Perancangan Sistem

Perancangan sistem merupakan suatu kegiatan pengembangan serta perbaikan terhadap sebuah sistem yang berjalan. Pada tahap ini dilakukan upaya memperbaiki sistem atau membangun sistem dengan memanfaatkan teknologi baru untuk mengurangi dan mengatasi berbagai permasalahan yang telah terjadi pada sistem yang lama sehingga dapat meningkatkan efektifitas dan efisiensi. Sistem informasi yang dirancang diharapkan memberi solusi alternatif baru yang memberi kemudahan terhadap proses persediaan bahan baku di PT Rekadaya Multi Adiprima.

5.2.1 Tujuan Sistem yang Diusulkan

Tujuan pengembangan sistem adalah untuk memberikan penjelasan dan mendefinisikan kepada *user* yang akan menggunakan program. Pembuatan program ini diharapkan dapat mengatasi kekurangan yang ada dan membantu *user* dalam pengecekan persediaan bahan baku di gudang. Adapun perancangan yang diusulkan merupakan langkah untuk lebih mengefektifkan dan

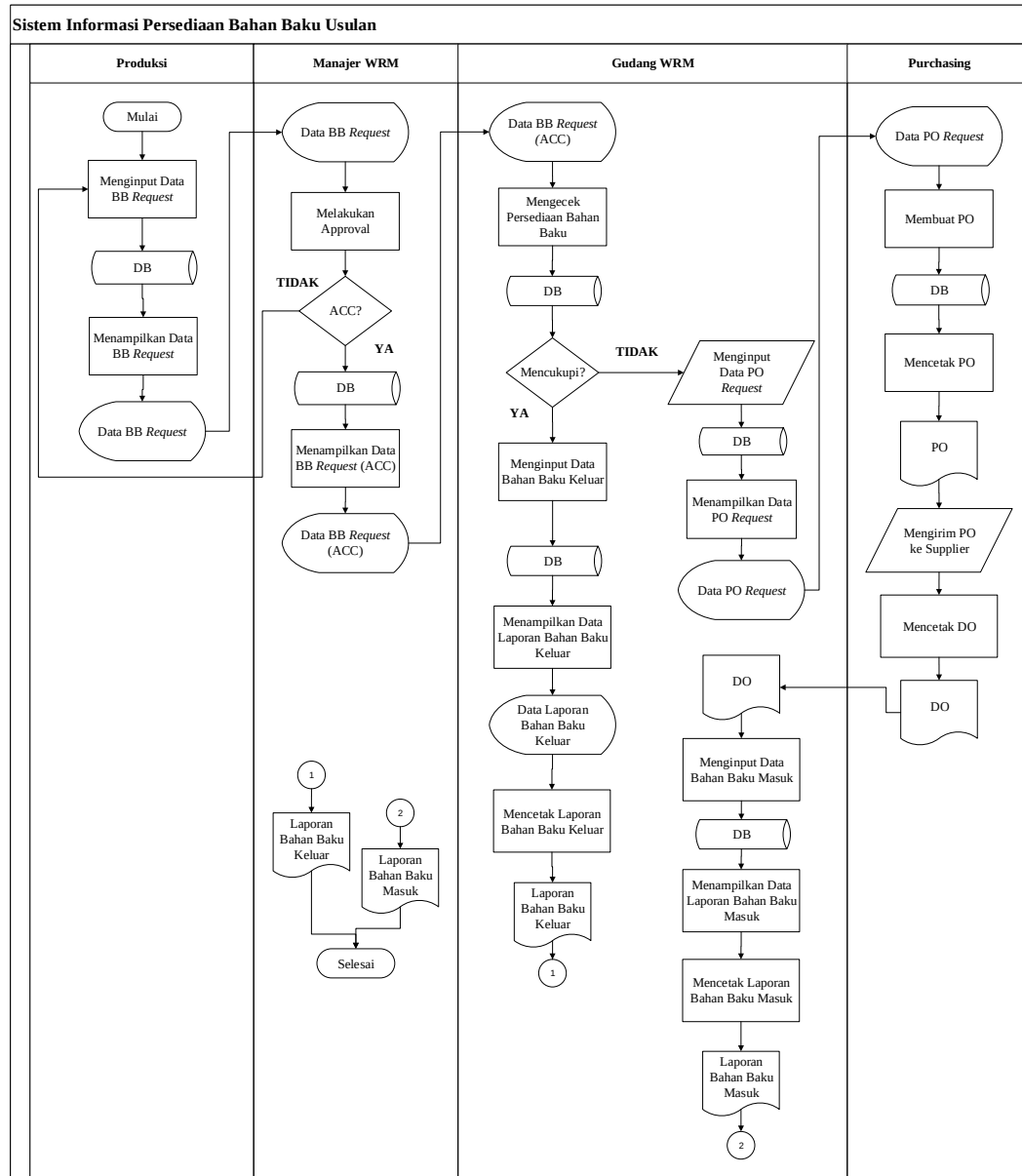
mengefisienkan sistem yang sudah terkomputerisasi dengan menggunakan sistem informasi.

Tujuan utama dari perancangan sistem adalah untuk memberikan gambaran secara umum kepada pemakai mengenai sistem informasi yang baru. Adapun tujuan dari perancangan sistem yang diusulkan yaitu:

1. Mengembangkan sistem yang sudah terkomputerisasi dengan menggunakan sistem informasi.
2. Memudahkan dalam pengecekan bahan baku di gudang.

5.2.2 Perancangan Sistem yang Diusulkan

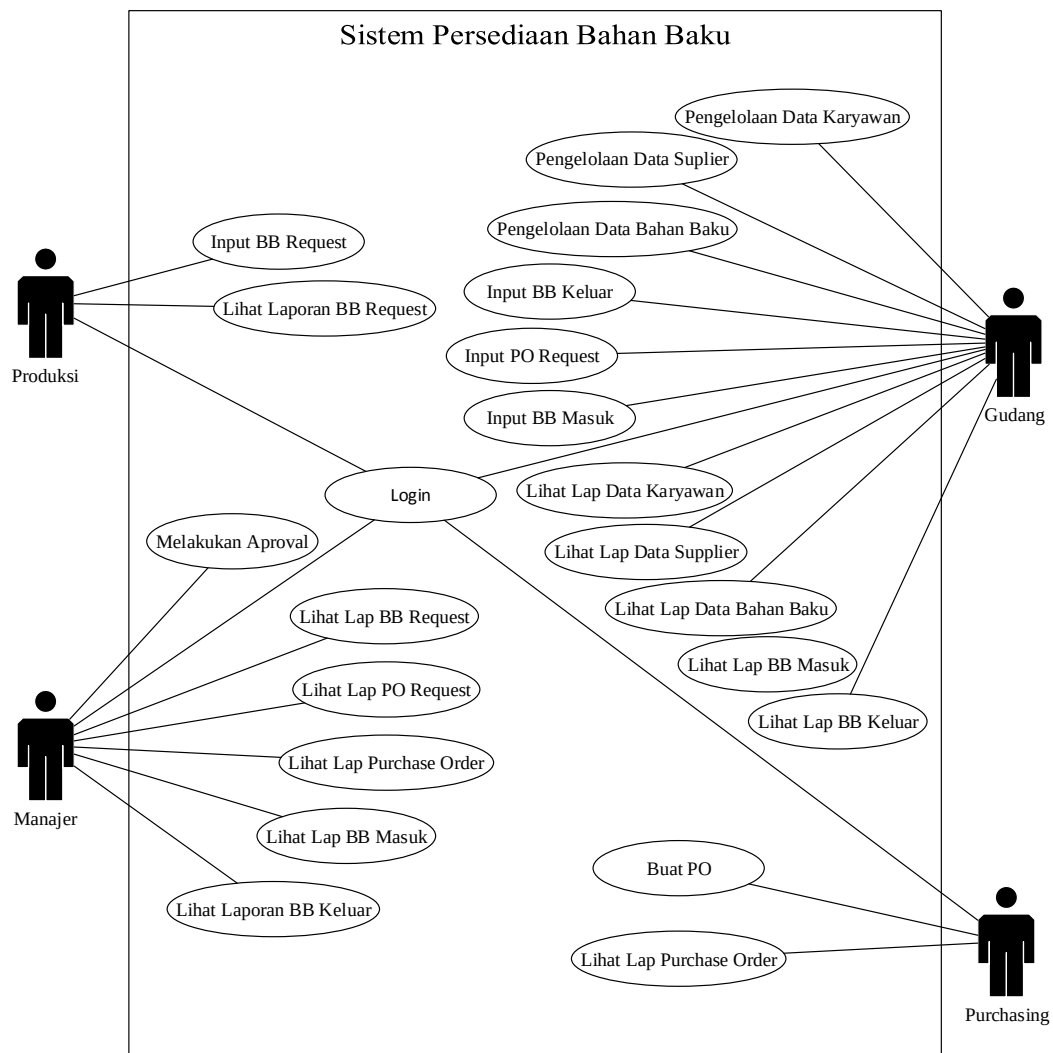
Proses yang diusulkan bertujuan untuk mengembangkan prosedur dan proses suatu data agar tujuan dari suatu organisasi dapat tercapai. Perancangan dibuat untuk meminimalkan kekurangan dari sistem lama yang kurang efektif dan efisien ke sistem baru yang terintegrasi dengan basis data sehingga lebih mudah dan cepat dalam pengerjaan proses pengecekan bahan baku di PT Rekadaya Multi Adiprima. *Flowmap* Program Usulan Sistem Persediaan Bahan Baku dapat dilihat pada Gambar V.1.



Gambar V.1 *Flowmap* Program Usulan Sistem Persediaan Bahan Baku
(Sumber: Hasil Analisis Data, 2017)

5.2.3 Use Case Diagram yang Diusulkan

Use case diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case diagram* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Alasan memilih *Use case diagram* untuk menggambarkan sistem yang diusulkan, agar mempermudah dalam menggambarkan dan membaca sistem informasi yang akan diusulkan. *Usecase diagram* usulan sistem persediaan bahan baku pada PT Rekadaya Multi Adiprima dapat dilihat pada Gambar V.2.



Gambar V.2 *Use Case Diagram* Sistem Persediaan Bahan Baku yang Diusulkan
(Sumber: Hasil Analisis Data, 2017)

Penjelasan *use case diagram* sistem saat ini, dapat dijelaskan sebagai berikut:

1. Definisi Aktor

Berikut ini adalah deskripsi pendefinisian aktor pada sistem informasi persediaan bahan baku.

Tabel V.2. Definisi Aktor Sistem Usulan

No.	Aktor	Deskripsi
1.	Produksi	Orang yang bertugas meminta bahan baku untuk diproduksi
2.	Gudang	Orang yang bertugas dan memiliki hak akses untuk melakukan operasi pengelolaan data bahan baku, data bahan baku masuk, data bahan baku keluar, laporan bahan baku masuk, bahan baku keluar, dan membuat permintaan PO.
3.	Manajer	Orang yang hanya mempunyai hak akses untuk menyetujui permintaan bahan baku produksi dan melihat laporan bahan baku masuk dan bahan baku keluar.
4.	<i>Purchasing</i>	Orang yang menerima dokumen permintaan PO dan membuat PO (<i>Purchase Order</i>).

(Sumber: Hasil Analisis Data, 2017)

2. Definisi Use Case

Pendefinisian *use case* pada *use case* sistem usulan, dapat dilihat pada tabel berikut ini.

Tabel V.3 Definisi Use Case Login Sistem Usulan

Nama Use Case	Login.
Deskripsi Use Case	<i>Use case</i> ini menggambarkan penginputan <i>username</i> dan <i>password</i> untuk masuk ke aplikasi.
Aktor	Gudang, Manajer, Produksi, Purchasing.
Relationship	

Tabel V.3 Definisi Use Case Login Sistem Usulan (Lanjutan)

<i>Normal Flow Events:</i>	1. Karyawan menginputkan <i>username</i> dan <i>password</i>. 2. Jika <i>username</i> dan <i>password</i> benar, maka akan masuk ke menu utama, tetapi jika salah maka akan ada alert <i>username</i> dan <i>password</i> salah dan perintah untuk melakukan input ulang <i>username</i> dan <i>password</i>.
----------------------------	---

(Sumber: Hasil Analisis Data, 2017)

Tabel V.4 Definisi Use Case Pengelolaan Data Bahan Baku Sistem Usulan

Nama Use Case	Pengelolaan Data Bahan Baku.
Deskripsi Use Case	Use case ini menggambarkan pengelolaan data bahan baku oleh Divisi Gudang.
Aktor	Gudang
Relationship	
<i>Normal Flow Events:</i>	1. Karyawan Gudang menginput data bahan baku yang belum terdapat pada <i>database</i> . 2. Karyawan Gudang meng-<i>update</i> data bahan baku pada <i>Database</i>. 3. Karyawan Gudang dapat menghapus data bahan baku pada <i>database</i>. 4. Karyawan Gudang mengecek persediaan bahan baku pada <i>database</i>. Jika bahan baku cukup maka akan dicetak form BKB dan mengeluarkan bahan baku. Jika bahan baku tidak cukup maka akan diinput data PO <i>Request</i>

(Sumber: Hasil Analisis Data, 2017)

Tabel V.5 Definisi Use Case Input BB Request Sistem Usulan

Nama Use Case	Input BB Request
Deskripsi Use Case	Use case ini menggambarkan Divisi Produksi memberikan permintaan bahan baku kepada Divisi Gudang.

Aktor	Produksi, Manajer.
-------	--------------------

Tabel V.5 Definisi Use Case Input BB Request Sistem Usulan (Lanjutan)

<i>Relationship</i>	
<i>Normal Flow Events:</i>	3. Karyawan Produksi meminta bahan baku. 4. Karyawan Gudang menerima BB <i>Request</i> dan Manager melakukan <i>Approval</i> .

(Sumber: Hasil Analisis Data, 2017)

Tabel V.6 Definisi Use Case Input BB Masuk Sistem Usulan

Nama Use Case	Input BB Masuk
Deskripsi Use Case	Use case ini menggambarkan penginputan data bahan baku yang masuk oleh Divisi Gudang.
Aktor	Gudang
<i>Relationship</i>	
<i>Normal Flow Events:</i>	1. Karyawan Gudang memilih menu transaksi. 2. Karyawan Gudang menginputkan data-data bahan baku masuk.

(Sumber: Hasil Analisis Data, 2017)

Tabel V.7 Definisi Use Case Input PO Request Sistem Usulan

Nama Use Case	Input PO Request
Deskripsi Use Case	Use case ini menggambarkan Divisi Gudang memberikan PO Request kepada Divisi Purchasing.
Aktor	Gudang

<i>Relationship</i>	
<i>Normal Flow Events:</i>	1. Karyawan Gudang memberikan PO Request. 2. Karyawan Purchasing menerima PO Request, memprosesnya dan mencetak PO .

(Sumber: Hasil Analisis Data, 2017)

Tabel V.8 Definisi Use Case Input BB Keluar Sistem Usulan

<i>Nama Use Case</i>	Input BB Keluar
<i>Deskripsi Use Case</i>	<i>Use case</i> ini menggambarkan penginputan data bahan baku yang keluar untuk diproduksi.
<i>Aktor</i>	Gudang
<i>Relationship</i>	
<i>Normal Flow Events:</i>	1. Karyawan Gudang memilih menu transaksi. 2. Karyawan Gudang menginputkan data-data bahan baku keluar.

(Sumber: Hasil Analisis Data, 2017)

Tabel V.9 Definisi Use Case Buat PO Sistem Usulan

<i>Nama Use Case</i>	Buat PO
<i>Deskripsi Use Case</i>	<i>Use case</i> ini menggambarkan proses pembuatan PO di <i>Purchasing</i> .
<i>Aktor</i>	<i>Purchasing</i>

<i>Relationship</i>	
<i>Normal Flow Events:</i>	1. Karyawan <i>Purchasing</i> memilih menu transaksi. 2. Karyawan <i>Purchasing</i> menerima data <i>PO Request</i> dan mencetak PO.

(Sumber: Hasil Analisis Data, 2017)

Tabel V.10 Definisi Use Case Melakukan Approval Sistem Usulan

Nama Use Case	Melakukan <i>Approval</i>
Deskripsi Use Case	<i>Use case</i> ini menggambarkan proses <i>Approval</i> pada permintaan bahan baku yang dilakukan oleh Manajer.
Aktor	Manajer
<i>Relationship</i>	
<i>Normal Flow Events:</i>	1. Manajer memilih menu Transaction Data, kemudian BB <i>Request</i> 2. Manajer memilih data BB <i>Request</i> berdasarkan nama bahan baku, kode bahan baku dan tanggal. 3. Manajer dapat mencetak data-data BB <i>Request</i>

(Sumber: Hasil Analisis Data, 2017)

Tabel V.11 Definisi Use Case Lihat Laporan Bahan Baku Masuk Sistem Usulan

Nama Use Case	Lihat Laporan Bahan Baku Masuk
Deskripsi Use Case	<i>Use case</i> ini menggambarkan fungsi untuk melihat data bahan baku masuk yang telah diinput.
Aktor	Karyawan Gudang, Manajer
<i>Relationship</i>	
<i>Normal Flow Events:</i>	1. Karyawan Gudang dan Manajer memilih menu laporan. 2. Karyawan Gudang dan Manajer memilih data bahan baku masuk berdasarkan nama bahan baku, kode bahan baku

	dan tanggal. 3. Karyawan Gudang dan Manajer dapat mencetak data-data bahan baku masuk
--	--

(Sumber: Hasil Analisis Data, 2017)

Tabel V.12 Definisi Use Case Lihat Laporan Bahan Baku Keluar Sistem Usulan

Nama Use Case	Lihat Laporan Bahan Baku Keluar
Deskripsi Use Case	Use case ini menggambarkan fungsi untuk melihat data bahan baku keluar yang telah diinput.
Aktor	Karyawan Gudang, Manajer
Relationship	
Normal Flow Events:	1. Karyawan Gudang dan Manajer memilih menu laporan. 2. Karyawan Gudang dan Manajer memilih data bahan baku keluar berdasarkan nama bahan baku, kode bahan baku, dan tanggal. 3. Karyawan Gudang dan Manajer dapat mencetak data-data bahan baku keluar

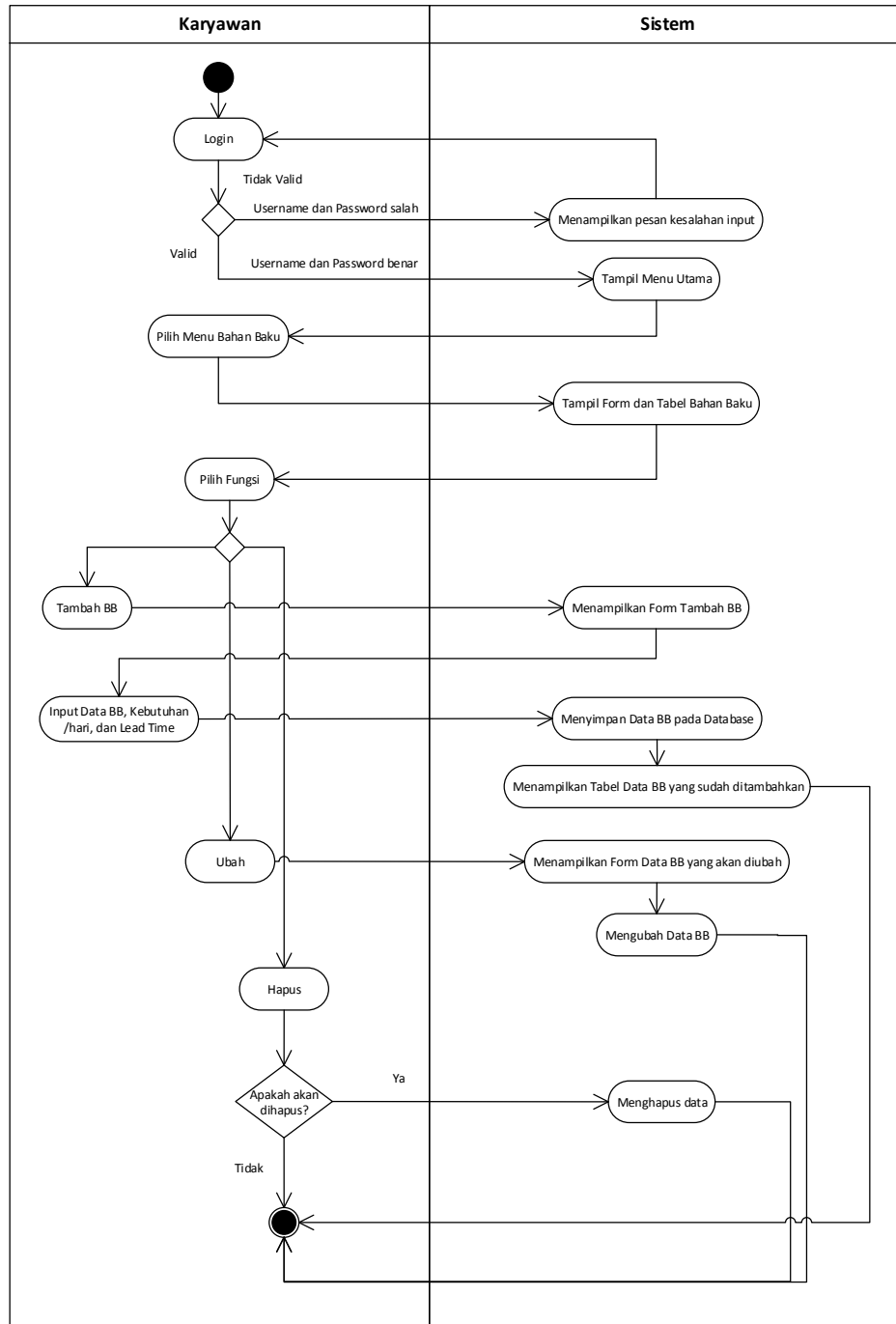
(Sumber: Hasil Analisis Data, 2017)

5.2.4 Activity Diagram yang Diusulkan

Activity diagram digunakan untuk menggambarkan kegiatan-kegiatan yang ada di dalam sistem. Agar lebih memahami sistem yang akan dibuat, maka perlu dibuatkan *activity diagram* tentang sistem informasi persediaan bahan baku ini.

1. Activity Diagram Kelola Data Bahan Baku

Activity Diagram Kelola Data Bahan Baku berikut ini merupakan diagram aktivitas yang menunjukkan fungsi-fungsi yang dapat dilakukan pada data bahan baku. *Activity diagram* data bahan baku dapat dilihat pada Gambar V.3 sebagai berikut:

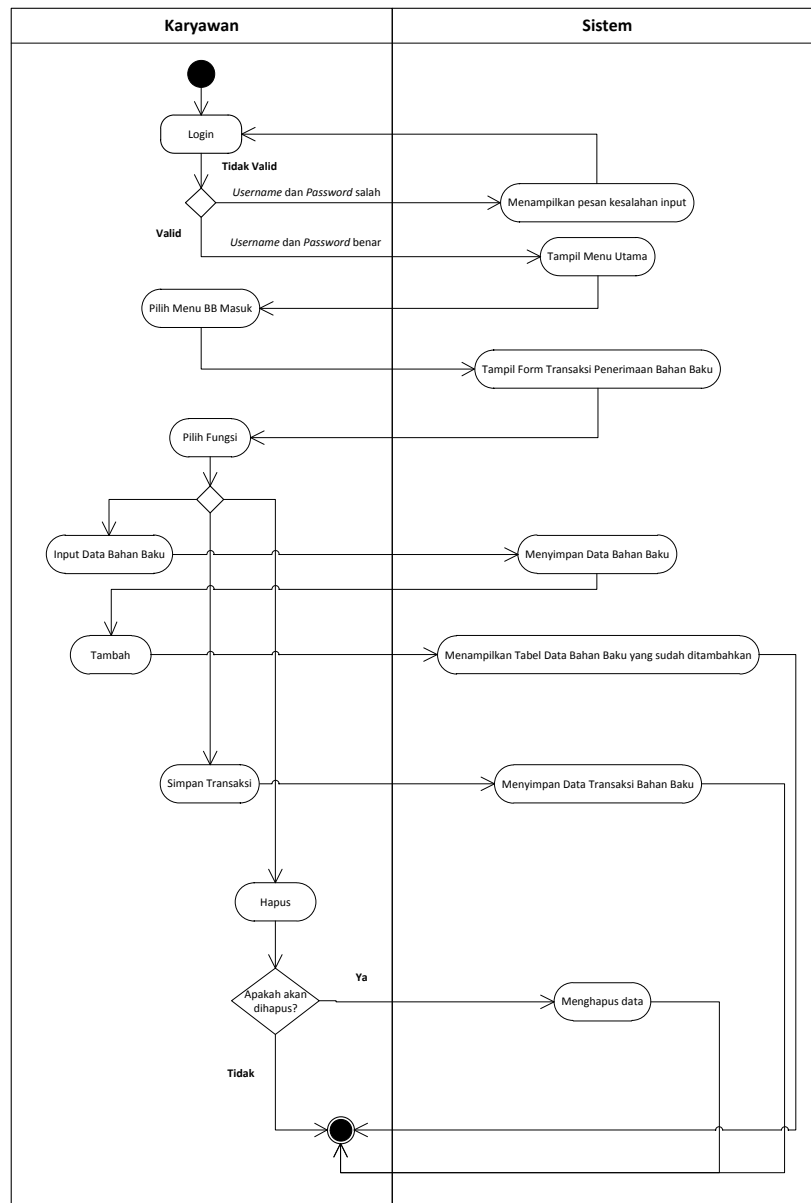


Gambar V.3 Activity Diagram Kelola Data Bahan Baku
(Sumber: Analisis Data, 2017)

2. Activity Diagram Input Data BB Masuk

Activity Diagram Input Data BB Masuk berikut ini merupakan diagram aktivitas yang menunjukkan fungsi-fungsi yang dapat dilakukan pada

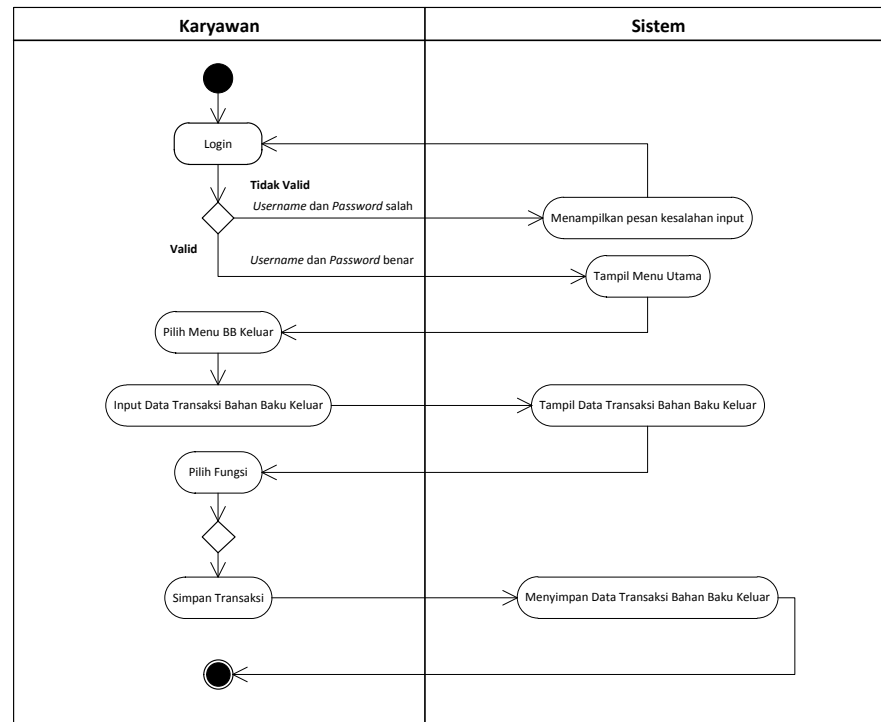
input data bahan baku masuk. Activity diagram input data bahan baku masuk dapat dilihat pada Gambar V.4 sebagai berikut:



Gambar V.4 Activity Diagram Input Data BB Masuk
(Sumber: Analisis Data, 2017)

3. Activity Diagram Input Data BB Keluar

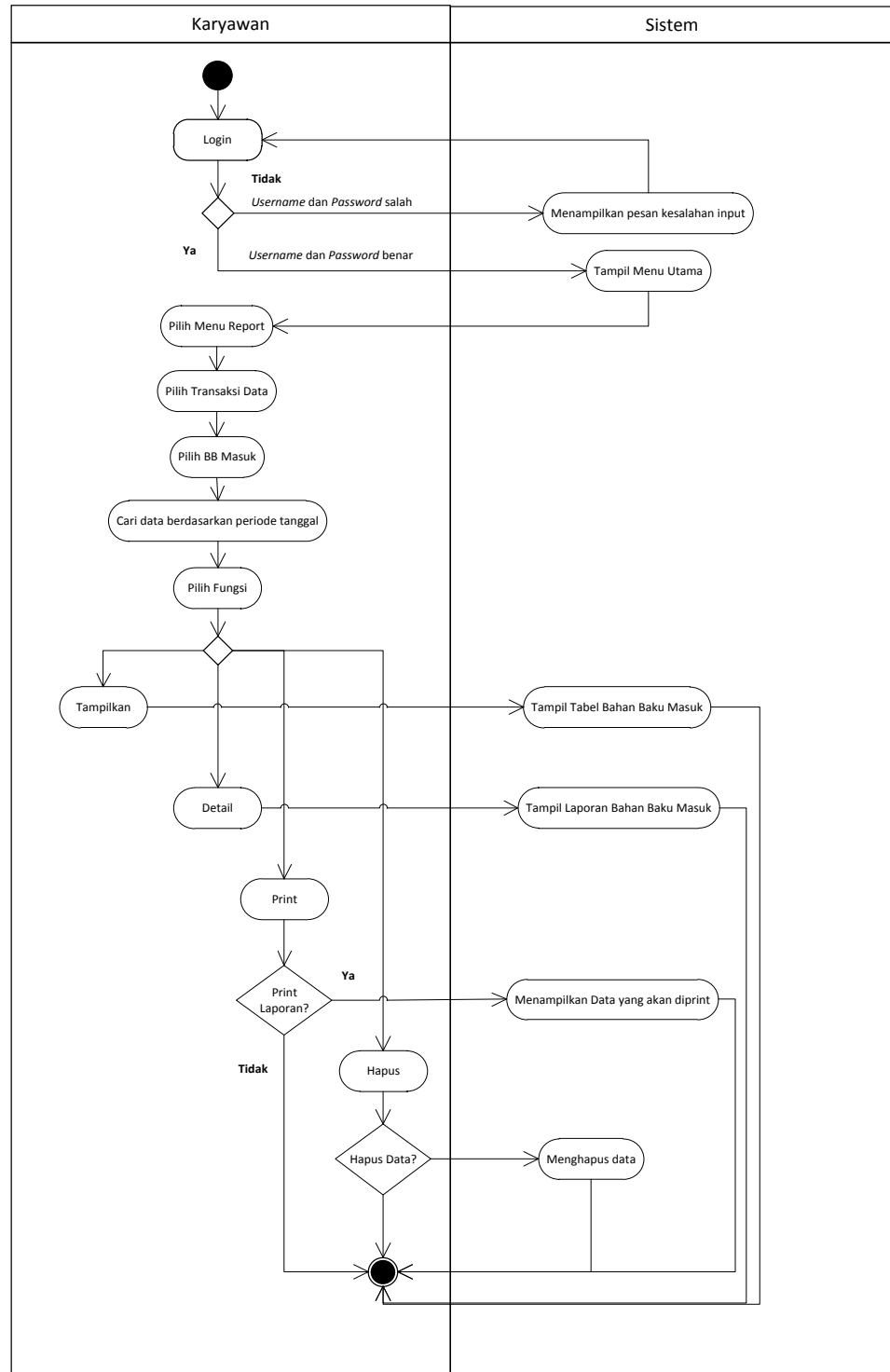
Activity Diagram Input Data BB Keluar berikut ini merupakan diagram aktivitas yang menunjukkan fungsi-fungsi yang dapat dilakukan pada *input data bahan baku keluar*. *Activity diagram input data bahan baku keluar* dapat dilihat pada Gambar V.5 sebagai berikut:



Gambar V.5 *Activity Diagram Input Data BB Keluar*
(Sumber: Analisis Data, 2017)

4. *Activity Diagram* Lihat Laporan BB Masuk

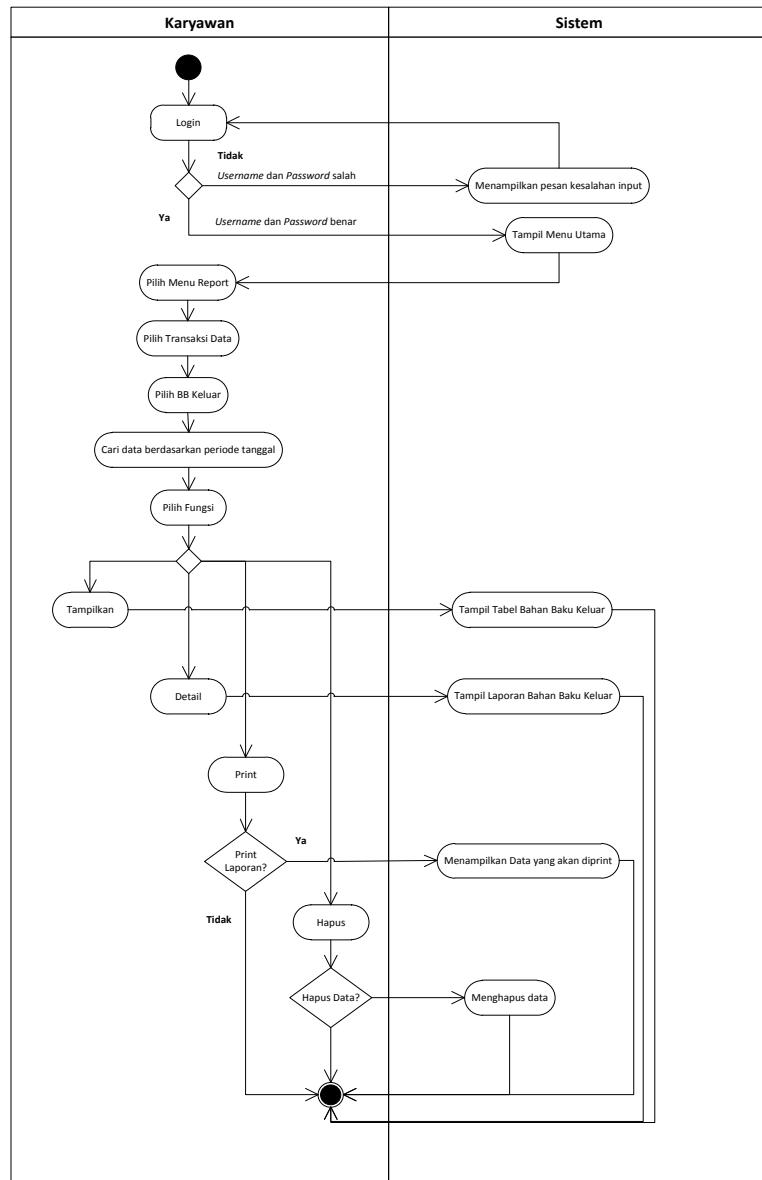
Activity Diagram Lihat Laporan BB Masuk berikut ini merupakan diagram aktivitas yang menunjukkan fungsi-fungsi yang dapat dilakukan pada lihat laporan bahan baku masuk. *Activity diagram* lihat laporan bahan baku masuk dapat dilihat pada Gambar V.6 sebagai berikut:



Gambar V.6 Activity Diagram Lihat Laporan BB Masuk
(Sumber: Analisis Data, 2017)

5. *Activiy Diagram* Lihat Laporan BB Keluar

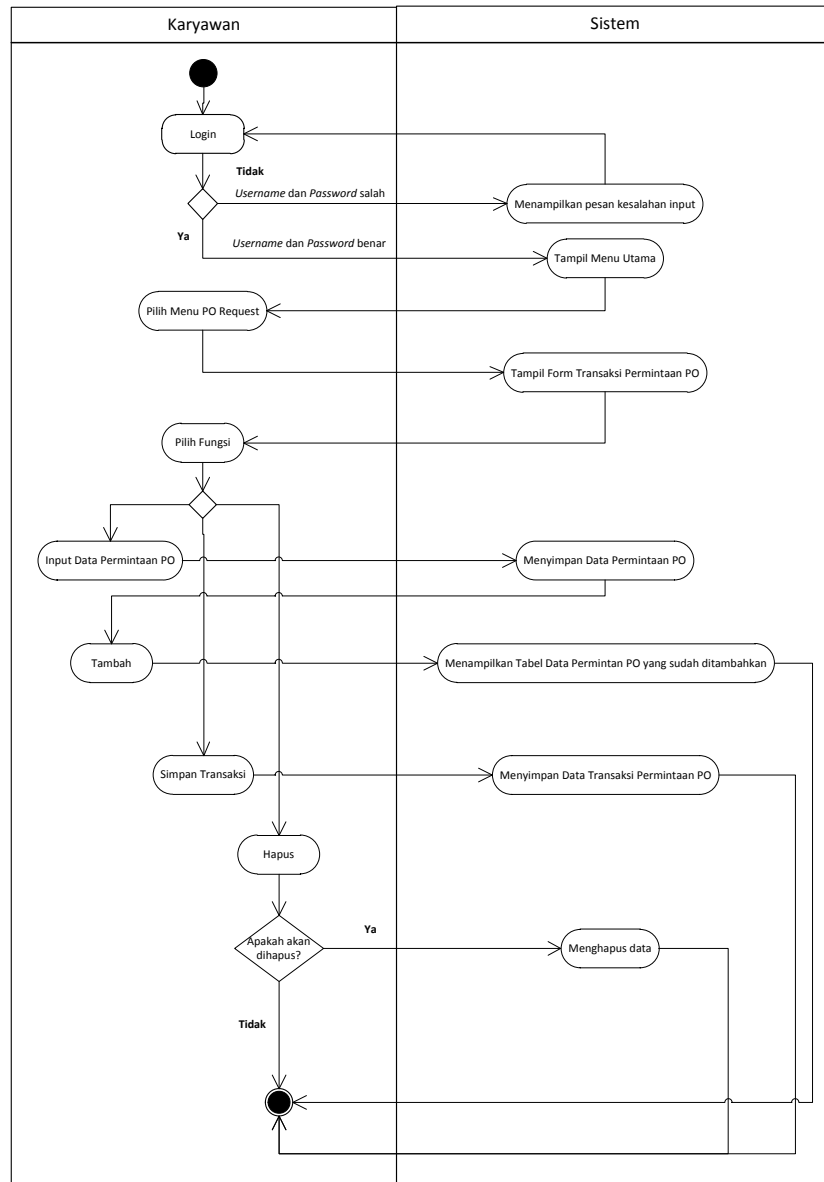
Activity Diagram Lihat Laporan BB Keluar berikut ini merupakan diagram aktivitas yang menunjukkan fungsi-fungsi yang dapat dilakukan pada lihat laporan bahan baku keluar. *Activity diagram* lihat laporan bahan baku keluar dapat dilihat pada Gambar V.7 sebagai berikut:



Gambar V.7 *Activity Diagram* Lihat Laporan BB Keluar
(Sumber: Analisis Data, 2017)

6. Activity Diagram PO Request

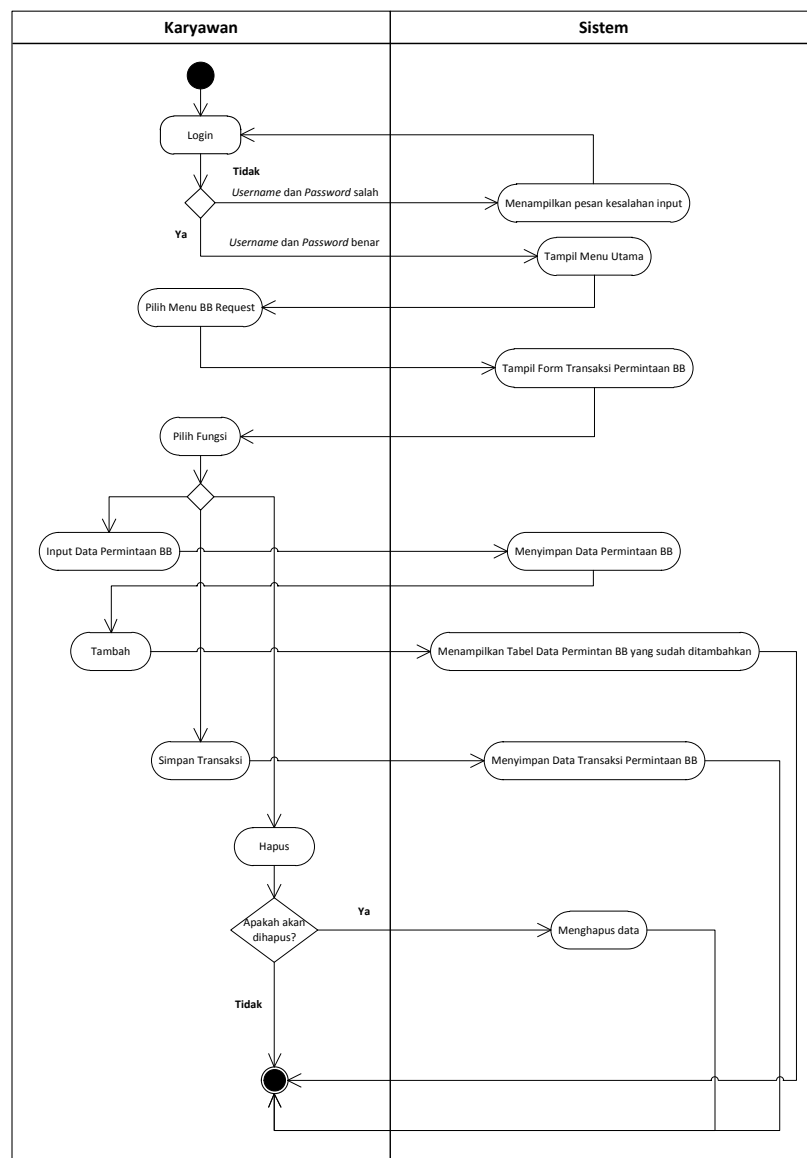
Activity Diagram PO Request berikut ini merupakan diagram aktivitas yang menunjukkan fungsi-fungsi yang dapat dilakukan pada permintaan PO yang ditujukan kepada Divisi *Purchasing*. Activity diagram PO Request dapat dilihat pada Gambar V.8 sebagai berikut:



Gambar V.8 Activity Diagram PO Request
(Sumber: Analisis Data, 2017)

7. Activity Diagram BB Request

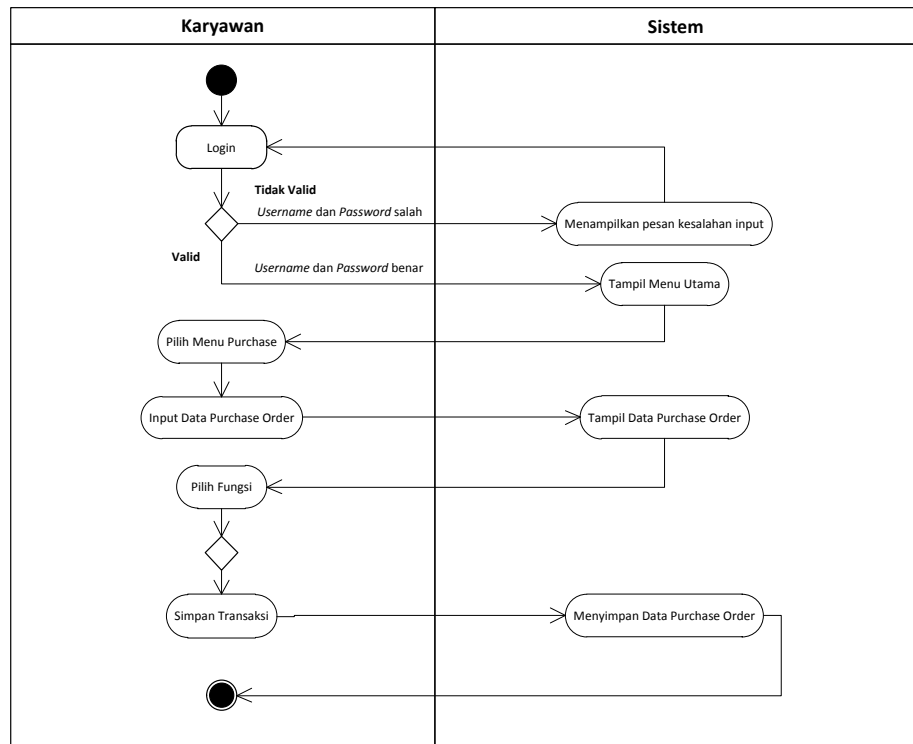
Activity Diagram BB Request berikut ini merupakan diagram aktivitas yang menunjukkan fungsi-fungsi yang dapat dilakukan pada permintaan bahan baku oleh Divisi Produksi. Activity diagram BB Request dapat dilihat pada Gambar V.9 sebagai berikut:



Gambar V.9 Activity Diagram BB Request
(Sumber: Analisis Data, 2017)

8. Activity Diagram Buat PO

Activity Diagram Buat PO berikut ini merupakan diagram aktivitas yang menunjukkan fungsi-fungsi yang dapat dilakukan pada pembuatan purchase order oleh Divisi Purchasing. Activity diagram Buat PO dapat dilihat pada Gambar V.10 sebagai berikut:



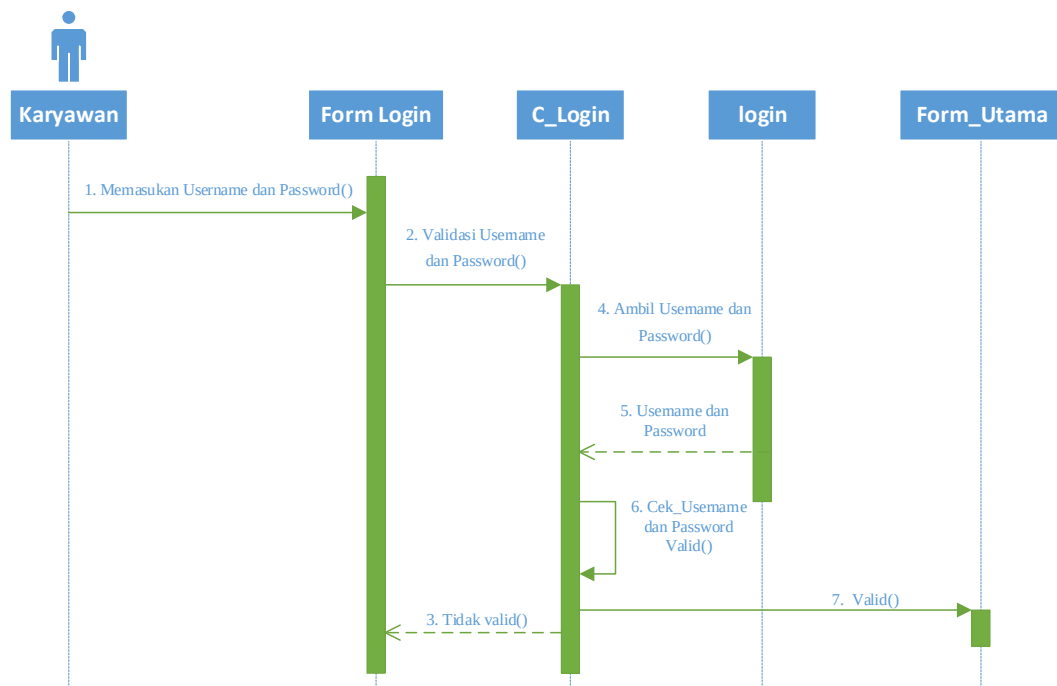
Gambar V.10 Activity Diagram Buat PO
(Sumber: Analisis Data, 2017)

5.2.5 Sequence Diagram yang Diusulkan

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup obyek dan *message* yang dikirimkan dan diterima obyek. *Sequence diagram* yang ditampilkan pada menu data master sistem ini hanya data master persediaan bahan baku karena untuk transaksi lain bentuknya sama. Berikut adalah *sequence diagram* dari sistem informasi persediaan bahan baku yang diusulkan:

1. Sequence Diagram Proses Login

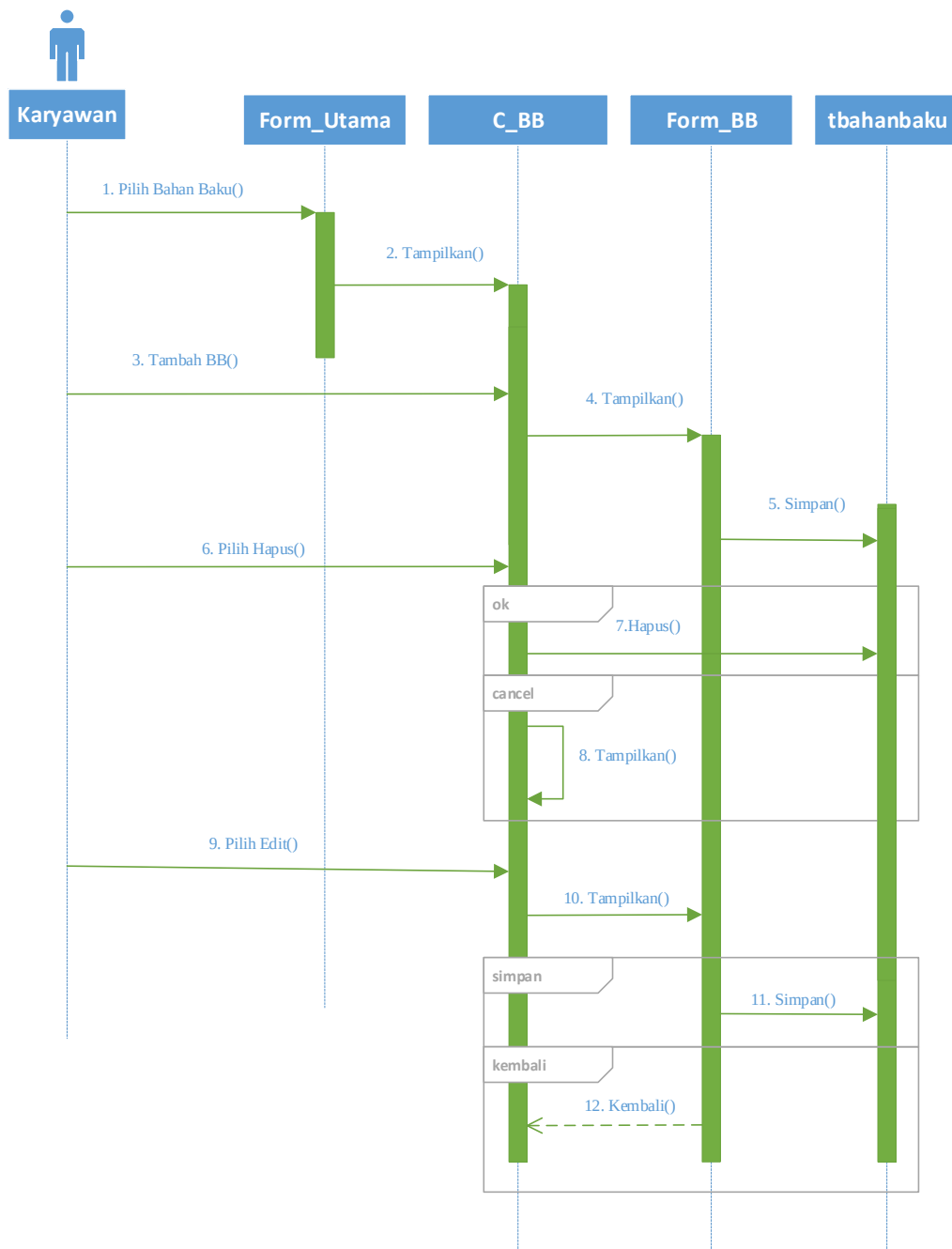
Sequence diagram login menjelaskan sebuah *sequence diagram* dalam proses *login*. Proses ini dilakukan oleh karyawan yang mempunyai hak akses sebelum masuk ke sistem. Adapun *sequence diagram* dari *use case login* dapat dilihat pada Gambar V.11.



Gambar V.11 Sequence Diagram Proses Login
(Sumber: Analisis Data, 2017)

2. Sequence Diagram Pengelolaan Data Bahan Baku

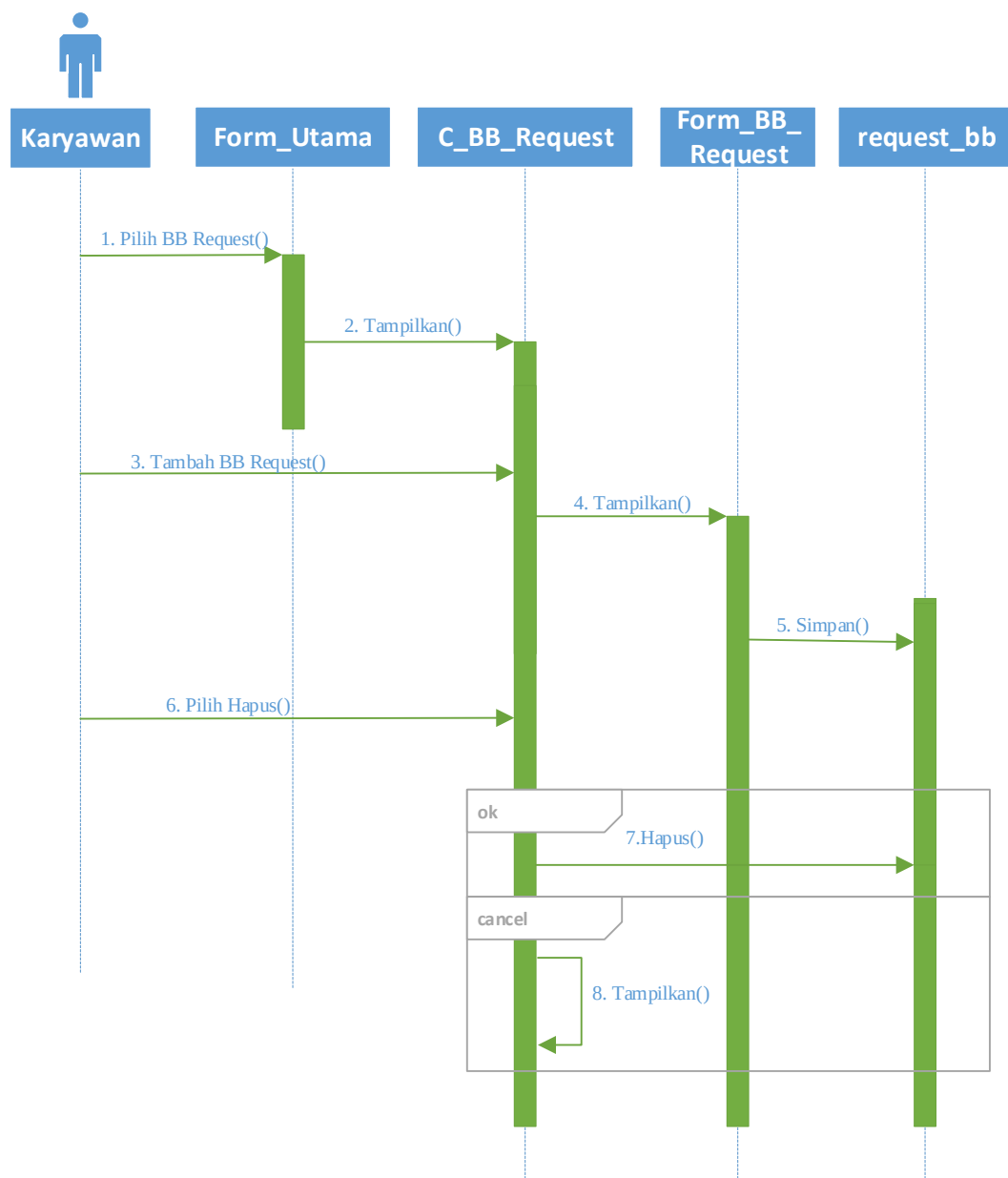
Sequence diagram pengelolaan data bahan baku menjelaskan sebuah *sequence diagram* dalam transaksi menambah, menghapus dan mengubah data bahan baku. Adapun *sequence diagram* dari *use case* pengelolaan data bahan baku dapat dilihat pada Gambar V.12.



Gambar V.12 *Sequence Diagram* Pengelolaan Data Bahan Baku
(Sumber: Analisis Data, 2017)

3. *Sequence Diagram Input BB Request*

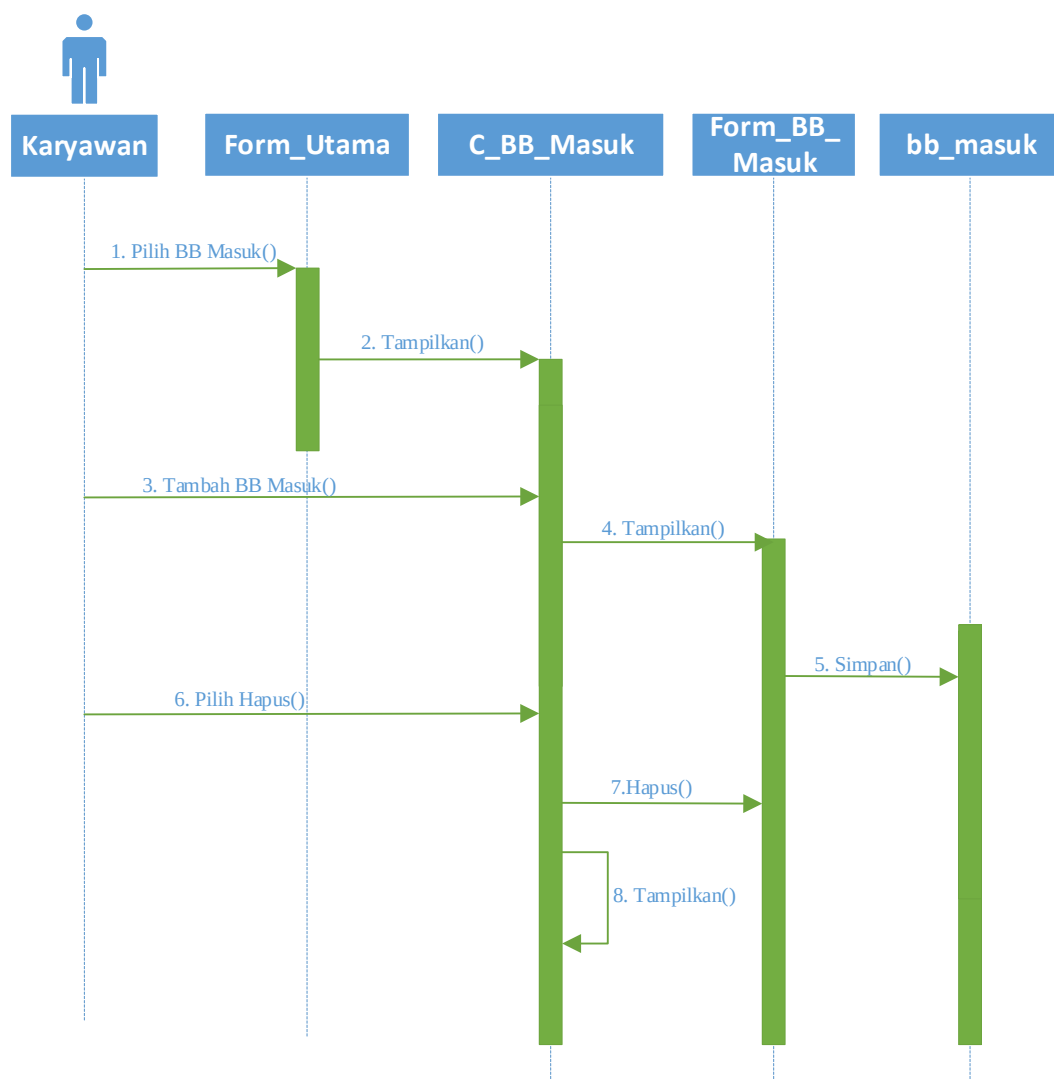
Pada *Sequence Diagram Input BB Request* dijelaskan pertama karyawan harus *login*, lalu masuk ke *main menu* pilih transaksi data *BB Request*, maka akan muncul form untuk *input* data permintaan BB. Setelah meng-*input* data akan diproses dan masuk ke *database*.



Gambar V.13 *Sequence Diagram Input BB Request*
(Sumber: Analisis Data, 2017)

4. *Sequence Diagram Input BB Masuk*

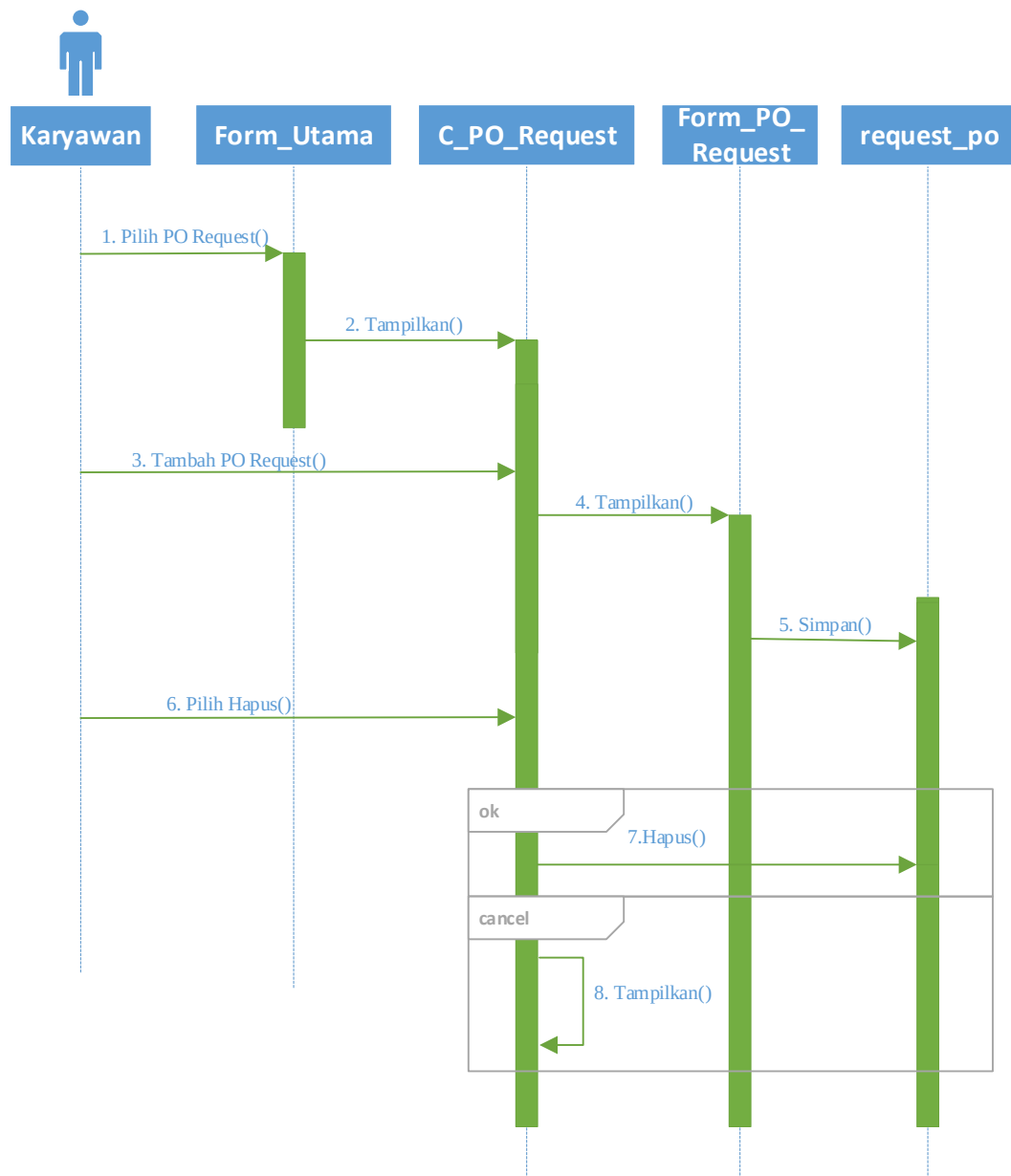
Pada *Sequence Diagram Input BB Masuk* dijelaskan pertama karyawan harus *login*. Karyawan akan masuk ke *main menu* pilih transaksi data BB Masuk, akan muncul form untuk *input* data bahan baku masuk, setelah meng-*input* data akan diproses dan masuk ke *database*.



Gambar V.14 *Sequence Diagram Input BB Masuk*
(Sumber: Analisis Data, 2017)

5. *Sequence Diagram Input PO Request*

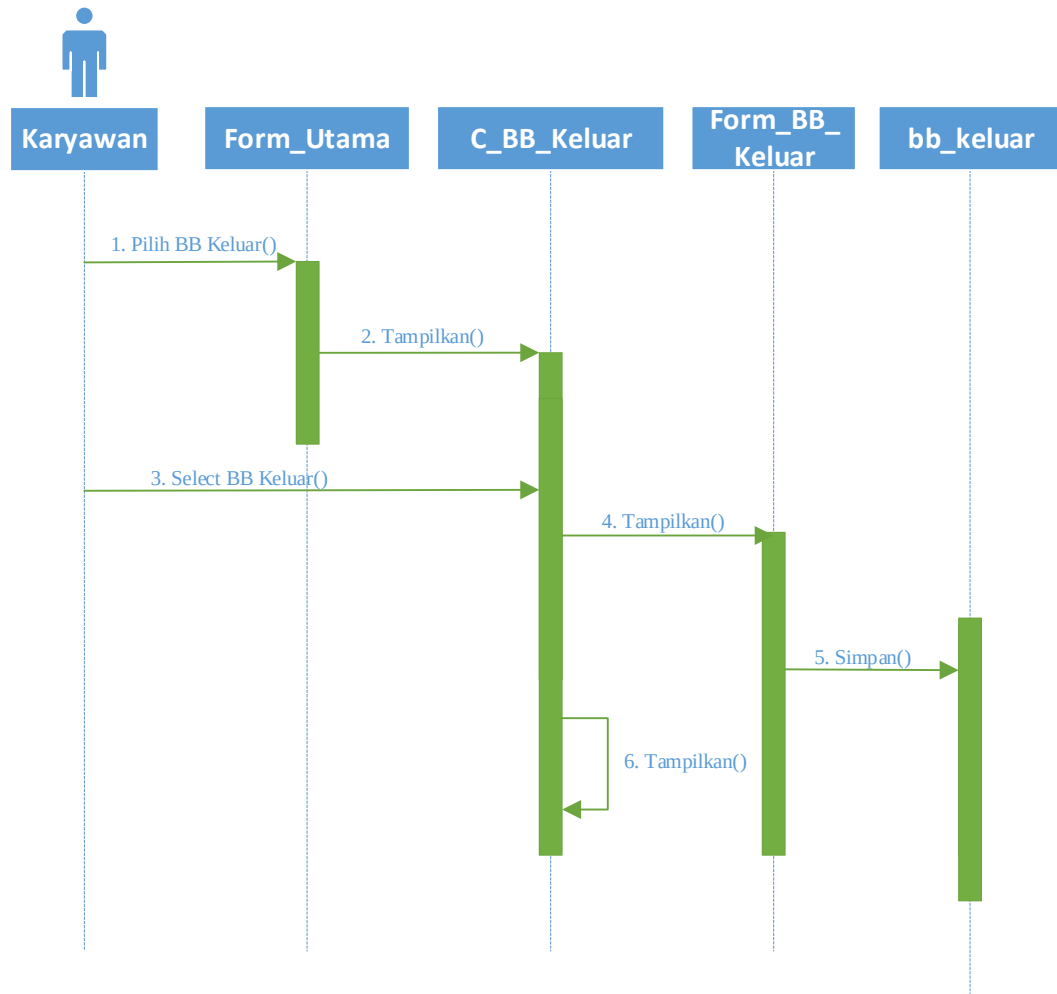
Pada *Sequence Diagram Input PO Request* dijelaskan pertama karyawan harus *login*, lalu masuk ke *main menu* pilih transaksi data *PO Request*, maka akan muncul form untuk *input* data permintaan PO. Setelah meng-*input* data akan diproses dan masuk ke *database*.



Gambar V.15 *Sequence Diagram Input PO Request*
(Sumber: Analisis Data, 2017)

6. *Sequence Diagram Input BB Keluar*

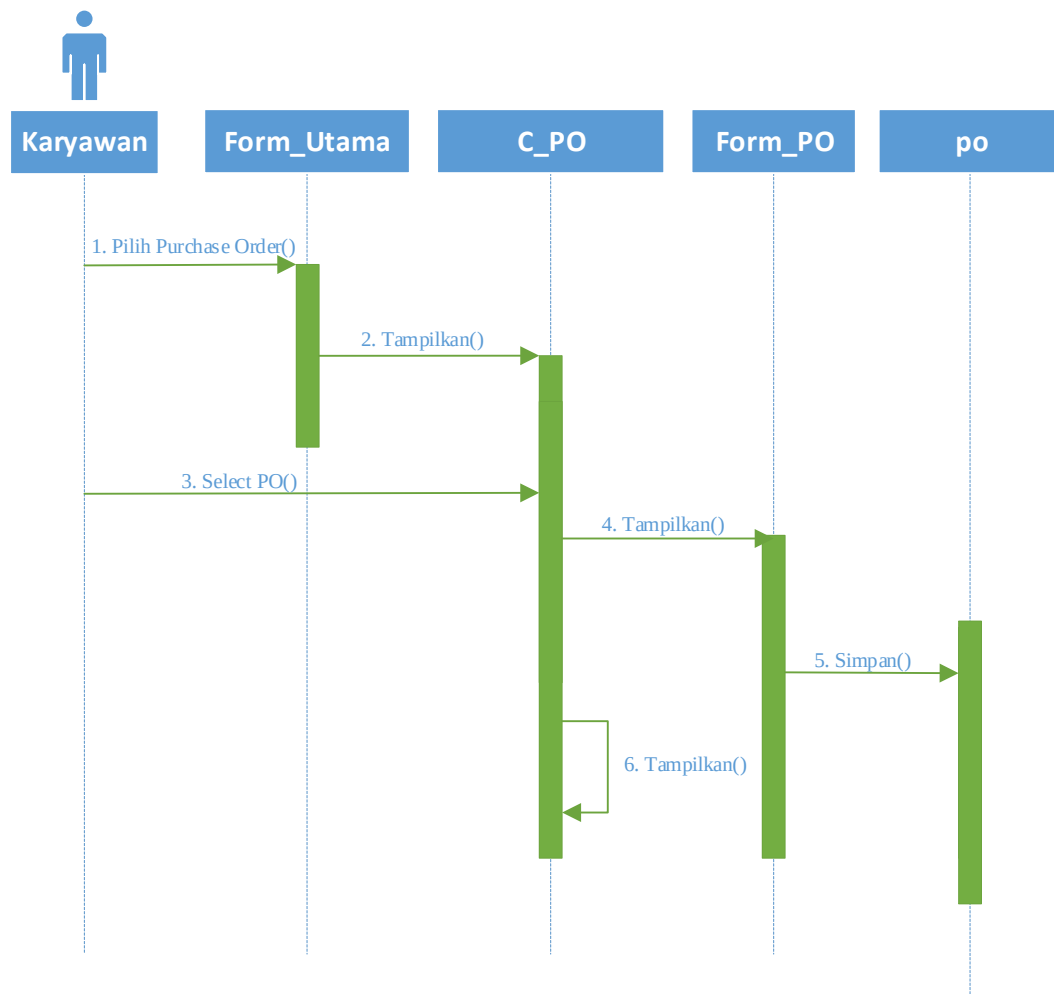
Pada *Sequence Diagram Input BB Keluar* dijelaskan pertama karyawan harus *login*, lalu masuk ke *main menu* pilih transaksi data BB Keluar. Kemudian akan muncul form untuk *input* data bahan baku keluar, setelah meng-*input* data akan diproses dan masuk ke *database*.



Gambar V.16 *Sequence Diagram Input BB Keluar*
(Sumber: Analisis Data, 2017)

7. *Sequence Diagram Buat PO*

Sequence diagram Buat PO menjelaskan sebuah *sequence diagram* dalam transaksi membuat *Purchase Order* (PO). Adapun *sequence diagram* dari *use case* buat PO dapat dilihat pada Gambar V.17.

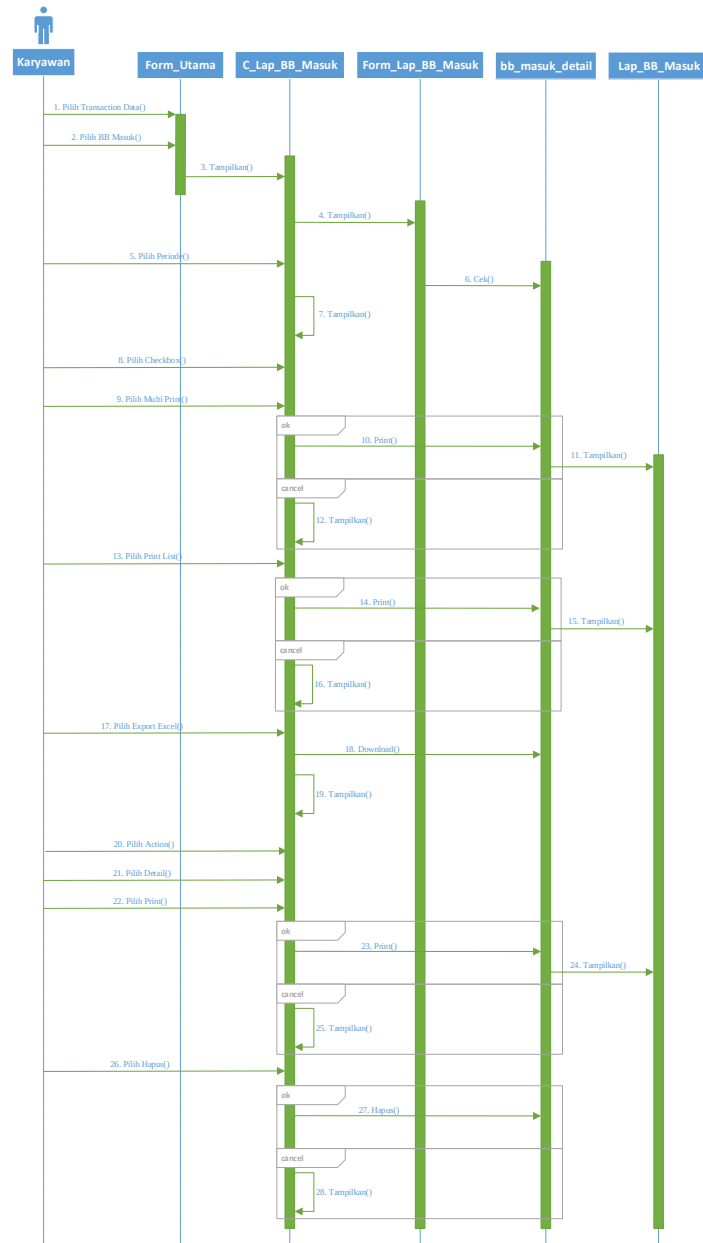


Gambar V.17 *Sequence Diagram* Buat PO
(Sumber: Analisis Data, 2017)

8. *Sequence Diagram* Lihat Laporan BB Masuk

Pada *Sequence Diagram* Lihat Laporan BB Masuk dijelaskan bahwa karyawan ingin mencari dan melihat laporan bahan baku masuk. Pertama karyawan harus *login*, masuk ke *main menu* pilih *report transaction* data

pilih BB Masuk, akan muncul kolom pencarian. Karyawan dapat memasukkan tanggal pada kolom pencarian periode data bahan baku masuk, setelah itu akan muncul data-data bahan baku yang sedang dicari.

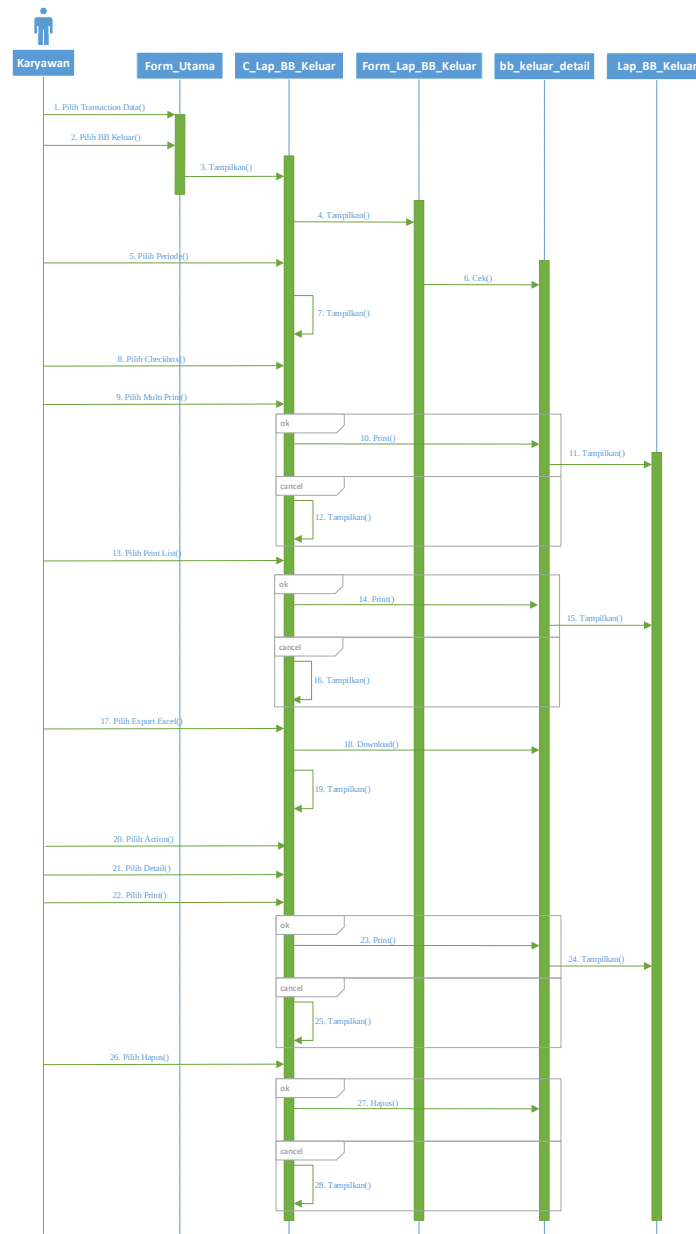


Gambar V.18 *Sequence Diagram* Lihat Laporan BB Masuk
(Sumber: Analisis Data, 2017)

9. *Sequence Diagram* Lihat Laporan BB Keluar

Pada *Sequence Diagram* Lihat Laporan BB Keluar dijelaskan bahwa karyawan ingin mencari dan melihat laporan bahan baku keluar. Pertama

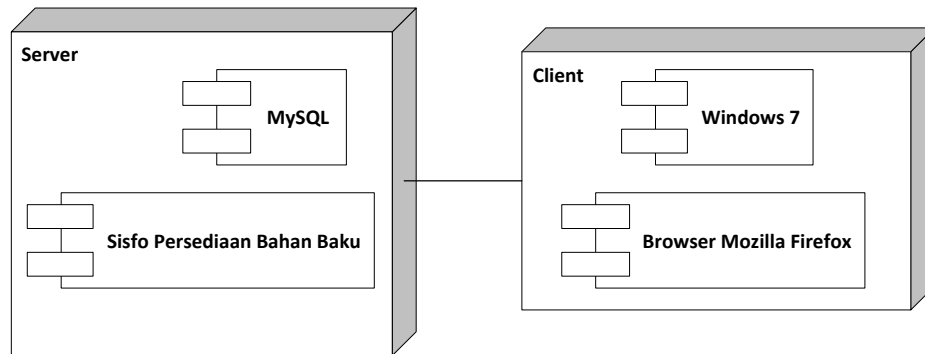
karyawan harus *login*, masuk ke *main menu* pilih *report transaction* data pilih data BB Keluar, akan muncul kolom pencarian. Karyawan dapat memasukkan tanggal pada kolom pencarian periode data bahan baku keluar, setelah itu akan muncul data-data bahan baku yang sedang dicari.



Gambar V.19 Sequence Diagram Lihat Laporan BB Keluar
(Sumber: Analisis Data, 2017)

5.2.6 Deployment Diagram yang Diusulkan

Deployment diagram digunakan pada divisi-divisi awal proses perancangan sistem untuk mendokumentasikan arsitektur fisik sebuah sistem. *Deployment diagram* sistem informasi persediaan bahan baku yang diusulkan dapat dilihat pada Gambar V.20.



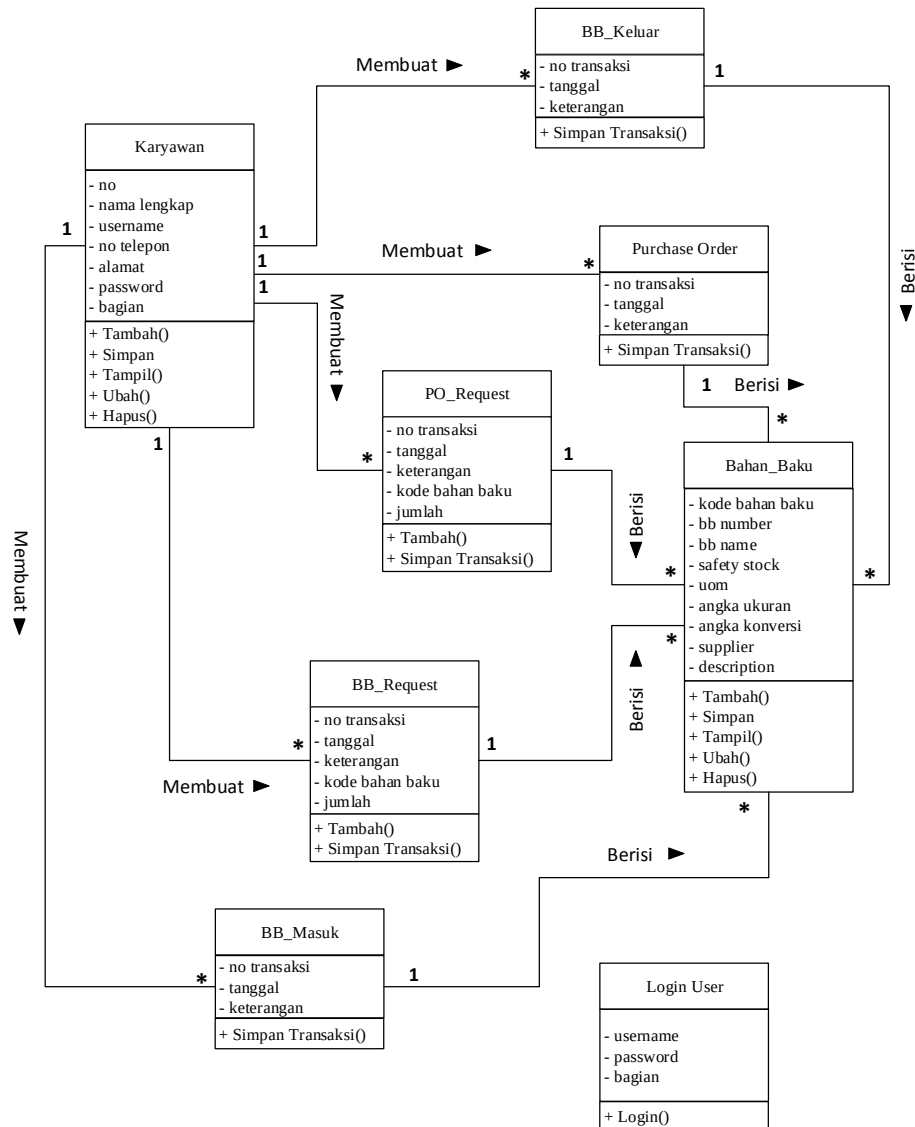
Gambar V.20 *Deployment Diagram* Sistem Informasi Persediaan Bahan Baku
(Sumber: Analisis Data, 2017)

Deployment Diagram Sistem Informasi Persediaan Bahan Baku sebagai *client* menggunakan PC dengan sistem operasi Windows 7 dan menggunakan Browser Mozilla Firefox, dan sebagai *server* yaitu *database* MySQL untuk menyimpan data dan menggunakan aplikasi berbasis web sistem informasi persediaan bahan baku.

5.2.7 *Class Diagram* yang Diusulkan

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

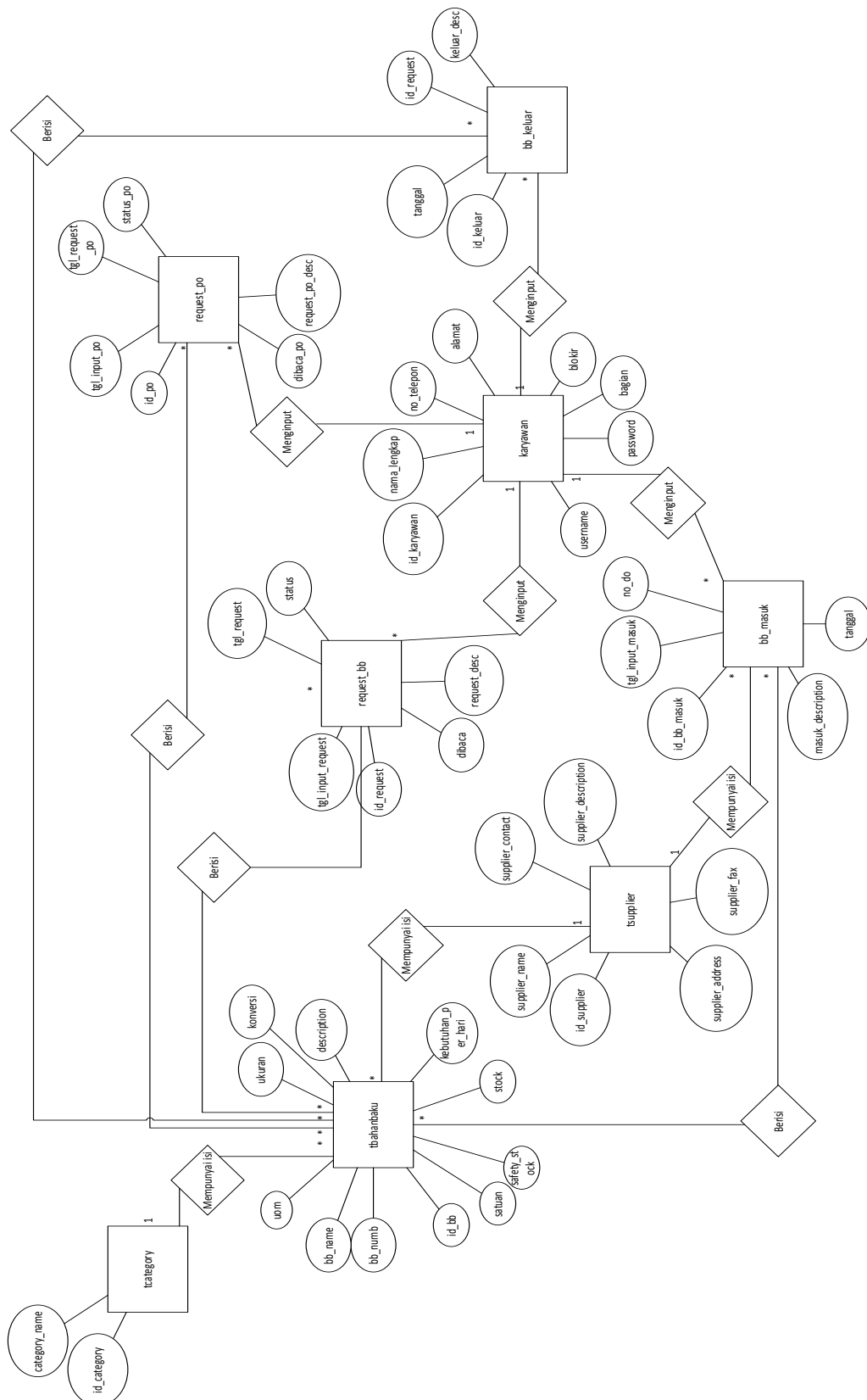
Selama proses analisis, *class diagram* memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama tahap desain, *class diagram* berperan dalam merangkap struktur dari semua kelas yang membentuk arsitektur yang dibuat. *Class diagram* sistem informasi persediaan bahan baku yang diusulkan dapat dilihat pada Gambar V.21.



Gambar V.21 *Class Diagram* Sistem Informasi Persediaan Bahan Baku
(Sumber: Analisis Data, 2017)

5.2.8 *Entity Relationship Diagram* yang Diusulkan

Entity Relationship Diagram (ERD) merupakan gambaran keterkaitan antar tabel beserta dengan *field-field* di dalamnya pada suatu *database* sistem. Setiap tabel pada umumnya memiliki keterkaitan hubungan. Keterkaitan antar tabel ini biasa disebut dengan relasi. *Entity Relationship Diagram* (ERD) usulan sistem persediaan bahan baku pada PT Rekadaya Multi Adiprima dapat dilihat pada Gambar V.22.



Gambar V.22 Entity Relationship Diagram Sistem Persediaan Bahan Baku Usulan
(Sumber: Analisis Data, 2017)

5.2.9 Kamus Data yang Diusulkan

Kamus data membantu dalam pendefinisian data agar pendefinisian data tersebut dapat dilakukan dengan lengkap dan terstruktur. Pembentukan kamus data dilaksanakan dalam tahap analisis dan perancangan suatu sistem. Berikut adalah kamus data yang menjelaskan isi entitas dari sistem:

1. Tabel *Master Karyawan*

Nama tabel : Karyawan

Fungsi : Menyimpan data karyawan

Tipe : *File Master*

Tabel V.13 Tabel *Master Karyawan*

No	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	Kode Karyawan	id_karyawan	Varchar	5	<i>Primary Key</i>
2	Nama Karyawan	nama_lengkap	Varchar	50	
3	No. Telepon Karyawan	no_telepon	Varchar	13	
4	Alamat Karyawan	Alamat	Varchar	100	
5	<i>Username</i>	<i>Username</i>	Varchar	20	
6	<i>Password</i>	<i>Password</i>	Text		
7	Bagian	Bagian	Varchar	20	
8	Blokir	Blokir	Enum('Y', 'N')		

(Sumber: Analisis Data, 2017)

2. Tabel *Master Supplier*

Nama tabel : *tsupplier*

Fungsi : Menyimpan data *supplier*

Tipe : *File Master*

Tabel V.14 Tabel *Master Supplier*

No	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	Kode <i>Supplier</i>	id_supplier	Varchar	5	<i>Primary Key</i>
2	Nama <i>Supplier</i>	supplier_name	Varchar	50	

No	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
3	No Telepon <i>Supplier</i>	<i>supplier_contact</i>	Varchar	13	
4	No Fax <i>Supplier</i>	<i>supplier_fax</i>	Varchar	13	
5	Alamat <i>Supplier</i>	<i>supplier_address</i>	Text		
6	Keterangan <i>Supplier</i>	<i>supplier_description</i>	Text		

(Sumber: Analisis Data, 2017)

3. Tabel *Master* Kategori

Nama tabel : *tcategory*

Fungsi : Menyimpan data kategori bahan baku

Tipe : *File Master*

Tabel V.15 Tabel *Master* Kategori

No	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	Kode Kategori	<i>id_category</i>	Varchar	5	<i>Primary Key</i>
2	Nama Kategori	<i>category_name</i>	Varchar	50	

(Sumber: Analisis Data, 2017)

4. Tabel *Master* Bahan Baku

Nama tabel : *tbahanbaku*

Fungsi : Menyimpan data bahan baku

Tipe : *File Master*

Tabel V.16 Tabel *Master* Bahan Baku

No	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	Kode Bahan Baku	<i>id_bb</i>	Varchar	10	<i>Primary Key</i>
2	Nomor Bahan Baku	<i>bb_numb</i>	Varchar	50	
3	Nama Bahan Baku	<i>bb_name</i>	Varchar	50	
4	UOM	<i>Uom</i>	Varchar	10	
5	Ukuran	<i>Ukuran</i>	Int	11	
6	Konversi	<i>Konversi</i>	Varchar	11	
7	Satuan	<i>Satuan</i>	Varchar	2	
8	Minimum Stok	<i>safety_stock</i>	Double		
9	Stok	<i>Stock</i>	Double		

No	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
10	Keterangan	<i>description</i>	Varchar	100	
11	Kode Kategori	<i>id_category</i>	Varchar	5	<i>Foreign Key</i>
12	Kode <i>Supplier</i>	<i>id_Supplier</i>	Varchar	5	<i>Foreign Key</i>
13	Kebutuhan Perhari	kebutuhan_per_hari	Int	8	

(Sumber: Analisis Data, 2017)

5. Tabel Transaksi Bahan Baku Masuk

Nama tabel : bb_masuk

Fungsi : Menyimpan data bahan baku masuk

Tipe : *File Transaksi*

Tabel V.17 Tabel Transaksi Bahan Baku Masuk

No	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	Kode Bahan Baku Masuk	id_bb_masuk	Varchar	10	<i>Primary Key</i>
2	Tanggal Input Bahan Baku Masuk	tgl_input_masuk	Datetime		
3	Nomor DO	no_do	Varchar	20	
4	Tanggal	Tanggal	Date		
5	Kode <i>Supplier</i>	<i>id_supplier</i>	Varchar	5	<i>Foreign Key</i>
6	Keterangan Bahan Baku Masuk	masuk_description	Varchar	200	
7	Kode Karyawan	id_karyawan	Varchar	5	<i>Foreign Key</i>

(Sumber: Analisis Data, 2017)

6. Tabel Transaksi Bahan Baku Masuk Detail

Nama tabel : bb_masuk_detail

Fungsi : Melihat data bahan baku masuk secara detail

Tipe : *File Transaksi*

Tabel V.18 Tabel Transaksi Bahan Baku Masuk Detail

No	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	Kode Bahan Baku Masuk	id_bb_masuk	Varchar	10	<i>Foreign Key</i>
2	Kode Bahan Baku	id_bb	Varchar	10	<i>Foreign Key</i>
3	Jumlah Bahan Baku Masuk	Jumlah	Int	11	

(Sumber: Analisis Data, 2017)

7. Tabel Transaksi Bahan Baku Keluar

Nama tabel : bb_keluar

Fungsi : Menyimpan data bahan baku keluar

Tipe : *File Transaksi***Tabel V.19 Tabel Transaksi Bahan Baku Keluar**

No	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	Kode Bahan Baku Keluar	id_keluar	Varchar	10	<i>Primary Key</i>
2	Tanggal	Tanggal	Date		
3	Kode <i>Request</i>	id_request	Varchar	10	<i>Foreign Key</i>
4	Keterangan Bahan Baku Keluar	keluar_desc	Varchar	100	
5	Kode Karyawan	id_karyawan	Varchar	5	<i>Foreign Key</i>

(Sumber : Analisis Data, 2017)

8. Tabel Bahan Baku Keluar Detail

Nama tabel : bb_keluar_detail

Fungsi : Melihat data bahan baku keluar secara detail

Tipe : *File Transaksi***Tabel V.20 Tabel Transaksi Bahan Baku Keluar Detail**

No	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	Kode Bahan Baku Keluar	id_keluar	Varchar	10	<i>Foreign Key</i>
2	Kode Bahan Baku	id_bb	Varchar	10	<i>Foreign Key</i>
3	Jumlah Bahan Baku	Jumlah	Double		

	Keluar				
--	--------	--	--	--	--

(Sumber : Analisis Data, 2017)

9. Tabel Transaksi Permintaan Bahan Baku

Nama tabel : *request_bb*

Fungsi : Menyimpan data permintaan bahan baku

Tipe : *File Transaksi*

Tabel V.21 Tabel Transaksi Permintaan Bahan Baku

No	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	Kode Request Bahan Baku	<i>id_request</i>	Varchar	10	<i>Primary Key</i>
2	Tanggal Input Request	<i>tgl_input_request</i>	Datetime		
3	Tanggal Request	<i>tgl_request</i>	Date		
4	Status	Status	Varchar	20	
5	Dibaca	Dibaca	Varchar	1	
6	Keterangan Request	<i>request_desc</i>	Varchar	100	
7	Kode Karyawan	<i>id_karyawan</i>	Varchar	10	<i>Foreign Key</i>

(Sumber: Analisis Data, 2017)

10. Tabel Transaksi Permintaan Bahan Baku Detail

Nama tabel : *request_bb_detail*

Fungsi : Melihat data permintaan bahan baku secara detail

Tipe : *File Transaksi*

Tabel V.22 Tabel Transaksi Permintaan Bahan Baku Detail

No	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	Kode Request Bahan Baku	<i>id_request</i>	Varchar	10	<i>Foreign Key</i>
2	Kode Bahan Baku	<i>Id_bb</i>	Varchar	10	<i>Foreign Key</i>
3	Jumlah Bahan Baku	Jumlah	Varchar	11	

(Sumber: Analisis Data, 2017)

11. Tabel Transaksi Permintaan PO

Nama tabel : *request_po*

Fungsi : Menyimpan data permintaan PO

Tipe : *File Transaksi*

Tabel V.23 Tabel Transaksi Permintaan PO

No	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	Kode <i>Request</i> PO	id_po	Varchar	10	<i>Primary Key</i>
2	Tanggal <i>Input</i> PO	tgl_input_po	Datetime		
3	Jumlah PO	tgl_request_po	Date		
4	Status PO	status_po	Varchar	20	
5	Dibaca	dibaca_po	Varchar	1	
6	Keterangan <i>Request</i> PO	request_po_desc	Varchar	100	
7	Kode Karyawan	id_karyawan	Varchar	10	<i>Foreign Key</i>

(Sumber: Analisis Data, 2017)

12. Tabel Transaksi Permintaan PO Detail

Nama tabel : *request_po_detail*

Fungsi : Melihat data permintaan PO

Tipe : *File Transaksi*

Tabel V.24 Tabel Transaksi Permintaan PO Detail

No	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	Kode <i>Request</i> PO	id_po	Varchar	10	<i>Foreign Key</i>
2	Kode Bahan Baku	id_bb	Varchar	10	<i>Foreign Key</i>
3	Jumlah PO	Jumlah_po	Varchar	11	

(Sumber: Analisis Data, 2017)

13. Tabel PO

Nama tabel : po
Fungsi : Melihat data PO
Tipe : *File Transaksi*

Tabel V.25 Tabel PO

No	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	Kode PO	kd_po	Varchar	10	<i>Primary Key</i>
2	Tanggal	Tanggal	Date		
3	ID Request	Jumlah_po	Varchar	10	
4	Keterangan Keluar	keluar_desc	Varchar	100	
5	Status	Status	Varchar	20	
6	ID Karyawan	id_karyawan	Varchar	5	

(Sumber: Analisis Data, 2017)

14. Tabel PO Detail

Nama tabel : po_detail
Fungsi : Melihat data PO
Tipe : *File Transaksi*

Tabel V.26 Tabel PO Detail

No	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	Kode PO	id_po	Varchar	10	
2	Kode Bahan Baku	id_bb	Varchar	10	
3	Jumlah PO	Jumlah	Double		

(Sumber: Analisis Data, 2017)

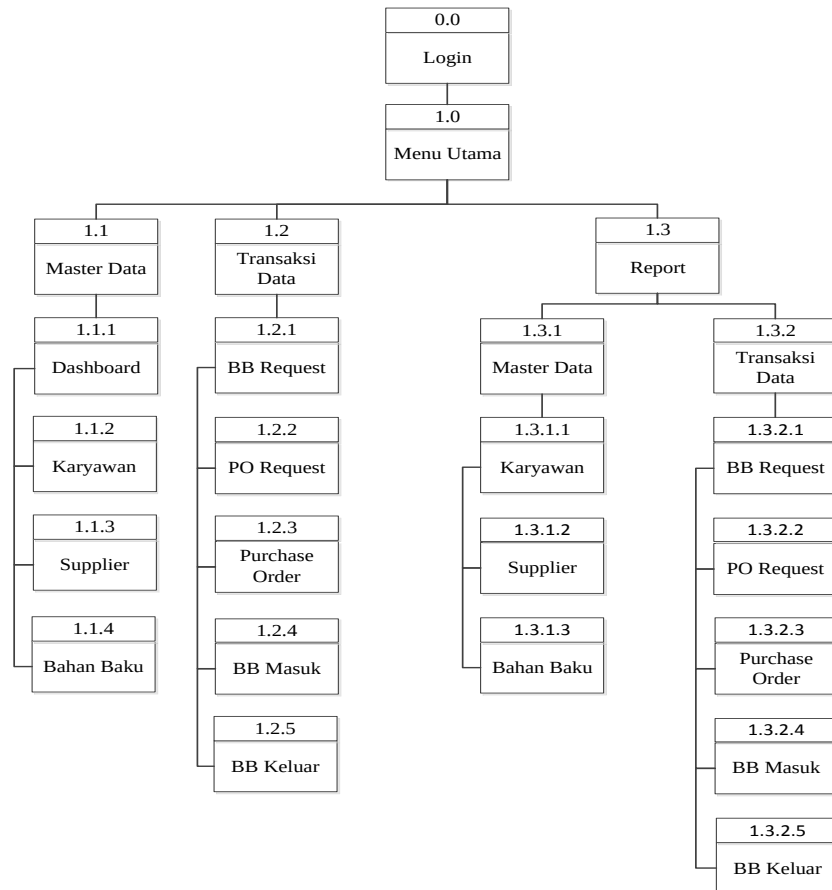
5.3 Analisis Desain Program

Tahap ini merupakan tahap kedua dalam metodologi *prototype evolutioner*, yaitu tahap membuat sebuah *prototype* dari program/aplikasi. Dimulai dengan analisis desain usulan yang meliputi pembuatan struktur menu

program, *interface* program dan *flowchart* program sampai program dapat dijalankan.

5.3.1 Struktur Menu Program

Berikut adalah struktur menu program persediaan bahan baku PT Rekadaya Multi Adiprima, yang dapat dilihat pada Gambar V.23.



Gambar V.23 Struktur Menu Program
(Sumber: Analisis Data, 2017)

Pada struktur menu program yang diawali dengan *login*, kemudian masuk ke menu utama dimana terdapat empat menu. Pertama menu master yang didalamnya terdapat menu data karyawan, *supplier* dan bahan baku. Kedua menu transaksi yang didalamnya terdapat menu *BB Request*, *PO Request*, *BB Masuk*, *BB Keluar*. Ketiga menu laporan yang didalamnya terdapat laporan *BB Request*, laporan *PO Request*, laporan *BB Masuk* dan laporan *BB Keluar*. Lalu yang

keempat menu *dashboard* yang menampilkan menu utama dan terdapat menu ubah password dan *logout*.

5.3.2 Perancangan *Interface* Program

Rancangan *interface* dari program persediaan bahan baku ini adalah sebagai berikut:

1. *Form Login*

Form login adalah *form* yang digunakan untuk masuk ke dalam program aplikasi. Untuk masuk ke dalam aplikasi, pengguna harus memasukkan nama pengguna dan sandi yang benar. Jika sukses *login* maka akan masuk ke menu utama, tetapi jika tidak sukses *login* maka akan ada *alert username* atau *password* salah. Rancangan *form login* dapat dilihat pada Gambar V.24.



Aplikasi Persediaan Bahan Baku

Username

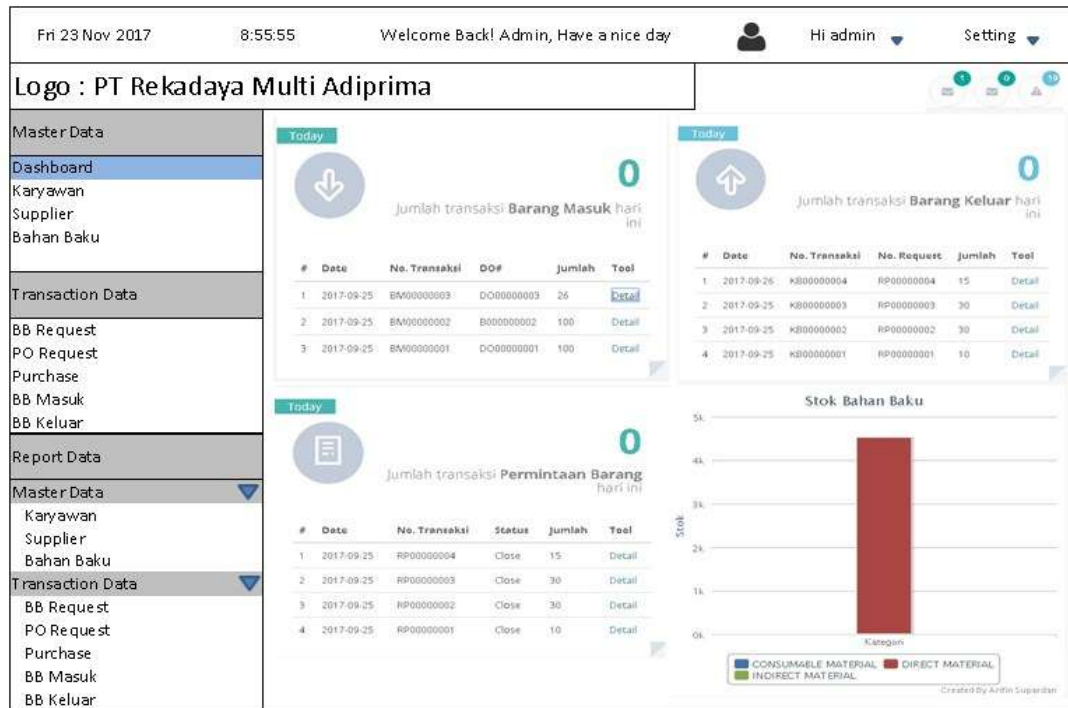
Password

Login

Gambar 5.24 Rancangan *Form Login*
(Sumber: Analisis Data, 2017)

2. Form Menu Utama

Form menu utama adalah tampilan awal pada aplikasi yang berisi empat menu yang masing-masing terdiri dari submenu. Tiga menu tersebut adalah *Master Data*, *Transaksi Data* dan *Report Data*. Rancangan menu utama dapat dilihat pada Gambar V.25.



Gambar V.25 Rancangan Form Menu Utama
(Sumber: Analisis Data, 2017)

3. Form Tambah Bahan Baku

Form Tambah Bahan Baku adalah form yang digunakan untuk menyimpan data-data persediaan bahan baku. Data persediaan bahan baku dibuat ketika ada bahan baku yang belum di-input. Form ini dapat dilihat pada Gambar V.25. Berikut penjelasan penggunaan pada form ini:

- Textfield* BB ID :Diberikan otomatis oleh system
- Textfield* BB Number :Untuk memasukkan nomor bahan baku
- Textfield* BB Name :Untuk memasukkan nama bahan baku

- | | |
|-----------------------------------|--|
| d. <i>Texfield Safety Stock</i> | :Untuk memasukkan <i>safety stock</i> |
| e. <i>Texfield UOM</i> | :Untuk memasukkan UOM |
| f. <i>Texfield Angka Ukuran</i> | :Untuk memasukkan angka dan ukuran |
| g. <i>Texfield Angka Konversi</i> | :Untuk memasukkan angka konversi |
| h. <i>Texfield Category</i> | :Untuk memasukkan kategori bahan baku |
| i. <i>Texfield Supplier</i> | :Untuk memasukkan nama <i>supplier</i> |
| j. <i>Texfield Description</i> | :Untuk memasukkan keterangan bahan baku. |
| k. Tombol Simpan | :Untuk menyimpan data |

Fri 23 Nov 2017 8:55:55 Welcome Back! Admin, Have a nice day Hi gudang

Logo : PT Rekadaya Multi Adiprima

Master Data Dashboard Profil Supplier Bahan Baku	Tambah Bahan Baku BB ID BB Number BB Name Safety Stock Uom --Pilih-- Angka Ukuran --Ukuran-- Angka Konversi --Category-- Supplier --Pilih-- Description Simpan
--	--

Gambar V.26 Rancangan *Form* Tambah Bahan Baku
(Sumber: Analisis Data, 2017)

4. Form Transaksi Penerimaan Bahan Baku

Form transaksi penerimaan bahan baku masuk (BB Masuk) digunakan untuk menyimpan data-data bahan baku yang masuk. Berikut penjelasan penggunaan pada form ini:

- a. *Textfield* No Transaksi :Diberikan otomatis oleh system
- b. *Datetime* Tanggal Masuk :Untuk memasukkan tanggal masuk bahan baku
- c. *Textfield* DO# :Untuk memasukkan nomor DO
- d. *Textfield* Supplier :Untuk memasukkan nama *supplier*
- e. *Textfield* Keterangan :Untuk memasukkan keterangan bahan baku masuk.
- f. *Textfield* Kode Bahan Baku :Untuk memasukkan kode bahan baku
- g. *Textfield* Jumlah :Untuk memasukkan jumlah bahan baku masuk.
- h. Tombol Tambah :Untuk menambah data bahan baku yang akan dikeluarkan.
- i. Tombol Simpan Transaksi :Untuk menyimpan data transaksi

The screenshot displays a web application interface for PT Rekadaya Multi Adiprima. The main section is titled 'Transaksi Penerimaan Bahan Baku (BB Masuk)'. It contains several input fields: 'No. Transaksi', 'Tanggal Masuk', 'DO#', 'Supplier' (a dropdown menu), 'Keterangan', 'Kode Bahan Baku', and 'Jumlah'. At the bottom right of this section are two buttons: 'Tambah' (blue) and 'Simpan Transaksi' (red). Below the form is a table titled 'DAFTAR BAHAN BAKU' with the following columns: 'No', 'No. BB', 'Nama Bahan Baku', 'UOM', and 'Jumlah'. The table is currently empty, with a 'GRAND TOTAL' row showing a value of '0' in the 'Jumlah' column. The left sidebar contains navigation links for 'Mater Data' (Dashboard, Profil, Supplier, Bahan Baku), 'Transaction Data' (BB Masuk, BB Keluar, PO Request), and 'Report Data' (Mater Data, Supplier, Bahan Baku, Transaction Data, BB Masuk, BB Keluar).

Gambar V.27 Rancangan Form Transaksi Penerimaan Bahan Baku
(Sumber: Analisis Data, 2017)

5. Form Transaksi Pengeluaran Bahan Baku

Form transaksi pengeluaran bahan baku (BB Keluar) digunakan untuk menyimpan data-data bahan baku yang keluar untuk diproduksi. Berikut penjelasan penggunaan pada form ini:

- Textfield* No Transaksi :Diberikan otomatis oleh system
- Textfield* Tanggal Keluar :Untuk memasukkan tanggal keluar bahan baku.
- Textfield Please Select* :Untuk memasukkan nomor bb request.
- Textfield* Keterangan :Untuk memasukkan keterangan keluar bahan baku.
- Tombol Simpan Transaksi :Untuk menyimpan data transaksi.

Logo : PT Rekadaya Multi Adiprima

Master Data

Dashboard

Profil

Supplier

Bahan Baku

Transaction Data

BB Masuk

BB Keluar

PO Request

Report Data

Master Data

Supplier

Bahan Baku

Transaction Data

BB Masuk

BB Keluar

Transaksi Pengeluaran Bahan Baku (BB Keluar)

No. Transaksi: BK00000005

Tanggal: 29-09-2017

Please Select

Keterangan

DAFTAR BAHAN BAKU

No	No. BB	Nama Bahan Baku	UOM	Size	Jumlah
TOTAL:					0

SIMPAN TRANSAKSI

Gambar V.28 Rancangan Form Transaksi Pengeluaran Bahan Baku
(Sumber: Analisis Data, 2017)

6. Form Transaksi Permintaan PO

Form Transaksi Permintaan PO (PO Request) digunakan untuk menyimpan data-data permintaan bahan baku yang akan dibeli oleh Divisi *Purchasing*. Berikut penjelasan penggunaan pada *form* ini:

- Textfield* No Transaksi :Diberikan otomatis oleh *system*
- Textfield* Tanggal :Untuk memasukkan tanggal input PO Request.
- Textfield* Keterangan :Untuk memasukkan keterangan PO Request.
- Textfield* Kode Bahan Baku :Untuk memasukkan kode bahan baku.
- Textfield* Jumlah :Untuk memasukkan jumlah bahan baku yang di request.
- Tombol Tambah :Untuk menambah data bahan baku yang akan di request.
- Tombol Simpan Transaksi :Untuk menyimpan data transaksi.

PT Rekadaya Multi Adiprima

Master Data

Dashboard

Karyawan

Supplier

Bahan Baku

Transaction Data

BB Request

PO Request

Purchase

BB Masuk

BB Keluar

Report Data

Master Data

Karyawan

Supplier

Bahan Baku

Transaction Data

BB Request

PO Request

Purchase

BB Masuk

BB Keluar

Transaksi Permintaan PO (PO Request)

No. Transaksi: PO00000003

Kode Bahan Baku: [input field]

Tanggal: 29-09-2017

Jumlah: [input field]

Keterangan: [input field]

Tambah [Simpan Transaksi]

DAFTAR BAHAN BAKU

No	No. B B	Nama Bahan Baku	UOM	Size	Jumlah
GRAND TOTAL:					0

Gambar V.29 Rancangan *Form* Transaksi Permintaan PO

(Sumber: Analisis Data, 2017)

7. Form Transaksi Permintaan Bahan Baku

Form transaksi permintaan bahan baku (*BB Request*) digunakan untuk menyimpan data-data permintaan bahan baku ke Divisi Gudang untuk diproduksi. Berikut penjelasan penggunaan pada *form* ini:

- a. *Textfield* No Transaksi :Diberikan otomatis oleh *system*
- b. *Textfield* Tanggal :Untuk memasukkan tanggal input *BB Request*.
- c. *Textfield* Keterangan :Untuk memasukkan keterangan *BB Request*.
- d. *Textfield* Kode Bahan Baku :Untuk memasukkan kode bahan baku.
- e. *Textfield* Jumlah :Untuk memasukkan jumlah bahan baku yang di *request*.
- f. Tombol Tambah :Untuk menambah data bahan baku yang akan di *request*.
- g. Tombol Simpan Transaksi :Untuk menyimpan data transaksi.

Fri 23 Nov 2017 8:55:55 Welcome Back! Admin, Have a nice day Hi produksi

Logo : PT Rekadaya Multi Adiprima

Master Data	Transaksi Permintaan Bahan Baku (BB Request)													
Dashboard Profile	No. Transaksi RP00000003	Kode Bahan Baku												
Transaction Data	Tanggal 29-09-2017	Jumlah												
BB Request	Keterangan	Tambah Simpan Transaksi												
Report Data	DAFTAR BAHAN BAKU													
BB Request	<table border="1"><thead><tr><th>No</th><th>No. BB</th><th>Nama Bahan Baku</th><th>UOM</th><th>Size</th><th>Jumlah</th></tr></thead><tbody><tr><td colspan="5">GRAND TOTAL:</td><td>0</td></tr></tbody></table>		No	No. BB	Nama Bahan Baku	UOM	Size	Jumlah	GRAND TOTAL:					0
No	No. BB	Nama Bahan Baku	UOM	Size	Jumlah									
GRAND TOTAL:					0									

Gambar V.30 Rancangan *Form* Transaksi Permintaan Bahan Baku
(Sumber: Analisis Data, 2017)

8. Form Laporan Bahan Baku Masuk

Form laporan bahan baku masuk digunakan untuk melihat bahan baku apa saja yang sudah masuk dan berapa total bahan baku yang masuk. Berikut penjelasan penggunaan pada form ini:

- Date Tanggal* : Untuk memasukkan periode tanggal pencarian laporan bahan baku masuk
- Tombol Tampilkan* : Untuk menampilkan laporan data bahan baku masuk.
- Tombol Action* : Untuk melihat detail laporan data bahan baku masuk.
- Tombol Print List* : Untuk mencetak laporan data bahan baku masuk.
- Tombol Export Excel* : Untuk men-download laporan data bahan baku masuk dalam format Excel.

Fri 23 Nov 2017 8:55:55 Welcome Back! Admin, Have a nice day Hi gudang ▼

Logo : PT Rekadaya Multi Adiprima

Master Data

Dash hboard

Profile

Supplier

Bahan Baku

Transaction Data

BB Masuk

BB Keluar

PO Request

Report Data

Master Data ▼

Supplier

Bahan Baku

Transaction Data ▼

BB Masuk

BB Keluar

Laporan Bahan Baku Masuk

Periode

#	Tanggal	ID Bahan Baku Masuk	DO#	Jumlah	Supplier	Tools
						Action▼
						Action▼
						Action▼

Showing 1 to 3 of 3 entries

Gambar V.31 Rancangan Form Laporan Bahan Baku Masuk
(Sumber: Analisis Data, 2017)

9. Form Laporan Bahan Baku Keluar

Form laporan bahan baku keluar digunakan untuk melihat bahan baku apa saja yang sudah keluar untuk diproduksi dan berapa total bahan baku yang keluar. Berikut penjelasan penggunaan pada *form* ini:

- a. *Date Tanggal* :Untuk memasukkan periode tanggal pencarian laporan bahan baku keluar.
- b. Tombol *Tampilkan* :Untuk menampilkan laporan data bahan baku keluar.
- c. Tombol *Action* :Untuk melihat detail laporan data bahan baku keluar.
- f. Tombol *Multi Print* :Untuk mencetak laporan data bahan baku keluar secara *optional*.
- g. Tombol *Print List* :Untuk mencetak laporan data bahan baku masuk.
- h. Tombol *Export Excel* :Untuk men-*download* laporan data bahan baku masuk dalam format *Excel*.

The screenshot displays a web application interface for generating a 'Laporan Bahan Baku Keluar' (Raw Material Out Report). The top navigation bar shows the date 'Fri 23 Nov 2017', time '8:55:55', and a welcome message 'Welcome Back! Admin, Have a nice day'. The user 'Higudang' is logged in. The sidebar menu on the left includes 'Master Data', 'Dashboard', 'Profile', 'Supplier', 'Bahan Baku', 'Transaction Data', 'BB Masuk', 'BB Keluar', and 'PO Request'. The main content area features a table with columns: '#', 'Tanggal', 'ID Bahan Baku Keluar', 'Request Number', 'Jumlah', 'Out By', and 'Tools'. Above the table, there is a date range selector set to '1-09-2017' to '30-09-2017' and a red 'Tampilkan' button. Below the table, there are three buttons: 'Multi Print' (red), 'Print List' (yellow), and 'Export Excel' (blue). The bottom of the form indicates 'Showing 1 to 3 of 3 entries' and includes navigation buttons for '<- Previous', '1', and 'Next ->'.

Gambar V.32 Rancangan Laporan Bahan Baku Keluar
(Sumber: Analisis Data, 2017)

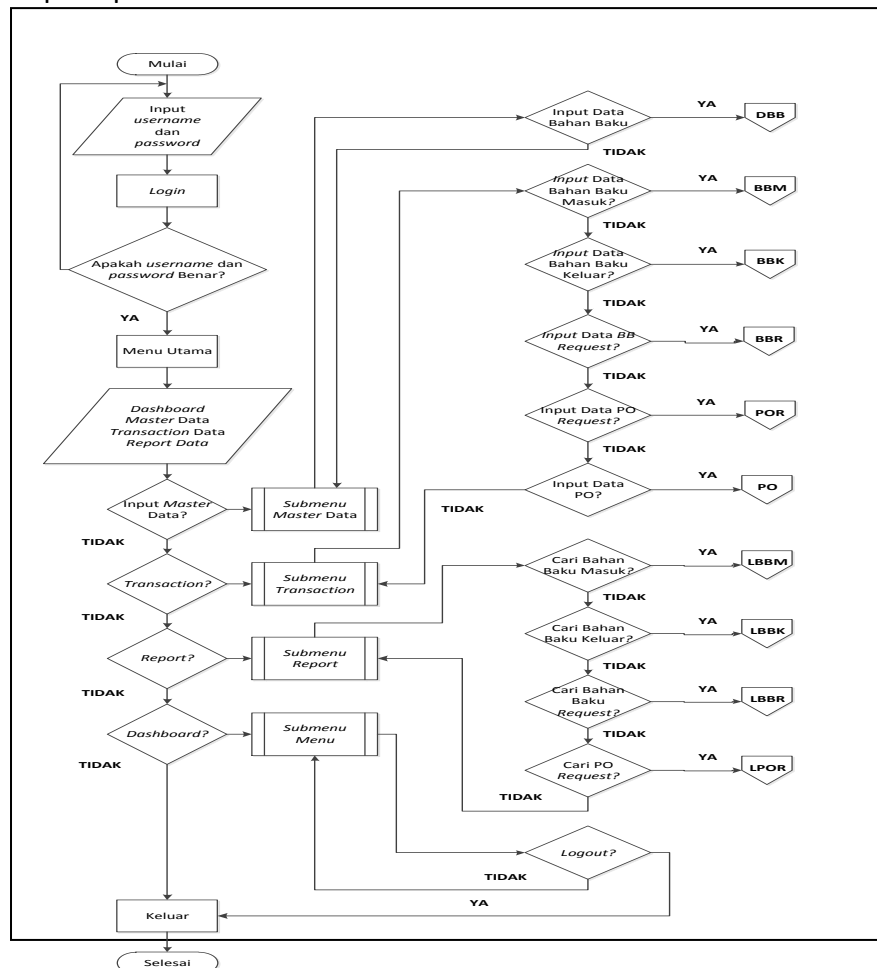
5.3.3 Flowchart Program

Pada sub bab ini dijelaskan bentuk *flowchart*/alir data dari program sistem informasi persediaan bahan baku sehingga *flowchart* ini dapat menggambarkan alur logika program tersebut dengan jelas, Divisi ini juga memperjelas urutan prosedur sistem dan spesifikasi proses.

Berikut adalah *flowchart-flowchart* yang ada pada program :

1. Flowchart Menu Utama

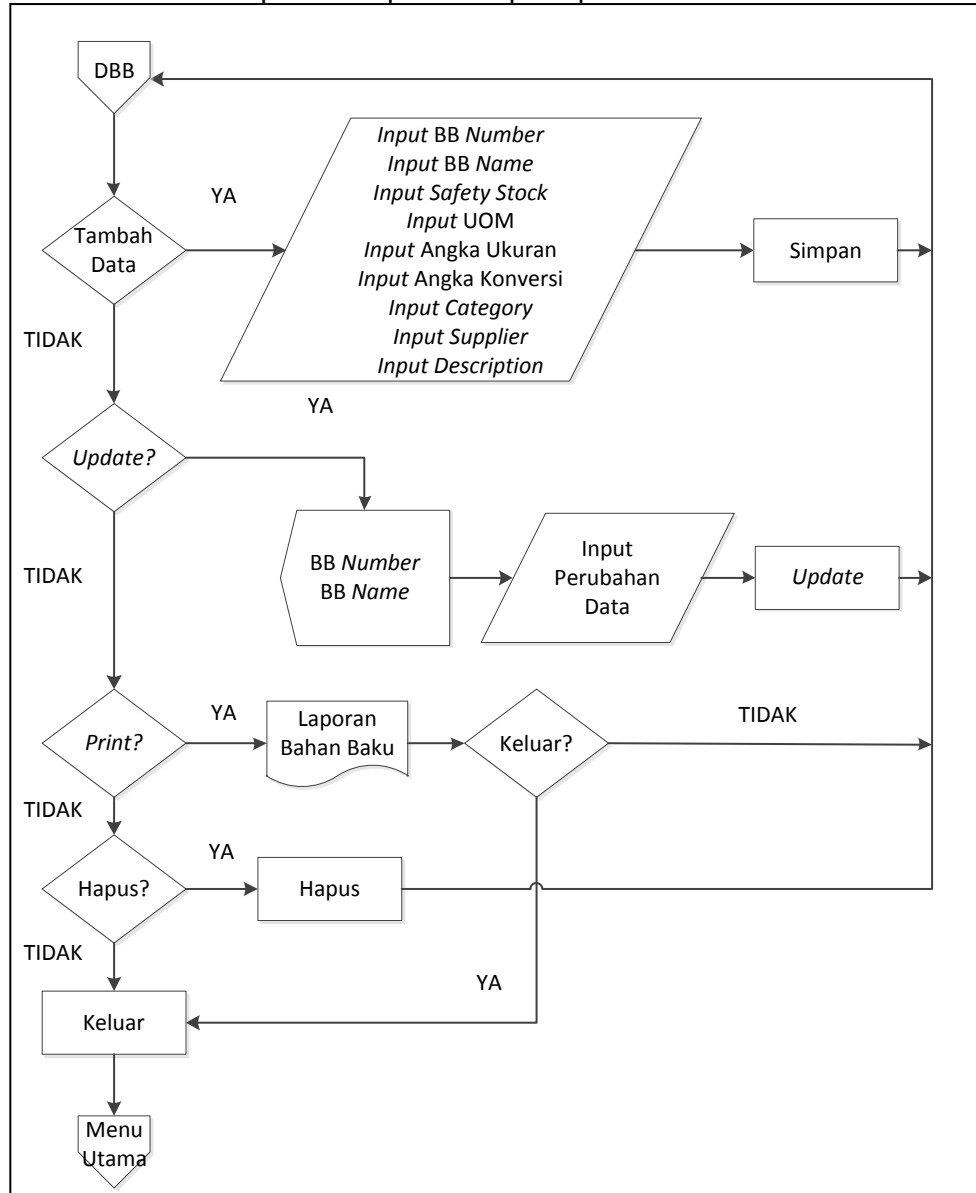
Flowchart Menu Utama menggambarkan alur logika yang sebenarnya. *Flowchart* ini juga memperjelas urutan prosedur sistem dan spesifikasi proses seperti pada Gambar V.33.



Gambar V.33 Flowchart Menu Utama
(Sumber: Analisis Data, 2017)

2. Flowchart Menu Bahan Baku

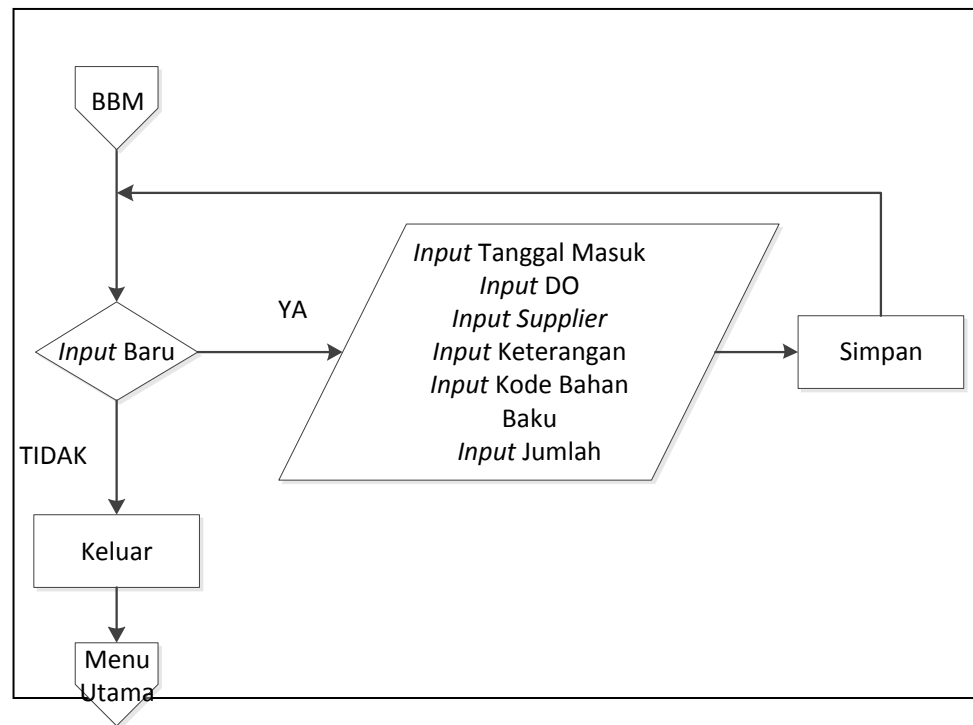
Flowchart Menu Bahan Baku menggambarkan alur logika penginputan data persediaan bahan baku. Flowchart ini juga memperjelas urutan prosedur sistem dan spesifikasi proses seperti pada Gambar V.34.



Gambar V.34 Flowchart Menu Bahan Baku
(Sumber: Analisis Data, 2017)

3. *Flowchart* BB Masuk

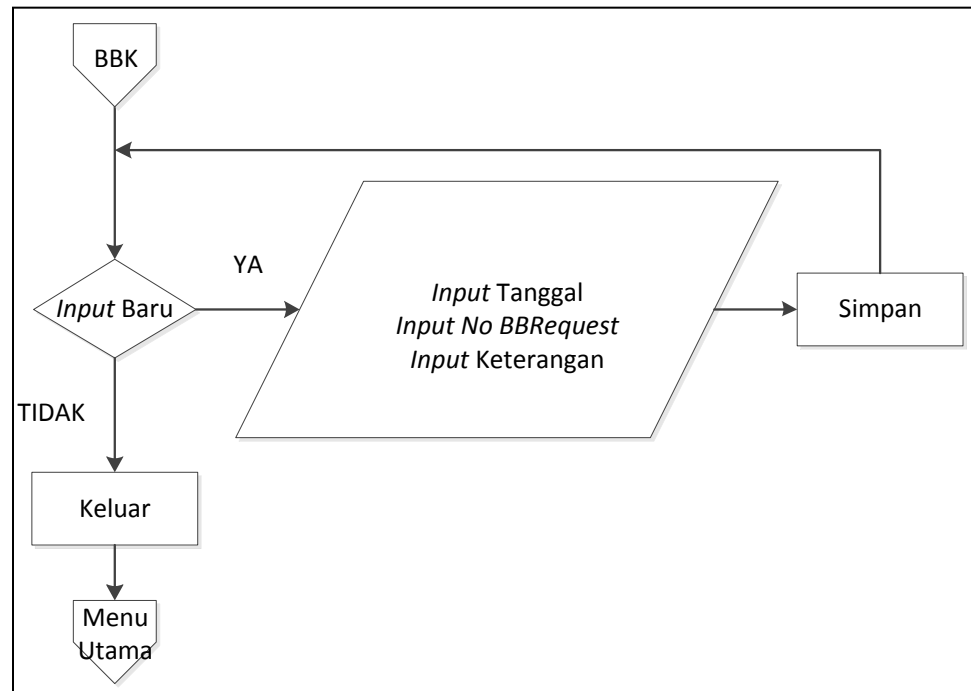
Flowchart BB Masuk menggambarkan alur logika penginputan data bahan baku masuk. *Flowchart* ini juga memperjelas urutan prosedur sistem dan spesifikasi proses seperti pada Gambar V.35.



Gambar V.35 *Flowchart* Menu BB Masuk
(Sumber: Analisis Data, 2017)

4. *Flowchart* BB Keluar

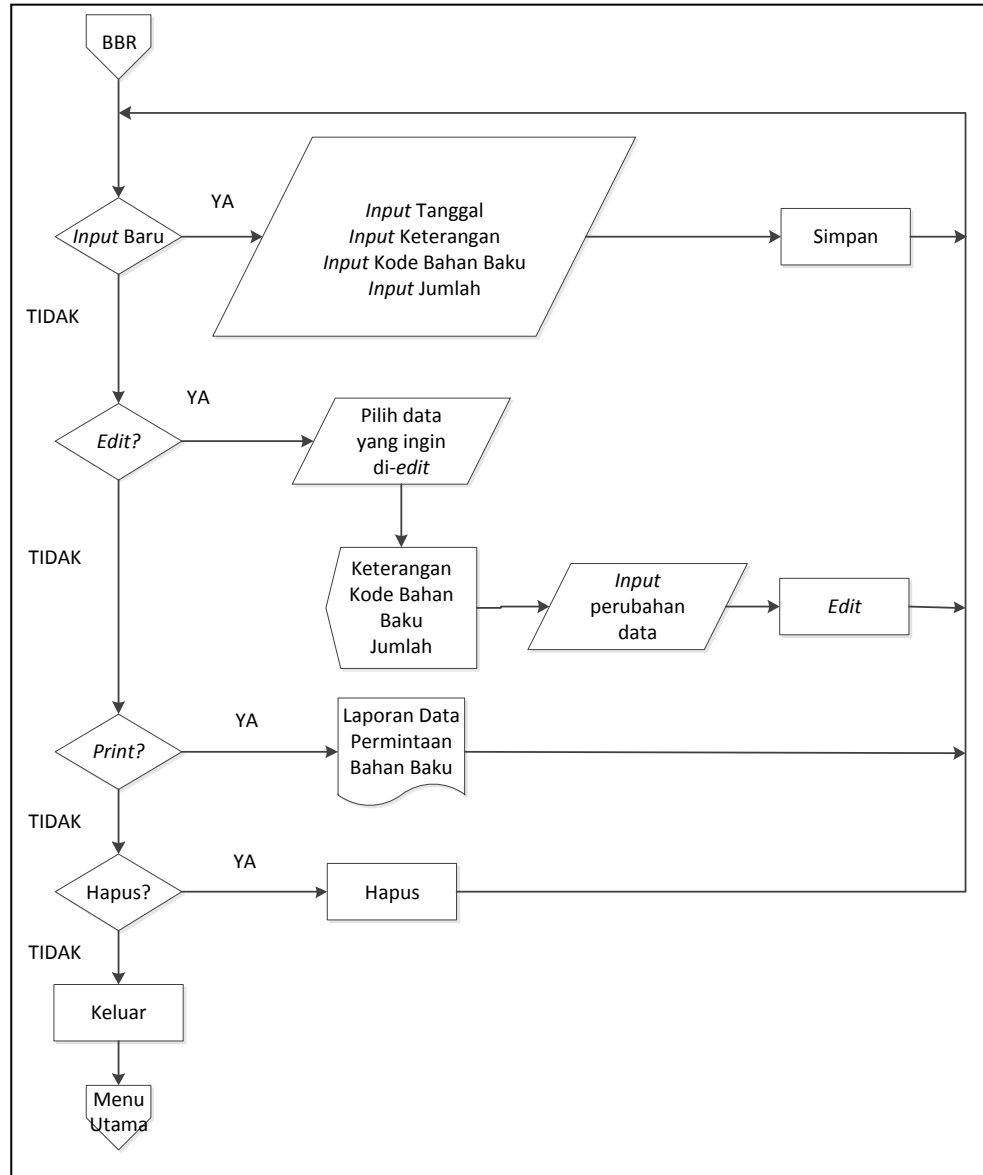
Flowchart BB Keluar menggambarkan alur logika penginputan data bahan baku keluar. *Flowchart* ini juga memperjelas urutan prosedur sistem dan spesifikasi proses seperti pada Gambar V.36.



Gambar V.36 *Flowchart* Menu BB Keluar
(Sumber: Analisis Data, 2017)

5. Flowchart BB Request

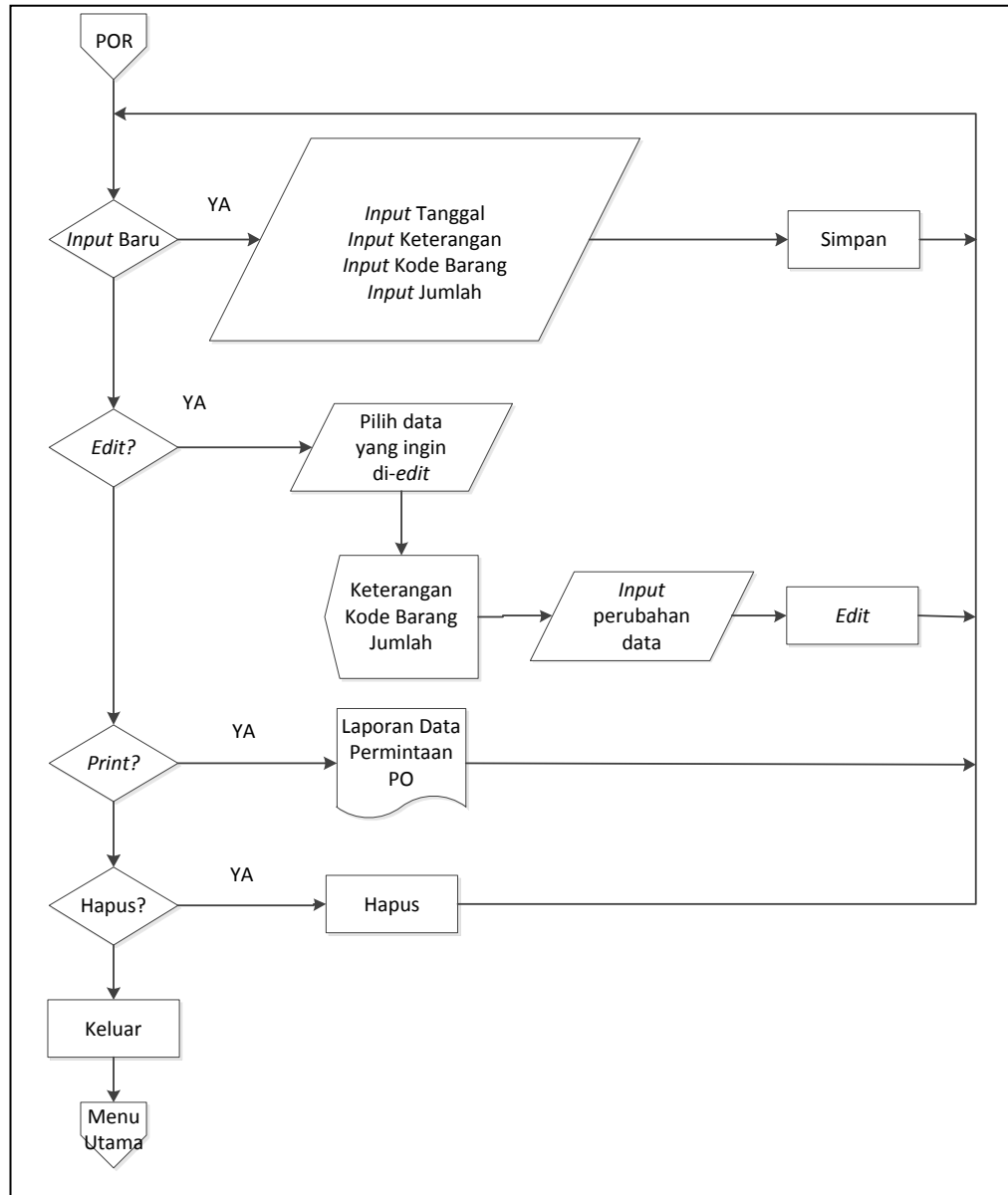
Flowchart BB Request menggambarkan alur logika penginputan data permintaan bahan baku produksi. *Flowchart* ini juga memperjelas urutan prosedur sistem dan spesifikasi proses seperti pada Gambar V.37.



Gambar V.37 *Flowchart BB Request*
(Sumber: Analisis Data, 2017)

6. Flowchart PO Request

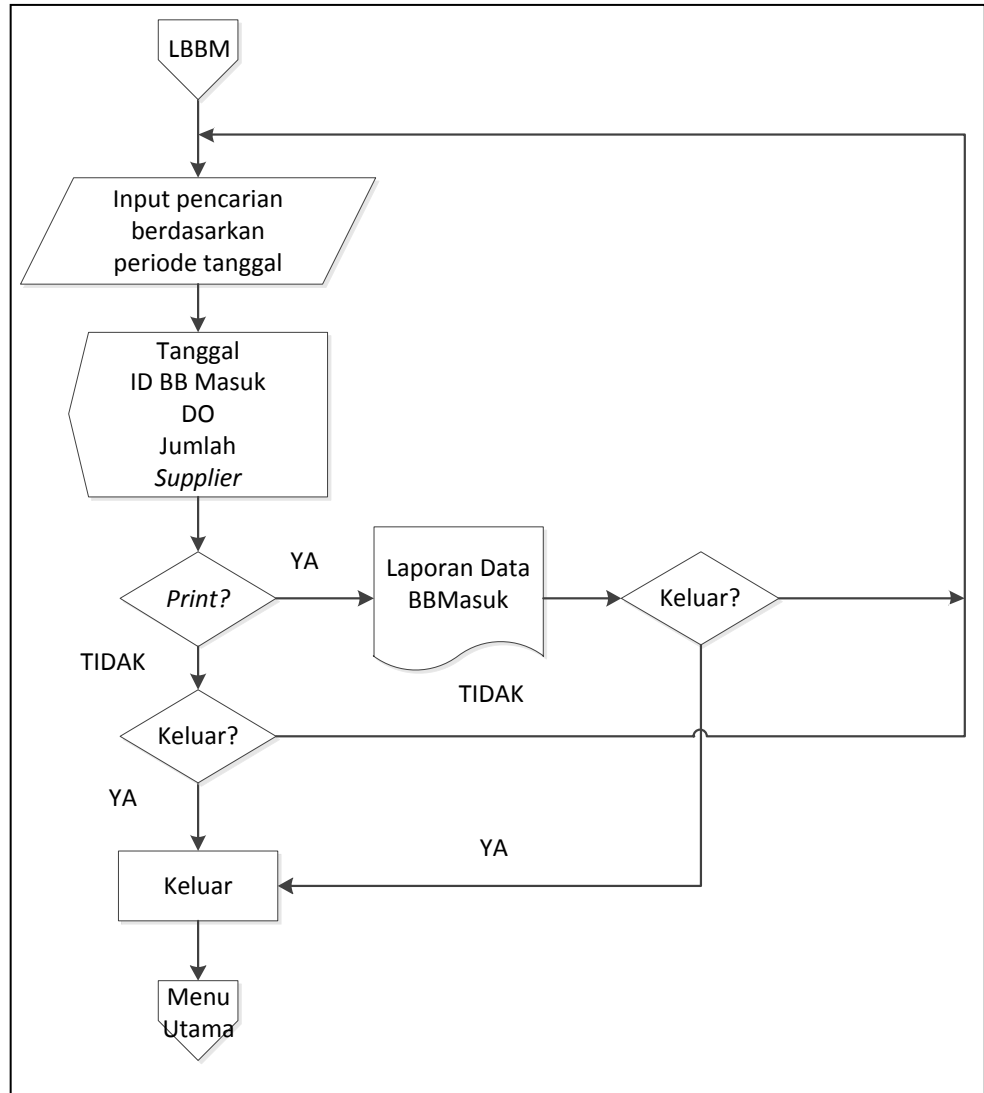
Flowchart PO Request menggambarkan alur logika penginputan data permintaan pembuatan PO. *Flowchart* ini juga memperjelas urutan prosedur sistem dan spesifikasi proses seperti pada Gambar V.38.



Gambar V.38 Flowchart Menu PO Request
(Sumber: Analisis Data, 2017)

7. Flowchart Laporan BB Masuk

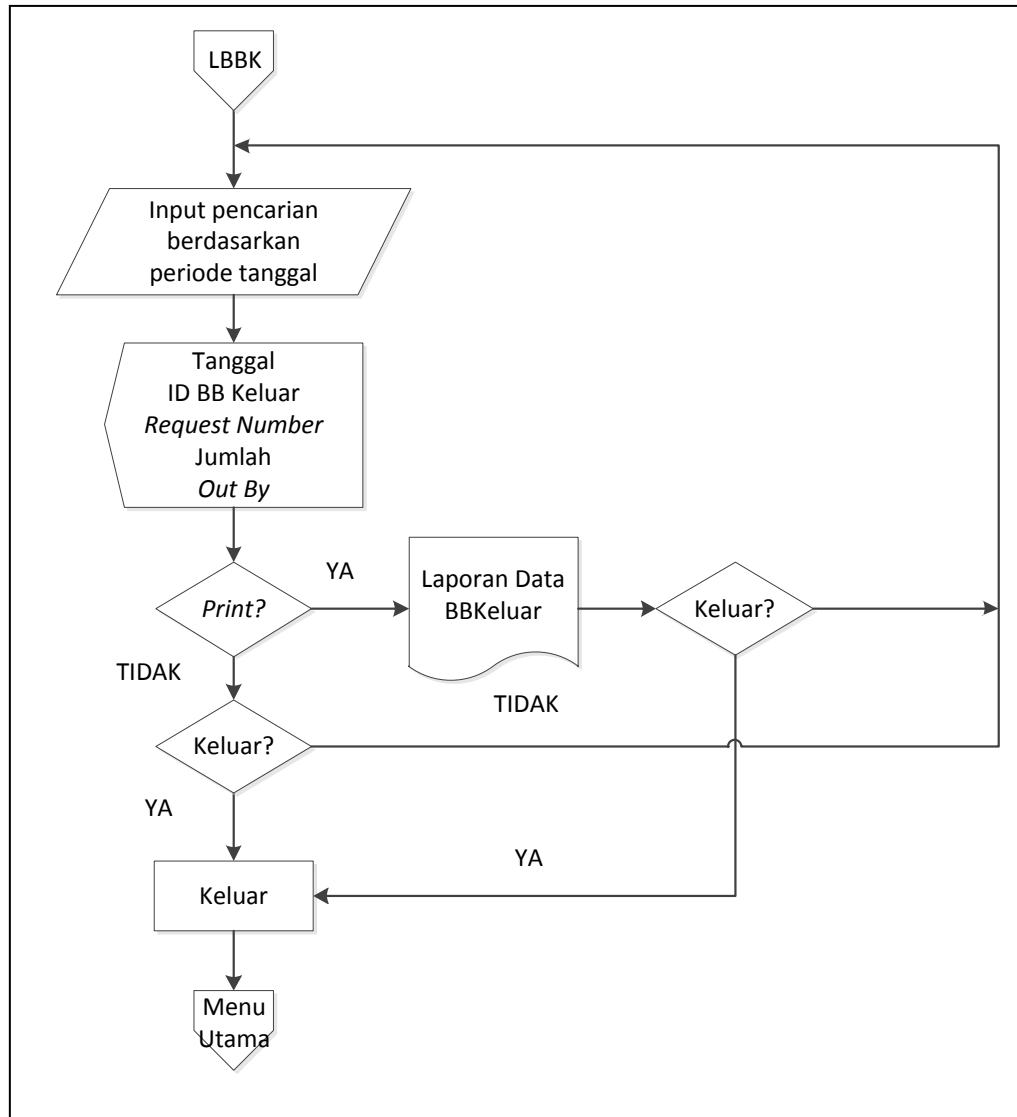
Flowchart Laporan BB Masuk menggambarkan alur logika untuk melihat laporan bahan baku masuk. *Flowchart* ini juga memperjelas urutan prosedur sistem dan spesifikasi proses seperti pada Gambar V.39.



Gambar V.39 *Flowchart* Menu Laporan BB Masuk
(Sumber: Analisis Data, 2017)

8. *Flowchart* Laporan BB Keluar

Flowchart Laporan BB Keluar menggambarkan alur logika untuk melihat laporan bahan baku keluar. *Flowchart* ini juga memperjelas urutan prosedur sistem dan spesifikasi proses seperti pada Gambar V.40.



Gambar V.40 *Flowchart* Menu Laporan BB Keluar
(Sumber: Analisis Data, 2017)

5.4 Analisis *Software* dan *Hardware*

1. Analisis kebutuhan *software*

Software yang dibutuhkan antara lain sebagai berikut:

- a. XAMPP 1.8.1 untuk pengelolaan *database*.
- b. *Browser* Mozilla Firefox atau Internet Explorer untuk menjalankan XAMPP.
- c. Sistem Operasi minimal Windows XP.

2. Analisis kebutuhan *hardware*

Untuk mendukung kebutuhan implementasi sistem persediaan bahan baku pada PT Rekadaya Multi Adiprima, diperlukan dukungan sistem komputer yang memadai dalam segi *hardware*, diantaranya sebagai berikut:

- a. *Processor* : minimal Pentium IV (1,8 GHz).
- b. RAM : minimal 512 MB.
- c. *Harddisk* : minimal 64 GB.
- d. Sistem Operasi : Microsoft Windows XP atau di atasnya.
- e. *Printer* untuk pencetakan dokumen.

BAB VI

KESIMPULAN DAN SARAN

1.10 Kesimpulan

Berdasarkan hasil rancang bangun sistem yang dilakukan mengenai sistem informasi persediaan bahan baku di PT Rekadaya Multi Adiprima dapat disimpulkan beberapa hal yaitu:

1. Aplikasi sistem informasi ini sudah berjalan secara komputerisasi sehingga pencatatan bahan baku yang masuk dan keluar gudang tidak akan terjadi lagi kerusakan, kehilangan dan ketidaksesuaian pada data bahan baku.
2. Dengan sistem ini, pembuatan laporan bahan baku dapat dikerjakan secara komputerisasi sehingga dapat dijadikan acuan yang lebih akurat untuk inspeksi bahan baku pada periode berikutnya.

1.11 Saran

Saran yang dapat diberikan untuk pengembangan sistem informasi persediaan bahan baku ini selanjutnya adalah sebagai berikut:

1. Melakukan pengembangan sistem *output* dan *interface* aplikasi sistem persediaan bahan baku yang lebih baik untuk kenyamanan pengguna.
2. Melakukan pelatihan terlebih dahulu kepada pengguna aplikasi sistem persediaan bahan baku untuk mencegah kesalahan pengguna.
3. Melakukan perawatan terhadap sistem informasi persediaan bahan baku tersebut agar jangka waktu penggunaan menjadi lebih lama.

DAFTAR PUSTAKA

Bahagia, Senator Nuh , 2006, *Sistem Inventori*, ITB, Bandung.

Denies, Alan, Wixom, Barbara Haley and Tegarden, David, 2012, *Systems Analysis and Design 5th Edition*, John Wiley&Sons, Inc.

Fairuz, 2017, *Stakeholder Sistem Informasi*, fairuzelsaid.wordpress.com, 2017.
(Tanggal Akses: 8 April 2017)

Fitz Gerald, Jerry, 2014, *Konsep Dasar Sistem*, Graha Ilmu, Yogyakarta.

Holy Igun, Yunarto dan Martinus Getty, Santika, 2005, *Business Concept Implementation Series in Inventory Management*. Elex Media. Jakarta.

Jogiyanto, 2005. *Analisis & Desain*, Andi, Yogyakarta.

O'Brien, James A, 2005, *Pengantar Sistem Informasi : Persepektif Bisnis dan Manajerial*, (12th edition), Salemba edition, Salemba Empat, Jakarta.

Oktarini, Amyra Widya, *Safety Stock*, awosassite.blogspot.co.id, 2017.
(Tanggal Akses: 5 November 2017)

Rangkuti, Freddy, 2002, *Manajemen Persediaan Aplikasi di Bidang Bisnis*, Andi, Yogyakarta.

Ristono, Agus, 2009, *Manajemen Persediaan*, Graha Ilmu, Yogyakarta.

S, Rosa A, Shalahuddin, M, 2013, *Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*, Informatika, Bandung.

Sutabri, Tata. S.Kom., MM, 2004, *Analisa Sistem Informasi*, Edisi Pertama, Andi, Yogyakarta.

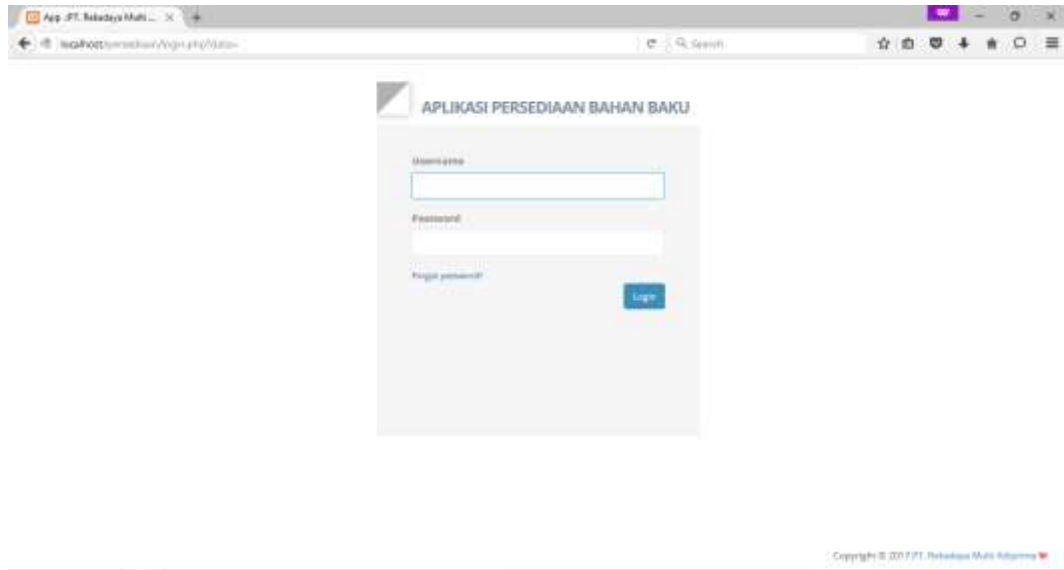
Tersine, Richard J., 1994, *Principles of Inventory and Materials Management*, United States of America, Prentice Hall.

Warman John, 2004, *Manajemen Pergudangan Seri Manajemen No.57*, Pustaka Sinar Harapan, Jakarta.

LAMPIRAN

a. Form login

Lampiran A



b. Menu Utama

Master Data

- Dashboard
- Penerimaan
- Supplier
- Bahan Baku

Transaksi Data

- BB Request
- PG Request
- BB Masuk
- BB Keluar

Report

- Master Data
- Transaction Data

Transaksi 0
 Jumlah transaksi Barang Masuk Hari ini

#	Date	No. Transaksi	Qty	Jumlah	Total
1	2017-11-12	0000000004	000000001	100	Detail
2	2017-09-25	0000000003	000000000	20	Detail
3	2017-09-25	0000000002	000000000	000	Detail
4	2017-09-25	0000000001	000000000	100	Detail

Transaksi 0
 Jumlah transaksi Barang Keluar Hari ini

#	Date	No. Transaksi	No. Request	Jumlah	Total
1	2017-11-12	0000000000	0000000000	20	Detail
2	2017-09-25	0000000005	0000000005	50	Detail
3	2017-09-25	0000000004	0000000004	75	Detail
4	2017-09-25	0000000003	0000000003	80	Detail
5	2017-09-25	0000000002	0000000002	30	Detail

Transaksi 1
 Jumlah transaksi Permintaan Barang Hari ini

Stok Bahan Baku

No.

Qty

Login.php

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <!-- <meta content="IE=edge" http-equiv="X-UA-Compatible"> -->
  <meta content="width=device-width, initial-scale=1"
name="viewport">
  <meta content="" name="description">
  <meta content="" name="author">
  <link href="asset/ico/favicon.ico" rel="shortcut icon">

  <title>App .:PT. Rekadaya Multi Adiprima:.</title>

  <!-- Bootstrap core CSS -->
  <link rel="stylesheet" href="asset/css/bootstrap.css">
  <!-- Bootstrap theme -->

  <!-- Custom styles for this template -->

  <link rel="stylesheet" href="asset/css/login.css">
  <link rel="stylesheet" href="asset/css/theme2.css">
  <link rel="stylesheet" href="asset/css/dashboard.css">
  <link rel="stylesheet" href="asset/css/style.css">
  <link rel="stylesheet" href="asset/css/dripicon.css">
  <link rel="stylesheet" href="asset/css/typicons.css" />
  <link rel="stylesheet" href="asset/css/responsive.css">
  <link rel="stylesheet" href="asset/js/tip/tooltipster.css">
  <link rel="stylesheet" type="text/css"
href="asset/js/vegas/jquery.vegas.css" />
  <link rel="stylesheet" type="text/css" href="asset/js/number-progress-
bar/number-pb.css">
  <!-- pace loader -->
  <script src="asset/js/pace/pace.js"></script>

</head>

<body>

  <div class="container-fluid ">
    <div class="row">
      <div class="col-md-4 col-md-offset-4">

```

```

<!-- Comtainer -->
<div class="paper-wrap bevel tlbr">
  <div id="paper-top">
    <div class="row">
      <div class="col-lg-12 no-pad">
        <a class="navbar-brand logo-text" href="#">Aplikasi
Persediaan Bahan Baku</a>
      </div>
    </div>
  </div>

  <!-- CONTENT -->
  <div style="min-height:400px;" class="wrap-fluid"
id="paper-bg">

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

          <form role="form" action="main/?page=login-cek"
method="post">
            <div class="form-group">
              <label>Username</label>
              <input type="text" name="username"
class="form-control" autofocus>
            </div>
            <div class="form-group">
              <label>Password</label>
              <input type="password" name="password"
class="form-control">
            </div>
            <div class=" pull-left">
              <a href="#" class="pull-right label-
forgot">Forgot password?</a>
            </div><br/>

            <button class="btn btn-primary pull-right"
type="submit" name="btnLogin">
              Login
            </button>
          </form>

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

```

```

        </div>
    </div>
</div>
</div>
</div>
<!-- / FOOTER -->
<div id="footer">
    <div class="devider-footer-left"></div>
    <div class="time">
        <p id="spanDate"></p>
        <p id="clock"></p>
    </div>
    <div class="copyright">Copyright &copy; 2017 <a href="#">PT.
    Rekadaya Multi Adiprima</a><i class="fontello-heart-filled text-
    red"></i>
    </div>
</div>
<!-- Container -->
</div>
</div>
</div>
</div>
</body>
</html>

```

Dashboard.php

```

<?php
if($_SESSION['bagian']=='Administrator') { ?>
<div class="row">
<?php # Perintah untuk menampilkan Semua Daftar Transaksi Permintaan
Material

    $jmlbr=mysql_num_rows(mysql_query("select * from barang_masuk
    where tanggal='".date('Y-m-d').""));
    $mySql = "SELECT * FROM barang_masuk ORDER BY
    id_barang_masuk DESC limit 5 ";
    $myQry = mysql_query($mySql, $koneksi) or die ("Query 1
    salah : ".mysql_error());
    ?>
<div class="col-lg-6">
    <div class="box">
        <!-- /.box-header -->
        <div class="box-body">
            <div class="news-widget">

```

```

        <h2>
            <span class="bg-green">Today</span>
        </h2>
        <i class="fontello-down-outline"></i>
        <h4 class="text-green"><?php echo $jmlbr; ?></h4>
        <h5>Jumlah transaksi <b>Barang Masuk</b> hari
    ini</h5>

        <div style="clear:both;"></div>
    </div>
<table class="table table-hover">
<thead>
    <tr>
        <th>#</th>
        <th>Date</th>
        <th>No. Transaksi</th>
        <th>DO#</th>
        <th>Jumlah</th>
        <th>Tool</th>
    </tr>
</thead>
<?php

    $nomor =0;
    while ($myData = mysql_fetch_array($myQry)) {
        $nomor++;

        # Membaca Kode Penjualan/ Nomor transaksi
        $noNota = $myData['id_barang_masuk'];

        # Menghitung Total Penjualan (belanja) setiap nomor
        transaksi
            $my2Sql = "SELECT SUM(jumlah) AS total
                        FROM barang_masuk_detail WHERE
id_barang_masuk='$noNota'";
            $my2Qry = mysql_query($my2Sql, $koneksiDB) or die
("Query 2 salah : ".mysql_error());
            $my2Data = mysql_fetch_array($my2Qry);
        ?>
    <tr>
        <td><?php echo $nomor; ?></td>
        <td><?php echo $myData['tanggal']; ?></td>
        <td><?php echo $myData['id_barang_masuk']; ?></td>
        <td><?php echo $myData['no_do']; ?></td>
        <td><?php echo $my2Data['total'];?></td>
    </tr>

```

```

        <td><a href="?page=detail-report-in-material&id=?php echo
$myData['id_barang_masuk']; ?>" title="Detail
Transaksi">Detail</a></td>
    </tr>
    <?php } ?>
</table>

    </div>
    <!-- /.box-body -->
</div>
    <!-- /.box -->
</div>
    <?php # Perintah untuk menampilkan Semua Daftar Transaksi
Permintaan Material

    $jmlbm=mysql_num_rows(mysql_query("select * from barang_keluar
where tanggal='".date('Y-m-d').""));
    $mySql2 = "SELECT * FROM barang_keluar ORDER BY
id_keluar DESC limit 5 ";
    $myQry2 = mysql_query($mySql2, $koneksi) or die ("Query 2
salah : ".mysql_error());
    ?>
<div class="col-lg-6">
    <div class="box">
        <div class="box-body">
            <div class="news-widget">
                <h2>
                    <span class="bg-aqua">Today</span>
                </h2>
                <i class="fontello-up-outline"></i>
                <h4 class="text-aqua"><?php echo $jmlbm; ?></h4>
                <h5>Jumlah transaksi <b>Barang Keluar</b> hari
ini</h5>
                <div style="clear:both;"></div>
            </div>
            <table class="table table-hover">
                <thead>
                    <tr>
                        <th>#</th>
                        <th>Date</th>
                        <th>No. Transaksi</th>
                        <th>No. Request</th>
                        <th>Jumlah</th>
                        <th>Tool</th>
                    </tr>
                </thead>
                <?php

```

```

$nomor =0;
while ($myData2 = mysql_fetch_array($myQry2)) {
    $nomor++;

    # Membaca Kode Penjualan/ Nomor transaksi
    $snoNota2 = $myData2['id_keluar'];

    # Menghitung Total Penjualan (belanja) setiap nomor
    transaksi
    $my2Sql2 = "SELECT SUM(jumlah) AS total
                FROM barang_keluar_detail WHERE
id_keluar='$snoNota2'";
    $my2Qry2 = mysql_query($my2Sql2, $koneksi) or die
("Query 2 salah : ".mysql_error());
    $my2Data2 = mysql_fetch_array($my2Qry2);

    ?>
    <tr>
    <td><?php echo $nomor; ?></td>
    <td><?php echo $myData2['tanggal']; ?></td>
    <td><?php echo $myData2['id_keluar']; ?></td>
    <td><?php echo $myData2['id_request']; ?></td>
    <td><?php echo $my2Data2['total']; ?></td>
    <td><a href="?page=detail-report-out-material&id=<?php echo
$myData2['id_keluar']; ?>" title="Detail Transaksi">Detail</a></td>
    </tr>
    <?php } ?>
</table>

</div>

</div>
<!-- /.box -->
</div>

<?php # Perintah untuk menampilkan Semua Daftar Transaksi Permintaan
Material

    $jmlR=mysql_num_rows(mysql_query("select * from request_part
where tgl_request='".date('Y-m-d')."'"));
    $mySql4 = "SELECT * FROM request_part ORDER BY
id_request DESC limit 5 ";
    $myQry4 = mysql_query($mySql4, $koneksi) or die ("Query 1
salah : ".mysql_error());
    ?>
    <div class="col-lg-6">
    <div class="box">
    <!-- /.box-header -->

```

```

<div class="box-body">
    <div class="news-widget">

        <h2>
            <span class="bg-green">Today</span>
        </h2>
        <i class="icon-document"></i>
        <h4 class="text-green"><?php echo $jmlR; ?></h4>
        <h5>Jumlah transaksi <b>Permintaan Barang</b>
hari ini</h5>

        <div style="clear:both;"></div>
    </div>

<table class="table table-hover">
<thead>
    <tr>
        <th>#</th>
        <th>Date</th>
        <th>No. Transaksi</th>
        <th>Status</th>
        <th>Jumlah</th>
        <th>Tool</th>
    </tr>
</thead>
<?php
    $nomor =0;
    while ($myData4 = mysql_fetch_array($myQry4)) {
        $nomor++;

        # Membaca Kode Penjualan/ Nomor transaksi
        $noNota4 = $myData4['id_request'];

        # Menghitung Total Penjualan (belanja) setiap nomor
transaksi
        $my2Sql4 = "SELECT SUM(jumlah_part) AS total4
                    FROM request_part_detail WHERE
id_request='$noNota4'";
        $my2Qry4 = mysql_query($my2Sql4, $koneksiDb) or die
("Query 2 salah : ".mysql_error());
        $my2Data4 = mysql_fetch_array($my2Qry4);
        ?>
    <tr>
        <td><?php echo $nomor; ?></td>
        <td><?php echo $myData4['tgl_request']; ?></td>
        <td><?php echo $myData4['id_request']; ?></td>
        <td><?php echo $myData4['status']; ?></td>

```

```

        <td><?php echo $my2Data4['total4'];?></td>
        <td><a href="?page=detail-report-request-material&id=?php echo
$myData4['id_request']; ?>" title="Detail Transaksi">Detail</a></td>
    </tr>
<?php } ?>
</table>

</div>
<!-- /.box-body -->
</div>
<!-- /.box -->
</div>

```

```

<div class="col-lg-5">
    <div class="box">
        <script src="../../asset/js/jquery.min.js" type="text/javascript"></script>
        <script src="../../asset/js/highcharts.js" type="text/javascript"></script>

        <script type="text/javascript">
            var chart1; // globally available
            $(document).ready(function() {
                chart1 = new Highcharts.Chart({
                    chart: {
                        renderTo: 'container',
                        type: 'column'
                    },
                    title: {
                        text: 'Stok Bahan Baku '
                    },
                    xAxis: {
                        categories: ['Kategori']
                    },
                    yAxis: {
                        title: {
                            text: 'Stok'
                        }
                    },
                    series:
                    [
                        <?php
                        # Perintah untuk menampilkan Semua Daftar Transaksi Penjualan
                        $mySql3 = "SELECT id_category, category_name FROM
tcategory ";
                        $myQry3 = mysql_query($mySql3, $koneksi) or die ("Query 1
salah : ".mysql_error());
                        while ($myData3 = mysql_fetch_array($myQry3)) {

```

```

# Membaca Kode Penjualan/ Nomor transaksi
$kdKategori = $myData3['id_category'];

# Menghitung Total Penjualan (belanja) setiap nomor
transaksi
    $my22Sql = "SELECT sum(stock) as jumlah FROM
tbarang WHERE id_category='$kdKategori'";
    $my22Qry = mysql_query($my22Sql, $koneksiDb) or die
("Query 2 salah : ".mysql_error());
    $my2Data3 = mysql_fetch_array($my22Qry);
    ?>
    {
        name: '<?php echo $myData3['category_name']; ?>',
        data: [<?php echo $my2Data3['jumlah']; ?>]
    },
    <?php } ?>
    ]
    });
    });
</script>

<div id='container'></div>
</div>
<!-- /.box -->
</div>

<?php
}elseif($_SESSION['bagian']=='Pimpinan') { ?>
<div class="row">
<?php # Perintah untuk menampilkan Semua Daftar Transaksi Permintaan
Material

    $jmlbr=mysql_num_rows(mysql_query("select * from barang_masuk
where tanggal='".$_date('Y-m-d').""));
    $mySql = "SELECT * FROM barang_masuk ORDER BY
id_barang_masuk DESC limit 5 ";
    $myQry = mysql_query($mySql, $koneksiDb) or die ("Query 1
salah : ".mysql_error());
    ?>
<div class="col-lg-6">
    <div class="box">
        <!-- /.box-header -->
        <div class="box-body">
            <div class="news-widget">

```

```

        <h2>
            <span class="bg-green">Today</span>
        </h2>
        <i class="fontello-down-outline"></i>
        <h4 class="text-green"><?php echo $jmlbr; ?></h4>
        <h5>Jumlah transaksi <b>Barang Masuk</b> hari
    ini</h5>

        <div style="clear:both;"></div>
    </div>
<table class="table table-hover">
<thead>
    <tr>
        <th>#</th>
        <th>Date</th>
        <th>No. Transaksi</th>
        <th>DO#</th>
        <th>Jumlah</th>
        <th>Tool</th>
    </tr>
</thead>
<?php

    $nomor =0;
    while ($myData = mysql_fetch_array($myQry)) {
        $nomor++;

        # Membaca Kode Penjualan/ Nomor transaksi
        $noNota = $myData['id_barang_masuk'];

        # Menghitung Total Penjualan (belanja) setiap nomor
        transaksi
            $my2Sql = "SELECT SUM(jumlah) AS total
                        FROM barang_masuk_detail WHERE
id_barang_masuk='$noNota'";
            $my2Qry = mysql_query($my2Sql, $koneksiDB) or die
("Query 2 salah : ".mysql_error());
            $my2Data = mysql_fetch_array($my2Qry);
        ?>
    <tr>
        <td><?php echo $nomor; ?></td>
        <td><?php echo $myData['tanggal']; ?></td>
        <td><?php echo $myData['id_barang_masuk']; ?></td>
        <td><?php echo $myData['no_do']; ?></td>
        <td><?php echo $my2Data['total'];?></td>
    </tr>

```

```

        <td><a href="?page=detail-report-in-material&id=?php echo
$myData['id_barang_masuk']; ?>" title="Detail
Transaksi">Detail</a></td>
    </tr>
<?php } ?>
</table>

    </div>
    <!-- /.box-body -->
</div>
<!-- /.box -->
</div>
<?php # Perintah untuk menampilkan Semua Daftar Transaksi
Permintaan Material

    $jmlbm=mysql_num_rows(mysql_query("select * from barang_keluar
where tanggal='".date('Y-m-d').""));
    $mySql2 = "SELECT * FROM barang_keluar ORDER BY
id_keluar DESC limit 5 ";
    $myQry2 = mysql_query($mySql2, $koneksi) or die ("Query 2
salah : ".mysql_error());
    ?>
<div class="col-lg-6">
    <div class="box">
        <div class="box-body">
            <div class="news-widget">
                <h2>
                    <span class="bg-aqua">Today</span>
                </h2>
                <i class="fontello-up-outline"></i>
                <h4 class="text-aqua"><?php echo $jmlbm; ?></h4>
                <h5>Jumlah transaksi <b>Barang Keluar</b> hari
ini</h5>
                <div style="clear:both;"></div>
            </div>
            <table class="table table-hover">
                <thead>
                    <tr>
                        <th>#</th>
                        <th>Date</th>
                        <th>No. Transaksi</th>
                        <th>No. Request</th>
                        <th>Jumlah</th>
                        <th>Tool</th>
                    </tr>
                </thead>
                <?php

```

```

$nomor =0;
while ($myData2 = mysql_fetch_array($myQry2)) {
    $nomor++;

    # Membaca Kode Penjualan/ Nomor transaksi
    $snoNota2 = $myData2['id_keluar'];

    # Menghitung Total Penjualan (belanja) setiap nomor
    transaksi
    $my2Sql2 = "SELECT SUM(jumlah) AS total
                FROM barang_keluar_detail WHERE
id_keluar='$snoNota2'";
    $my2Qry2 = mysql_query($my2Sql2, $koneksi) or die
("Query 2 salah : ".mysql_error());
    $my2Data2 = mysql_fetch_array($my2Qry2);

    ?>
    <tr>
    <td><?php echo $nomor; ?></td>
    <td><?php echo $myData2['tanggal']; ?></td>
    <td><?php echo $myData2['id_keluar']; ?></td>
    <td><?php echo $myData2['id_request']; ?></td>
    <td><?php echo $my2Data2['total'];?></td>
    <td><a href="?page=detail-report-out-material&id=<?php echo
$myData2['id_keluar']; ?>" title="Detail Transaksi">Detail</a></td>
    </tr>
    <?php } ?>
</table>

</div>

</div>
<!-- /.box -->
</div>

<?php # Perintah untuk menampilkan Semua Daftar Transaksi Permintaan
Material

    $jmlR=mysql_num_rows(mysql_query("select * from request_part
where tgl_request='".date('Y-m-d')."'"));
    $mySql4 = "SELECT * FROM request_part ORDER BY
id_request DESC limit 5 ";
    $myQry4 = mysql_query($mySql4, $koneksi) or die ("Query 1
salah : ".mysql_error());
    ?>
    <div class="col-lg-6">
    <div class="box">
    <!-- /.box-header -->

```

```

<div class="box-body">
    <div class="news-widget">

        <h2>
            <span class="bg-green">Today</span>
        </h2>
        <i class="icon-document"></i>
        <h4 class="text-green"><?php echo $jmlR; ?></h4>
        <h5>Jumlah transaksi <b>Permintaan Barang</b>
hari ini</h5>

        <div style="clear:both;"></div>
    </div>

<table class="table table-hover">
<thead>
    <tr>
        <th>#</th>
        <th>Date</th>
        <th>No. Transaksi</th>
        <th>Status</th>
        <th>Jumlah</th>
        <th>Tool</th>
    </tr>
</thead>
<?php
    $nomor =0;
    while ($myData4 = mysql_fetch_array($myQry4)) {
        $nomor++;

        # Membaca Kode Penjualan/ Nomor transaksi
        $noNota4 = $myData4['id_request'];

        # Menghitung Total Penjualan (belanja) setiap nomor
transaksi
        $my2Sql4 = "SELECT SUM(jumlah_part) AS total4
                    FROM request_part_detail WHERE
id_request='$noNota4'";
        $my2Qry4 = mysql_query($my2Sql4, $koneksiDb) or die
("Query 2 salah : ".mysql_error());
        $my2Data4 = mysql_fetch_array($my2Qry4);
        ?>
    <tr>
        <td><?php echo $nomor; ?></td>
        <td><?php echo $myData4['tgl_request']; ?></td>
        <td><?php echo $myData4['id_request']; ?></td>
        <td><?php echo $myData4['status']; ?></td>

```

```

        <td><?php echo $my2Data4['total4'];?></td>
        <td><a href="?page=detail-report-request-material&id=<?php echo
$myData4['id_request']; ?>" title="Detail Transaksi">Detail</a></td>
    </tr>
<?php } ?>
</table>

    </div>
    <!-- /.box-body -->
</div>
<!-- /.box -->
</div>

```

```

<div class="col-lg-5">
    <div class="box">
        <script src="../../asset/js/jquery.min.js" type="text/javascript"></script>
        <script src="../../asset/js/highcharts.js" type="text/javascript"></script>

```

```

            <script type="text/javascript">
                var chart1; // globally available
$(document).ready(function() {
    chart1 = new Highcharts.Chart({
        chart: {
            renderTo: 'container',
            type: 'column'
        },
        title: {
            text: 'Stok Bahan Baku '
        },
        xAxis: {
            categories: ['Kategori']
        },
        yAxis: {
            title: {
                text: 'Stok'
            }
        },
        series:
        [
            <?php
            # Perintah untuk menampilkan Semua Daftar Transaksi Penjualan
            $mySql3 = "SELECT id_category, category_name FROM
tcategory ";
            $myQry3 = mysql_query($mySql3, $koneksi) or die ("Query 1
salah : ".mysql_error());
            while ($myData3 = mysql_fetch_array($myQry3)) {

```

```

# Membaca Kode Penjualan/ Nomor transaksi
$kdKategori = $myData3['id_category'];

# Menghitung Total Penjualan (belanja) setiap nomor
transaksi
    $my22Sql = "SELECT sum(stock) as jumlah FROM
tbarang WHERE id_category='$kdKategori'";
    $my22Qry = mysql_query($my22Sql, $koneksiDb) or die
("Query 2 salah : ".mysql_error());
    $my2Data3 = mysql_fetch_array($my22Qry);
    ?>
    {
        name: '<?php echo $myData3['category_name']; ?>',
        data: [<?php echo $my2Data3['jumlah']; ?>]
    },
    <?php } ?>
    ]
    });
    });
</script>

<div id='container'></div>
</div>
<!-- /.box -->
</div>

<?php
}elseif($_SESSION['bagian']=='Gudang') { ?>
<div class="row">
<?php # Perintah untuk menampilkan Semua Daftar Transaksi Permintaan
Material

    $jmlbr=mysql_num_rows(mysql_query("select * from barang_masuk
where tanggal='".$_date('Y-m-d').""));
    $mySql = "SELECT * FROM barang_masuk ORDER BY
id_barang_masuk DESC limit 5 ";
    $myQry = mysql_query($mySql, $koneksiDb) or die ("Query 1
salah : ".mysql_error());
    ?>
<div class="col-lg-6">
    <div class="box">
        <!-- /.box-header -->
        <div class="box-body">
            <div class="news-widget">

```

```

<h2>
    <span class="bg-green">Today</span>
</h2>
<i class="fontello-down-outline"></i>
<h4 class="text-green"><?php echo $jmlbr; ?></h4>
<h5>Jumlah transaksi <b>Barang Masuk</b> hari
ini</h5>

    <div style="clear:both;"></div>
</div>
<table class="table table-hover">
<thead>
<tr>
<th>#</th>
<th>Date</th>
<th>No. Transaksi</th>
<th>DO#</th>
<th>Jumlah</th>
<th>Tool</th>
</tr>
</thead>
<?php

    $nomor =0;
    while ($myData = mysql_fetch_array($myQry)) {
        $nomor++;

        # Membaca Kode Penjualan/ Nomor transaksi
        $noNota = $myData['id_barang_masuk'];

        # Menghitung Total Penjualan (belanja) setiap nomor
transaksi
        $my2Sql = "SELECT SUM(jumlah) AS total
                    FROM barang_masuk_detail WHERE
id_barang_masuk='$noNota'";
        $my2Qry = mysql_query($my2Sql, $koneksiDb) or die
("Query 2 salah : ".mysql_error());
        $my2Data = mysql_fetch_array($my2Qry);
        ?>
<tr>
<td><?php echo $nomor; ?></td>
<td><?php echo $myData['tanggal']; ?></td>
<td><?php echo $myData['id_barang_masuk']; ?></td>
<td><?php echo $myData['no_do']; ?></td>
<td><?php echo $my2Data['total'];?></td>

```

```

        <td><a href="?page=detail-report-in-material&id=?php echo
$myData['id_barang_masuk']; ?>" title="Detail
Transaksi">Detail</a></td>
    </tr>
    <?php } ?>
</table>

    </div>
    <!-- /.box-body -->
</div>
<!-- /.box -->
</div>
    <?php # Perintah untuk menampilkan Semua Daftar Transaksi
Permintaan Material

    $jmlbm=mysql_num_rows(mysql_query("select * from barang_keluar
where tanggal='".date('Y-m-d')."'"));
    $mySql2 = "SELECT * FROM barang_keluar ORDER BY
id_keluar DESC limit 5 ";
    $myQry2 = mysql_query($mySql2, $koneksi) or die ("Query 2
salah : ".mysql_error());
    ?>
<div class="col-lg-6">
    <div class="box">
        <div class="box-body">
            <div class="news-widget">
                <h2>
                    <span class="bg-aqua">Today</span>
                </h2>
                <i class="fontello-up-outline"></i>
                <h4 class="text-aqua"><?php echo $jmlbm; ?></h4>
                <h5>Jumlah transaksi <b>Barang Keluar</b> hari
            </h5>
        </div>
        <div style="clear:both;"></div>
    </div>

    <table class="table table-hover">
<thead>
    <tr>
        <th>#</th>
        <th>Date</th>
        <th>No. Transaksi</th>
        <th>No. Request</th>
        <th>Jumlah</th>
        <th>Tool</th>
    </tr>
</thead>
    <?php

```

```

$nomor =0;
while ($myData2 = mysql_fetch_array($myQry2)) {
    $nomor++;

    # Membaca Kode Penjualan/ Nomor transaksi
    $snoNota2 = $myData2['id_keluar'];

    # Menghitung Total Penjualan (belanja) setiap nomor
    transaksi
    $my2Sql2 = "SELECT SUM(jumlah) AS total
                FROM barang_keluar_detail WHERE
id_keluar='$snoNota2'";
    $my2Qry2 = mysql_query($my2Sql2, $koneksi) or die
("Query 2 salah : ".mysql_error());
    $my2Data2 = mysql_fetch_array($my2Qry2);

    ?>
    <tr>
    <td><?php echo $nomor; ?></td>
    <td><?php echo $myData2['tanggal']; ?></td>
    <td><?php echo $myData2['id_keluar']; ?></td>
    <td><?php echo $myData2['id_request']; ?></td>
    <td><?php echo $my2Data2['total'];?></td>
    <td><a href="?page=detail-report-out-material&id=<?php echo
$myData2['id_keluar']; ?>" title="Detail Transaksi">Detail</a></td>
    </tr>
    <?php } ?>
</table>

    </div>

    </div>
    <!-- /.box -->
</div>

<div class="col-lg-10">
    <div class="box">
    <script src="../../asset/js/jquery.min.js" type="text/javascript"></script>
    <script src="../../asset/js/highcharts.js" type="text/javascript"></script>

        <script type="text/javascript">
            var chart1; // globally available
$(document).ready(function() {
    chart1 = new Highcharts.Chart({
        chart: {
            renderTo: 'container',
            type: 'column'

```

```

    },
    title: {
        text: 'Stok Bahan Baku '
    },
    xAxis: {
        categories: ['Kategori']
    },
    yAxis: {
        title: {
            text: 'Stok'
        }
    },
    series:
    [
        <?php
        # Perintah untuk menampilkan Semua Daftar Transaksi Penjualan
        $mySql3 = "SELECT id_category, category_name FROM
tcategory ";
        $myQry3 = mysql_query($mySql3, $koneksiDb) or die ("Query 1
salah : ".mysql_error());
        while ($myData3 = mysql_fetch_array($myQry3)) {

            # Membaca Kode Penjualan/ Nomor transaksi
            $kdKategori = $myData3['id_category'];

            # Menghitung Total Penjualan (belanja) setiap nomor
transaksi
            $my22Sql = "SELECT sum(stock) as jumlah FROM
tbarang WHERE id_category='$kdKategori'";
            $my22Qry = mysql_query($my22Sql, $koneksiDb) or die
("Query 2 salah : ".mysql_error());
            $my2Data3 = mysql_fetch_array($my22Qry);

            ?>
            {
                name: '<?php echo $myData3['category_name']; ?>',
                data: [<?php echo $my2Data3['jumlah']; ?>]
            },
            <?php } ?>
        ]
    });
});
</script>

```

<div id='container'></div>

```

        </div>
    <!-- /.box -->
</div>
<?php
}
elseif ($_SESSION['bagian']=='Purchasing'){
    ?>
    <div class="row">

<div class="col-lg-10">
    <div class="box">
<script src="../../asset/js/jquery.min.js" type="text/javascript"></script>
<script src="../../asset/js/highcharts.js" type="text/javascript"></script>

        <script type="text/javascript">
            var chart1; // globally available
$(document).ready(function() {
    chart1 = new Highcharts.Chart({
        chart: {
            renderTo: 'container',
            type: 'column'
        },
        title: {
            text: 'Stok Bahan Baku '
        },
        xAxis: {
            categories: ['Kategori']
        },
        yAxis: {
            title: {
                text: 'Stok'
            }
        },
        series:
        [
            <?php
            # Perintah untuk menampilkan Semua Daftar Transaksi Penjualan
            $mySql3 = "SELECT id_category, category_name FROM
tcategory ";
            $myQry3 = mysql_query($mySql3, $koneksi) or die ("Query 1
salah : ".mysql_error());
            while ($myData3 = mysql_fetch_array($myQry3)) {

                # Membaca Kode Penjualan/ Nomor transaksi

```

```

$kdKategori = $myData3['id_category'];

# Menghitung Total Penjualan (belanja) setiap nomor
transaksi
    $my22Sql = "SELECT sum(stock) as jumlah FROM
tbarang WHERE id_category='$kdKategori'";
    $my22Qry = mysql_query($my22Sql, $koneksi) or die
("Query 2 salah : ".mysql_error());
    $my2Data3 = mysql_fetch_array($my22Qry);
    ?>
    {
        name: '<?php echo $myData3['category_name']; ?>',
        data: [<?php echo $my2Data3['jumlah']; ?>]
    },
    <?php } ?>
    ]
    });
    });
</script>

```

```

<div id='container'></div>
</div>
<!-- /.box -->
</div>
<?php
}
elseif ($_SESSION['bagian']=='Produksi'){
    ?>
    <div class="row">
        <?php # Perintah untuk menampilkan Semua Daftar Transaksi
Permintaan Material

        $jmlR=mysql_num_rows(mysql_query("select * from request_part
where tgl_request='".date('Y-m-d')."'"));
        $mySql4 = "SELECT * FROM request_part ORDER BY
id_request DESC limit 5 ";
        $myQry4 = mysql_query($mySql4, $koneksi) or die ("Query 1
salah : ".mysql_error());
        ?>
        <div class="col-lg-6">
            <div class="box">
                <!-- /.box-header -->
                <div class="box-body">
                    <div class="news-widget">

```

```

<h2>
    <span class="bg-green">Today</span>
</h2>
<i class="icon-document"></i>
<h4 class="text-green"><?php echo $jmlR; ?></h4>
<h5>Jumlah transaksi <b>Permintaan Barang</b>
hari ini</h5>

    <div style="clear:both;"></div>
</div>
<table class="table table-hover">
<thead>
    <tr>
        <th>#</th>
        <th>Date</th>
        <th>No. Transaksi</th>
        <th>Status</th>
        <th>Jumlah</th>
        <th>Tool</th>
    </tr>
</thead>
<?php
    $nomor =0;
    while ($myData4 = mysql_fetch_array($myQry4)) {
        $nomor++;

        # Membaca Kode Penjualan/ Nomor transaksi
        $noNota4 = $myData4['id_request'];

        # Menghitung Total Penjualan (belanja) setiap nomor
transaksi
        $my2Sql4 = "SELECT SUM(jumlah_part) AS total4
                    FROM request_part_detail WHERE
id_request='$noNota4'";
        $my2Qry4 = mysql_query($my2Sql4, $koneksi) or die
("Query 2 salah : ".mysql_error());
        $my2Data4 = mysql_fetch_array($my2Qry4);
        ?>
    <tr>
        <td><?php echo $nomor; ?></td>
        <td><?php echo $myData4['tgl_request']; ?></td>
        <td><?php echo $myData4['id_request']; ?></td>
        <td><?php echo $myData4['status']; ?></td>
        <td><?php echo $my2Data4['total4'];?></td>
        <td><a href="?page=detail-report-request-material&id=<?php echo
$myData4['id_request']; ?>" title="Detail Transaksi">Detail</a></td>

```

```

        </tr>
        <?php } ?>
    </table>

    </div>
    <!-- /.box-body -->
</div>
<!-- /.box -->
</div>

<div class="col-lg-5">
    <div class="box">
        <script src="../../asset/js/jquery.min.js" type="text/javascript"></script>
        <script src="../../asset/js/highcharts.js" type="text/javascript"></script>

        <script type="text/javascript">
            var chart1; // globally available
            $(document).ready(function() {
                chart1 = new Highcharts.Chart({
                    chart: {
                        renderTo: 'container',
                        type: 'column'
                    },
                    title: {
                        text: 'Stok Bahan Baku '
                    },
                    xAxis: {
                        categories: ['Kategori']
                    },
                    yAxis: {
                        title: {
                            text: 'Stok'
                        }
                    },
                    series:
                    [
                        <?php
                        # Perintah untuk menampilkan Semua Daftar Transaksi Penjualan
                        $mySql3 = "SELECT id_category, category_name FROM
tcategory ";
                        $myQry3 = mysql_query($mySql3, $koneksi) or die ("Query 1
salah : ".mysql_error());
                        while ($myData3 = mysql_fetch_array($myQry3)) {

                            # Membaca Kode Penjualan/ Nomor transaksi
                            $kdKategori = $myData3['id_category'];

```

```

# Menghitung Total Penjualan (belanja) setiap nomor
transaksi
    $my22Sql = "SELECT sum(stock) as jumlah FROM
tbarang WHERE id_category='$kdKategori'";
    $my22Qry = mysql_query($my22Sql, $koneksi) or die
("Query 2 salah : ".mysql_error());
    $my2Data3 = mysql_fetch_array($my22Qry);
    ?>
    {
        name: '<?php echo $myData3['category_name']; ?>',
        data: [<?php echo $my2Data3['jumlah']; ?>]
    },
    <?php } ?>
]
});
});
</script>

```

```

<div id='container'></div>
</div>
<!-- /.box -->
</div>

<?php }else{
    echo "<h2>Selamat datang .....!</h2>";
    echo "<b>Anda belum login, silahkan <a href='?page=Login'
alt='Login'>login </a>untuk mengakses sitem ini ";
}
?>

</div>

```