

No. Dok : 6555

DJ 655.785 Pan R

**RANCANG BANGUN SISTEM INFORMASI MANAJEMEN  
GUDANG BAHAN BAKU PADA PT MULTI PRIMA  
SEJAHTERA TBK MENGGUNAKAN  
CODEIGNITER 3.1.8 DAN MYSQL 5.6.26**

**TUGAS AKHIR**

Diajukan Untuk Memenuhi Sebagian Syarat Penyelesaian Program Sarjana  
Terapan Program Studi Sistem Informasi Industri Otomotif Pada  
Politeknik STMI Jakarta Kementerian Perindustrian R.I

OLEH  
HENDRA DANOPAN  
1315029

**DATA BUKU PERPUSTAKAAN**

Tgl Terima	24/07/22
No Induk Buku	488/S110/SB/TA/22



**POLITEKNIK STMI JAKARTA  
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA  
2019**

**SUMBANGAN ALUMNI**

**POLITEKNIK STMI JAKARTA  
KEMENTERIAN PERINDUSTRIAN RI  
TANDA PERSETUJUAN DOSEN PEMBIMBING**

JUDUL TUGAS AKHIR:

**RANCANG BANGUN SISTEM INFORMASI MANAJEMEN  
GUDANG BAHAN BAKU PADA PT MULTI PRIMA  
SEJAHTERA TBK MENGGUNAKAN  
CODEIGNITER 3.1.8 DAN MYSQL 5.6.26**

DISUSUN OLEH:

**NAMA : HENDRA DANOPAN  
NIM : 1315029  
PROGRAM STUDI : SISTEM INFORMASI INDUSTRI OTOMOTIF**

Telah diperiksa dan disetujui untuk memenuhi sebagian syarat penyelesaian Program Sarjana Terapan Program Studi Sistem Informasi Industri Otomotif pada Politeknik STMI Jakarta.

Jakarta, 09 Agustus 2019  
Dosen Pembimbing



Triana Fatmawati, S.T., M.T.  
NIP. 19800514.200502.2.001

**POLITEKNIK STMI JAKARTA  
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA**

**LEMBAR PENGESAHAN**

**JUDUL TUGAS AKHIR:  
RANCANG BANGUN SISTEM INFORMASI MANAJEMEN GUDANG  
BAHAN BAKU PADA PT MULTI PRIMA SEJAHTERA TBK  
MENGGUNAKAN CODEIGNITER 3.1.8 DAN MYSQL 5.6.26**

**Disusun Oleh:**

Nama : Hendra Danopan  
NIM : 1315029  
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 Selasa Tanggal 17 September 2019.

Jakarta, 18 September 2019

Dosen Pembimbing

Triana Fatmawati, S.T., MT  
NIP : 198005142005022001

Ketua Penguji

Lucky Heriyanto, S.T., M.T.I  
NIP : 197908202009011009

Dosen Penguji

Dedy Trisanto, S.Kom, MMSI  
NIP : 197805052005021002

Dosen Penguji

Ahmad Juniar, S.Kom, M.T  
NIP : 197906052006041002



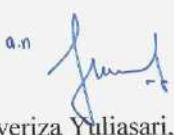
**LEMBAR BIMBINGAN PENYUSUNAN TUGAS AKHIR**

Nama : Hendra Danopan  
NIM : 1315029  
Judul Tugas Akhir : Rancang Bangun Sistem Informasi Manajemen Gudang Bahan Baku Pada PT Multi Prima Sejahtera Tbk Menggunakan CodeIgniter 3.1.8 Dan MySQL 5.6.26  
Pembimbing : Triana Fatmawati, S.T., M.T.

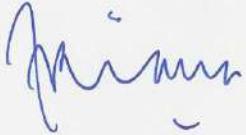
Tanggal	Bab	Keterangan	Paraf
29 Mei 2019	Bab I	Pendahuluan	
18 Juni 2019	Bab I	Perbaikan Pendahuluan	
19 Juni 2019	Bab II	Landasan Teori	
21 Juni 2019	Bab II	Perbaikan Landasan Teori	
27 Juni 2019	Bab III	Metodologi Penelitian	
3 Juli 2019	Bab III	Perbaikan Metodologi Penelitian	
11 Juli 2019	Bab IV	Pengumpulan dan Pengolahan Data	
22 Juli 2019	Bab V	Analisis dan Pembahasan	
6 Agustus 2019	Bab V	Perbaikan Analisis dan Pembahasan	
	Program	Demo Program	
19 Agustus 2019	Bab V	Perbaikan Analisis dan Pembahasan	

Mengetahui,

Ketua Program Studi  
Sistem Informasi Industri  
Otomotif

a.n  
  
Noveriza Yuliasari, M.T.  
NIP: 197811212009012003

Dosen Pembimbing

  
Triana Fatmawati, S.T., M.T.  
NIP: 19800514.200502.2.001

## LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : Hendra Danopan

NIM : 1315029

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

**“Rancang Bangun Sistem Informasi Manajemen Gudang Bahan Baku Pada PT Multi Prima Sejahtera Tbk Menggunakan CodeIgniter 3.1.8 Dan MySQL 5.6.26”.**

- **Dibuat** dan diselesaikan sendiri, dengan menggunakan literatur hasil kuliah, survei 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, 09 Agustus 2019



## ABSTRAK

PT Multi Prima Sejahtera Tbk merupakan perusahaan yang bergerak dalam bidang manufaktur. Kegiatan utama perusahaan adalah memproduksi busi (*spark plug*) untuk kendaraan bermotor baik mobil maupun motor. PT Multi Prima Sejahtera Tbk membutuhkan pengembangan sebuah sistem pada sistem manajemen gudang bahan baku khususnya proses penerimaan dan pengeluaran bahan baku. Terdapat beberapa masalah yang ada pada proses penerimaan dan pengeluaran bahan baku di Bagian Gudang PT Multi Prima Sejahtera Tbk, seperti data yang terkait dengan proses penerimaan dan pengeluaran bahan baku masih disimpan dan diolah secara manual menggunakan *form-form* yang kemudian ditumpuk di dalam sebuah arsip penyimpanan. Hal ini dapat membuat data yang terkait dengan proses penerimaan dan pengeluaran bahan baku sulit dicari, mudah hilang dan rusak. Selain itu, belum adanya sistem informasi pada proses penerimaan dan pengeluaran bahan baku menyebabkan karyawan sulit memperoleh informasi mengenai data bahan baku yang diterima dan dikeluarkan sehingga proses penerimaan dan pengeluaran bahan baku membutuhkan waktu yang cukup lama. Pengembangan sistem informasi manajemen gudang bahan baku khususnya sistem informasi penerimaan dan pengeluaran bahan baku menggunakan metode *waterfall*. Analisis dan perancangan yang digunakan adalah analisis dan rancangan berorientasi objek. Pemodelan sistem menggunakan *Unified Modeling Language* (UML), pemodelan data menggunakan *Entity Relationship Diagram* (ERD) dan *Conceptual Data Model* (CDM), perancangan sistem menggunakan *Windows Navigation Diagram* (WND) dan perancangan antarmuka. Pembuatan aplikasi sistem informasi penerimaan dan pengeluaran bahan baku menggunakan CodeIgniter 3.1.8 dan MySQL 5.6.26, agar dapat memudahkan perusahaan dalam mengelola dan melihat informasi yang ada serta dapat menghemat waktu pencarian data. Sistem informasi manajemen gudang bahan baku khususnya sistem informasi penerimaan dan pengeluaran bahan baku merupakan aplikasi yang sesuai untuk mempermudah dalam proses penerimaan dan pengeluaran bahan baku.

**Kata kunci:** Sistem Informasi, Penerimaan Bahan Baku, Pengeluaran Bahan Baku, *Waterfall*, UML, CodeIgniter 3.1.8, MySQL 5.6.26.

## KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya, penulis dapat menyelesaikan Tugas Akhir ini dengan judul **“Rancang Bangun Sistem Informasi Manajemen Gudang Bahan Baku Pada PT Multi Prima Sejahtera Tbk Menggunakan CodeIgniter 3.1.8 Dan MySQL 5.6.26”**.

Laporan ini disusun guna memenuhi sebagian syarat yang harus dipenuhi dalam menempuh jenjang Sarjana Terapan Program Studi Sistem Informasi Industri Otomotif pada Politeknik STMI Jakarta.

Adapun tujuan dari Tugas Akhir itu sendiri adalah untuk menganalisis sistem dari suatu perusahaan, selain itu juga untuk membangun hubungan yang baik antara kampus dengan perusahaan serta mengetahui mengenai penerapan ilmu dan teknologi yang sesuai dengan bidang yang ditekuni selama ini.

Dalam penyusunan laporan Tugas Akhir ini, penulis banyak mendapat bantuan serta bimbingan dari berbagai pihak. Pada kesempatan ini tidak lupa penulis sampaikan ucapan terima kasih kepada:

1. Tuhan Yang Maha Esa atas rahmat serta kemudahan yang diberikan.
2. Kedua orang tua serta keluarga tercinta yang telah memberikan do'a, dukungan, pengorbanan, semangat dan kasih sayang hingga saat ini.
3. Bapak Dr. Mustofa, S.T., M.T., selaku Direktur Politeknik STMI Jakarta.
4. Ibu Noveriza Yuliasari, M.T., selaku Ketua Program Studi Sistem Informasi Industri Otomotif.
5. Ibu Triana Fatmawati, S.T., M.T., selaku dosen pembimbing yang telah memberikan arahan dan masukan dalam penyusunan laporan Tugas Akhir.
6. Bapak Drs. Lukman Djaja, M.B.A. selaku Presiden Direktur PT Multi Prima Sejahtera Tbk.
7. Bapak Sanggar Sadewo, Bapak Washadi, Bapak Sepurohim, Bapak Purwanto, Bapak Dzain Aldy Faisal, Bapak Renaldy, Bapak Ojan

Nasrulah dan seluruh pegawai di PT Multi Prima Sejahtera Tbk yang telah membantu memberi arahan serta informasi untuk pengumpulan data.

8. Seluruh dosen Politeknik STMI Jakarta yang telah memberikan ilmunya guna menambah pengetahuan dan pengalaman yang sangat bermanfaat.
9. Teman-teman mahasiswa/i Politeknik STMI Jakarta Program Studi Sistem Informasi Industri Otomotif terutama SA01 2015 atas kebersamaan, solidaritas, canda tawa dan motivasinya selama ini.
10. Teman-teman seperjuangan yang dinamakan Pejuang September atas semangat, kebersamaan, solidaritas dan canda tawa.
11. Teman-teman seperjuangan Warung Mbak Yati dan Warung Mpo yang selalu memberikan support dan solidaritas tanpa batas.

Semoga Tuhan Yang Maha Esa membalas kebaikan dan ketulusan semua pihak yang telah membantu menyelesaikan laporan ini dengan melimpahkan rahmat dan karunia-Nya. Besar harapan Penulis bahwa laporan ini dapat memberikan manfaat serta pengetahuan bagi pembacanya. Terima kasih.

Jakarta, 09 Agustus 2019

Penulis

## DAFTAR ISI

	Halaman
Halaman Judul.....	i
Lembar Persetujuan Dosen Pembimbing .....	ii
Lembar Bimbingan Dosen Pembimbing .....	iii
Lembar Pernyataan Keaslian.....	iv
Abstrak .....	v
Kata Pengantar .....	vi
Daftar Isi.....	viii
Daftar Gambar.....	xii
Daftar Tabel .....	xiv
Daftar Lampiran .....	xvi
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Pokok Permasalahan .....	2
1.3 Tujuan Penelitian .....	3
1.4 Batasan Masalah.....	3
1.5 Manfaat Tugas Akhir .....	4
1.6 Sistematika Penulisan.....	4
<b>BAB II LANDASAN TEORI.....</b>	<b>6</b>
2.1 Pengertian Rancang Bangun .....	6
2.2 Konsep Dasar Sistem .....	6
2.2.1 Tujuan Sistem .....	7
2.2.2 Karakteristik Sistem.....	7
2.3 Informasi .....	9
2.3.1 Pengelompokan Informasi .....	11
2.4 Konsep Dasar Sistem Informasi.....	11
2.5 Penerimaan Barang .....	11
2.6 Pengeluaran Barang .....	12

2.7	Bahan Baku .....	13
2.8	<i>Literature Review</i> .....	13
2.9	Pengertian Gudang .....	18
2.10	Busi ( <i>Spark Plug</i> ) .....	18
2.10.1	Tingkat Panas Busi.....	19
2.10.2	Kaitan Tingkat Panas Busi Dengan Konsumsi Bahan Bakar .....	19
2.10.3	Tujuan Pembuatan Busi Dengan Tingkat Panas Berbeda .....	19
2.11	<i>Systems Development Life Cycle</i> (SDLC).....	20
2.12	<i>System Requirements</i> .....	22
2.13	<i>Flowmap</i> .....	22
2.14	<i>Object Oriented Analysis and Design</i> .....	24
2.14.1	Prinsip Umum <i>Object Oriented Analysis and Design</i> ...	25
2.14.2	Karakteristik <i>Object Oriented Analysis and Design</i> .....	26
2.15	<i>UML (Unified Modeling Language)</i> .....	27
2.15.1	<i>Use Case Diagram</i> .....	28
2.15.2	<i>Activity Diagram</i> .....	30
2.15.3	<i>Sequence Diagram</i> .....	31
2.15.4	<i>Class Diagram</i> .....	33
2.15.5	<i>Deployment Diagram</i> .....	35
2.16	<i>Entity Relationship Diagram</i> (ERD).....	36
2.17	<i>Conceptual Data Model</i> (CDM) .....	38
2.18	Kamus Data .....	38
2.19	<i>Windows Navigation Diagram</i> (WND).....	39
2.20	<i>PHP (Hypertext Preprocessor)</i> .....	41
2.21	<i>Database</i> .....	41
2.22	<i>MySQL</i> .....	42
2.23	<i>CodeIgniter</i> .....	43
2.24	<i>Blackbox Testing</i> .....	45

<b>BAB III</b>	<b>METODOLOGI PENELITIAN .....</b>	<b>47</b>
3.1	Metodologi Penelitian .....	47
3.2	Jenis dan Sumber Data .....	47
3.3	Metode Pengumpulan Data .....	48
3.4	Metode Pengembangan Sistem .....	49
3.5	Kerangka Penelitian .....	50
<b>BAB IV</b>	<b>PENGUMPULAN DAN PENGOLAHAN DATA.....</b>	<b>56</b>
4.1	Sejarah Perusahaan.....	56
4.2	Profil Perusahaan .....	57
4.3	Visi dan Misi Perusahaan.....	58
4.4	Moto Karyawan.....	58
4.5	Kebijakan Mutu.....	60
4.6	Jam Kerja dan Jumlah Karyawan.....	61
4.7	Lokasi Perusahaan.....	62
4.8	Identitas Anak Perusahaan dan Asosiasi.....	63
4.9	Struktur Kepemilikan Saham .....	64
4.10	Struktur Organisasi .....	66
4.11	<i>Job Description</i> .....	68
4.12	Produk Perusahaan .....	69
4.13	Bahan Baku Perusahaan .....	72
4.14	<i>Layout Pabrik</i> .....	74
4.15	Daftar Pelanggan.....	75
4.16	Analisis Sistem Berjalan .....	77
4.16.1	Analisis Dokumen.....	78
4.16.2	Analisis Proses Bisnis .....	81
<b>BAB V</b>	<b>ANALISIS DAN PEMBAHASAN.....</b>	<b>84</b>
5.1	Analisis Kebutuhan Sistem .....	84
5.1.1	<i>Non Functional Requirement</i> .....	85
5.2	Urutan Prosedur dan <i>Flowmap</i> Usulan .....	86
5.3	Analisis Sistem Informasi Usulan .....	89
5.3.1	<i>Use Case Diagram</i> .....	89

5.3.1.1	Definisi Aktor .....	90
5.3.1.2	Definisi <i>Use Case</i> .....	91
5.3.2	<i>Activity Diagram</i> .....	95
5.3.3	<i>Sequence Diagram</i> .....	102
5.3.4	<i>Class Diagram</i> .....	110
5.3.5	<i>Deployment Diagram</i> .....	111
5.3.6	<i>Entity Relationship Diagram</i> (ERD) .....	112
5.3.7	<i>Conceptual Data Model</i> (CDM) .....	113
5.3.8	Kamus Data .....	114
5.4	Analisis Desain Sistem.....	118
5.4.1	<i>Windows Navigation Diagram</i> .....	118
5.4.2	Perancangan <i>Interface</i> Sistem.....	120
5.5	Pengujian Sistem.....	127
<b>BAB VI</b>	<b>KESIMPULAN DAN SARAN .....</b>	<b>128</b>
6.1	Kesimpulan .....	128
6.2	Saran.....	128
<b>Daftar Pustaka.....</b>		<b>129</b>
<b>Lampiran .....</b>		<b>132</b>

## DAFTAR GAMBAR

	Halaman
Gambar II.1 Model <i>Waterfall</i> .....	20
Gambar II.2 Contoh <i>Windows Navigation Diagram</i> .....	40
Gambar III.1 Kerangka Penelitian .....	54
Gambar IV.1 Logo Perusahaan PT Multi Prima Sejahtera Tbk.....	57
Gambar IV.2 Moto dan Budaya Karyawan PT Multi Prima Sejahtera Tbk ....	59
Gambar IV.3 Kebijakan Mutu PT Multi Prima Sejahtera Tbk .....	60
Gambar IV.4 Struktur Kepemilikan Saham PT Multi Prima Sejahtera Tbk....	65
Gambar IV.5 Struktur Organisasi Pabrik PT Multi Prima Sejahtera Tbk.....	67
Gambar IV.6 Struktur Organisasi Bagian Gudang.....	68
Gambar IV.7 <i>Layout</i> Pabrik PT Multi Prima Sejahtera Tbk.....	75
Gambar IV.8 Surat Jalan .....	78
Gambar IV.9 <i>Form</i> Laporan Penerimaan Barang .....	79
Gambar IV.10 Bon Permintaan Dan Penyerahan Barang .....	80
Gambar IV.11 <i>Flowmap</i> Berjalan .....	82
Gambar V.1 <i>Flowmap</i> Usulan .....	88
Gambar V.2 <i>Use Case Diagram</i> .....	89
Gambar V.3 <i>Activity Diagram</i> Proses <i>Login</i> .....	96
Gambar V.4 <i>Activity Diagram</i> Mengelola Data Master .....	97
Gambar V.5 <i>Activity Diagram</i> Menginput Data Surat Jalan .....	98
Gambar V.6 <i>Activity Diagram</i> Membuat Laporan Penerimaan.....	99
Gambar V.7 <i>Activity Diagram</i> Memberikan Persetujuan Surat Jalan .....	100
Gambar V.8 <i>Activity Diagram</i> Membuat Bon Permintaan dan Penyerahan	101
Gambar V.9 <i>Activity Diagram</i> Memberikan Persetujuan Bon Permintaan ..	102
Gambar V.10 <i>Sequence Diagram</i> Proses <i>Login</i> .....	103
Gambar V.11 <i>Sequence Diagram</i> Mengelola Data Bahan Baku.....	104
Gambar V.12 <i>Sequence Diagram</i> Mengelola Data Rak .....	105
Gambar V.13 <i>Sequence Diagram</i> Menginput Data Surat Jalan .....	106

Gambar V.14	<i>Sequence Diagram</i> Membuat Laporan Penerimaan.....	107
Gambar V.15	<i>Sequence Diagram</i> Memberikan Persetujuan Surat Jalan .....	108
Gambar V.16	<i>Sequence Diagram</i> Membuat Bon Permintaan .....	109
Gambar V.17	<i>Sequence Diagram</i> Memberikan Persetujuan Bon Permintaan	110
Gambar V.18	<i>Class Diagram</i> .....	111
Gambar V.19	<i>Deployment Diagram</i> .....	112
Gambar V.20	<i>Entity Relationship Diagram</i> .....	113
Gambar V.21	<i>Conceptual Data Model</i> .....	114
Gambar V.22	<i>Windows Navigation Diagram</i> .....	119
Gambar V.23	<i>Interface Form Login</i> .....	120
Gambar V.24	<i>Interface Halaman Utama</i> .....	121
Gambar V.25	<i>Interface Data Bahan Baku</i> .....	122
Gambar V.26	<i>Interface Data Rak</i> .....	123
Gambar V.27	<i>Interface Data Surat Jalan</i> .....	124
Gambar V.28	<i>Interface Data Surat Jalan Detail</i> .....	125
Gambar V.29	<i>Interface Data Permintaan</i> .....	126
Gambar V.30	<i>Interface Data Permintaan Detail</i> .....	127

## DAFTAR TABEL

	Halaman
Tabel II.1	<i>Literature Review</i> ..... 15
Tabel II.2	Simbol-Simbol <i>Flowmap</i> ..... 23
Tabel II.3	Simbol-Simbol <i>Use Case Diagram</i> ..... 28
Tabel II.4	Simbol-Simbol <i>Activity Diagram</i> ..... 30
Tabel II.5	Simbol-Simbol <i>Sequence Diagram</i> ..... 32
Tabel II.6	Simbol-Simbol <i>Class Diagram</i> ..... 34
Tabel II.7	Simbol-Simbol <i>Deployment Diagram</i> ..... 36
Tabel II.8	Simbol-Simbol <i>Entity Relationship Diagram</i> ..... 36
Tabel II.9	Simbol-Simbol <i>Conceptual Data Model</i> ..... 38
Tabel II.10	Simbol-Simbol Kamus Data ..... 39
Tabel II.11	Tipe Data MySQL ..... 43
Tabel IV.1	Jam Kerja Karyawan PT Multi Prima Sejahtera Tbk ..... 61
Tabel IV.2	Jumlah Karyawan PT Multi Prima Sejahtera Tbk ..... 61
Tabel IV.3	Tabel Pemegang Saham PT Multi Prima Sejahtera Tbk ..... 63
Tabel IV.4	Daftar Produk PT Multi Prima Sejahtera Tbk ..... 70
Tabel IV.5	Daftar Bahan Baku PT Multi Prima Sejahtera Tbk ..... 73
Tabel IV.6	Daftar Nama Pelanggan PT Multi Prima Sejahtera Tbk ..... 76
Tabel V.1	Analisis Kebutuhan Sistem ..... 84
Tabel V.2	Definisi Aktor ..... 90
Tabel V.3	<i>Use Case Login</i> ..... 91
Tabel V.4	<i>Use Case Mengelola Data Master</i> ..... 91
Tabel V.5	<i>Use Case Menginput Surat Jalan</i> ..... 92
Tabel V.6	<i>Use Case Membuat Laporan Penerimaan</i> ..... 93
Tabel V.7	<i>Use Case Memberikan Persetujuan Surat Jalan</i> ..... 94
Tabel V.8	<i>Use Case Membuat Bon Permintaan</i> ..... 94
Tabel V.9	<i>Use Case Memberikan Persetujuan Bon Permintaan</i> ..... 95
Tabel V.10	Tabel Data Bahan Baku ..... 115

Tabel V.11	Tabel Data <i>Supplier</i> .....	115
Tabel V.12	Tabel Data Karyawan.....	115
Tabel V.13	Tabel Data Rak.....	116
Tabel V.14	Tabel Data <i>Login</i> .....	116
Tabel V.15	Tabel Data Surat Jalan .....	116
Tabel V.16	Tabel Data Surat Jalan Detail.....	117
Tabel V.17	Tabel Data Permintaan .....	117
Tabel V.18	Tabel Data Permintaan Detail .....	118

## **DAFTAR LAMPIRAN**

	Halaman
Lampiran A Lembar Kegiatan .....	L-1
Lampiran B <i>Blackbox Testing</i> .....	L-3
Lampiran C Kode Program .....	L-14
Lampiran D Tampilan Antar Muka .....	L-31

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Dengan perkembangan teknologi yang semakin canggih ini, banyak perusahaan menerapkan teknologi informasi sebagai sarana untuk mempermudah pekerjaan. Dengan adanya kemajuan perkembangan teknologi menjadikan pengolahan data menjadi informasi semakin mudah. Pengolahan data dan informasi yang tersedia secara tepat, cepat dan akurat merupakan hal penting yang dibutuhkan dalam suatu perusahaan. Suatu perusahaan harus menerapkan sistem informasi yang dapat digunakan dalam proses jalannya perusahaan agar dapat bersaing dengan perusahaan lain.

Penerapan sistem informasi sangat diperlukan untuk meningkatkan kegiatan perusahaan terutama pada proses penerimaan dan pengeluaran bahan baku. Proses penerimaan dan pengeluaran bahan baku merupakan hal yang sangat perlu dipertimbangkan oleh suatu perusahaan, karena proses tersebut mempunyai peran yang sangat penting dalam pencapaian dan kesuksesan suatu perusahaan itu sendiri. Dengan adanya sistem informasi pada proses penerimaan dan pengeluaran bahan baku, perusahaan mempunyai stabilitas dalam mengontrol bahan baku yang diterima dan dikeluarkan di gudang itu sendiri.

Bahan baku adalah bahan yang digunakan dalam membuat produk dimana bahan tersebut secara menyeluruh tampak pada produk jadinya. Bahan baku terbagi menjadi dua bagian yaitu bahan baku langsung (*direct material*) dan bahan baku tidak langsung (*indirect material*). Bahan baku langsung merupakan bahan baku yang merupakan bagian suatu produk jadi. Sedangkan bahan baku tidak langsung adalah bahan baku yang ikut berperan dalam proses produksi tetapi tidak secara langsung tampak pada produk jadi yang dihasilkan.

PT Multi Prima Sejahtera Tbk merupakan perusahaan yang bergerak dalam bidang manufaktur. Kegiatan utama perusahaan adalah memproduksi busi (*spark plug*) untuk kendaraan bermotor baik mobil maupun motor. Pada saat ini

sistem informasi penerimaan dan pengeluaran bahan baku yang berjalan di PT Multi Prima Sejahtera Tbk masih dilakukan secara manual. Bagian Gudang bertugas untuk menerima bahan baku yang dikirim dari *supplier* dan mengeluarkan bahan baku untuk Bagian Produksi. Selain itu, Bagian Gudang juga bertugas untuk mengolah data barang masuk dan keluar gudang.

Pada saat ini, terdapat beberapa masalah yang ada pada proses penerimaan dan pengeluaran bahan baku di Bagian Gudang PT Multi Prima Sejahtera Tbk, seperti data yang terkait dengan proses penerimaan dan pengeluaran bahan baku masih disimpan dan diolah secara manual menggunakan *form-form* yang kemudian ditumpuk di dalam sebuah arsip penyimpanan. Hal ini dapat membuat data yang terkait dengan proses penerimaan dan pengeluaran bahan baku sulit dicari, mudah hilang dan rusak. Selain itu, belum adanya sistem informasi pada proses penerimaan dan pengeluaran bahan baku menyebabkan karyawan sulit memperoleh informasi mengenai data bahan baku yang diterima dan dikeluarkan sehingga proses penerimaan dan pengeluaran bahan baku membutuhkan waktu yang cukup lama.

Untuk meningkatkan kinerja dalam proses penerimaan dan pengeluaran bahan baku, maka perlu diterapkan sistem yang terkomputerisasi dengan cara merancang sistem informasi menggunakan basis data yang baik. Penelitian Tugas Akhir ini akan menganalisis dan merancang sistem informasi penerimaan dan pengeluaran bahan baku untuk memudahkan Bagian Gudang memperoleh informasi mengenai data bahan baku yang diterima dan dikeluarkan secara cepat dan tepat. Adapun judul Tugas Akhir ini adalah “Rancang Bangun Sistem Informasi Penerimaan Dan Pengeluaran Bahan Baku Pada Bagian Gudang Di PT Multi Prima Sejahtera Tbk Menggunakan CodeIgniter Dan MySQL”.

## 1.2 Pokok Permasalahan

Berdasarkan latar belakang di atas, maka penulis dapat mengidentifikasi permasalahan yang terjadi di Bagian Gudang adalah sebagai berikut:

1. Data yang terkait dengan proses penerimaan dan pengeluaran bahan baku masih disimpan dan diolah secara manual menggunakan *form-form*

yang kemudian ditumpuk di dalam sebuah arsip penyimpanan. Hal ini dapat membuat data yang terkait dengan proses penerimaan dan pengeluaran bahan baku sulit dicari, mudah hilang dan rusak.

2. Belum adanya sistem informasi pada proses penerimaan dan pengeluaran bahan baku menyebabkan karyawan sulit memperoleh informasi mengenai data bahan baku yang diterima dan dikeluarkan sehingga proses penerimaan dan pengeluaran bahan baku membutuhkan waktu yang cukup lama.

### **1.3 Tujuan Penelitian**

Adapun tujuan dari penulisan Tugas Akhir ini adalah sebagai berikut:

1. Menganalisis, merancang dan membangun sistem informasi penerimaan dan pengeluaran bahan baku dengan media penyimpanan data dalam bentuk *database* agar dapat menyimpan data lebih aman dan terstruktur.
2. Menganalisis, merancang dan membangun sistem informasi penerimaan dan pengeluaran bahan baku yang dapat memberikan kemudahan pengolahan data bahan baku yang diterima dan dikeluarkan sehingga proses penerimaan dan pengeluaran bahan baku menjadi lebih optimal.

### **1.4 Batasan Masalah**

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

1. Penelitian ini membahas manajemen gudang bahan baku khususnya sistem informasi penerimaan dan pengeluaran bahan baku.
2. Penelitian dilakukan pada Bagian Gudang selama satu bulan mulai dari 8 Agustus sampai dengan 7 September 2018.
3. Penelitian hanya sebatas mengenai penerimaan bahan baku dari *supplier* dan pengeluaran bahan baku ke Bagian Produksi yang terjadi pada Bagian Gudang.
4. Penelitian tidak membahas mengenai kualitas bahan baku, permintaan pembelian bahan baku dan pengembalian bahan baku.

## **1.5 Manfaat Tugas Akhir**

Adapun manfaat bagi perusahaan yang dapat diambil dari penelitian Tugas Akhir ini yaitu memiliki aplikasi penerimaan dan pengeluaran bahan baku yang sudah berbasis *web*, agar dapat membantu perusahaan dalam proses penerimaan dan pengeluaran bahan baku.

## **1.6 Sistematika Penulisan**

Penulisan Tugas Akhir ini disusun berdasarkan hal-hal yang berhubungan erat dengan hasil pengamatan sehingga dapat memperoleh gambaran yang lebih jelas mengenai Tugas Akhir yang dilaksanakan. Adapun tahapan-tahapan dalam penulisan laporan ini adalah sebagai berikut:

### **BAB I PENDAHULUAN**

Pada bab ini memuat latar belakang, pokok permasalahan, tujuan penelitian, batasan masalah, manfaat penelitian dan sistematika penulisan.

### **BAB II LANDASAN TEORI**

Pada bab ini membahas tentang berbagai teori yang diperoleh dari buku-buku literatur ataupun berbagai macam referensi yang berkaitan dengan tema yang diambil. Selain itu, pada bab ini juga dibahas tentang penelitian terdahulu yang berkaitan dengan topik penelitian ini. Teori-teori yang dipaparkan pada laporan ini adalah rancang bangun, sistem informasi, konsep dasar penerimaan dan pengeluaran bahan baku serta *tools* pemodelan sistem khususnya *Unified Modeling Language* (UML), *Hypertext Preprocessor* (PHP), *CodeIgniter* dan MySQL.

### **BAB III METODOLOGI PENELITIAN**

Pada bab ini membahas tentang langkah-langkah atau tahapan yang akan dilakukan dalam pemecahan masalah termasuk metodologi pengembangan sistem yang digunakan.

**BAB IV PENGUMPULAN DAN PENGOLAHAN DATA**

Pada bab ini membahas tentang data yang telah diperoleh berdasarkan pengamatan di PT Multi Prima Sejahtera Tbk terutama dalam proses penerimaan dan pengeluaran bahan baku meliputi proses bisnis sistem yang berjalan dan dokumen yang terlibat pada sistem berjalan.

**BAB V ANALISIS DAN PEMBAHASAN**

Pada bab ini berisi analisis rinci dari pengolahan data, yakni mulai dari analisis sistem yang meliputi diagram alir sistem usulan, pemodelan sistem dengan *Unified Modelling Language* (UML), pemodelan basis data dengan *Entity Relationship Diagram* (ERD), *Conceptual Data Model* (CDM), kamus data, perancangan antar muka, perancangan menu dengan *Windows Navigation Diagram* (WND) dan pembuatan spesifikasi sistem yang diperlukan untuk implementasi sistem.

**BAB VI KESIMPULAN DAN SARAN**

Pada bab ini diuraikan beberapa kesimpulan dari hasil penulisan Tugas Akhir dan saran untuk penerapan sistem informasi penerimaan dan pengeluaran bahan baku untuk perusahaan dan pengembangan penelitian selanjutnya.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Pengertian Rancang Bangun**

Rancang merupakan serangkaian prosedur untuk menerjemahkan hasil analisis dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan. Rancangan sistem adalah penentuan proses dan data yang diperlukan oleh sistem baru (McLeod, 2007).

Menurut Pressman (2015), perancangan atau rancang merupakan serangkaian prosedur untuk menterjemahkan hasil analisa dan sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem di implementasikan. Menurut Pressman (2015), pengertian pembangunan atau bangun sistem adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada secara keseluruhan.

#### **2.2 Konsep Dasar Sistem**

Terdapat dua kelompok pendekatan di dalam pendefinisian sistem, yaitu kelompok yang menekankan pada prosedur dan kelompok yang menekankan pada elemen atau komponennya. Pendekatan yang menekankan pada prosedur mendefinisikan sistem sebagai suatu jaringan kerja prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Sedangkan pendekatan sistem yang lebih menekankan pada elemen atau komponen mendefinisikan sistem sebagai kumpulan elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Kedua kelompok definisi ini adalah benar dan tidak bertentangan (Sutabri, 2012).

Banyak ahli mengajukan konsep sistem dengan deskripsi berbeda tetapi prinsipnya hampir sama dengan konsep dasar sistem umumnya. Secara ringkas menjelaskan bahwa sistem adalah (Fatta, 2017):

1. Komponen-komponen yang saling berhubungan satu sama lain.
2. Suatu keseluruhan tanpa memisahkan komponen pembentuknya.
3. Bersama-sama dalam mencapai tujuan.
4. Memiliki *input* dan *output* yang dibutuhkan oleh sistem lainnya.
5. Terdapat proses yang mengubah *input* menjadi *output*.
6. Menunjukkan adanya entropi.
7. Memiliki aturan.
8. Memiliki subsistem yang lebih kecil.
9. Memiliki deferensiasi antar subsistem.
10. Memiliki tujuan yang sama meskipun mulai berbeda.

### **2.2.1 Tujuan Sistem**

Susanto (2013) mengemukakan bahwa tujuan sistem merupakan target atau sasaran akhir yang ingin dicapai oleh suatu sistem. Agar supaya target tersebut bisa tercapai, maka target atau sasaran tersebut harus diketahui terlebih dahulu ciri-ciri atau kriterianya. Upaya mencapai suatu sasaran tanpa mengetahui ciri-ciri atau kriteria dari sasaran tersebut kemungkinan besar sasaran tersebut tidak akan pernah tercapai. Ciri-ciri atau kriteria dapat juga digunakan sebagai tolak ukur dalam menilai suatu keberhasilan suatu sistem dan menjadi dasar dilakukannya suatu pengendalian.

### **2.2.2 Karakteristik Sistem**

Menurut Sutabri (2012), dalam buku yang berjudul Analisis Sistem Informasi, model umum sebuah sistem terdiri dari *input*, proses, dan *output*. Hal ini merupakan konsep sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat beberapa masukan dan keluaran sekaligus. Adapun karakteristik yang dimaksud adalah sebagai berikut:

1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Suatu sistem dapat mempunyai sistem yang lebih besar yang disebut dengan supra sistem.

2. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.

3. Lingkungan Luar Sistem (*Environtment*)

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem. Lingkungan luar sistem yang menguntungkan merupakan energi bagi sistem tersebut, yang dengan demikian lingkungan luat tersebut harus selalu dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus dikendalikan.

4. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsitem ke subsistem yang lain. Keluaran suatu subsistem akan menjadi masukan untuk subsistem yang lain dengan melewati penghubung.

5. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Sebagai contoh, di dalam suatu unit sistem komputer, “program” adalah *maintenance input* yang digunakan untuk mengoperasikan komputer. Sementara “data” adalah *signal input* yang akan diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain. Contoh sistem informasi, keluaran yang dihasilkan adalah informasi yang mana informasi ini dapat digunakan sebagai masukan untuk

pengambilan keputusan atau hal-hal lain yang merupakan *input* bagi subsistem lainnya.

#### 7. Pengolah Sistem (Proses)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran. Sebagai contoh, sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

#### 8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat *deterministic*. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

### 2.3 Informasi

Informasi merupakan data yang sudah diolah yang ditujukan untuk seseorang, organisasi ataupun siapa saja yang membutuhkan. Informasi akan menjadi berguna apabila objek yang menerima informasi membutuhkan informasi tersebut (Sutabri, 2012). Romey & Steinbart (2015) mengemukakan kriteria informasi yang baik sebagai berikut:

#### 1. Relevan

Informasi yang relevan adalah informasi yang:

##### a. Memiliki umpan balik

Informasi memungkinkan pengguna untuk menegaskan atau mengoreksi ekspektasi mereka di masa lalu.

##### b. Memiliki manfaat prediktif

Informasi dapat membantu pengguna untuk memprediksi masa yang akan datang berdasarkan hasil masa lalu dan kejadian masa kini.

#### 2. Andal

Informasi harus bebas dari pengertian yang menyesatkan dan kesalahan material, menyajikan setiap fakta secara jujur. Informasi yang akurat memenuhi karakteristik:

a. Penyajian jujur

Informasi menggambarkan dengan jujur transaksi serta peristiwa lainnya yang seharusnya disajikan atau yang secara wajar dapat diharapkan untuk disajikan.

b. Netralitas

Informasi diarahkan pada kebutuhan umum dan tidak berpihak pada kebutuhan pihak tertentu.

3. Lengkap

Informasi disajikan selengkap mungkin, yaitu mencakup semua informasi yang dapat mempengaruhi pengambilan keputusan. Informasi yang lengkap memenuhi karakteristik:

a. Disajikan dengan lengkap.

b. Sesuai dengan ketentuan dan kebutuhan.

4. Tepat waktu

Informasi disajikan tepat waktu sehingga dapat berpengaruh dan berguna dalam pengambilan keputusan. Informasi yang tepat waktu memenuhi karakteristik:

a. Tersedia pada saat dibutuhkan.

b. Informasi yang disajikan terbaru.

5. Dapat dipahami

Informasi yang disajikan dalam informasi manajemen keuangan dinyatakan dalam bentuk serta istilah yang disesuaikan dengan batas pemahaman para pengguna.

6. Dapat diverifikasi

Informasi yang disajikan dalam informasi manajemen keuangan dapat diuji, dan apabila pengujian dilakukan lebih dari sekali oleh pihak yang berbeda, hasilnya tetap menunjukkan simpulan yang tidak berbeda jauh.

7. Dapat diakses

Informasi tersedia pada saat dibutuhkan dan dengan format yang dapat digunakan.

### **2.3.1 Pengelompokan Informasi**

Sutabri (2012) mengemukakan bahwa informasi merupakan proses lebih lanjut dari data yang sudah memiliki nilai tambah. Informasi dapat dikelompokkan menjadi 3 bagian, yaitu:

- 1. Informasi Strategis**

Informasi ini digunakan untuk mengambil keputusan jangka panjang, yang mencakup informasi eksternal, rencana perluasan perusahaan, dan sebagainya.

- 2. Informasi Taktis**

Informasi ini dibutuhkan untuk mengambil keputusan jangka menengah, seperti informasi tren penjualan yang dapat dimanfaatkan untuk menyusun rencana penjualan.

- 3. Informasi Teknis**

Informasi ini dibutuhkan untuk keperluan operasional sehari-hari, seperti informasi persediaan stok, retur penjualan dan laporan kas harian.

### **2.4 Konsep Dasar Sistem Informasi**

Sutabri (2012) mengemukakan bahwa sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu.

Pratama (2014) mengemukakan bahwa sistem informasi merupakan gabungan dari empat bagian utama yaitu perangkat lunak (*software*), perangkat keras (*hardware*), infrastruktur dan Sumber Daya Manusia (SDM) yang terlatih. Keempat bagian utama ini saling berkaitan untuk menciptakan sebuah sistem yang dapat mengolah sebuah data menjadi informasi yang bermanfaat.

### **2.5 Penerimaan Barang**

Menurut Damiri (2004) tentang penerimaan barang dalam sebuah organisasi atau sebuah perusahaan, sesi ini biasanya ada di bawah organisasi *accounting* sama halnya dengan bagian gudang. Pemegang jabatannya, biasanya

setingkat dengan *supervisor*, mengingat tanggung jawabnya terhadap perusahaan cukup besar dan penting. Seksi ini disebut Penerimaan Barang (*Receiving*). Tugas dan tanggung jawab awalnya adalah menerima barang-barang komoditas untuk operasional perusahaan yang masuk melalui pembelian oleh Departemen pembelian sesuai dengan pesanan, meliputi harga, kualitas dan kuantitas. Barang-barang yang secara resmi diterima oleh penerima barang, dibuatkan tanda terima dan selanjutnya diserahkan kepada bagian (departemen) pemesanan (*requested*).

Untuk penerimaan barang-barang komoditas yang kriterianya kurang diketahui dan untuk tidak terjadi kesalahan, maka dia dapat meminta bantuan kepada pemesan yang diberi wewenang serta bertanggung jawab untuk kedatangan barang-barang tersebut. Ada 4 hal utama di bawah ini yang harus diketahui oleh seorang bagian penerimaan barang, yaitu (Damiri, 2004):

1. Kuantitas barang yang diterima harus sama dengan yang tertera pada daftar *slip* penyerahan (faktur, *invoice*) serta sama dengan yang tercantum dengan *order* pembelian (*request*).
2. Kualitas barang yang diterima haruslah sama dengan yang ada dalam spesifikasi pembelian yang standar.
3. Harga dalam *slip* penyerahan (faktur, *invoice*) haruslah sama dengan yang tercantum dalam *order* pembelian (*request*) yaitu ditetapkan oleh Departemen Pembelian.
4. Pemasok dalam faktur penyerahan haruslah sama dengan yang tercantum dalam *order* yang diterima dari Departemen Pembelian.

## 2.6 Pengeluaran Barang

Menurut Arianti (2003) sistem pengeluaran barang digunakan dalam perusahaan untuk mengeluarkan barang yang dibutuhkan perusahaan dalam proses produksi. Fungsi gudang mengeluarkan barang sesuai dengan daftar kebutuhan barang yang telah disetujui dan diotorisasi oleh kepala bagian gudang. Penentuan barang yang dikeluarkan sesuai dengan daftar kebutuhan barang. Otorisasi dari setiap adanya pengeluaran barang. Fungsi gudang tidak hanya berhenti pada pembuatan dokumen penyerahan barang saja, akan tetapi juga

meliputi pembuatan laporan pengeluaran gudang untuk kepentingan manajemen. Bagian yang terkait dalam transaksi pengeluaran barang dilaksanakan oleh bagian–bagian yang terpisah, yaitu: bagian gudang, bagian produksi, dan bagian akuntansi.

## 2.7 Bahan Baku

Menurut Hanggana (2006) pengertian bahan baku adalah sesuatu yang digunakan untuk membuat barang jadi, bahan pasti menempel menjadi satu dengan barang jadi. Dalam sebuah perusahaan bahan baku dan bahan penolong memiliki arti yang sangat penting, karena menjadi modal terjadinya proses produksi sampai hasil produksi.

Nasution & Prasetyawan (2008) mengemukakan bahwa bahan baku adalah barang-barang yang dibeli dari pemasok (*supplier*) dan akan digunakan atau diolah menjadi produk jadi yang akan dihasilkan oleh perusahaan. Thontowie (2011) menyebutkan bahwa bahan baku yang digunakan dalam proses produksi dapat dibedakan menjadi dua yaitu:

1. Bahan baku langsung (*direct material*), merupakan bahan baku utama yang mempengaruhi proses produksi. Contoh produsen mobil: baja, besi alumunium.
2. Bahan baku tidak langsung (*indirect material*), merupakan bahan pembantu yang diperlukan untuk memperlancar proses produksi. Contoh produsen mobil: bor, oli.

## 2.8 *Literature Review*

*Literature review* adalah uraian tentang teori, temuan dan bahan penelitian lainnya yang diperoleh dari bahan acuan untuk dijadikan landasan kegiatan penelitian untuk menyusun kerangka pemikiran yang jelas dari perumusan masalah yang ingin diteliti. *Literature review* memiliki peran penting dalam membuat suatu tulisan ataupun karangan ilmiah, karena dapat memberikan ide dan tujuan tentang topic penelitian yang akan dilakukan. Pada umumnya berisi ulasan, rangkuman & pemikiran penulis tentang beberapa pustaka (buku, jurnal,

majalah) yang berkaitan dengan topik yang dibahas. Menurut Wahono (2016), yang disebut dengan literatur ilmiah dapat berupa:

1. *Paper* dari Jurnal Ilmiah
2. *Paper* dari *Conference (Proceedings)*
3. *Thesis* dan Disertasi
4. *Report* (laporan) dari Organisasi yang Terpercaya
5. Buku *Textbook*

Tabel II.1 berikut ini adalah *literature review* dari beberapa jurnal yang telah dianalisis perbedaan dan persamaan yang ada pada beberapa jurnal:

Tabel II.1 *Literature Review*

No	Peneliti, Tahun	Topik dan Objek Penelitian	Permasalahan	Kesimpulan Penelitian	Hasil Review
1	Junaidi, Retno Setianingsih dan Khusnul Khotimah, 2015	Rancang Bangun Sistem Penerimaan dan Pengeluaran Barang Menggunakan Java Aplikasi	Pencatatan dalam transaksi baik permintaan, penerimaan maupun pengeluaran barang masih menggunakan cara manual. Sehingga, dokumen yang berkaitan mudah hilang, rusak dan pembuatan laporan masih kurang efektif.	Sistem aplikasi dapat memberikan kemudahan dalam melakukan transaksi, pencatatan stok pesediaan barang serta sangat membantu dalam hal pembuatan laporan yang bertujuan untuk mempermudah mendapatkan informasi, karena sistem aplikasi terletak pada proses <i>input</i> dan <i>output</i> .	Persamaan dari hasil <i>review</i> penelitian dengan laporan penelitian ini yaitu kesamaan dalam membahas rancang bangun sistem penerimaan barang dan menggunakan MySql dalam pembuatan program.  Perbedaan dari hasil <i>review</i> penelitian yang dilakukan dengan laporan penelitian ini adalah program yang digunakan menggunakan Netbeans sedangkan dalam penelitian ini menggunakan <i>Framework CodeIgniter</i> .

Tabel II.1 *Literature Review* (lanjutan)

No	Peneliti, Tahun	Topik dan Objek Penelitian	Permasalahan	Kesimpulan Penelitian	Hasil Review
2	Abdur Rochman dan Yanti Hardiyanti, 2016	Rancangan Sistem Informasi Penerimaan Barang Jadi Pada PT Shyang Yao Fung Dengan Metodologi Berorientasi Objek	Perhitungan jumlah barang masih dilakukan secara manual sehingga memerlukan proses yang cukup lama.	Merancang dan membangun sebuah sistem baru menggunakan metode elisitasi dan merancang diagram dengan menggunakan UML ( <i>Unified Modeling Language</i> ) yang bertujuan untuk mencegah terjadinya kesalahan <i>input</i> data dari proses manual sebelumnya.	Persamaan dari hasil <i>review</i> penelitian yang dilakukan oleh Abdur Rochman dan Yanti Hardiyanti dengan laporan penelitian ini adalah adanya kesamaan dalam pembahasan rancangan sistem informasi penerimaan barang dan menggunakan UML ( <i>Unified Modeling Language</i> ) serta menggunakan MySQL sebagai <i>database</i> .  Perbedaan dari hasil <i>review</i> penelitian yang dilakukan oleh Abdur Rochman dan Yanti Hardiyanti dengan laporan penelitian ini adalah pembahasan, dimana pada penelitian tersebut membahas tentang penerimaan barang jadi, sedangkan pada laporan penelitian ini membahas tentang penerimaan bahan baku yang belum menggunakan metode tertentu untuk pengelolaan penyimpanan barang.

Tabel II.1 *Literature Review* (lanjutan)

No	Peneliti, Tahun	Topik dan Objek Penelitian	Permasalahan	Kesimpulan Penelitian	Hasil Review
3	Auliya Rahman, 2016	Analisis dan Implementasi Sistem Aplikasi <i>Inventory Items</i> Berbasis <i>Web</i> Menggunakan <i>Framewrok CodeIgniter</i>	Pencatatan inventaris barang masih dibuat dengan cara manual. Sehingga membutuhkan waktu yang lama dalam penulisan laporan.	Dengan adanya aplikasi ini dapat memberikan kelebihan dan kemudahan serta dapat membantu meringankan pekerjaan karyawan mencari data barang inventaris dalam pembuatan laporan.	Persamaan dari hasil <i>review</i> penelitian yang dilakukan oleh Auliya Rahman dengan laporan penelitian ini adalah adanya kesamaan dalam menggunakan <i>Framework CodeIgniter</i> . Perbedaan dari hasil <i>review</i> penelitian yang dilakukan oleh Auliya Rahman dengan laporan penelitian ini adalah dalam topik pembahasan, yaitu pada penelitian tersebut membahas tentang persediaan barang sedangkan pada penelitian ini membahas tentang penerimaan barang.

(Sumber: Hasil Analisis, 2019)

## 2.9 Pengertian Gudang

Definisi gudang menurut Lambert (2001) adalah bagian dari sistem logistik perusahaan yang menyimpan produk-produk (*raw material, parts, goods-in-process, finished goods*) pada dan antara titik sumber (*point-of-origin*) dan titik konsumsi (*point-of-consumption*) dan menyediakan informasi kepada manajemen mengenai status, kondisi dan disposisi dari item-item yang disimpan.

Tujuan dari adanya tempat penyimpanan dan fungsi dari pergudangan secara umum adalah memaksimalkan penggunaan sumber-sumber yang ada disamping memaksimalkan pelayanan terhadap pelanggan dengan sumber yang terbatas. Sumber daya gudang dan pergudangan adalah ruangan, peralatan dan personil. Pelanggan membutuhkan gudang dan fungsi pergudangan untuk dapat memperoleh barang yang diinginkan secara tepat dan dalam kondisi yang baik. Maka dalam perancangan gudang dan sistem pergudangan diperlukan untuk hal-hal berikut ini (Purnomo, 2004):

1. Memaksimalkan penggunaan ruang.
2. Memaksimalkan menggunakan peralatan.
3. Memaksimalkan penggunaan tenaga kerja.
4. Memaksimalkan kemudahan dalam penerimaan seluruh material dan penerimaan barang.

## 2.10 Busi (*Spark Plug*)

Busi berfungsi untuk menghasilkan cetus listrik yang diperlukan untuk membakar gas (uap bensin dan udara dalam silinder pada akhir kompresi). Menurut Daryanto (2003), secara garis besar busi dibedakan menjadi dua jenis, yaitu:

1. Busi Dingin

Busi dingin adalah busi yang menyerap serta membuang atau melepaskan panas dengan cepat. Jenis ini biasanya dipakai untuk mesin yang temperatur dalam ruang bakarnya tinggi. Busi dingin memiliki nilai panas lebih tinggi yaitu 8-12.

## 2. Busi Panas

Busi panas adalah busi yang menyerap serta membuang atau melepaskan panas dengan lambat. Jenis ini hanya digunakan untuk mesin yang temperatur dalam ruang bakarnya rendah. Busi panas memiliki nilai panas lebih rendah yaitu 2-7.

### 2.10.1 Tingkat Panas Busi

Menurut Daryanto (2003), tingkat panas busi menunjukkan sampai berapa tinggi panasnya busi dapat bekerja. Tanpa adanya tingkat panas pada busi yang disesuaikan dengan keadaan kerja mesin maka akan terjadi endapan karbon pada elektroda-elektroda yang dapat mengurangi loncatan api dan tenaga mesin menjadi rendah, selain itu busi-busi kemungkinan dapat memijar dan menyebabkan terjadinya pembakaran pendahuluan, yang mana campuran akan terbakar sebelum busi mengeluarkan loncatan api.

### 2.10.2 Kaitan Tingkat Panas Busi Dengan Konsumsi Bahan Bakar

Bila nilai panas dari sebuah busi tinggi maka percikan bunga api yang dihasilkan akan lebih besar. Bila busi yang digunakan dengan tingkat panas lebih rendah (busi panas) maka suhu di dalam ruang bakar juga menjadi rendah sehingga busi tidak mampu untuk membakar campuran bahan bakar dan udara akibat suhu mesin tidak ideal (Denso, 2008).

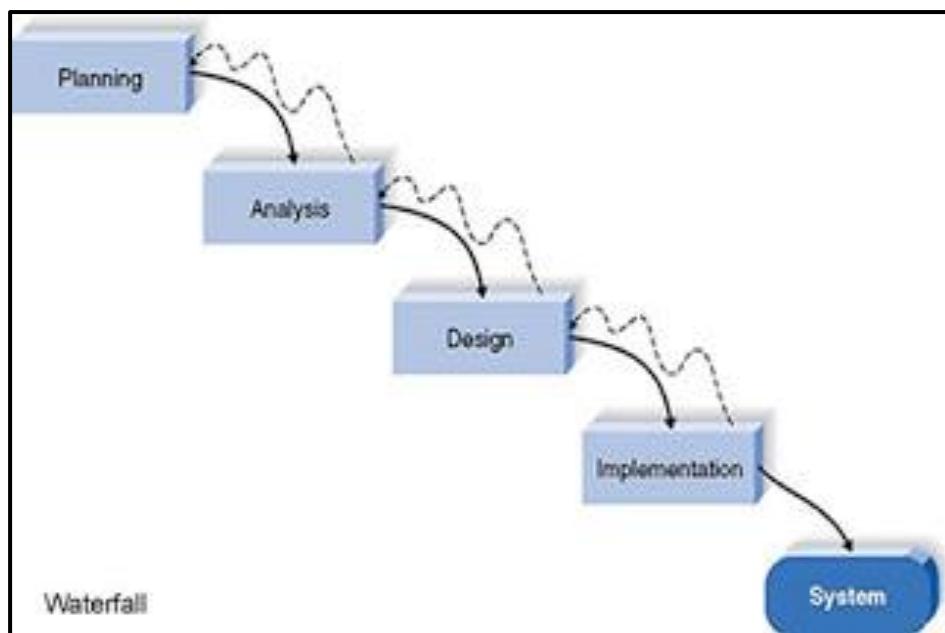
### 2.10.3 Tujuan Pembuatan Busi Dengan Tingkat Panas Berbeda

Menurut Daryanto (2003), busi pada umumnya direncanakan menurut keadaan panas dan temperatur di dalam ruang bakar mesin. Busi pada umumnya diciptakan dengan tingkat panas yang berbeda bertujuan untuk menyesuaikan dengan keadaan kerja mesin. Busi dingin diciptakan untuk kendaraan yang temperatur ruang bakarnya tinggi, sedangkan pada busi panas diciptakan untuk kendaraan yang temperatur ruang bakarnya lebih rendah.

## 2.11 Systems Development Life Cycle (SDLC)

Menurut Dennis (2015) siklus hidup pengembangan sistem adalah proses memahami bagaimana suatu sistem informasi dapat mendukung kebutuhan bisnis dengan merancang sistem, membangunnya dan memberikannya kepada *user*.

Metode pengembangan sistem dengan model *waterfall*, menurut Pressman (2015), model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Nama model ini sebenarnya adalah “*linear sequential model*”. Model ini sering disebut juga dengan “*classic life cycle*” atau metode *waterfall*. Model ini termasuk ke dalam model generik pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Gambar II.1 berikut ini adalah langkah-langkah yang ada dalam metode *waterfall*.



Gambar II.1 Model *Waterfall*  
(Sumber: Dennis, Wixom & Tegarden, 2015)

Berikut ini adalah tahapan-tahapan yang harus dilalui pada metode *waterfall*:

### 1. Perencanaan (*Planning*)

Dalam tahapan ini, menjelaskan dan mengargumentasikan untuk melanjutkan proyek yang telah dipilih, rencana kerja yang matang juga disusun untuk menjalankan tahapan-tahapan lainnya. Pada tahap ini ditentukan secara detail rencana kerja yang harus dikerjakan, durasi yang diperlukan masing-masing tahap, sumber daya manusia, perangkat lunak, dokumentasi, perangkat keras, maupun finansial diestimasi. Kesalahan pada tahap ini akan mengakibatkan keuntungan yang diperolah tidak maksimal, bahkan bisa rugi. Pada tahapan ini peran manajemen sistem informasi berpengalaman sangat dibutuhkan.

### 2. Analisis (*Analysis*)

Tahap kedua, adalah tahap analisis, yaitu berusaha mengenali segenap permasalahan yang muncul pada pengguna dengan mendekomposisi dan merealisasikan komponen-komponen sistem. Tujuan utama dari tahap analisis adalah untuk memahami dan mendokumentasikan kebutuhan bisnis dan persyaratan proses dari sistem baru. Menganalisa kebutuhan sebagai bahan dalam membuat spesifikasi di tahapan selanjutnya.

### 3. Perancangan (*Design*)

Tahap Perancangan (*design*), mencoba mencari solusi permasalahan yang didapat dari tahap analisis. Tahapan mengubah kebutuhan yang masih berupa konsep menjadi spesifikasi sistem yang nyata untuk diimplementasikan. Jika pada tahapan analisis *form requirement to specification*, maka tahapan desain adalah *form specification to implementation*.

### 4. Implementasi (*Implementation*)

Tahap implementasi, dimana mengimplementasikan perancangan sistem ke situasi nyata atau desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca. Disini mulai berurusan dengan pemilihan perangkat keras dan penyusunan perangkat lunak aplikasi (pengkodean/*coding*).

## 5. Sistem

Pada tahapan sistem dilakukan pengujian (*testing*) dan pemeliharaan, yang dapat digunakan untuk menentukan apakah sistem/perangkat lunak yang dibuat sudah sesuai dengan kebutuhan pengguna atau belum. Jika belum, proses selanjutnya adalah bersifat *iterative*, yaitu kembali ke tahap sebelumnya. Tahap pemeliharaan dan perawatan dimana mulai melakukan pengoperasian sistem dan jika diperlukan melakukan perbaikan-perbaikan kecil. Kemudian jika waktu penggunaan sistem habis, maka akan masuk lagi pada tahap perencanaan (*design*).

### 2.12 *System Requirements*

*System requirements* atau kebutuhan sistem adalah semua aktivitas yang harus dilakukan atau didukung oleh sistem baru dan batasan-batasan yang harus dicapai sistem baru (Satzinger, 2010). *System requirements* dibagi menjadi 2 kategori, yaitu:

1. *Functional requirements* atau kebutuhan fungsional adalah aktivitas yang harus dilakukan oleh sistem dengan contoh-contoh sebagai berikut:
  - a. Sistem dapat melakukan input pendataan buku.
  - b. Sistem dapat melakukan input pendataan anggota.
  - c. Sistem dapat melakukan transaksi peminjaman.
2. *Non-functional requirements* atau kebutuhan non-fungsional adalah karakteristik dari sistem selain aktivitas yang harus dilakukan atau didukung dengan contoh-contoh sebagai berikut:
  - a. Sistem dapat dijalankan oleh beberapa *software web browser* diantaranya Internet Explore, Google Chrome dan Mozilla Firefox.
  - b. Sistem harus dapat memastikan bahwa data yang digunakan dalam sistem harus terlindung dari akses yang tidak berwenang.
  - c. Besarnya program dari sistem maksimal sebesar 100 MB.

### 2.13 *Flowmap*

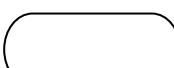
Menurut Jogiyanto (2005) *flowmap* atau bagan alir adalah bagan yang menunjukkan aliran di dalam program atau prosedur sistem secara logika.

*Flowmap* ini berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *flowmap* ini harus dapat memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi. Adapun pedoman-pedoman dalam pembuatan *flowmap* adalah sebagai berikut:

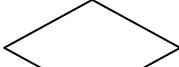
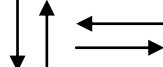
1. *Flowmap* sebaiknya digambarkan dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
2. Kegiatan di dalam *flowmap* harus ditunjukkan dengan jelas.
3. Harus ditunjukkan dari mana kegiatan akan dimulai dan dimana akan berakhir.
4. Masing-masing kegiatan di dalam *flowmap* sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan.
5. Masing-masing kegiatan di dalam *flowmap* harus di dalam urutan yang semestinya.
6. Kegiatan yang terpotong dan akan disambung ditempat lain harus ditunjukkan dengan jelas menggunakan simbol penghubung.

Adapun simbol-simbol yang sering digunakan dalam *flowmap* dapat dilihat pada tabel II.2 berikut ini:

Tabel II.2 Simbol-Simbol *Flowmap*

Simbol	Keterangan
 Terminal	<ul style="list-style-type: none"> <li>• Menyatakan permulaan atau akhir suatu program.</li> </ul>
 Input-Output	<ul style="list-style-type: none"> <li>• Menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.</li> </ul>
 Process	<ul style="list-style-type: none"> <li>• Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.</li> </ul>

Tabel II.2 Simbol-Simbol *Flowmap* (lanjutan)

Simbol	Keterangan
 <i>Decision</i>	<ul style="list-style-type: none"> <li>Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya atau tidak.</li> </ul>
 <i>Arsip</i>	<ul style="list-style-type: none"> <li>Menunjukkan pengarsipan data.</li> </ul>
 <i>Manual Process</i>	<ul style="list-style-type: none"> <li>Menunjukkan pengolahan yang tidak dilakukan oleh komputer/pekerjaan manual.</li> </ul>
 <i>Predafined Process</i>	<ul style="list-style-type: none"> <li>Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.</li> </ul>
 <i>Document</i>	<ul style="list-style-type: none"> <li>Menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau <i>output</i> dicetak di kertas.</li> </ul>
 <i>Display</i>	<ul style="list-style-type: none"> <li>Menampilkan ke monitor.</li> </ul>
 <i>Flow</i>	<ul style="list-style-type: none"> <li>Menyatakan jalannya arus suatu proses.</li> </ul>

(Sumber: Jogyianto, 2005)

## 2.14 Object Oriented Analysis and Design

Menurut Mathiassen (2000) *Object Oriented Analysis and Design* (OOAD) adalah metode untuk menganalisa dan merancang sistem dengan pendekatan berorientasi *object*. *Object* diartikan sebagai suatu entitas yang memiliki identitas, *state* dan *behavior*.

Pada analisis, identitas sebuah *object* menjelaskan bagaimana seorang user membedakannya dari *object* lain, dan *behavior object* digambarkan melalui *event* yang dilakukannya. Sedangkan pada perancangan, identitas sebuah *object*

digambarkan dengan cara bagaimana *object* lain mengenalinya sehingga dapat diakses dan *behavior object* digambarkan dengan *operation* yang dapat dilakukan *object* tersebut yang dapat mempengaruhi *object* lain dalam sistem.

Menurut Satzinger (2010) pengertian *Object Oriented Analysis* (OOA) merupakan semua jenis objek yang melakukan pekerjaan dalam sistem dan menunjukkan interaksi pengguna apa yang dibutuhkan untuk meyelesaikan tugas-tugas. Objek diartikan sebagai suatu hal dalam sistem komputer yang dapat merespon pesan-pesan. *Object Oriented Design* (OOD) adalah semua jenis objek yang diperlukan untuk berkomunikasi dengan orang dan perangkat dalam sistem, menunjukkan bagaimana objek berinteraksi untuk menyelesaikan tugas dan menyempurnakan definisi dari masing-masing objek sehingga dapat diimplementasikan dengan bahasa tertentu.

Sedangkan menurut Whitten (2007), *Object Oriented Analysis and Design* (OOAD) adalah suatu kumpulan alat dan teknik untuk mengembangkan suatu sistem yang akan menggunakan teknologi objek untuk membangun sebuah sistem dan piranti lunak.

#### **2.14.1 Prinsip Umum *Object Oriented Analysis and Design***

*Object Oriented Analysis and Design* memiliki beberapa prinsip-prinsip umum diantaranya (Hariyanto, 2007):

1. Model *the context*: Sistem yang bermanfaat sesuai dengan konteks OOAD. *Emphasize the architecture* merupakan arsitektur yang mudah dipahami yang memfasilitasi kolaborasi antara *designer* dan *programmer*.
2. *Reuse Patterns*: Dibangun berdasarkan gagasan-gagasan yang kuat dan komponen *pretested* memperbaiki kualitas sistem dan produktivitas dari proses *development*.
3. *Tailor the method to suit specific projects*: Setiap usaha *development* masing-masing mempunyai tantangan yang unik. OOAD harus disesuaikan dengan kebutuhan-kebutuhan yang khusus dari situasi analisis dan desain yang diberikan.

### 2.14.2 Karakteristik *Object Oriented Analysis and Design*

Dalam pendekatan berorientasi objek ada 4 pilar utama yang harus dipahami dalam pendekatan berorientasi objek yaitu karakteristik. Karakteristik suatu program termasuk OOAD, apabila terdapat abstraksi, pembungkusan (*encapsulation*), *polymorphism* dan turunan (*inheritance*) (Hariyanto, 2007).

#### 1. *Abstraction*

Kemampuan untuk menjadikan dalam bentuk yang lebih sederhana. Hal ini juga dikenal dalam metodologi pendekatan struktur yaitu dekomposisi seperti menyederhanakan suatu sistem dalam bentuk *Context Diagram*.

#### 2. *Encapsulation*

Merupakan suatu karakteristik OOAD dimana program terbungkus data dan perilaku, artinya lebih memperhatikan aspek internal daripada aspek eksternal. Contoh dalam program terdapat tombol *button close* di dalamnya ada *method system.exit(0)* untuk keluar dari sistem java. Berbeda dengan metodologi terdahulu, metodologi ini menggabungkan atribut dan fungsi/ proses ke dalam suatu objek yang disebut dengan *encapsulation*. Setiap objek dapat menyembunyikan kompleksitasnya dan berhubungan dengan objek lain dengan mengirim pesan yang dapat dikenal dan diproses oleh objek penerima.

#### 3. *Polymorphism*

Suatu mekanisme yang memungkinkan suatu objek memiliki semua atau sebagian definisi dari objek induk. *Polymorphism* adalah kemampuan suatu atribut atau method dapat berubah dalam berbagai bentuk dalam implementasi. Contoh pada obyek mobil, walaupun minibus dan truk merupakan jenis obyek mobil yang sama, namun memiliki juga perbedaan. Misalnya suara truk lebih keras dari pada minibus, hal ini juga berlaku pada obyek anak (*child*) melakukan metode yang sama dengan algoritma berbeda dari obyek induknya. Hal ini yang disebut *polymorphism*, teknik atau konsep dasar lainnya adalah ruang lingkup/

pembatasan. Artinya setiap obyek mempunyai ruang lingkup kelas, atribut dan metode yang dibatasi.

#### 4. *Inheritance*

Merupakan suatu karakteristik OOAD dimana suatu kelas (*parent/base class*) dapat diturunkan ke kelas lain (*child/derived class*), sehingga kelas anak dapat memiliki data atau perilaku kelas orang tuanya. Contoh dengan beberapa buah mobil yang mempunyai kegunaan yang berbeda-beda. Ada mobil bak terbuka seperti truk, bak tertutup seperti sedan dan minibus. Walaupun demikian obyek-obyek ini memiliki kesamaan yaitu teridentifikasi sebagai obyek mobil, obyek ini dikatakan obyek induk (*parent*). Sedangkan minibus obyek anak (*child*), berarti semua operasi yang berlaku pada mobil berlaku pada minibus.

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

Menurut Nugroho (2010), UML (*Unified Modeling Language*) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma (berorientasi objek). Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih maksimal dipelajari dan dipahami.

Menurut Dennis (2015) UML (*Unified Modeling Language*) merupakan kosakata umum berbasis objek dan diagram teknik yang cukup efektif untuk memodelkan setiap proyek pengembangan sistem mulai tahap analisis sampai tahap desain dan implementasi. UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk:

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

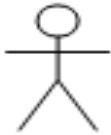
Blok pembangunan utama UML adalah diagram. Beberapa diagram ada yang rinci dan lainnya ada yang bersifat umum (misalnya diagram kelas). Para

pengembang sistem berorientasi objek menggunakan bahasa model untuk mengambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. UML memungkinkan para anggota tim untuk bekerja sama dengan bahasa model yang sama dalam mengaplikasikan beragam sistem. Intinya, UML merupakan alat komunikasi yang konsisten dalam mensupport para pengembang sistem saat ini.

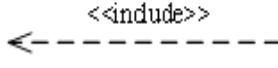
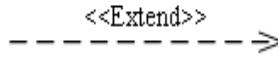
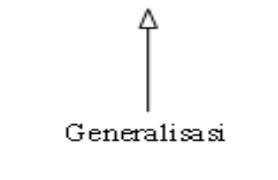
### **2.15.1 Use Case Diagram**

Menurut Dennis, Wixom & Tegarden (2015) *Use Case Diagram* secara grafis menggambarkan, interaksi secara sistem, sistem eksternal dan pengguna. Dengan kata lain *use case diagram* secara grafis mendeskripsikan siapa yang akan menggunakan sistem dan dalam cara apa pengguna (*user*) mengharapkan interaksi dengan sistem itu. *Use Case* secara naratif digunakan untuk secara tekstual menggambarkan sekuensi langkah-langkah dari tiap interaksi. *Use Case Diagram* merupakan suatu diagram yang menangkap kebutuhan bisnis untuk sistem dan untuk menggambarkan interaksi antara sistem dan lingkungannya. Adapun simbol-simbol yang sering digunakan dalam *Use Case Diagram* dapat dilihat pada tabel II.3 berikut ini:

Tabel II.3 Simbol-Simbol *Use Case Diagram*

Simbol	Deskripsi
 <b>Actor/Role</b>  <i>&lt;&lt;Actor&gt;&gt;</i> <b>Actor/Role</b>	<ul style="list-style-type: none"> <li>Menyatakan seseorang atau sistem yang mendapatkan keuntungan dari sistem.</li> <li>Digambarkan sebagai gambar <i>stick/gambar orang (default)</i> atau jika bukan seorang aktor manusia, digambarkan dengan suatu kotak dengan tanda <i>&lt;&lt;actor&gt;&gt;</i> di dalamnya (alternatif).</li> <li>Dilabelkan dengan peran/<i>role</i> dari aktor.</li> </ul>

Tabel II.3 Simbol-Simbol *Use Case Diagram* (lanjutan)

Simbol	Deskripsi
	<ul style="list-style-type: none"> <li>Merepresentasikan bagian utama dari fungsionalitas suatu sistem.</li> <li>Dapat berupa perluasan <i>use case</i> lain.</li> <li>Dapat termasuk di dalam <i>use case</i> lain.</li> <li>Diletakkan di dalam batas sistem.</li> <li>Dinamakan dengan frasa kata kerja.</li> </ul>
	<ul style="list-style-type: none"> <li>Menyertakan nama subjek di dalam maupun di atas.</li> <li>Merepresentasikan ruang lingkup dari subjek, sistem atau proses bisnis.</li> </ul>
	<ul style="list-style-type: none"> <li>Menghubungkan suatu aktor dengan <i>use case</i> dengan interaksi antara keduanya.</li> </ul>
	<ul style="list-style-type: none"> <li>Merepresentasikan fungsionalitas suatu <i>use case</i> dengan <i>use case</i> lainnya.</li> <li>Disimbolkan dengan anak panah dari sebuah <i>use case</i> dasar ke <i>use case</i> yang digunakan.</li> </ul>
	<ul style="list-style-type: none"> <li>Merepresentasikan perluasan (<i>extend</i>) dari <i>use case</i> lain untuk menyertakan perilaku opsional/<i>optional</i> (tidak wajib).</li> <li>Disimbolkan dengan anak panah yang digambarkan dari perluasan <i>use case</i> ke <i>use case</i> dasar.</li> </ul>
	<ul style="list-style-type: none"> <li>Merepresentasikan <i>use case</i> khusus ke satu <i>use case</i> yang lebih umum.</li> <li>Disimbolkan dengan anak panah yang digambarkan dari <i>use case</i> khusus ke <i>use case</i> umum.</li> </ul>

(Sumber: Dennis, Wixom &amp; Tegarden, 2015)

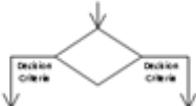
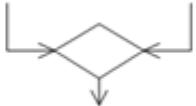
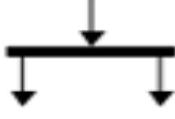
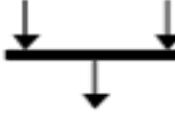
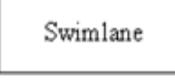
### 2.15.2 Activity Diagram

Secara grafis untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis maupun *use case*. *Activity Diagram* dapat juga digunakan untuk memodelkan *action* yang akan dilakukan saat operasi dieksekusi, dan memodelkan hasil dari *action* tersebut. Pengertian *Activity Diagram* adalah yang menggambarkan alur kerja bisnis independen dari *class*, aliran kegiatan dalam *use case*, atau desain rinci sebuah metode (Dennis, Wixom & Roth, 2012). Adapun simbol-simbol yang sering digunakan dalam *Activity Diagram* dapat dilihat pada tabel II.4 berikut ini:

Tabel II.4 Simbol-Simbol *Activity Diagram*

No	Simbol	Nama	Deskripsi
1		<i>Action</i>	Merupakan notasi yang <i>simple</i> , dan tidak untuk dianalisis perilakunya. Dinamakan sesuai aksi yg dilakukan.
2		<i>Activity</i>	Digunakan untuk mewakili satu <i>set</i> aksi yang dilakukan.
3		<i>Object Node</i>	Digunakan untuk mewakili objek yang terhubung ke satu <i>set</i> arus objek.
4		<i>Control Flow</i>	Menampilkan urutan eksekusi.
5		<i>Object Flow</i>	Menunjukkan aliran dari sebuah objek dari satu aktivitas ke aktivitas lainnya.
6		<i>Initial Node</i>	Menggambarkan permulaan dari suatu <i>set</i> aktivitas atau aksi.

Tabel II.4 Simbol-Simbol *Activity Diagram* (lanjutan)

No	Simbol	Nama	Deskripsi
7		<i>Final-Activity Node</i>	Menggambarkan akhir dari setiap aktivitas.
8		<i>Final-Flow Node</i>	Menggambarkan akhir aliran <i>control</i> spesifik atau aliran objek.
9		<i>Decision Node</i>	Menggambarkan suatu kondisi untuk memastikan bahwa arah panah/arrah objek hanya memiliki satu jalur ke bawah.
10		<i>Merge Node</i>	Digunakan untuk membawa aliran keputusan yang berbeda ke satu <i>decision node</i> .
11		<i>Fork Node</i>	Menggambarkan percabangan dari satu aliran aktivitas.
12		<i>Join Node</i>	Menyatukan beberapa percabangan dari aliran aktivitas.
13		<i>Swimlane</i>	Menggambarkan akhir aliran <i>control</i> spesifik atau aliran objek.

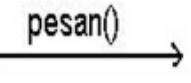
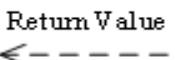
(Sumber: Dennis, Wixom &amp; Roth, 2012)

### 2.15.3 Sequence Diagram

*Sequence Diagram* merupakan urutan model dinamis yang menggambarkan contoh *class* yang berpartisipasi dalam *use case* dan pesan yang lewat di antara mereka dari waktu ke waktu (Dennis, Wixom & Roth, 2012). *Sequence Diagram* merupakan diagram interaksi yang disusun berdasarkan urutan

waktu. Cara membaca diagram sekuensial dari atas ke bawah. Setiap diagram sekuensial mempresentasikan satu *flow* dari beberapa *flow* di dalam *use case*. Adapun simbol-simbol *Sequence Diagram* dapat dilihat pada tabel II.5 berikut ini:

Tabel II.5 Simbol-Simbol *Sequence Diagram*

No	Simbol	Nama	Deskripsi
1	 <b>Actor/Role</b> <code>&lt;&lt;Actor&gt;&gt;</code> <b>Actor/Role</b>	<i>Actor</i>	<ul style="list-style-type: none"> <li>• Orang atau sistem yang berasal dari luar sistem yang dapat memberikan manfaat.</li> <li>• Berpartisipasi secara berurutan mengirim atau menerima pesan.</li> <li>• Ditempatkan di bagian atas diagram.</li> </ul>
2	 <b>objek:kelas</b>	<i>Object</i>	<ul style="list-style-type: none"> <li>• Berpartisipasi secara berurutan mengirim atau menerima pesan.</li> <li>• Ditempatkan di bagian atas diagram.</li> </ul>
3		<i>Lifeline</i>	<ul style="list-style-type: none"> <li>• Menandakan kehidupan sebuah objek selama <i>sequence</i>.</li> <li>• Berisi sebuah X pada titik dimana kelas tidak lagi berinteraksi.</li> </ul>
4		<i>Execution Occurrence</i>	<ul style="list-style-type: none"> <li>• Menandakan sebuah persegi panjang yang sempit ditempatkan diatas sebuah garis hidup.</li> <li>• Menandakan ketika suatu objek mengirim atau menerima pesan.</li> </ul>
5	 <b>pesan()</b>	<i>Message</i>	<ul style="list-style-type: none"> <li>• Menyampaikan informasi dari satu objek ke objek yang lain.</li> </ul>
6	 <b>Return Value</b>	<i>Message (return)</i>	<ul style="list-style-type: none"> <li>• Pesan yang dikirim untuk diri sendiri.</li> </ul>

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

No	Simbol	Nama	Deskripsi
7		<i>Guard Condition</i>	<ul style="list-style-type: none"> <li>Menggambarkan tes yang harus dipenuhi untuk pesan yang akan dikirim.</li> </ul>
8		<i>Object Destruction</i>	<ul style="list-style-type: none"> <li>Merupakan sebuah X ditempatkan pada akhir suatu garis hidup untuk menunjukkan bahwa itu akan keluar dari eksistensi.</li> </ul>
9		<i>Frame</i>	<ul style="list-style-type: none"> <li>Mengindikasikan konteks dari <i>sequence diagram</i>.</li> </ul>

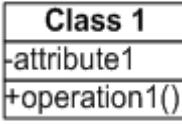
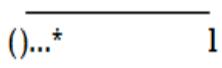
(Sumber: Dennis, Wixom & Roth, 2012)

#### 2.15.4 *Class Diagram*

*Class Diagram* adalah ilustrasi antara *class* yang dimodelkan di dalam sistem. *Class Diagram* sangat mirip dengan diagram hubungan entitas (ERD). *Class Diagram* menggambarkan *class* yang meliputi atribut, perilaku dan *states*, sementara dalam ERD hanya mencakup atribut (Dennis, Wixom & Roth, 2012).

*Class Diagram* menggambarkan *class* dan hubungan antar-*class* di dalam sistem. *Class Diagram* dibangun berdasarkan *use case diagram*, *sequence diagram*, atau *collaboration diagram* yang telah dibuat sebelumnya. Diagram *Class* memberikan pandangan secara luas dari suatu sistem dengan menunjukkan kelas-kelasnya dan hubungan mereka. Diagram *Class* bersifat statis, menggambarkan hubungan apa yang terjadi bukan yang terjadi jika mereka berhubungan. Adapun simbol-simbol *Class Diagram* dapat dilihat pada tabel II.6 berikut ini:

Tabel II.6 Simbol-Simbol *Class Diagram*

No	Simbol	Nama	Deskripsi
1		<i>Class</i>	<ul style="list-style-type: none"> <li>Mewakili jenis orang, tempat atau hal yang sistem harus menangkap dan menyimpan informasi.</li> <li>Memiliki nama yang diketik dengan huruf tebal dan berpusat di atas kompartemen.</li> <li>Memiliki daftar atribut di tengah kompartemen.</li> <li>Memiliki daftar operasi.</li> </ul>
2	Nama atribut ()	<i>Attribut</i>	<ul style="list-style-type: none"> <li>Merupakan sifat yang menggambarkan bagian suatu objek.</li> <li>Dapat diturunkan dari atribut lain, ditunjukkan oleh penempatan garis miring sebelum nama atribut.</li> </ul>
3	Nama operasi/ <i>operation</i> ()	<i>Operation</i>	<ul style="list-style-type: none"> <li>Mewakili suatu aksi atau fungsi dari suatu <i>class</i>.</li> <li>Dapat diklasifikasikan sebagai konstruktor (<i>method</i> yang secara otomatis dipanggil/dijalankan pada sebuah <i>class</i>), <i>query</i>, atau perubahan operasi. Tanda kurung mengandung informasi atau parameter yang dibutuhkan untuk menampilkan operasi.</li> </ul>
4		<i>AssociatedWith</i>	<ul style="list-style-type: none"> <li>Mewakili sebuah hubungan antara beberapa <i>class</i> atau <i>class</i> dengan <i>class</i>.</li> <li>Digambarkan dengan kata kerja atau nama, yang mewakili sebuah hubungan.</li> <li>Tersedia diantara satu atau lebih <i>class</i>.</li> </ul>

Tabel II.6 Simbol-Simbol *Class Diagram* (lanjutan)

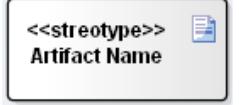
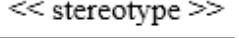
No	Simbol	Nama	Deskripsi
5		<i>Generalization</i>	<ul style="list-style-type: none"> <li>Merupakan sebuah <i>taxonomic relationship</i> antara <i>class</i> yang lebih umum dengan <i>class</i> yang lebih khusus.</li> <li>Mewakili hubungan antara beberapa <i>class</i> (<i>class</i> banyak).</li> </ul>
6	 0..* <i>IsPartOf</i> ▶ 1	<i>Aggregation</i>	<ul style="list-style-type: none"> <li>Mewakili bagian hubungan yang logis antara beberapa <i>class</i> atau <i>class</i> dengan <i>class</i>.</li> <li>Merupakan bagian dari <i>association</i>.</li> </ul>
7	 1..* <i>IsPartOf</i> ▶ 1	<i>Composition</i>	<ul style="list-style-type: none"> <li>Mewakili bagian hubungan yang fisik antara beberapa <i>class</i> atau <i>class</i> dengan <i>class</i>.</li> <li>Merupakan bagian dari <i>association</i>.</li> </ul>

(Sumber: Dennis, Wixom &amp; Roth, 2012)

### 2.15.5 Deployment Diagram

*Deployment diagram* digunakan untuk mewakili hubungan antara komponen-komponen *hardware* yang digunakan dalam infrastruktur fisik sistem informasi. Misalnya, ketika membuat suatu sistem informasi terdistribusi yang akan menggunakan jaringan yang 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.*, 2015). Adapun simbol-simbol *Deployment Diagram* dapat dilihat pada tabel II.7 berikut ini:

Tabel II.7 Simbol-Simbol *Deployment Diagram*

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

(Sumber: Dennis *et al.*, 2015)

## 2.16 Entity Relationship Diagram (ERD)

Menurut Mulyani (2016) *Entity Relationship Diagram* (ERD) adalah *tools* yang digunakan untuk melakukan pemodelan data secara abstrak dengan tujuan untuk mendeskripsikan atau menggambarkan struktur dari data yang digunakan. Adapun fungsi utama ERD yaitu sebagai alat untuk memodelkan hasil dari analisis data, sebagai alat untuk memodelkan data konseptual dan sebagai alat untuk memodelkan objek-objek dalam suatu sistem. Adapun simbol-simbol ERD dapat dilihat pada tabel II.8 berikut ini:

Tabel II.8 Simbol-Simbol *Entity Relationship Diagram*

No	Simbol	Nama	Deskripsi
1		Entitas	<ul style="list-style-type: none"> <li>• Kumpulan dari objek yang dapat diidentifikasi secara unik.</li> </ul>

Tabel II.8 Simbol-Simbol *Entity Relationship Diagram* (lanjutan)

No	Simbol	Nama	Deskripsi
2		Relasi	<ul style="list-style-type: none"> <li>Hubungan yang terjadi antara satu atau lebih entitas. Jenis hubungan antara lain: satu ke satu, satu ke banyak, dan banyak ke banyak.</li> </ul>
3		Atribut	<ul style="list-style-type: none"> <li>Karakteristik dari <i>entity</i> atau relasi yang merupakan penjelasan detail tentang entitas.</li> </ul>
4		Garis	<ul style="list-style-type: none"> <li>Hubungan antara <i>entity</i> dengan atributnya dan himpunan entitas dengan himpunan relasi.</li> </ul>

(Sumber: Mulyani, 2016)

Menurut Mulyani (2016), ERD berfungsi untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. Pada dasarnya ada tiga komponen yang digunakan, yaitu:

1. *Entity* (Entitas)

*Entity* merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol dari *entity* ini biasanya digambarkan dengan persegi panjang.

2. Atribut

Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain. Gambar atribut diwakili oleh simbol elips.

3. Hubungan/Relasi

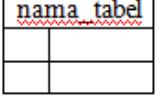
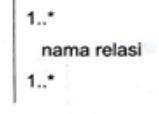
Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Relasi dapat digambarkan sebagai berikut: Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dalam satu basis data yaitu:

- a. Satu ke satu (*one to one*) hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B.
- b. Satu ke banyak (*one to many*) setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A.
- c. Banyak ke banyak (*many to many*) setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B.

### 2.17 *Conceptual Data Model (CDM)*

Menurut Rosa dan Shalahuddin (2013), CDM (*Conceptual Data Model*) atau model konsep data merupakan konsep yang berkaitan dengan pandangan pemakai terhadap data yang disimpan dalam basis data. CDM telah dibuat dalam bentuk tabel-tabel tanpa tipe data yang menggambarkan relasi antar tabel untuk keperluan implementasi ke basis data. CDM merupakan hasil penjabaran dari ERD. Adapun simbol-simbol CDM dapat dilihat pada tabel II.9 berikut ini:

Tabel II.9 Simbol-Simbol *Conceptual Data Model*

No	Simbol	Deskripsi
1	<b>Entitas/table</b> 	Entitas atau tabel yang menyimpan data dalam basis data.
2	<b>Relasi</b> 	Relasi antar tabel yang terdiri atas nama relasi dan <i>multiplicity</i> .

(Sumber: Rosa dan Shalahuddin, 2013)

### 2.18 **Kamus Data**

Menurut Rosa dan Shalahudin (2013), kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan). Kamus data dalam implementasi program dapat menjadi parameter

masukan atau keluaran dari sebuah fungsi atau prosedur. Kamus data biasanya berisi:

1. Nama: nama dari data.
2. Deskripsi: merupakan deskripsi data.
3. Informasi tambahan: seperti tipe data, nilai data, batas nilai data dan komponen yang membentuk data.

Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan seperti tabel II.10 berikut ini:

Tabel II.10 Simbol-Simbol Kamus Data

No	Simbol	Keterangan
1	=	Disusun atau terdiri dari
2	+	Dan
3	[]	Baik ... atau ...
4	{ } <sup>n</sup>	n kali diulang/bernilai banyak
5	( )	Data opsional
6	* ... *	Batas Komentar

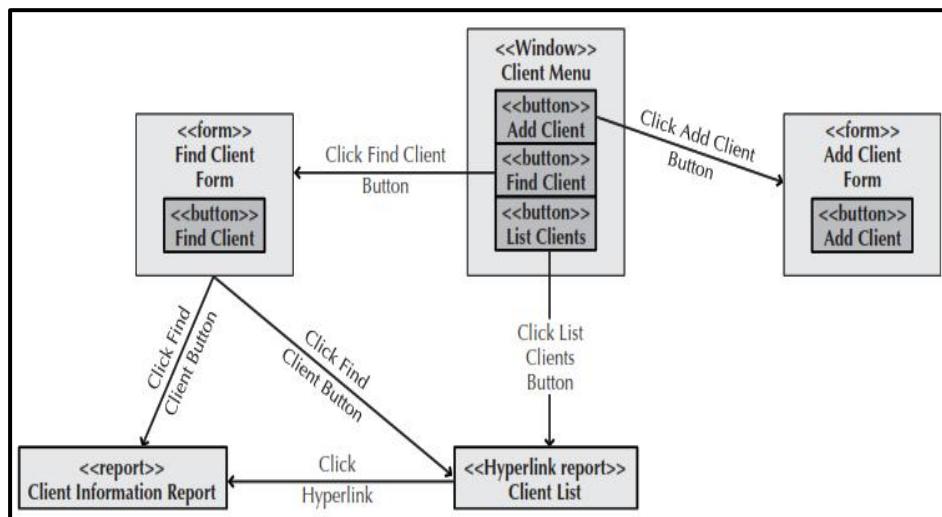
(Sumber: Rosa dan Shalahudin, 2013)

## 2.19 Windows Navigation Diagram (WND)

Menurut Dennis (2015) dalam bukunya *Systems Analysis & Design* menjelaskan *Windows Navigation Diagram* (WND) digunakan untuk menunjukkan bagaimana semua layar, formulir dan laporan yang digunakan oleh sistem terkait dan bagaimana pengguna berpindah dari satu ke lainnya. Sebagian besar sistem memiliki beberapa WND, satu untuk setiap bagian utama dari sistem.

WND sangat mirip dengan status antarmuka pengguna, karena keduanya memodelkan perubahan status. Status antarmuka pengguna biasanya memodelkan perubahan status suatu objek, sedangkan WND memodelkan perubahan status antarmuka pengguna. Dalam WND, setiap keadaan antarmuka pengguna direpresentasikan sebagai sebuah kotak. Sebuah kotak biasanya terkait dengan komponen antarmuka pengguna, seperti jendela, formulir, tombol atau laporan.

Transisi dimodelkan sebagai panah berkepala tunggal atau berkepala dua. Satu panah yang mengarah menunjukkan bahwa kembali ke keadaan panggilan tidak diperlukan, sedangkan panah berkepala ganda menunjukkan pengembalian yang diminta. Item terakhir yang akan dijelaskan dalam WND adalah stereotipe. Sebuah stereotip dimodelkan sebagai item teks yang diapit oleh guillemet atau kurung sudut (<>). Stereotip mewakili jenis komponen antarmuka pengguna kotak pada diagram. Misalnya, menu klien adalah jendela, sedangkan temukan formulir klien adalah formulir. Gambar II.2 berikut ini adalah contoh dari WND yang ada pada umumnya.



Gambar II.2 Contoh Windows Navigation Diagram

(Sumber: Dennis, 2015)

Struktur navigasi dasar dari suatu antarmuka mengikuti struktur dasar dari proses bisnis itu sendiri, sebagaimana didefinisikan dalam kasus penggunaan dan model perilaku. Analis mulai dengan kasus penggunaan penting dan mengembangkan aliran fundamental kontrol sistem saat bergerak dari satu objek ke objek lainnya. Analis kemudian memeriksa skenario penggunaan untuk melihat seberapa baik WND mendukungnya. Cukup sering, skenario penggunaan mengidentifikasi jalur melalui WND yang lebih rumit dari yang seharusnya. Analis kemudian mengerjakan ulang WND untuk menyederhanakan kemampuan antarmuka untuk mendukung skenario penggunaan, kadang-kadang dengan membuat perubahan besar pada struktur menu, kadang-kadang dengan menambahkan cara pintas.

## 2.20 PHP (*Hypertext Preprocessor*)

PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1994. PHP (PHP: *Hypertext Preprocessor*) adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis. Karena PHP merupakan *server-side scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi di *server* kemudian hasilnya dikirimkan ke *browser* dalam format HTML. Dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh *user* sehingga keamanan dalam *web* lebih terjamin (Sutaji, 2012).

Menurut Sidik (2012), PHP secara umum dikenal dengan sebagai bahasa pemrograman *script-script* yang membuat dokumen HTML secara *on the fly* yang dieksekusi di *server web*, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemrograman *server side*.

PHP dirancang untuk membentuk halaman *web* yang dinamis, yaitu halaman *web* yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman *web*. Untuk menjalankan sistem PHP dibutuhkan 3 komponen, di antaranya adalah sebagai berikut (Sutaji, 2012):

1. *Webserver*
2. Program PHP
3. *Database Server*

Adapun salah satu kelebihan dari bahasa PHP di antaranya adalah PHP mendukung banyak *database*. Berikut adalah *database* yang dapat didukung oleh PHP yaitu Adabas D, MSQL, dBase, MySQL, Direct MS-SQL, ODBC, Empress, Oracle (OCI7 dan OCI8), Filepro(Read only), Ovrimos, FrontBase, PostgreSQL, Hyperwave, Solid, IBM DB2, Sybase, Informix, Unix DBM, Ingres dan Velocis.

## 2.21 *Database*

Basis data (*database*) merupakan kumpulan *file/tabel/arsip* yang saling berhubungan yang disimpan dalam media penyimpanan elektronis (Fathansyah, 2012). Basis Data dapat didefinisikan dalam sejumlah sudut pandang seperti:

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan *file/table* yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

## 2.22 MySQL

MySQL adalah sebuah program pembuat dan pengelola *database* atau yang sering disebut dengan DBMS (*Database Management System*), sifat dari DBMS ini adalah *open source*. MySQL sebenarnya produk yang berjalan pada *platform* Linux, dengan adanya perkembangan dan banyaknya pengguna serta lisensi dari *database* ini adalah *open source*, maka para pengembang kemudian merilis versi *Windows* (Nugroho, 2010).

MySQL 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).

MySQL mendukung banyak tipe data yang dapat disimpan pada sebuah kolom. Penjelasan singkat kegunaan masing-masing jenis data akan sangat bermanfaat dalam memilih suatu jenis data yang dipakai dalam merancang tabel. Beberapa jenis data yang tersedia pada MySQL dapat dilihat pada tabel II.11 (Sutaji, 2012).

Tabel II.11 Tipe Data MySQL

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. Bahkan 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: Sutaji, 2012)

## 2.23 CodeIgniter

Menurut Supono dan Putratama (2016) *Code Igniter* adalah aplikasi *open source* yang berupa *framework* dengan model *MVC* (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan PHP. Sedangkan menurut Purbadian (2016) *Code Igniter* adalah sebuah *framework* yang dibuat menggunakan bahasa pemrograman PHP yang bertujuan untuk memudahkan para *programmer web* untuk membuat atau mengembangkan aplikasi berbasis *web*.

Menurut Widodo (2013) *Codeigniter* adalah merupakan *framework* PHP yang diklaim memiliki eksekusi tercepat dibandingkan dengan *framework*

lainnya. *Codeigniter* bersifat *open source* dan menggunakan model *basic MVC (Model, View, Controller)*, yang merupakan model konsep modern *framework* yang digunakan saat ini.

Menurut Hakim (2010) *CodeIgniter* adalah sebuah *framework* PHP yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi *web* berbasis PHP dibanding jika menulis semua kode program dari awal. *CodeIgniter* pertama kali dibuat oleh Rick Ellis, CEO Ellislab, Inc. (<http://ellislab.com>), sebuah perusahaan yang memproduksi CMS (*Content Management System*) yang cukup handal, yaitu *Expression Engine* (<http://www.expressionengine.com>). Saat ini, *CodeIgniter* dikembangkan dan *di-maintain* oleh *Expression Engine Development Team*. Adapun beberapa keuntungan menggunakan *CodeIgniter*, diantaranya (Hakim, 2010):

1. Gratis

*CodeIgniter* berlisensi di bawah Apache/BSD *open source*.

2. Ditulis menggunakan PHP 4

Meskipun *CodeIgniter* dapat berjalan di PHP 5, namun sampai saat ini kode program *CodeIgniter* masih dibuat dengan menggunakan PHP 4.

3. Berukuran kecil

Ukuran *CodeIgniter* yang kecil merupakan keunggulan tersendiri. Dibanding dengan *framework* lain yang berukuran besar.

4. Menggunakan konsep MVC

*CodeIgniter* menggunakan konsep MVC yang memungkinkan pemisahan *layer application-logic* dan *presentation*.

5. URL yang sederhana

Secara *default*, URL yang dihasilkan *CodeIgniter* sangat bersih dan *Search Engine Friendly* (SEF).

6. Memiliki paket *library* yang lengkap

*CodeIgniter* mempunyai *library* yang lengkap untuk mengerjakan operasi-operasi yang umum dibutuhkan oleh sebuah aplikasi berbasis *web*, misalnya mengakses *database*, mengirim *email*, memvalidasi *form*, menangani *session* dan sebagainya.

7. *Extensible*

Sistem dapat dikembangkan dengan mudah menggunakan *plugin* dan *helper*, atau dengan menggunakan *hooks*.

8. Tidak memerlukan *template engine*

Meskipun *CodeIgniter* dilengkapi dengan *template* sederhana yang dapat digunakan, tetapi hal ini tidak mengharuskan kita untuk menggunakananya.

9. Dokumentasi lengkap dan jelas

Dari sekian banyak *framework*, *CodeIgniter* adalah satu-satunya *framework* dengan dokumentasi yang lengkap dan jelas.

10. Komunitas

Komunitas *CodeIgniter* saat ini berkembang pesat. Salah satu komunitasnya bisa dilihat di (<http://codeigniter.com/forum/>).

Berdasarkan pengertian di atas dapat ditarik kesimpulan bahwa *Codeigniter* adalah aplikasi *open source* yang merupakan *framework* PHP dengan eksekusi tercepat dan menggunakan model *basic MVC* (*Model, View, Controller*) untuk membangun sebuah *website* yang dinamis dan bertujuan untuk memudahkan para *programmer web* untuk membuat atau mengembangkan aplikasi berbasis *web*.

## **2.24 Blackbox Testing**

Menurut Soetam (2011), berpendapat bahwa *blackbox testing* adalah tipe *testing* yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para *tester* memandang perangkat lunak seperti layaknya sebuah “kotak hitam” yang tidak penting dilihat isinya, tapi cukup dikenai proses *testing* di bagian luar.

*Black-Box Testing* adalah metode pengujian perangkat lunak yang menguji fungsionalitas aplikasi tanpa mengintip ke dalam struktur internal atau cara kerjanya. Metode pengujian ini dapat diterapkan secara virtual untuk setiap tingkat pengujian perangkat lunak (unit, integrasi, sistem dan penerimaan).

Kadang-kadang disebut sebagai pengujian berbasis spesifikasi (Patton, 2005). Contoh kegiatan yang dilakukan pada *black-box testing* yaitu:

1. Membuat *test case* untuk menguji fungsi-fungsi yang ada pada aplikasi.
2. Membuat *test case* untuk menguji kesesuaian alur kerja suatu fungsi di aplikasi dengan *requirement* yang dibutuhkan *user* untuk fungsi tersebut.
3. Mencari *bugs/error* dari tampilan (*interface*) aplikasi.

*Black-box testing* merupakan pengujian yang berpusat pada kebutuhan fungsional perangkat lunak dimana memungkinkan untuk memperoleh sekumpulan kondisi input yang secara penuh memeriksa fungsional dari sebuah aplikasi. *Black-box testing* berusaha menemukan kesalahan-kesalahan seperti kesalahan fungsi dan kesalahan tampilan aplikasi. *Black-box testing* dapat digunakan untuk menguji aplikasi konvensional dan aplikasi yang berorientasi objek (Pressman, 2015).

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Metodologi Penelitian**

Menurut Sugiyono (2013), metode penelitian pada dasarnya merupakan cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu. Berdasarkan hal tersebut terdapat empat kata kunci yang perlu diperhatikan yaitu cara ilmiah, data, tujuan dan kegunaan.

Menurut Darmadi (2013), metode penelitian adalah suatu cara ilmiah untuk mendapatkan data dengan tujuan kegunaan tertentu. Cara ilmiah berarti kegiatan penelitian itu didasarkan pada ciri-ciri keilmuan yaitu rasional, empiris dan sistematis. Berdasarkan pemaparan di atas dapat disimpulkan bahwa metode penelitian adalah suatu cara ilmiah untuk memperoleh data dengan tujuan dan kegunaan tertentu.

#### **3.2 Jenis dan Sumber Data**

Menurut Arikunto (2010) sumber data dalam penelitian adalah subjek dari mana data dapat diperoleh. Apabila peneliti menggunakan kuesioner atau wawancara dalam pengumpulan datanya, maka sumber data disebut responden, yaitu orang yang merespon atau menjawab pertanyaan-pertanyaan peneliti, baik pertanyaan tertulis maupun lisan. Dalam suatu penelitian terdapat dua sumber data yang dipakai, data tersebut adalah sebagai berikut:

##### **1. Sumber Data Primer**

Sumber data primer adalah data asli yang dikumpulkan sendiri diperoleh secara langsung dari PT Multi Prima Sejahtera Tbk, dimana pengumpulan data dan informasi dilakukan langsung dari objek yang diteliti melalui hasil wawancara yaitu dengan melakukan tanya jawab langsung kepada Bagian Gudang untuk memperoleh informasi mengenai objek penelitian. Data-data tersebut adalah data yang digunakan dalam proses penerimaan dan

pengeluaran bahan baku diantaranya, analisis sistem yang berjalan, analisis dokumen yang berjalan serta proses bisnis sistem yang akan diusulkan.

## 2. Sumber Data Sekunder

Sumber data sekunder adalah data yang diperoleh dari sumber yang tidak langsung atau melalui sumber lain yang sudah tersedia. Dimana pengumpulan data dan informasi didapat melalui catatan atau buku-buku yang ada pada perusahaan, literatur-literatur serta sumber lain yang berhubungan dengan objek penelitian.

### 3.3 Metode Pengumpulan Data

Menurut Sugiyono (2013) teknik pengumpulan data merupakan langkah yang paling strategis dalam penelitian, karena tujuan utama dari penelitian adalah mendapatkan data. Adapun cara yang digunakan untuk mengumpulkan data dan informasi dalam penelitian ini adalah:

#### 1. Studi Lapangan

Melakukan pengumpulan data secara langsung untuk mendapatkan data yang sesuai, yaitu dengan cara:

##### a. Pengamatan

Melakukan pengamatan secara langsung pada Bagian Gudang di PT Multi Prima Sejahtera Tbk dengan mengamati proses penerimaan dan pengeluaran bahan baku yang sedang berjalan. Hal yang diamati adalah proses bahan baku masuk dan keluar gudang, *supplier* yang mengirim bahan baku dan mengetahui jumlah bahan baku yang diterima dan dikeluarkan.

##### b. Wawancara

Pengambilan data dengan cara berdialog dan bertanya dengan karyawan divisi terkait tentang segala hal yang diperlukan pada penyusunan Tugas Akhir khususnya bagaimana sistem informasi penerimaan dan pengeluaran bahan baku berjalan di PT Multi Prima Sejahtera Tbk serta dokumen-dokumen yang terlibat pada proses penerimaan dan pengeluaran bahan baku tersebut.

## 2. Riset Pustaka

Mengumpulkan data dan menambah referensi dengan membaca buku-buku, literatur, artikel di internet atau sumber tertulis lain yang berhubungan dengan topik yang dibahas guna melengkapi data yang diperlukan dalam penelitian ini, agar dalam praktik dan teori tidak jauh berbeda.

### 3.4 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan adalah metode pengembangan sistem *waterfall*. Metode *waterfall* memiliki kelebihan yaitu memiliki proses yang lebih terstruktur, hal ini membuat kualitas *software* baik dan tetap terjaga. Dengan adanya urutan yang pasti, dapat dilihat perkembangan tahap demi tahap secara pasti. Selain itu, model ini merupakan jenis model yang bersifat dokumen lengkap sehingga proses pemeliharaan dapat dilakukan dengan mudah. Metode *waterfall* ini terdiri dari tahap *planning*, *analysis*, *design* dan *implementation*. Berikut ini akan dijelaskan secara singkat mengenai tahapan-tahapan yang dilakukan dalam pengembangan sistem dengan metode *waterfall*. Berikut ini merupakan tahapan-tahapan metode *waterfall* menurut (Dennis, 2015):

#### 1. Tahap Perencanaan (*Planning*)

Pada tahap ini akan dilakukan perencanaan untuk membangun suatu sistem dengan memahami konteks bisnis aplikasi yang akan dibuat.

#### 2. Tahap Analisis (*Analysis*)

Pada tahap ini, merupakan proses analisis kebutuhan sistem. Analis mengumpulkan data-data sebagai bahan pengembangan sistem. Pengumpulan data dapat dilakukan dengan teknik wawancara maupun teknik observasi.

#### 3. Tahap Desain (*Design*)

Proses desain pengembang melakukan desain seperti desain struktur data dan representasi antarmuka. Data-data yang didapat dari tahap analisis diterapkan dalam tahap desain.

#### 4. Tahap Implementasi (*Implementation*)

Pada tahap ini, pengembang akan membangun sebuah sistem dengan rancangan yang sudah ada kemudian diterjemahkan ke dalam sebuah *code* atau aktivitas *coding*.

### 3.5 Kerangka Penelitian

Kerangka penelitian merupakan gambaran langkah-langkah kegiatan mulai dari awal hingga akhir. Adapun langkah-langkah atau tahapan dalam kerangka penelitian tersebut adalah sebagai berikut:

#### 1. Studi Pendahuluan

Pada tahap ini merupakan tahapan awal yang dilakukan untuk memulai penelitian. Proses penanganan yang dilakukan dengan mengetahui terlebih dahulu alur proses bisnis yang berjalan pada PT Multi Prima Sejahtera Tbk. Studi pendahuluan ini dimaksud untuk mengetahui gambaran alur bisnis yang terjadi pada Bagian Gudang saat ini serta mengetahui masalah yang terjadi pada Bagian Gudang tersebut. Langkah-langkah yang dilakukan dalam studi pendahuluan untuk mendapatkan informasi adalah melakukan observasi dan wawancara terhadap pihak-pihak terkait yang berada pada Bagian Gudang. Selain itu, studi pendahuluan juga dilakukan dengan cara membaca buku, literatur, serta sumber lain guna mendapatkan beberapa referensi untuk melakukan penelitian.

#### 2. Identifikasi Masalah

Pada tahap ini peneliti melakukan penelitian dengan menganalisis sistem penerimaan dan pengeluaran bahan baku yang sedang berjalan sehingga dapat diidentifikasi permasalahan yang ada pada proses penerimaan dan pengeluaran bahan baku di PT Multi Prima Sejahtera Tbk. Masalah yang terjadi pada Bagian Gudang PT Multi Prima Sejahtera Tbk adalah data yang terkait dengan proses penerimaan dan pengeluaran bahan baku masih disimpan dan diolah secara manual menggunakan *form-form* yang kemudian ditumpuk di dalam sebuah arsip penyimpanan. Hal ini dapat membuat data yang terkait dengan proses penerimaan dan pengeluaran bahan baku sulit

dicari, mudah hilang dan rusak. Selain itu, belum adanya sistem informasi pada proses penerimaan dan pengeluaran bahan baku menyebabkan karyawan sulit memperoleh informasi mengenai data bahan baku yang diterima dan dikeluarkan sehingga proses penerimaan dan pengeluaran bahan baku membutuhkan waktu yang cukup lama. Adapun cara mengidentifikasi masalah dalam penelitian ini dengan melakukan:

- a. Melakukan observasi selama satu bulan dengan pengumpulan data di PT Multi Prima Sejahtera Tbk pada penerimaan dan pengeluaran bahan baku di Bagian Gudang.
- b. Melakukan wawancara pada Bagian Gudang mengenai proses penerimaan dan pengeluaran bahan baku.

### 3. Identifikasi Solusi

Pada tahap setelah melakukan identifikasi suatu masalah, langkah berikutnya adalah identifikasi solusi yang bertujuan untuk mengetahui maksud dan tujuan dari penelitian, maksud dan tujuan dari melakukan penelitian di Bagian Gudang antara lain:

- a. Menganalisis, merancang dan membangun sistem informasi penerimaan dan pengeluaran bahan baku dengan media penyimpanan data dalam bentuk *database* agar dapat menyimpan data lebih aman dan terstruktur.
- b. Menganalisis, merancang dan membangun sistem informasi penerimaan dan pengeluaran bahan baku yang dapat memberikan kemudahan pengolahan data bahan baku yang diterima dan dikeluarkan sehingga proses penerimaan dan pengeluaran bahan baku menjadi lebih optimal.

### 4. Penerapan Metode *Waterfall*

Pengembangan sistem yang digunakan dalam penelitian Tugas Akhir ini adalah menggunakan metode *waterfall*. Metode *waterfall* ini memiliki beberapa tahapan yaitu:

#### a. Perencanaan (*Planning*)

Pada tahap ini diawali dengan memahami konteks bisnis aplikasi yang akan dibuat berdasarkan hasil observasi dan wawancara pada karyawan PT Multi Prima Sejahtera Tbk.

### b. Analisis (*Analysis*)

Pada tahap ini menganalisis sistem informasi penerimaan dan pengeluaran bahan baku yang sedang berjalan dan memberikan usulan pengembangan sistem berdasarkan kebutuhan pengguna terhadap sistem informasi yang akan dibuat. Kemudian mengolah data yang didapat dengan menganalisis dokumen dan analisis proses bisnis yang berjalan sehingga menghasilkan gambaran fungsionalitas sistem serta batasan-batasan yang disediakan oleh sistem. Tahap ini menghasilkan daftar kebutuhan fungsional dan kebutuhan non fungsional sistem. Kebutuhan fungsional adalah aktivitas yang harus dilakukan oleh sistem dengan contoh-contoh sebagai berikut:

- Sistem dapat melakukan *input* pendataan buku,
- Sistem dapat melakukan *input* pendataan anggota,
- Sistem dapat melakukan transaksi peminjaman.

Sedangkan kebutuhan non fungsional adalah karakteristik dari sistem selain aktivitas yang harus dilakukan atau didukung dengan contoh-contoh sebagai berikut:

- Sistem dapat dijalankan oleh beberapa *software web browser* diantaranya Internet Explore, Google Chrome dan Mozilla Firefox,
- Sistem harus dapat memastikan bahwa data yang digunakan dalam sistem harus terlindung dari akses yang tidak berwenang,
- Besarnya program dari sistem maksimal sebesar 100 MB.

Pada tahap ini juga dilakukan usulan perbaikan proses bisnis.

### c. Desain (*Design*)

Pada tahap ini akan dirancang sebuah desain struktur data dan representasi antarmuka sistem informasi penerimaan dan pengeluaran bahan baku. Berikut tahapan desain yang nantinya akan memuat proses-proses diantaranya:

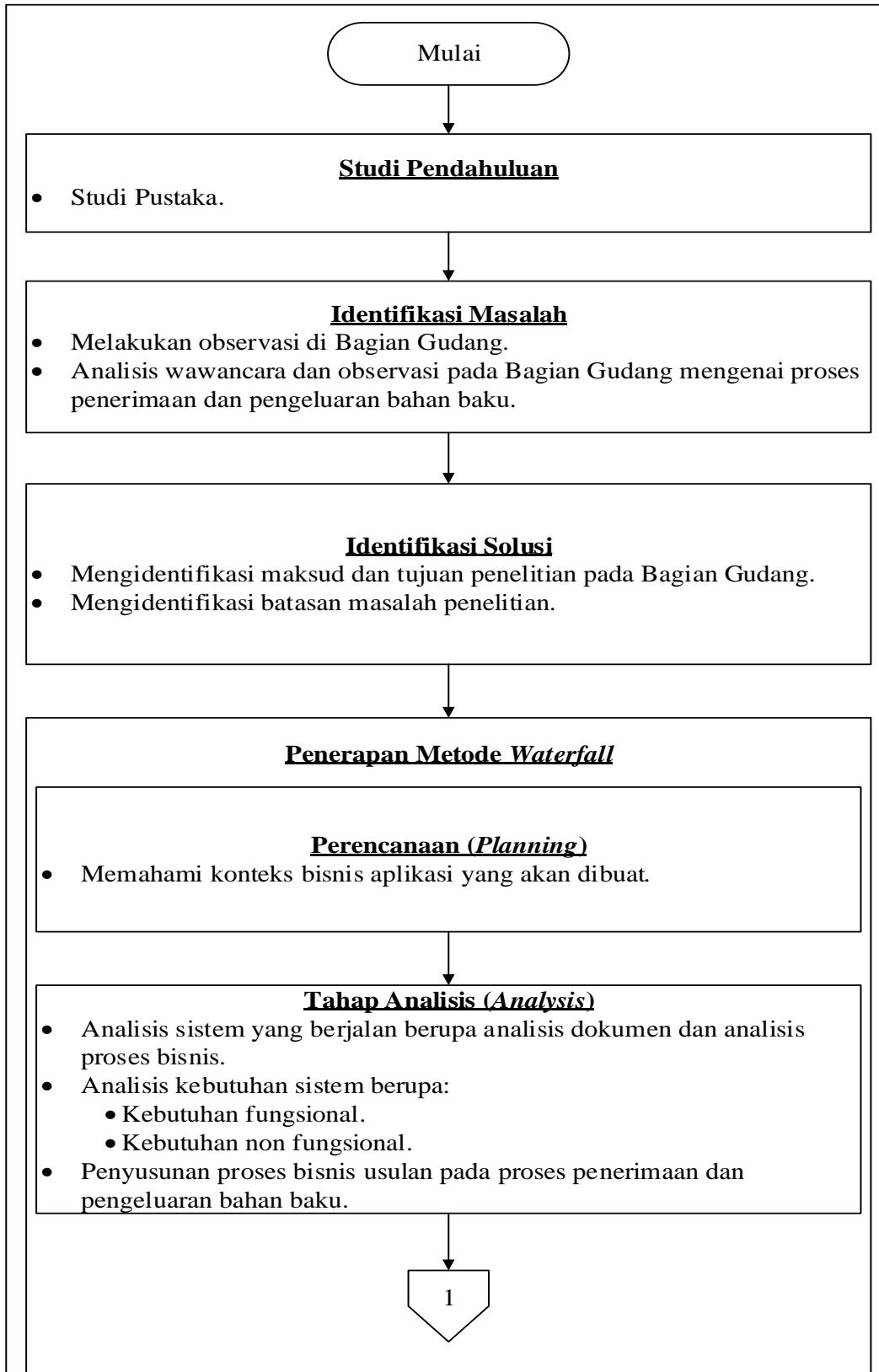
- 1) Pemodelan sistem menggunakan *Unified Modeling Language* (UML) yaitu menggunakan *usecase diagram*, *activity diagram*, *sequence diagram*, *class diagram* dan *deployment diagram*.

- 2) Pemodelan basis data dengan menggunakan *Entity Relationship Diagram* (ERD), *Conceptual Data Model* (CDM) dan kamus data.
- 3) Merancang sistem dengan menggunakan *Windows Navigation Diagram* (WND) dan rancangan antar muka.

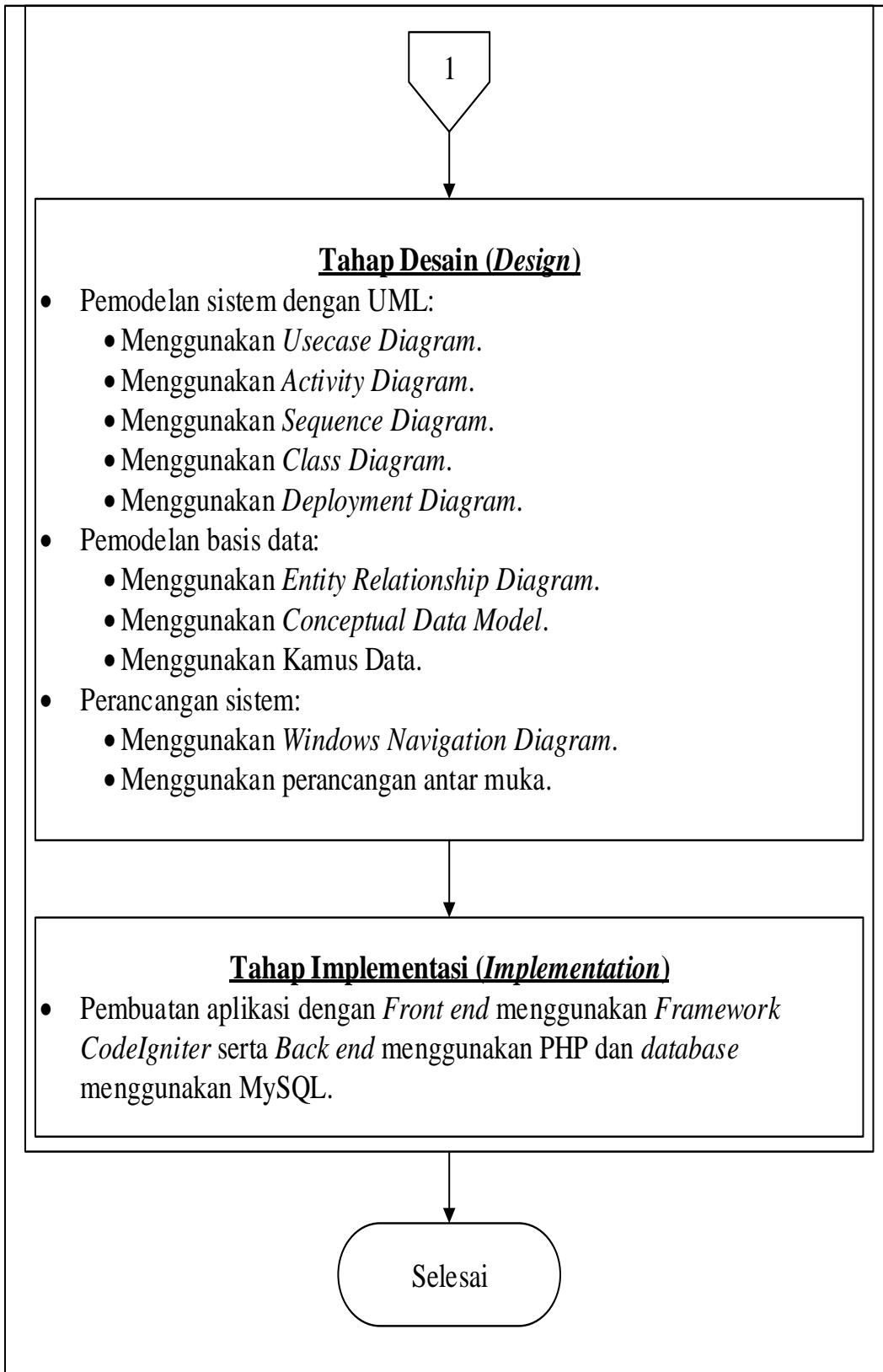
d. Implementasi (*Implementation*)

Pada tahap ini, dokumentasi dari tahap-tahap sebelumnya akan diterjemahkan ke sebuah aplikasi dengan melakukan aktivitas pengkodean untuk membuat sebuah aplikasi sistem informasi penerimaan dan pengeluaran bahan baku dengan *Front end* yang digunakan adalah *framework CodeIgniter* dan *Back end* yang digunakan yaitu PHP serta *database* yang digunakan yakni MySql. Selain itu, dilakukan sebuah pengujian dengan menggunakan *blackbox testing* yang bertujuan untuk menguji coba apakah setiap fungsi atau tombol-tombol yang terdapat pada sistem dapat berjalan sesuai yang direncanakan atau diharapkan oleh penulis.

Berikut kerangka penelitian yang digunakan dalam laporan Tugas Akhir ini yang terdapat pada gambar III.1:



Gambar III.1 Kerangka Penelitian  
 (Sumber: Pengolahan Data, 2019)



Gambar III.1 Kerangka Penelitian (lanjutan)  
(Sumber: Pengolahan Data, 2019)

## **BAB IV**

### **PENGUMPULAN DAN PENGOLAHAN DATA**

#### **4.1 Sejarah Perusahaan**

Multi Prima Sejahtera Tbk (LPIN) didirikan tanggal 07 Januari 1982 dengan nama PT Lippo Champion Glory dan memulai kegiatan usaha komersialnya pada tahun 1987. Kantor pusat LPIN berdomisili di Karawaci Office Park Blok M No. 39-50 Lippo Karawaci, Tangerang 15139, sedangkan pabriknya berlokasi di Jl. Kabupaten No. 454, Desa Tlajung Udik, Kecamatan Gunung Putri, Bogor Jawa Barat. LPIN beberapa kali melakukan perubahan nama, antara lain:

1. PT Lippo Champion Glory, per 07-Jan-1982;
2. PT Champion Spark Plug Industries, per 21-Sep-1989;
3. PT Lippo Industries (Lippo Industries Tbk), per 21-Ags-1990 (1996);
4. Lippo Enterprises Tbk, per 1997;
5. Multi Prima Sejahtera Tbk, per tahun 2001.

Pemegang saham yang memiliki 5% atau lebih saham Multi Prima Sejahtera Tbk adalah Pacific Asia Holdings Limited, Cook Islands, dengan persentase kepemilikan sebesar 25,00%. Pada tahun 1990, LPIN memperoleh pernyataan efektif dari Bapepam-LK untuk melakukan Penawaran Umum Perdana Saham LPIN (IPO) kepada masyarakat sebanyak 1.250.000 dengan nilai nominal Rp1.000,- per saham dengan harga penawaran Rp3.000,- per saham. Saham-saham tersebut dicatatkan pada Bursa Efek Indonesia (BEI) pada tanggal 17 Oktober 1994.

PT Multi Prima Sejahtera Tbk (PT MPS) merupakan perusahaan yang bergerak dalam bidang manufaktur. Kegiatan utama perusahaan adalah memproduksi busi (*spark plug*) untuk kendaraan baik mobil maupun motor. Perusahaan mengadakan perjanjian lisensi dengan Federal Mogul Pty. Ltd., (FM), Australia untuk memproduksi dan menjual busi merek “Champion”. Pada tanggal 8 Maret 2000, sejak saat itu perusahaan mempunyai perikatan dan komitmen untuk membeli komponen utama busi berbentuk insulator bermerek “Champion”.

Tentunya sebuah perusahaan mempunyai sebuah logo sebagai lambang dari perusahaannya. Desain logo perusahaan adalah pembeda visual suatu perusahaan dengan perusahaan lain. Didalam desain logo yang bagus, akan terlihat filosofi dan misi dari perusahaan tersebut. PT MPS mempunyai sebuah logo perusahaan seperti pada gambar IV.1.



Gambar IV.1 Logo Perusahaan PT Multi Prima Sejahtera Tbk  
(Sumber: PT Multi Prima Sejahtera Tbk, 2018)

#### **4.2 Profil Perusahaan**

Profil perusahaan merupakan produk tulisan praktisi Public Relation yang berisi gambaran umum perusahaan. Gambaran ini tidak sepenuhnya detail, lengkap, dan mendalam. Adapun profil dari PT Multi Prima Sejahtera Tbk adalah sebagai berikut:

Nama Perusahaan	: PT Multi Prima Sejahtera Tbk
Bidang Usaha	: Produsen Suku Cadang Kendaraan Bermotor
Tahun Bediri	: 1982
Alamat Kantor & Pabrik: ( <i>Office</i> ) Karawaci Office Park Blok M No. 39-50 Lippo Karawaci, Tangerang 15139	
(Factory) Jl. Kabupaten No. 454, Desa Tlajung Udik, Kecamatan Gunung Putri, Bogor Jawa Barat	
Telepon	: (021)867-29-09
Fax	: (021)867-28-78
E-mail	: <a href="mailto:corporatesecretary@multiprimasejahtera.net">corporatesecretary@multiprimasejahtera.net</a>
Website	: <a href="http://www.multiprimasejahtera.net/">http://www.multiprimasejahtera.net/</a>
Status Perusahaan	: Swasta Nasional
Bank	: Bank CIMB
Total Karyawan	: 117 karyawan
Luas Tanah	: 14.215 m <sup>2</sup>

Luas Bangunan	: 3.591 m <sup>2</sup>
Direktur Utama	: Drs. Lukman Djaja, MBA
Direktur Penjualan	: Drs. Lukman Djaja, MBA
Direktur Marketing	: Drs. Lukman Djaja, MBA
Direktur Asuransi	: Hartono Tjahjana. G
Direktur Akuntansi	: Hartono Tjahjana. G
Direktur Produksi	: Made Saputra Djaya

#### **4.3 Visi dan Misi Perusahaan**

Visi adalah serangkaian kata yang menunjukkan impian, cita-cita atau nilai inti sebuah organisasi, perusahaan atau instansi. Visi merupakan tujuan masa depan sebuah instansi, organisasi atau perusahaan. Visi juga adalah pikiran-pikiran yang ada didalam benak para pendiri. Pikiran-pikiran tersebut adalah gambaran tentang masa depan yang ingin dicapai.

Misi adalah tahapan-tahapan yang harus dilalui untuk mencapai visi tersebut. Selain itu, misi juga merupakan deskripsi atau tujuan mengapa perusahaan, organisasi atau instansi tersebut berada di tengah-tengah masyarakat. PT Multi Prima Sejahtera Tbk merupakan perusahaan industri yang melayani produksi barang komponen otomotif yaitu busi kendaraan bermotor. PT Multi Prima Sejahtera Tbk mempunyai Visi dan Misi sebagai berikut:

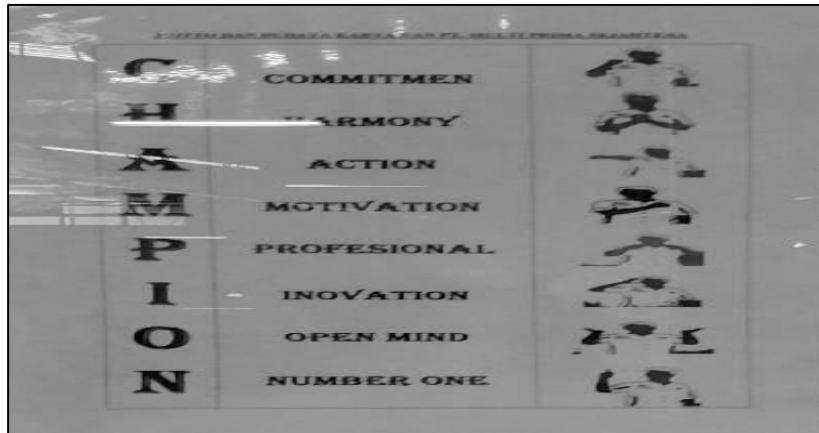
Visi : Menjadi perusahaan manufaktur dan trading yang professional dengan memperdayakan keahlian local dan senantiasa mencari peluang usaha guna meningkatkan nilai tambah bagi perseroan.

Misi : Menjaga kualitas produk dan pelayanan untuk mencapai kepuasan pelanggan dengan SDM yang berkualitas.

#### **4.4 Moto Karyawan**

Moto atau dalam bahasa Inggris dituliskan dengan moto, merupakan sebuah kalimat ataupun kata yang dijadikan sebagai prinsip dan semboyan dalam kehidupan. Moto adalah kata-kata atau seruan yang menggambarkan motivasi, semangat, dan tujuan dari seseorang atau individu.

PT Multi Prima Sejahtera Tbk memiliki moto dan budaya karyawan yang diterapkan di setiap masing masing individu. Moto PT Multi Prima Sejahtera Tbk bisa kita lihat pada gambar IV.2.



Gambar IV.2 Moto dan Budaya Karyawan PT Multi Prima Sejahtera Tbk  
(Sumber: PT Multi Prima Sejahtera Tbk, 2018)

1. *Commitment* (Komitmen)

Suatu keadaan dimana seseorang membuat perjanjian (keterikatan), baik kepada diri sendiri maupun kepada orang lain yang tercermin dalam tindakan/perilaku tertentu yang dilakukan secara sukarela maupun terpaksa.

2. *Harmony* (Harmoni)

Harmoni adalah kerja sama antara berbagai faktor dengan sedemikian rupa hingga faktor-faktor tersebut dapat menghasilkan suatu kesatuan yang luhur.

3. *Action* (Aksi)

Aksi adalah tindakan yang dilakukan karyawan perusahaan tersebut.

4. *Motivation* (Motivasi)

Motivasi adalah proses yang menjelaskan intensitas, arah, dan ketekunan seorang individu untuk mencapai tujuannya.

5. *Professional* (Profesional)

Profesional adalah orang yang memiliki profesi atau pekerjaan yang dilakukan dengan memiliki kemampuan yang tinggi dan berpegang teguh kepada nilai moral yang mengarahkan serta mendasari perbuatan.

6. *Innovation* (Inovasi)

Inovasi adalah suatu proses dan/ atau hasil pengembangan pemanfaatan suatu produk/ sumber daya yang telah ada sebelumnya, sehingga memiliki nilai yang lebih berarti.

7. *Open Mind* (Pikiran Terbuka)

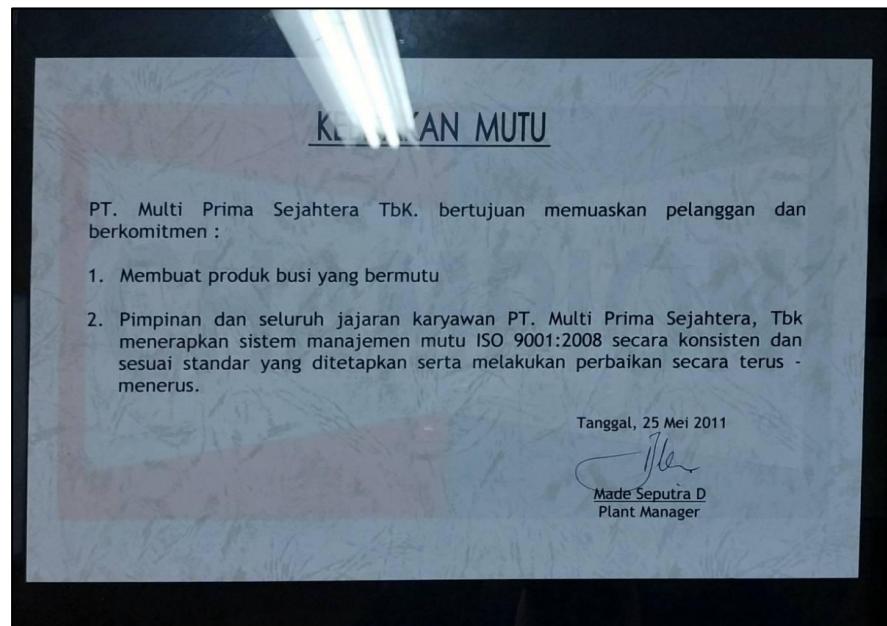
Pikiran terbuka adalah ketika anda tahu bahwa anda benar, tetapi anda juga tahu anda bisa saja salah dan mau untuk mendengar opini atau sudut pandang orang lain yang berbeda maupun yang tidak sesuai dengan pemikiran anda.

8. *Number One* (Nomor Satu)

Nomor satu adalah angka pertama pada bilangan. Makna nomor satu pada moto PT Multi Prima Sejahtera Tbk adalah semua moto yang sudah disebutkan di atas dijadikan nomor satu atau dijadikan moto penting pada PT Multi Prima Sejahtera Tbk.

#### **4.5 Kebijakan Mutu**

PT Multi Prima Sejahtera Tbk memiliki mutu jaminan atas barang produksi yang bisa kita lihat pada gambar IV.3 berikut ini:



Gambar IV.3 Kebijakan Mutu PT Multi Prima Sejahtera Tbk  
 (Sumber: PT Multi Prima Sejahtera Tbk, 2018)

#### 4.6 Jam Kerja dan Jumlah Karyawan

Tenaga kerja adalah setiap orang yang mampu melakukan pekerjaan guna menghasilkan barang atau jasa untuk memenuhi kebutuhan sendiri maupun masyarakat. Waktu kerja di PT Multi Prima Sejahtera Tbk yaitu 5 hari kerja dalam seminggu (Senin-Jumat). PT Multi Prima Sejahtera Tbk memberlakukan sistem *shift* yang terbagi menjadi 3 yaitu Non *Shift*, *Shift 1*, *Shift 2* di dalam produksinya. Jam kerja karyawan terdapat pada tabel IV.1 berikut ini, sedangkan jumlah karyawan pada PT Multi Prima Sejahtera Tbk terdapat pada tabel IV.2 berikut ini:

Tabel IV.1 Jam Kerja Karyawan PT Multi Prima Sejahtera Tbk

Hari	Jenis	Masuk	Pulang
Senin Sampai Jumat	<i>Non Shift</i>	07.00	16.10
	<i>Shift 1</i>	06.00	15.10
	<i>Shift 2</i>	03.00	24.00

(Sumber: PT Multi Prima Sejahtera Tbk, 2018)

Tabel IV.2 Jumlah Karyawan PT Multi Prima Sejahtera Tbk

No	Nama Jabatan	Karyawan (orang)
1	<i>Plant Manager</i>	1
2	<i>Management Representative</i>	1
3	<i>Sistem Support</i>	1
4	<i>Customer Service</i>	1
5	Pengendali Dokumen	1
6	<i>Internal Lead Auditor</i>	1
7	Divisi Purchasing	1
8	Divisi Gudang	8
9	Divisi Quality Assurance	7
10	Divisi Maintenance	14
11	Divisi Produksi	71
12	Divisi Personalia & GA	17
Jumlah Karyawan		124

(Sumber: PT Multi Prima Sejahtera Tbk, 2018)

#### **4.7 Lokasi Perusahaan**

Lokasi merupakan salah satu kegiatan awal yang harus dilakukan sebelum perusahaan mulai beroperasi. Penentuan lokasi yang tepat akan mempengaruhi kemampuan perusahaan dalam melayani konsumen, mendapatkan bahan-bahan mentah yang cukup, mendapatkan tenaga kerja dengan mudah, serta memungkinkan diadakannya perluasan usaha.

Kesalahan dalam pemilihan lokasi akan mengakibatkan biaya transportasi yang tinggi, kekurangan tenaga kerja, kehilangan kesempatan dalam bersaing, tidak tersedianya bahan baku yang cukup, dan sebagainya. PT Multi Prima Sejahtera Tbk memiliki kantor pusat, kantor pemasaran, dan pabrik yang masing-masing berada di lokasi yang berbeda. Berikut adalah informasi mengenai lokasi perusahaan saat ini, yaitu:

1. Kantor Pusat

Karawaci Office Park Blok M, No. 39/50, LippoKarawaci, Tanggerang  
15139 Telp/fax: (021) 5589 767/(021) 5589 823;

2. Kantor Pemasaran

Gedung Bank Lippo Lantai 2, Jln Kebon Sirih Raya No. 33, Jakarta  
10340 Telp/fax: (021) 319 00857, 315 2042 / (021) 2300450;

3. Pabrik

Jln Kabupaten No. 454, Desa Tlajung Udik, Kecamatan Gunung Putri,  
Kabupaten Bogor, Jawa Barat–Indonesia Telp/fax: (021) 867 2909/(021)  
867 2878.

PT Multi Prima Sejahtera Tbk mendirikan pabrik yang berlokasi di Gunung Putri–Bogor. Pabrik tersebut dibangun di atas luas tanah 14.215 m<sup>2</sup> dengan luas bangunan pabrik sebesar 3.591 m<sup>2</sup>. Kapasitas produksi yang dapat dihasilkan oleh PT Multi Prima Sejahtera setiap tahunnya adalah sebesar 9.000.000 pcs/tahun. Mesin proses produksi yang ada di pabrik terdiri dari 2 (dua) bagian, yaitu:

1. *Manufacturing*

- *Cold Forming Machine (NEP)*
- *Multi Spinder Bar Machine (ASM)*

- *Upright Side Wire Welder (USW)*
  - *Reed Thread Roller Machine (RRT)*
  - *Counter Bor (CB)*
2. *Assembling*
- *Core Tamping Machine (CTM)*
  - *Plug Tamping Machine (PT)*
  - *Gasket Crimping Machine (GCF)*
  - *Oil Spark Test (OST)*
  - *Bending Trim and Gap (BTG)*

#### **4.8 Identitas Anak Perusahaan dan Asosiasi**

PT Multi Prima Sejahtera Tbk memiliki anak perusahaan dan asosiasi yang memiliki saham terbesar dalam suatu perusahaan yang mempunyai kepemilikan langsung. Tabel pemegang saham terdapat pada tabel IV.3 berikut ini:

Tabel IV.3 Tabel Pemegang Saham PT Multi Prima Sejahtera Tbk

Nama Perusahaan	Bidang Usaha	Kepemilikan	Status Operasi	Alamat
PT Metropolitan Sinar Indah	Perdagangan Umum dan Investasi	100%	Tidak Beroperasi	Karawaci Office Park Blok M No, 39-50 Lippo Karawaci, Tangerang 15139
PT Multi Usaha Wisesa	Perdagangan Umum dan Investasi	100%	Tidak Beroperasi	Gedung CIMB Niaga Lt. 2 Jln. Kebon Sirih Raya No. 33, Jakarta Pusat 10340
PT Champion Multi Usaha	Perdagangan Umum dan Investasi	100%	Tidak Beroperasi	Gedung CIMB Niaga Lt. 2 Jln. Kebon Sirih Raya No. 33, Jakarta Pusat 10340

Tabel IV.3 Tabel Pemegang Saham PT Multi Prima Sejahtera Tbk (lanjutan)

Nama Perusahaan	Bidang Usaha	Kepemilikan	Status Operasi	Alamat
PT Cipta Selaras Majujaya	Perdagangan Umum dan Investasi	100%	Tidak Beroperasi	Karawaci Office Park Blok L No. 55 Lippo Karawaci, Tangerang 15139
PT Karya Indah Selaras Jaya	Perdagangan Umum dan Investasi	100%	Tidak Beroperasi	Karawaci Office Park Blok L No. 55 Lippo Karawaci, Tangerang 15139
PT Cipta Global Internasional	Perdagangan Umum dan Investasi	100%	Tidak Beroperasi	Karawaci Office Park Blok L No. 55 Lippo Karawaci, Tangerang 15139
PT Walsin Lippo Industries	Manufaktur Wire	30%	Beroperasi	Jln. M.H. Thamrin Blok A1 No. 1 Delta Silicon Industrial Park, Lippo Cikarang, Bekasi 17650
PT Walsin Lippo Kabel	Perindustrian Umum	30%	Tidak Beroperasi	Jln. M.H. Thamrin Blok A1 No. 1 Delta Silicon Industrial Park, Lippo Cikarang, Bekasi 17650

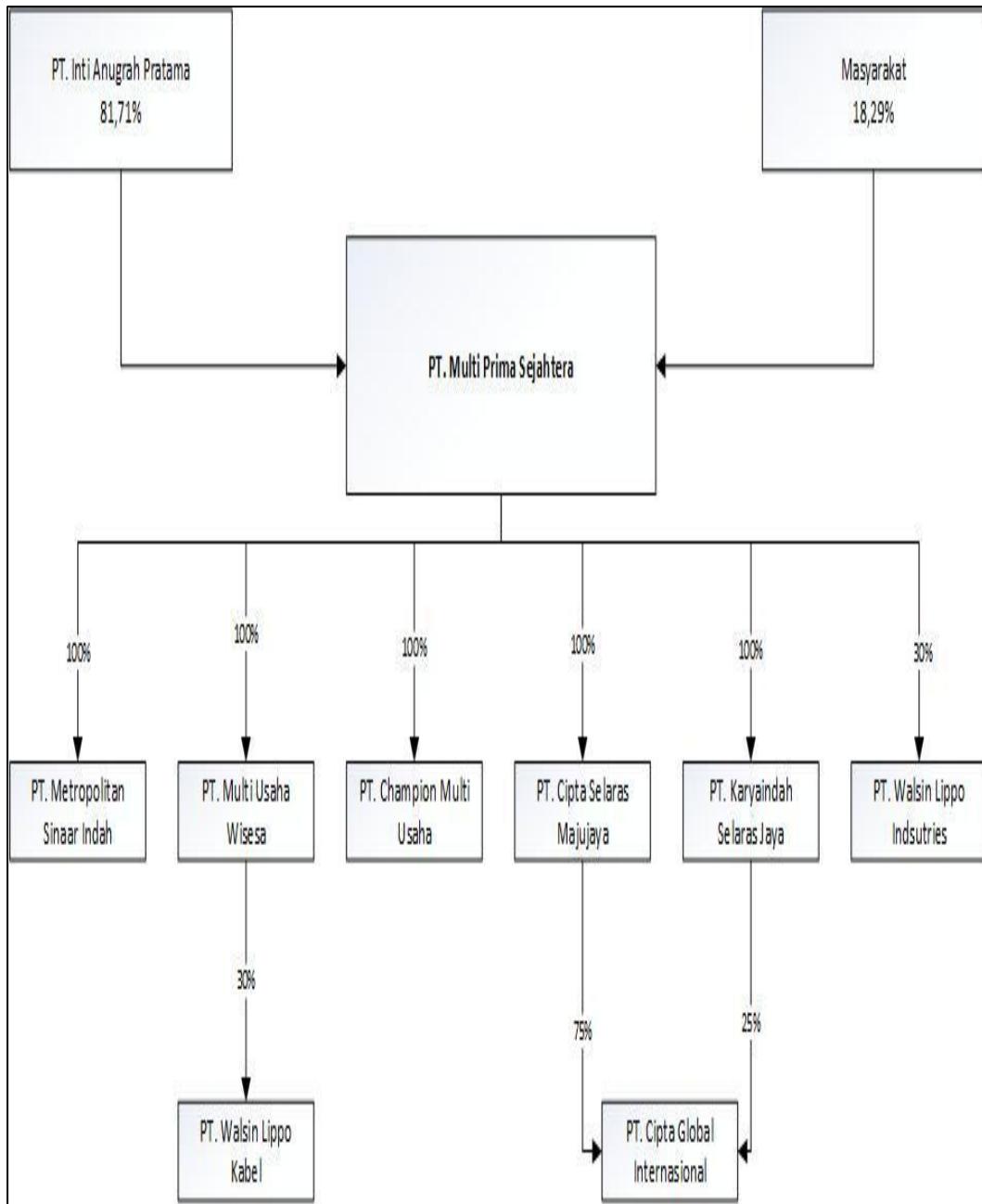
(Sumber: PT Multi Prima Sejahtera Tbk, 2018)

#### 4.9 Struktur Kepemilikan Saham

Struktur kepemilikan dapat berupa investor individual, pemerintah, dan institusi swasta. PT Multi Prima Sejahtera memiliki beberapa pemegang saham yang membangun perusahaan berdiri. PT Multi Prima Sejahtera Tbk mempunyai

sebuah struktur kepemilikan yang sudah ada dan berfungsi untuk merincikan beberapa perusahaan yang memiliki saham pada PT MPS.

Berdasarkan struktur kepemilikan tersebut PT MPS diharapkan lebih transparan dan lebih berkembang di bidang otomotif atau lebih signnifikan yaitu dalam memproduksi busi kendaraan bermotor. Pada gambar IV.4 dibawah ini menjelaskan struktur kepemilikan PT MPS yang sudah ada.



Gambar IV.4 Struktur Kepemilikan Saham PT Multi Prima Sejahtera Tbk  
(Sumber: PT Multi Prima Sejahtera Tbk, 2018)

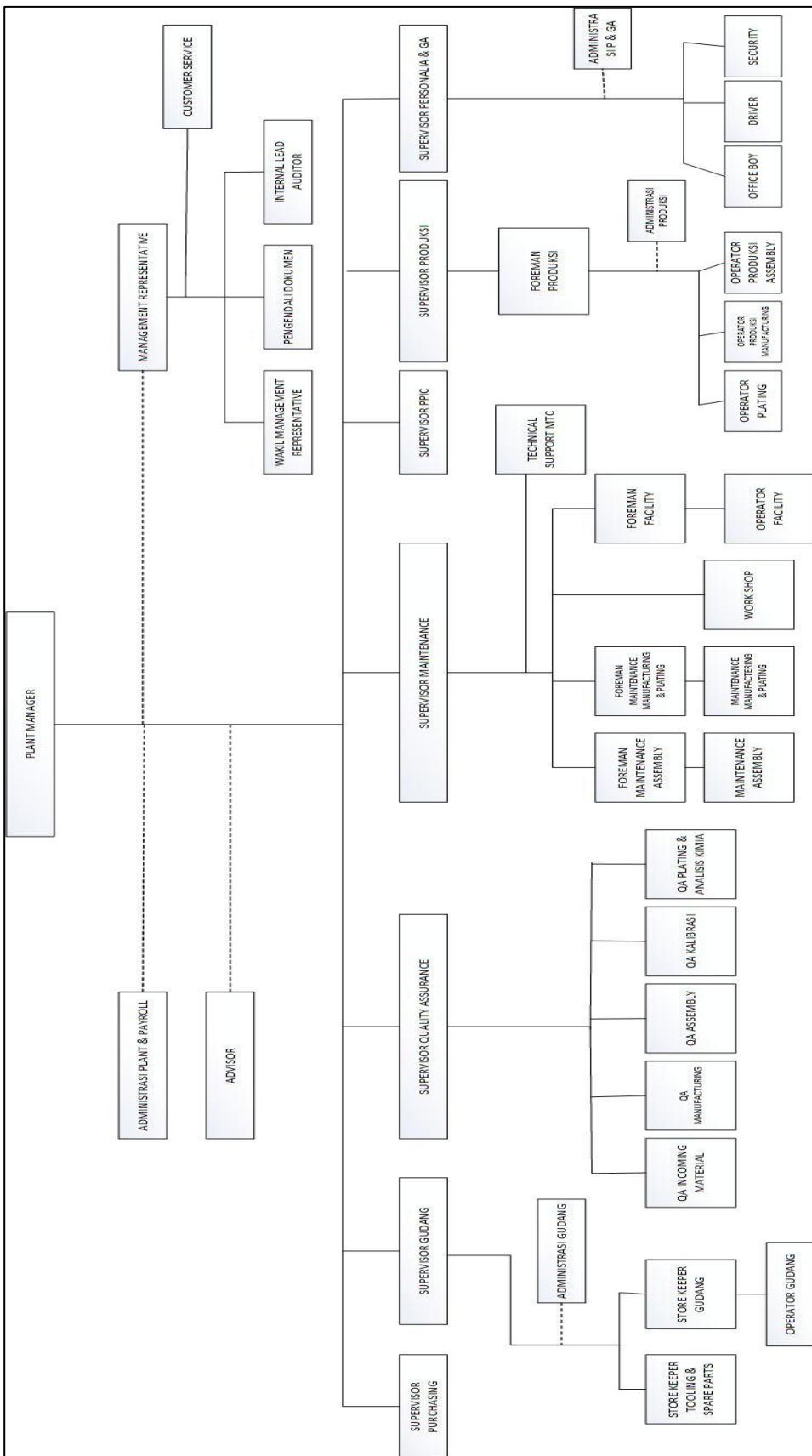
#### **4.10 Struktur Organisasi**

Roda perusahaan dapat bergerak secara efektif dan efisien, jika setiap komponen dalam perusahaan tersebut berfungsi secara optimal. Oleh karena itu, pimpinan perusahaan harus berupaya untuk membagi tugas dan menempatkan semua sumber daya perusahaan, khususnya SDM, dalam posisi yang tepat sesuai bidang keahlian masing-masing. Hal ini menjadikan setiap individu yang terdapat dalam perusahaan tersebut memiliki gambaran jelas mengenai kedudukan, fungsi, hak dan kewajibannya.

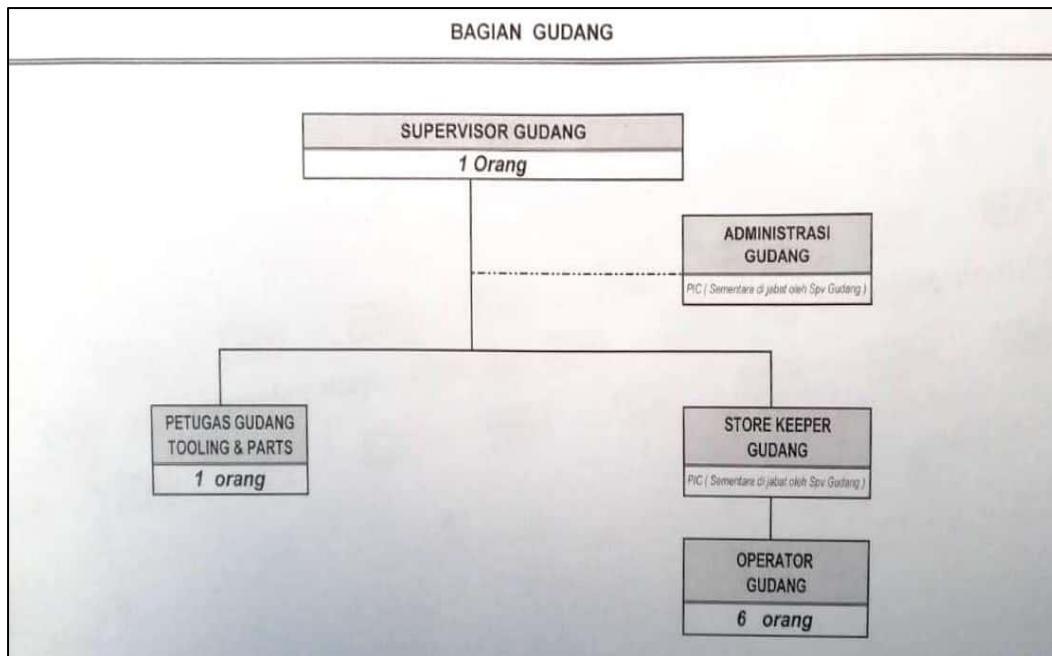
Selain itu, puncak pimpinan juga bisa mengetahui dengan mudah komponen mana yang memiliki kinerja, fungsi dan peran yang tak sesuai harapan perusahaan. Dengan demikian, akan lebih cepat untuk mengambil sebuah tindakan keputusan yang terbaik bagi perusahaan. Struktur organisasi perusahaan adalah sebuah garis hierarki (bertingkat) yang mendeskripsikan komponen-komponen yang menyusun perusahaan dimana setiap individu (sumber daya manusia) yang berada pada lingkup perusahaan tersebut memiliki posisi dan fungsi masing-masing.

Struktur organisasi menggambarkan dengan jelas pemisahan kegiatan pekerjaan antara yang satu dengan yang lain dan bagaimana hubungan aktivitas dan fungsi dibatasi. Dalam struktur organisasi yang baik harus menjelaskan hubungan wewenang siapa melapor kepada siapa, jadi ada satu pertanggungjawaban apa yang akan dikerjakan. PT Multi Prima Sejahtera Tbk memiliki struktur organisasi secara menyeluruh dan bagian pabrik.

Adapun gambar IV.5 berikut ini merupakan struktur organisasi pabrik PT Multi Prima Sejahtera Tbk dan pada gambar IV.6 berikut adalah struktur organisasi Bagian Gudang PT Multi Prima Sejahtera Tbk.



Gambar IV.5 Struktur Organisasi Pabrik PT Multi Prima Sejahtera Tbk  
 (Sumber: PT Multi Prima Sejahtera Tbk, 2018)



Gambar IV.6 Struktur Organisasi Bagian Gudang  
(Sumber: PT Multi Prima Sejahtera Tbk, 2018)

#### 4.11 *Job Description*

*Job Description* merupakan sebuah kumpulan informasi jabatan dan disusun secara sistematis yang dapat mengidentifikasi dan menguraikan suatu jabatan atau posisi tertentu. *Job Description* membuat status setiap jabatan menjadi jelas akan fungsi, peran, hasil, dan tanggung jawabnya. Pada struktur organisasi PT Multi Prima Sejahtera Tbk, maka *Job Description* dari masing-masing bagian sebagai berikut:

##### 1. *Supervisor Gudang*

Tugas:

- Pengelola dan penanggung jawab aktivitas di dalam gudang.
- Bertanggung jawab atas ketepatan laporan penerimaan barang.
- Bertanggung jawab atas ketepatan laporan pengeluaran barang.
- Bertanggung jawab atas barang masuk ataupun keluar.
- Mengawasi kinerja *staff* gudang.

##### 2. *Administrasi Gudang*

Tugas:

- Mengelola administrasi dan dokumentasi gudang.

- b. Menerima surat jalan dari *supplier*.
  - c. Membuat laporan mengenai barang masuk ataupun keluar gudang.
3. Petugas Gudang *Tooling and Parts/Store Keeper Tooling and Parts*
- a. Mencatat semua peralatan dan komponen-komponen yang berhubungan dengan perusahaan yang ada di gudang.
  - b. Menjaga semua peralatan dan komponen-komponen yang berhubungan dengan perusahaan yang ada di gudang.
  - c. Merawat semua peralatan dan komponen-komponen yang berhubungan dengan perusahaan yang ada di gudang.
4. *Store Keeper* Gudang
- a. Mencatat semua barang-barang yang berhubungan dengan perusahaan yang ada di gudang.
  - b. Menjaga semua barang-barang yang berhubungan dengan perusahaan yang ada di gudang.
  - c. Merawat semua barang-barang yang berhubungan dengan perusahaan yang ada di gudang.
5. Operator Gudang
- a. Menerima barang masuk ke gudang ataupun barang keluar dari gudang.
  - b. Mengambil barang/menyiapkan barang yang akan dikirim dari gudang.
  - c. Meletakkan barang pada tempat yang ditentukan.

#### **4.12 Produk Perusahaan**

Kata produk berasal dari bahasa Inggris *product* yang berarti "sesuatu yang diproduksi oleh tenaga kerja atau sejenisnya". Bentuk kerja dari kata *product*, yaitu *produce*, merupakan serapan dari bahasa latin *prōdūce(re)*, yang berarti (untuk) memimpin atau membawa sesuatu untuk maju. Dalam manufaktur, produk dibeli dalam bentuk barang mentah dan dijual sebagai barang jadi. Produk yang berupa barang mentah seperti metal atau hasil pertanian sering pula disebut sebagai komoditas. PT Multi Prima Sejahtera Tbk memproduksi 2 jenis busi yaitu busi motor dan busi mobil. Produk busi ini terdapat pada tabel IV.4 di bawah ini.

Tabel IV.4 Daftar Produk PT Multi Prima Sejahtera Tbk

No	Tipe	Model Kendaraan	Gambar
1	Z9Y/ P-Z11HC	<ul style="list-style-type: none"> <li>• Yamaha (Nouvo, Mio, Vega, Crypton, Jupiter)</li> <li>• Honda (Astrea, Win 100, Fit S, Supra X, Revo)</li> <li>• Suzuki (New Smash, Shogun, Spin, Titan, Skydrive, Arashi)</li> <li>• Kawasaki (Blitz, Kaze, Zone, ZX)</li> <li>• Kanzen (Pesona, Scudetto, Taurus)</li> <li>• Kymco ( Easy, Free MX, Libero, Metica, Spike, Trend, Visa)</li> </ul>	
2	L95YC/ PL95YC	<ul style="list-style-type: none"> <li>• Yamaha (Fiz-R)</li> <li>• Suzuki (Econos, RC Bravo, TRS, TRZ)</li> <li>• Suzuki Mobil (Carry ST 20)</li> <li>• Piaggio ( Corsa, Super, Sprint)</li> </ul>	
3	A8YC	<ul style="list-style-type: none"> <li>• Yamaha (Scorpio Z)</li> <li>• Honda (GL, CB, CG, GL Max, Mega Pro)</li> <li>• Suzuki (Thunder 250)</li> </ul>	
4	RG4HC	<ul style="list-style-type: none"> <li>• Honda (Beat, Scoopy, Supra X)</li> </ul>	
5	PN 7 YC	<ul style="list-style-type: none"> <li>• Suzuki (Satria R, RGR 150,)</li> <li>• Kawasaki (Ninja R, Ninja RR)</li> </ul>	

Tabel IV.4 Daftar Produk PT Multi Prima Sejahtera Tbk (lanjutan)

No	Tipe	Model Kendaraan	Gambar
6	P10Y	• Honda (C90)	
7	PRZ9HC (Tipe Lama)	• Suzuki (Shogun 125, New Shogun, Spin, Skydrive, Arashi)	
8	RG6YC	• Yamaha (Jupiter MX 135LC, Mio Sporty, Vixion) • Honda (Blade, CBR 150R, NSR 150R, Absolut Revo, Supra X, CS 1, Kirana) • Suzuki (Satria F150, Suzuki FXR, Thunder 125)	
9	N12YC	• Mitsubishi (Le Dan Gan 1.8 DOHC, Kuda 1.6, Jet Star) • Suzuki Mobil (Carry ST100, Katana, Swift, Forsa, Karimun) • Toyota (Land Cruiser, Kijang 2.0, Corona RX50, Crown, Corolla) • Daihatsu (Hijet 55 WIDE, Zebra, Ceria, Charage 1.0 XG) • Honda Mobil (Civic Excelent, Accord Elegent)	
10	RHG4HC	• Honda (Vario)	

Tabel IV.4 Daftar Produk PT Multi Prima Sejahtera Tbk (lanjutan)

No	Tipe	Model Kendaraan	Gambar
11	RN9HC	<ul style="list-style-type: none"> <li>• Mitsubishi (Colt L300, Pajero, Gan 1.6, Lancer LA, Lancer GLX, Galant Sigma, Galant 2.9, Eterna 2.0 SOHC)</li> <li>• Suzuki (Forsa Emenity, Esteem 1.3, Vitara, Escudo, Karimun Estilo)</li> <li>• Toyota (Starlet Ep71, Corolla EE80)</li> <li>• Nissan (Terano, New Terano, Terano Patrol)</li> <li>• Honda (Accord Excutive 1.6)</li> </ul>	
12	RC12YC	<ul style="list-style-type: none"> <li>• Mitsubishi (Le Dan Gan 1.5, Galant V6 2.0, Galant VR 2.0, Kuda 2.0, Galant 11 1.8)</li> <li>• Suzuki (APV, Carry 1.5, Grand Escudo, Aerio, Grand Vitara, New Swift)</li> <li>• Toyota (Vios 1.5, New Vios, Yaris, Avanza, Corolla AE)</li> <li>• Nissan (Sentra, Infinity)</li> </ul>	

(Sumber: PT Multi Prima Sejahtera Tbk, 2018)

#### 4.13 Bahan Baku Perusahaan

Bahan baku merupakan salah satu unsur yang paling aktif di dalam perusahaan yang secara terus menerus diperoleh, diubah yang kemudian diolah kembali. Tanpa bahan baku proses produksi tidak akan berjalan. PT Multi Prima Sejahtera Tbk mendapatkan bahan baku (*raw material*) dari pemasoknya sebelum diolah. Tabel IV.5 dibawah ini adalah daftar bahan baku pembuatan produk yang ada di PT Multi Prima Sejahtera Tbk.

Tabel IV.5 Daftar Bahan Baku PT Multi Prima Sejahtera Tbk

No	Bahan Baku	Pemasok	Gambar Bahan Baku
1	<i>Nickel Wire</i>	Federal Mogul China	
2	Insulator	Federal Mogul China	
3	<i>Center Electrode</i>	Federal Mogul China	
4	<i>Spring</i>	Federal Mogul China	
5	Resistor	Federal Mogul China	
6	Terminal Stud	Federal Mogul China, PT Ginsa Inti Pratama	

Tabel IV.5 Daftar Bahan Baku PT Multi Prima Sejahtera Tbk (lanjutan)

No	Bahan Baku	Pemasok	Gambar Bahan Baku
7	<i>Inside Gasket</i>	Federal Mogul China	
8	<i>Outside Gasket</i>	Federal Mogul China, Copper & Automotive Washer CO. LTD	
9	Terminal Nut	Federal Mogul China	

(Sumber: PT Multi Prima Sejahtera Tbk, 2018)

#### 4.14 *Layout Pabrik*

Tata letak pabrik merupakan suatu landasan utama dalam dunia industri. Tata letak pabrik yang terencana dengan baik akan ikut menentukan efisiensi dan efektivitas kegiatan produksi dan dalam beberapa hal akan juga menjaga kelangsungan hidup atau keberhasilan suatu perusahaan.

*Layout Pabrik* adalah cara penempatan fasilitas-fasilitas produksi guna memperlancar proses produksi yang efektif dan efisien. Fasilitas pabrik dapat berupa mesin-mesin, alat-alat produksi, alat pengangkutan bahan, dan peralatan pengawasan. PT Multi Prima Sejahtera Tbk memiliki *layout* pabrik seperti gambar IV.7 berikut ini.



Gambar IV.7 Layout Pabrik PT Multi Prima Sejahtera Tbk  
 (Sumber: PT Multi Prima Sejahtera Tbk, 2018)

#### 4.15 Daftar Pelanggan

Pelanggan pada dasarnya berasal dari konsumen biasa yang mencoba menggunakan jasa ataupun produk dari sebuah perusahaan. Pelanggan terbentuk dari pola kerja sama saling menguntungkan yang terjadi dalam proses kerja sama antara penyedia layanan dan pengguna layanan. PT Multi Prima Sejahtera Tbk adalah perusahaan yang sangat mengutamakan kepuasan pelanggan. Daftar pelanggan PT Multi Prima Sejahtera Tbk bisa dilihat pada tabel IV.6 berikut ini.

Tabel IV.6 Daftar Nama Pelanggan PT Multi Prima Sejahtera Tbk

No	Kode	Nama Pelanggan	Alamat	Kontak
1	AM163	Aneka Motor	Jln. Wonogiri Sari Barat No. 498 Pring Sewu, Lampung	0729-370434
2	ASM08	Aneka Sakti Motor	Jln. M.T. Haryono No. 301 Semarang	024-3546354
3	BJM17	Berkah Jaya Motor	Jln. Raya Cimanuk No. 101 A Jaya Raga, Garut	0262-232336
4	BM096	Baru Motor	Jln. A. Yani No.14 Purwodadi	0292-421160
5	CH001	Cahaya Hasil	Jln. Pawiyatan No. 7 Surabaya	031-70993728
6	CM089	Cahaya Motor	Jln. Raya Banjar No. 220 Parung Sari, Banjar, Jawa Barat	0265-741312
7	DAM03	Ani Motor	Jln. Raya Timur Pedan, Klaten	0272-897374
8	DM085	Devi Motor	Jln. Raya Tambak Boyo Ambarawa, Semarang	0298-594878
9	EM008	Eta Motor	Jln. A. R. Hakim No. 85, Karawang	0267-401142
10	ES004	Era Sehati 1 Motor	Jln. Raya Tayu Km. 7 Ngablak, Jepara	0295-4545400
11	FS002	Felix Sunarso	Jln. Melong Asih No. 61 Bandung, Jawa Barat	022-6016678
12	FTR01	Bambang S	Jln. Soekarno Hatta, Pasuruan	0343-424227
13	G0008	Gunanto	Jln. Brigjen Katamso No. 264 Yogyakarta	0274-371466
14	G0007	Gemini Motor	Jln. Panglima Sudirman 114 Mojokerto	0321-322606
15	HJM08	Harapan Jaya Motor	Jln. Seth Adjji No. 11 A Palangkaraya	0511-704445
16	IM040	Indo Motor	Jln. Raya Tuparev No. 50 C	0231-3305000
17	JM070	Jaya Mulya	Jln. Kyai Tamin No. 130 Malang	0341-326069
18	KM114	King Motor	Jln. Kantor Pos No. 46 Banjar	0265-742675
19	LM048	Lely Motor	Jln. Pemuda No. 37 Blora	0296-531742

Tabel IV.6 Daftar Nama Pelanggan PT Multi Prima Sejahtera Tbk (lanjutan)

No	Kode	Nama Pelanggan	Alamat	Kontak
20	MM080	Medan Motor	Jln. HOS Cokroaminoto No. 2 Cianjur	0263-263714
21	NMN01	Naga Mas Motor	Jln. Panglima Sudirman No. 136 Lumajang	0334-881988
22	OM011	Osaka Motor	Jln. Nusantara A10, Jember	0331-488626
23	P0010	PT Hyundai Indonesia Motor	Jln. Kebon Sirih 17-19 Jakarta	021-8854220
24	R0010	Rasa	Jln. Wachid Hasyim No. 8 Lumajang	0334-881415
25	MM174	Merapi Motor	Jln. Sumbing No. 29B Wonosobo	0286-323967
26	HSM03	PT Hyundai Sukses Mandiri	Jln. Jendral Sudirman No. 633- 639 Bandung	022-6002920
27	CV012	CV Semangat Baru	Jln. Ratna Gang Kumala No. 2 Bali	0361-222709
28	PO146	PT Mega Anugrah Mandiri	Jln. Gaharu No. 86A Medan	061-6610673

(Sumber: PT Multi Prima Sejahtera Tbk, 2018)

#### 4.16 Analisis Sistem Berjalan

Analisis sistem yang berjalan pada Bagian Gudang dilakukan dengan teknik pengumpulan data untuk mengidentifikasi penyebab masalah yang terjadi pada sistem informasi penerimaan dan pengeluaran bahan baku Bagian Gudang dengan melakukan wawancara terhadap karyawan dan observasi pada Bagian Gudang PT Multi Prima Sejahtera Tbk.

Sistem yang berjalan pada Bagian Gudang masih dilakukan secara manual dengan *form-form* yang ditulis tangan oleh karyawan. Beberapa tahapan dalam analisis sistem informasi penerimaan dan pengeluaran bahan baku Bagian Gudang PT Multi Prima Sejahtera Tbk adalah sebagai berikut:

1. Analisis Dokumen;
2. Analisis Proses Bisnis Penerimaan Dan Pengeluaran Bahan Baku.

#### 4.16.1 Analisis Dokumen

Analisis dokumen berikut akan menjelaskan dokumen-dokumen apa saja yang digunakan dalam proses penerimaan dan pengeluaran bahan baku pada Bagian Gudang. Adapun dokumen-dokumen yang terkait dalam proses bisnis Bagian Gudang diantaranya adalah:

1. Surat Jalan

Surat Jalan merupakan dokumen yang didapatkan dari *supplier*. Berperan sebagai dokumen penyerahan bahan baku yang telah dipesan dengan kesepakatan bersama. Surat Jalan dibuat *supplier* berdasarkan *purchase order* yang telah diterima dari konsumen (dapat dilihat pada gambar IV.8).

PT. GINSA INTI PRATAMA				
BELI:				
PT. MULTI PRIMA SEJAHTERA JALAN RAYA TELAJUNG UDIK NO. 454 GUNUNG PUTRI BOGOR	TANGGAL : 20-Feb-13	NO. SURAT JALAN : SO.GINSA/1302/00476	NO. PO : 377/MPS/12-12	NO. MOBIL : B-9435-CQB
NO	JENIS BARANG	QTY (PCS)	KETERANGAN	
1	TERMINAL STUD.M4X4,5X16	16,215		
TANDA TERIMA			HORMAT KAMI	
 <b>PT. MULTI PRIMA SEJAHTERA Tbk</b>			 <b>PT. GINSA INTI PRATAMA</b>	

Gambar IV.8 Surat Jalan  
(Sumber: PT Multi Prima Sejahtera Tbk, 2018)

Keterangan dari *field-field* pada gambar IV.8 adalah:

- No. Surat Jalan: berisi nomor surat jalan
- No. PO: berisi nomor *purchase order*
- No. Mobil: berisi Tanda Nomor Kendaraan Bermotor pengirim
- No: berisi nomor urut dokumen

- Jenis Barang: berisi jenis bahan baku yang dibeli
  - Qty (Kg): berisi banyaknya bahan baku dalam satuan kilogram
  - Keterangan: berisi keterangan dari bahan baku

## 2. *Form Laporan Penerimaan Barang*

*Form* Laporan Penerimaan Barang merupakan dokumen yang terdapat di Bagian Gudang. Berperan sebagai laporan pendataan bahan baku yang sudah diterima. *Form* Laporan Penerimaan Barang dibuat berdasarkan kebutuhan pendataan pada Bagian Gudang (dapat dilihat pada gambar IV.9).

Gambar IV.9 *Form Laporan Penerimaan Barang*  
(Sumber: PT Multi Prima Sejahtera Tbk, 2018)

Keterangan dari *field-field* pada gambar IV.9 adalah:

- No. Form: berisi nomor *form* laporan penerimaan barang
  - Revisi: berisi nomor revisi
  - No. Kontrol: berisi nomor kontrol

- No. PO: berisi nomor *purchase order*
- Referensi: berisi data referensi yang dibutuhkan
- Tgl. Terima: berisi tanggal terima bahan baku
- No. Urut: berisi nomor urut dokumen
- Nama Barang: berisi nama bahan baku
- Part No: berisi nomor bahan baku
- Jumlah Terima: berisi jumlah bahan baku diterima
- Unit: satuan bahan baku yang diterima
- Keterangan: berisi keterangan bahan baku yang diterima

### 3. Bon Permintaan Dan Penyerahan Barang

Bon Permintaan Dan Penyerahan Barang merupakan dokumen yang terdapat di Bagian Gudang. Berperan sebagai pendataan bahan baku yang diminta dan dikeluarkan ke Bagian Produksi. Bon Permintaan Dan Penyerahan Barang dibuat berdasarkan kebutuhan bahan baku yang diperlukan Bagian Produksi (dapat dilihat pada gambar IV.10).

<b>MULTI PRIMA SEJAHTERA Tbk</b> Bagian Gudang		<i>(Signature)</i>	No. Form : 0109/GD/PMB Revisi : I			
<b>BON PERMINTAAN DAN PENYERAHAN BARANG</b>						
Diminta Oleh : UMA - 05 - 07. 2019		Diserahkan Oleh : <i>Huk</i>		Diterima Oleh : <i>Vatin</i>		
Dept / Seksi : <i>produksi</i>		Dept / Seksi : <i>lantai</i>		Dept / Seksi : <i>prod.</i>		
A.C.C. Kepala Seksi : <i>✓</i>		A.C.C. Kepala Seksi : <i>✓</i>		Tanggal : <i>25-9-19</i>		
NO.	NAMA BARANG <i>Arabik Komponen Biji</i>	PART NO.	JUMLAH	SATUAN	JOB NO.	KETERANGAN
✓	Steel	7800807	18.000	Fcs		
✓	Ore	8512480	18.000	"		
✓	Werk	6800018	18.000	"		
✓	Outide	6801029	18.000	"	✓ 84	
✓	Der 1		18.000	"		
✓	lo		1.800	"		
✓	200		90	"	✓	

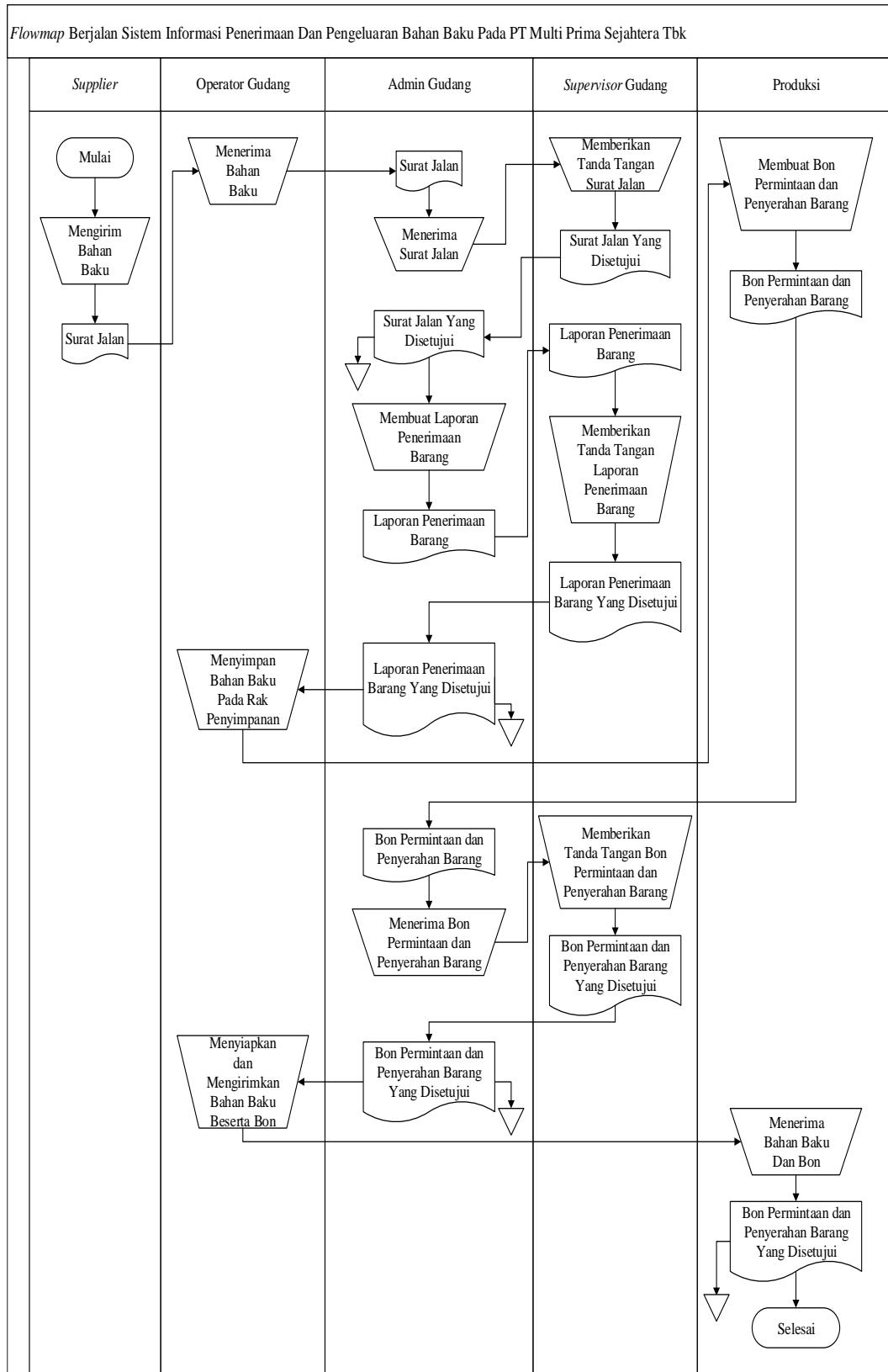
Gambar IV.10 Bon Permintaan Dan Penyerahan Barang  
(Sumber: PT Multi Prima Sejahtera Tbk, 2018)

Keterangan dari *field-field* pada gambar IV.10 adalah:

- No. Form: berisi nomor *form* bon permintaan dan penyerahan barang
- Revisi: berisi nomor revisi
- No.: berisi nomor kontrol
- Diminta Oleh: berisi data peminta
- Diserahkan Oleh: berisi data yg menyerahkan
- Diterima Oleh: berisi data yg menerima
- Dept/Seksi: berisi data divisi yang meminta
- Dept/Seksi: berisi data divisi yang menyerahkan
- Dept/Seksi: berisi data divisi yang menerima
- Acc Kepala Seksi: berisi tanda tangan divisi yang meminta
- Acc Kepala Seksi: berisi tanda tangan divisi yang menyerahkan
- Tanggal: berisi tanggal penyerahan bahan baku
- No.: berisi nomor urut data
- Nama Barang: berisi nama bahan baku
- Part No: berisi nomor bahan baku
- Jumlah: berisi jumlah bahan baku
- Satuan: berisi satuan bahan baku
- Job No: berisi nomor job
- Keterangan: berisi keterangan

#### 4.16.2 Analisis Proses Bisnis

Kegiatan manajemen gudang bahan baku pada PT Multi Prima Sejahtera Tbk terdiri dari penerimaan dan pengeluaran bahan baku. Penerimaan bahan baku yang dikirim oleh *supplier* kepada Bagian Gudang dan pengeluaran bahan baku kepada Bagian Produksi. Berikut adalah kegiatan penerimaan dan pengeluaran bahan baku pada PT Multi Prima Sejahtera Tbk diuraikan sebagai berikut (dapat dilihat pada gambar IV.11).



Gambar IV.11 *Flowmap Berjalan Sistem Informasi Manajemen Gudang Bahan Baku*  
(Sumber: Pengolahan Data, 2018)

Adapun proses manajemen gudang bahan baku pada Bagian Gudang adalah sebagai berikut:

1. *Supplier* mengirim bahan baku ke Bagian Gudang.
2. Operator Gudang menerima bahan baku.
3. Admin Gudang menerima surat jalan.
4. *Supervisor* Gudang memberikan tanda tangan surat jalan.
5. Admin Gudang membuat laporan penerimaan barang.
6. *Supervisor* Gudang memberikan tanda tangan laporan penerimaan barang.
7. Admin Gudang mengarsipkan laporan penerimaan barang yang disetujui.
8. Operator Gudang menyimpan bahan baku pada rak penyimpanan.
9. Produksi membuat bon permintaan dan penyerahan barang.
10. Admin Gudang menerima bon permintaan dan penyerahan barang.
11. *Supervisor* Gudang memberikan tanda tangan bon permintaan dan penyerahan barang.
12. Admin Gudang mengarsipkan bon permintaan dan penyerahan barang yang disetujui.
13. Operator Gudang menyiapkan dan mengirimkan bahan baku beserta bon.
14. Produksi menerima bahan baku dan bon.

## BAB V

### ANALISIS DAN PEMBAHASAN

#### 5.1 Analysis Kebutuhan Sistem

Berdasarkan hasil dari pengumpulan dan pengolahan data pada sistem informasi penerimaan dan pengeluaran bahan baku, maka dapat diidentifikasi bahwa dibutuhkannya analisis kebutuhan sistem untuk mendukung sistem baru dan batasan-batasan pada sistem baru. Berikut merupakan analisis kebutuhan sistem informasi penerimaan dan pengeluaran bahan baku yang terdapat pada tabel V.1 berikut ini:

Tabel V.1 Analisis Kebutuhan Sistem

No.	Permasalahan	Kebutuhan <i>User</i>	Solusi	<i>Functional Requirement</i>
1.	Data yang terkait dengan proses penerimaan dan pengeluaran bahan baku masih disimpan dan diolah secara manual menggunakan <i>form-form</i> yang kemudian ditumpuk di dalam sebuah arsip penyimpanan.	Sistem dengan media penyimpanan data dalam bentuk <i>database</i> agar dapat menyimpan data lebih aman dan terstruktur.	Menganalisis, merancang dan membangun sistem informasi penerimaan dan pengeluaran bahan baku dengan media penyimpanan data dalam bentuk <i>database</i> agar dapat menyimpan data lebih aman dan terstruktur.	Sistem dapat menyimpan, menambah, mengubah dan menghapus data bahan baku pada proses penerimaan dan pengeluaran bahan baku.

Tabel V.1 Analisis Kebutuhan Sistem (lanjutan)

No.	Permasalahan	Kebutuhan User	Solusi	Functional Requirement
2.	Belum adanya sistem informasi pada proses penerimaan dan pengeluaran bahan baku.	Sistem yang dapat memberikan kemudahan pengolahan data bahan baku yang diterima dan dikeluarkan sehingga proses penerimaan dan pengeluaran bahan baku menjadi lebih optimal.	Menganalisis, merancang dan membangun sistem informasi penerimaan dan pengeluaran bahan baku yang dapat memberikan kemudahan pengolahan data bahan baku yang diterima dan dikeluarkan sehingga proses penerimaan dan pengeluaran bahan baku menjadi lebih optimal.	Sistem dapat mengelola data bahan baku yang diterima dan dikeluarkan serta mampu memberikan kemudahan pengolahan data bahan baku yang diterima dan dikeluarkan.

(Sumber: Hasil Analisis, 2019)

### 5.1.1 Non Functional Requirement

*Non functional requirement* secara umum berisi batasan-batasan pada pelayanan atau fungsi yang disediakan oleh sistem. Berikut adalah *non functional requirement* dari sistem informasi penerimaan dan pengeluaran bahan baku yang diusulkan antara lain:

#### 1. Operasional

Sistem dapat diakses oleh pengguna melalui *browser* yang mendukung format PHP, membutuhkan *web server* dan *database*. Seperti Mozilla Firefox, Apache dan MySQL.

#### 2. Antarmuka (*Interface*)

Sistem yang sederhana, mudah dipahami (*user friendly*) sehingga pengguna dapat menggunakan aplikasi dengan mudah dan nyaman, sistem hanya menyediakan satu pilihan bahasa saja.

### 3. Keamanan

Sistem menggunakan *username* dan *password* untuk masuk ke dalam sistem dan memiliki hak akses.

### 4. *Supportability*

- Minimal RAM 1 GB.
- Minimal *harddisk* 64 GB.
- Menggunakan *mouse* dan *keyboard*.
- Menggunakan *printer* untuk mencetak laporan.

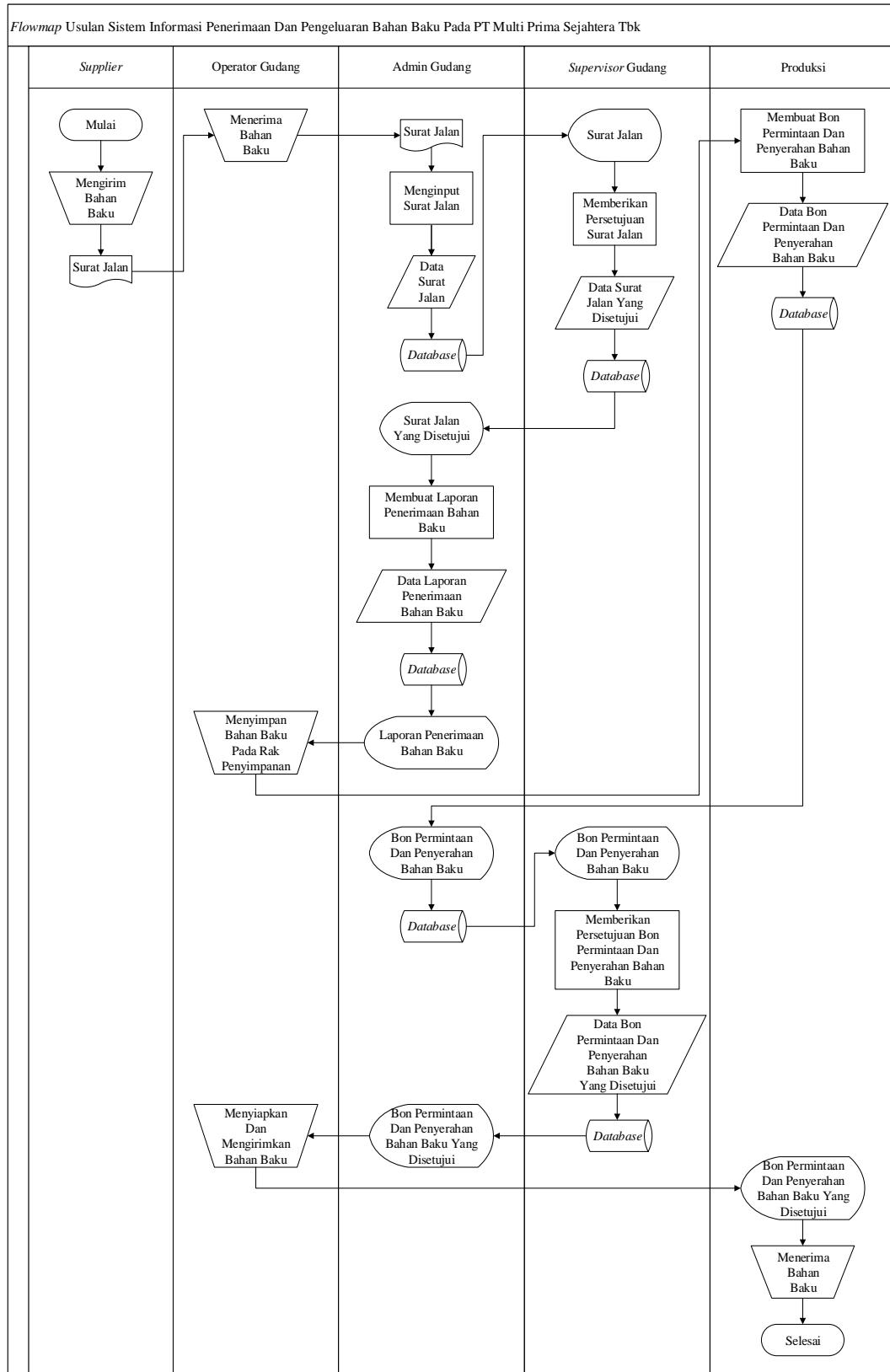
## 5.2 Urutan Prosedur dan *Flowmap* Usulan

Urutan prosedur dan *flowmap* pada sistem informasi penerimaan dan pengeluaran bahan baku yang diusulkan adalah sebagai berikut:

1. Dimulai dengan *supplier* mengirim bahan baku.
2. Operator Gudang menerima bahan baku.
3. Admin Gudang menginput surat jalan.
4. Admin Gudang menyimpan data surat jalan ke dalam *database*.
5. *Supervisor* Gudang memberikan persetujuan surat jalan.
6. *Supervisor* Gudang menyimpan data surat jalan yang sudah disetujui ke dalam *database*.
7. Admin Gudang membuat laporan penerimaan bahan baku.
8. Admin Gudang menyimpan data laporan penerimaan bahan baku ke dalam *database*.
9. Operator Gudang menyimpan bahan baku pada rak penyimpanan.
10. Produksi membuat bon permintaan dan penyerahan bahan baku.
11. Produksi menyimpan data bon permintaan dan penyerahan bahan baku ke dalam *database*.
12. *Supervisor* Gudang memberikan persetujuan bon permintaan dan penyerahan bahan baku.
13. *Supervisor* Gudang menyimpan data bon permintaan dan penyerahan bahan baku yang disetujui.
14. Operator Gudang menyiapkan dan mengirimkan bahan baku.

15. Produksi menerima bahan baku.

Adapun aliran *flowmap* pada usulan sistem informasi penerimaan dan pengeluaran bahan baku di Bagian Gudang pada PT Multi Prima Sejahtera dapat dilihat pada gambar V.1 berikut ini:



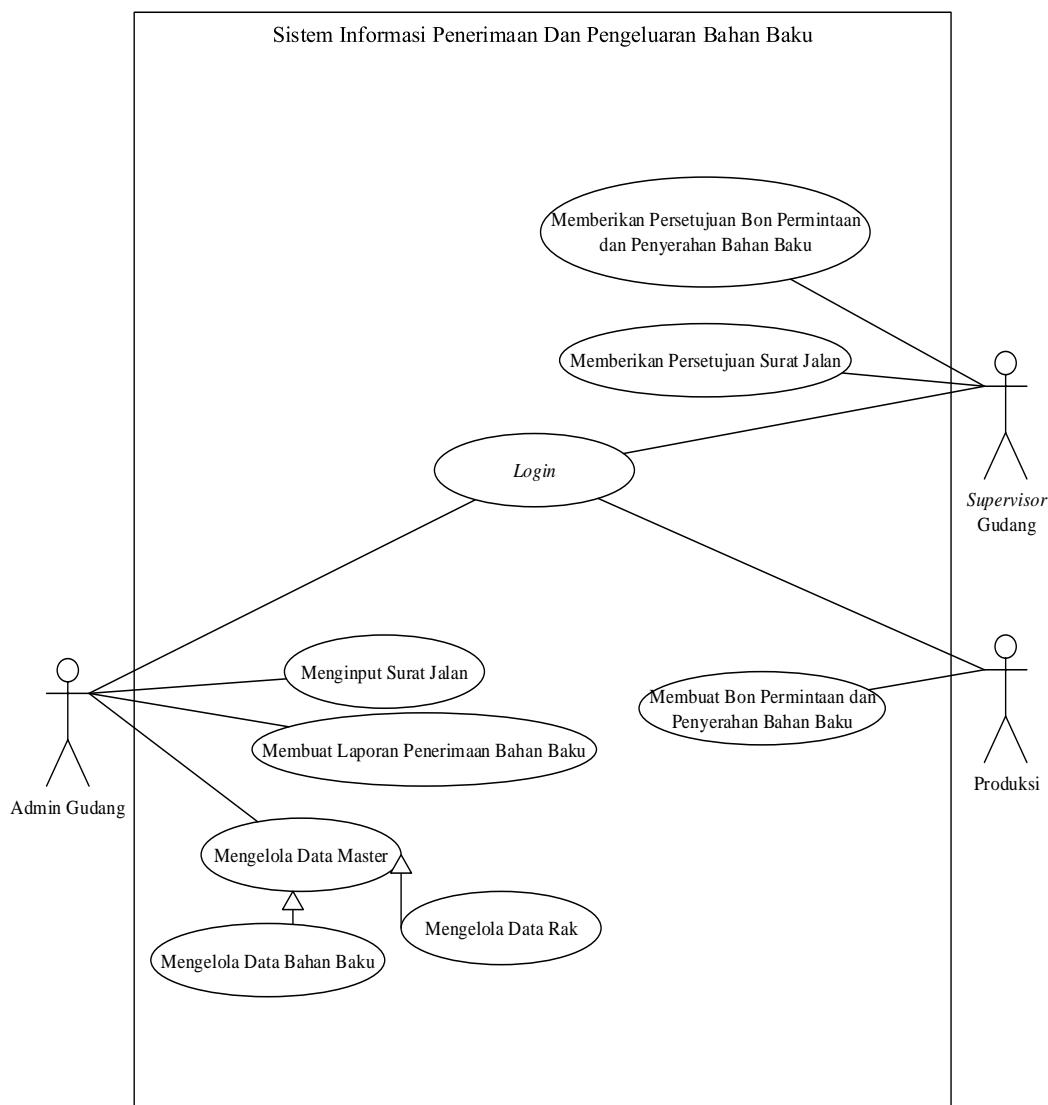
Gambar V.1 *Flowmap Usulan Sistem Informasi Manajemen Gudang Bahan Baku*  
(Sumber: Hasil Analisis, 2019)

### 5.3 Analisis Sistem Informasi Usulan

Analisis sistem informasi penerimaan dan pengeluaran bahan baku pada Bagian Gudang menggunakan *tools* pemodelan sistem UML (*Unified Modeling Language*) yang meliputi pembuatan *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram* dan *Deployment Diagram*.

#### 5.3.1 Use Case Diagram

Berdasarkan analisis yang telah dilakukan, maka *use case diagram* sistem informasi penerimaan dan pengeluaran bahan baku dapat dilihat pada gambar V.2 berikut ini:



Gambar V.2 *Use Case Diagram* Sistem Informasi Manajemen Gudang Bahan Baku  
 (Sumber: Hasil Analisis, 2019)

### 5.3.1.1 Definisi Aktor

Pendefinisian aktor pada *use case diagram* perancangan sistem informasi penerimaan dan pengeluaran bahan baku usulan dapat dilihat pada tabel V.2 berikut ini:

#### 1. Definisi Aktor

Aktor merupakan penggambaran tokoh atau sistem yang memperoleh keuntungan dan berada di luar dari sistem. Pendefinisian aktor pada *use case diagram* sistem informasi penerimaan dan pengeluaran bahan baku pada PT Multi Prima Sejahtera Tbk dapat dilihat pada tabel V.2 berikut ini:

Tabel V.2 Definisi Aktor

No.	Aktor	Definisi
1.	Supervisor Gudang	<i>Supervisor</i> Gudang adalah orang yang bertugas sebagai pengawas, penanggung jawab dan pengarah <i>staff</i> bawahan. Tugas <i>Supervisor</i> Gudang PT Multi Prima Sejahtera dalam sistem ini adalah memberikan persetujuan surat jalan dan memberikan persetujuan bon permintaan dan penyerahan bahan baku. Untuk dapat mengakses kegiatan tersebut <i>Supervisor</i> Gudang diharuskan melakukan <i>login</i> terlebih dahulu.
2.	Admin Gudang	Admin Gudang adalah sutradara dibalik segala operasional gudang. Tugas Admin Gudang PT Multi Prima Sejahtera dalam sistem ini adalah meng <input type="text"/> surat jalan, membuat laporan penerimaan bahan baku dan mengelola data bahan baku dan data rak. Untuk dapat mengakses kegiatan tersebut Admin Gudang diharuskan melakukan <i>login</i> terlebih dahulu.
3.	Produksi	Produksi adalah organisasi yang bertanggung jawab penuh dalam proses produksi di suatu perusahaan. Tugas Produksi PT Multi Prima Sejahtera dalam sistem ini adalah membuat bon permintaan dan penyerahan bahan baku. Untuk dapat mengakses kegiatan tersebut Produksi diharuskan melakukan <i>login</i> terlebih dahulu.

(Sumber: Hasil Analisis, 2019)

### 5.3.1.2 Definisi Use Case

Berikut adalah skenario *use case diagram* sistem informasi penerimaan dan pengeluaran bahan baku yang diusulkan:

#### 1. Skenario *Use Case Diagram Login*

Tabel V.3 *Use Case Login*

Nama Use Case	<i>Login</i>
Aktor	<i>Supervisor Gudang, Admin Gudang dan Produksi</i>
Deskripsi	<i>Use Case</i> ini menggambarkan proses pengguna masuk ke sistem informasi penerimaan bahan baku.
<i>Normal Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor membuka aplikasi.</li> <li>2. Sistem akan menampilkan halaman <i>login</i>.</li> <li>3. Aktor mengisi <i>Username</i> dan <i>Password</i>.</li> <li>4. Aktor mengklik tombol <i>Login</i>.</li> <li>5. Sistem akan memvalidasi <i>Username</i> dan <i>Password</i>.</li> <li>6. Jika <i>Username</i> dan <i>Password</i> valid, sistem akan menampilkan halaman utama.</li> <li>7. Jika <i>Username</i> dan <i>Password</i> tidak valid, sistem akan meminta aktor memasukkan <i>Username</i> dan <i>Password</i> sekali lagi.</li> </ol>

(Sumber: Hasil Analisis, 2019)

#### 2. Skenario *Use Case Diagram Mengelola Data Master*

Tabel V.4 *Use Case Mengelola Data Master*

Nama Use Case	Mengelola Data Master
Aktor	Admin Gudang
Deskripsi	<i>Use Case</i> ini menggambarkan proses mengelola data master seperti data master bahan baku dan data master rak.
<i>Normal Flow</i>	<ol style="list-style-type: none"> <li>1. Admin Gudang memilih menu data master.</li> <li>2. Admin Gudang memilih sub menu data master.</li> <li>3. Jika Admin Gudang ingin mengelola data master bahan baku maka Admin Gudang mengklik bahan baku, jika Admin Gudang ingin mengelola data master rak maka Admin Gudang mengklik rak.</li> <li>4. Sistem menampilkan data sub menu data master yang dipilih.</li> </ol>

Tabel V.4 Use Case Mengelola Data Master (lanjutan)

	<p>5. Admin Gudang dapat melakukan aksi tambah dengan mengklik tambah.</p> <p>6. Sistem menampilkan halaman tambah pada data master yang dipilih.</p> <p>7. Admin Gudang mengisi data lalu mengklik simpan.</p> <p>8. Sistem mengupdate <i>database</i> lalu menampilkan data master yang dipilih.</p> <p>9. Admin Gudang dapat melakukan aksi ubah dengan mengklik ubah.</p> <p>10. Sistem menampilkan halaman ubah pada data master yang dipilih.</p> <p>11. Admin Gudang mengubah data lalu mengklik simpan.</p> <p>12. Sistem mengupdate <i>database</i> lalu menampilkan data master yang dipilih.</p> <p>13. Admin Gudang dapat melakukan aksi hapus dengan mengklik hapus.</p> <p>14. Sistem menampilkan <i>pop up</i> hapus.</p> <p>15. Admin Gudang mengklik oke lalu sistem mengupdate <i>database</i> lalu menampilkan data master yang dipilih.</p> <p>16. Admin Gudang dapat mengklik batal maka sistem akan menampilkan data master yang dipilih.</p>
--	---

(Sumber: Hasil Analisis, 2019)

### 3. Skenario Use Case Diagram Menginput Surat Jalan

Tabel V.5 Use Case Menginput Surat Jalan

Nama Use Case	Menginput Surat Jalan
Aktor	Admin Gudang
Deskripsi	Use Case ini menggambarkan proses menginput surat jalan.
Normal Flow	<p>1. Admin Gudang memilih data transaksi dan memilih sub menu data transaksi.</p> <p>2. Admin Gudang mengklik surat jalan.</p> <p>3. Sistem menampilkan halaman surat jalan.</p>

Tabel V.5 Use Case Menginput Surat Jalan (lanjutan)

	<p>4. Admin Gudang dapat melakukan aksi tambah dengan mengklik tambah.</p> <p>5. Sistem menampilkan halaman tambah surat jalan.</p> <p>6. Admin Gudang mengisi data lalu mengklik simpan.</p> <p>7. Sistem menampilkan halaman tambah surat jalan detail.</p> <p>8. Admin Gudang mengisi data lalu mengklik simpan.</p> <p>9. Sistem mengupdate <i>database</i> lalu menampilkan data surat jalan.</p> <p>10. Admin Gudang dapat melakukan aksi hapus dengan mengklik hapus.</p> <p>11. Sistem menampilkan <i>pop up</i> hapus.</p> <p>12. Admin Gudang mengklik oke lalu sistem mengupdate <i>database</i> lalu menampilkan data surat jalan.</p> <p>13. Admin Gudang dapat mengklik batal maka sistem akan menampilkan data surat jalan.</p>
--	--

(Sumber: Hasil Analisis, 2019)

#### 4. Skenario Use Case Diagram Membuat Laporan Penerimaan Bahan Baku

Tabel V.6 Use Case Membuat Laporan Penerimaan

Nama Use Case	Membuat Laporan Penerimaan Bahan Baku
Aktor	Admin Gudang
Deskripsi	<i>Use Case</i> ini menggambarkan proses membuat laporan penerimaan bahan baku.
Normal Flow	<p>1. Admin Gudang memilih data laporan dan memilih sub menu laporan.</p> <p>2. Admin Gudang mengklik data penerimaan.</p> <p>3. Sistem menampilkan halaman rekap data surat jalan.</p> <p>4. Admin Gudang memilih tanggal awal dan tanggal akhir yang ditentukan, lalu mengklik rekap data.</p> <p>5. Sistem menampilkan data surat jalan yang disetujui.</p> <p>6. Admin Gudang mengklik aksi laporan pada surat jalan yang dipilih berdasarkan tanggal yang ditentukan.</p> <p>7. Sistem menampilkan laporan penerimaan bahan baku.</p>

(Sumber: Hasil Analisis, 2019)

## 5. Skenario *Use Case Diagram* Memberikan Persetujuan Surat Jalan

Tabel V.7 *Use Case* Memberikan Persetujuan Surat Jalan

Nama <i>Use Case</i>	Memberikan Persetujuan Surat Jalan
Aktor	<i>Supervisor</i> Gudang
Deskripsi	<i>Use Case</i> ini menggambarkan proses memberikan persetujuan surat jalan.
<i>Normal Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Supervisor</i> Gudang memilih data transaksi dan memilih sub menu data transaksi.</li> <li>2. <i>Supervisor</i> Gudang mengklik surat jalan.</li> <li>3. Sistem menampilkan halaman surat jalan.</li> <li>4. <i>Supervisor</i> Gudang memilih surat jalan.</li> <li>5. <i>Supervisor</i> Gudang memberikan persetujuan dengan mengklik setuju.</li> <li>6. Sistem mengubah status dari belum disetujui menjadi disetujui.</li> <li>7. Sistem mengupdate <i>database</i>.</li> </ol>

(Sumber: Hasil Analisis, 2019)

## 6. Skenario *Use Case Diagram* Membuat Bon Permintaan dan Penyerahan

Tabel V.8 *Use Case* Membuat Bon Permintaan dan Penyerahan

Nama <i>Use Case</i>	Menginput Bon Permintaan Dan Penyerahan
Aktor	Produksi
Deskripsi	<i>Use Case</i> ini menggambarkan proses membuat bon permintaan dan penyerahan.
<i>Normal Flow</i>	<ol style="list-style-type: none"> <li>1. Produksi memilih data transaksi dan memilih sub menu data transaksi.</li> <li>2. Produksi mengklik bon permintaan dan penyerahan.</li> <li>3. Sistem menampilkan halaman bon permintaan dan penyerahan.</li> <li>4. Produksi dapat melakukan aksi tambah dengan mengklik tambah.</li> <li>5. Sistem menampilkan halaman tambah bon permintaan dan penyerahan.</li> <li>6. Produksi mengisi data lalu mengklik simpan.</li> <li>7. Sistem menampilkan halaman tambah bon permintaan dan penyerahan detail.</li> </ol>

Tabel V.8 *Use Case* Membuat Bon Permintaan (lanjutan)

	<p>8. Produksi mengisi data lalu mengklik simpan.</p> <p>9. Sistem mengupdate <i>database</i> lalu menampilkan data bon permintaan dan penyerahan.</p> <p>10. Produksi dapat melakukan aksi hapus dengan mengklik hapus.</p> <p>11. Sistem menampilkan <i>pop up</i> hapus.</p> <p>12. Produksi mengklik oke lalu sistem mengupdate <i>database</i> lalu menampilkan data bon permintaan dan penyerahan.</p> <p>13. Produksi dapat mengklik batal maka sistem akan menampilkan data bon permintaan dan penyerahan.</p>
--	--

(Sumber: Hasil Analisis, 2019)

#### 7. Skenario *Use Case Diagram* Memberikan Persetujuan Bon Permintaan dan Penyerahan

Tabel V.9 *Use Case* Memberikan Persetujuan Bon Permintaan

Nama <i>Use Case</i>	Memberikan Persetujuan Bon Permintaan dan Penyerahan
Aktor	<i>Supervisor Gudang</i>
Deskripsi	<i>Use Case</i> ini menggambarkan proses memberikan persetujuan bon permintaan dan penyerahan.
Normal Flow	<p>1. <i>Supervisor Gudang</i> memilih data transaksi dan memilih sub menu data transaksi.</p> <p>2. <i>Supervisor Gudang</i> mengklik bon permintaan dan penyerahan.</p> <p>3. Sistem menampilkan halaman bon permintaan dan penyerahan.</p> <p>4. <i>Supervisor Gudang</i> memilih bon permintaan dan penyerahan.</p> <p>5. <i>Supervisor Gudang</i> memberikan persetujuan dengan mengklik setuju.</p> <p>6. Sistem mengubah status dari belum disetujui menjadi disetujui.</p> <p>7. Sistem mengupdate <i>database</i>.</p>

(Sumber: Hasil Analisis, 2019)

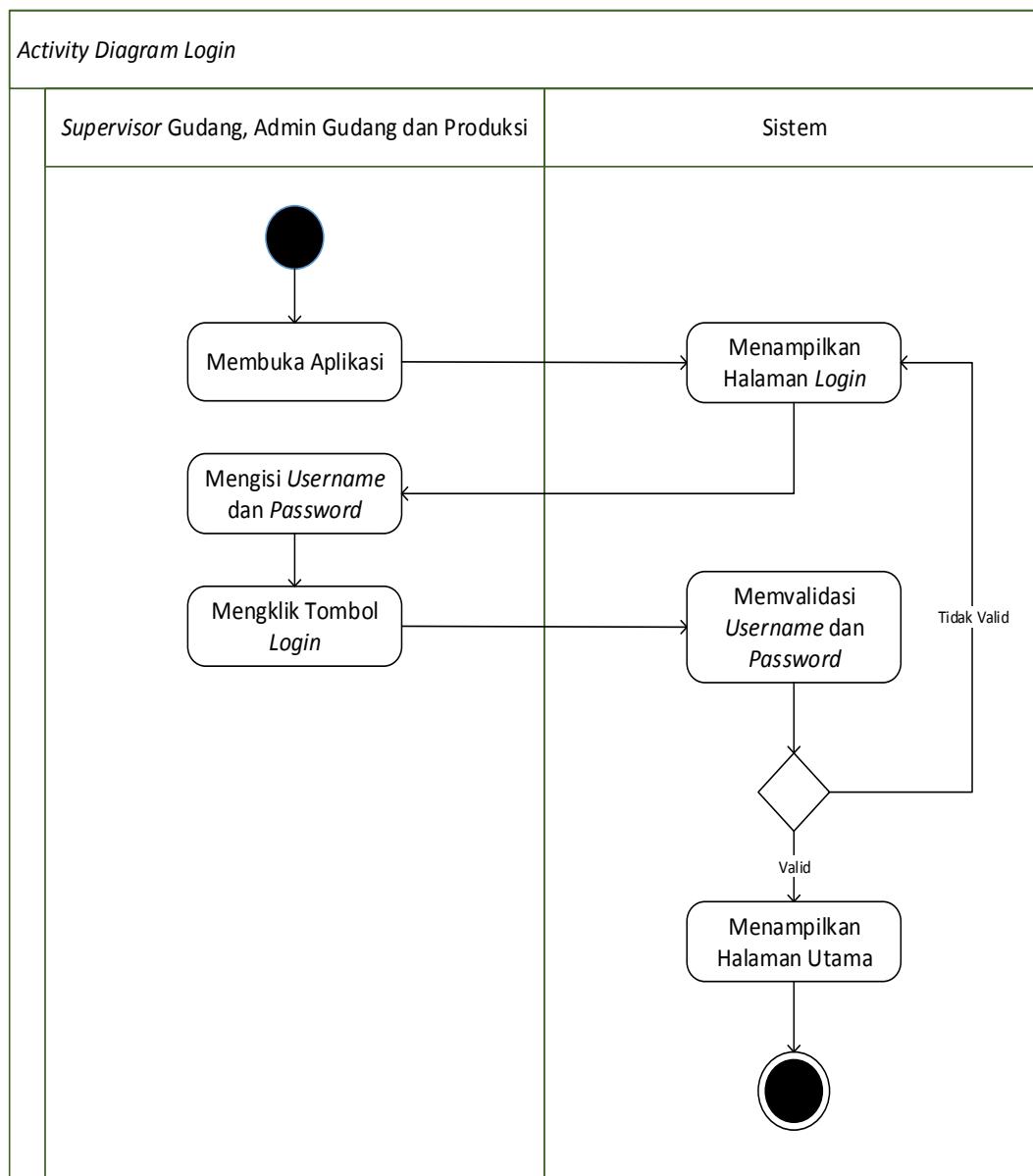
#### 5.3.2 *Activity Diagram*

*Activity diagram* digunakan untuk menggambarkan aliran kerja tiap *use case* pada sistem informasi penerimaan dan pengeluaran bahan baku yang ada

pada Bagian Gudang. *Activity diagram* masing-masing *use case* digambarkan pada gambar-gambar berikut ini:

1. *Activity Diagram Proses Login*

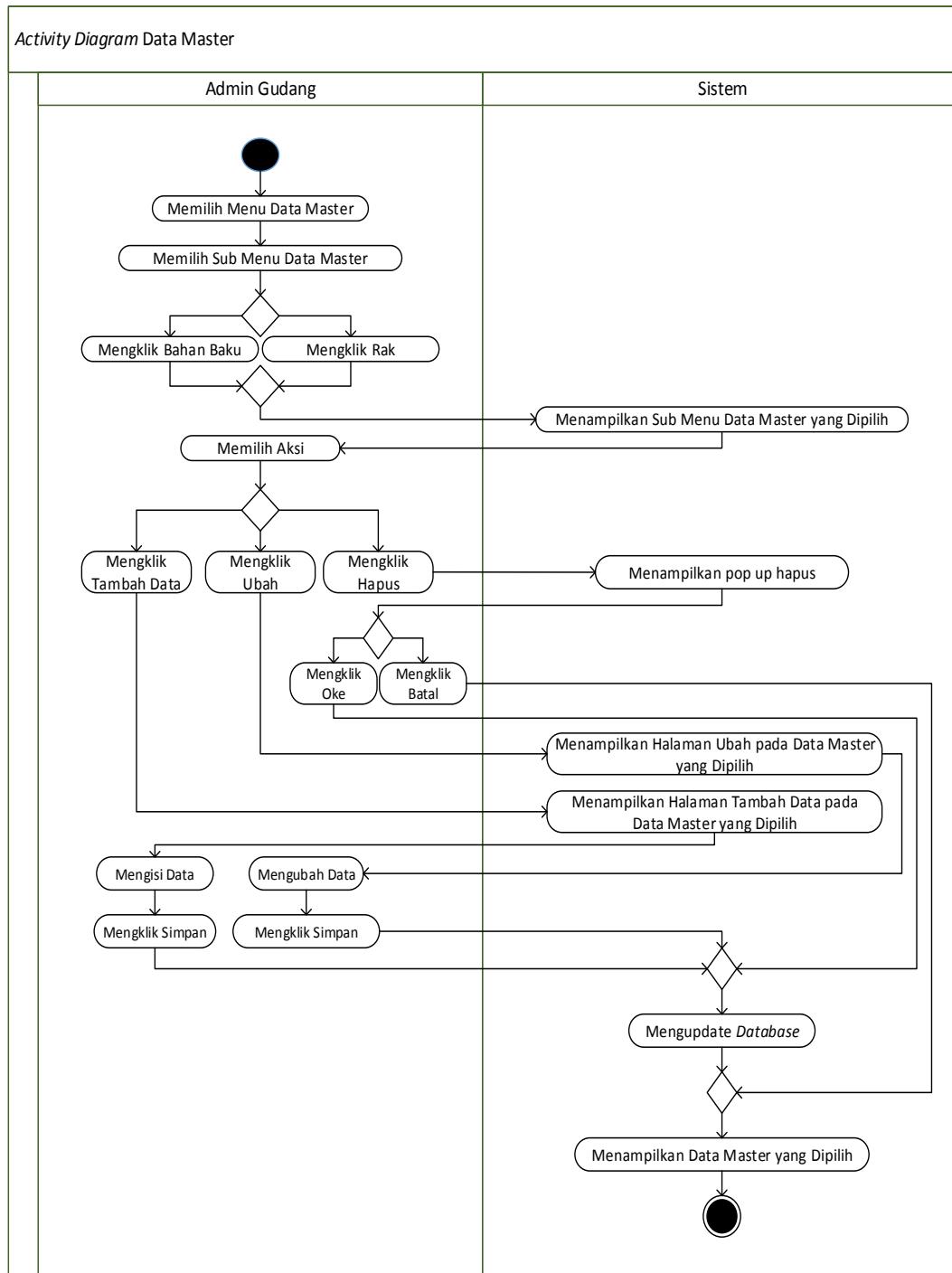
*Activity diagram* berikut ini menggambarkan aktivitas saat melakukan proses *login*. Pengguna mengisi *username* dan *password* untuk dapat masuk ke sistem. *Activity diagram* yang diusulkan dapat dilihat pada gambar V.3 sebagai berikut:



Gambar V.3 *Activity Diagram* Proses *Login*  
(Sumber: Hasil Analisis, 2019)

## 2. Activity Diagram Mengelola Data Master

*Activity diagram* mengelola data master berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada saat mengelola data master. *Activity diagram* yang diusulkan dapat dilihat pada gambar V.4 sebagai berikut:

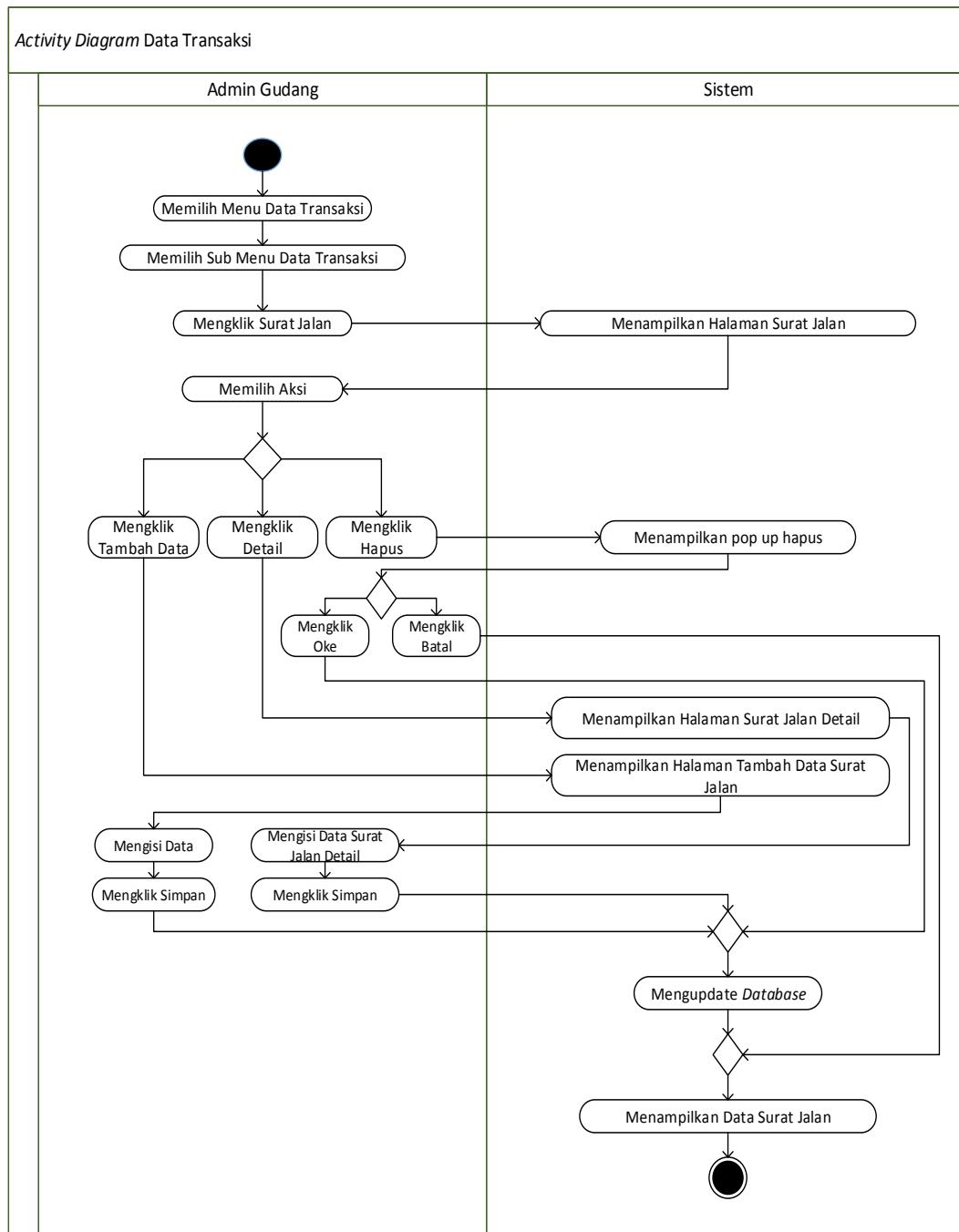


Gambar V.4 *Activity Diagram* Mengelola Data Master

(Sumber: Hasil Analisis, 2019)

### 3. Activity Diagram Menginput Surat Jalan

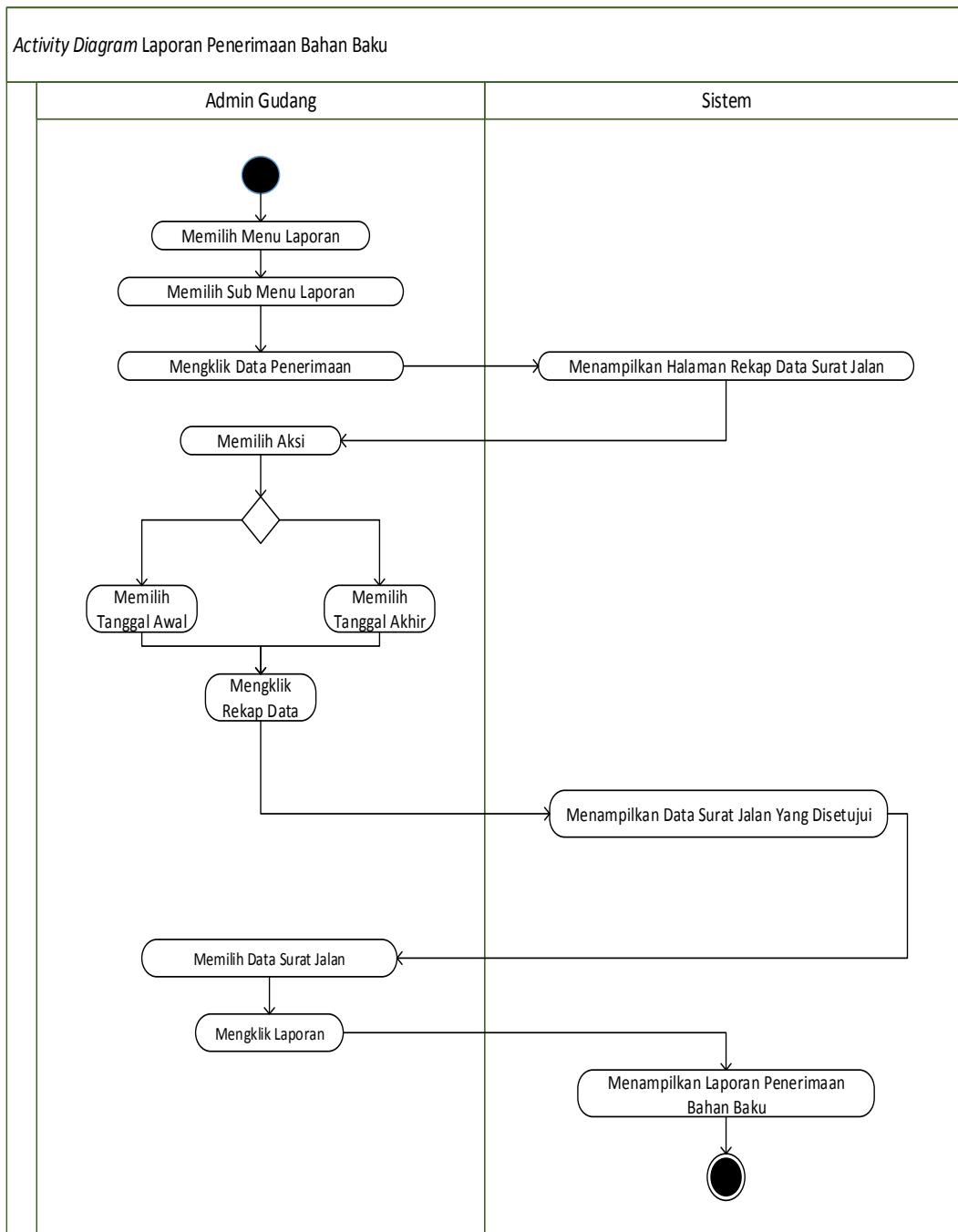
*Activity diagram* menginput data surat jalan berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada saat menginput data surat jalan. *Activity diagram* yang diusulkan dapat dilihat pada gambar V.5 sebagai berikut:



Gambar V.5 *Activity Diagram* Menginput Data Surat Jalan  
(Sumber: Hasil Analisis, 2019)

#### 4. *Activity Diagram* Membuat Laporan Penerimaan Bahan Baku

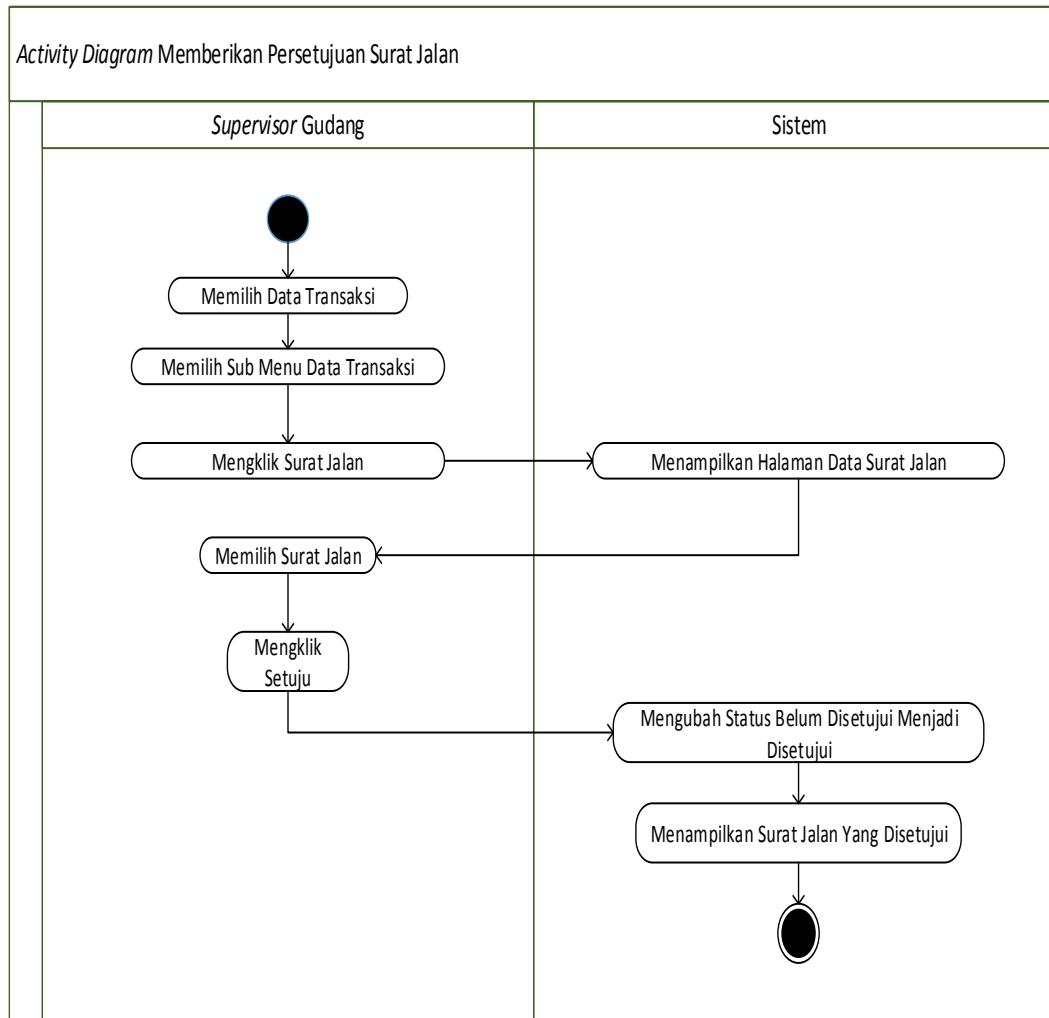
*Activity diagram* membuat laporan penerimaan bahan baku berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada proses membuat laporan penerimaan bahan baku. *Activity diagram* yang diusulkan dapat dilihat pada gambar V.6 sebagai berikut:



Gambar V.6 *Activity Diagram* Membuat Laporan Penerimaan  
(Sumber: Hasil Analisis, 2019)

5. *Activity Diagram* Memberikan Persetujuan Surat Jalan

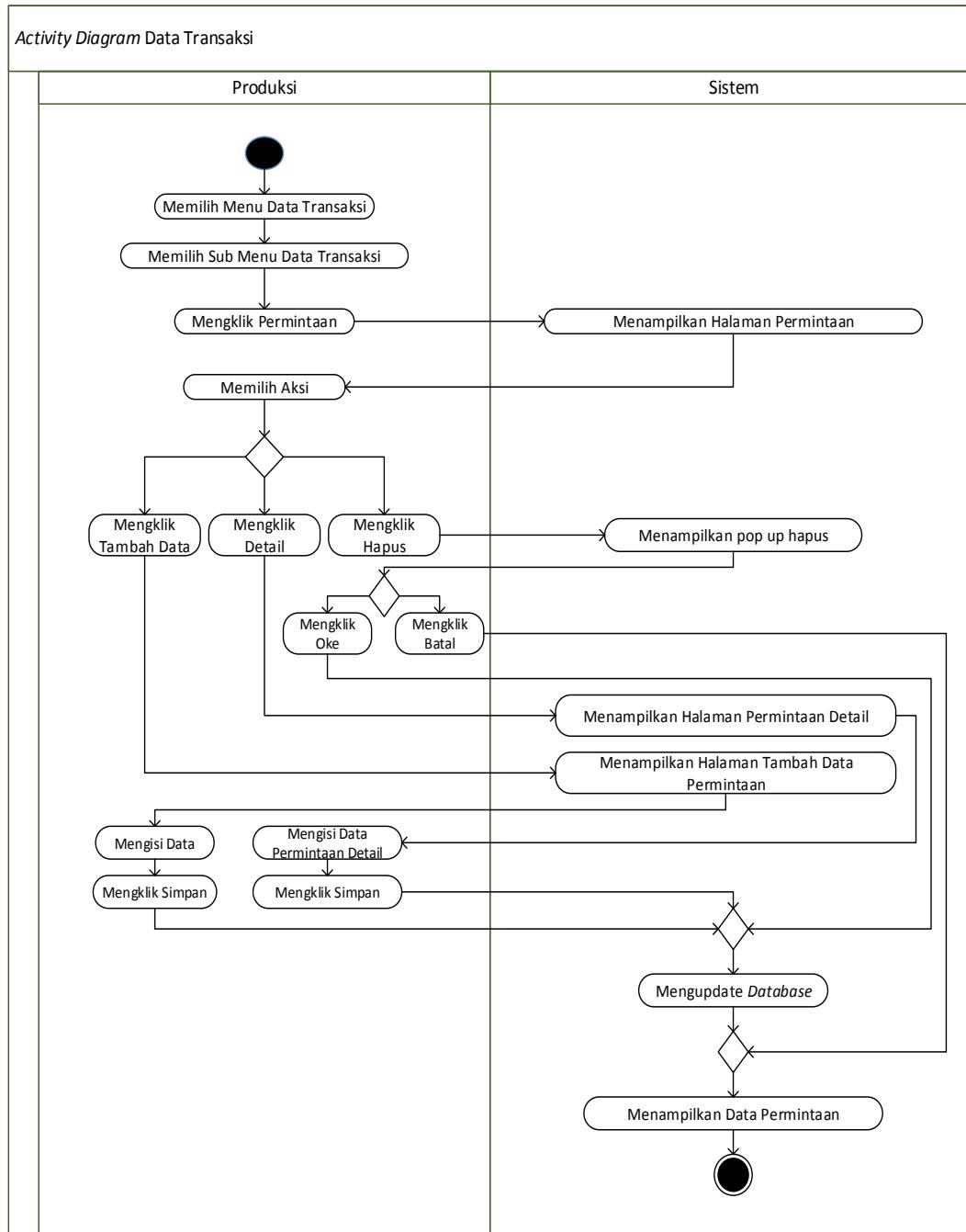
*Activity diagram* memberikan persetujuan surat jalan berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada saat proses memberikan persetujuan surat jalan. *Activity diagram* yang diusulkan dapat dilihat pada gambar V.7 sebagai berikut:



Gambar V.7 *Activity Diagram* Memberikan Persetujuan Surat Jalan  
(Sumber: Hasil Analisis, 2019)

6. *Activity Diagram* Membuat Bon Permintaan dan Penyerahan

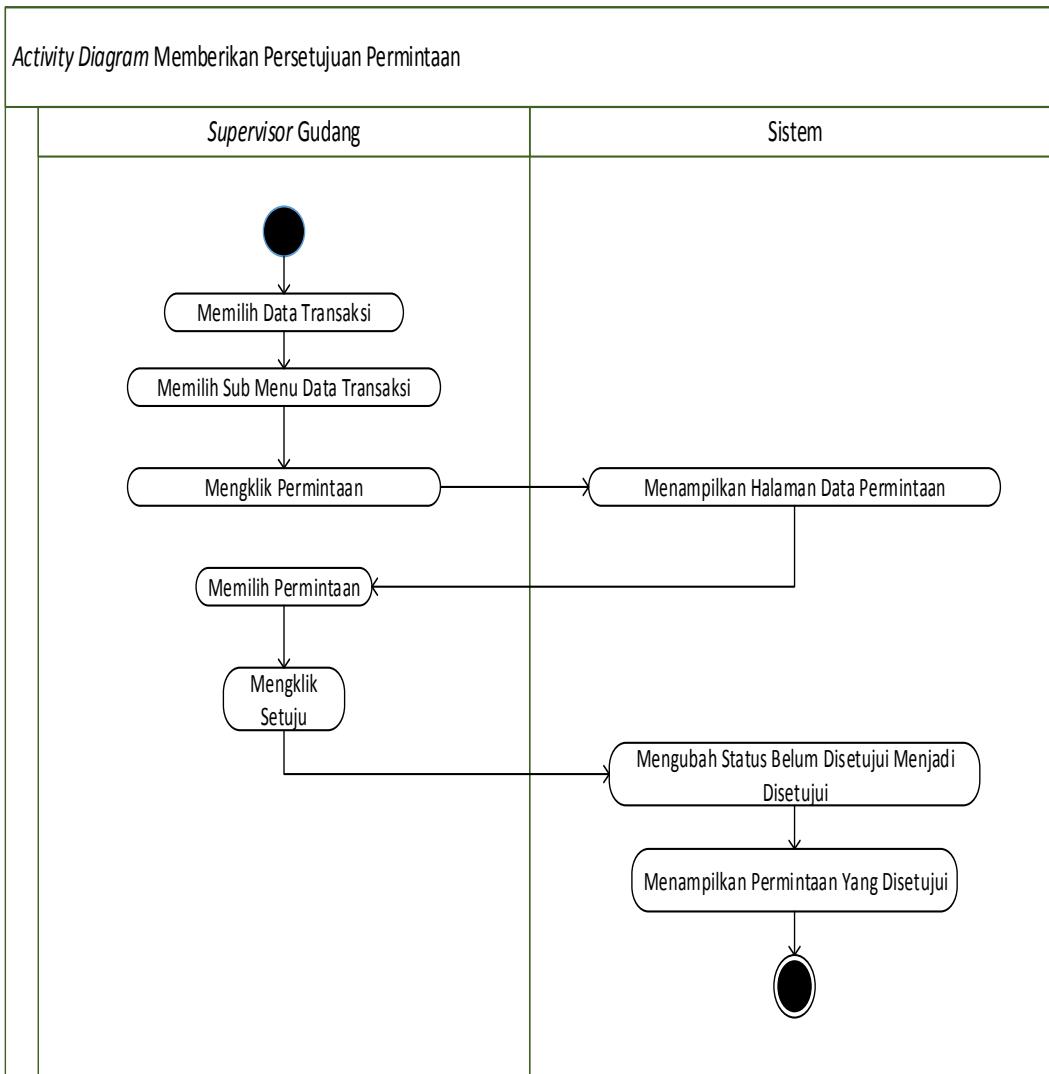
*Activity diagram* membuat bon permintaan dan penyerahan berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada saat proses membuat bon permintaan dan penyerahan. *Activity diagram* yang diusulkan dapat dilihat pada gambar V.8 sebagai berikut:



Gambar V.8 *Activity Diagram* Membuat Bon Permintaan dan Penyerahan  
(Sumber: Hasil Analisis, 2019)

#### 7. *Activity Diagram* Memberikan Persetujuan Bon Permintaan

*Activity diagram* memberikan persetujuan permintaan dan penyerahan berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada saat proses memberikan persetujuan permintaan dan penyerahan. *Activity diagram* yang diusulkan dapat dilihat pada gambar V.9 sebagai berikut:



Gambar V.9 *Activity Diagram* Memberikan Persetujuan Bon Permintaan  
 (Sumber: Hasil Analisis, 2019)

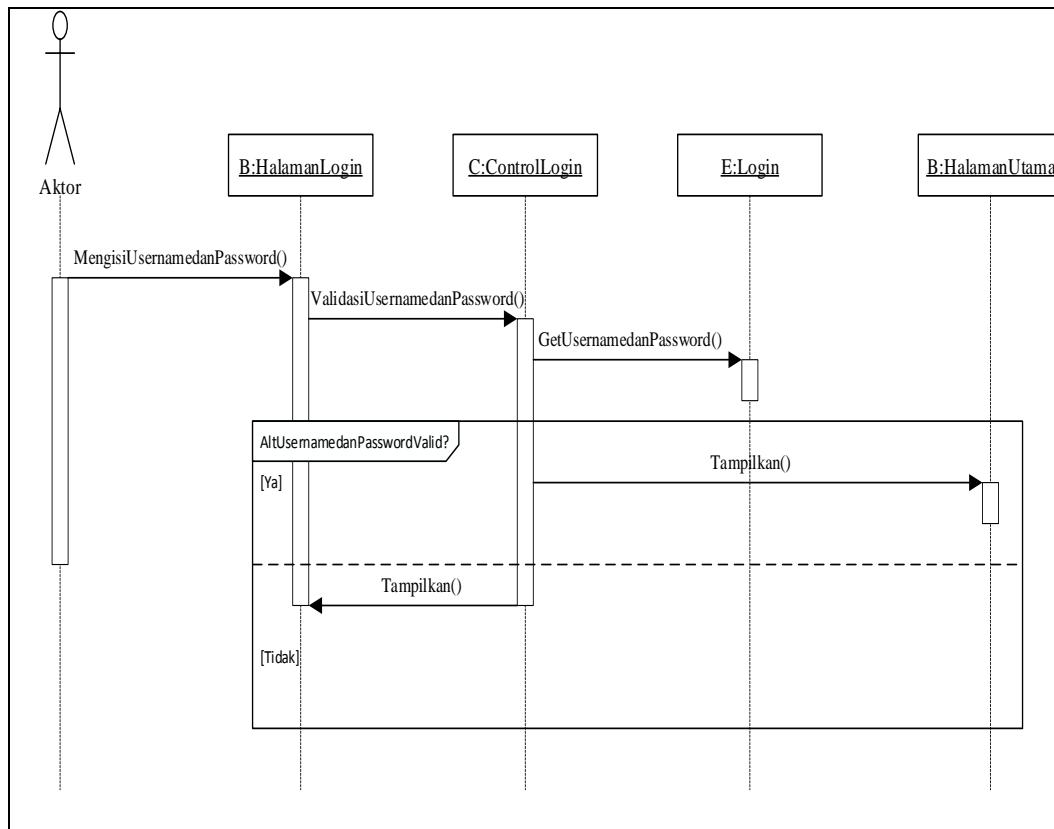
### 5.3.3 Sequence Diagram

*Sequence diagram* menggambarkan interaksi yang terjadi pada suatu objek *use case diagram* ketika melakukan suatu proses tertentu, dimana urutan proses ketika melakukan suatu proses tertentu dapat diketahui dengan melihat gambaran pada diagram.

Hubungan yang ada pada gambar-gambar di bawah ini adalah proses yang dilakukan oleh sistem sesuai dengan objek pada *use case diagram*. Berikut adalah *sequence diagram* pada sistem informasi penerimaan dan pengeluaran bahan baku:

### 1. Sequence Diagram pada Proses Login

*Sequence diagram login* menggambarkan interaksi yang terjadi pada proses *login*. Proses ini dilakukan oleh seluruh aktor dalam *use case diagram*. *Sequence diagram* proses login digambarkan pada gambar V.10.

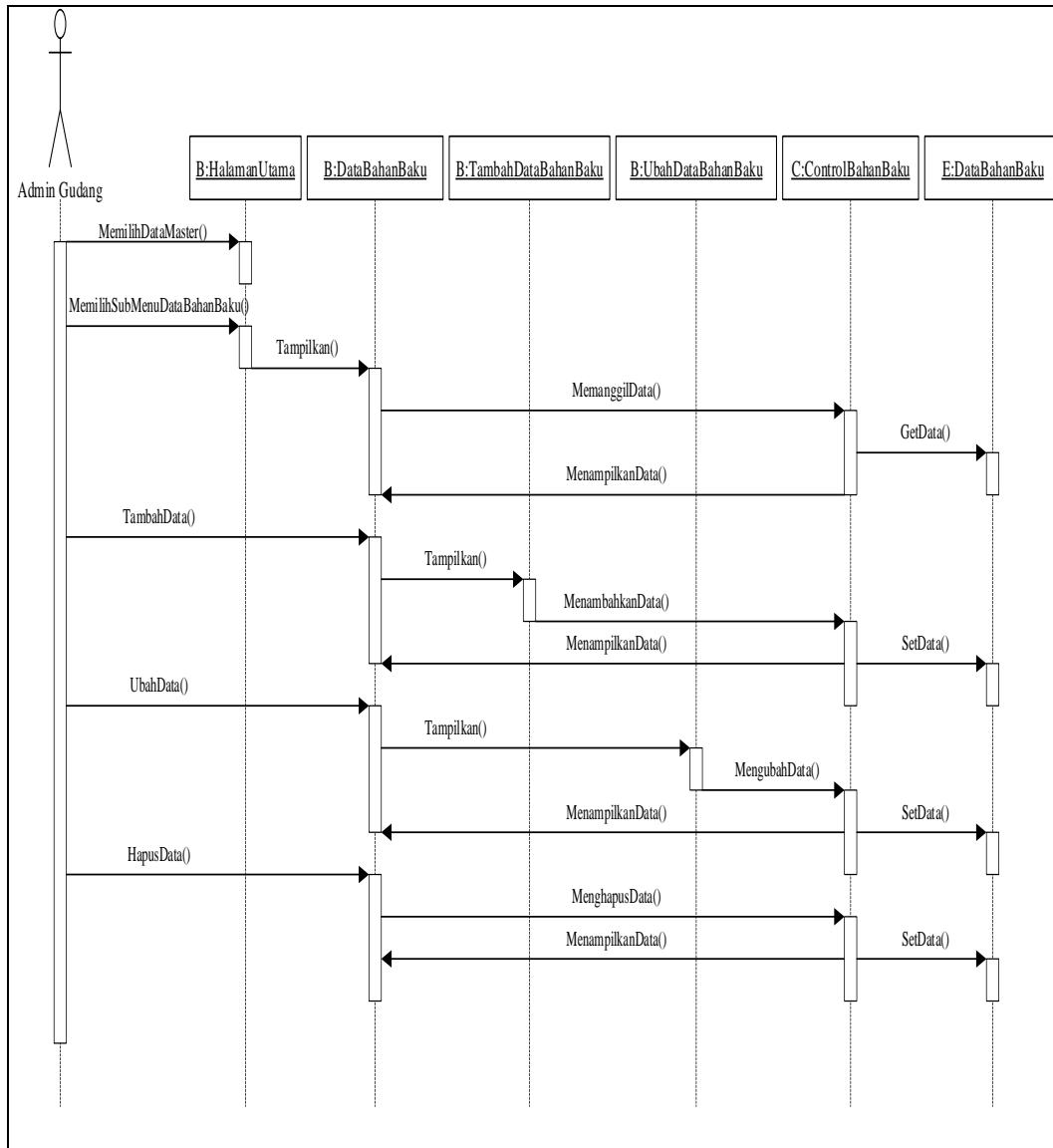


Gambar V.10 *Sequence Diagram* Proses Login

(Sumber: Hasil Analisis, 2019)

### 2. Sequence Diagram Mengelola Data Bahan Baku

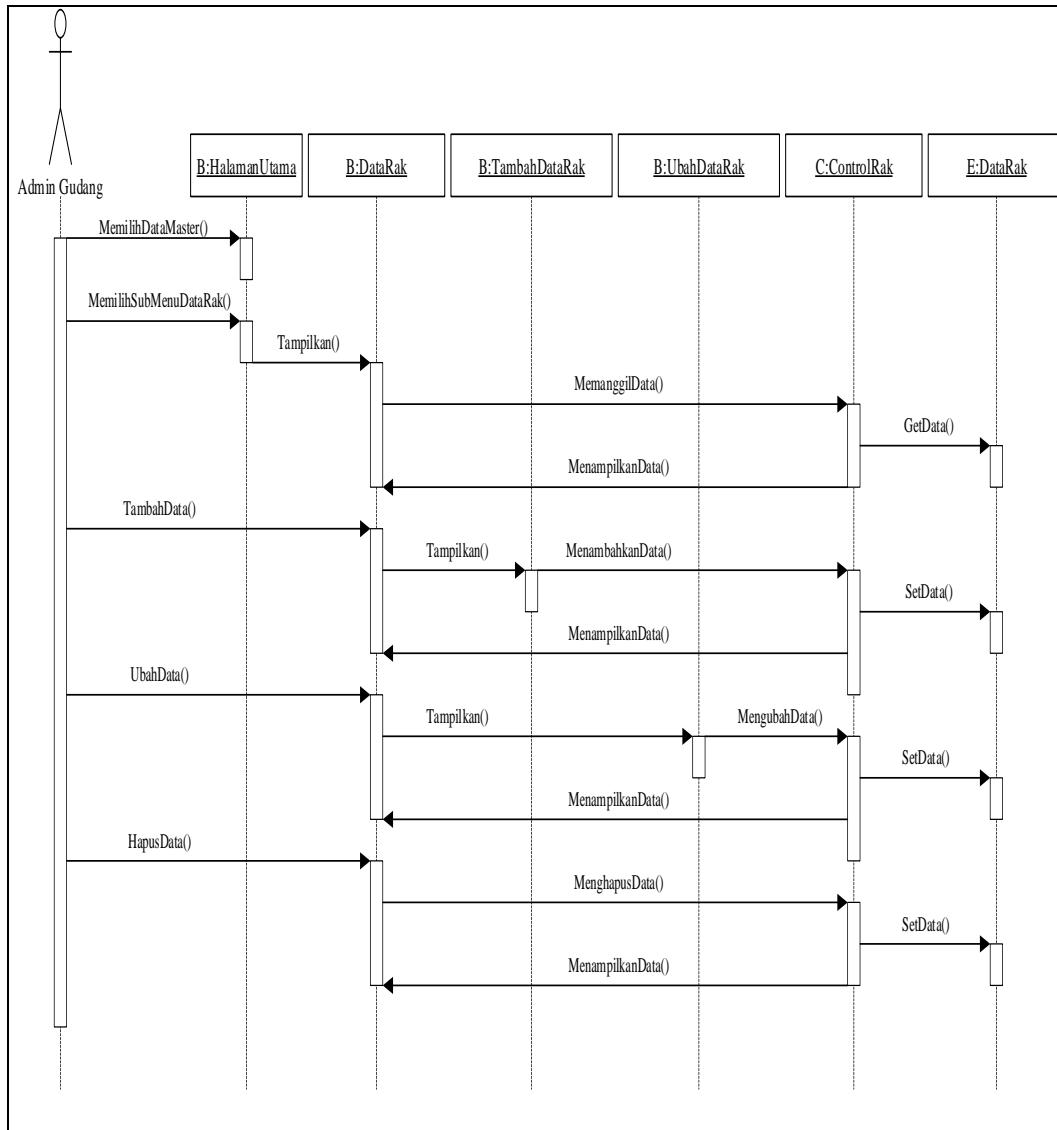
*Sequence diagram* mengelola data bahan baku menggambarkan interaksi yang terjadi pada saat dilakukan proses mengelola data bahan baku. *Sequence diagram* mengelola data bahan baku memiliki beberapa kegiatan yang dapat dilakukan seperti menambah data bahan baku, mengubah data bahan baku dan menghapus data bahan baku. *Sequence diagram* mengelola data bahan baku digambarkan pada gambar V.11.



Gambar V.11 *Sequence Diagram* Mengelola Data Bahan Baku  
 (Sumber: Hasil Analisis, 2019)

### 3. *Sequence Diagram* Mengelola Data Rak

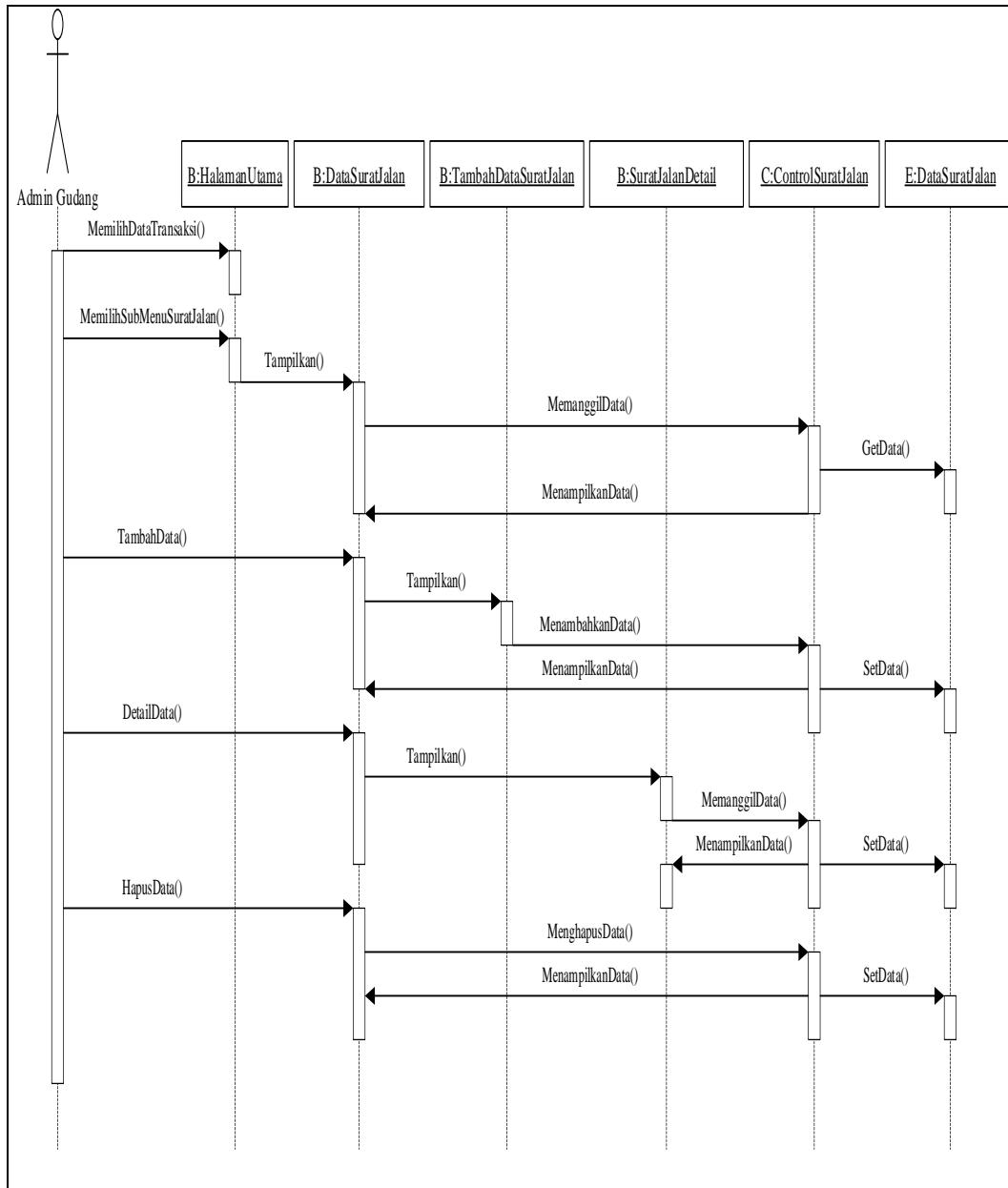
*Sequence diagram* mengelola data rak menggambarkan interaksi yang terjadi pada saat dilakukan proses mengelola data rak. *Sequence diagram* mengelola data rak memiliki beberapa kegiatan yang dapat dilakukan seperti menambah data rak, mengubah data rak dan menghapus data rak. *Sequence diagram* mengelola data rak digambarkan pada gambar V.12.



Gambar V.12 *Sequence Diagram* Mengelola Data Rak  
 (Sumber: Hasil Analisis, 2019)

#### 4. *Sequence Diagram* Menginput Data Surat Jalan

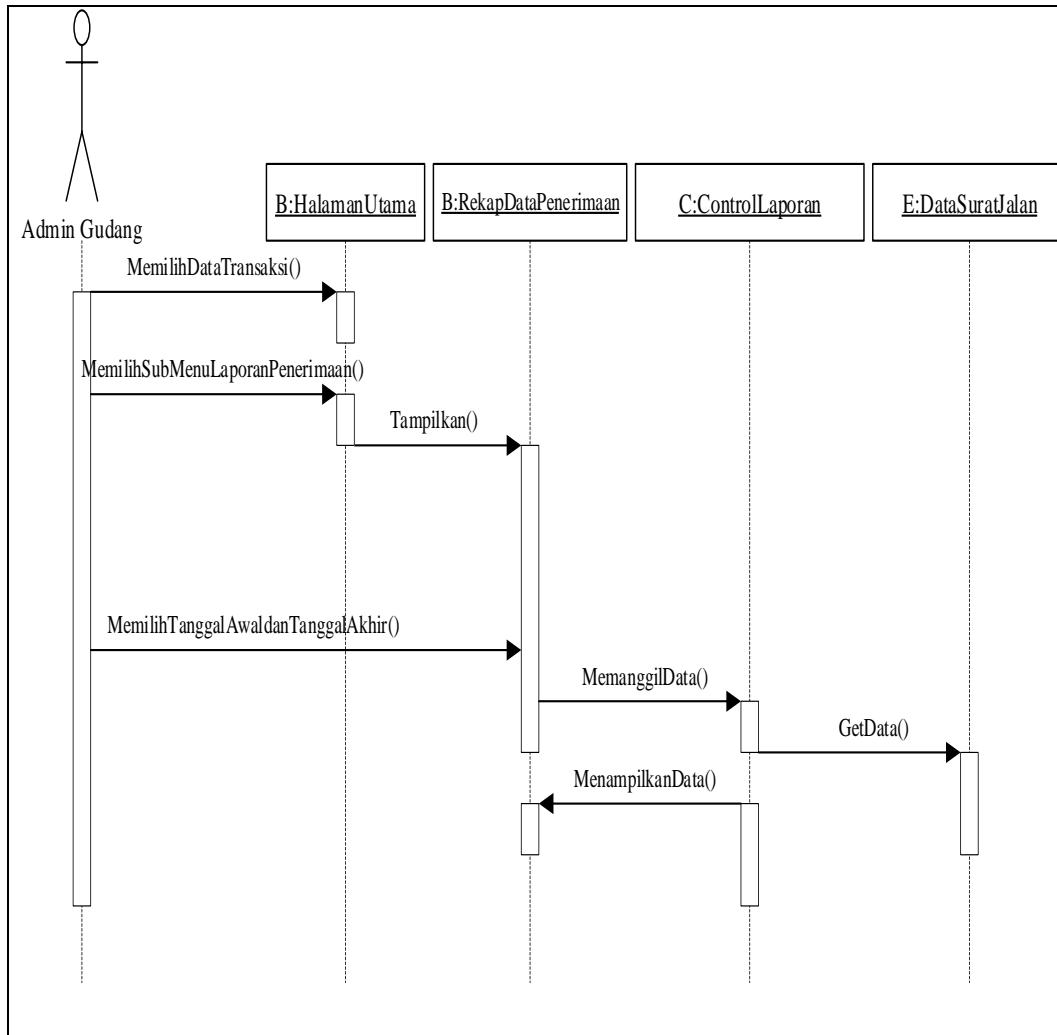
*Sequence diagram* menginput data surat jalan menggambarkan interaksi yang terjadi pada saat dilakukan proses menginput data surat jalan. *Sequence diagram* menginput data surat jalan memiliki beberapa kegiatan yang dapat dilakukan seperti menambah data surat jalan dan menghapus data surat jalan. *Sequence diagram* menginput data surat jalan digambarkan pada gambar V.13.



Gambar V.13 *Sequence Diagram* Menginput Data Surat Jalan  
 (Sumber: Hasil Analisis, 2019)

##### 5. *Sequence Diagram* Membuat Laporan Penerimaan Bahan Baku

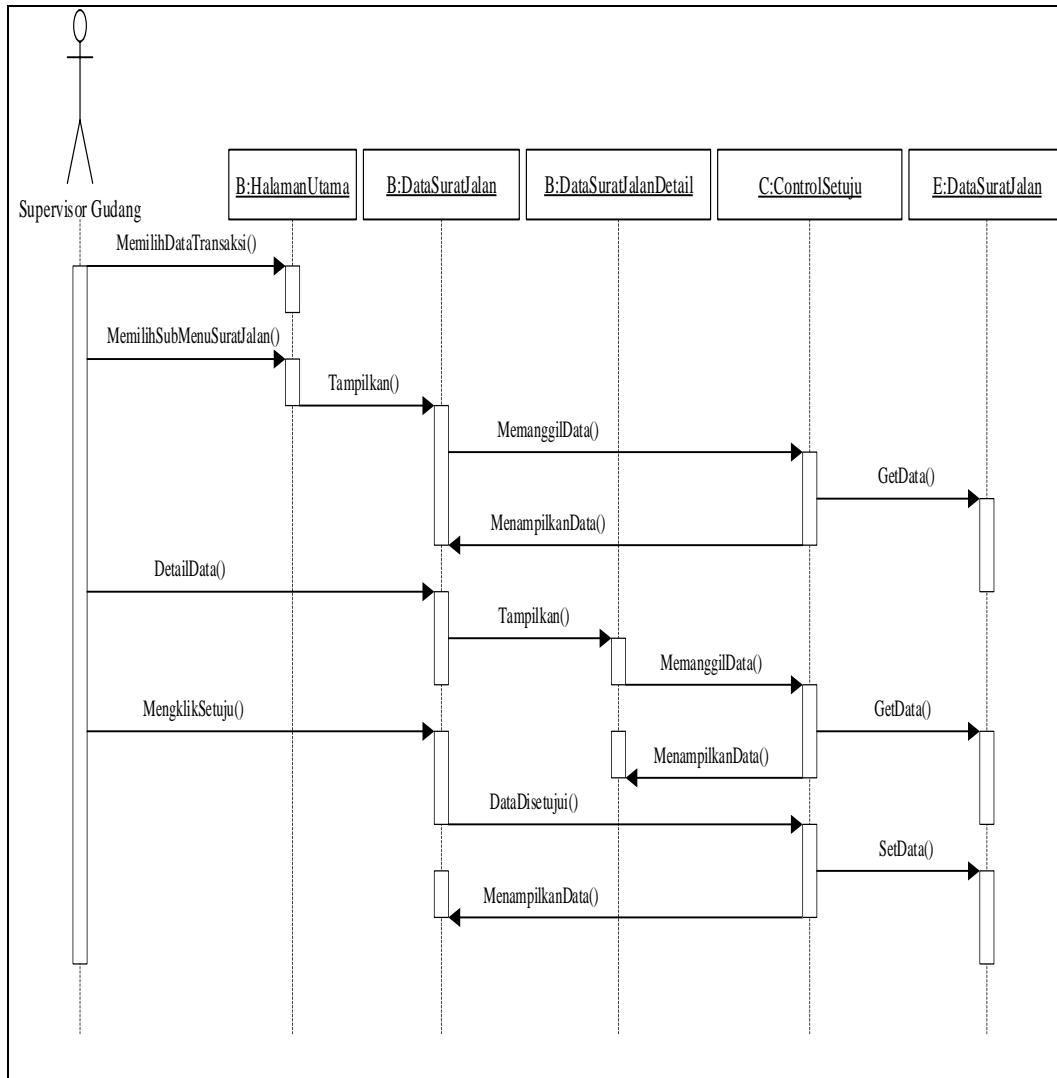
*Sequence diagram* membuat laporan penerimaan bahan baku menggambarkan interaksi yang terjadi pada saat dilakukan proses membuat laporan penerimaan bahan baku. *Sequence diagram* membuat laporan penerimaan bahan baku digambarkan pada gambar V.14.



Gambar V.14 *Sequence Diagram* Membuat Laporan Penerimaan  
 (Sumber: Hasil Analisis, 2019)

#### 6. *Sequence Diagram* Memberikan Persetujuan Surat Jalan

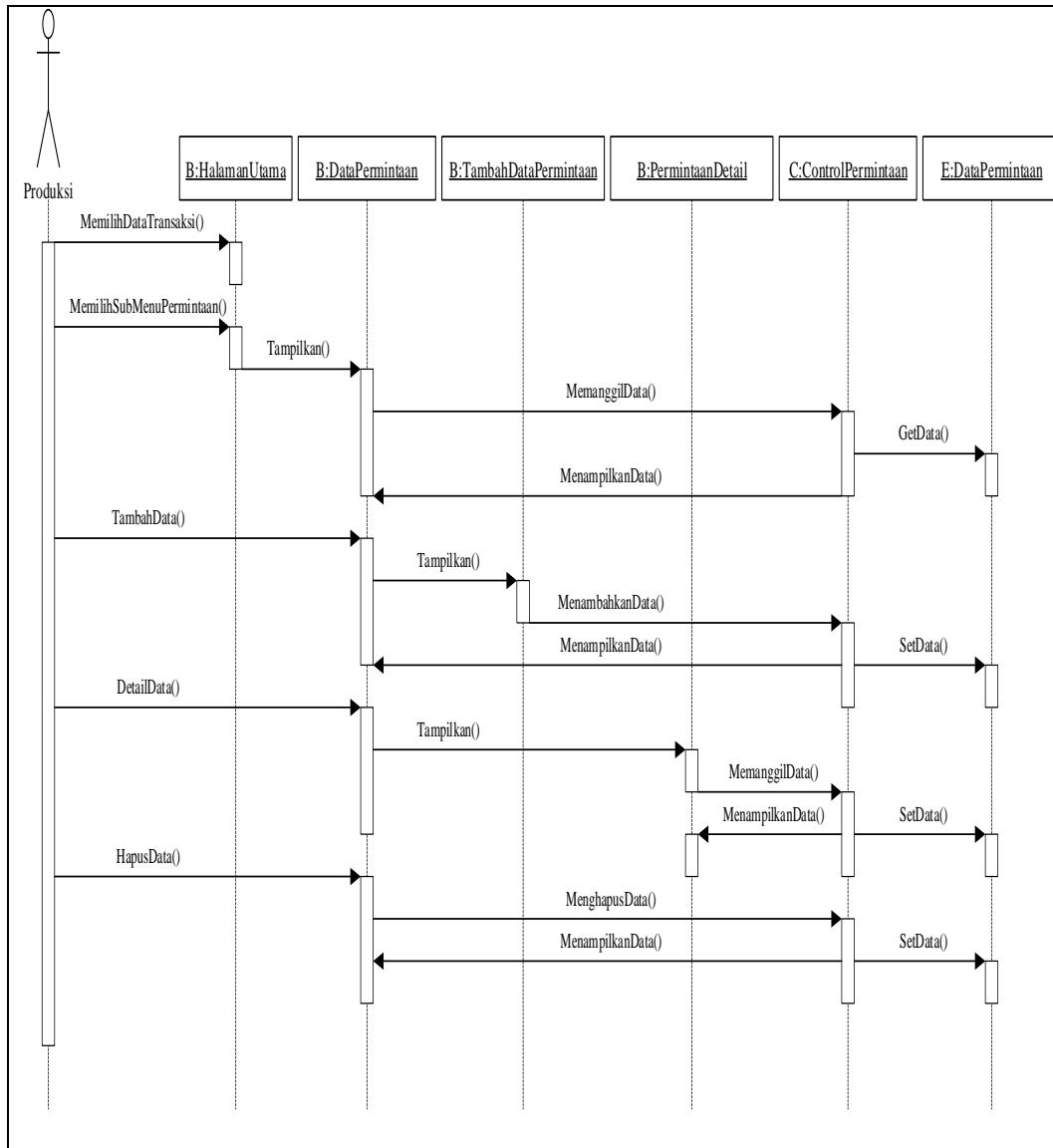
*Sequence diagram* memberikan persetujuan surat jalan menggambarkan interaksi yang terjadi pada saat dilakukan proses memberikan persetujuan surat jalan. *Sequence diagram* memberikan persetujuan surat jalan digambarkan pada gambar V.15.



Gambar V.15 *Sequence Diagram* Memberikan Persetujuan Surat Jalan  
 (Sumber: Hasil Analisis, 2019)

## 7. *Sequence Diagram* Membuat Bon Permintaan dan Penyerahan

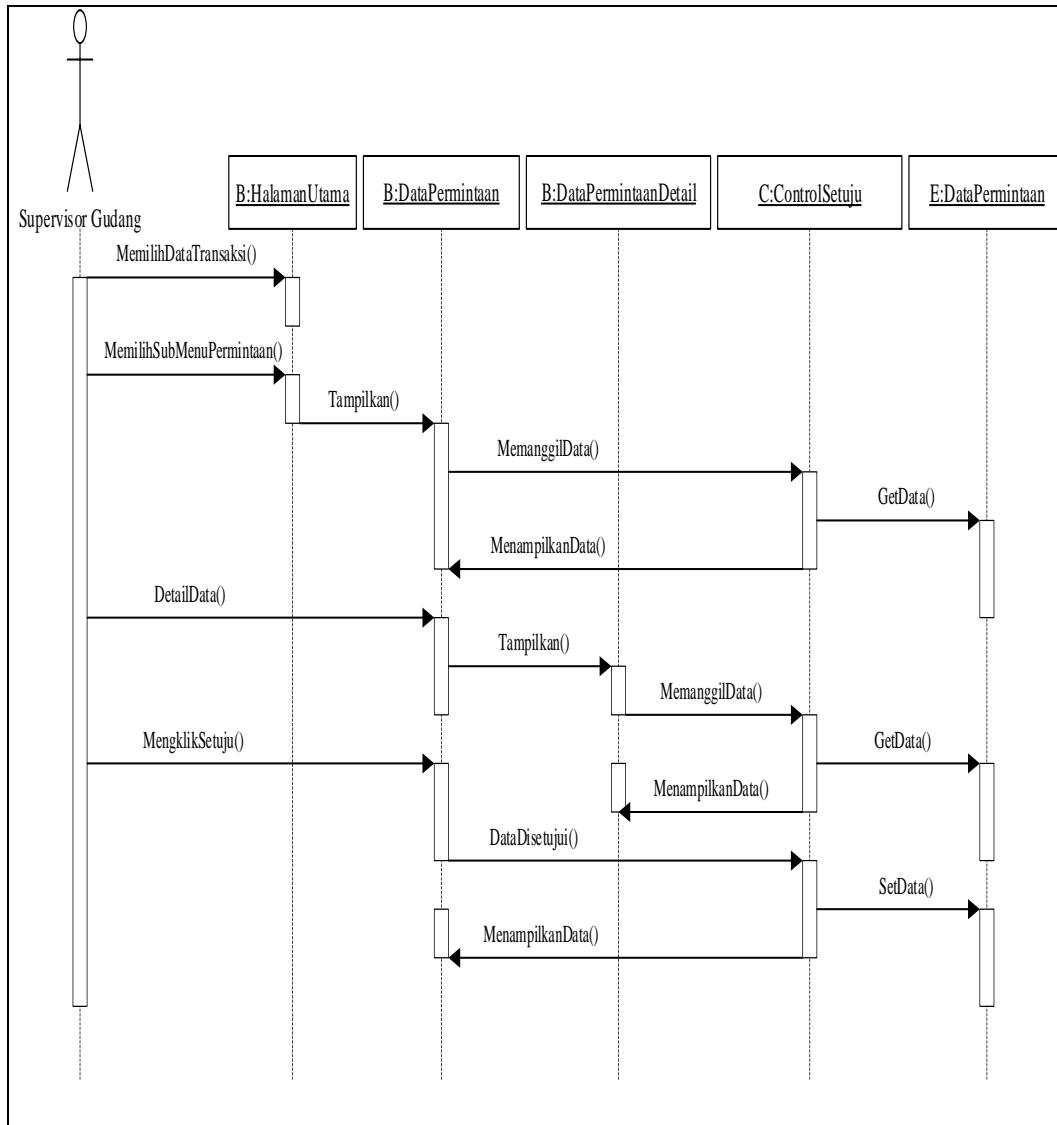
*Sequence diagram* membuat permintaan dan penyerahan menggambarkan interaksi yang terjadi pada saat dilakukan proses membuat permintaan dan penyerahan. *Sequence diagram* membuat permintaan dan penyerahan memiliki beberapa kegiatan yang dapat dilakukan seperti menambah data dan menghapus data. *Sequence diagram* membuat permintaan dan penyerahan digambarkan pada gambar V.16.



Gambar V.16 *Sequence Diagram* Membuat Bon Permintaan  
 (Sumber: Hasil Analisis, 2019)

#### 8. *Sequence Diagram* Memberikan Persetujuan Bon Permintaan

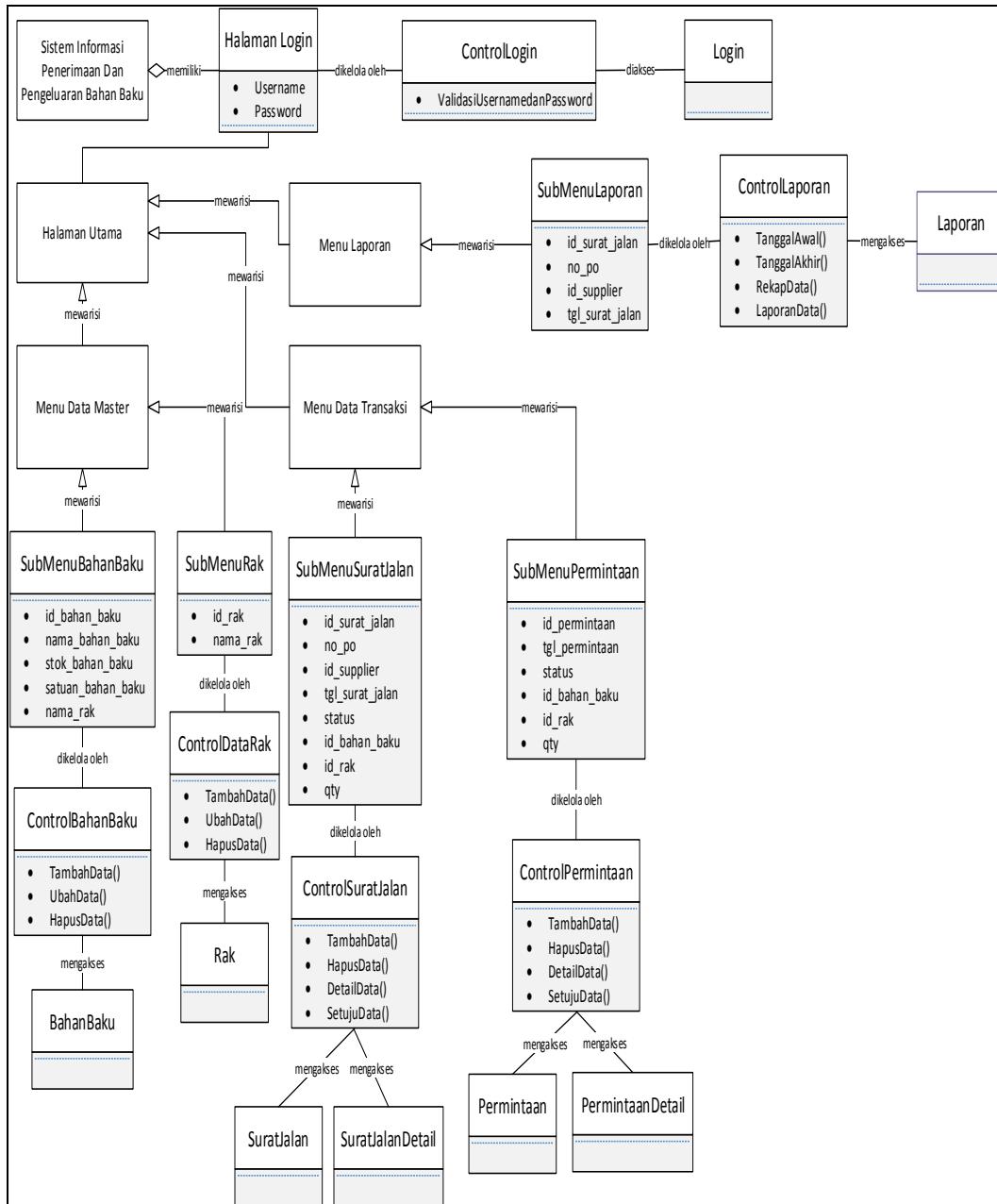
*Sequence diagram* memberikan persetujuan permintaan menggambarkan interaksi yang terjadi pada saat dilakukan proses memberikan persetujuan permintaan. *Sequence diagram* memberikan persetujuan permintaan digambarkan pada gambar V.17.



Gambar V.17 *Sequence Diagram* Memberikan Persetujuan Bon Permintaan  
(Sumber: Hasil Analisis, 2019)

#### 5.3.4 Class Diagram

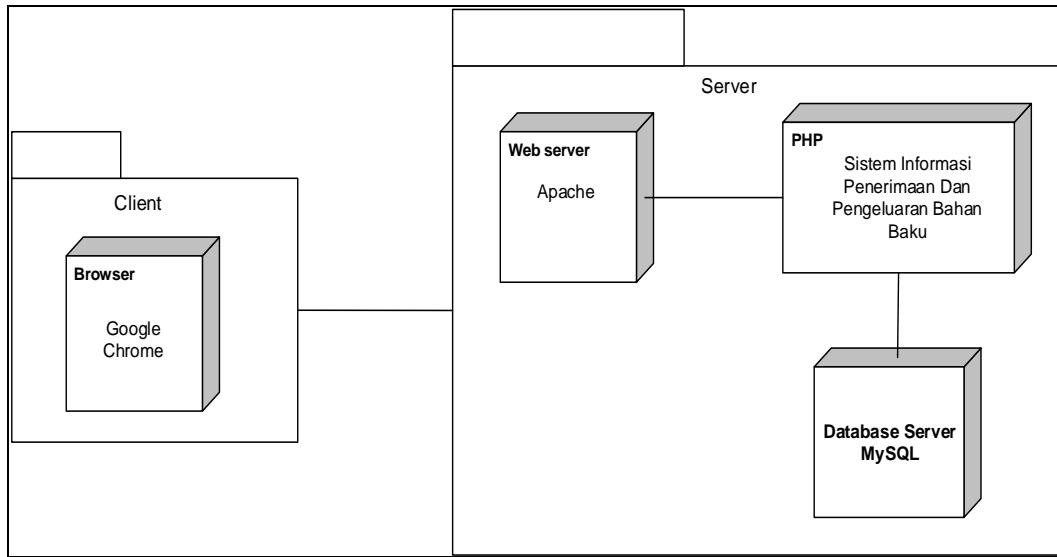
*Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem, *class diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem. *Class diagram* sistem informasi penerimaan dan pengeluaran bahan baku yang diusulkan dapat dilihat pada gambar V.18.



Gambar V.18 Class Diagram Sistem Informasi Manajemen Gudang Bahan Baku  
 (Sumber: Hasil Analisis, 2019)

### 5.3.5 Deployment Diagram

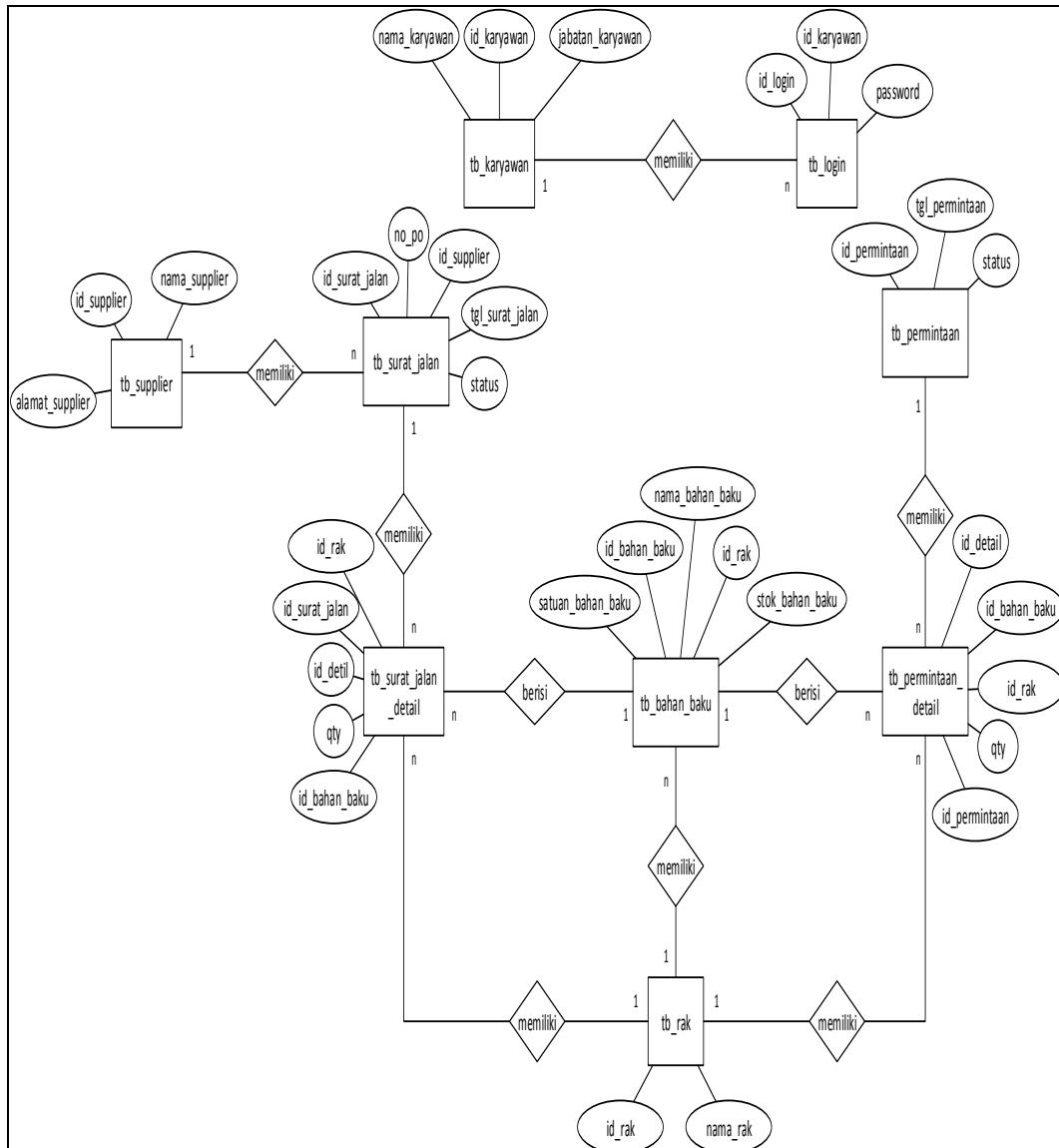
*Deployment diagram* digunakan pada bagian-bagian awal proses perancangan sistem untuk mendokumentasikan arsitektur fisik sebuah sistem. *Deployment diagram* sistem informasi penerimaan dan pengeluaran bahan baku yang diusulkan dapat dilihat pada gambar V.19.



Gambar V.19 *Deployment Diagram* Sistem Informasi Manajemen Gudang Bahan Baku  
 (Sumber: Hasil Analisis, 2019)

### 5.3.6 Entity Relationship Diagram (ERD)

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. Untuk menggembarkannya digunakan beberapa notasi dan simbol. ERD sistem informasi penerimaan dan pengeluaran bahan baku yang diusulkan dapat dilihat pada gambar V.20.

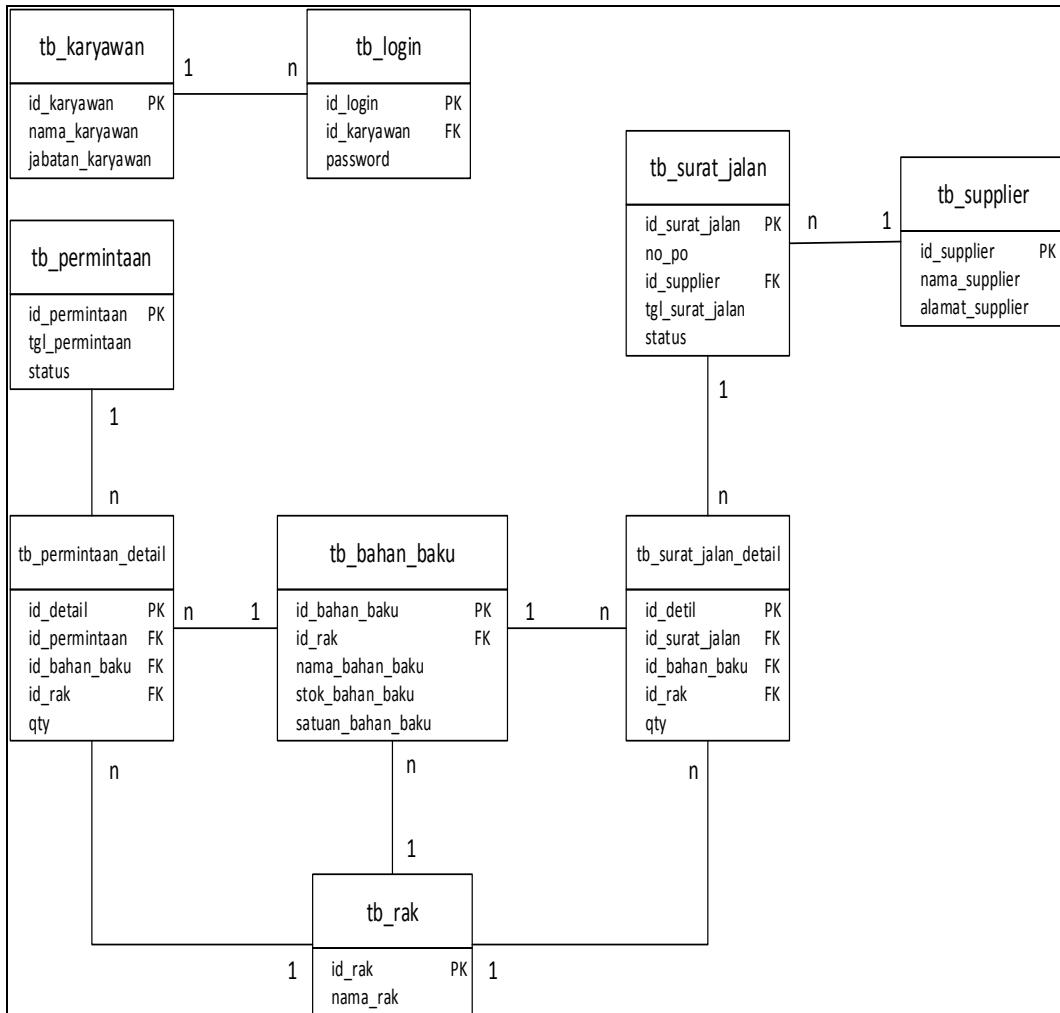


Gambar V.20 *Entity Relationship Diagram* Sistem Informasi Manajemen Gudang Bahan Baku

(Sumber: Hasil Analisis, 2019)

### 5.3.7 Conceptual Data Model (CDM)

*Conceptual Data Model* digunakan untuk menggambarkan atau merepresentasikan seluruh muatan informasi yang dikandung oleh basis data. Berikut ini merupakan *Conceptual Data Model* dari sistem informasi penerimaan dan pengeluaran bahan baku yang diusulkan (seperti gambar V.21).



Gambar V.21 *Conceptual Data Model* Sistem Informasi Manajemen Gudang Bahan Baku  
 (Sumber: Hasil Analisis, 2019)

### 5.3.8 Kamus Data

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

#### 1. Spesifikasi Tabel Data Bahan Baku

Nama Tabel : tb\_bahan\_baku

Fungsi : Untuk menyimpan data bahan baku

Tipe : File Data Master

Tabel V.10 Tabel Data Bahan Baku

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	ID Bahan Baku	id_bahan_baku	Varchar	20	Primary Key
2.	Nama Rak	id_rak	Varchar	20	Foreign Key
3.	Nama Bahan Baku	nama_bahan_baku	Varchar	20	Not Null
4.	Stok Bahan Baku	stok_bahan_baku	Int	20	Not Null
5.	Satuan Bahan Baku	satuan_bahan_baku	Varchar	20	Not Null

(Sumber: Hasil Analisis, 2019)

2. Spesifikasi Tabel Data *Supplier*

Nama Tabel : tb\_supplier

Fungsi : Untuk menyimpan data *supplier*

Tipe : File Data Master

Tabel V.11 Tabel Data *Supplier*

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	ID Supplier	id_supplier	Varchar	20	Primary Key
2.	Nama Supplier	nama_supplier	Varchar	30	Not Null
3.	Alamat Supplier	alamat_supplier	Varchar	70	Not Null

(Sumber: Hasil Analisis, 2019)

## 3. Spesifikasi Tabel Data Karyawan

Nama Tabel : tb\_karyawan

Fungsi : Untuk menyimpan data karyawan

Tipe : File Data Master

Tabel V.12 Tabel Data Karyawan

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	ID Karyawan	id_karyawan	Varchar	20	Primary Key
2.	Nama Karyawan	nama_karyawan	Varchar	20	Not Null
3.	Jabatan Karyawan	jabatan_karyawan	Varchar	20	Not Null

(Sumber: Hasil Analisis, 2019)

4. Spesifikasi Tabel Data Rak

Nama Tabel : tb\_rak

Fungsi : Untuk menyimpan data rak

Tipe : File Data Master

Tabel V.13 Tabel Data Rak

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	ID Rak	id_rak	Varchar	20	<i>Primary Key</i>
2.	Nama Rak	nama_rak	Varchar	20	<i>Not Null</i>

(Sumber: Hasil Analisis, 2019)

5. Spesifikasi Tabel Data *Login*

Nama Tabel : tb\_login

Fungsi : Untuk menyimpan data *login*

Tipe : File Data Master

Tabel V.14 Tabel Data *Login*

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	ID Login	id_login	Varchar	20	<i>Primary Key</i>
2.	ID Karyawan	id_karyawan	Varchar	20	<i>Foreign Key</i>
3.	Password	Password	Varchar	20	<i>Not Null</i>

(Sumber: Hasil Analisis, 2019)

6. Spesifikasi Tabel Data Surat Jalan

Nama Tabel : tb\_surat\_jalan

Fungsi : Untuk menyimpan data surat jalan

Tipe : File Data Transaksi

Tabel V.15 Tabel Data Surat Jalan

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	ID Surat Jalan	id_surat_jalan	Varchar	20	<i>Primary Key</i>
2.	No PO	no_po	Varchar	30	<i>Not Null</i>
3.	ID Supplier	id_supplier	Varchar	20	<i>Foreign Key</i>

Tabel V.15 Tabel Data Surat Jalan (lanjutan)

4.	Tanggal Surat Jalan	tgl_surat_jalan	Date		<i>Not Null</i>
5.	Status	Status	Varchar	20	<i>Not Null</i>

(Sumber: Hasil Analisis, 2019)

#### 7. Spesifikasi Tabel Data Surat Jalan Detail

Nama Tabel : tb\_surat\_jalan\_detail

Fungsi : Untuk menyimpan data surat jalan detail

Tipe : File Data Transaksi

Tabel V.16 Tabel Data Surat Jalan Detail

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	ID Surat Jalan Detail	id_detil	Int	11	<i>Primary Key</i>
2.	ID Surat Jalan	id_surat_jalan	Varchar	255	<i>Foreign Key</i>
3.	Nama Bahan Baku	id_bahan_baku	Varchar	20	<i>Foreign Key</i>
4.	Nama Rak	id_rak	Varchar	20	<i>Foreign Key</i>
5.	Qty	Qty	Int	11	<i>Not Null</i>

(Sumber: Hasil Analisis, 2019)

#### 8. Spesifikasi Tabel Data Permintaan

Nama Tabel : tb\_permintaan

Fungsi : Untuk menyimpan data permintaan

Tipe : File Data Transaksi

Tabel V.17 Tabel Data Permintaan

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	ID Permintaan	id_permintaan	Varchar	20	<i>Primary Key</i>
2.	Tanggal Permintaan	tgl_permintaan	Date		<i>Not Null</i>
3.	Status	Status	Varchar	20	<i>Not Null</i>

(Sumber: Hasil Analisis, 2019)

9. Spesifikasi Tabel Data Permintaan Detail

Nama Tabel : tb\_permintaan\_detail

Fungsi : Untuk menyimpan data permintaan detail

Tipe : File Data Transaksi

Tabel V.18 Tabel Data Permintaan Detail

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	ID Permintaan Detail	id_detail	Int	11	<i>Primary Key</i>
2.	ID Permintaan	id_permintaan	Varchar	20	<i>Foreign Key</i>
3.	Nama Bahan Baku	id_bahan_baku	Varchar	20	<i>Foreign Key</i>
4.	Nama Rak	id_rak	Varchar	20	<i>Foreign Key</i>
5.	Qty	qty	Int	11	<i>Not Null</i>

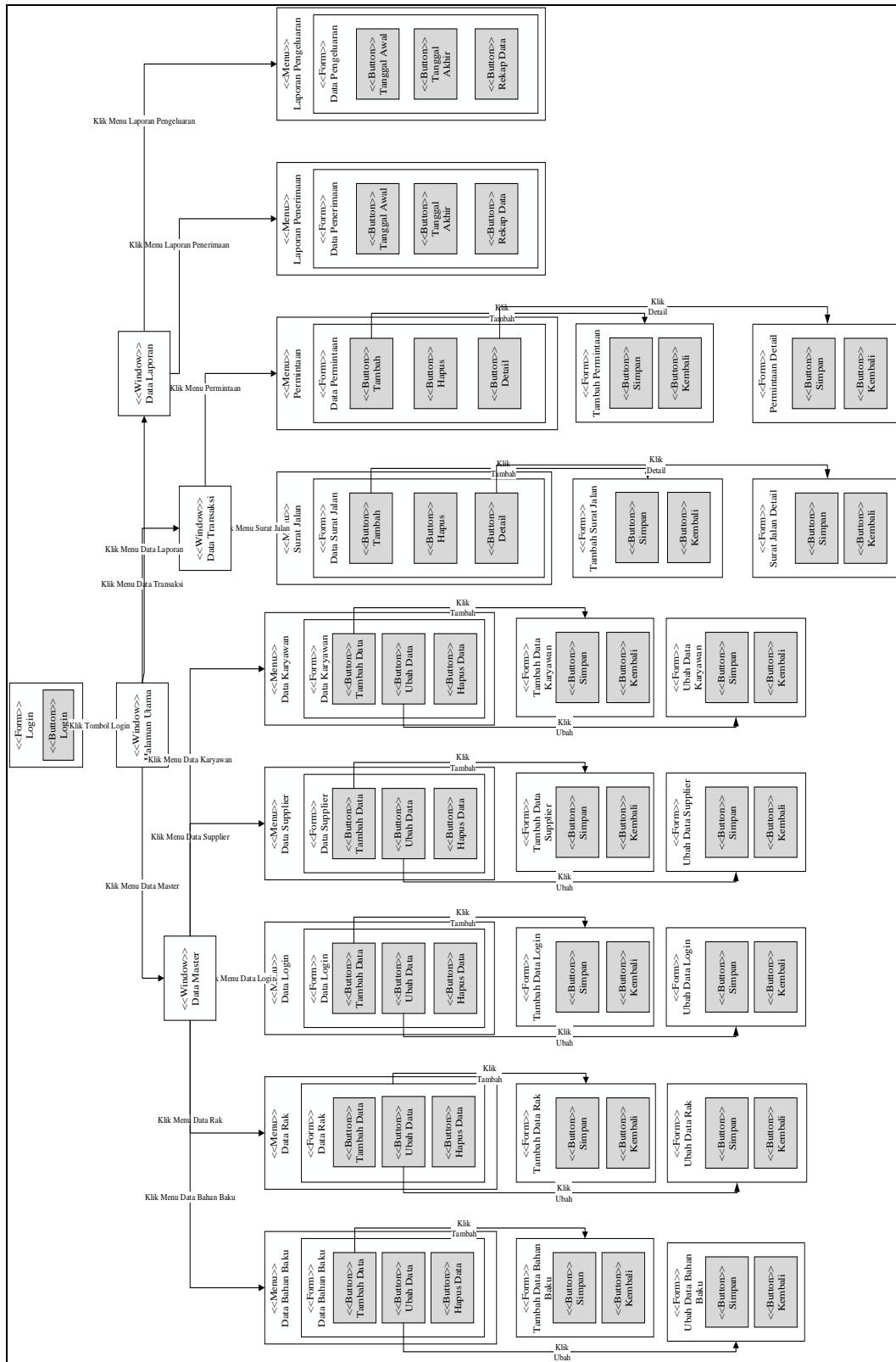
(Sumber: Hasil Analisis, 2019)

## 5.4 Analisis Desain Sistem

Tahap ini dimulai dengan analisis desain usulan yang meliputi pembuatan WND (*Windows Navigation Diagram*) dan *interface* aplikasi sistem informasi penerimaan dan pengeluaran bahan baku.

### 5.4.1 Windows Navigation Diagram (WND)

Dengan *Windows Navigation Diagram* kita dapat dengan mudah melihat skema sistem, sehingga akan memudahkan menganalisa sistem. Berikut ini merupakan contoh *Windows Navigation Diagram* usulan pada sistem informasi penerimaan dan pengeluaran bahan baku. Dapat dilihat pada gambar V.22 berikut ini.



Gambar V.22 Windows Navigation Diagram Sistem Informasi Manajemen Gudang Bahan Baku

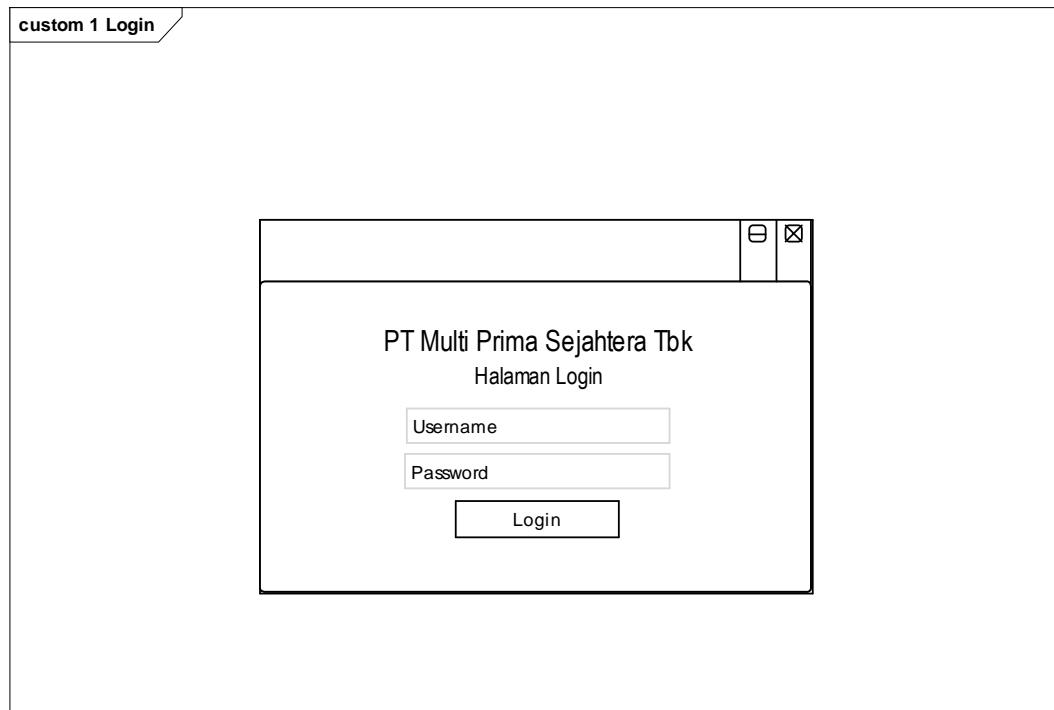
(Sumber: Hasil Analisis, 2019)

### 5.4.2 Perancangan *Interface* Sistem

Rancangan *interface* dari sistem informasi penerimaan dan pengeluaran bahan baku ini adalah sebagai berikut:

#### 1. *Form Login*

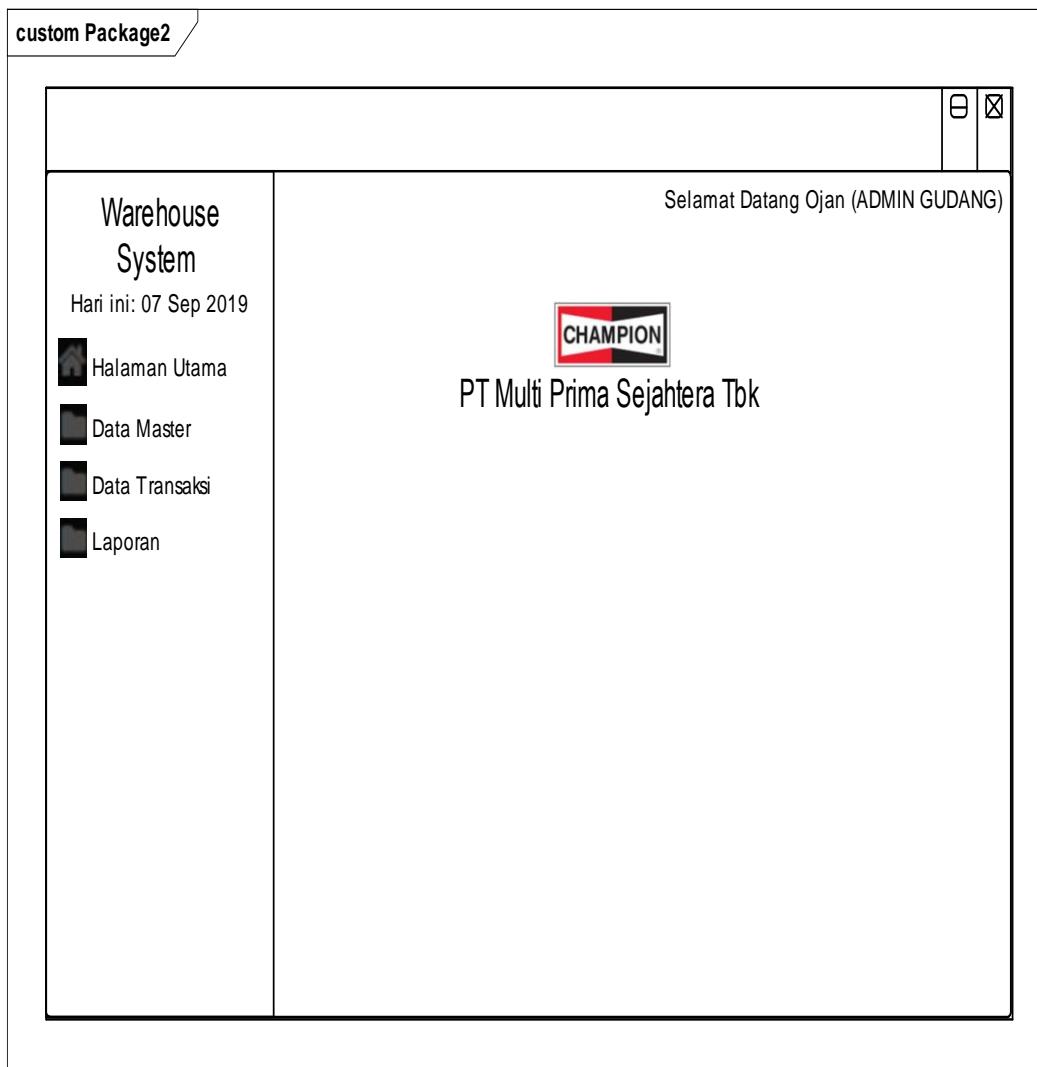
*Form login* adalah *form* yang digunakan untuk masuk ke dalam program aplikasi. Untuk masuk ke dalam aplikasi, pengguna harus memasukkan *username* dan *password* yang benar. Rancangan *form login* dapat dilihat pada gambar V.23 berikut ini:



Gambar V.23 *Interface Form Login*  
(Sumber: Hasil Analisis, 2019)

#### 2. *Form Halaman Utama*

*Form halaman utama* adalah tampilan awal pada aplikasi yang berisi menu-menu seperti Data Master, Data Transaksi, dan Laporan. Rancangan halaman utama dapat dilihat pada gambar V.24 berikut ini:



Gambar V.24 *Interface Halaman Utama*  
(Sumber: Hasil Analisis, 2019)

### 3. *Form Data Bahan Baku*

*Form* data bahan baku adalah *form* yang digunakan untuk mengelola data master bahan baku. Rancangan *interface* dari *form* data bahan baku dapat dilihat pada gambar V.25.

custom Webpage Wireframe

						<input type="checkbox"/>	<input checked="" type="checkbox"/>
<b>Warehouse System</b> Hari ini: 07 Sep 2019 <ul style="list-style-type: none"> <li><span style="color: #ccc;">#</span> Halaman Utama</li> <li><span style="color: #ccc;">#</span> Data Master</li> <li><span style="color: #ccc;">#</span> Data Transaksi</li> <li><span style="color: #ccc;">#</span> Laporan</li> </ul>	Selamat Datang Ojan (ADMIN GUDANG)						
	<b>Data Bahan Baku</b>						
	<b>Tambah Data Bahan Baku</b>						
	ID Bahan Baku	Nama Bahan Baku	Stok Bahan Baku	Satuan Bahan Baku	Nama Rak	Aksi	
						<span style="color: #ccc;">#</span> Ubah <span style="color: red;">#</span> Hapus	

Gambar V.25 *Interface Data Bahan Baku*  
 (Sumber: Hasil Analisis, 2019)

#### 4. *Form Data Rak*

*Form* data rak adalah *form* yang digunakan untuk mengelola data master rak. Rancangan *interface* dari *form* data rak dapat dilihat pada gambar V.26.

custom Webpage Wireframe

		Selamat Datang Ojan (ADMIN GUDANG)	
<p><b>Warehouse System</b></p> <p>Hari ini: 07 Sep 2019</p> <ul style="list-style-type: none"> <li><span style="color: #ccc;">#</span> Halaman Utama</li> <li><span style="color: #ccc;">#</span> Data Master</li> <li><span style="color: #ccc;">#</span> Data Transaksi</li> <li><span style="color: #ccc;">#</span> Laporan</li> </ul>	<p><b>Data Rak</b></p> <p><span style="border: 1px solid black; padding: 2px;">Tambah Data Rak</span></p>		
	<span style="border: 1px solid black; padding: 2px;">ID Rak</span>	<span style="border: 1px solid black; padding: 2px;">Nama Rak</span>	<span style="border: 1px solid black; padding: 2px;">Aksi</span>
			<span style="color: #ccc;">#</span> Ubah <span style="color: #ccc;">#</span> Hapus

Gambar V.26 *Interface Data Rak*

(Sumber: Hasil Analisis, 2019)

##### 5. *Form Data Surat Jalan*

*Form* data surat jalan adalah *form* yang digunakan untuk mengelola data transaksi surat jalan. Rancangan *interface* dari *form* data surat jalan dapat dilihat pada gambar V.27.

Gambar V.27 *Interface* Data Surat Jalan

(Sumber: Hasil Analisis, 2019)

## 6. *Form* Data Surat Jalan Detail

*Form* data surat jalan detail adalah *form* yang digunakan untuk mengelola data transaksi surat jalan detail. Rancangan *interface* dari *form* data surat jalan detail dapat dilihat pada gambar V.28.

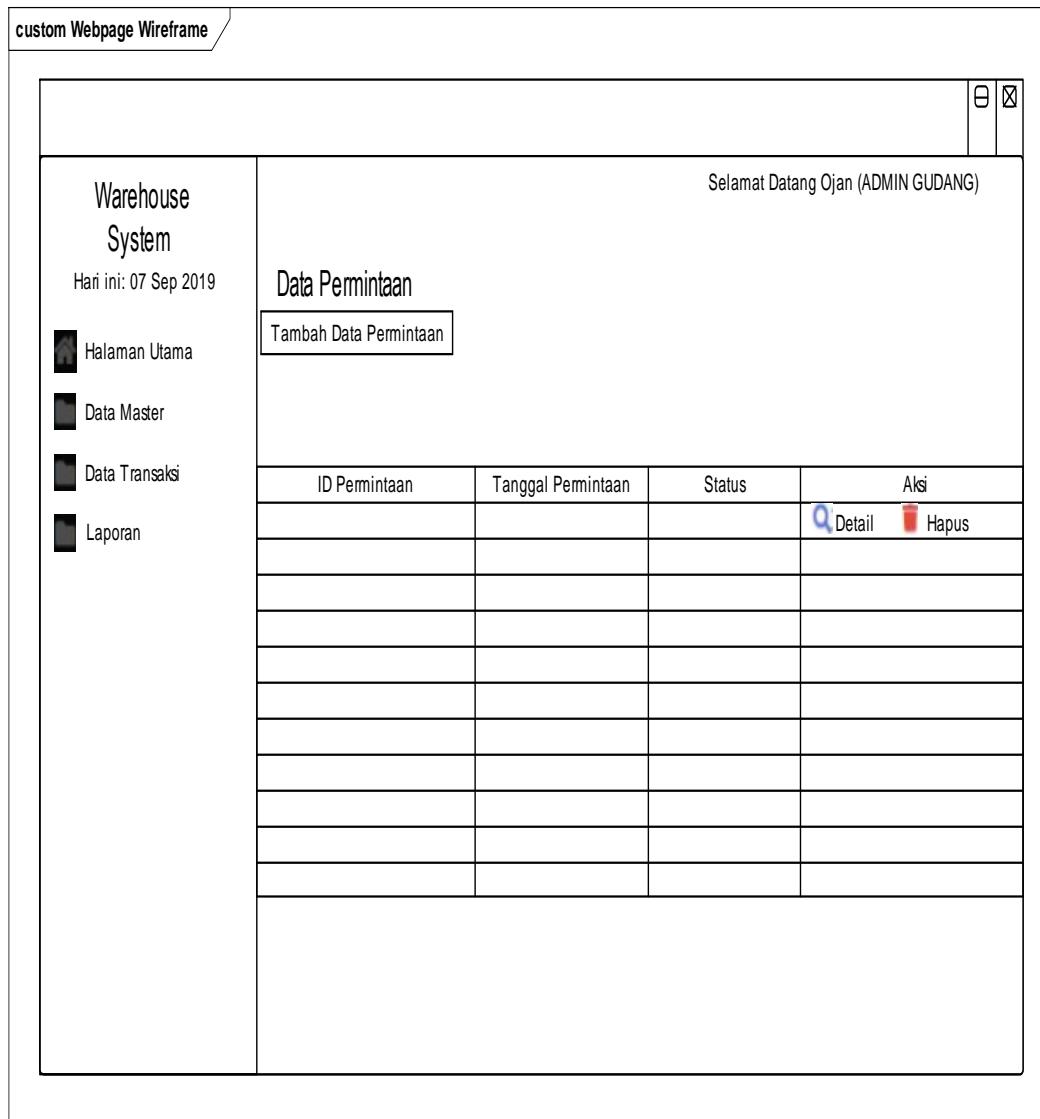
custom Webpage Wireframe

Warehouse System Hari ini: 07 Sep 20 <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Halaman Utama</li> <li><input type="checkbox"/> Data Master</li> <li><input type="checkbox"/> Data Transaksi</li> <li><input type="checkbox"/> Laporan</li> </ul>	Selamat Datang Ojan (ADM IN GUDANG) <b>Data Surat Jalan Detail</b> ID Surat Jalan <input type="text"/> Nama Bahan Baku <input type="text" value="Value1"/> <input type="button" value="▼"/> Rak <input type="text"/> Qty Bahan Baku <input type="text"/> <input type="button" value="Simpan"/> <input type="button" value="Kembali"/>																														
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>ID Surat Jalan Detail</th> <th>ID SuratJalan</th> <th>Nama Bahan Baku</th> <th>Qty</th> <th>Action</th> </tr> </thead> <tbody> <tr><td></td><td></td><td></td><td></td><td><input type="button" value="Hapus"/></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> </tbody> </table>		ID Surat Jalan Detail	ID SuratJalan	Nama Bahan Baku	Qty	Action					<input type="button" value="Hapus"/>																				
ID Surat Jalan Detail	ID SuratJalan	Nama Bahan Baku	Qty	Action																											
				<input type="button" value="Hapus"/>																											

Gambar V.28 *Interface* Data Surat Jalan Detail  
 (Sumber: Hasil Analisis, 2019)

## 7. Form Data Permintaan

*Form* data permintaan adalah *form* yang digunakan untuk mengelola data transaksi permintaan. Rancangan *interface* dari *form* data permintaan dapat dilihat pada gambar V.29.



Gambar V.29 *Interface Data Permintaan*

(Sumber: Hasil Analisis, 2019)

#### 8. *Form* Data Permintaan Detail

*Form* data permintaan detail adalah *form* yang digunakan untuk mengelola data transaksi permintaan detail. Rancangan *interface* dari *form* data permintaan detail dapat dilihat pada gambar V.30.

custom Webpage Wireframe

<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <b>Warehouse System</b>            Hari ini: 07 Sep 20         </div> <ul style="list-style-type: none"> <li><span style="color: #ccc;">House</span> Halaman Utama</li> <li><span style="color: #ccc;">Database</span> Data Master</li> <li><span style="color: #ccc;">Database</span> Data Transaksi</li> <li><span style="color: #ccc;">Database</span> Laporan</li> </ul>	<div style="text-align: right; margin-bottom: 10px;"> <span style="border: 1px solid #ccc; padding: 2px 5px; border-radius: 5px; margin-right: 10px;"></span> Selamat Datang Ojan (ADMIN GUDANG)         </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <b>Data Permintaan Detail</b> </div> <div style="display: flex; justify-content: space-between; align-items: center;"> <div style="flex: 1;">           ID Permintaan  <input type="text"/> </div> <div style="flex: 1;">           Nama Bahan Baku  <input type="text" value="Value1"/> </div> </div> <div style="display: flex; justify-content: space-between; align-items: center;"> <div style="flex: 1;">           Rak  <input type="text"/> </div> <div style="flex: 1;">           Qty Bahan Baku  <input type="text"/> </div> </div> <div style="display: flex; justify-content: space-around; align-items: center; margin-top: 10px;"> <span style="border: 1px solid #ccc; padding: 2px 5px; border-radius: 5px; margin-right: 10px;">Simpan</span> <span style="border: 1px solid #ccc; padding: 2px 5px; border-radius: 5px;">Kembali</span> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>ID Permintaan Detail</th> <th>ID Permintaan</th> <th>Nama Bahan Baku</th> <th>Qty</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td style="text-align: center;"><span style="color: red;">Delete</span> Hapus</td> </tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table> </div>	ID Permintaan Detail	ID Permintaan	Nama Bahan Baku	Qty	Action					<span style="color: red;">Delete</span> Hapus																									
ID Permintaan Detail	ID Permintaan	Nama Bahan Baku	Qty	Action																																
				<span style="color: red;">Delete</span> Hapus																																

Gambar V.30 *Interface* Data Permintaan Detail  
 (Sumber: Hasil Analisis, 2019)

## 5.5 Pengujian Sistem

Pengujian sistem yang dilakukan untuk sistem informasi penerimaan dan pengeluaran bahan baku dapat dilakukan menggunakan *blackbox testing*. *Blackbox testing* merupakan pengujian apakah fungsionalitas dari aplikasi yang dibangun sesuai dengan apa yang diharapkan tanpa menguji desain dan kode program. *Blackbox testing* menguji sistem berdasarkan *use case* yang telah dibuat. Untuk lebih lanjut mengenai proses pengujian *blackbox testing* pada sistem informasi penerimaan dan pengeluaran bahan baku dapat dilihat pada Lampiran B.

## **BAB VI**

### **KESIMPULAN DAN SARAN**

#### **6.1 Kesimpulan**

Berdasarkan hasil rancang bangun sistem yang dilakukan mengenai sistem informasi penerimaan dan pengeluaran bahan baku pada PT Multi Prima Sejahtera Tbk dapat disimpulkan beberapa hal, yaitu:

1. Sistem informasi penerimaan dan pengeluaran bahan baku dengan media penyimpanan data dalam bentuk *database* diharapkan dapat menyimpan data lebih aman dan terstruktur.
2. Sistem informasi penerimaan dan pengeluaran bahan baku yang telah dibangun diharapkan dapat memberikan kemudahan pengolahan data bahan baku yang diterima dan dikeluarkan sehingga proses penerimaan dan pengeluaran bahan baku menjadi lebih optimal.

#### **6.2 Saran**

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

1. Mengimplementasikan hasil dari penelitian berupa sistem informasi untuk proses penerimaan dan pengeluaran bahan baku pada Bagian Gudang di PT Multi Prima Sejahtera Tbk.
2. Sistem informasi penerimaan dan pengeluaran bahan baku dapat disosialisasikan kepada *user* agar sistem informasi ini berjalan dengan baik sehingga proses penerimaan dan pengeluaran bahan baku menjadi lebih optimal.

**LAMPIRAN A**  
**DAFTAR HADIR KEGIATAN PRAKTIK KERJA LAPANGAN**

1	NAMA MAHASISWA	HENDRA DANOPAN
2	NIM	1315029
3	PROGRAM STUDI	SISTEM INFORMASI INDUSTRI OTOMOTIF

1	NAMA PERUSAHAAN	PT MULTI PRIMA SEJAHTERA TBK
2	ALAMAT/ TELEPON/ EMAIL	JL. RAYA TLAJUNG UDIK NO.454 GUNUNG PUTRI, BOGOR 16962
3	DEPARTMEN/ UNIT KERJA	GUDANG
4	NAMA PEMBIMBING/ HP	SANGGAR SADEWO

Tanggal	Uraian Kegiatan	Waktu Datang	Waktu Pulang	Paraf
08/08/2018	Melakukan Pengamatan Proses Penerimaan Bahan Baku Pada Bagian Gudang	08 : 00	15 : 00	<i>lui</i>
09/08/2018	Melakukan Pengamatan Proses Penerimaan Bahan Baku Pada Bagian Gudang	08 : 00	15 : 00	<i>lui</i>
10/08/2018	Melakukan Pengamatan Proses Pengeluaran Produk Pada Bagian Gudang	08 : 00	15 : 00	<i>lui</i>
13/08/2018	Melakukan Pengamatan Proses Pengeluaran Produk Pada Bagian Gudang	08 : 00	15 : 00	<i>lui</i>
14/08/2018	Melakukan Pengamatan Proses Pengeluaran Bahan Baku Pada Bagian Gudang	08 : 00	15 : 00	<i>lui</i>
15/08/2018	Melakukan Pengamatan Proses Pengeluaran Bahan Baku Pada Bagian Gudang	08 : 00	15 : 00	<i>lui</i>
16/08/2018	Melakukan Pengamatan Proses <i>Quality Assurance On Production</i> Pada Bagian QA	08 : 00	15 : 00	<i>lui</i>
20/08/2018	Melakukan Pengamatan Proses <i>Quality Assurance After Production</i> Pada Bagian QA	08 : 00	15 : 00	<i>lui</i>
21/08/2018	Melakukan Pengamatan Proses <i>Quality Assurance After Production</i> Pada Bagian QA	08 : 00	15 : 00	<i>lui</i>
23/08/2018	Melakukan Pengamatan Proses Produksi Pada Bagian Produksi	08 : 00	15 : 00	<i>lui</i>

24/08/2018	Melakukan Pengamatan Proses Produksi Pada Bagian Produksi	08 : 00	15 : 00	<i>lui</i>
27/08/2018	Melakukan Pengamatan Proses Produksi Pada Bagian Produksi	08 : 00	15 : 00	<i>lui</i>
28/08/2018	Melakukan Pengamatan Proses Produksi Pada Bagian Produksi	08 : 00	15 : 00	<i>lui</i>
29/08/2018	Melakukan Pengamatan Proses <i>Maintenance</i> Mesin Pada Bagian <i>Maintenance</i>	08 : 00	15 : 00	<i>lui</i>
30/08/2018	Melakukan Pengamatan Proses <i>Maintenance</i> Mesin Pada Bagian <i>Maintenance</i>	08 : 00	15 : 00	<i>lui</i>
31/08/2018	Melakukan Pengamatan Proses <i>Maintenance</i> Mesin Pada Bagian <i>Maintenance</i>	08 : 00	15 : 00	<i>lui</i>
03/09/2018	Melakukan Pengamatan Proses <i>Maintenance</i> Mesin Pada Bagian <i>Maintenance</i>	08 : 00	15 : 00	<i>lui</i>
04/09/2018	Melakukan Pengamatan Proses Pada Bagian HRD	08 : 00	15 : 00	<i>lui</i>
05/09/2018	Melakukan Pengamatan Proses Penyimpanan Bahan Baku Pada Bagian Gudang	08 : 00	15 : 00	<i>lui</i>
06/09/2018	Melakukan Pengamatan Proses Penyimpanan Bahan Baku Pada Bagian Gudang	08 : 00	15 : 00	<i>lui</i>
07/09/2018	Melakukan Pengamatan Proses Penyimpanan Bahan Baku Pada Bagian Gudang	08 : 00	15 : 00	<i>lui</i>

Bogor, 07 September 2018

Pembimbing Praktik Kerja Lapangan

*lui*  
PT. MULTI PRIMA SEJAHTERA Tbk

( Sanggar Sadewo )

## **LAMPIRAN B**

### ***BLACK BOX TESTING***

#### 1. Halaman *Login*

Deskripsi : Menguji fungsi pada halaman *login*.

Pengujii : Bagas Eko Saputro (1315051)

No.	Test Case	Expected Result	Actual Result	Result
1.	Mengklik tombol <i>login</i> tanpa memasukkan <i>username</i> dan <i>password</i>	Sistem akan menampilkan pesan “Coba lagi, kombinasi ID Bahan baku dan Password Salah”	Sistem menampilkan pesan “Coba lagi, kombinasi ID Bahan baku dan Password Salah”	Valid
2.	Mengklik tombol <i>login</i> dengan hanya memasukkan <i>username</i>	Sistem akan menampilkan pesan “Coba lagi, kombinasi ID Bahan baku dan Password Salah”.	Sistem menampilkan pesan “Coba lagi, kombinasi ID Bahan baku dan Password Salah”.	Valid
3.	Mengklik tombol <i>login</i> dengan hanya memasukkan <i>password</i>	Sistem akan menampilkan pesan “Coba lagi, kombinasi ID Bahan baku dan Password Salah”.	Sistem menampilkan pesan “Coba lagi, kombinasi ID Bahan baku dan Password Salah”.	Valid
4.	Menginput <i>username</i> dengan benar sedangkan menginput <i>password</i> salah atau sebaliknya, lalu mengklik tombol <i>login</i>	Sistem menampilkan pesan “Coba lagi, kombinasi ID Bahan baku dan Password Salah”.	Sistem menampilkan pesan “Coba lagi, kombinasi ID Bahan baku dan Password Salah”.	Valid
5.	Menginput <i>username</i> dan <i>password</i> dengan benar, lalu mengklik tombol <i>login</i>	Sistem akan menerima akses <i>login</i> dan menampilkan menu utama berdasarkan hak aksesnya.	Sistem menerima akses <i>login</i> dan menampilkan menu utama berdasarkan hak aksesnya.	Valid

#### 2. Menu Halaman Utama

Deskripsi : Menguji fungsi pada menu Halaman Utama.

Penguji : Bagas Eko Saputro (1315051)

No.	Test Case	Expected Result	Actual Result	Result
1.	Mengklik menu halaman utama	Sistem akan menampilkan halaman utama.	Sistem menampilkan halaman utama.	Valid

### 3. Sub Menu Bahan baku

Deskripsi : Menguji fungsi pada sub menu bahan baku.

Penguji : Bagas Eko Saputro (1315051)

No.	Test Case	Expected Result	Actual Result	Result
1.	Mengklik sub menu bahan baku	Sistem akan menampilkan halaman data bahan baku.	Sistem menampilkan halaman data bahan baku.	Valid
2.	Menambah data bahan baku dengan mengklik tombol tambah data bahan baku	Tombol dapat diklik dan sistem akan menampilkan form data bahan baku.	Tombol dapat diklik dan sistem menampilkan form data bahan baku.	Valid
3.	Mengklik tombol simpan dengan mengkosongkan semua atau beberapa kolom	Data tidak dapat disimpan dan akan menampilkan pesan "Please fill out this field".	Data tidak dapat disimpan dan menampilkan pesan "Please fill out this field".	Valid
4.	Mengisi kolom isian lalu mengklik tombol simpan.	Data dapat disimpan dan akan menampilkan halaman data bahan baku.	Data dapat disimpan dan menampilkan halaman data bahan baku.	Valid
5.	Mengubah data bahan baku dengan mengklik tombol ubah.	Tombol dapat diklik dan sistem akan menampilkan ubah data bahan baku.	Tombol dapat diklik dan sistem menampilkan ubah data bahan baku.	Valid

No.	Test Case	Expected Result	Actual Result	Result
6.	Melakukan	Tombol dapat	Tombol dapat	Valid

	perubahan data bahan baku dengan mengisi semua <i>field</i> lalu mengklik tombol simpan.	diklik dan data bahan baku akan diubah.	diklik dan sistem mengubah data bahan baku.	
7.	Mengklik tombol kembali pada form kelola data bahan baku	Tombol dapat diklik dan akan kembali ke halaman data bahan baku.	Tombol dapat diklik dan dapat kembali ke halaman data bahan baku.	<i>Valid</i>
8.	Menghapus data bahan baku dengan mengklik tombol hapus	Sistem akan menampilkan pesan “anda yakin mau menghapus data ini?”.	Sistem menampilkan pesan “anda yakin mau menghapus data ini?”.	<i>Valid</i>
9.	Mengklik tombol cancel pada kotak dialog hapus	Tombol dapat diklik dan data akan batal untuk dihapus	Tombol dapat diklik dan data batal untuk dihapus	<i>Valid</i>
10.	Mengklik tombol hapus pada kotak dialog hapus	Tombol dapat diklik dan data akan terhapus.	Tombol dapat diklik dan data terhapus dari tabel data bahan baku.	<i>Valid</i>

#### 4. Sub Menu Rak

Deskripsi : Menguji fungsi pada sub menu rak.

Penguji : Bagas Eko Saputro (1315051)

No.	Test Case	Expected Result	Actual Result	Result
1.	Mengklik sub menu rak	Sistem akan menampilkan halaman data rak.	Sistem menampilkan halaman data rak.	<i>Valid</i>
2.	Menambah data rak dengan mengklik tombol tambah data rak	Tombol dapat diklik dan sistem akan menampilkan <i>form</i> data rak.	Tombol dapat diklik dan sistem menampilkan <i>form</i> data rak.	<i>Valid</i>

No.	Test Case	Expected Result	Actual Result	Result
3.	Mengklik tombol simpan dengan mengkosongkan semua atau beberapa kolom	Data tidak dapat disimpan dan akan menampilkan pesan “Please fill out this field”.	Data tidak dapat disimpan dan menampilkan pesan “Please fill out this field”.	Valid
4.	Mengisi kolom isian lalu mengklik tombol simpan.	Data dapat disimpan dan akan menampilkan halaman data rak.	Data dapat disimpan dan menampilkan halaman data rak.	Valid
5.	Mengubah data rak dengan mengklik tombol ubah.	Tombol dapat diklik dan sistem akan menampilkan ubah data rak.	Tombol dapat diklik dan sistem menampilkan ubah data rak.	Valid
6.	Melakukan perubahan data rak dengan mengisi semua <i>field</i> lalu mengklik tombol simpan.	Tombol dapat diklik dan data rak akan diubah.	Tombol dapat diklik dan sistem mengubah data rak.	Valid
7.	Mengklik tombol kembali pada form kelola data rak	Tombol dapat diklik dan akan kembali ke halaman data rak.	Tombol dapat diklik dan dapat kembali ke halaman data rak.	Valid
8.	Menghapus data rak dengan mengklik tombol hapus	Sistem akan menampilkan pesan “anda yakin mau menghapus data ini?”.	Sistem menampilkan pesan “anda yakin mau menghapus data ini?”.	Valid
9.	Mengklik tombol cancel pada kotak dialog hapus	Tombol dapat diklik dan data akan batal untuk dihapus	Tombol dapat diklik dan data batal untuk dihapus	Valid
10.	Mengklik tombol hapus pada kotak dialog hapus	Tombol dapat diklik dan data akan terhapus.	Tombol dapat diklik dan data terhapus dari tabel data rak.	Valid

## 5. Sub Menu *Supplier*

Deskripsi : Menguji fungsi pada sub menu *supplier*.

Pengujii : Bagas Eko Saputro (1315051)

No.	Test Case	Expected Result	Actual Result	Result
1.	Mengklik sub menu <i>supplier</i>	Sistem akan menampilkan halaman data <i>supplier</i> .	Sistem menampilkan halaman data <i>supplier</i> .	Valid
2.	Menambah data <i>supplier</i> dengan mengklik tombol tambah data <i>supplier</i>	Tombol dapat diklik dan sistem akan menampilkan form data <i>supplier</i> .	Tombol dapat diklik dan sistem menampilkan form data <i>supplier</i> .	Valid
3.	Mengklik tombol simpan dengan mengkosongkan semua atau beberapa kolom	Data tidak dapat disimpan dan akan menampilkan pesan “Please fill out this field”.	Data tidak dapat disimpan dan menampilkan pesan “Please fill out this field”.	Valid
4.	Mengisi kolom isian lalu mengklik tombol simpan.	Data dapat disimpan dan akan menampilkan halaman data <i>supplier</i> .	Data dapat disimpan dan menampilkan halaman data <i>supplier</i> .	Valid
5.	Mengubah data <i>supplier</i> dengan mengklik tombol ubah.	Tombol dapat diklik dan sistem akan menampilkan ubah data <i>supplier</i> .	Tombol dapat diklik dan sistem menampilkan ubah data <i>supplier</i> .	Valid
6.	Melakukan perubahan data <i>supplier</i> dengan mengisi semua <i>field</i> lalu mengklik tombol simpan.	Tombol dapat diklik dan data <i>supplier</i> akan diubah.	Tombol dapat diklik dan sistem mengubah data <i>supplier</i> .	Valid

No.	Test Case	Expected Result	Actual Result	Result
7.	Mengklik tombol kembali pada form kelola data <i>supplier</i>	Tombol dapat diklik dan akan kembali ke halaman data <i>supplier</i> .	Tombol dapat diklik dan dapat kembali ke halaman data <i>supplier</i> .	Valid
8.	Menghapus data <i>supplier</i> dengan mengklik tombol hapus	Sistem akan menampilkan pesan “anda yakin mau menghapus data ini?”.	Sistem menampilkan pesan “anda yakin mau menghapus data ini?”.	Valid
9.	Mengklik tombol cancel pada kotak dialog hapus	Tombol dapat diklik dan data akan batal untuk dihapus	Tombol dapat diklik dan data batal untuk dihapus	Valid
10.	Mengklik tombol hapus pada kotak dialog hapus	Tombol dapat diklik dan data akan terhapus.	Tombol dapat diklik dan data terhapus dari tabel data <i>supplier</i> .	Valid

## 6. Sub Menu Karyawan

Deskripsi : Menguji fungsi pada sub menu karyawan.

Penguji : Bagas Eko Saputro (1315051)

No.	Test Case	Expected Result	Actual Result	Result
1.	Mengklik sub menu karyawan	Sistem akan menampilkan halaman data karyawan.	Sistem menampilkan halaman data karyawan.	Valid
2.	Menambah data karyawan dengan mengklik tombol tambah data karyawan	Tombol dapat diklik dan sistem akan menampilkan form data karyawan.	Tombol dapat diklik dan sistem menampilkan form data karyawan.	Valid
3.	Mengklik tombol simpan dengan mengkosongkan semua atau beberapa kolom	Data tidak dapat disimpan dan akan menampilkan pesan “Please fill out this field”.	Data tidak dapat disimpan dan menampilkan pesan “Please fill out this field”.	Valid

No.	<i>Test Case</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Result</i>
4.	Mengisi kolom isian lalu mengklik tombol simpan.	Data dapat disimpan dan akan menampilkan halaman data karyawan.	Data dapat disimpan dan menampilkan halaman data karyawan.	<i>Valid</i>
5.	Mengubah data karyawan dengan mengklik tombol ubah.	Tombol dapat diklik dan sistem akan menampilkan ubah data karyawan.	Tombol dapat diklik dan sistem menampilkan ubah data karyawan.	<i>Valid</i>
6.	Melakukan perubahan data karyawan dengan mengisi semua <i>field</i> lalu mengklik tombol simpan.	Tombol dapat diklik dan data karyawan akan diubah.	Tombol dapat diklik dan sistem mengubah data karyawan.	<i>Valid</i>
7.	Mengklik tombol kembali pada form kelola data karyawan	Tombol dapat diklik dan akan kembali ke halaman data karyawan.	Tombol dapat diklik dan dapat kembali ke halaman data karyawan.	<i>Valid</i>
8.	Menghapus data karyawan dengan mengklik tombol hapus	Sistem akan menampilkan pesan “anda yakin mau menghapus data ini?”.	Sistem menampilkan pesan “anda yakin mau menghapus data ini?”.	<i>Valid</i>
9.	Mengklik tombol cancel pada kotak dialog hapus	Tombol dapat diklik dan data akan batal untuk dihapus	Tombol dapat diklik dan data batal untuk dihapus	<i>Valid</i>
10.	Mengklik tombol hapus pada kotak dialog hapus	Tombol dapat diklik dan data akan terhapus.	Tombol dapat diklik dan data terhapus dari tabel data karyawan.	<i>Valid</i>

## 7. Sub Menu *Login*

Deskripsi : Menguji fungsi pada sub menu *login*.

Penguji : Bagas Eko Saputro (1315051)

No.	Test Case	Expected Result	Actual Result	Result
1.	Mengklik sub menu <i>login</i>	Sistem akan menampilkan halaman data <i>login</i> .	Sistem menampilkan halaman data <i>login</i> .	Valid
2.	Menambah data <i>login</i> dengan mengklik tombol tambah data <i>login</i>	Tombol dapat diklik dan sistem akan menampilkan form data <i>login</i> .	Tombol dapat diklik dan sistem menampilkan form data <i>login</i> .	Valid
3.	Mengklik tombol simpan dengan mengkosongkan semua atau beberapa kolom	Data tidak dapat disimpan dan akan menampilkan pesan "Please fill out this field".	Data tidak dapat disimpan dan menampilkan pesan "Please fill out this field".	Valid
4.	Mengisi kolom isian lalu mengklik tombol simpan.	Data dapat disimpan dan akan menampilkan halaman data <i>login</i> .	Data dapat disimpan dan menampilkan halaman data <i>login</i> .	Valid
5.	Mengubah data <i>login</i> dengan mengklik tombol ubah.	Tombol dapat diklik dan sistem akan menampilkan ubah data <i>login</i> .	Tombol dapat diklik dan sistem menampilkan ubah data <i>login</i> .	Valid
6.	Melakukan perubahan data <i>login</i> dengan mengisi semua <i>field</i> lalu mengklik tombol simpan.	Tombol dapat diklik dan data <i>login</i> akan diubah.	Tombol dapat diklik dan sistem mengubah data <i>login</i> .	Valid
7.	Mengklik tombol kembali pada form kelola data <i>login</i>	Tombol dapat diklik dan akan kembali ke halaman data <i>login</i> .	Tombol dapat diklik dan dapat kembali ke halaman data <i>login</i> .	Valid
8.	Menghapus data <i>login</i> dengan mengklik tombol hapus	Sistem akan menampilkan pesan "anda yakin mau menghapus	Sistem menampilkan pesan "anda yakin mau menghapus	Valid

		menghapus data ini?".	data ini?".	
No.	Test Case	Expected Result	Actual Result	Result
9.	Mengklik tombol cancel pada kotak dialog hapus	Tombol dapat diklik dan data akan batal untuk dihapus	Tombol dapat diklik dan data batal untuk dihapus	Valid
10.	Mengklik tombol hapus pada kotak dialog hapus	Tombol dapat diklik dan data akan terhapus.	Tombol dapat diklik dan data terhapus dari tabel data login.	Valid

## 8. Sub Menu Surat Jalan

Deskripsi : Menguji fungsi pada sub menu surat jalan.

Pengujii : Bagas Eko Saputro (1315051)

No.	Test Case	Expected Result	Actual Result	Result
1.	Mengklik sub menu surat jalan	Sistem akan menampilkan halaman data surat jalan.	Sistem menampilkan halaman data surat jalan.	Valid
2.	Menambah data surat jalan dengan mengklik tombol tambah data surat jalan	Tombol dapat diklik dan sistem akan menampilkan form data surat jalan.	Tombol dapat diklik dan sistem menampilkan form data surat jalan.	Valid
3.	Mengklik tombol simpan dengan mengkosongkan semua atau beberapa kolom	Data tidak dapat disimpan dan akan menampilkan pesan "Please fill out this field".	Data tidak dapat disimpan dan menampilkan pesan "Please fill out this field".	Valid
4.	Mengisi kolom isian lalu mengklik tombol simpan.	Data dapat disimpan dan akan menampilkan halaman data surat jalan.	Data dapat disimpan dan menampilkan halaman data surat jalan.	Valid

No.	<i>Test Case</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Result</i>
5.	Mengklik tombol setuju pada halaman data surat jalan	Tombol dapat diklik dan sistem akan menampilkan pesan “Data berhasil disetujui”.	Tombol dapat diklik dan sistem akan menampilkan pesan “Data berhasil disetujui”.	Valid
6.	Mengklik tombol detail pada halaman data surat jalan	Tombol dapat diklik dan sistem akan menampilkan halaman Data Surat Jalan Detail.	Tombol dapat diklik dan sistem menampilkan halaman Data Surat Jalan Detail.	Valid
7.	Mengklik tombol kembali pada halaman data surat jalan	Tombol dapat diklik dan akan kembali ke halaman data surat jalan.	Tombol dapat diklik dan dapat kembali ke halaman data surat jalan.	Valid

#### 9. Sub Menu Permintaan

Deskripsi : Menguji fungsi pada sub menu permintaan.

Penguji : Bagas Eko Saputro (1315051)

No.	<i>Test Case</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Result</i>
1.	Mengklik sub menu permintaan	Sistem akan menampilkan halaman data permintaan.	Sistem menampilkan halaman data permintaan.	Valid
2.	Mengklik tombol detail pada halaman data permintaan	Tombol dapat diklik dan sistem akan menampilkan halaman Data Permintaan Detail.	Tombol dapat diklik dan sistem menampilkan halaman Data Permintaan Detail.	Valid
3.	Mengklik tombol kembali pada halaman data permintaan	Tombol dapat diklik dan akan kembali ke halaman data permintaan.	Tombol dapat diklik dan dapat kembali ke halaman data permintaan	Valid

## 10. Sub Menu Data Penerimaan

Deskripsi : Menguji fungsi pada sub menu data pelatihan.

Penguji : Bagas Eko Saputro (1315051)

No.	Test Case	Expected Result	Actual Result	Result
1.	Mengklik sub menu penerimaan	Sistem akan menampilkan halaman rekap data penerimaan.	Sistem menampilkan halaman rekap data penerimaan.	Valid
2.	Mengklik tombol rekap data pada halaman data penerimaan	Tombol dapat diklik dan sistem akan menampilkan Data Penerimaan.	Tombol dapat diklik dan sistem akan menampilkan Data Penerimaan.	Valid

## 11. Sub Menu Data Pengeluaran

Deskripsi : Menguji fungsi pada sub menu buat kebutuhan.

Penguji : Bagas Eko Saputro (1315051)

No.	Test Case	Expected Result	Actual Result	Result
1.	Mengklik sub menu pengeluaran	Sistem akan menampilkan halaman rekap bon permintaan dan penyerahan bahan baku.	Sistem menampilkan halaman halaman rekap bon permintaan dan penyerahan bahan baku.	Valid
2.	Mengklik tombol rekap data pada halaman rekap bon permintaan dan penyerahan bahan baku.	Tombol dapat diklik dan sistem akan menampilkan rekap bon permintaan dan penyerahan bahan baku..	Tombol dapat diklik dan sistem akan menampilkan rekap bon permintaan dan penyerahan bahan baku.	Valid

## **LAMPIRAN C**

### **TRANSKRIP KODE PROGRAM**

#### **1. Surat Jalan**

##### **a. Model (m\_suratjalan.php)**

```
<?php

class m_suratjalan extends CI_model {

    public function kode()
    {
        $q      =      $this->db->query("select
MAX(RIGHT(id_surat_jalan,3))      as      code_max      from
tb_surat_jalan");
        $code = "";
        if($q->num_rows()>0){
            foreach($q->result() as $cd){
                $tmp = ((int)$cd->code_max)+1;
                $code = sprintf("%03s", $tmp);
            }
        }else{
            $code = "001";
        }
        return "SJ".$code;
    }

    function get_rak_bykode($kode) {
        $hsl=$this->db->query("SELECT
tb_bahan_baku.id_bahan_baku,          tb_bahan_baku.id_rak,
tb_rak.nama_rak, tb_bahan_baku.nama_bahan_baku
        FROM `tb_bahan_baku`, tb_rak
        WHERE tb_bahan_baku.id_rak = tb_rak.id_rak AND
tb_bahan_baku.id_bahan_baku = '$kode'");
        if($hsl->num_rows()>0){
            foreach ($hsl->result() as $data) {
                $hasil=array(
                    'id_rak'      => $data->id_rak,
                    'nama_rak'    => $data->nama_rak
                );
                return $hasil;
            }
        }
    }

    public function kodi()
    {
        $q      =      $this->db->query("select
MAX(RIGHT(no_po,3))  as codi_max  from tb_surat_jalan");
        $codi = "";
        if($q->num_rows()>0){
            foreach($q->result() as $cd){
                $tmp = ((int)$cd->codi_max)+1;
                $codi = sprintf("%03s", $tmp);
            }
        }
    }
}
```

```

        }else{
            $codi = "001";
        }
        return "PO".$codi;
    }

    public function tampilkan_data()
    {
        $this->db->select('*');
        $this->db->from('tb_surat_jalan');
        $this->db->join('tb_supplier',
'tb_supplier.id_supplier = tb_surat_jalan.id_supplier');
        $this->db->order_by('id_surat_jalan', 'DESC');
        return $this->db->get();
    }

    public function tampilkan_data_dtl()
    {
        $a = $this->uri->segment('3');
        $hasil = $this->db->query("SELECT
tb_surat_jalan_detail.id_detil,
tb_surat_jalan_detail.id_surat_jalan,
tb_bahan_baku.nama_bahan_baku, tb_rak.nama_rak,
tb_surat_jalan_detail.qty
FROM tb_surat_jalan_detail, tb_bahan_baku,
tb_rak
WHERE tb_surat_jalan_detail.id_bahan_baku
= tb_bahan_baku.id_bahan_baku AND
tb_surat_jalan_detail.id_rak =
tb_rak.id_rak
AND tb_surat_jalan_detail.id_surat_jalan =
'$a' order by tb_surat_jalan_detail.id_detil desc");
        return $hasil;
    }

    function construct()
    {
        parent::__construct();
    }

    function get()
    {
        $query = $this->db->query('SELECT * FROM
tb_supplier');
        return $query->result();
    }

    function got()
    {
        $query = $this->db->query('SELECT * FROM
tb_bahan_baku');
        return $query->result();
    }

```

```

public function simpan($data)
{
    $this->db->insert('tb_surat_jalan', $data);
}

public function simpan_detil($data)
{
    $this->db->insert('tb_surat_jalan_detail',
$data);
}

public function hapus($id)
{
    $this->db->delete('tb_surat_jalan',
['id_surat_jalan' => $id]);
    $this->db->delete('tb_surat_jalan_detail',
['id_surat_jalan' => $id]);
}

public function hapus_detail($id)
{
    $this->db->delete('tb_surat_jalan_detail',
['id_detil' => $id]);
}

public function get_data($id)
{
    $vans = array('id_surat_jalan' => $id);
    $this->db->select('*');
    $this->db->from('tb_surat_jalan');
    $this->db->join('tb_supplier',
'tb_supplier.id_supplier = tb_surat_jalan.id_supplier');
    $this->db->join('tb_bahan_baku',
'tb_bahan_baku.id_bahan_baku = tb_surat_jalan.id_bahan_baku');
    $this->db->join('tb_rak', 'tb_rak.id_rak = tb_surat_jalan.id_rak');
    return $this->db->get_where('', $vans);
}

public function isi()
{
    $a = $this->uri->segment('3');
    $hasil = $this->db->query("SELECT
tb_surat_jalan.id_surat_jalan, tb_surat_jalan.no_po,
tb_supplier.nama_supplier, tb_supplier.alamat_supplier,
tb_surat_jalan.status, tb_surat_jalan.tgl_surat_jalan
FROM tb_surat_jalan, tb_supplier
WHERE
tb_surat_jalan.id_supplier = tb_supplier.id_supplier
AND tb_surat_jalan.id_surat_jalan = '$a'");

    return $hasil;
}

```

```

public function detail()
{
    $a = $this->uri->segment('3');
    $hasil = $this->db->query("SELECT
tb_surat_jalan.id_surat_jalan,tb_surat_jalan_detail.id_deti
l,tb_surat_jalan_detail.id_bahan_baku,tb_bahan_baku.satuan_
bahan_baku,
tb_supplier.nama_supplier,
tb_supplier.alamat_supplier, tb_bahan_baku.nama_bahan_baku,
tb_rak.nama_rak, tb_surat_jalan_detail.qty
FROM tb_surat_jalan,tb_surat_jalan_detail,
tb_bahan_baku, tb_rak, tb_supplier
WHERE tb_surat_jalan_detail.id_bahan_baku
= tb_bahan_baku.id_bahan_baku AND
tb_surat_jalan.id_surat_jalan =
tb_surat_jalan_detail.id_surat_jalan AND
tb_surat_jalan_detail.id_rak =
tb_rak.id_rak AND
tb_surat_jalan.id_supplier =
tb_supplier.id_supplier
AND tb_surat_jalan.id_surat_jalan =
'$a'");

    return $hasil;
}

function update($data, $id)
{
    $this->db->where('id_surat_jalan', $id);
    $this->db->update('tb_surat_jalan', $data);
}

function validasi($id)
{
    $this->db->query("UPDATE tb_surat_jalan SET status='1'
where id_surat_jalan='$id'");

    $bnyak = $this->db->query("SELECT
id_bahan_baku,qty FROM tb_surat_jalan_detail WHERE
ID_SURAT_JALAN = '$id'");
    foreach ($bnyak->result() as $bb) {
        $this->db->query("UPDATE tb_bahan_baku SET
stok_bahan_baku = stok_bahan_baku +".$bb->qty. " WHERE
id_bahan_baku=". "'$bb->id_bahan_baku'");
    }
}

function update_minus($id)
{

    $bnyak = $this->db->query("SELECT id_bahan_baku
FROM tb_surat_jalan_detail WHERE ID_SURAT_JALAN = '$id'");
    foreach ($bnyak->result() as $bb) {

```

```

        $cek = $this->db->query("SELECT * FROM
tb_bahan_baku WHERE id_bahan_baku = '". $bb-
>id_bahan_baku."'");
        foreach ($cek->result() as $bc) {
            if ($bc->stok_bahan_baku > 0 && $bc-
>kekurangan < 0) {
                $stok = $bc->stok_bahan_baku;
                $kurang = $bc->kekurangan;
                $this->db->query("UPDATE
tb_bahan_baku SET kekurangan = ".$kurang." + ".$stok.,
stok_bahan_baku = ".$stok." +
".$kurang." WHERE id_bahan_baku='". $bc->id_bahan_baku."'");
            }
        }
        $cek = $this->db->query("SELECT * FROM
tb_bahan_baku WHERE id_bahan_baku = '". $bb-
>id_bahan_baku."'");
        foreach ($cek->result() as $bd) {
            if ($bd->stok_bahan_baku < 0) {
                $this->db->query("UPDATE
tb_bahan_baku SET
stok_bahan_baku = 0 WHERE
id_bahan_baku='". $bc->id_bahan_baku."'");
            }
            if ($bd->kekurangan > 0) {
                $this->db->query("UPDATE
tb_bahan_baku SET
kekurangan = 0 WHERE
id_bahan_baku='". $bc->id_bahan_baku."'");
            }
        }
    }
}

```

## b. Controller (suratjalan.php)

```

<?php

class suratjalan extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model('m_suratjalan');
        $this->load->library('form_validation');
    }

    public function index()
    {
        $data['code'] = $this->m_suratjalan->kode();
        $data['codi'] = $this->m_suratjalan->kodi();
        $data['record'] = $this->m_suratjalan-
>tampilkan_data();
        $data['id_supplier'] = $this->m_suratjalan-
>get();
    }
}

```

```

        $this->template-
>load('template','suratjalan/lihat_data', $data);
}

public function index_detail()
{
    $data['record']      =      $this->m_suratjalan-
>tampilkan_data();
    $data['id_bahan_baku'] =  $this->m_suratjalan-
>got();
    $this->template-
>load('template','suratjalan/lihat_data', $data);
}

public function validasi()
{
    $id = $this->uri->segment(3);
    $simpan = $this->m_suratjalan->validasi($id);
    $simpan      =      $this->m_suratjalan-
>update_minus($id);
    $this->session->set_flashdata('message',   '<div
class="alert alert-warning" role="alert">
                Data Berhasil Disetujui<button
type="button" class="close" data-dismiss="alert" aria-
label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>');
    redirect('suratjalan');
}

public function tambah()
{
    $data['code'] = $this->m_suratjalan->kode();
    $data['codi'] = $this->m_suratjalan->kodi();
    $data['id_supplier'] = $this->m_suratjalan-
>get();
    $this->template->load('template',
'suratjalan/form_input', $data);
}

public function tambah_detail()
{
    $data['code']      =      $this->m_suratjalan-
>kode();
    $data['bahan_baku'] =  $this->m_suratjalan-
>got();
    $this->template->load('template',
'suratjalan/form_detail', $data);
}

public function detail($id)
{
    $data['judul'] = 'Detail Data Surat Jalan';
    $data['suratjalan'] = $this->m_suratjalan-
>detail($id);
}

```

```

        $this->template->load('template',
'suratjalan/detail_data', $data);
    }

    public function simpan()
    {
        $id = $this-
>input->post('id_surat_jalan');
        $no_po = $this-
>input->post('no_po');
        $id_supplier = $this-
>input->post('id_supplier');
        $tgl_surat_jalan = $this->input-
>post('tgl_surat_jalan');
        $validasi_surat_jalan = $this->input-
>post('validasi_surat_jalan');

        $data = array('id_surat_jalan' => $id, 'no_po' => $no_po, 'id_supplier' => $id_supplier, 'tgl_surat_jalan' => $tgl_surat_jalan, 'status' => 0);
        $simpan = $this->m_suratjalan->simpan($data);
        $this->session->set_flashdata('message', '<div class="alert alert-success" role="alert">
            Data Berhasil Ditambahkan <button type="button" class="close" data-dismiss="alert" aria-label="Close">
                <span aria-hidden="true">&times;</span>
            </button>
        </div>');
        redirect('suratjalan/form_detail/'.$id);
    }

    public function simpan_detil()
    {
        $id_surat_jalan = $this-
>input->post('id_surat_jalan');
        $id_bahan_baku = $this-
>input->post('id_bahan_baku');
        $id_rak = $this->input->post('id_rak');
        $qty_bahan_baku = $this-
>input->post('qty');

        $data = array('id_surat_jalan' => $id_surat_jalan, 'id_bahan_baku' => $id_bahan_baku, 'id_rak' => $id_rak, 'qty' => $qty_bahan_baku);
        $simpan = $this->m_suratjalan->simpan_detil($data);
        $this->session->set_flashdata('message', '<div class="alert alert-success" role="alert">
            Data Berhasil Ditambahkan <button type="button" class="close" data-dismiss="alert" aria-label="Close">
        </div>');
    }
}

```

```

        <span aria-hidden="true">&times;</span>
    </button>
</div>');
    redirect($_SERVER['HTTP_REFERER']);
}

function form_detail(){
    $data['bahan_baku'] = $this->m_suratjalan-
>got();
    $data['record'] = $this->m_suratjalan-
>tampilkan_data_dtl();
    $this->template->load('template',
'suratjalan/form_detail',$data);
}

function get_rak(){
    $a = $this->uri->segment(3);
    $data = $this->m_suratjalan->get_rak_bykode($a);
    echo json_encode($data);
}

public function hapus($id)
{
    $id = $this->uri->segment(3);
    $this->m_suratjalan->hapus($id);
    $this->session->set_flashdata('message', '<div
class="alert alert-danger" role="alert">
    Data Berhasil Dihapus <button
type="button" class="close" data-dismiss="alert" aria-
label="Close">
        <span aria-hidden="true">&times;</span>
    </button>
</div>');
    redirect('suratjalan');
}

public function hapus_detail($id)
{
    $id = $this->uri->segment(3);
    $this->m_suratjalan->hapus_detail($id);
    $this->session->set_flashdata('message', '<div
class="alert alert-danger" role="alert">
    Data Berhasil Dihapus <button
type="button" class="close" data-dismiss="alert" aria-
label="Close">
        <span aria-hidden="true">&times;</span>
    </button>
</div>');
    redirect($_SERVER['HTTP_REFERER']);
}

public function ubah()
{
    $id = $this->uri->segment(3);
    $data['id_supplier'] = $this->m_suratjalan-
>get();
}

```

```

        $data['id_bahan_baku'] = $this->m_suratjalan-
>got();
        $data['record'] = $this->m_suratjalan-
>get_data($id)->row_array();
        $this->template->load('template',
'suratjalan/form_edit', $data);
    }

    public function edit()
    {
        $id = $this-
>input->post('id_surat_jalan');
        $no_po = $this-
>input->post('no_po');
        $id_supplier = $this-
>input->post('id_supplier');
        $id_bahan_baku = $this-
>input->post('id_bahan_baku');
        $qty_bahan_baku = $this-
>input->post('qty_bahan_baku');
        $tgl_surat_jalan = $this->input-
>post('tgl_surat_jalan');
        $validasi_tgl = $this-
>input->post('validasi_tgl');
        $validasi_surat_jalan = $this->input-
>post('validasi_surat_jalan');

        $data = array('id_surat_jalan' => $id, 'no_po' => $no_po, 'id_supplier' => $id_supplier, 'id_bahan_baku' => $id_bahan_baku, 'qty_bahan_baku' => $qty_bahan_baku, 'tgl_surat_jalan' => $tgl_surat_jalan, 'validasi_tgl' => $validasi_tgl, 'validasi_surat_jalan' => $validasi_surat_jalan);
        $simpan = $this->m_suratjalan->update($data, $id);
        $this->session->set_flashdata('message', '<div class="alert alert-success" role="alert"> Data Berhasil Diubah <button type="button" class="close" data-dismiss="alert" aria-label="Close"> <span aria-hidden="true">&times;</span> </button> </div>');
        redirect('suratjalan');
    }

    function construct()
    {
        parent::__construct();
    }

    public function getReport()
    {
        $data['judul'] = 'Detail Data Surat Jalan';
        $data['i_suratjalan'] = $this->m_suratjalan-
>isi()->row_array();
    }
}

```

```

        $data['record'] = $this->m_suratjalan-
>detail();
        $this->template->load('template',
'suratjalan/laporan', $data);
    }
}

```

## 2. Permintaan

### a. Model (m\_permintaan.php)

```

<?php

class m_permintaan extends CI_model {

    public function kode()
    {
        $q = $this->db->query("select
MAX(RIGHT(id_permintaan,3)) as code_max from
tb_permintaan");
        $code = "";
        if($q->num_rows()>0){
            foreach($q->result() as $cd){
                $tmp = ((int)$cd->code_max)+1;
                $code = sprintf("%03s", $tmp);
            }
        }else{
            $code = "001";
        }
        return "PM".$code;
    }

    function get_rak_bykode($kode){
        $hsl=$this->db->query("SELECT
tb_bahan_baku.id_bahan_baku, tb_bahan_baku.id_rak,
tb_rak.nama_rak, tb_bahan_baku.nama_bahan_baku
FROM `tb_bahan_baku`, tb_rak
WHERE tb_bahan_baku.id_rak = tb_rak.id_rak AND
tb_bahan_baku.id_bahan_baku = '$kode'");
        if($hsl->num_rows()>0){
            foreach ($hsl->result() as $data) {
                $hasil=array(
                    'id_rak'      => $data->id_rak,
                    'nama_rak'    => $data->nama_rak
                );
                return $hasil;
            }
        }
    }

    public function tampilkan_data()
    {
        return $this->db->query('SELECT * FROM
`tb_permintaan` ORDER BY `tb_permintaan`.`id_permintaan` DESC');
    }

    public function tampilkan_data_dtl()

```

```

{
    $a = $this->uri->segment('3');
    $hasil = $this->db->query("SELECT
tb_permintaan_detail.id_detail,
tb_permintaan_detail.id_permintaan,
tb_bahan_baku.nama_bahan_baku, tb_rak.nama_rak,
tb_permintaan_detail.qty
        FROM tb_permintaan_detail, tb_bahan_baku,
tb_rak
        WHERE tb_permintaan_detail.id_bahan_baku =
tb_bahan_baku.id_bahan_baku AND
tb_permintaan_detail.id_rak =
tb_rak.id_rak
        AND tb_permintaan_detail.id_permintaan =
'$a' order by tb_permintaan_detail.id_detail desc");
    return $hasil;
}

public function construct()
{
    $this->load->database();
    parent::__construct();
}

function got()
{
    $query = $this->db->query('SELECT * FROM
tb_bahan_baku');
    return $query->result();
}

public function simpan($data)
{
    $this->db->insert('tb_permintaan', $data);
}

public function simpan_detil($data)
{
    $this->db->insert('tb_permintaan_detail',
$data);
}

public function hapus($id)
{
    $this->db->delete('tb_permintaan',
['id_permintaan' => $id]);
    $this->db->delete('tb_permintaan_detail',
['id_permintaan' => $id]);
}

public function hapus_detail($id)
{
    $this->db->delete('tb_permintaan_detail',
['id_detail' => $id]);
}

```

```

public function get_data($id)
{
    $vans = array('id_permintaan' => $id);
    $this->db->select('*');
    $this->db->from('tb_permintaan');
    $this->db->join('tb_bahan_baku',
'tb_bahan_baku.id_bahan_baku =
tb_permintaan.id_bahan_baku');
    $this->db->join('tb_rak', 'tb_rak.id_rak =
tb_permintaan.id_rak');
    return $this->db->get_where('', $vans);
}

public function isi()
{
    $a = $this->uri->segment('3');
    // echo $a;
    // die();
    $hasil = $this->db->query("SELECT
tb_permintaan.id_permintaan, tb_permintaan.status,
tb_permintaan.tgl_permintaan
        FROM tb_permintaan
        WHERE tb_permintaan.id_permintaan =
'$a'");
    return $hasil;
}

public function detail()
{
    $a = $this->uri->segment('3');
    // echo $a;
    // die();
    $hasil = $this->db->query("SELECT
tb_permintaan.id_permintaan,tb_permintaan_detail.id_deta
il,tb_permintaan_detail.id_bahan_baku,tb_bahan_baku.satu
an_bahan_baku, tb_bahan_baku.nama_bahan_baku,
tb_rak.nama_rak, tb_permintaan_detail.qty
        FROM tb_permintaan,tb_permintaan_detail,
tb_bahan_baku, tb_rak
        WHERE tb_permintaan_detail.id_bahan_baku =
tb_bahan_baku.id_bahan_baku AND
                tb_permintaan.id_permintaan =
tb_permintaan_detail.id_permintaan AND
                tb_permintaan_detail.id_rak =
tb_rak.id_rak
                AND tb_permintaan.id_permintaan = '$a')");
    return $hasil;
}

function update($data, $id)
{
    $this->db->where('id_permintaan', $id);
    $this->db->update('tb_permintaan', $data);
}

function validasi($id)

```

```

{
    $this->db->query("UPDATE tb_permintaan SET status='1'
where id_permintaan='$id'");

    $bnyak = $this->db->query("SELECT
id_bahan_baku,qty FROM tb_permintaan_detail WHERE
id_permintaan = '$id'");
    foreach ($bnyak->result() as $bb) {
        // echo $bb->id_bahan_baku.'-'.$bb->qty.
        '<br>';
        $this->db->query("UPDATE tb_bahan_baku SET
stok_bahan_baku = stok_bahan_baku - ".$bb->qty. " WHERE
id_bahan_baku=". "'".$bb->id_bahan_baku'"');
        // $this->load->database();
    }
}

function update_stock_minus($id)
{

    $bnyak = $this->db->query("SELECT
id_bahan_baku,qty FROM tb_permintaan_detail WHERE
id_permintaan = '$id'");
    foreach ($bnyak->result() as $bs) {
        $id_bahan_baku= $bs->id_bahan_baku;
    }

    $bnyak = $this->db->query("SELECT * FROM
tb_bahan_baku WHERE id_bahan_baku = '$id_bahan_baku'");
    foreach ($bnyak->result() as $bb) {
        if ($bb->stok_bahan_baku < 0 ) {
            $this->db->query("UPDATE tb_bahan_baku SET
kekurangan =".$bb->stok_bahan_baku. " WHERE
id_bahan_baku=". "'".$bb->id_bahan_baku.'"');

            $this->db->query("UPDATE tb_bahan_baku SET
stok_bahan_baku = 0 WHERE id_bahan_baku=". "'".$bb-
>id_bahan_baku.'"");
        }
    }
}

```

### b. Controller (C\_Permintaan.php)

```

<?php

class permintaan extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model('m_permintaan');
        $this->load->library('form_validation');
    }

    public function index()
    {

```

```

        $data['code'] = $this->m_permintaan->kode();
        $data['record'] = $this->m_permintaan-
>tampilkan_data();
        $this->template-
>load('template','permintaan/lihat_data', $data);
    }

    public function index_detail()
    {
        $data['record'] = $this->m_permintaan-
>tampilkan_data();
        $data['id_bahan_baku'] = $this->m_permintaan-
>got();
        $this->template-
>load('template','permintaan/lihat_data', $data);
    }

    public function validasi($id)
    {
        $id = $this->uri->segment(3);
        $this->m_permintaan->validasi($id);
        $this->m_permintaan->update_stock_minus($id);
        $this->load->database();
        $this->session->set_flashdata('message', '<div
class="alert alert-warning" role="alert">
            Data Berhasil Disetujui <button
type="button" class="close" data-dismiss="alert" aria-
label="Close">
                <span aria-hidden="true">&times;</span>
            </button>
        </div>');
        redirect('permintaan');
    }

    public function tambah()
    {
        $data['code'] = $this->m_permintaan->kode();
        $this->template->load('template',
'permintaan/form_input', $data);
    }

    public function tambah_detail()
    {
        $data['code']           = $this->m_permintaan-
>kode();
        $data['bahan_baku'] = $this->m_permintaan-
>got();
        $this->template->load('template',
'permintaan/form_detail', $data);
    }

    public function detail($id)
    {
        $data['judul'] = 'Detail Data Permintaan';
        $data['permintaan'] = $this->m_permintaan-
>tampilkan_data($id);
    }

```

```

        $this->template->load('template',
'permintaan/detail', $data);
}

public function simpan()
{
    $id = $this-
>input->post('id_permintaan');
    $tgl_permintaan = $this-
>input->post('tgl_permintaan');

    $data = array('id_permintaan' => $id,
'tgl_permintaan' => $tgl_permintaan, 'status' => 0);
    $simpan = $this->m_permintaan->simpan($data);
    $this->session->set_flashdata('message', '<div
class="alert alert-success" role="alert">
        Data Berhasil Ditambahkan <button
        type="button" class="close" data-dismiss="alert" aria-
        label="Close">
            <span aria-hidden="true">&times;</span>
        </button>
    </div>');
    redirect('permintaan/form_detail/'.$id);
}

public function simpan_detil()
{
    $id_permintaan = $this-
>input->post('id_permintaan');
    $id_bahan_baku = $this-
>input->post('id_bahan_baku');
    $id_rak = $this->input->post('id_rak');
    $qty_bahan_baku = $this-
>input->post('qty');

    $data = array('id_permintaan' => $id_permintaan,
'id_bahan_baku' => $id_bahan_baku, 'id_rak' => $id_rak,
'qty' => $qty_bahan_baku);
    $simpan = $this->m_permintaan-
>simpan_detil($data);
    $this->session->set_flashdata('message', '<div
class="alert alert-success" role="alert">
        Data Berhasil Ditambahkan <button
        type="button" class="close" data-dismiss="alert" aria-
        label="Close">
            <span aria-hidden="true">&times;</span>
        </button>
    </div>');
    redirect($_SERVER['HTTP_REFERER']);
}

function form_detail(){
    $data['bahan_baku'] = $this->m_permintaan-
>got();
}

```

```

        $data['record'] = $this->m_permintaan-
>tampilkan_data_dtl();
        $this->template->load('template',
'permintaan/form_detail',$data);
    }

    function get_rak(){
        // $b    = $this->uri->segment(3);
        $a    = $this->uri->segment(3);
        $data = $this->m_permintaan->get_rak_bykode($a);
        echo json_encode($data);
    }

    public function hapus($id)
    {
        $id = $this->uri->segment(3);
        $this->m_permintaan->hapus($id);
        $this->session->set_flashdata('message', '<div
class="alert alert-danger" role="alert">
            Data Berhasil Dihapus <button
type="button" class="close" data-dismiss="alert" aria-
label="Close">
                <span aria-hidden="true">&times;</span>
                </button>
            </div>');
        redirect('permintaan');
    }

    public function hapus_detail($id)
    {
        $id = $this->uri->segment(3);
        $this->m_permintaan->hapus_detail($id);
        $this->session->set_flashdata('message', '<div
class="alert alert-danger" role="alert">
            Data Berhasil Dihapus <button
type="button" class="close" data-dismiss="alert" aria-
label="Close">
                <span aria-hidden="true">&times;</span>
                </button>
            </div>');
        redirect($_SERVER['HTTP_REFERER']);
    }
    function construct()
    {
        $this->load->database();
        parent::__construct();
    }

    public function getReport()
    {
        $data['judul'] = 'Detail Data Permintaan';
        $data['i_permintaan'] = $this->m_permintaan-
>isi()->row_array();
        $data['record'] = $this->m_permintaan-
>detail();
    }
}

```

```
        $this->template->load('template',
'permintaan/laporan', $data);
    }
}
```

## LAMPIRAN D TAMPILAN ANTAR MUKA



Gambar Tampilan *Login*

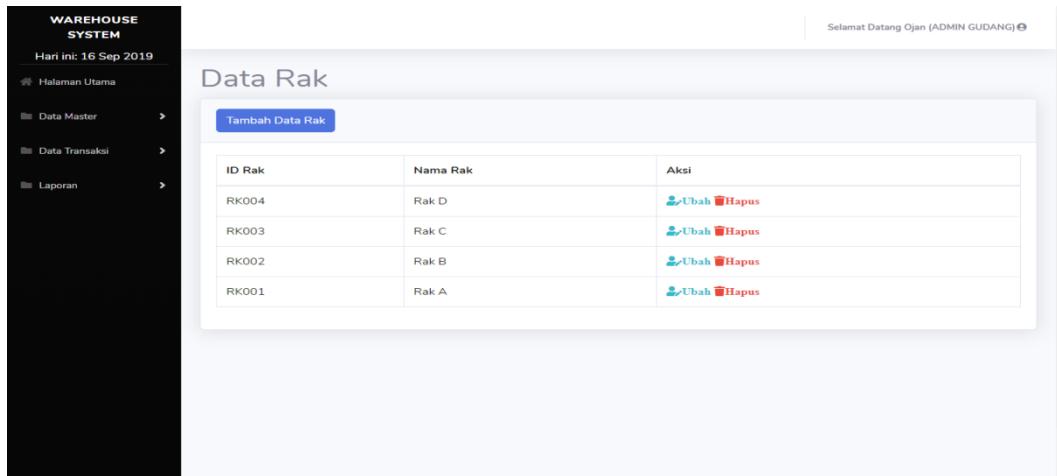


Gambar Tampilan Halaman Utama

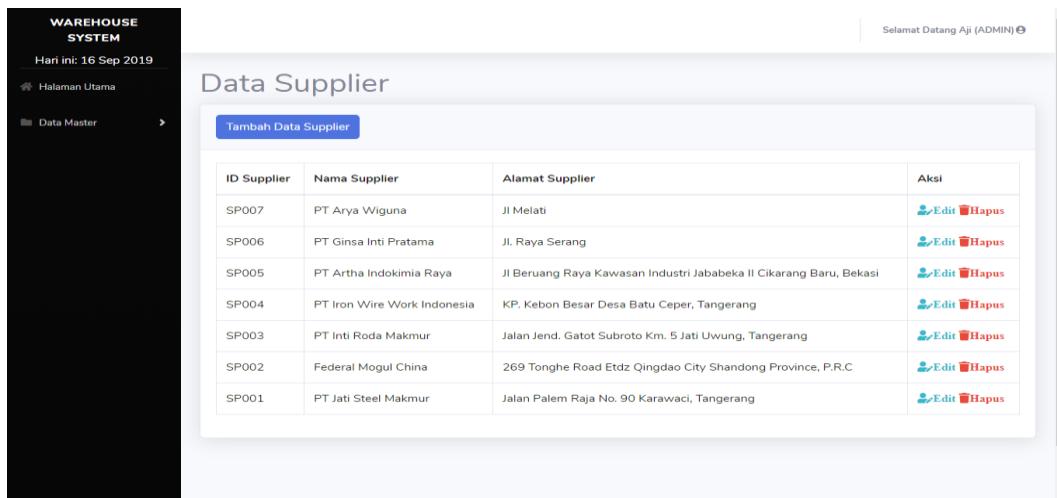
The screenshot shows a sub-menu for raw material data ('Data Bahan Baku'). The top bar includes the 'WAREHOUSE SYSTEM' header, the date 'Hari ini: 16 Sep 2019', and the welcome message 'Selamat Datang Ojan (ADMIN GUDANG)'. The main content area has a title 'Data Bahan Baku' and a sub-header 'Tambah Data Bahan Baku'. Below this is a table listing raw materials:

ID Bahan Baku	Nama Bahan Baku	Stok Bahan Baku	Satuan Bahan Baku	Nama Rak	Kekurangan	Aksi
BB010		0	kg	Rak A	0	
BB009	Terminal Nut	1000	pcs	Rak D	0	
BB008	Outside Gasket	1000	pcs	Rak D	0	
BB007	Inside Gasket	1000	pcs	Rak C	0	
BB006	Terminal Stud	1798	pcs	Rak C	0	
BB005	Resistor	1002	pcs	Rak B	0	
BB004	Spring	1000	pcs	Rak B	0	
BB003	Center Elektrode	1004	pcs	Rak B	0	
BB002	Insulator	1000	pcs	Rak A	0	
BB001	Nickel Wire	32	pcs	Rak A	0	

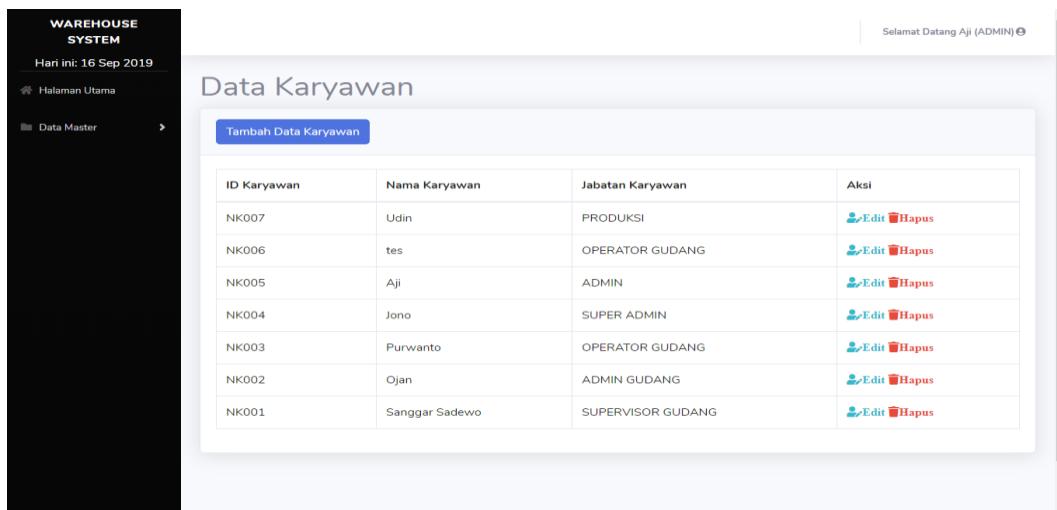
Gambat Tampilan Sub Menu Data Bahan Baku



Gambar Tampilan Sub Menu Data Rak



Gambar Tampilan Sub Menu Data Supplier



Gambar Tampilan Sub Menu Data Karyawan

**WAREHOUSE SYSTEM**  
Hari ini: 16 Sep 2019  
Selamat Datang Aji (ADMIN)

### Data Login

Tambah Data Login

ID Login	ID Karyawan	Password	Aksi
LG004	NK007	777	<a href="#">Edit</a> <a href="#">Hapus</a>
LG003	NK005	333	<a href="#">Edit</a> <a href="#">Hapus</a>
LG002	NK002	222	<a href="#">Edit</a> <a href="#">Hapus</a>
LG001	NK001	111	<a href="#">Edit</a> <a href="#">Hapus</a>

Gambar Tampilan Sub Menu Data Login

**WAREHOUSE SYSTEM**  
Hari ini: 16 Sep 2019  
Selamat Datang Udin (PRODUKSI)

### Data Permintaan

Tambah Data Permintaan

ID Permintaan	Tanggal Permintaan	Status	Aksi
PM006	2019-09-13	Disetujui	<a href="#">Q.Detail</a>
PM005	2019-09-10	Disetujui	<a href="#">Q.Detail</a>
PM004	2019-09-10	Disetujui	<a href="#">Q.Detail</a>
PM003	2019-09-08	Disetujui	<a href="#">Q.Detail</a>
PM002	2019-09-06	Disetujui	<a href="#">Q.Detail</a>
PM001	2019-09-05	Disetujui	<a href="#">Q.Detail</a>

Gambar Tampilan Sub Menu Data Permintaan

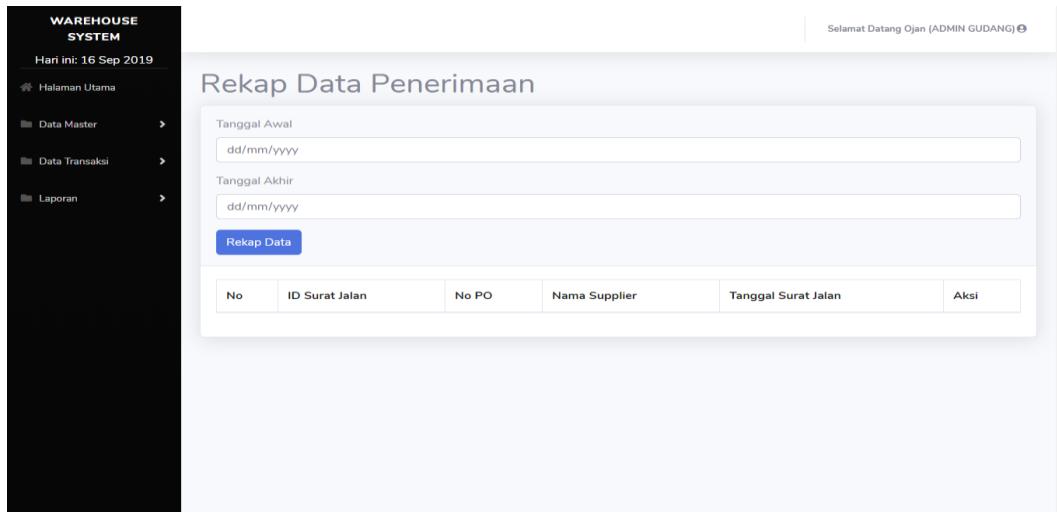
**WAREHOUSE SYSTEM**  
Hari ini: 16 Sep 2019  
Selamat Datang Ojan (ADMIN GUDANG)

### Data Surat Jalan

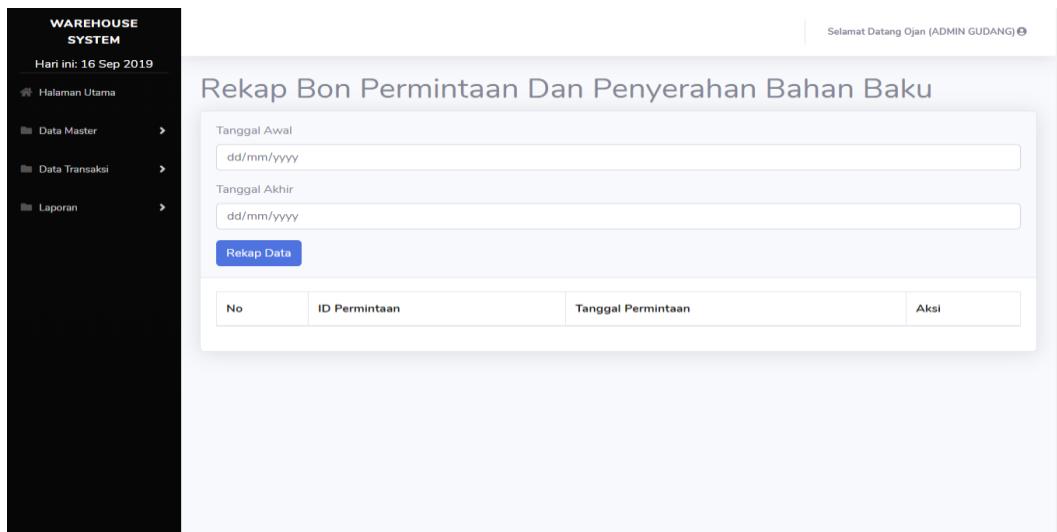
Tambah Data Surat Jalan

ID Surat Jalan	No PO	Nama Supplier	Tanggal Surat Jalan	Status	Aksi
SJ007	PO007	PT Jati Steel Makmur	2019-09-13	Disetujui	<a href="#">Q.Detail</a>
SJ006	PO006	Federal Mogul China	2019-09-13	Disetujui	<a href="#">Q.Detail</a>
SJ005	PO005	PT Jati Steel Makmur	2019-09-10	Disetujui	<a href="#">Q.Detail</a>
SJ003	PO003	PT Inti Roda Makmur	2019-09-06	Disetujui	<a href="#">Q.Detail</a>
SJ002	PO002	PT Arya Wiguna	2019-09-03	Disetujui	<a href="#">Q.Detail</a>
SJ001	PO001	PT Ginsa Inti Pratama	2019-09-02	Disetujui	<a href="#">Q.Detail</a>

Gambar Tampilan Sub Menu Data Surat Jalan



Gambar Tampilan Sub Menu Laporan Data Penerimaan



Gambar Tampilan Menu Data Laporan Data Pengeluaran

## **LAMPIRAN**