

**RANCANG BANGUN SISTEM INFORMASI
PENERIMAAN DAN PENGIRIMAN KOMPONEN
MENGUNAKAN *JAVA J2SE* DAN *MYSQL 5.1.37*
PADA PT KRAMA YUDHA RATU MOTOR**

TUGAS AKHIR

Diajukan Untuk Memenuhi Syarat Program Sarjana Terapan
Program Studi Sistem Informasi Industri Otomotif Politeknik STMI Jakarta

**DISUSUN OLEH
RAHMEILIA KHAIRUNNISA
1314063**



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

POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

LEMBAR PENGESAHAN

Judul Tugas Akhir : **RANCANG BANGUN SISTEM INFORMASI
PENERIMAAN DAN PENGIRIMAN KOMPONEN
MENGUNAKAN JAVA J2SE DAN MYSQL 5.1.37
PADA PT KRAMA YUDHA RATU MOTOR**

Disusun Oleh :
Nama : Rahmeilia Khairunnisa
Nim : 1314063
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 Rabu tanggal 19 September 2018.

Jakarta, 20 September 2018

Dosen Pembimbing

Ketua Penguji

Noveriza Yuliasari, S.Si, MT
NIP. 197811212009012003

Ridzky Kramanandita, S.Kom, MT
NIP. 197403022002121001

Dosen Penguji

Dosen Penguji

Lucky Heriyanto, MTI
NIP. 197908202009011009

Lucyana Tresia, MT
NIP. 197803012008032001

POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

LEMBAR PERSETUJUAN DOSEN PEMBIMBING

Judul Tugas Akhir : **RANCANG BANGUN SISTEM INFORMASI
PENERIMAAN DAN PENGIRIMAN KOMPONEN
MENGUNAKAN JAVA J2SE DAN MYSQL 5.1.37
PADA PT KRAMA YUDHA RATU MOTOR**

Disusun Oleh :
Nama : Rahmeilia Khairunnisa
Nim : 1314063
Program Studi : Sistem Informasi Industri Otomotif
Tanggal Seminar : 03 September 2018
Tanggal Sidang : 19 September 2018
Tanggal Lulus : 19 September 2018

Jakarta, 20 September 2018

Menyetujui

Dosen Pembimbing

Noveriza Yuliasari, S.Si, MT
NIP. 197811212009012003

**POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA**

TANDA PERSETUJUAN DOSEN PEMBIMBING

Judul Tugas Akhir : **RANCANG BANGUN SISTEM INFORMASI
PENERIMAAN DAN PENGIRIMAN KOMPONEN
MENGUNAKAN JAVA J2SE DAN MYSQL 5.1.37 PADA PT
KRAMA YUDHA RATU MOTOR**

Disusun Oleh

Nama : Rahmeilia Khairunnisa
Nim : 1314063
Program Studi : Sistem Informasi Industri Otomotif

Jakarta, 08 Agustus 2018

Menyetujui
Dosen Pembimbing

Noveriza Yuliasari, S.Si, MT
NIP. 197811212009012003

POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : Rahmeilia Khairunnisa

Nim : 1314063

Program Studi : Sistem Informasi Industri Otomotif

Dengan ini menyatakan bahwa karya Tugas Akhir yang saya buat dengan judul:

“RANCANG BANGUN SISTEM INFORMASI PENERIMAAN DAN PENGIRIMAN KOMPONEN MENGGUNAKAN JAVA J2SE DAN MYSQL 5.1.37 PADA PT KRAMA YUDHA RATU MOTOR”. Merupakan dibuat dan diselesaikan sendiri, dengan menggunakan literatur hasil kuliah, survei lapangan, dosen pembimbing, melalui tanya jawab maupun asistensi serta buku-buku acuan yang tertera dalam referensi pada karya Tugas Akhir ini.

Jika terbukti saya tidak memenuhi apa yang telah dinyatakan di atas, maka karya Tugas Akhir saya dibatalkan.

Jakarta, 08 Agustus 2018

Yang Membuat Pernyataan,

Rahmeilia Khairunnisa



LEMBAR BIMBINGAN PENYUSUNAN TUGAS AKHIR

Nama : Rahmeilia Khairunnisa

NIM : 1314063

Judul TA : Rancang Bangun Sistem Informasi Penerimaan Dan Pengiriman Komponen Menggunakan Java J2SE Dan Mysql 5.1.37 Pada PT Krama Yudha Ratu Motor

Pembimbing : Noveriza Yuliasari, S.Si, MT

Tanggal	BAB	Keterangan	Paraf
24/04/2018	Proposal	Acc Proposal	
02/05/2018	BAB I	Bimbingan BAB I	
04/05/2018	BAB I	Revisi BAB I	
16/05/2018	BAB I, II	Revisi BAB I, Bimbingan BAB II, Daftar	
18/05/2018	BAB	Pustaka	
25/05/2018	I,II,III	Revisi BAB I, BAB II, Bimbingan BAB III	
10/07/2018	BAB III,	Revisi BAB III, Bimbingan BAB IV	
13/07/2018	IV	Bimbingan BAB V <i>Flowmap</i>	
24/07/2018	BAB V	Revisi <i>Flowmap</i> , Bimbingan BAB V UML	
27/07/2018	BAB V	Revisi UML	
01/08/2018	BAB V	Bimbingan ERD, CDM, Kamus Data	
03/08/2018	BAB V	Revisi ERD, CDM, Bimbingan WND, BAB	
06/08/2018	BAB V, VI	VI	
07/08/2018	BAB V	Revisi ERD, CDM, Bimbingan Rancangan	
13/08/2018	BAB V	<i>Interface</i>	
	BAB V, VI	Revisi ERD, CDM	
	BAB I, II,	Revisi ERD, CDM, BAB VI	
	III, IV, V,	<i>Review All</i> dan Demo Program	
	VI		

Mengetahui,
Ketua Program Studi
Pembimbing
Sistem Informasi Industri Otomotif

Drs. Jacob Saragih, MM
Yuliasari, S.Si, MT
NIP: 195404281986031002
197811212009012003

Noveriza

NIP :

ABSTRAK

Sistem informasi merupakan hal yang penting dalam sebuah perusahaan. Dengan adanya sistem informasi, perusahaan dapat menjamin kualitas informasi, yaitu informasi dapat disajikan dengan baik, cepat dan mudah dipahami. PT. Krama Yudha Ratu Motor (PT. KRM) merupakan sebuah perusahaan manufaktur yang bergerak di bidang otomotif yang memiliki spesialisasi merakit berbagai macam kendaraan niaga Mitsubishi di Indonesia. Salah satu divisi yang dimiliki PT KRM adalah divisi *Part Control* yang bertugas melakukan penerimaan dan pengiriman komponen produksi. Untuk menghasilkan produk sesuai dengan target produksi, membutuhkan suatu sistem informasi penerimaan dan pengiriman komponen agar pemenuhan kebutuhan komponen untuk produksi dapat terpenuhi dengan baik sesuai dengan rencana yang telah ditentukan. Sistem Informasi yang diusulkan akan mempermudah dalam melakukan pengolahan data komponen antar bagian *Part Control* dan Produksi sehingga lebih cepat dan mudah. Pengembangan sistem informasi penerimaan dan pengiriman komponen pada PT KRM menggunakan model *prototype evolutioner*. Analisis dan perancangan yang digunakan adalah analisis dan perancangan berorientasi objek. Pemodelan Sistem dengan *Unified Modeling Language* (UML), pemodelan data dengan *ERD* dan Kamus Data, desain sistem menggunakan *windows navigation diagram* dan desain *interface*. Perancangan dan pembuatan sistem informasi penerimaan dan pengiriman komponen ini menggunakan bahasa pemrograman *Java J2SE* dan perangkat basis data *MySQL*. Sistem informasi penerimaan dan pengiriman komponen ini dirancang agar dapat menunjang proses penerimaan dan pengiriman komponen serta mempersingkat waktu dalam pembuatan laporan penerimaan dan pengiriman komponen yang sebelumnya sekitar 1 jam, dengan adanya sistem informasi penerimaan dan pengiriman komponen dapat dihasilkan dalam waktu 30 menit. Untuk penerapan sistem yang baru, disarankan untuk melakukan sosialisasi kepada bagian terkait dan pemeliharaan aplikasi agar dapat berjalan dengan baik.

Kata kunci : Sistem Informasi, Penerimaan, Pengiriman, Komponen, *Prototype Evolutioner*, *Unified Modelling Language* (UML), *Java J2SE*, *Netbeans* 8.2, *MySQL*.

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul **“RANCANG BANGUN SISTEM INFORMASI PENERIMAAN DAN PENGIRIMAN KOMPONEN MENGGUNAKAN JAVA J2SE DAN MYSQL 5.1.37 PADA PT KRAMA YUDHA RATU MOTOR”** berjalan baik dan benar.

Penulisan Tugas Akhir ini bertujuan untuk memenuhi syarat penyelesaian program sarjana terapan pada program studi Sistem Informasi Industri Otomotif pada Politeknik STMI Jakarta.

Terselesaikannya tugas akhir ini tidak terlepas dari bantuan moril maupun materil, bantuan, bimbingan, dukungan serta cinta dari berbagai pihak. Maka dengan segala ketulusan dan kerendahan hati, penulis menyampaikan ucapan terimakasih kepada:

1. Allah SWT yang telah memberikan segala sesuatu serta kelancaran dan kemudahan dalam menyelesaikan Tugas Akhir ini.
2. Ayah dan Mamak di Medan yang selalu memberikan dukungan, bantuan, cinta dan do'a serta mendorong penulis untuk segera menyelesaikan Tugas Akhir ini.
3. Bapak Drs. Jacob Saragih, MM selaku Ketua Program Studi Sistem Informasi Industri Otomotif Politeknik STMI Jakarta.
4. Ibu Noveriza Yuliasari, S.Si., M.T selaku Dosen Pembimbing Tugas Akhir yang telah bersedia membantu memberikan arahan, bimbingan serta penjelasan dalam penyusunan Tugas Akhir.
5. Seluruh dosen yang terlibat dalam proses kegiatan belajar mengajar dan secara tidak langsung membantu penyelesaian laporan kerja lapangan ini.
6. PPJ Ishmah Afiifatul Afnan, Erlin Puspita Dewi, Felly Hedriyani Laili dan Nax Caem Ivana Andira Wiraputri, Nurul Sukma Dewi, Nurul Istiqomah, Maya Lestari yang telah membantu, menghibur, mensupport dan selalu ada buat penulis serta selalu menyemangati untuk menyelesaikan Tugas Akhir ini.

7. Teman-teman seperjuangan mahasiswa/i Politeknik STMI Jakarta Program Studi Sistem Informasi Industri Otomotif SA02 dan SA01 2014 atas kebersamaan dan motivasinya selama ini.
8. Bapak Suryadi selaku pembimbing Praktik Kerja Lapangan, Bapak Nugroho, Bapak Rahmat, Bapak Anjar, Bapak Fauzi, Bapak Tulus, Bapak Fajar, Bapak Ahyar, Bu Sheena serta seluruh karyawan PT Krama Yudha Ratu Motor yang telah memberi arahan serta informasi untuk pengumpulan data kepada penulis sehingga penulis bisa menyelesaikan Laporan Praktik Kerja Lapangan ini.
9. Serta semua pihak yang tidak disebutkan satu persatu yang telah memberikan kritik, saran dan bantuan dalam penyusunan laporan ini.

Semoga segala bantuan yang telah diberikan kepada penulis mendapat balasan yang setimpal dari Allah SWT. Penulis menyadari masih banyak kekurangan dalam penyusunan Tugas Akhir ini, untuk itu penulis mohon kritik dan saran yang bersifat membangun demi kesempurnaan penulisan di masa yang akan datang. Semoga laporan ini dapat memberikan manfaat dan informasi yang berguna bagi kami dan para pembaca.

Jakarta, 13 Agustus 2018

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	
LEMBAR PENGESAHAN	
LEMBAR PERSETUJUAN DOSEN PEMBIMBING	
LEMBAR BIMBINGAN DENGAN DOSEN PEMBIMBING	
LEMBAR PERNYATAAN KEASLIAN	
ABSTRAK	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR GAMBAR	xii
DAFTAR TABEL	xv
 BAB I PENDAHULUAN	 1
1.1 Latar Belakang	1
1.2 Pokok Permasalahan	2
1.3 Tujuan Penelitian	3
1.4 Batasan Masalah.....	3
1.5 Manfaat Tugas Akhir	4
1.6 Sistematika Penulisan.....	4
 BAB II LANDASAN TEORI	 6
2. 1 Pengertian Rancang Bangun	6
2. 2 Konsep Dasar Sistem Informasi	6
2.2.1 Pengertian Sistem	7
2.2.2 Pengertian Informasi	8
2.2.3 Pengertian Sistem Informasi	9
2.2.4 Komponen Sistem Informasi	10
2. 3 Pengertian Sistem Informasi Penerimaan dan Pengiriman Bahan Baku atau Komponen.....	13
2.3.1 Pengertian Penerimaan	14

2.3.2	Pengertian Pengiriman	15
2.3.3	Pengertian Bahan Baku atau Komponen	15
2. 4	Pengembangan Sistem	16
2.4.1	Model Prototipe Secara Umum	19
2.4.2	Evolutionary Prototype	20
2. 6	Unified Modelling Language (UML).....	22
2. 5	Flowchart.....	24
2.6.1	Use Case Diagram	26
2.6.2	Activity Diagram	27
2.6.3	Sequence Diagram	29
2.6.4	Class Diagram.....	31
2.6.5	Deployment Diagram.....	32
2. 7	Entity Relationship Diagram (ERD)	33
2. 8	Conceptual Data Model (CDM).....	34
2. 9	Kamus Data	37
2. 10	Windows Navigation Diagram (WND)	38
2. 11	Database	39
2. 12	Java.....	39
2. 13	Netbeans	41
2. 14	MySQL.....	43
2.6.4	Keunggulan MySQL	44
2.6.4	Tipe Data MySQL.....	45
2. 14	XAMPP	46
BAB III	METODOLOGI PENELITIAN.....	47
3.1	Metodologi Penelitian	47
3.2	Jenis dan Sumber Data	47
3.3	Metode Pengumpulan Data	48
3.4	Metode Pendekatan dan Pengembangan Sistem	49
3.4.1	Metode Pendekatan Sistem.....	49
3.4.2	Metode Pengembangan Sistem.....	50
3.5	Kerangka Penelitian	50

BAB IV	PENGUMPULAN DAN PENGOLAHAN DATA	54
4. 1	Sekilas Perusahaan	54
4. 2	Profil Perusahaan	55
4. 3	Visi dan Misi Perusahaan	56
4. 4	Struktur organisasi Part Control PT. KRM	57
4. 5	Produk PT. KRM	60
4. 6	Komponen	61
4. 7	Vendor Perusahaan	61
4. 8	Man Power Part Control PT KRM	62
4. 9	Alat Pendukung Produksi	63
4. 10	Dokumen Penerimaan Komponen	69
4. 11	Prosedur Penerimaan Komponen	69
BAB V	ANALISIS DAN PEMBAHASAN	72
5. 1	Analisis Kebutuhan Functional Sistem	72
5. 2	Prosedur Sistem Informasi Penerimaan dan Pengiriman Komponen	73
5. 3	Pemodelan Sistem Informasi Penerimaan dan Pengiriman Komponen	76
5.3.1	<i>Use case diagram</i>	76
5.3.2	<i>Activity Diagram</i>	84
5.3.3	<i>Sequence Diagram</i>	92
5.3.4	<i>Class Diagram</i>	99
5.3.5	<i>Deployment Diagram</i>	99
5. 4	Pemodelan Data	100
5.4.1.	<i>Entity Relationship Diagram (ERD)</i>	100
5.4.2.	<i>Conceptual Data Model (CDM)</i>	101
5. 5	Kamus Data	102
5. 6	Analisis Desain Program	106
5.6.1	<i>WND (Windows Navigation Diagram)</i>	106
5.6.2	Perancangan <i>Interface</i> Sistem	107
5. 7	Implementasi Sistem <i>Software</i> dan <i>Hardware</i>	118

BAB VI PENUTUP	119
6. 1 Kesimpulan	119
6. 2 Saran	119
DAFTAR PUSTAKA	120
LAMPIRAN	

DAFTAR GAMBAR

	Halaman
Gambar II.1 Sistem Dengan Tiga Aspek, Prosedur, Dan Pengguna.....	7
Gambar II.2 Pilar Kualitas Informasi	8
Gambar II.3 Komponen-Komponen Sistem Informasi	13
Gambar II.4 Ilustrasi Model Prototipe	19
Gambar II.5 <i>Evolutionary Prototype Model</i>	21
Gambar II.6 Contoh Windows Navigation Diagram	38
Gambar III.1 Kerangka Penelitian.....	53
Gambar IV.1 Logo Sertifikat Yang Diperoleh	55
Gambar IV.2 Logo Pt Krama Yudha Ratu Motor	56
Gambar IV.3 Pabrik Pt Krama Yudha Ratu Motor	56
Gambar IV.4 Struktur Organisasi Bagian Pc Pt. Krm	58
Gambar IV.5 Produk Pt. Krm	60
Gambar IV.6 Kendaraan <i>Towing</i>	63
Gambar IV.7 Alat Angkut <i>Dolly</i>	64
Gambar IV.8 <i>Box Kitting</i>	64
Gambar IV.9 Rencana Produksi Harian	65
Gambar IV.10 <i>Order List</i>	66
Gambar IV.11 <i>Delivery Note</i>	66
Gambar IV.12 Dokumen Permohonan Surat Jalan.....	67
Gambar IV.13 Surat Pengiriman Komponen.....	68
Gambar IV.14 Surat Permintaan <i>Advance Part</i>	68
Gambar IV.15 Laporan Harian <i>Part Control</i>	69
Gambar IV.16 Aliran Dokumen Sistem Informasi Penerimaan Komponen	71
Gambar V.1 <i>Flowmap</i> Sistem Informasi Penerimaan Dan Pengiriman Komponen Usulan	75
Gambar V.2 <i>Use Case Diagram</i> Sistem Informasi Penerimaan Dan Pengiriman Komponen Usulan.....	87
Gambar V.3 Activity Diagram Login.....	84

Gambar V.4	<i>Activity Diagram</i> Mengelola Data Master	85
Gambar V.5	<i>Activity Diagram</i> Mengecek Stok Komponen	86
Gambar V.6	<i>Activity Diagram</i> Membuat Permintaan Advance Part	87
Gambar V.7	<i>Activity Diagram</i> Menginput Komponen Masuk	88
Gambar V.8	<i>Activity Diagram</i> Membuat Permohonan Surat Jalan Pengiriman Suplemen	89
Gambar V.9	<i>Activity Diagram</i> Membuat Surat Jalan Pengiriman Suplemen	90
Gambar V.10	<i>Activity Diagram</i> View Surat Jalan	91
Gambar V.11	<i>Activity Diagram</i> Mencetak Laporan	91
Gambar V.12	<i>Sequence Diagram</i> Login	92
Gambar V.13	<i>Sequence Diagram</i> Mengelola Data Master	93
Gambar V.14	<i>Sequence Diagram</i> Mengecek Stok Komponen	94
Gambar V.15	<i>Sequence Diagram</i> Membuat Permintaan Advance Part	94
Gambar V.16	<i>Sequence Diagram</i> Menginput Komponen Masuk.....	95
Gambar V.17	<i>Sequence Diagram</i> Membuat Permohonan Surat Jalan Pengiriman.....	96
Gambar V.18	<i>Sequence Diagram</i> Membuat Permohonan Surat Jalan Pengiriman Suplemen	97
Gambar V.19	<i>Sequence Diagram</i> View Surat Jalan	98
Gambar V.20	<i>Sequence Diagram</i> Mencetak Laporan Komponen Masuk	98
Gambar V.21	<i>Class Diagram</i> Sistem Informasi Penerimaan Dan Pengiriman Komponen Usulan	99
Gambar V.22	<i>Deployment Diagram</i> Sistem Informasi Penerimaan Dan Pengiriman Komponen Usulan.....	100
Gambar V.23	ERD Konseptual Sistem Informasi Penerimaan Dan Pengiriman Komponen Usulan	101
Gambar V.24	CDM Sistem Informasi Penerimaan Dan Pengiriman Komponen Usulan	102
Gambar V.25	WND Sistem Informasi Penerimaan Dan Pengiriman Komponen	107
Gambar V.26	<i>Interface Login</i>	108

Gambar V.27	<i>Interface Login</i>	108
Gambar V.28	<i>Interface Form Data Master Komponen</i>	109
Gambar V.29	<i>Interface Form Data Master Vendor</i>	110
Gambar V.30	<i>Interface Form Data Master User</i>	111
Gambar V.31	<i>Interface Form Mengecek Stok Komponen</i>	112
Gambar V.32	<i>Interface Form Advance Part</i>	113
Gambar V.33	<i>Interface Form Komponen Masuk</i>	114
Gambar V.34	<i>Interface Form Permohonan Surat Jalan</i>	115
Gambar V.35	<i>Interface Form Surat Jalan</i>	116
Gambar V.36	<i>Interface Form Laporan Penerimaan Komponen</i>	117
Gambar V.37	<i>Interface Form Laporan Pengiriman Komponen</i>	117

DAFTAR TABEL

	Halaman
Tabel II.1 Simbol <i>Flow Direction</i>	23
Tabel II.2 Simbol Proses.....	23
Tabel II.3 Simbol <i>Input Dan Output</i>	24
Tabel II.4 Jenis Diagram Pada Uml Dan Fungsi-Fungsinya	25
Tabel II.5 Simbol-Simbol <i>Use Case Diagram</i>	26
Tabel II.6 Simbol-Simbol <i>Activity Diagram</i>	28
Tabel II.7 Simbol-Simbol <i>Sequence Diagram</i>	30
Tabel II.8 Simbol-Simbol <i>Class Diagram</i>	32
Tabel II.9 Simbol-Simbol <i>Deployment Diagram</i>	33
Tabel II.10 Simbol-Simbol <i>Entity Relationship Diagram</i> (Erd).....	34
Tabel II.11 Simbol-Simbol <i>Conceptual Data Model</i> (Cdm)	35
Tabel II.12 Simbol-Simbol Aturan <i>Conceptual Data Model</i> (Cdm)	35
Tabel II.13 Contoh Kamus Data Untuk Tabel Pemasok.....	37
Tabel II.14 Jenis Data Pada <i>Mysql</i>	45
Tabel IV.1 Data <i>Vendor Pt. Krm</i>	62
Tabel IV.2 <i>Man Power Pc Pt. Krm</i>	62
Tabel V.1 Kebutuhan <i>Functional Sistem</i>	72
Tabel V.2 Definisi Aktor <i>Use Case Diagram</i> Sistem Usulan.....	76
Tabel V.3 Definisi <i>Use Case Diagram</i> Sistem Usulan	78
Tabel V.4 Definisi <i>Use Case Login</i>	78
Tabel V.5 Definisi <i>Use Case</i> Mengelola Data <i>Master</i>	79
Tabel V.6 Definisi <i>Use Case</i> Mengecek Stok Komponen	80
Tabel V.7 Definisi <i>Use Case</i> Membuat <i>Advance Part</i>	80
Tabel V.8 Definisi <i>Use Case</i> Menginput Komponen Masuk	81
Tabel V.9 Definisi <i>Use Case</i> Membuat Permohonan Surat Jalan Pengiriman Suplemen.....	82
Tabel V.10 Definisi <i>Use Case</i> Membuat Surat Jalan Pengiriman Suplemen	82
Tabel V.11 Definisi <i>Use Case View</i> Surat Jalan Pengiriman Suplemen.....	83

Tabel V.12	Definisi <i>Use Case</i> Mencetak Laporan Penerimaan Komponen ..	83
Tabel V.13	Tabel Pengguna	103
Tabel V.14	Tabel Komponen	103
Tabel V.15	Tabel <i>Vendor</i>	103
Tabel V.16	Tabel Komponen Masuk	104
Tabel V.17	Tabel Detail Komponen Masuk	104
Tabel V.18	Tabel Permohonan Surat Jalan Pengiriman Suplemen	105
Tabel V.19	Tabel Detail Permohonan.....	105
Tabel V.20	Tabel Surat Jalan	106

BAB I

PENDAHULUAN

1.7 Latar Belakang

Seiring dengan semakin ketatnya persaingan dalam dunia bisnis, baik industri jasa maupun manufaktur, keberadaan pengolahan data menjadi informasi secara terkomputerisasi menjadi sangat penting. Hal ini dikarenakan pengolahan data secara terkomputerisasi dapat memberikan kontribusi yang besar bagi kinerja suatu perusahaan untuk meningkatkan kualitas dan kuantitas produk yang dihasilkan. Jika dibandingkan pengolahan data secara manual, pengolahan data secara terkomputerisasi memiliki kelebihan, seperti pengolahan data yang cepat, akurat, serta mendukung pengolahan data dalam skala besar.

Pengolahan data secara terkomputerisasi berupa sistem informasi sangat diperlukan bagi setiap perusahaan yang berada dalam lingkup industri untuk membuat suatu kemajuan dalam pelaksanaan produksi di perusahaan tersebut. Sistem informasi menjadikan setiap perusahaan lebih efektif dan efisien pada saat pemberian data kepada pekerja yang terlibat di dalam perusahaan sebagai sarana pengambilan keputusan. Selain itu, sistem informasi juga dapat diterapkan atau dipakai dalam bagian-bagian yang terdapat dalam suatu perusahaan, seperti pengadaan barang, penerimaan komponen, administrasi, penjadwalan, produksi, dan pengiriman. Salah satu kegiatan yang menggunakan sistem informasi di dalam perusahaan yaitu kegiatan penerimaan dan pengiriman komponen produksi. Pengelolaan data penerimaan dan pengiriman komponen yang terhambat atau tidak berjalan dengan optimal dapat menyebabkan kesalahan dalam pencatatan jumlah komponen masuk dan keluar serta ketersediaan stok komponen yang mempengaruhi kegiatan produksinya. Pengaruh tersebut yaitu produksi terhambat atau justru tidak dapat dilaksanakan. Hal ini disebabkan karena dalam proses produksi memerlukan komponen untuk menghasilkan suatu produk.

Indonesia merupakan negara dengan jumlah industri yang besar, salah satunya adalah Industri Manufaktur Otomotif. PT Krama Yudha Ratu Motor (PT KRM) merupakan perusahaan industri yang bergerak dibidang Industri Manufaktur Otomotif

khususnya produksi kendaraan roda empat (mobil). Untuk menciptakan produk yang berkualitas tinggi dibutuhkan komponen yang baik serta pengendalian komponen yang terstruktur dengan sistematis. PT KRM bekerjasama dengan beberapa *vendor* untuk mendapatkan komponen tersebut.

Pada saat ini sistem informasi pada bagian *part control* PT Krama Yudha Ratu Motor masih dilakukan secara manual dan semi komputerisasi. Dalam pembuatan laporan komponen masuk, admin harus menunggu surat jalan atau DN dari bagian *part control*. *Part control* seringkali tidak langsung memberikan DN kepada admin dan menunggu DN sampai terkumpul terlebih dahulu kemudian memberikannya kepada admin, sehingga dalam pembuatan laporan penerimaan komponen, admin membutuhkan waktu yang cukup lama yang seharusnya beberapa menit menjadi 1 jam karena admin kesulitan dalam melihat dan menginput data yang terdiri dari 8 *sheet* dalam 1 *file* dan terkadang salah mengisi baris pada *excel* karena dalam 1 *sheet* terdapat 60-200 baris. Sedangkan pembuatan surat pengiriman komponen ke bagian produksi masih manual yaitu admin harus mengisi manual pada lembar surat jalan pengiriman komponen. Selain itu, kurangnya tempat penyimpanan dokumen yang hanya memiliki 1 meja untuk 16-20 lembar dokumen setiap harinya membuat para pekerja di bagian *Part Control* sering menyimpan dokumen di lokasi yang tidak semestinya sehingga sering terjadi kesulitan mencari dokumen apabila dibutuhkan dan juga beresiko terjadi kehilangan dokumen yang disimpan.

Untuk mengatasi masalah tersebut, diperlukan aplikasi untuk membantu perusahaan dalam mengatasi masalah tersebut. Adapun judul Tugas Akhir ini adalah “Rancang Bangun Sistem Informasi Penerimaan dan Pengiriman Komponen Menggunakan Java Netbeans 8.2 dan MySQL 5.1.37 pada PT Krama Yudha Ratu Motor”.

1.8 Pokok Permasalahan

Permasalahan yang terjadi pada PT Krama Yudha Ratu Motor adalah sebagai berikut:

1. Pengisian *form* penerimaan komponen dan laporan harian dilakukan dengan *input* melalui *Microsoft Excel* dimana admin seringkali salah ketik karena kesulitan dalam melihat data pada *sheet* dan dokumen yang berbeda.

2. Pembuatan surat jalan pengiriman komponen masih manual yaitu dengan mengisi manual lembar pengiriman komponen.
3. Admin menyimpan dokumen di sembarang tempat yang disebabkan kurangnya tempat penyimpanan dokumen sehingga berisiko terjadinya kehilangan data, sulit mencari data dan kerusakan data.

1.9 Tujuan Penelitian

Adapun tujuan dari penelitian Tugas Akhir ini adalah merancang dan membangun sistem informasi penerimaan komponen yang dapat:

1. Membantu bagian *Part Control* PT Krama Yudha Ratu Motor dalam melihat data komponen produksi serta mempermudah *Part Control* untuk membuat surat jalan pengiriman komponen serta laporan penerimaan dan pengiriman komponen lebih cepat dan sederhana dengan menggunakan aplikasi yang sudah terintegrasi.
2. Menyediakan fasilitas penyimpanan data dokumen-dokumen komponen dengan menggunakan *database* agar dapat disimpan di satu lokasi dan mencegah terjadinya kehilangan data, kesulitan mencari data dan kerusakan data.

1.10 Batasan Masalah

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

1. Penelitian dilakukan di PT Krama Yudha Ratu Motor selama 1 bulan mulai dari tanggal 05 Juli 2017 sampai tanggal 31 Juli 2017.
2. Penelitian hanya sebatas mengenai pengolahan data pada proses pengiriman suplemen komponen dan penerimaan komponen dari *vendor*, tidak mencakup pembelian dan pengembalian komponen cacat.

1.11 Manfaat Tugas Akhir

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

1. Bagi penulis

Sebagai alat untuk mempraktekkan teori-teori yang telah diperoleh selama perkuliahan, sehingga penulis dapat menambah pengetahuan secara praktis tentang

masalah-masalah yang dihadapi di dunia kerja dan yang terjadi pada perusahaan, khususnya perusahaan manufaktur.

2. Bagi perusahaan

Hasil penelitian ini agar dapat diimplementasikan di perusahaan untuk membantu kinerja perusahaan, karena dengan adanya sistem informasi penerimaan komponen dengan menggunakan *Java Netbeans* 8.2 dan *MySQL* 5.1.37. ini dapat membantu perusahaan untuk membuat laporan penerimaan dan pengiriman komponen.

3. Bagi pihak lain

Hasil penelitian ini diharapkan dapat memberikan sumbangan pemikiran dan sebagai referensi bagi peneliti lain yang melakukan penelitian yang merupakan pengembangan penelitian ini.

1.12 Sistematika Penulisan

Tugas Akhir ini disusun secara sistematis guna memberikan gambaran yang jelas mengenai isi dan pembahasan yang ada di dalamnya. Adapun sistematika penulisan Tugas Akhir ini diuraikan dalam enam bab, yaitu:

BAB I PENDAHULUAN

Pada bab ini memuat latar belakang, pokok permasalahan, tujuan penelitian, batasan masalah, manfaat Tugas Akhir, 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. Teori-teori yang dipaparkan adalah seputar tentang konsep dasar sistem informasi, pengembangan sistem (*System Development Life Cycle*), Diagram Alur (*Flowchart*), *Unified Modeling Language* (UML), *Windows Navigation Diagram*, *Entity Relationship Diagram* (ERD), konsep dasar *Java*, *Database*, *Netbeans*, *MySQL* dan *XAMPP*.

BAB III METODOLOGI PENELITIAN

Bab ini membahas tentang langkah-langkah yang akan dilakukan dalam identifikasi masalah, metode pengumpulan data, serta metodologi pendekatan dan pengembangan sistem yang digunakan.

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Bab ini membahas pengumpulan data berupa proses bisnis dan dokumen yang terlibat pada sistem yang sedang berjalan, pengolahan hasil pengamatan proses bisnis pada sistem yang berjalan yang berkaitan dengan penerimaan dan pengiriman komponen.

BAB V ANALISIS DAN PEMBAHASAN

Bab ini berisi analisis dari sistem informasi yang sedang berjalan dan analisis permasalahan yang ada di lapangan. Setelah itu dilakukan analisis sistem usulan dan perancangan sistem usulan dengan menggunakan *Flowchart*, *Unified Modeling Language (UML)*, *Windows Navigation Diagram*, perancangan basis data, Implementasi dengan menggunakan *Java 2 Standard Edition (J2SE)*, *Java Development Kit (JDK)*, Netbeans, dan MySQL.

BAB VI PENUTUP

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

BAB II

LANDASAN TEORI

2.1 Pengertian Rancang Bangun

Menurut Pressman (2002) rancang bangun merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan *detail* bagaimana komponen-komponen sistem diimplementasikan.

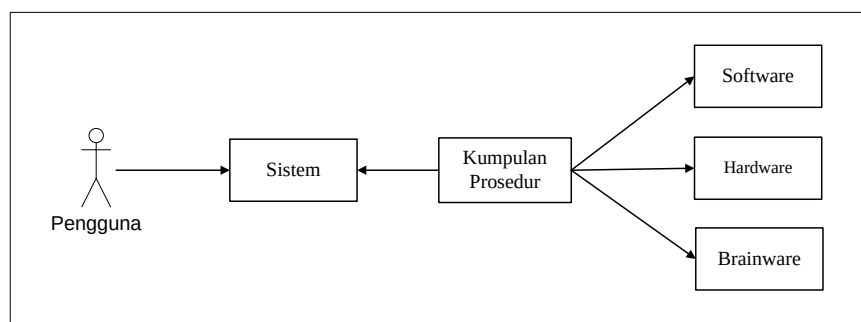
Sedangkan menurut Mahyudanil (2014) rancang bangun adalah tahap setelah analisis dari siklus pengembangan sistem yang merupakan pendefinisian dari kebutuhan-kebutuhan fungsional, serta menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, termasuk menyangkut menkonfigurasi dari komponen-komponen perangkat keras dan perangkat lunak dari suatu sistem.

2.2 Konsep Dasar Sistem Informasi

Sistem informasi merupakan gabungan dari empat bagian utama. Keempat bagian tersebut mencakup 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 data menjadi informasi yang bermanfaat. Di dalamnya juga termasuk proses perencanaan, kontrol, koordinasi, dan pengambilan keputusan. Sehingga sebuah sistem yang mengolah data menjadi informasi yang akan disajikan dan digunakan oleh pengguna, maka sistem informasi merupakan sebuah sistem yang kompleks. Bukan hanya komputer saja yang bekerja (beserta *software* dan *hardware* di dalamnya), namun juga manusia (dengan *brainware* yang dimiliki). Manusia (pengguna/aktor) dalam hal ini menggunakan seluruh ide, pemikiran, perhitungan, untuk dituangkan ke dalam sistem informasi yang digunakan (Pratama, 2014).

2.2.1 Pengertian Sistem

Sistem didefinisikan sebagai sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama. Secara garis besar, sebuah sistem terdiri atas tiga komponen utama. Ketiga komponen tersebut mencakup *software*, *hardware*, dan *brainware*. Ketiga komponen ini saling berkaitan satu sama lain (Pratama, 2014). Gambar II.1 bagan di bawah ini mengilustrasikan sebuah sistem yang terdiri atas tiga aspek (*software*, *hardware*, dan *brainware*), prosedur, serta pengguna (manusia) itu sendiri.



Gambar II.1 Sistem dengan Tiga Aspek, Prosedur, dan Pengguna
(Sumber: Pratama, 2014)

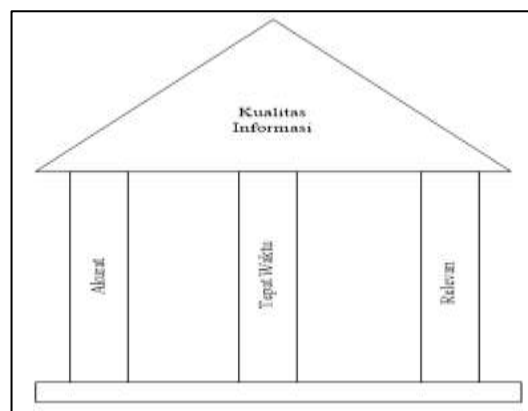
Kata sistem sendiri dari bahasa Latin “*Systema*” dan bahasa Yunani “*Sustema*” adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi. Keempat bagian utama ini saling berkaitan untuk menciptakan sebuah sistem yang dapat mengolah data menjadi informasi yang bermanfaat. Di dalamnya juga termasuk proses perencanaan, kontrol, koordinasi, dan pengambilan keputusan. Sehingga sebagai sebuah sistem yang mengolah data menjadi informasi yang akan disajikan dan digunakan oleh pengguna. Bukan hanya komputer saja yang bekerja (beserta *software* dan *hardware* di dalamnya), namun juga manusia (dengan *brainware* yang dimiliki). Manusia (pengguna atau aktor) dalam hal ini menggunakan seluruh ide, pemikiran, perhitungan, untuk dituangkan ke dalam sistem yang digunakan.

Pada dasarnya manusia hidup di dunia penuh dengan sistem. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi dengan suatu model matematika. Apa yang terlihat disekeliling kita dalam kehidupan sehari-hari sebenarnya merupakan kumpulan dari sebuah sistem, dan sistem tersebut tersusun atau terorganisir dengan kompleks. Dikatakan kompleks karena dunia ini memang tersusun dari beberapa subsistem yang berbeda satu dengan yang lain dan mereka berinteraksi

pada tingkatan tertentu. Sedangkan sistem dalam sudut pandang perusahaan atau organisasi adalah tempat sumber daya manusia yang berinteraksi untuk merealisasikan formulasi tujuan yang telah ditetapkan.

2.2.2 Pengertian Informasi

Informasi atau dalam Bahasa Inggrisnya adalah *information*, berasal dari kata *informacion* Bahasa Prancis. Kata tersebut diambil dari Bahasa Latin, yaitu *informationem* yang artinya konsep, ide, garis besar. Informasi adalah suatu data yang sudah diolah atau diproses sehingga menjadi suatu bentuk yang memiliki arti bagi penerima informasi yang memiliki nilai bermanfaat (Gelinas dan Dull, 2012). Dalam buku Mustakini (2012) menggambarkan kualitas dari informasi dengan bentuk bangunan yang ditunjang oleh tiga buah pilar.



Gambar II.2 Pilar kualitas informasi
(Sumber: Mustakini, 2012)

1. Akurat

Berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan (*noise*) yang dapat merubah atau merusak informasi.

2. Tepat pada waktunya

Berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan di dalam pengambilan keputusan. Bila pengambilan keputusan terlambat maka dapat berakibat fatal untuk organisasi.

3. Relevan

Berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lain berbeda. Misalnya informasi mengenai sebab-musabab kerusakan mesin produksi kepada akuntan perusahaan adalah kurang relevan dan akan lebih relevan bila ditunjukkan kepada ahli teknik perusahaan.

2.2.3 Pengertian Sistem Informasi

Sistem informasi merupakan gabungan dari empat bagian utama. Keempat bagian tersebut mencakup 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 data menjadi informasi yang bermanfaat. Di dalamnya juga termasuk proses perencanaan, kontrol, koordinasi, dan pengambilan keputusan. Sehingga sebuah sistem yang mengolah data menjadi informasi yang akan disajikan dan digunakan oleh pengguna, maka sistem informasi merupakan sebuah sistem yang kompleks. Bukan hanya komputer saja yang bekerja (beserta *software* dan *hardware* di dalamnya), namun juga manusia (dengan *brainware* yang dimiliki). Manusia (pengguna/aktor) dalam hal ini menggunakan seluruh ide, pemikiran, perhitungan, untuk dituangkan ke dalam sistem informasi yang digunakan (Pratama, 2014).

2.2.4 Komponen Sistem Informasi

Komponen Sistem Informasi menurut Pratama (2014), disebutkan mengenai adanya komponen-komponen di dalam sebuah sistem informasi. Komponen-komponen ini memiliki fungsi dan tugas masing-masing yang saling berkaitan satu sama lain. Keterkaitan antar komponen ini membentuk suatu kesatuan kerja, yang menjadikan sistem informasi dapat mencapai tujuan dan fungsi yang ingin dicapai oleh pengguna dan pengembang sistem informasi bersangkutan.

Komponen-komponen yang terdapat di dalam semua jenis sistem informasi mencakup tujuh poin (Pratama, 2014). Berikut ketujuh komponen sistem informasi, antara lain:

1. *Input* (Masukan)

Sebuah informasi berasal dari data yang telah diolah dan diverifikasi sehingga akurat, bermanfaat, dan memiliki nilai. Komponen *input* ini berfungsi untuk menerima semua *input* (masukan) dari pengguna. Masukan yang diterima dalam bentuk data. Data ini berasal dari satu maupun beberapa buah sumber.

Jika dilihat dari cakupan organisasi/tempat dimana sistem informasi berada, maka data dapat digolongkan ke dalam data internal dan data eksternal. Data internal merupakan data yang berasal dari dalam organisasi/tempat yang bersangkutan. Data eksternal merupakan data yang berasal dari luar organisasi/tempat yang bersangkutan (misal data yang berasal dari sumber referensi internet).

2. *Output* (Keluaran)

Sebuah sistem informasi akan menghasilkan keluaran (*output*) berupa informasi. Komponen keluaran berfungsi untuk menyajikan hasil akhir ke pengguna sistem informasi. Informasi yang disajikan ini merupakan hasil dari pengolahan data yang telah dimasukkan sebelumnya. Pada komponen *output*, informasi yang disajikan disesuaikan dengan data yang di-*input* dan fungsionalitas dari sistem informasi bersangkutan.

3. *Software* (Perangkat Lunak)

Komponen *software* (perangkat lunak) mencakup semua perangkat lunak yang digunakan di dalam sistem informasi. Adanya komponen perangkat lunak ini akan membantu sistem informasi di dalam menjalankan tugasnya dan untuk dapat dijalankan sebagaimana mestinya. Komponen perangkat lunak ini melakukan proses pengolahan data, penyajian informasi, perhitungan data, dan lain-lain. Komponen perangkat lunak mencakup sistem operasi, aplikasi dan *driver*.

2. *Hardware* (Perangkat Keras)

Komponen *hardware* (perangkat keras) mencakup semua perangkat keras komputer yang digunakan secara fisik di dalam sistem informasi, baik di komputer *server* maupun di komputer *client*. Komponen *hardware* (perangkat keras) ini meliputi komputer *server* beserta komponen di dalamnya, komputer *desktop* beserta komponen di dalamnya, komputer jinjing beserta komponen di dalamnya, *mobile device* (*tablet*, *smartphone*), dan lain-lain. Termasuk juga di dalamnya *hub*, *switch*, *router*, yang berperan di dalam jaringan komputer (untuk media komunikasi di dalam sistem informasi).

3. *Database (Basis Data)*

Mengingat bahwa sistem informasi menyajikan informasi yang berasal dari satu maupun beberapa data yang dimasukkan dan diolah, maka tentu diperlukan sebuah aplikasi penyimpanan, mengolah, dan menyajikan data dan informasi tersebut secara komputerisasi. Komponen basis data berfungsi untuk menyimpan semua data dan informasi ke dalam satu atau beberapa tabel. Setiap tabel memiliki *field* masing-masing. Setiap tabel memiliki fungsi penyimpanan masing-masing, serta antar tabel dapat juga terjadi relasi (hubungan).

4. Kontrol dan Prosedur

Kontrol dan prosedur adalah dua buah komponen yang menjadi satu. Komponen kontrol berfungsi untuk mencegah terjadinya beragam gangguan dan ancaman terhadap data dan informasi yang ada di dalam sistem informasi, termasuk juga sistem informasi itu sendiri beserta fisiknya (dalam hal ini komputer *server*).

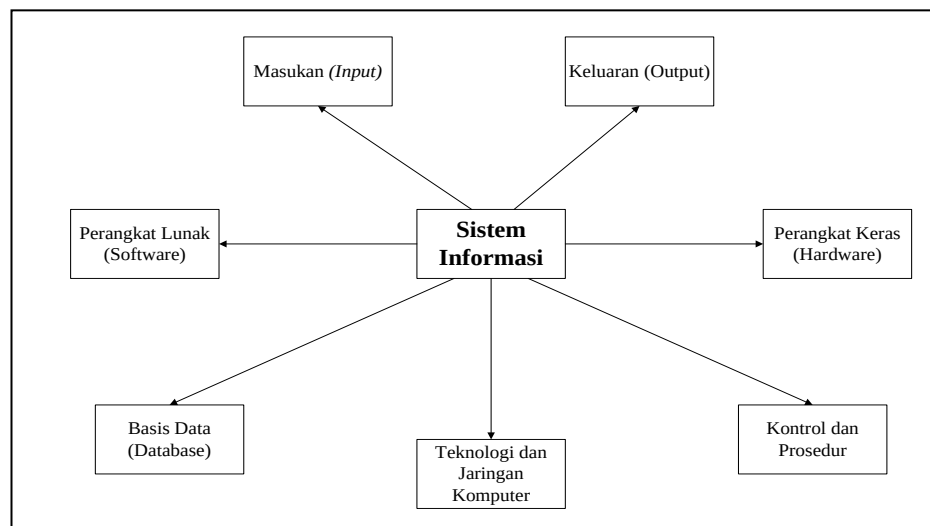
Komponen prosedur mencakup semua prosedur dan aturan yang harus dilakukan dan wajib ditaati bersama, guna mencapai tujuan yang diinginkan. Komponen ini berkaitan dengan komponen kontrol dalam hal pencegahan terhadap kemungkinan ancaman dan gangguan yang terjadi pada sistem informasi, yang berpengaruh terhadap layanan yang diberikan, informasi yang disajikan, dan tingkat kepuasan pengguna.

5. Teknologi dan Jaringan Komputer

Komponen terakhir di dalam sistem informasi ini, yaitu teknologi dan jaringan komputer, memegang peranan terpenting untuk sebuah sistem informasi. Komponen teknologi mengatur *software*, *hardware*, *database*, kontrol dan prosedur, *input*, *output*, sehingga sistem dapat berjalan dan terkendali dengan baik. Misalnya teknologi yang digunakan berupa sistem operasi *Linux*, *Apache web server*, *MySQL database server* (untuk *software*), seperangkat komputer *server* merek XEON (untuk *hardware*), *database MySQL* (untuk *database*), serta proses enkripsi, sensor, dan sejumlah ISO terkait dengan pencegahan ancaman dan gangguan keamanan informasi yang ada (untuk kontrol dan prosedur).

Komponen jaringan komputer berperan di dalam menghubungkan sistem informasi dengan sebanyak mungkin pengguna, baik melalui kabel jaringan (*wired*) maupun tanpa kabel (*wireless*). Jaringan komputer dapat berupa jaringan

lokal (*private*) hingga jaringan internet (*public*). Hal ini bergantung pada kebutuhan, biaya, kebijakan, situasi, dan kondisi yang ada.



Gambar II.3 Komponen-Komponen Sistem Informasi
(Sumber: Pratama, 2014)

2.3 Pengertian Sistem Informasi Penerimaan dan Pengiriman Bahan Baku atau Komponen

Sistem informasi penerimaan dan pengiriman bahan baku adalah sistem informasi manajemen yang menyediakan informasi untuk digunakan oleh fungsi penerimaan dan pengiriman bahan baku. Sistem informasi penerimaan bahan baku mendukung fungsi penerimaan yang meliputi semua aktivitas yang berkaitan dengan proses saat bahan baku datang dan bahan baku masuk ke gudang (Zakiyudin, 2012). Sistem informasi digunakan untuk manajemen operasi dan pemrosesan transaksi pendukung semua perusahaan yang harus merencanakan, memonitor, dan mengendalikan persediaan, pembelian, serta arus barang dan jasa. Oleh karenanya, perusahaan-perusahaan seperti perusahaan transportasi, grosir, peritel, lembaga keuangan, dan perusahaan jasa harus menggunakan sistem informasi penerimaan dan pengiriman bahan baku untuk mengatur kedatangan bahan baku mereka agar tidak terjadi kecurangan (Wilkinson, 2012).

2.3.1 Pengertian Penerimaan

Transaksi pembelian lokal bahan baku melibatkan bagian-bagian produksi,

gudang, pembelian, penerimaan barang, dan asuransi. Dokumen sumber dan dokumen pendukung yang dibuat dalam transaksi pembelian lokal bahan baku terdiri dari prosedur permintaan pembelian lain, prosedur order pembelian, prosedur penerimaan barang digudang (Mulyadi, 2010).

a. **Prosedur Penerimaan Pembelian Bahan Baku**

Jika persediaan bahan baku yang ada digudang sudah mencapai tingkat minimum pemesanan kembali, bagian gudang kemudian membuat surat permintaan pembelian untuk dikirim untuk dikirim ke bagian pembelian.

b. **Prosedur Order Pembelian**

Bagian pembelian melaksanakan pembelian atas dasar surat permintaan pembelian dari bagian gudang. Untuk pemilihan pemasok, bagian pembelian mengirimkan surat permintaan penawaran harga kepada para pemasok, yang berisi permintaan informasi harga dan syarat-syarat pembelian dari masing-masing pemasok tersebut. Setelah pemasok yang dianggap baik dipilih, bagian pembelian kemudian membuat surat order pembelian untuk dikirim kepada pemasok yang dipilih.

c. **Prosedur Penerimaan Bahan Baku**

Pemasok mengirim bahan baku kepada perusahaan sesuai dengan surat order pembelian yang diterimanya. Bagian penerimaan bertugas menerima barang, memcocokkan kualitas, kuantitas, jenis serta spesifikasi bahan baku yang diterima sesuai dengan surat order pembelian.

d. **Prosedur Pencatatan Penerimaan Bahan Baku di Bagian Gudang.**

Bagian penerimaan menyerahkan bahan baku yang diterima dari pemasok ke bagian gudang. Bagian gudang menyimpan bahan baku tersebut dan mencatat jumlah bahan baku dalam kartu gudang. Kartu gudang ini digunakan oleh bagian gudang untuk mencatat mutasi tiap-tiap barang digudang.

2.3.2 Pengertian Pengiriman

Pengiriman adalah proses pemindahan secara fisik barang dari gudang ketempat tujuan yang disesuaikan dengan dokumen pemesanan dan pengiriman serta dalam kondisi yang sesuai dengan persyaratan penanganan barangnya.

Menurut Kotler (2007), pengiriman/distribusi terdiri dari seperangkat yang melibatkan perencanaan, penerapan dan pengendalian arus bahan dan produk akhir

dari titik asal ke titik penggunaan untuk memenuhi kebutuhan dan menciptakan laba. Distribusi melibatkan perencanaan, pengimplementasian, dengan pengendalian arus fisik bahan mentah, barang jadi dan informasi-informasi yang terkait mulai dari titik asal barang hingga titik konsumsi untuk memenuhi kebutuhan dan memperoleh laba.

2.3.3 Pengertian Bahan Baku atau Komponen

Kelancaran proses produksi sangat ditentukan oleh tersedianya bahan baku dalam jumlah dan ukuran yang sesuai dengan kebutuhan perusahaan. Hal ini disebabkan karena bahan baku merupakan faktor utama dalam pelaksanaan proses produksi pada suatu perusahaan.

Menurut Assauri (2016) pengertian bahan baku meliputi semua bahan yang dipergunakan dalam perusahaan pabrik, kecuali terhadap bahan-bahan yang secara fisik akan digabungkan dengan produk yang dihasilkan oleh perusahaan pabrik tersebut. Sedangkan komponen adalah suatu barang yang terdiri dari beberapa suku cadang yang membentuk satu kesatuan dan mempunyai fungsi tertentu. Komponen adalah keseluruhan makna yang terdiri dari sejumlah elemen, dimana elemen yang satu dengan yang lainnya memiliki ciri khusus yang berbeda-beda (Buntarto, 2015).

Komponen menurut PT Krama Yudha Ratu Motor merupakan unsur yang berhubungan langsung terhadap barang yang akan di produksi. Komponen juga dapat disebut dengan bahan baku. Komponen ini terbagi menjadi dua jenis, yaitu komponen lokal dan komponen CKD (*Completely Knock Down*). Komponen yang berasal dari *vendor* dalam negeri disebut lokal sedangkan komponen yang harus dibeli dari *vendor* dari luar negeri disebut CKD.

2. 4 Pengembangan Sistem

Pengembangan sistem dapat berarti menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada. Sistem yang lama perlu diperbaiki oleh karena beberapa hal, yaitu (Sutabri, 2004): Munculnya masalah pada sistem yang lama, Untuk meraih kesempatan, Adanya instruksi.

Proses perancangan atau pengembangan sistem informasi, mulai dari konsep sampai dengan implementasinya disebut dengan istilah *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* (SDLC) adalah proses

mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya berdasarkan *best practice* atau cara-cara yang sudah teruji baik (Rosa dan Shalahuddin, 2011). Menurut Rosa dan Shalahuddin (2011), tahapan-tahapan yang ada pada SDLC secara global adalah sebagai berikut:

1. Inisiasi (*initiation*)

Tahap ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak.

2. Pengembangan konsep sistem (*system concept development*)

Mendefinisikan lingkup konsep termasuk dokumen lingkup sistem analisis manfaat biaya, manajemen rencana dan pembelajaran kemudahan sistem.

3. Perencanaan (*planning*)

Mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya. Menyediakan dasar untuk mendapatkan sumber daya (*resources*) yang dibutuhkan untuk memperoleh solusi.

4. Analisis kebutuhan (*requirements analysis*)

Menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *user*. Membuat dokumen kebutuhan fungsional.

5. Desain (*design*)

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

6. Pengembangan (*development*)

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

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

Mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional. Dengan diarahkan

oleh staf penjamin kualitas (*quality assurance*) dan *user* sehingga menghasilkan laporan analisis pengujian.

8. Implementasi (*implementation*)

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

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

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

10. Disposisi (*disposition*)

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

System Development Life Cycle (SDLC) adalah pendekatan bertahap untuk melakukan analisa dan membangun rancangan sistem dengan menggunakan siklus yang spesifik terhadap kegiatan pengguna (Kendall & Kenneth, 2003). Selain itu, *System Development Life Cycle* (SDLC) adalah sebuah proses memahami bagaimana sistem informasi dapat mendukung kebutuhan bisnis, merancang sistem, membangun sistem, dan memberikannya kepada pengguna (Dennis, Wixom, & Tegarden, 2005).

System Development Life Cycle (SDLC) juga merupakan pusat pengembangan sistem informasi yang efisien. SDLC terdiri dari 4 (empat) langkah kunci yaitu (Valacich, George, & Hoffer, 2012):

1. Perencanaan

Fase perencanaan adalah sebuah proses dasar untuk memahami mengapa sebuah sistem harus dibangun. Pada fase ini diperlukan analisa kelayakan dengan mencari data atau melakukan proses *information gathering* kepada pengguna.

2. Analisa

Fase analisa adalah sebuah proses investigasi terhadap sistem yang sedang berjalan dengan tujuan untuk mendapatkan jawaban mengenai pengguna sistem, cara kerja sistem dan waktu penggunaan sistem. Dari proses analisa ini akan didapatkan cara untuk membangun sistem baru.

3. Rancangan

Fase perancangan merupakan proses penentuan cara kerja sistem dalam hal *architechture design*, *interface design*, *database* dan spesifikasi *file*, dan *program design*. Hasil dari proses perancangan ini akan didapatkan spesifikasi sistem.

4. Implementasi

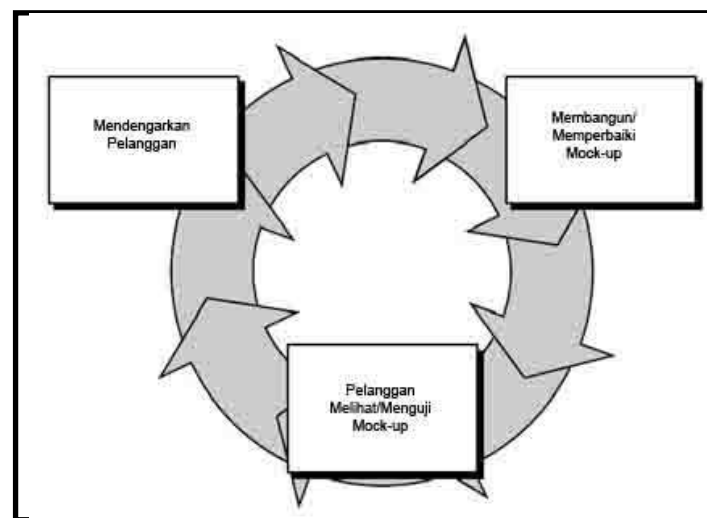
Fase implementasi adalah proses pembangunan dan pengujian sistem, instalasi sistem, dan rencana dukungan sistem. Berdasarkan pada penjelasan di atas maka SDLC dapat disimpulkan sebagai sebuah siklus untuk membangun sistem dan memberikannya kepada pengguna melalui tahapan perencanaan, analisa, perancangan dan implementasi dengan cara memahami dan menyeleksi keadaan dan proses yang dilakukan pengguna untuk dapat mendukung kebutuhan pengguna. Untuk menggunakan SDLC maka dibutuhkan sumber data awal dari pengguna yang dijadikan acuan dalam perencanaan, analisa, perancangan dan implementasi. Penggunaan acuan ini dimaksudkan agar sistem yang dibangun bisa menjembatani kebutuhan pengguna dari permasalahan yang dihadapinya.

2.4.1 Model Prototipe Secara Umum

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

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

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



Gambar II.4 Ilustrasi Model Prototipe
Sumber: Rosa dan Shalahuddin (2011)

Mock-up adalah sesuatu yang digunakan sebagai model desain yang digunakan untuk mengajar, demonstrasi, evaluasi desain, promosi atau keperluan lain. Sebuah *mock-up* disebut sebagai prototipe perangkat lunak jika menyediakan atau mampu mendemonstrasikan sebagian besar fungsi sistem perangkat lunak dan memungkinkan pengujian desain sistem perangkat lunak. Iterasi terjadi pada pembuatan prototipe sampai sesuai dengan keinginan pelanggan atau *user* (Rosa dan Shalahuddin, 2011).

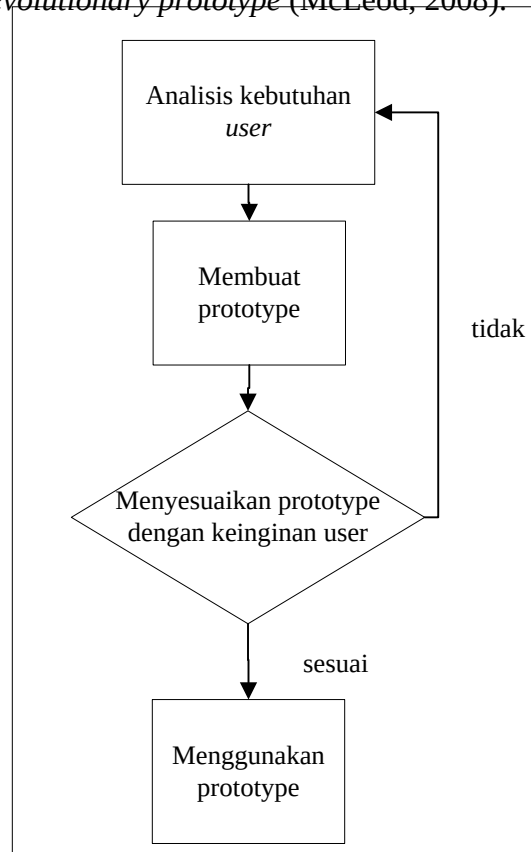
Kelemahan model prototipe adalah sebagai berikut (Rosa dan Shalahuddin, 2011):

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

2.4.2 *Evolutionary Prototype*

Evolutionary prototype yaitu, *prototype* yang secara terus menerus dikembangkan hingga *prototype* tersebut memenuhi fungsi dan prosedur yang dibutuhkan oleh sistem. Pada pendekatan evolusioner, suatu *prototype* dibangun

berdasarkan pada kebutuhan dan pemahaman secara umum. *Prototype* kemudian diubah dan dievolusikan daripada dibuang. *Prototype* yang dibuang biasanya digunakan dengan aspek sistem yang dimengerti secara luas dan dibangun atas kekuatan tahapan *evolutionary prototype* (McLeod, 2008).



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

Penjelasan dari gambar tersebut adalah sebagai berikut:

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

2.5 *Flowchart*

Flowchart merupakan penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program. *Flowchart* menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. (Fathansyah, 2015).

Flowchart digunakan untuk alat bantu komunikasi dan dokumentasi. Pada waktu menggambar suatu diagram alir, analisis sistem dapat mengikuti ketentuan-ketentuan sebagai berikut (Mustakini, 2012):

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

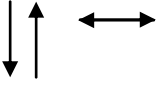
Simbol-simbol *flowchart* yang digunakan merupakan simbol-simbol *flowchart* standar yang dikeluarkan oleh ANSI dan ISO. Berikut simbol-simbol standar yang digunakan untuk menggambarkan diagram alir sesuai kegunaan simbol (Fathansyah, 2015):

1. *Flow Direction Symbols*

Simbol yang dipakai untuk menghubungkan antara simbol yang satu dengan simbol lainnya atau disebut juga *connecting line*. Tabel II.1 menjelaskan tentang simbol-simbol penghubung dalam menggambarkan diagram alir.

Tabel II.1 Simbol *Flow Direction*

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

	<i>Arus/Flow</i>	Penghubung antara prosedur/proses.
	<i>Connector</i>	Simbol keluar/masuk prosedur atau proses dalam lembar/halaman yang sama.
	<i>Off-line Connector</i>	Simbol keluar/masuk prosedur atau proses dalam lembar/halaman yang lain.

(Sumber: Fathansyah, 2015)

2. Simbol Proses

Simbol yang menunjukkan jenis operasi pengolahan dalam suatu prosedur. Tabel II.2 menjelaskan tentang simbol-simbol proses dalam menggambarkan diagram alir.

Tabel II.2 Simbol Proses

Simbol	Nama	Keterangan
	<i>Predafined Process</i>	Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam <i>storage</i> .
	<i>Process</i>	Simbol yang menunjukan pengolahan yang dilakukan komputer.
	Terminal	Simbol untuk permulaan atau akhir dari suatu program
	Manual Input	Simbol untuk pemasukan data secara manual on-line keyboard.
	Decision	Simbol untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban/aksi.

(Sumber: Fathansyah, 2015)

3. Simbol *Input* dan *Output*

Simbol yang dipakai untuk menyatakan jenis peralatan yang digunakan sebagai media *input* atau *output*. Tabel II.3 menjelaskan tentang simbol-simbol *input* dan *output* dalam menggambarkan diagram alir.

Tabel II.3 Simbol *Input* dan *Output*

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

	<i>Input-Output</i>	Simbol yang menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.
	<i>Document</i>	Simbol yang menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau <i>output</i> dicetak di kertas.
	<i>Disk and On-line Storage</i>	Simbol untuk menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i> .

(Sumber: Fathansyah, 2015)

2.6 Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Tujuan dari UML adalah untuk menyediakan kosa kata yang umum dari istilah-istilah berbasis objek dan teknik yang cukup banyak untuk memodelkan proyek pengembangan sistem dari analisis ke desain. Menurut Dennis (2010), diagram-diagram yang ada dalam UML terbagi menjadi dua bagian utama yaitu, *Struktur diagram* dan *Behavior diagram*. *Structure diagram* biasanya digunakan untuk merepresentasikan data dan hubungan statik yang ada di dalam sebuah sistem informasi. Sedangkan *Behavior diagram* menyediakan para analis dengan sebuah gambaran hubungan yang dinamis antara instansi atau objek yang merepresentasikan sistem informasi bisnis. Versi 2.0 ini sudah diterima oleh *Object Management Group* (OMG) pada tahun 2003. Versi UML ini mendefinisikan suatu set dari 14 teknik untuk pemodelan sistem. Berikut adalah tabel dari jenis diagram yang ada dan fungsi-fungsinya:

Tabel II.4 Jenis Diagram Pada UML dan Fungsi-fungsinya

Nama Diagram	Fungsi
<i>Structure Diagrams</i>	
<i>Class Diagram</i>	Menggambarkan hubungan antara pemodelan-pemodelan <i>class</i> di dalam sistem
<i>Object Diagram</i>	Menggambarkan hubungan antara pemodelan-pemodelan objek di dalam sistem

<i>Package Diagram</i>	Mengelompokkan elemen UML lainnya menjadi satu untuk membentuk level konstruksi yang lebih tinggi
<i>Deployment Diagram</i>	Menampilkan arsitektur fisik dari suatu sistem. Bisa juga digunakan untuk menunjukkan komponen <i>software</i> sedang dijadikan arsitektur fisik suatu sistem
<i>Component Diagram</i>	Menggambarkan hubungan fisik diantara komponen-komponen <i>software</i>
Behavior Diagram	
<i>Activity Diagram</i>	Menggambarkan proses bisnis masing-masing <i>class</i>
<i>Sequence Diagram</i>	Memodelkan kebiasaan dari objek-objek dalam suatu <i>use case</i> . Fokus dalam aktifitas berdasarkan urutan waktu
<i>Communication Diagram</i>	Memodelkan kebiasaan dari objek-objek dalam suatu <i>use case</i> . Fokus dalam komunikasi antara satu set dari kolaborasi objek dari suatu objek
<i>Interaction Overview Diagram</i>	Menggambarkan suatu ikhtisar alur dari kontrol suatu proses
<i>Timing Diagram</i>	Menggambarkan interaksi yang terjadi diantara suatu set dari objek-objek dan perubahan keadaan selama perjalanan waktu

Tabel II.4 Jenis Diagram Pada UML dan Fungsi-fungsinya (Lanjutan)

Nama Diagram	Fungsi
<i>Behavioral State Machine Diagram</i>	Memeriksa kebiasaan dari suatu <i>class</i>
<i>Protocol State Machine Diagram</i>	Menggambarkan ketergantungan diantara perbedaan-perbedaan <i>interface</i> dari suatu <i>class</i>
<i>Use Case Diagram</i>	Menangkap kebutuhan bisnis untuk sistem dan untuk menggambarkan interaksi diantara sistem dan lingkungannya

(Sumber: Dennis, 2010)

2.6.1 Use Case Diagram

Diagram *use case* dibuat untuk menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Suatu landasan pola pikir yang ditekankan dalam

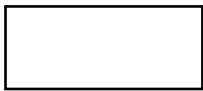
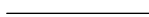
diagram *use case* ini adalah “apa” yang dapat diperbuat oleh sistem, dan bukan “bagaimana” sistem melakukannya.

Diagram *use case* menggambarkan kebutuhan sistem dari sudut pandang *user*, yaitu proses yang dilakukan oleh sistem dalam melayani *user* yang berinteraksi dengan sistem tersebut. Diagram ini sangat tepat untuk menggambarkan hubungan yang terjadi antara sistem dengan *user*nya.

Tabel II.5 Simbol-Simbol *Use Case Diagram*

No.	Simbol	Deskripsi
1.	<i>Use Case</i> 	– Merepresentasikan bagian utama dari sistem secara fungsional. – Diletakan didalam <i>system boundary</i>. – Dilabelkan dengan frasa kata kerja deskriptif
2.	<i>Actor</i> 	– Seseorang atau sistem yang mendapatkan keuntungan dari sistem – Dilabelkan dengan peran – Bisa diasosiasikan dengan aktor lainnya berdasarkan spesialisasi

Tabel II.5 Simbol-Simbol *Use Case Diagram* (Lanjutan)

No.	Simbol	Deskripsi
3.	<i>Boundary</i> 	– Nama dari sistem terdapat didalam atau diatas – Merepresentasikan ruang lingkup dari sistem
4.	<i>Association</i> 	– Menghubungkan suatu aktor dengan use case dengan apa interaksinya
5.	<i>Extend</i> 	– Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu.
6.	<i>Generalization</i> 	– Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

7.	<i>Include</i> 	– Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.
----	---	---

(Sumber: Dennis, 2010)

2.6.2 Activity Diagram

Digunakan untuk model perilaku dalam independen proses bisnis benda. Dalam banyak hal, diagram aktivitas dapat dipandang sebagai diagram aliran data yang canggih yang digunakan dalam hubungannya dengan analisis terstruktur. Namun, tidak seperti aliran data diagram, diagram aktivitas termasuk notasi yang membahas pemodelan paralel, kegiatan bersamaan dan proses. (Dennis, 2010)

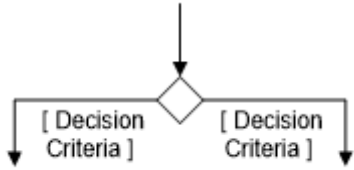
Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

- Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
- Urutan atau pengelompokan tampilan dari sistem atau *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan
- Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya
- Rancangan menu yang akan ditampilkan pada perangkat lunak

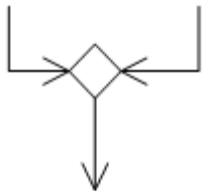
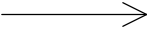
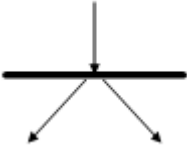
Berikut adalah simbol-simbol yang ada pada *Activity Diagram*:

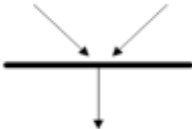
Tabel II.6 Simbol-Simbol *Activity Diagram*

No.	Simbol	Deskripsi
1.	<i>Initial Node</i> 	Merupakan tanda awal dari sebuah aktifitas.
2.	<i>Activity</i> 	Merupakan sebuah gambaran aktifitas yang terjadi

3.	<i>Decision Node</i> 	<i>Decision node</i> digunakan untuk merepresentasikan suatu alur logika yang timbul dari sekumpulan / urutan aktivitas pada suatu proses bisnis. Alur logika ini merupakan pilihan atas jalur aktivitas yang bernilai "<i>True</i>" dan "<i>False</i>", dan hanya salah satu dari jalur tersebut yang akan dipilih sesuai dengan syarat/kriteria pada <i>Decision node</i> yang telah ditentukan
----	---	--

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

No.	Simbol	Deskripsi
4.	Merge Node 	Merge node digunakan untuk menyatukan / menutup alur logika yang sebelumnya dibentuk oleh Decision node
5.	Final-Activity Node 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
7.	<i>Control flow</i> 	Menunjukkan urutan eksekusi
8.	<i>Fork</i> 	Fork node digunakan untuk membagi (<i>split</i>) sekumpulan aktivitas dalam suatu proses bisnis, dimana sekumpulan aktivitas tersebut dapat berjalan secara paralel dalam satu kondisi waktu yang sama.

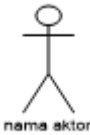
9.	<i>Join</i> 	Join node digunakan untuk menyatukan / menutup aktivitas yang berjalan paralel yang sebelumnya dibentuk oleh Fork node
----	--	--

(Sumber: Dennis, 2010)

2.6.3 Sequence Diagram

Sequence diagram adalah salah satu dari dua jenis diagram interaksi. Mereka menggambarkan benda-benda yang berpartisipasi dalam kasus penggunaan dan pesan yang melewati antara mereka dari waktu ke waktu untuk satu *use case*. Sebuah diagram *sequence* adalah model dinamis yang menunjukkan urutan eksplisit pesan yang lewat di antara objek dalam interaksi didefinisikan. Karena urutan diagram menekankan pemesanan berbasis waktu kegiatan yang terjadi di antara set benda, mereka sangat membantu untuk memahami spesifikasi *real-time* dan kompleks menggunakan kasus (Dennis, 2010). Berikut simbol yang ada pada *Sequence* Diagram:

Tabel II.7 Simbol-Simbol *Sequence Diagram*

No.	Simbol	Deskripsi
1.	Aktor 	Orang atau sistem yang berasal dari manfaat dan eksternal ke sistem yang berpartisipasi secara berurutan dengan mengirim dan atau menerima pesan
2.	Garis Hidup/<i>lifeline</i> 	Menyatakan kehidupan suatu objek.
3.	Objek 	Berpartisipasi secara berurutan dengan mengirim dan atau menerima pesan yang ditempatkan diatas diagram.
4.	<i>Execution Occurrence</i> 	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.

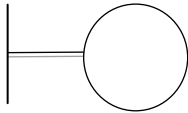
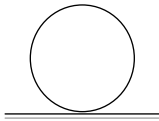
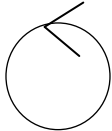
5.	<i>Message</i> <i>aMessage()</i> 	Pesan yang menggambarkan komunikasi yang terjadi antar objek.
----	--	---

Tabel II.7 Simbol-Simbol *Sequence Diagram*(Lanjutan)

No.	Simbol	Deskripsi
6.	<i>Object Destruction</i> X	Ditempatkan pada akhir dari suatu objek <i>lifeline</i> untuk menunjukkan bahwa itu akan keluar dari eksistensi.

(Sumber: Dennis, 2010)

Tabel II.7 Simbol-Simbol *Sequence Diagram* (lanjutan)

No.	Simbol	Deskripsi
Adapun jenis-jenis objek <i>Sequence Diagram</i> sebagai berikut:		
1.	<i>Boundary</i> 	Berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>formentry</i> dan <i>form</i> cetak.
2.	<i>Entity Class</i> 	Merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
3.	<i>Control Class</i> 	Suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.

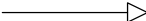
(Sumber: Gata, 2013)

2.6.4 Class Diagram

Class diagram merupakan salah satu diagram dalam konsep *Unified Modelling Language* (UML) yang menjelaskan mengenai berbagai jenis objek yang terdapat dalam sistem beserta beberapa hubungan antar objek tersebut. Sebuah *Class*

merupakan deskripsi dari sekumpulan objek yang memiliki properti (*attribute*), operasi (*method*), relasi (*association*), dan tingkah laku (*behavior*) yang sama. Sebuah *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda /fungsi). (Dennis, 2010). Berikut adalah simbol-simbol class diagram menurut Dennis 2010:

Tabel II.8 Simbol-Simbol *Class Diagram*

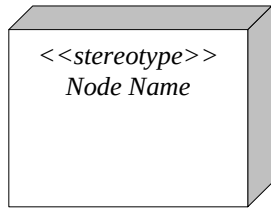
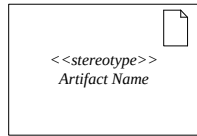
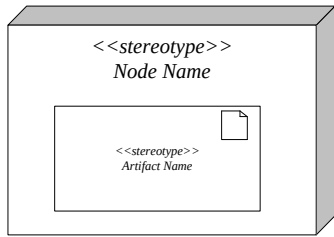
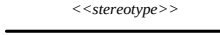
No.	Simbol	Deskripsi
1.	Class Nama Class	Class digunakan sebagai <i>template</i> dari kumpulan objek.
2.	Asosiasi berpanah 	Asosiasi (<i>Association</i>) yaitu hubungan statis antar class yang direpresentasikan dengan kata kerja (<i>Verb</i>). Umumnya menggambarkan class yang memiliki atribut berupa class lain, atau class yang harus mengetahui eksistensi class lain. Panah <i>navigability</i> menunjukkan arah <i>query</i> antar <i>class</i> .
3.	Asosiasi 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	Generalization 	Merupakan sebuah <i>taxonomic relationship</i> antara <i>class</i> yang lebih umum dengan <i>class</i> yang lebih khusus
5.	Aggregation 	Menggambarkan suatu <i>class</i> terdiri dari <i>class</i> lain atau suatu <i>class</i> adalah bagian dari <i>class</i> lain.
6.	Multiplicity 10..1 (0..*)2..4 1..*2, 4..6, 8 *	<i>Multiplicity</i> merupakan gambaran dari sejumlah objek (<i>instance</i>) pada suatu class yang terlibat dalam asosiasi dengan sejumlah objek pada class yang lainnya

(Sumber: Dennis, 2010)

2.6.5 Deployment Diagram

Deployment diagram digunakan untuk mewakili hubungan diantara komponen-komponen perangkat keras dalam infrastruktur fisik sebuah sistem informasi. Misalnya, ketika merancang sistem informasi yang akan terdistribusi menggunakan *wide area network*, sebuah *deployment diagram* dapat digunakan untuk menunjukkan hubungan komunikasi antara node yang berbeda dalam jaringan. *Deployment diagram* juga dapat digunakan untuk mewakili komponen-komponen perangkat lunak dan bagaimana mereka dikerahkan dalam arsitektur atau infrastruktur dari suatu sistem informasi. Berikut adalah simbol-simbol *deployment diagram* menurut Dennis 2010:

Tabel II.9 Simbol-Simbol *Deployment Diagram*

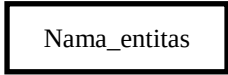
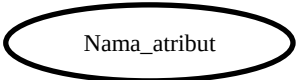
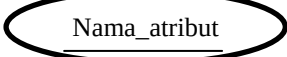
Nama Simbol	Deskripsi	Simbol
<i>Node</i>	Sumber daya komputasi seperti komputer klien, <i>server</i> atau perangkat jaringan individu.	
<i>Artifact</i>	Spesifikasi sebuah <i>software</i> atau <i>database</i> , contoh: <i>database</i> atau tabel atau <i>layer</i> .	
<i>Node with a deployed artifact</i>	Menggambarkan sebuah artefak yang ditempatkan pada node fisik. Mendukung pemodelan distribusi perangkat lunak melalui jaringan.	
<i>Communication path</i>	Merupakan asosiasi antara dua <i>node</i> . Memungkinkan <i>node</i> untuk bertukar pesan.	

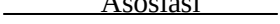
(Sumber: Dennis, 2010)

2.7 Entity Relationship Diagram (ERD)

(Rosa dan Shalahuddin, 2015) ERD adalah pemodelan awal basis data yang dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. (Rosa dan Shalahuddin, 2015) menjelaskan simbol-simbol yang digunakan dalam ERD, yaitu:

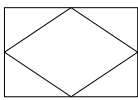
Tabel II.10 Simbol-simbol *Entity Relationship Diagram (ERD)*

No.	Simbol	Deskripsi
1.	Entitas 	Entitas merupakan data inti yang akan disimpan
2.	Atribut 	Field atau kolom data yang butuh disimpan dalam suatu entitas
3.	Atribut kunci primer 	Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record yang diinginkan.

5.	Relasi 	Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja
6.	Asosiasi 	Penghubung antara relasi dan entitas

(Sumber: Rosa dan Shalahuddin, 2015)

Tabel II.9 Simbol-simbol *Entity Relationship Diagram* (lanjutan)

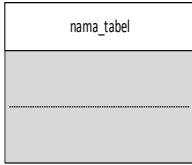
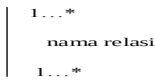
No	Simbol	Nama	Deskripsi
7.		<i>Associative Entity</i>	Hanya bisa terdapat antara 2 entitas dan menggambarkan hubungan diantara 2 entitas tersebut.

(Sumber: Hoffer, 2010)

2.8 Conceptual Data Model (CDM)

(Rosa dan Shalahuddin, 2016) CDM (*Conceptual Data Model*) atau model konsep data adalah konsep yang berkaitan dengan pandangan pemakai terhadap data yang disimpan dalam basis data. CDM dibuat sudah dalam bentuk tabel-tabel tanpa tipe data yang menggambarkan relasi antar tabel untuk keperluan implementasi ke basis data. CDM merupakan hasil penjabaran lebih lanjut dari ERD. Berikut adalah simbol-simbol yang ada pada CDM:

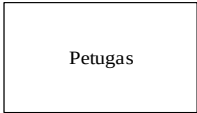
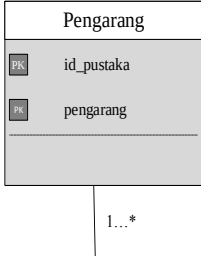
Tabel II.11 Simbol-simbol *Conceptual Data Model* (CDM)

Simbol	Deskripsi
Entitas/tabel 	Entitas atau tabel yang menyimpan data dalam basis data
Relasi 	Relasi antar tabel yang terdiri atas nama relasi dan <i>multiplicity</i>

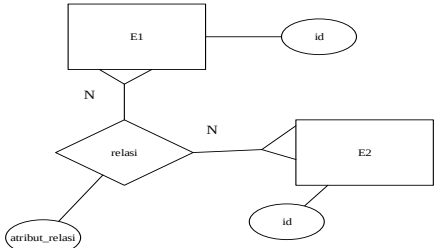
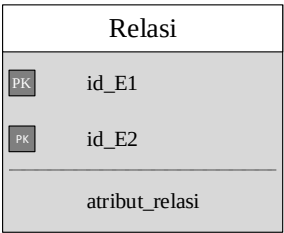
Aturan untuk mengubah ERD menjadi CDM secara umum adalah sebagai berikut:

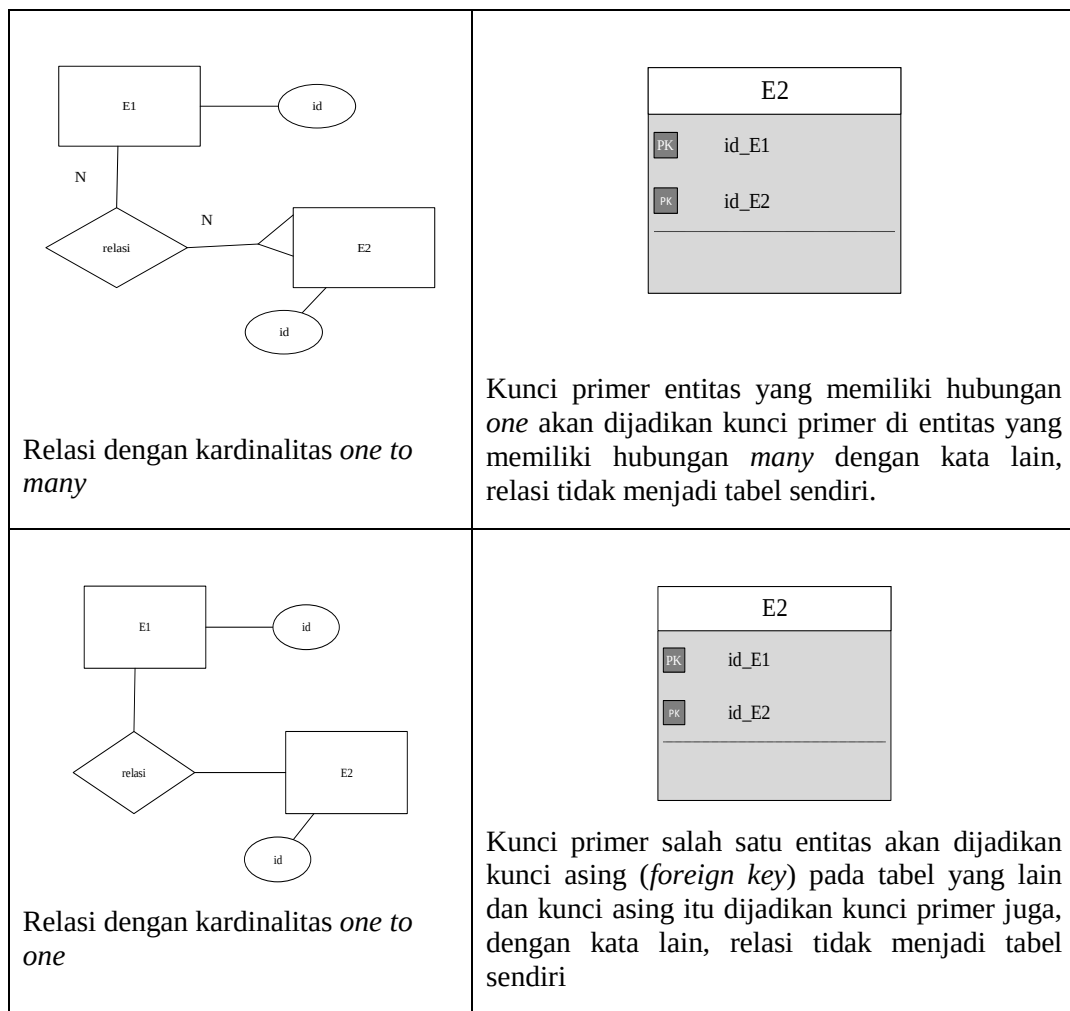
Tabel II.12 Simbol-simbol Aturan *Conceptual Data Model* (CDM)

ERD	CDM
-----	-----

 Entitas	 Menjadi sebuah tabel tersendiri
 Atribut <i>multivalue</i>	 Menjadi sebuah tabel tersendiri dengan kunci primer (primary key) adalah kunci primer pada entitas dan memiliki atribut dengan nama seperti pada atribut entitas

Tabel II.12 Simbol-simbol Aturan *Conceptual Data Model* (CDM) (Lanjutan)

ERD	CDM
 Relasi dengan kardinalitas <i>many to many</i>	 Menjadi sebuah tabel tersendiri dengan kunci primer adalah atribut yang menjadi kunci primer di kedua entitas yang direlasikannya.



(Sumber: Rosa dan Shalahuddin, 2016)

2.9 Kamus Data

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

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

Keuntungan penggunaan kamus data adalah (Sommerville, 2003):

1. Kamus data merupakan mekanisme untuk manajemen nama. Banyak orang yang harus menciptakan nama untuk entitas dan relasi ketika mengembangkan model

sistem yang besar. Nama-nama ini harus dipakai secara konsisten dan tidak boleh bentrok. Kamus data dapat memeriksa keunikan nama dan memberitahu analis persyaratan sekiranya terjadi duplikasi nama.

2. Kamus data sebagai tempat penyimpanan informasi yang dapat menghubungkan analis, desain, implementasi dan evolusi. Sementara sistem dikembangkan, informasi diambil untuk memberitahu perkembangan informasi baru ditambahkan pada sistem. Semua informasi mengenai entitas berada pada satu tempat. Berikut adalah contoh penulisan kamus data:

Spesifikasi tabel pemasok

Nama Tabel : Pemasok

Tipe : File Master

Tabel II.13 Contoh Kamus Data Untuk Tabel Pemasok

No.	Nama Elemen	Akronim	Tipe	Width	Keterangan
1.	ID pemasok	ID_ pemasok	Varchar	10	<i>Primary Key</i>
2.	Nama Pemasok	Nama_Pemasok	char	40	

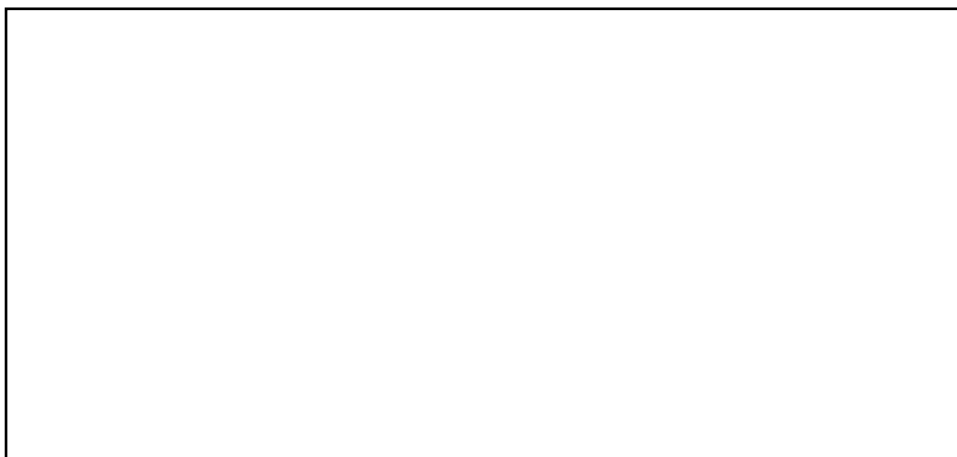
Tabel II.13 Contoh Kamus Data Untuk Tabel Pemasok (lanjutan)

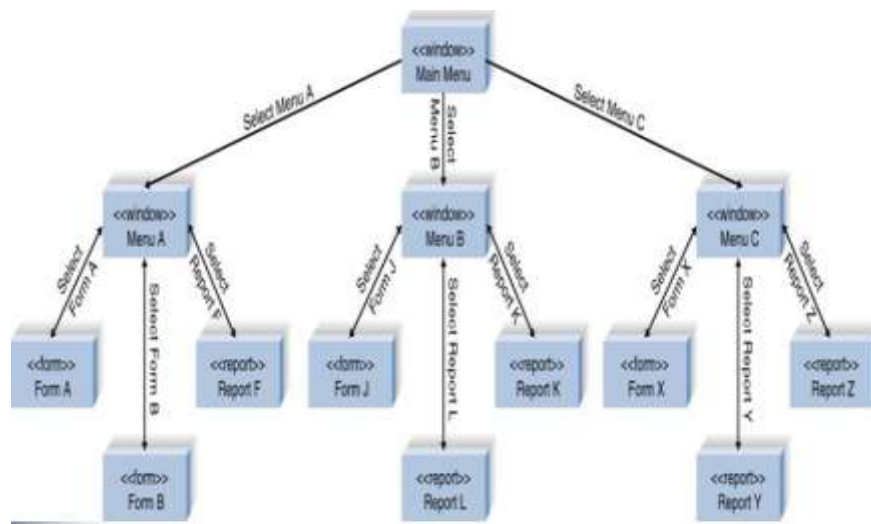
No.	Nama Elemen	Akronim	Tipe	Width	Keterangan
3.	Alamat pemasok	Alamat	Varchar	100	
4.	Nomor telepon	Telepon	Varchar	12	

(Sumber: Jogiyanto, 2005)

2. 10 Windows Navigation Diagram (WND)

Windows Navigation Diagram merupakan *statechart* diagram khusus yang berfokus pada *user interface* (Alan Dennis, 2011) Diagram ini menunjukkan *window-window* tersebut. Sebuah *window* dapat digambarkan sebagai sebuah *state*. *State* ini memiliki nama dan berisi gambar miniature *window*. Transisi antar *state* dipicu oleh ditekannya sebuah tombol yang menghubungkan dua *window*.





Gambar II.6 Contoh *Windows Navigation Diagram*
(Sumber: Alan Dennis, 2010)

2.11 Database

Basis Data terdiri dari 2 kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/ berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya (Fathansyah, 2015). Sebagai satu kesatuan istilah, Basis data (Database) sendiri 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 (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan file/table yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

2.12 Java

Java adalah sebuah bahasa pemrograman yang populer untuk para akademisi dan praktisi komputer. *Java* dikembangkan pertama kali oleh James Gosling dari Sun Microsystems pada tahun 1990-an. *Java* pertama kali dikembangkan untuk memenuhi kebutuhan akan sebuah bahasa komputer yang ditulis satu kali dan dapat dijalankan pada sebagian sistem komputer berbeda tanpa perubahan kode berarti. Kebanyakan bahasa komputer yang ada memiliki keterbatasan migrasi sistem yang berbeda (Irawan, 2007).

Java diciptakan sebagai sebuah bahasa baru dengan implementasi yang berbeda. Bahasa *Java* merupakan bahasa berorientasi objek yang diturunkan dari C++ dengan banyak penyempurnaan. Pada umumnya, para pakar pemrograman berpendapat bahwa bahasa *Java* memiliki konsep yang konsisten dengan teori pemrograman objek dan aman untuk digunakan. Kini universitas-universitas di berbagai negara berpaling dari Pascal atau C++ kemudian memilih *Java* sebagai bahasa untuk belajar pemrograman (Irawan, 2007).

Bahasa pemrograman *Java* memiliki komponen penting (Irawan, 2007):

1. *JVM (Java Virtual Machine)*

Java dapat berjalan pada sebuah sistem operasi membutuhkan *Java Virtual Machine (JVM)*. *JVM* sendiri terdiri dari *Java Runtime Environment (JRE)* dan *Java Development Kit (JDK)*. *Sun Microsystems* mengeluarkan tiga kelas paket *Java*, yaitu *J2-SE JRE* (hanya berisi *JRE*), *J2- SE SDK* (berisi *JDK + JRE*), dan *J2-EE SDK* (berisi *JDK+JRE* dan *tools* untuk aplikasi *enterprise*).

2. *IDE (Integrated Development Environment)*

IDE (Integrated Development Environment) adalah sebuah editor pemrograman sebuah bahasa. Untuk *Java* sendiri ada banyak *IDE* yang tersedia baik yang bersifat gratis (*freeware*) ataupun yang berbayar. Beberapa *IDE* yang populer antara lain, *Jcreator*, *Netbeans*, *Jbuilder*.

3. *Class*

Unit yang paling mendasar dalam pemrograman *java* adalah *class*. *Class* adalah komponen aplikasi yang menangani kode dan data dalam pemrograman *Java*.

Kelebihan *Java* yaitu:

1. Sederhana dan Ampuh

Java dirancang untuk mudah dipelajari, terutama bagi *programmer* yang telah mengenal *C/C++* akan mudah sekali untuk berpindah ke *Java*.

2. Aman

Java dirancang sebagai bahasa pemrograman yang handal dan aman. Aplikasi-aplikasi yang dibangun dengan bahasa *Java* sangat handal dengan manajemen memori yang bagus. Aplikasi *Java* juga dikenal sangat *secure*, yaitu kasus-kasus seperti *buffer overflow* yang umumnya menjadi lubang keamanan aplikasi-aplikasi berbasis *C/C++* tidak terjadi di *Java*, karena pengaturan *security*nya yang bagus.

3. Berorientasi-Objek

Paradigma pemrograman berorientasi objek merupakan paradigma pemrograman masa depan. *Java* merupakan bahasa pemrograman berorientasi objek.

4. Kokoh

Java membatasi anda dari beberapa hal kunci supaya anda dapat menemukan kesalahan lebih cepat saat mengembangkan program. *Java* langsung memeriksa program saat anda menuliskannya, dan sekali lagi ketika program di jalankan.

5. Interaktif

Java memiliki beberapa kemampuan yang memungkinkan program melakukan beberapa hal pada saat bersamaan, tanpa harus kesulitan menangani proses yang akan terjadi selanjutnya.

6. Netral Terhadap Berbagai Arsitektur

Java telah mengambil beberapa keputusan yang sulit dalam pembuatan bahasa *Java* dan bagaimana program dijalankan, jadi anda dapat sepenuhnya percaya “tuliskan sekali, jalan di mana saja, kapan saja, selamanya”.

2.13 Netbeans

NetBeans mengacu pada kedua kerangka *platform* untuk aplikasi *desktop Java*, dan sebuah lingkungan pengembangan terpadu (IDE) untuk pengembangan dengan *Java*, *JavaScript*, *PHP*, *Python*, *Ruby*, *Groovy*, *C*, *C++*, *Scala*, *Clojure*, dan lain-lain (Wikipedia, 2013). *NetBeans IDE* ditulis dalam *Java* dan berjalan di berbagai macam sistem operasi, dimana *JVM* diinstal, termasuk *Windows*, *Mac OS*, *Linux*, dan *Solaris*. Sebuah *JDK* diperlukan untuk pengembangan fungsionalitas *Java*, tetapi tidak diperlukan untuk pembangunan di bahasa pemrograman lain (Wikipedia, 2013).

Platform NetBeans memungkinkan aplikasi untuk dikembangkan dari satu set komponen perangkat lunak modular yang disebut modul. Aplikasi berbasis *platform NetBeans* (termasuk *IDE NetBeans*) dapat diperpanjang oleh pengembang pihak ketiga (Wikipedia, 2013). *Netbeans* terdiri dari beberapa komponen, yaitu:

1. Main Windows

Jendela utama ini bagian dari IDE yang mempunyai fungsi yang sama dengan semua fungsi utama dari program aplikasi *Windows* lainnya. Jendela utama *Netbeans* terbagi menjadi tiga bagian, berupa *Main Menu*, *Toolbar* dan *Component Palette*.

2. Toolbar

Netbeans memiliki beberapa *toolbar* yang masing-masing memiliki perbedaan fungsi dan setiap tombol pada bagian *toolbar* berfungsi sebagai

pengganti suatu menu perintah yang sering digunakan. *Toolbar* sering disebut juga dengan *Speedbar*. *Toolbar* terletak pada bagian bawah baris menu. Pada kondisi default Netbeans memiliki beberapa bagian *toolbar*, antara lain:

- a. *Build Main Project*
- b. *Clean and Build Main Project*
- c. *Run Main Project*
- d. *Debug Main Project*
- e. *Profile Main Project*.

3. *Component Palette*

Component Palette berisi kumpulan ikon yang melambangkan komponen-komponen yang terdapat pada VCL (Visual Component Library), pada *Component Palette* akan ditemukan beberapa *page control*, seperti *Swing Containers*, *Swing Controls*, *Swing Menus*, *Swing Windows*, *AWT*, *Beans*, dan *Java Persistence*.

4. *Form Designer*

Form Designer merupakan suatu objek yang dapat dipakai sebagai tempat untuk merancang program aplikasi. Form berbentuk sebuah meja kerja yang dapat diisi dengan komponen-komponen yang diambil dari *Component Palette*.

5. *Source Editor*

Source Editor merupakan tempat untuk menuliskan kode program. Pada bagian ini dapat dituliskan pernyataan-pernyataan dalam Object Java. Keuntungan bagi pemakai Netbeans adalah bahwa tidak perlu menuliskan kode-kode sumber, karena Netbeans telah menyediakan kerangka penulisan sebuah program.

6. *Object Inspector*

Object Inspector digunakan untuk mengubah properti atau karakteristik dari sebuah komponen. *Object Inspector* terdiri dari dua tab, yaitu *Properties* dan *Events*.

2.14 MySQL

My Structured Query Language (MySQL) adalah *Relational Database Management System* (RDBMS) yang sangat cepat dan kuat. Sebuah basis data

memungkinkan *user* untuk melakukan penyimpanan yang efisien, pencarian dan pengurutan data. MySQL *server* memiliki kemampuan melakukan kontrol akses terhadap data untuk memastikan bahwa setiap *user* dapat bekerja dengan sesuai, menyediakan akses yang cepat, dan meyakinkan bahwa hanya *user* yang mempunyai hak akses yang dapat mengaksesnya. Oleh karena itu, MySQL merupakan *database server* yang yang dapat digunakan banyak *user* dan banyak pekerjaan (Welling dan Thomson, 2003).

MySQL adalah salah satu *database server* yang menggunakan SQL. SQL (*Structured Query Language*) adalah bahasa pertanyaan (*query language*) yang distandarisasi untuk menanyakan informasi dari sebuah basis data (Welling dan Thomson, 2003). MySQL mendukung banyak tipe data yang dapat disimpan pada sebuah kolom. Terdapat tiga kategori tipe data yang didukung oleh MySQL, yaitu tipe data numerik, string, serta penganggalan dan waktu. Sebuah data yang akan disimpan harus sesuai dengan tipe data yang bersangkutan.

2.14.1 Keunggulan MySQL

Berikut keunggulan dari MySQL, di antaranya adalah (Sutaji, 2012):

1. *Portability*
Dapat berjalan stabil pada berbagai sistem operasi, di antaranya: Windows, Linux, FreeBSD, MacOS X Server, Solaris, Asigma.
2. *Open source*
Didistribusikan secara gratis di bawah lisensi dari *General Public License* (GPL), dimana setiap orang bebas untuk menggunakannya tetapi tidak boleh untuk dijadikan program induk turunan bersifat *close source* (komersial).
3. *Multi User*
Dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan.
4. *Performance tuning*
Memiliki kecepatan yang tinggi dalam menangani *query*.
5. *Column types*
Memiliki tipe data yang sangat kompleks, seperti: *signed/unsigned integer*, *float*, *double*, *char*, *varchar*, *text*, *blob*, *date*, *time*, *datetime*, *timestamp*, *year* dan *enum*.

6. *Command* dan *function*

Memiliki operator dan fungsi penuh yang mendukung *select* dan *where* dalam *query*.

7. *Security*

Memiliki beberapa lapisan sekuritas seperti tingkat *subnet mask*, *hostname*, *privilege user* dengan sistem perijinan yang mendetail serta *password* yang terenkripsi.

8. *Scalability* dan *limits*

Mampu menangani basis data dalam jumlah besar, dengan jumlah *field* lebih dari 50 juta, 60 ribu tabel, dan 5 miliar *record*. Batas indeks mencapai 32 buah per tabel.

9. *Localization*

Dapat mendeteksi pesan kesalahan (*error code*) pada klien dengan menggunakan lebih dari 20 bahasa.

10. *Connectivity*

Dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, *Unix Socket*, *Named Pipes*.

11. *Interface*

Memiliki antarmuka terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan API.

12. *Client* dan *tools*

Dilengkapi dengan berbagai *tool* yang dapat digunakan untuk administrasi basis data sekaligus dokumen petunjuk *online*.

13. Struktur tabel

Memiliki struktur tabel yang lebih fleksibel dalam menangani *alter* tabel dibandingkan dengan PostgreSQL dan Oracle.

2.14.2 Tipe Data MySQL

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 *table*. Beberapa jenis data yang tersedia pada MySQL dapat dilihat pada Tabel II.14 (Sutaji, 2012).

Tabel II.14 Jenis Data pada MySQL

Jenis Data	Keterangan
CHAR	Sebuah <i>string</i> dengan panjang tetap. Sisa Jumlah karakter yang belum terisi akan diisi dengan spasi, akan tetapi spasi ini dibuang jika data dipanggil. Jangkauan nilai M adalah 1-255 karakter
VARCHAR	<i>String</i> dengan panjang berupa variabel. M bisa mencapai 65535
DATE	Data berupa tanggal. Format tanggal dalam bentuk 'YYYY-MM-DD'

Tabel II.14 Jenis Data pada MySQL (Lanjutan)

Jenis Data	Keterangan
ENUM	Sebuah <i>enumeration</i> . Sebuah obyek string yang hanya boleh memiliki satu nilai, yang terambil dari 'value1','value2','...',NULL atau nilai spesial "" <i>error</i> . Sebuah <i>enum</i> dapat menampung 65535
TIME	Data berupa waktu. Format waktu dalam bentuk 'HH:MM:SS'
TINYINT	Bilangan antara -128 sampai dengan +127
SMALLINT	Bilangan antara -32768 sampai dengan +32767
INT	Bilangan antara -2147683648 sampai dengan +2147683647
FLOAT	Bilangan <i>floating point</i> yang kecil (presisi tunggal). Jangkauan nilai yang diizinkan adalah -3.402823466E+38 s/d -1.175494351e-38,0.
DOUBLE	Bilangan <i>floating point</i> dengan ukuran normal (presisi ganda). Jangkauan nilai yang diizinkan adalah -1.7976931348623157E+308 s/d -2.225073858507201E-308,0 dan 2.225073858507201E-308 s/d 1.7976931348623157E+308
TEXT, BLOB	Sebuah TEXT atau BLOB dengan panjang karakter maksimum 65535 karakter

(Sumber: Sutaji, 2012)

2.15. XAMPP

XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolahan data MySQL di komputer lokal. XAMPP berperan sebagai *web server* pada komputer. XAMPP juga dapat disebut sebagai sebuah *CPanel server virtual*, yang dapat membantu melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses internet (Kadir, 2009). Fungsi lainnya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL *database*, dan penerjemah bahasa yang ditulis

dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X, *Apache*, MySQL, PHP dan Perl. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis.

Daftar

2. 1	Pengertian Rancang Bangun	25
2. 2	Konsep Dasar Sistem Informasi.....	25
2.2.1	Pengertian Sistem.....	26
2.2.2	Pengertian Informasi	27
2.2.3	Pengertian Sistem Informasi.....	28
2.2.4	Komponen Sistem Informasi.....	28
2. 3	Pengertian Sistem Informasi Penerimaan dan Pengiriman Bahan Baku atau Komponen 31	
2.3.1	Pengertian Penerimaan	31
2.3.2	Pengertian Pengiriman	32
2.3.3	Pengertian Bahan Baku atau Komponen.....	33
2. 4	Pengembangan Sistem	33
2.4.1	Model Prototipe Secara Umum	36
2.4.2	<i>Evolutionary Prototype</i>	37
2. 5	<i>Flowchart</i>	39
2. 6	<i>Unified Modelling Language (UML)</i>	41
2.6.1	<i>Use Case Diagram</i>	42
2.6.2	<i>Activity Diagram</i>	44
2.6.3	<i>Sequence Diagram</i>	46
2.6.4	<i>Class Diagram</i>	47
2.6.5	<i>Deployment Diagram</i>	48
2. 7	<i>Entity Relationship Diagram (ERD)</i>	49
2. 8	<i>Conceptual Data Model (CDM)</i>	50
2. 9	Kamus Data	52
2. 10	<i>Windows Navigation Diagram (WND)</i>	53
2. 11	<i>Database</i>	55
2. 12	<i>Java</i>	55
2. 13	<i>Netbeans</i>	57
2. 14	<i>MySQL</i>	58

BAB III

METODOLOGI PENELITIAN

3.6 Metodologi Penelitian

Metodologi Penelitian merupakan cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu (Sugiyono, 2013). Metodologi penelitian berupa tata cara bagaimana suatu penelitian dilaksanakan atau cara yang ditempuh sehubungan dengan penelitian yang dilakukan, dan memilih langkah-langkah sistematis. Metodologi penelitian mempelajari cara-cara melakukan pengamatan dengan pemikiran yang tepat secara terpadu melalui tahapan-tahapan yang disusun secara ilmiah untuk mencari, menyusun, serta menganalisis dan menyimpulkan data-data berdasarkan fakta-fakta secara ilmiah. Untuk menghasilkan penelitian tugas akhir yang lebih lengkap diperlukan suatu metode dalam penelitian yang telah dipersiapkan sesuai dengan masalah yang akan dibahas.

3.7 Jenis dan Sumber Data

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

1. Data primer

Data yang diperoleh melalui pengamatan langsung pada PT Krama Yudha Ratu Motor di bagian *Part Control* penerimaan dan pengiriman komponen terhadap sistem yang sedang berjalan dan wawancara dengan pegawai sebagai sumber informasinya. Dalam penelitian ini tidak terdapat data primer.

2. Data Sekunder

Data yang diperoleh dari data yang tersedia dan telah terlebih dahulu dikumpulkan dan dilaporkan oleh orang lain, buku-buku dan kajian ilmiah dari berbagai sumber yang berkaitan dengan penelitian. Data tersebut adalah data alur proses penerimaan komponen sampai pengiriman suplemen komponen serta dokumen yang digunakan dalam penelitian ini adalah surat komponen masuk (*delivery note*), surat pengiriman komponen, dan laporan penerimaan pengiriman komponen, data umum perusahaan, profil perusahaan dan struktur organisasi perusahaan.

3.8 Metode Pengumpulan Data

Menurut Sugiyono (2013) teknik pengumpulan data merupakan langkah yang paling strategis dalam penelitian, karena tujuan utama dari penelitian adalah mendapatkan data. Penelitian ini menggunakan beberapa teknik pengumpulan data, yaitu:

1. Studi Lapangan

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

- a. Pengamatan, tahap ini dilakukan secara langsung di Bagian *Part Control*, Bagian admin *part control* PT Krama Yudha Ratu Motor dengan mengamati prosedur Penerimaan dan Pengiriman Komponen yang sedang berjalan mulai dari penerimaan komponen dari *vendor* sampai pengiriman suplemen komponen. Melalui teknik ini, data yang dibutuhkan diamati, dikumpulkan dan diolah sebagai bahan dalam penelitian.
- b. Wawancara, yaitu mencari data yang dibutuhkan secara langsung dengan memberikan pertanyaan tentang proses bisnis pembelian bahan baku yang diperlukan pada penyusunan Tugas Akhir ini. Wawancara ini dilakukan kepada karyawan pada Bagian *Part Control*, Bagian *Office*, Admin, dan *Foreman*.

2. Studi Pustaka

Tahap ini merupakan tahap awal pengumpulan data dari sumber-sumber seperti buku yang berhubungan dengan masalah yang sedang dianalisis. Studi pustaka

yang dilakukan berkaitan dengan analisis dan desain sistem informasi, pemrograman Java, pemrograman MySQL, sistem informasi produksi, pengembangan perangkat lunak, dan juga beberapa contoh *coding* program melalui media *internet*. Referensi yang diperoleh, kemudian dikaji sebagai dasar dalam menyelesaikan penelitian.

3.9 Metode Pendekatan dan Pengembangan Sistem

Metode pendekatan analisis dan desain yang digunakan yaitu perancangan berorientasi objek yang merupakan alat bantu dalam *Object Oriented Analysis and Design*. Adapun metode pengembangan perangkat lunak yang digunakan adalah *prototype evolutioner*.

3.4.3 Metode Pendekatan Sistem

Metode pendekatan yang digunakan adalah metode pendekatan berorientasi objek (*Object Oriented*). Seiring dengan berkembangnya pemrograman berbasis objek maka analisis dan desain sistem juga dapat menggunakan konsep objek. Pendekatan baru untuk pengembangan sistem ini sering disebut juga sebagai pendekatan ketiga, setelah pendekatan yang berorientasi data dan berorientasi proses. *Object Oriented Analysis and Design* adalah metode pendekatan sistem yang lebih menekankan objek dibandingkan dengan data atau proses. Ada beberapa ciri khas dari pendekatan ini, yaitu *object*, *inheritance* dan *object class*.

Object class adalah sekumpulan objek yang berbagi struktur yang sama dan perilaku yang sama. *Inheritance* merupakan properti yang muncul ketika tipe entitas atau *object class* disusun secara hierarki dan setiap tipe entitas atau *object class* menerima atau mewarisi atribut dan metode dari pendahulunya.

3.4.4 Metode Pengembangan Sistem

Metode pengembangan aplikasi Penerimaan dan Pengiriman Komponen ini menggunakan metode model *prototype* jenis *evolutionary*. Model *prototype* cocok digunakan untuk menjabarkan kebutuhan *user* secara lebih detail karena *user* sering kali kesulitan menyampaikan kebutuhannya secara *detail* tanpa melihat gambaran yang jelas (Rosa dan Shalahuddin, 2011).

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

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

3.10 Kerangka Penelitian

Dalam penelitian yang dibahas dalam tugas akhir ini, dilakukan langkah-langkah atau tahapan dalam penelitian yang ada pada Gambar III.1 sesuai dengan metodologi penelitian yang digunakan dalam penelitian tersebut. Penjelasan langkah-langkah atau tahapan dalam penelitian tersebut adalah sebagai berikut:

1. Studi Pendahuluan

Tujuan melakukan studi pendahuluan adalah untuk mendapatkan pengetahuan umum mengenai sistem yang sedang diteliti. Pada tahap ini penulis mengumpulkan data yang diperlukan dengan cara observasi langsung terhadap sistem yang berjalan, serta wawancara dengan pihak yang mempunyai hubungan dan pengetahuan terhadap sistem yang berjalan. Data yang dikumpulkan berkaitan dengan kebutuhan *user* terhadap sistem yang akan dikembangkan, seperti alur dokumen sistem yang sedang berjalan, arus informasi yang terdapat dalam sistem yang sedang berjalan serta dokumen-dokumen yang digunakan pada sistem yang sedang berjalan.

2. Identifikasi dan perumusan masalah

Tahap ini penulis mengidentifikasi dan menetapkan permasalahan yang terjadi selama penelitian pada sistem penerimaan dan pengiriman komponen yang sedang berjalan pada PT Krama Yudha Ratu Motor.

3. Studi Literatur

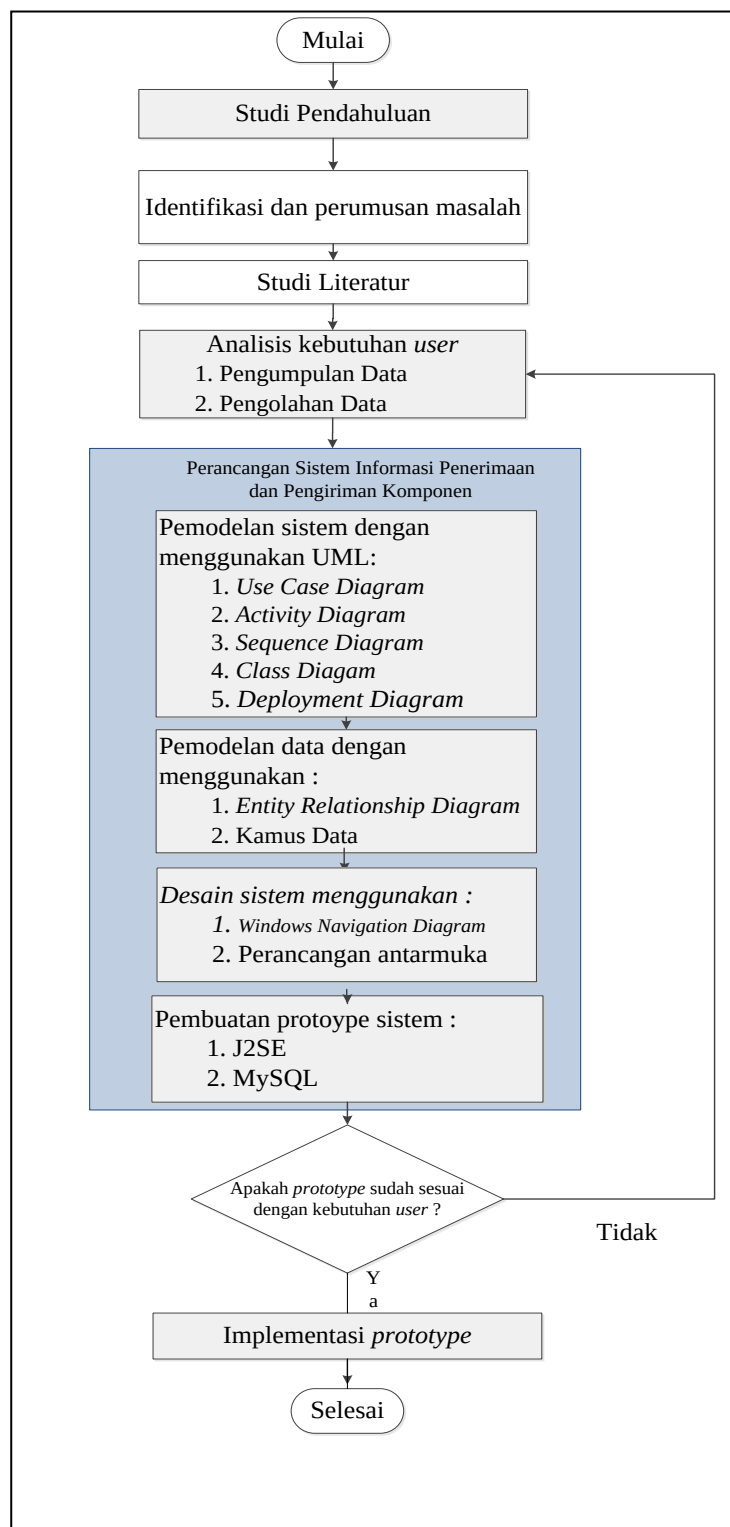
Penulis melakukan studi literatur yang lebih mendalam dengan tujuan untuk mendapatkan landasan teori yang dapat mendukung pemecahan masalah yang sedang diteliti. Teori yang didapatkan merupakan langkah awal peneliti agar lebih memahami permasalahan yang sedang diteliti dengan benar sesuai dengan kerangka berpikir ilmiah.

4. Perancangan Sistem Informasi Penerimaan dan Penerimaan Komponen
Kegiatan analisis untuk mengembangkan sistem informasi dilakukan melalui tahapan-tahapan sebagai berikut:
 - a. Analisis kebutuhan *user*
Melakukan analisis kembali terhadap data yang telah didapat pada penelitian sebelumnya, sehingga dapat diketahui kebutuhan *user* terhadap aplikasi yang akan dibuat.
 - b. Pemodelan sistem
Membuat pemodelan sistem berbasis objek dengan menggunakan *Unified Modeling Language (UML) Diagram*, yaitu:
 - a. *Use Case Diagram*
 - b. *Activity Diagram*
 - c. *Sequence Diagram*
 - d. *Class Diagram*
 - e. *Deployment Diagram*
 - c. Pemodelan data
Membuat desain pemodelan data dengan menggunakan *Entity Relationship Diagram* dan Kamus Data.
 - d. Desain sistem
Membuat desain sistem dengan *Windows Navigation Diagram* dan rancangan antar muka.
5. Pembuatan *prototype* sistem
Tahap selanjutnya adalah membuat perancangan *prototype* sistem dan pengkodean sistem. *Prototype* yang dibuat adalah :
 - a. Membuat aplikasi J2SE dengan menggunakan *tools* Netbeans.
 - b. Membuat *database* dengan perangkat lunak basis data MySQL.
6. Menyesuaikan *Prototype* dengan Keinginan *User*

Pada tahap ini, *prototype* yang diusulkan akan diperlihatkan dan diuji coba terlebih dahulu kepada *user* apakah sudah sesuai dengan keinginan *user* atau belum. Jika belum maka dilakukan analisis kembali, tetapi apabila sesuai maka tahap selanjutnya adalah implementasi penggunaan *prototype* tersebut.

7. Implementasi Sistem

Sebelum melakukan implementasi sistem usulan, ada 2 (dua) hal yang harus dipenuhi yaitu perangkat keras yang akan digunakan seperti laptop atau *Personal Computer* (PC), dan perangkat lunak seperti *Java Development Kit* dan *MySQL Service*.



Gambar III.1 Kerangka Penelitian

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Sekilas Perusahaan

PT. Krama Yudha Ratu Motor (PT. KRM) merupakan sebuah perusahaan manufaktur yang bergerak di bidang otomotif yang memiliki spesialisasi merakit berbagai macam kendaraan niaga Mitsubishi di Indonesia. PT KRM didirikan pada tanggal 1 Juni 1973 sebagai perusahaan swasta dengan awalnya 100% modalnya merupakan penanaman modal dalam negeri (PMDN), sejak tahun 2012 status menjadi PMA (Penanaman Modal Asing). Produksi komersial PT. Krama Yudha Ratu Motor mulai tahun 1975 dengan produk yaitu kendaraan niaga CJM, L300, ZC, TD/FM (Fuso), TD S/W (*cold diesel /canter*). PT KRM sudah mendapatkan sertifikat ISO 9001 sejak tahun 2002 untuk sistem prosedur yang digunakan dan mempunyai dokumen proses bisnis yaitu SSP (Standar Sistem Prosedur), manual mutu, dan ISO 14001-1996 sejak tahun 2003, namun secara komersial perusahaan ini berdiri pada tahun 1975 berdasarkan legalitas administrasi sebagai berikut:

1. Akte Notaris No. 16 Tanggal 15 April 1973.
2. Pengadilan Negeri Jakarta No. 1374 Tanggal 18 Maret 1974.
3. Departemen Perindustrian RI no. 27/BP/A/6D/1974 Tanggal 20 Maret 1974.
4. Departemen Kehakiman RI No. WA/5/108/28/1974 Tanggal 7 November 1974.
5. BKPM No. B.92/A/BKPM/XI/1973 Tanggal 7 November 1974.

Didasari ISO 9001:2008 dan ISO 14001:2004 dengan *subject* utama perakitan/*assembling* kendaraan bermotor roda 4 dan atau lebih melakukan pengendalian sistem manajemen mutu seperti penerimaan barang, penyimpanan, distribusi ke *line* produksi untuk selanjutnya melakukan perakitan pengelasan di bagian *welding*, pengecatan di bagian *painting* dan perakitan atau pemasangan *parts* di bagian *trimming*. Seluruh proses produksi dikendalikan dengan inspeksi ketat, baik dari seluruh pelaku proses produksi maupun dari bagian *inspector*. Ruang lingkup penerapan adalah dari penerimaan *part* proses perakitan sampai dengan *delivery* ke pelanggan dan diterapkan di seluruh Departemen, dan ruang lingkup sistem manajemen lingkungan adalah seluruh area perusahaan.

PT Krama Yudha Ratu Motor merupakan perusahaan yang patut diacungi jempol, karena perusahaan ini sudah mendapat beberapa sertifikat dan prestasi yang sudah diraih. Namun semua itu tidak lepas dari komitmen semua karyawan dalam mencapai visi misi PT Krama Yudha Ratu Motor dalam mencapai suatu tujuan.



PT. KRM ini merupakan dari bagian Krama Yudha Mitsubishi Group (KYMG). Awal berdirinya KYMG adalah akibat dari banyaknya kendaraan bermotor dari Eropa yang diimpor ke Indonesia untuk mengurangi pengimporan kendaraan tersebut maka para pengusaha melakukan pertemuan dan sepakat mendirikan suatu perakitan kendaraan bermotor di Indonesia dengan menggunakan lisensi dari Mitsubishi Motor Corporation Jepang. PT. Krama Yudha ini juga memiliki anak perusahaan di beberapa tempat, yaitu PT. Krama Yudha Ratu Motor (KRM), PT. Mitsubishi Krama Yudha Motor dan Manufacturing (MKM) I dan II, PT. Krama Yudha Tiga Berlian (KTB). PT. KRM merupakan *assembler* atau perakitan kendaraan bermotor *merk* Mitsubishi sehingga tidak memiliki desain dan pengembangan karena seluruh desain dan pengembangan merupakan hak agen tunggal pemegang merk (ATPM).

4.2 Profil Perusahaan

PT. KRM memiliki profil perusahaan sebagai berikut:

Nama Perusahaan	: PT Krama Yudha Ratu Motor
Tanggal berdiri	: 01 Juni 1973
Jumlah karyawan	: 1,063 (April 2017)
Alamat	: Jl. Raya Bekasi Km. 21-22 Cakung, Jakarta - Indonesia
Telp	: 4602905
Pekerja	

<i>Direct</i>	: 661 Orang
<i>Indirect</i>	: 295 Orang
<i>General</i>	: 220 Orang
Total	: 1176 Orang



Gambar IV.2 Logo PT Krama Yudha Ratu Motor
(Sumber : PT Krama Yudha Ratu Motor, 2017)



Gambar IV.3 Pabrik PT Krama Yudha Ratu Motor
(Sumber : PT Krama Yudha Ratu Motor, 2017)

4.3 Visi dan Misi Perusahaan

Visi adalah sesuatu yang ingin dicapai perusahaan atau organisasi di masa yang akan datang. Visi terkonsentrasi ke masa depan (jangka panjang) dan cenderung merupakan pernyataan yang sifatnya strategis. Jadi visi secara garis besar adalah arah dan pandangan suatu perusahaan untuk memberikan pelayanan, bekerjasama dengan instansi pemerintah maupun swasta, dan membantu konsumen dalam memenuhi permintaannya. Visi PT Krama Yudha Ratu Motor (PT. KRM) “**Menjadi Perusahaan terunggul dan sanggup bersaing di tingkat regional maupun global.**”

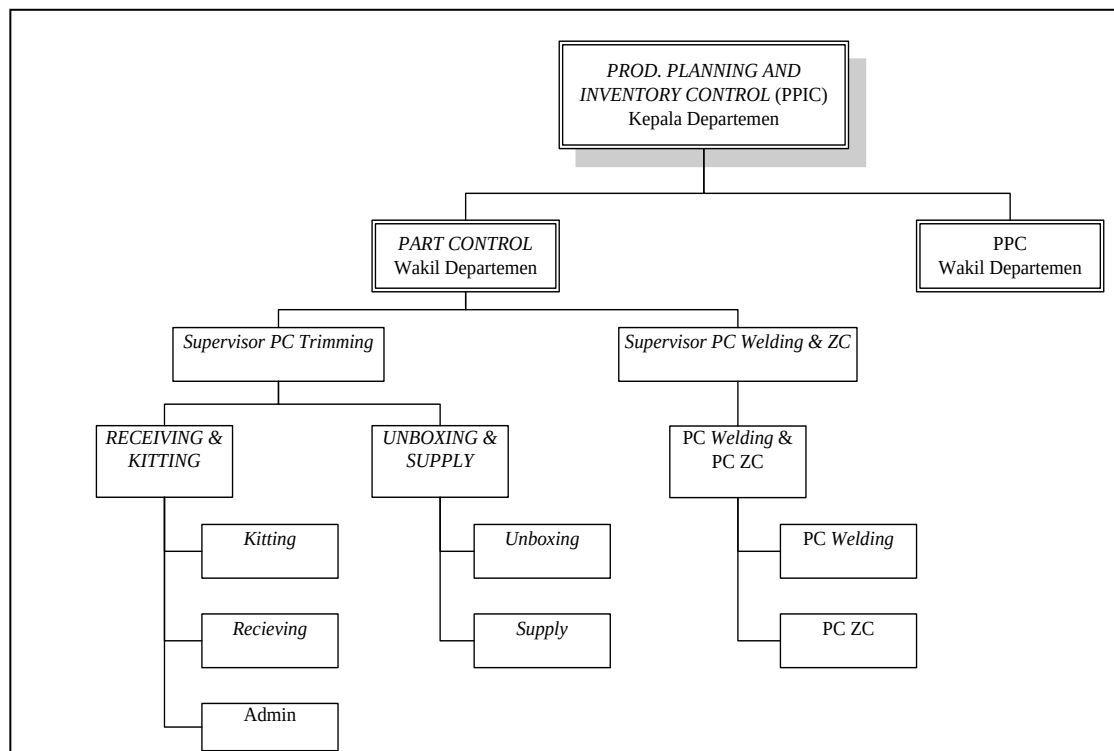
Misi adalah apa yang sedang (akan) dilakukan atau ingin dicapai dalam waktu (sangat) dekat atau saat ini. Misi lebih terkonsentrasi ke saat ini dan merupakan target-target yang sifatnya lebih operasional yang mungkin dikaitkan dengan konsumen, proses-proses dalam perusahaan atau organisasi, serta tingkat kinerja yang diinginkan.

Jadi misi secara garis besar adalah sasaran atau tujuan dari suatu perusahaan yang ingin dicapai agar perusahaan dapat bersaing dengan perusahaan lain, bertahan dan mampu berkembang lebih maju dalam memuaskan konsumen. Berikut adalah misi PT Krama Yudha Ratu Motor:

1. Menjadi perusahaan yang cukup memuaskan *shareholder* dan kesejahteraan yang baik bagi seluruh karyawan.
2. Membangun sumber daya manusia yang handal dan sanggup mengantisipasi perkembangan regional maupun global.
3. Komitmen akan industri yang ramah lingkungan.

4.4 Struktur organisasi *Part Control* PT. KRM

Struktur organisasi adalah proses untuk merancang struktur formal, untuk menunjukkan adanya pembagian kerja dan menunjukkan bagaimana fungsi-fungsi tersebut terintegrasi dengan baik agar tujuan suatu organisasi dapat dicapai dengan efektif dan efisien. Adapun struktur organisasi PC PT Krama Yudha Ratu Motor:



Gambar IV.4 Struktur Organisasi bagian PC PT. KRM
(Sumber : PT Krama Yudha Ratu Motor, 2017)

Berikut adalah uraian tugas dari masing-masing jabatan yang tercantum pada struktur organisasi di atas:

1. *Production Planning Control And Inventory Control*

- a. Membuat rencana kerja dan melaksanakan pengawasan serta pengendalian pada bagian PPC, *Part Control & Information Technology* (IT).
- b. Memberdayakan dan memonitor kegiatan proses produksi dengan menggunakan tenaga kerja yang cukup sesuai dengan kondisi alat dan mesin agar mencapai produktivitas sesuai standar.
- c. Mengevaluasi dan mengembangkan Sumber Daya Manusia (SDM) secara terpadu untuk mencapai sasaran yang telah ditentukan.
- d. Menjalankan, memelihara, meningkatkan sistem manajemen perusahaan.
- e. Melakukan identifikasi, mengelolah dan memonitoring lingkungan ditempat kerja masing-masing.

2. Wakil Departemen

- a. Membuat *order* dan memonitor penerimaan bahan bantu untuk kebutuhan produksi.
- b. Mengawasi dan mengendalikan penerimaan, penyimpanan, pemeliharaan dan pendistribusian *parts* ke *line* produksi
- c. Berkoordinasi dengan bagian terkait dalam pengaturan dan perubahan jadwal produksi apabila terjadi keterlambatan pengadaan *parts* dan bahan bantu.
- d. Memberikan *feedback* kepada *vendor* maupun distributor atas terjadinya keterlambatan penerimaan *part*.
- e. Mengendalikan *Line Damage* dan membuat *order* pengadaan *part* penggantinya.
- f. Bertanggung jawab atas penyimpanan dan penggunaan kunci gudang.

3. *Supervisor PC Trimming*

- a. Mengontrol barang masuk dari *supplier* untuk keperluan produksi lini *Trimming*
- b. Memeriksa jadwal ketepatan barang masuk
- c. Melakukan identifikasi, mengelolah dan memonitoring lingkungan ditempat kerja masing-masing.

4. *Supervisor PC Welding & ZC*

- a. Mengontrol barang masuk dari *supplier* untuk keperluan produksi lini

Welding dan lini produksi mobil *sport* tipe ZC

- b. Memeriksa jadwal ketepatan barang masuk
 - c. Melakukan identifikasi, mengelola dan memonitoring lingkungan ditempat kerja masing-masing.
5. *Receiving & Kitting*
- a. Menerima bahan baku dan mengecek jumlah barang masuk sesuai dengan permintaan.
 - b. Mengelompokkan setiap *part* kedalam urutan lini produksi
 - c. Menyusun *part* kedalam *box kitting*
6. *Unboxing & Supply*
- a. Membuka dan mengecek *part* yang berasal dari *supplier* lokal maupun CKD
 - b. Memasok *part* yang telah disusun ke lini produksi
7. *PC Welding & PC ZC*
- a. Menerima bahan baku dan mengecek jumlah barang masuk sesuai dengan permintaan.
 - b. Memasok *part* ke lini *Welding*
8. Admin
- a. Membuat laporan penerimaan *part* harian dan bulanan

4.5 Produk PT. KRM

PT. Krama Yudha Ratu Motor mulai merakit produk Mitsubishi sejak tahun 1975 hingga saat ini dengan produk komersil 5 model mobil yaitu Fuso (FM/FN), *Colt Diesel* (TD), *Colt L-300* (SL), *Colt T120 SS* (CJM), sedangkan tipe jenis kendaraan *passenger* yaitu *Outlander Sports* (ZC).



Gambar IV.5 Produk PT. KRM
(Sumber: PT. Krama Yudha Ratu Motor, 2016)

4.6 Komponen

Pada PT Krama Yudha Ratu Motor, komponen merupakan unsur yang berhubungan langsung terhadap barang yang akan diproduksi. Komponen juga dapat disebut dengan bahan baku. Begitu pentingnya bahan baku yang diperlukan oleh bagian produksi membuat perusahaan harus cepat dan tepat dalam memenuhi ketersediaannya. Jika ketersediaan komponen ini tidak berjalan secara optimal, maka kegiatan produksi akan terhambat atau mungkin tidak berjalan. Oleh sebab itu, maka perusahaan harus memiliki strategi untuk memenuhi ketersediaan komponen yang dibutuhkan oleh bagian produksi.

PT KRM merupakan sebuah perusahaan otomotif manufaktur yang memproduksi kendaraan niaga Mitsubishi di Indonesia. Untuk memproduksi sebuah kendaraan tentunya diperlukan komponen sebagai unsur dalam proses produksinya. Dalam pengadaannya, komponen ini terbagi menjadi dua jenis, yaitu komponen lokal dan komponen CKD (*Completely Knock Down*). Komponen

yang berasal dari *vendor* dalam negeri disebut lokal sedangkan komponen yang harus dibeli dari *vendor* dari luar negeri disebut CKD. Karena komponen sangat diperlukan untuk memproduksi pesanan dari KTB, maka PT KRM harus memenuhi kebutuhan komponen dengan optimal. Optimal yang dimaksud adalah PT KRM harus mampu mengelola komponen sesuai dengan kebutuhan. Dalam mencapai target produksi bulanan, komponen yang digunakan oleh PT KRM berjumlah 3.440.378 komponen.

4.7 Vendor Perusahaan

Dalam upaya memenuhi kebutuhan dari pelanggan PT. Krama Yudha Ratu Motor senantiasa mencari bahan baku yang terbaik untuk memuaskan pelanggan, ada standarisasi dalam memilih bahan baku tersebut dan berasal dari *supplier* lokal maupun *supplier* luar (CKD). Berikut adalah beberapa perusahaan yang menjalin hubungan kerjasama dengan PT. Krama Yudha Ratu Motor dalam pembelian bahan baku (*Parts*):

Tabel IV.1 Data *Vendor* PT. KRM

No	Nama <i>Vendor</i>	Nama <i>Part</i>
1.	Asahimas <i>Flat Glass</i> ,Tbk	<i>Glass Windshield (Green)</i>
		<i>Glass Fr Door Window Lh (Green)</i>
		<i>Glass Fr Door Window Rh (Green)</i>
2.	Braja Mukti Cakra	Hub, Fr <i>Wheel</i>
		<i>Knuckle</i>
3.	Caturindo Agungjaya <i>Rubber</i>	<i>Pad Door Glass</i>
		<i>W/Strip Fr Door Open Lh</i>
		<i>W/Strip Fr Door Open Rh</i>
		<i>Duct, Rear Vent</i>
4.	Cipta Nissin <i>Industries</i>	<i>Converter Assy, Catalytic</i>
5.	Denso Sales Indonesia	<i>Radiator Assy (Al)</i>
		<i>Horn,Assy Low Sld</i>
		<i>Air Filter L300</i>
		<i>Condenser Tank Assy</i>

(Sumber: PT. Krama Yudha Ratu Motor, 2017)

4.8 Man Power Part Control PT KRM

Bagian PPIC bertugas dalam penerimaan komponen dimana orang-orang yang terlibat di dalam proses *Part Control* yaitu:

Tabel IV.2 *Man Power* PC PT. KRM

No.	Jabatan	Tugas	Jumlah
1	<i>Supervisor</i>	Mengawasi proses <i>part control</i> agar berjalan sesuai dengan fungsinya.	1 orang
2	Asisten <i>Supervisor</i>	Membantu <i>supervisor</i> dalam menjalankan tugasnya.	1 orang
3	<i>Foreman</i>	Kepala bagian yang mengatur kegiatan <i>part control</i>	3 orang
4	Asisten Foreman	Membantu <i>foreman</i> dalam menjalankan tugasnya.	12 orang
5	<i>Receiver</i>	Mengangkut barang masuk dengan menggunakan <i>forklift</i> dan <i>reach lift</i> .	18 orang
6	<i>Checker</i>	Mengecek komponen masuk dari <i>vendor</i> dan mengecek kelengkapannya.	1 orang

Tabel IV.2 *Man Power* PC PT. KRM (Lanjutan)

No.	Jabatan	Tugas	Jumlah
7	Admin	Mendata komponen masuk dan membuat laporan.	2 orang
8	<i>Man power</i>	Pekerja yang tidak memiliki nama khusus seperti yang bertugas untuk membuat <i>box kitting</i> , dan mendistribusikan komponen ke lini produksi dan sebagainya.	72 orang

(Sumber: PT. Krama Yudha Ratu Motor, 2017)

4.9 Alat Pendukung Produksi

Untuk melancarkan proses produksi tentunya butuh alat-alat yang dapat membantu kinerja produksi. Alat ini digunakan untuk memindahkan bahan baku dari gudang menuju tempat produksi. Dengan adanya alat ini karyawan tidak perlu lagi mengangkat bahan baku yang akan diproduksi secara manual, dengan begitu proses produksi bahan baku ini akan lebih mudah dan cepat. Berikut merupakan alat-alat pendukung yang digunakan dalam proses pengangkutan barang.

1. *Towing*

Towing merupakan alat pengangkut semacam mobil kecil biasanya digunakan untuk mengangkut *part-part* yang lumayan besar dan jumlah yang banyak.



Gambar IV.6 Kendaraan *Towing*
(Sumber : PT Krama Yudha Ratu Motor, 2017)

2. *Dolly*

Merupakan alat pengangkut yang menyerupai rak. *Dolly* merupakan alat pendukung yang baru digunakan di PT. KRM. Muatan yang dapat dibawa untuk 1 (satu) unit mobil jadi.



Gambar IV.7 Alat angkut *Dolly*
(Sumber : PT Krama Yudha Ratu Motor, 2017)

3. *Box Kitting*

Merupakan wadah untuk membawa *part-part* dengan lebih mudah tanpa harus takut tertinggal di *warehouse* karena sudah ditempatkan dalam satu wadah.



Gambar IV.8 *Box kitting*
(Sumber : PT Krama Yudha Ratu Motor, 2017)

4. 10 Dokumen Penerimaan Komponen

Proses penerimaan komponen merupakan kegiatan untuk mengendalikan komponen masuk dan keluar serta menyediakan komponen sesuai dengan kebutuhan proses produksi. Kegiatan ini sangat penting dan berpengaruh terhadap kelancaran proses produksi. Dokumen yang terlibat dalam penerimaan komponen adalah sebagai berikut:

1. *Daily Schedule*

Daily Schedule adalah dokumen yang berisi target produksi harian PT. KRM, sehingga bagian *Part Control* dapat mengendalikan serta menyediakan *part* sesuai kebutuhan produksi.



PT KRAMAYUDHA RATU MOTOR

Departemen PPC-PPC

Kode Dokumen : PRM-PPC-7.1-001

Tanggal Revisi : 21 JANUARI 2012

Revisi : 01

SHIFT 1 DAY

RENCANA PRODUKSI PC / PPC

TANGGAL : 17-Jul-17

ITEM KOTING	TYPE	CMEURO 2				L306 EURO 2				TD												FURNO MONTRO 2												TOTAL	
		PU	PU	Ida	2	PU	PU	Ida	QUAT	RL1	T1				T3				T4				T5				T6				T7				
		STD	SR	BS	WAT	STD	SR	BS	QUAT	RL1	PR	PR	PR	PR	PR	PR	PR	PR	PR	PR	PR	PR	PR	PR	PR	PR	PR	PR	PR	PR	PR				
		STD	SR	BS	WAT	STD	SR	BS	QUAT	RL1	PR	PR	PR	PR	PR	PR	PR	PR	PR	PR	PR	PR	PR	PR	PR	PR	PR	PR	PR	PR	PR				
WELDING	CABIN	55				82																										173			
	ROBOT					83																										83			
FRONT AXLE														41				22		112													175		
TON	SCRT1					79							43																				122		
	SCRT2																	57		72												129			
	THAN 1	58																														58			
T.OFF	THAN 1					79							43																				122		
	THAN 2																	60		49												15			
	THAN 3	58																														58			
DOCKING	PCNT1					80							42																				122		
	PCNT2																	60		67												5			
PRA CONVEYOR																		60		74												1			

Gambar IV.9 Rencana Produksi Harian
(Sumber : PT Krama Yudha Ratu Motor, 2017)

2. Order List

Order List adalah dokumen pemesanan *part* yang dilakukan oleh pelanggan (KTB) dan dikirimkan via *email* oleh admin KTB ke admin *Part Control*. *Order List* berupa *file excel* yang memuat jumlah pemesanan selama 10 hari, admin bertugas menginput data komponen masuk sesuai atau tidak dengan pesanan yang sudah dilakukan.

	A	B	C	D	E	F	G	H	I
1	Nama Vendor	No. PO	Delivery Note	Nama Komponen	Tanggal	Jumlah Pesan	Jumlah Masuk	Kekurangan	Type
2	GEMALA KEMPA DAYA	1530523593	M30A11GJ5TZ	FRAME ASSY TZ	2-Dec-17	24	10	-14	TZ
3	GEMALA KEMPA DAYA	1530523594	KM704090TG	FRAME ASSY TG	4-Dec-17	96	96	0	TG
4	GEMALA KEMPA DAYA	1530523594	M30A11GJ7TR	FRAME ASSY TR	4-Dec-17	24	24	0	TR
5	GEMALA KEMPA DAYA	1530523594	M30A11GJ6TV	FRAME ASSY TV	4-Dec-17	48	48	0	TV
6	GEMALA KEMPA DAYA	1530523594	M30A11GJ6TX	FRAME ASSY TX	4-Dec-17	36	36	0	TX
7	GEMALA KEMPA DAYA	1530523595	KM704090TG	FRAME ASSY TG	5-Dec-17	84	84	0	TG
8	GEMALA KEMPA DAYA	1530523595	M30A11GJ7TR	FRAME ASSY TR	5-Dec-17	24	24	0	TR
9	GEMALA KEMPA DAYA	1530523595	M30A11GJ6TU	FRAME ASSY TU	5-Dec-17	48	48	0	TU
10	GEMALA KEMPA DAYA	1530523595	M30A11GJ6TX	FRAME ASSY TX	5-Dec-17	12	12	0	TX
11	GEMALA KEMPA DAYA	1530523596	M30A11GJ5TH	FRAME ASSY TH	6-Dec-17	12	12	0	TH
12	GEMALA KEMPA DAYA	1530523596	M30A11GJ7TR	FRAME ASSY TR	6-Dec-17	24	24	0	TR

Gambar IV.10 Order List
(Sumber : PT Krama Yudha Ratu Motor, 2017)

3. Delivery Note

Delivery Note atau biasa disebut DN merupakan surat jalan dari *vendor* sebagai bukti pengiriman komponen. Admin bertugas mengecek apakah nomor DN sesuai atau tidak dengan nomor DN pada *order list* serta menginput jumlah komponen masuk yang tertera di DN.

Gambar IV.11 Delivery Note
(Sumber : PT Krama Yudha Ratu Motor, 2017)

4. Dokumen Permohonan Surat Jalan

Dokumen ini digunakan apabila ada komponen perakitan atau yang disebut suplemen yang dipasang diluar pabrik KRM, pengiriman komponen tersebut

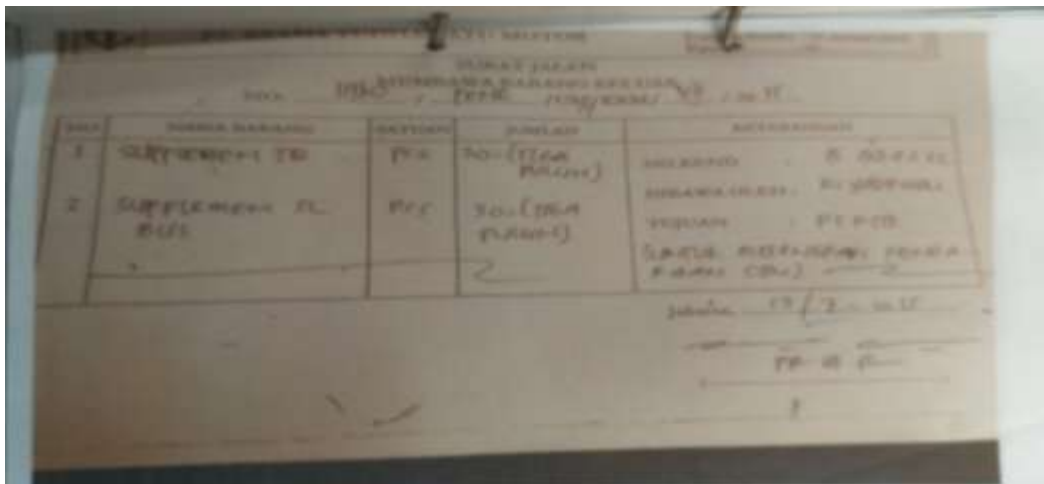
menggunakan surat jalan atau biasa disebut surat pengiriman komponen. Sebelum dikeluarkan komponen dari *Part Control*, diperlukan izin dengan mengajukan dokumen permohonan surat jalan sebagai berikut:

[illegible]

Gambar IV.12 Dokumen Permohonan Surat Jalan
(Sumber : PT Krama Yudha Ratu Motor, 2017)

5. Surat Pengiriman Komponen

Surat Pengiriman Komponen adalah surat yang dikeluarkan apabila permohonan surat jalan pengiriman komponen sudah disetujui oleh kepala departemen yang berisi data komponen yang akan keluar sebagai berikut:



Gambar IV.13 Surat Pengiriman Komponen
(Sumber : PT Krama Yudha Ratu Motor, 2017)

6. Surat Permintaan *Advance Part*

Surat Permintaan *Advance Part* adalah dokumen yang dikeluarkan apabila terjadi kekurangan *part* pada proses produksi sehingga meminta *vendor* untuk mengirimkan pesanan *part* pada tanggal selanjutnya agar dikirim lebih awal. Surat ini dikirimkan via *email* ke *vendor* berupa *file excel*.

PERMINTAAN ADVANCE DELIVERY										
VENDOR	Purchasing Document	Material Number	Description	Type	OLD			ADVANCE		
					Div Date	Tracking No.	Sch Qty	Div Date	Tracking No.	Sch Qty
SHIN HEUNG INDONESIA	1530517535	MK434981	LATCH/FR DOOR, LH	TQ, T2, TY, TX, T, W, TV, TU, TG, T, T, TL, TH, TK, TF	9-Sep-17		120	4-Sep-17		60

Gambar IV.14 Surat Permintaan *Advance Part*
(Sumber : PT Krama Yudha Ratu Motor, 2017)

7. Laporan Harian *Part Control*

Laporan harian yang dibuat oleh admin *Part Control* berupa laporan masalah yang terjadi pada penerimaan barang. Laporan ini akan diserahkan kepada pelanggan (KTB).



PT. KIRANA YUDHA RATU MOTOR

Jl. Raya Bekasi Km. 21-22 Planda Terse - Cikarang
Telp. 4952905, Fax. 4952904, P.O. Box-13613 Jakarta 13660

Daily Report Of Part Control Local

GROUP

DATE : 18 JULI 2017

DELIVERY PROBLEM

NO	REKOR	DELIVERY DATE	MATERIAL NUMBER	PART NAME	DELIVERY DATE	TITLE	SCHEDULE QTY	QTY DELIVER	DIFF	PROBLEM
1	BRAJA MUKTI CAGRA	150807131	MS755770	KNOCKLE	15-06-17	DU DUF DA DU QUF CA DUC	120	81	-39	NOVEN DELIVERY
2	BRAJA MUKTI CAGRA	150807131	MS755780	KNOCKLE	15-06-17	DU DUF DA DU QUF CA DUC	120	81	-39	NOVEN DELIVERY
3	INDOKARLO PERWASA	150813170	MS48251A	CONNECTOR	15-06-17	DU DUF DA DU QUF CA DUC	180	140	-40	NOVEN DELIVERY
4	KARTYA SAHANA UMGAH	150814530	MS35952	SPRINT ASSY LMR PML CTR	15-06-17	DU DUF DA DU QUF CA DUC	120	82	-38	NOVEN DELIVERY
5	WIND GOSIE TECHNOLOGY INDOEST	150814327	KMB4831C	FRONT GRILLE ASSY	17-06-17	DU DUF DA DU QUF CA DUC	180	173	-7	NOVEN DELIVERY
6	TRIMTEKA CITRAWASTA	150814310	MS48341D	LINK ASSY ACCEL	15-06-17	DU DUF DA DU QUF CA DUC	180	182	2	NOVEN DELIVERY
7										
8										
9										

TENDANCE MATTERS OF PERSON

NO	NAME	NAME	KEY

LINE STOP REPORT

NO	PART NAME	CAUSE	REMARK	REMARK

REMARKS

DATE

REMARKS

DATE

Gambar IV.15 Laporan Harian *Part Control*
(Sumber : PT Krama Yudha Ratu Motor, 2017)

4. 11 Prosedur Penerimaan Komponen

Adapun prosedur sistem informasi dalam penerimaan komponen:

1. Admin *Part Control* mengunduh data *Order List* dari perusahaan induk yang merupakan pelanggan KRM
2. Admin *Part Control* menerima *daily schedule* dari PPC.
3. Admin memeriksa stok untuk mencocokkan jumlah komponen yang dipesan dengan produksi harian apakah mencukupi atau tidak.
4. Jika stok mencukupi, *Vendor* akan mengirimkan komponen berdasarkan *Order List*. Jika tidak mencukupi, *Part Control* akan memberikan surat permintaan *advance part* ke *vendor*.

5. *Vendor* menerima *order* dan mengirimkan komponen ke pabrik KRM dengan membawa DN.
6. *Part Control* memeriksa kesesuaian DN dan memberikan ke Admin.
7. Admin menerima DN *checked* dan menginput data komponen masuk pada *file order list*.
8. Admin membuat laporan harian penerimaan komponen.
9. *Part Control* menyusun komponen sesuai dengan produksi harian KRM.
10. Komponen yang sudah disusun kemudian diletakkan ke *dolly* dan dikirimkan ke bagian produksi.
11. Bagian produksi menerima komponen dari PC.
12. Apabila produksi membutuhkan suplemen, maka dibuatkan permohonan surat jalan pengiriman komponen ke Admin.
13. Admin menerima surat permohonan pengiriman komponen.
14. Admin menyetujui permohonan pengiriman komponen dan mengeluarkan surat pengiriman komponen ke PC.
15. *Part Control* menerima surat pengiriman komponen dan mengirimkan suplemen ke gudang KTB.

Berikut merupakan *flowmap* penerimaan komponen di bagian *Part Control* pada PT. Krama Yudha Ratu Motor:

‘Gambar IV.16 Aliran Dokumen Sistem Informasi Penerimaan Komponen
(Sumber : PT Krama Yudha Ratu Motor, 2017)

Contents

5. 1	Analisis Kebutuhan <i>Functional</i> Sistem	72
5. 2	Prosedur Sistem Informasi Penerimaan dan Pengiriman Komponen.....	73
5. 3	Pemodelan Sistem Informasi Penerimaan dan Pengiriman Komponen	76
	5.3.1 <i>Use case diagram</i>	76
	5.3.2 <i>Activity Diagram</i>	82
	5.3.3 <i>Sequence Diagram</i>	91
	5.3.4 <i>Class Diagram</i>	97
	5.3.5 <i>Deployment Diagram</i>	98
5. 4	Pemodelan Data	99
	5.4.1..... <i>Entity Relationship Diagram (ERD)</i> . 99	
	5.4.2..... <i>Conceptual Data Model (CDM)</i> 100	
5. 5	Kamus Data	101
5. 6	Analisis Desain Program	104
	5.6.1 WND (<i>Windows Navigation Diagram</i>).....	104
	5.6.2 Perancangan <i>Interface</i> Sistem	105
5. 7	Implementasi Sistem <i>Software</i> dan <i>Hardware</i>	115

BAB V

ANALISIS DAN PEMBAHASAN

5.1 Analisis Kebutuhan *Functional* Sistem

Berdasarkan hasil pengolahan data dan analisis sistem, maka dibutuhkan perancangan aplikasi sistem informasi penerimaan dan pengiriman komponen pada Divisi *Part Control* pada PT Krama Yudha Ratu Motor. Berikut adalah daftar kebutuhan *Functional* sistem untuk aplikasi sistem informasi penerimaan dan pengiriman komponen pada Divisi *Part Control*.

Tabel V.1 Kebutuhan *Functional* Sistem

No.	Masalah	Kebutuhan User	Kebutuhan <i>Functional</i> Sistem	Uraian
1.	Pengisian <i>form</i> penerimaan komponen dan laporan harian dilakukan dengan <i>input</i> melalui <i>Microsoft Excel</i> dimana admin seringkali salah ketik karena kesulitan dalam melihat data pada <i>sheet</i> dan dokumen yang berbeda.	a. Dapat mengisi <i>form</i> penerimaan komponen dengan mudah dan cepat	a. Terdapat form penerimaan komponen yang simple untuk mencegah kesalahan dalam menginput jumlah komponen masuk	a. Proses yang dapat dilakukan pada penerimaan komponen yaitu: Dapat menambah dan mencari data penerimaan komponen
		b. Pembuatan Laporan harian penerimaan dan pengiriman komponen lebih cepat tanpa input ulang data	b. Menyediakan menu laporan harian	b. Pembuatan laporan harian penerimaan dan pengiriman komponen secara otomatis dan terdapat menu untuk mencetak laporan

Tabel V.1 Kebutuhan *Functional* Sistem (Lanjutan)

No.	Masalah	Kebutuhan User	Kebutuhan <i>Functional</i> Sistem	Uraian
2.	Pembuatan surat pengiriman komponen masih manual yaitu dengan mengisi manual lembar pengiriman komponen.	User ingin ada nya <i>form</i> pengiriman komponen	Terdapat menu pengiriman komponen	Proses yang dapat dilakukan pada menu pengiriman komponen yaitu: -Dapat membuat surat pengiriman dan surat jalan pengiriman komponen.
3.	Admin menyimpan dokumen di sembarang tempat yang disebabkan kurangnya tempat penyimpanan dokumen sehingga berisiko terjadinya kehilangan data, sulit mencari data dan kerusakan data.	Pengelolaan data yang mudah dan cepat	Pengelolaan Data Master	Data master diantaranya adalah data pengguna, data komponen, data <i>vendor</i> . Proses yang dapat dilakukan pada pengelolaan data tersebut adalah: - Dapat menambah, mengubah, mencari dan menghapus data tersebut.

(Sumber: Hasil analisis, 2018)

5.2 Prosedur Sistem Informasi Penerimaan dan Pengiriman Komponen

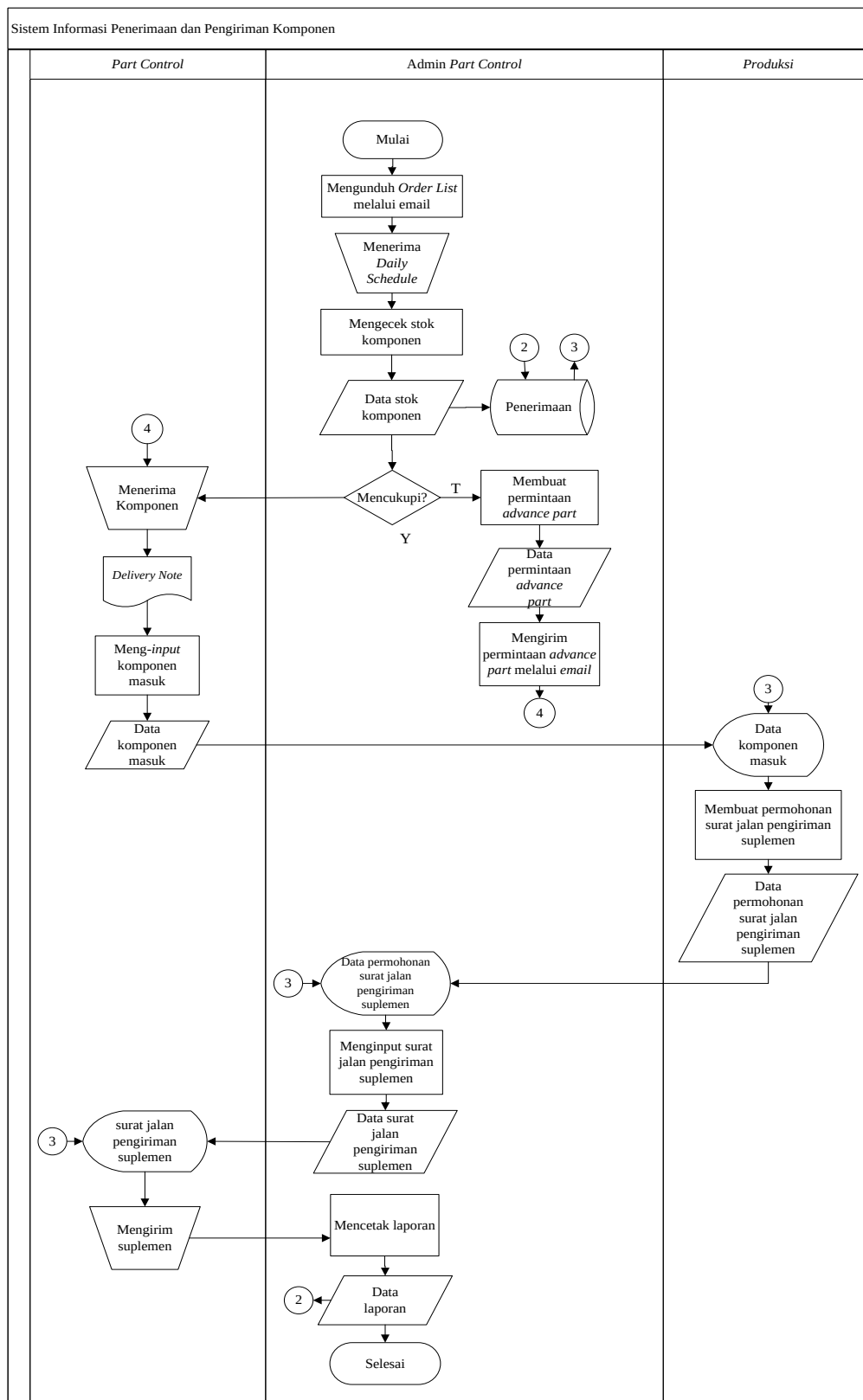
Prosedur sistem informasi penerimaan dan pengiriman komponen yang diusulkan melalui tahapan sebagai berikut:

1. Admin *Part Control* mengunduh *Order List* bulanan dari perusahaan induk KTB melalui *email* dan menerima *daily schedule* dari PPC.
2. Kemudian admin *Part Control* mengecek stok komponen ke sistem untuk mengetahui apakah komponen mencukupi atau tidak. Apabila stok mencukupi,

bagian *Part Control* akan menerima komponen. Jika stok tidak mencukupi, maka admin akan membuat permintaan *advance part* dan masuk ke *database* sistem.

3. Permintaan *advance part* yang dibuat oleh admin melalui sistem akan dikirimkan melalui *email*.
4. Bagian *Part Control* menerima komponen beserta *Delivery Note*.
5. *Part Control* menginput data komponen masuk ke sistem setelah menerima *Delivery Note*, kemudian tersimpan ke *database*.
6. Bagian Produksi membuat permohonan surat jalan pengiriman suplemen kepada Admin *Part Control*.
7. Admin *Part Control* melihat data permohonan surat jalan pengiriman suplemen dan menginput surat jalan pengiriman komponen ke sistem, kemudian tersimpan dalam *database*.
8. *Part control* melihat data surat jalan pengiriman komponen yang sudah tersimpan di *database*.
9. *Part Control* mengirimkan suplemen komponen sesuai dengan surat jalan.
10. Admin *Part Control* melihat data komponen masuk dan keluar pada sistem dan mencetak laporan penerimaan dan pengiriman komponen.

Seperti yang dapat dilihat pada Gambar V.1 *Flowmap* Sistem Informasi Penerimaan dan Pengiriman Komponen usulan sebagai berikut:



Gambar V.1 Flowmap Sistem Informasi Penerimaan dan Pengiriman Komponen usulan
(Sumber: Hasil analisis, 2018)

5.3 Pemodelan Sistem Informasi Penerimaan dan Pengiriman Komponen

Pemodelan sistem informasi penerimaan dan pengiriman komponen yang diusulkan menggunakan pemodelan sistem UML (*Unified Modeling Language*) yang meliputi pembuatan *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*, *Deployment Diagram* dan juga pembuatan Kamus Data. Tahapan analisis ini akan memberikan gambaran mengenai aliran informasi dan data pada sistem informasi yang akan dibangun sebagai berikut:

5.3.1 *Use case diagram*

Sistem Informasi Penerimaan dan Pengiriman Komponen memiliki keterkaitan antar aktor, diantaranya *Admin Part Control*, *Part Control*, dan Produksi. Setiap aktor memiliki kepentingan-kepentingan yang berkaitan dengan sistem tersebut. Adapun untuk detail dari analisis adalah sebagai berikut:

1. Definisi Aktor

Pendefinisian aktor pada *use case diagram* sistem informasi penerimaan dan pengiriman komponen usulan:

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

No.	Aktor	Definisi
1.	Admin <i>Part Control</i>	Admin <i>Part Control</i> adalah aktor yang bertugas mengelola data <i>master</i> , mengunduh data <i>order list</i> melalui email, melakukan pengecekan stok komponen, membuat <i>advance part</i> , mencetak laporan penerimaan dan pengiriman komponen, dan menginput surat jalan pengiriman suplemen.
2.	<i>Part Control</i>	<i>Part Control</i> adalah aktor yang bertugas menerima dan mengirim komponen serta menginput data komponen masuk.
4.	Produksi	Produksi adalah aktor yang bertugas membuat permohonan surat jalan pengiriman suplemen.

(Sumber: Hasil Analisis, 2018)

3.	Mengecek stok komponen	Proses mengecek ketersediaan stok komponen pada gudang.
4.	Membuat permintaan <i>advance part</i>	Proses membuat surat permintaan <i>advance part</i> (permintaan mempercepat pengiriman komponen dari tanggal yang ditentukan) apabila stok komponen tidak mencukupi.
5.	Menginput komponen masuk	Proses <i>input</i> data komponen masuk.
6.	Membuat permohonan surat jalan pengiriman suplemen	Proses membuat permohonan surat jalan pengiriman suplemen kepada admin <i>part control</i> .
7.	Membuat surat jalan pengiriman suplemen	Proses membuat surat jalan pengiriman suplemen yang telah disetujui.
8.	View surat jalan pengiriman suplemen	Proses melihat surat jalan pengiriman suplemen untuk mengirimkan suplemen.
9.	Mencetak laporan penerimaan dan pengiriman komponen	Proses mencetak laporan penerimaan dan pengiriman komponen.

(Sumber: hasil analisis, 2018)

3. Skenario *Use Case*

Skenario jalannya masing-masing *use case* pada sistem informasi penerimaan dan pengiriman komponen usulan dapat dilihat pada poin berikut:

1. *Use case login*

Berikut skenario *use case login* yang terdapat pada tabel V.4:

Tabel V.4 Definisi *Use Case Login*

Nama Use Case	<i>Login</i>
<i>Primary Actor</i>	Admin <i>Part Control</i> , <i>Part Control</i> , dan Produksi
<i>Use Case Description</i>	<i>Use case</i> ini menggambarkan <i>login</i> sesuai dengan hak akses dari <i>user</i> . Dalam hal ini hak aksesnya sesuai jabatan.
<i>Relationship</i>	<i>Association</i> : Admin <i>Part Control</i> , <i>Part Control</i> , dan Produksi.

Tabel V.4 Definisi *Use Case Login* (Lanjutan)

Nama Use Case	<i>Login</i>
<i>Normal Flow Event</i>	1. User membuka aplikasi. 2. Sistem Menampilkan Form Login. 3. User mengisi user name dan password pada form login. 4. Sistem melakukan validasi login.

	5. Jika user name dan password valid, maka muncul halaman utama. 6. Jika user name dan password salah (tidak valid), maka menampilkan pesan <i>login</i> gagal.
--	---

(Sumber: Hasil Analisis, 2018)

2. Use case mengelola data *master*

Berikut skenario *Use case* mengelola data *master*:

Tabel V.5 Definisi *Use case* mengelola data *master*

Nama Use Case	Use case mengelola data <i>master</i>
Primary Actor	Admin Part Control
Use Case Description	Use Case ini adalah proses untuk melakukan pengolahan data <i>master</i> diantaranya adalah data <i>user</i> , data komponen, dan data <i>vendor</i> .
Relationship	Association: Admin Part Control Include: Login
Normal Flow Event	a. Admin Part Control memilih menu data <i>master</i>. b. Admin Part Control memilih sub menu data <i>master</i>. c. Sistem menampilkan data sub menu data <i>master</i> yang dipilih. 4. Admin Part Control dapat melakukan proses tambah, ubah, cari dan hapus data <i>user</i>, data komponen, dan data <i>vendor</i>. 5. Sistem akan memperbarui <i>database</i>. 6. Sistem akan menampilkan data.

(Sumber: Hasil Analisis, 2018)

3. Use case mengecek stok komponen

Berikut skenario *Use case* mengecek stok komponen yang terdapat pada tabel V.6:

Tabel V.6 Definisi *Use case* mengecek stok komponen

Nama Use Case	Use case mengecek stok komponen
Primary Actor	Admin Part Control
Use Case Description	Use Case ini adalah proses untuk melakukan pengecekan ketersediaan stok apakah mencukupi atau tidak.
Relationship	Association: Admin Part Control Include: Login

<i>Normal Flow Event</i>	1. Admin <i>Part Control</i> memilih menu pengelolaan. 2. Admin <i>Part Control</i> memilih sub menu pengecekan stok. 3. Sistem menampilkan data sub menu pengecekan stok yang dipilih. 4. Admin <i>Part Control</i> memasukkan target produksi harian lalu mengklik tombol cek. 5. Sistem menampilkan ketersediaan stok.
--------------------------	---

(Sumber: Hasil Analisis, 2018)

4. Use case membuat advance part

Berikut skenario Use case membuat advance part:

Tabel V.7 Definisi Use case membuat advance part

<i>Nama Use Case</i>	<i>Use case membuat advance part</i>
<i>Primary Actor</i>	<i>Admin Part Control</i>
<i>Use Case Description</i>	<i>Use Case ini adalah proses membuat surat permintaan advance part ke vendor saat stok komponen tidak mencukupi.</i>
<i>Relationship</i>	<i>Association: Admin Part Control Include: Login, Mengecek stok komponen</i>

Tabel V.7 Definisi Use case membuat advance part (Lanjutan)

<i>Nama Use Case</i>	<i>Use case membuat advance part</i>
<i>Normal Flow Event</i>	1. Admin <i>Part Control</i> mengklik tombol advance part. 2. Sistem menampilkan form permintaan advance part. 3. Admin <i>Part Control</i> memasukkan data part yang akan di-advance. 4. Admin <i>Part Control</i> mengklik kirim email. 5. Sistem menampilkan halaman advance part.

(Sumber: Hasil Analisis, 2018)

5. Use case menginput komponen masuk

Berikut skenario Use case menginput komponen masuk yang terdapat pada tabel V.8:

Tabel V.8 Definisi Use case menginput komponen masuk

<i>Nama Use Case</i>	<i>Use case menginput komponen masuk</i>
<i>Primary Actor</i>	<i>Part Control</i>
<i>Use Case Description</i>	<i>Use Case ini adalah proses menginput data penerimaan komponen masuk</i>

<i>Relationship</i>	<i>Association: Admin Part Control</i> <i>Include: Login</i>
<i>Normal Flow Event</i>	1. <i>Part Control</i> memilih menu pengelolaan. 2. <i>Part Control</i> memilih sub menu komponen masuk. 3. Sistem menampilkan form komponen masuk. 4. <i>Part Control</i> dapat melakukan proses tambah, ubah, cari dan hapus data penerimaan komponen. 5. Sistem akan memperbarui <i>database</i>. 6. Sistem akan menampilkan data.

(Sumber: Hasil Analisis, 2018)

6. *Use case* Membuat permohonan surat jalan pengiriman suplemen

Berikut skenario *Use case* Membuat permohonan surat jalan pengiriman suplemen yang terdapat pada tabel V.9:

Tabel V.9 Definisi *Use case* Membuat permohonan surat jalan pengiriman suplemen

<i>Nama Use Case</i>	<i>Use case</i> Membuat permohonan surat jalan pengiriman suplemen
<i>Primary Actor</i>	Produksi
<i>Use Case Description</i>	<i>Use Case</i> ini adalah proses Membuat permohonan surat jalan pengiriman suplemen komponen kepada admin <i>part control</i>
<i>Relationship</i>	<i>Association: Produksi</i> <i>Include: Login</i>
<i>Normal Flow Event</i>	1. Produksi memilih menu pengelolaan 2. Produksi memilih sub menu permohonan surat jalan 3. Sistem menampilkan <i>form</i> permohonan surat jalan 4. Produksi mengisi form permohonan surat jalan 5. Produksi dapat melakukan proses tambah, ubah, cari dan hapus data permohonan surat jalan. Sistem akan menampilkan data permohonan surat jalan pengiriman suplemen

(Sumber: Hasil Analisis, 2018)

7. *Use case* Membuat surat jalan pengiriman suplemen

Berikut skenario *Use case* Membuat surat jalan pengiriman suplemen yang terdapat pada tabel V.10:

Tabel V.10 Definisi *Use case* Membuat surat jalan pengiriman suplemen

<i>Nama Use Case</i>	<i>Use case</i> Membuat surat jalan pengiriman suplemen
<i>Primary Actor</i>	Admin <i>Part control</i>
<i>Use Case Description</i>	<i>Use Case</i> ini adalah proses Membuat surat jalan pengiriman suplemen
<i>Relationship</i>	<i>Association: Admin Part control</i> <i>Include: Login, Melakukan persetujuan pengiriman suplemen</i>
<i>Normal Flow Event</i>	1. Admin <i>Part control</i> mengisi <i>form</i> surat jalan 2. Admin <i>Part control</i> dapat melakukan proses tambah,

	ubah, cari dan hapus data surat jalan. 3. Sistem menampilkan data surat jalan
--	--

(Sumber: Hasil Analisis, 2018)

8. Use case View surat jalan pengiriman suplemen

Berikut skenario Use case View surat jalan pengiriman suplemen yang terdapat pada tabel V.11:

Tabel V.11 Definisi Use case View surat jalan pengiriman suplemen

<i>Nama Use Case</i>	<i>Use case View surat jalan pengiriman suplemen</i>
<i>Primary Actor</i>	<i>Part control</i>
<i>Use Case Description</i>	<i>Use Case ini adalah proses Melihat surat jalan pengiriman komponen yang sudah dibuat oleh admin part control</i>
<i>Relationship</i>	<i>Association: Part control</i> <i>Include: Login, Membuat surat jalan pengiriman suplemen</i>
<i>Normal Flow Event</i>	1. <i>Part control</i> memilih menu surat jalan pada halaman utama 2. Sistem menampilkan data surat jalan

(Sumber: Hasil Analisis, 2018)

9. Use case mencetak laporan

Berikut skenario Use case mencetak laporan yang terdapat pada tabel V.12:

Tabel V.12 Definisi Use case mencetak laporan penerimaan komponen

<i>Nama Use Case</i>	<i>Use case mencetak laporan penerimaan komponen</i>
<i>Primary Actor</i>	<i>Admin Part Control</i>
<i>Use Case Description</i>	<i>Use Case ini adalah proses mencetak laporan penerimaan komponen</i>
<i>Relationship</i>	<i>Association: Admin Part Control</i> <i>Include: Login, menginput komponen masuk</i>
<i>Normal Flow Event</i>	1. <i>Admin Part Control</i> memilih menu laporan penerimaan komponen. 2. <i>Admin Part Control</i> memasukkan tanggal laporan yang ingin di cetak. 3. Sistem menampilkan data laporan harian yang ingin dicetak.

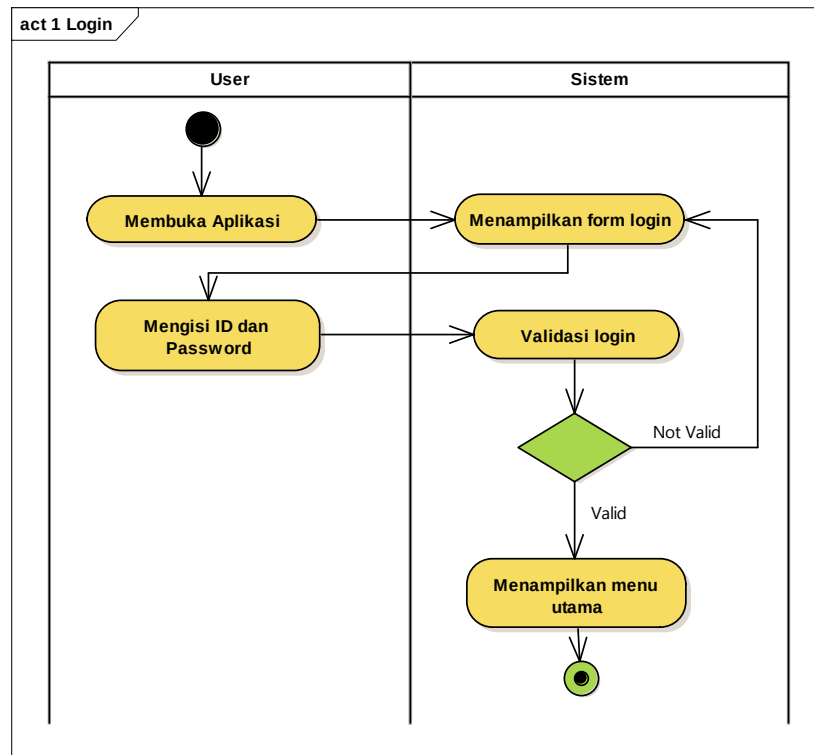
(Sumber: Hasil Analisis, 2018)

5.3.2 Activity Diagram

Activity diagram digunakan untuk menggambarkan kegiatan-kegiatan yang ada di dalam sistem. Agar memahami sistem yang akan dibuat, maka perlu dibuatkan activity diagram tentang sistem informasi penerimaan dan pengiriman komponen ini sebagai berikut:

1. Activity Diagram Login

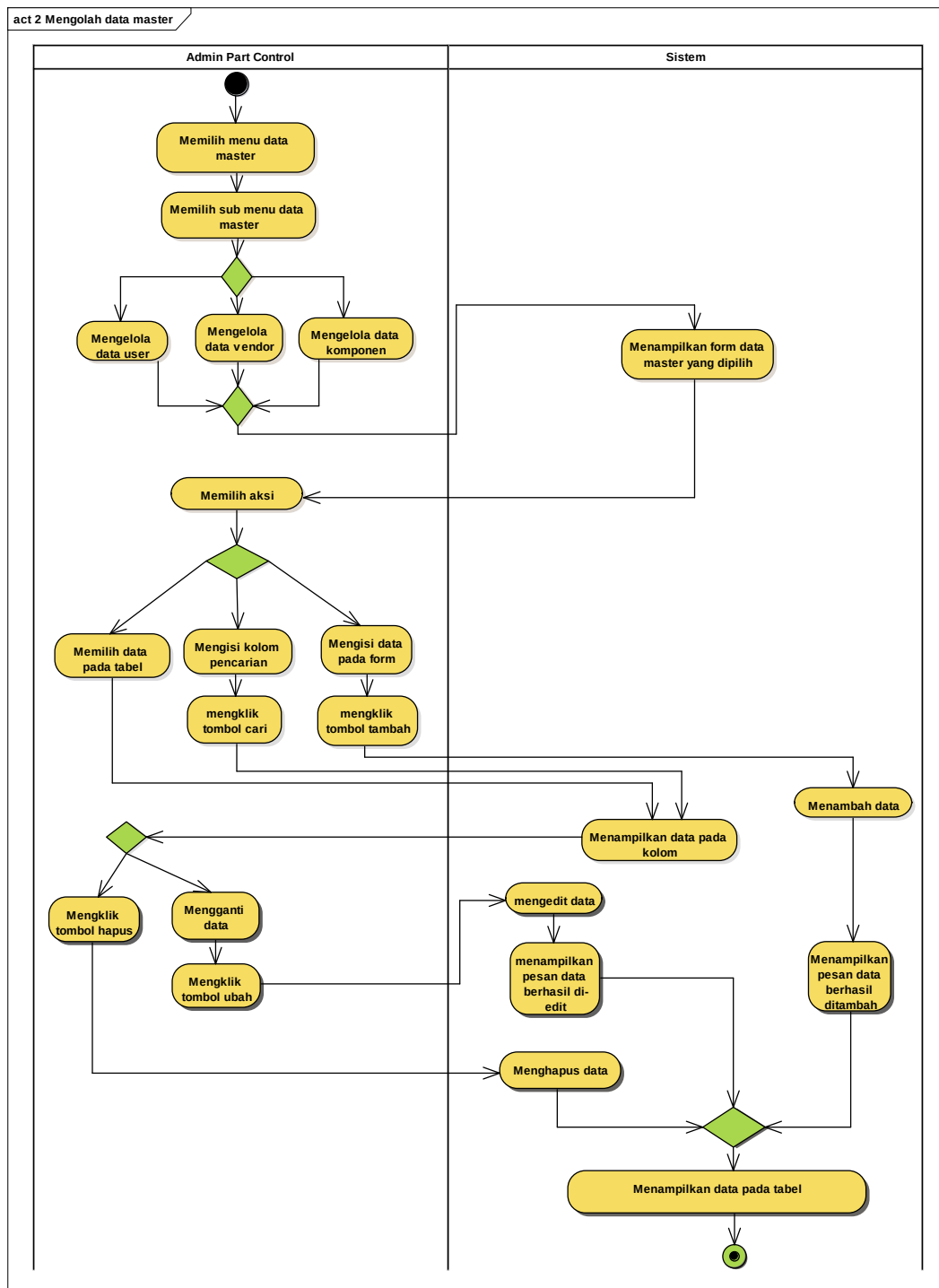
Activity diagram login ini menjelaskan alur aktivitas yang dilakukan oleh *user*, yaitu *Admin Part Control*, *Part Control*, dan *Produksi* untuk dapat masuk ke dalam sistem informasi penerimaan dan pengiriman komponen. *Activity diagram* dapat dilihat pada gambar berikut:



Gambar V.3 Activity Diagram Login
(Sumber: Hasil Analisis, 2018)

2. Activity Diagram Mengelola Data Master

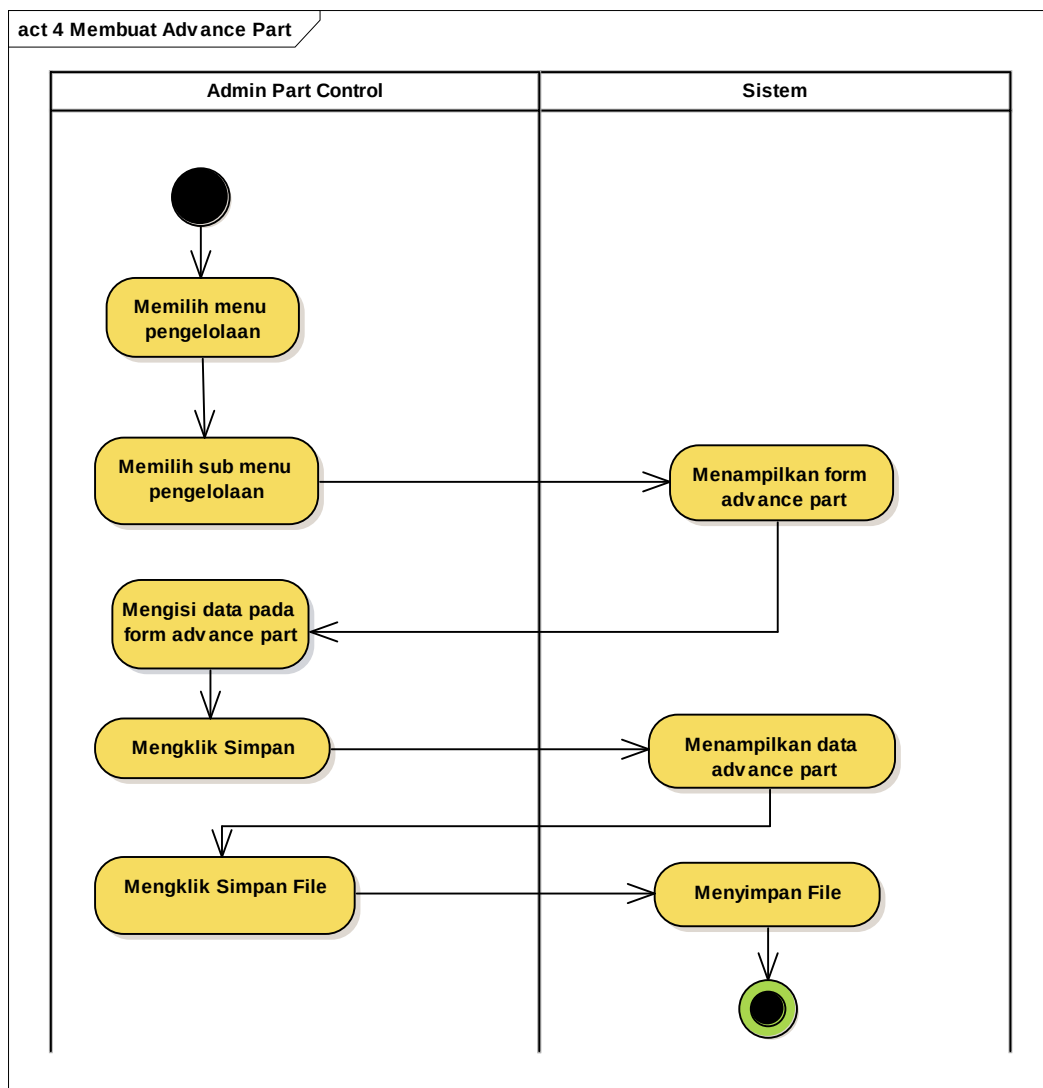
Activity diagram mengelola data master ini menjelaskan alur aktivitas yang dilakukan oleh *Admin Part Control* agar data yang ada dalam *database* selalu diperbarui. *Activity diagram* mengelola data master dapat dilihat pada Gambar V.4 sebagai berikut:



Gambar V.4 Activity Diagram Mengelola Data Master
(Sumber: Hasil Analisis, 2018)

3. Activity Diagram Mengecek Stok Komponen

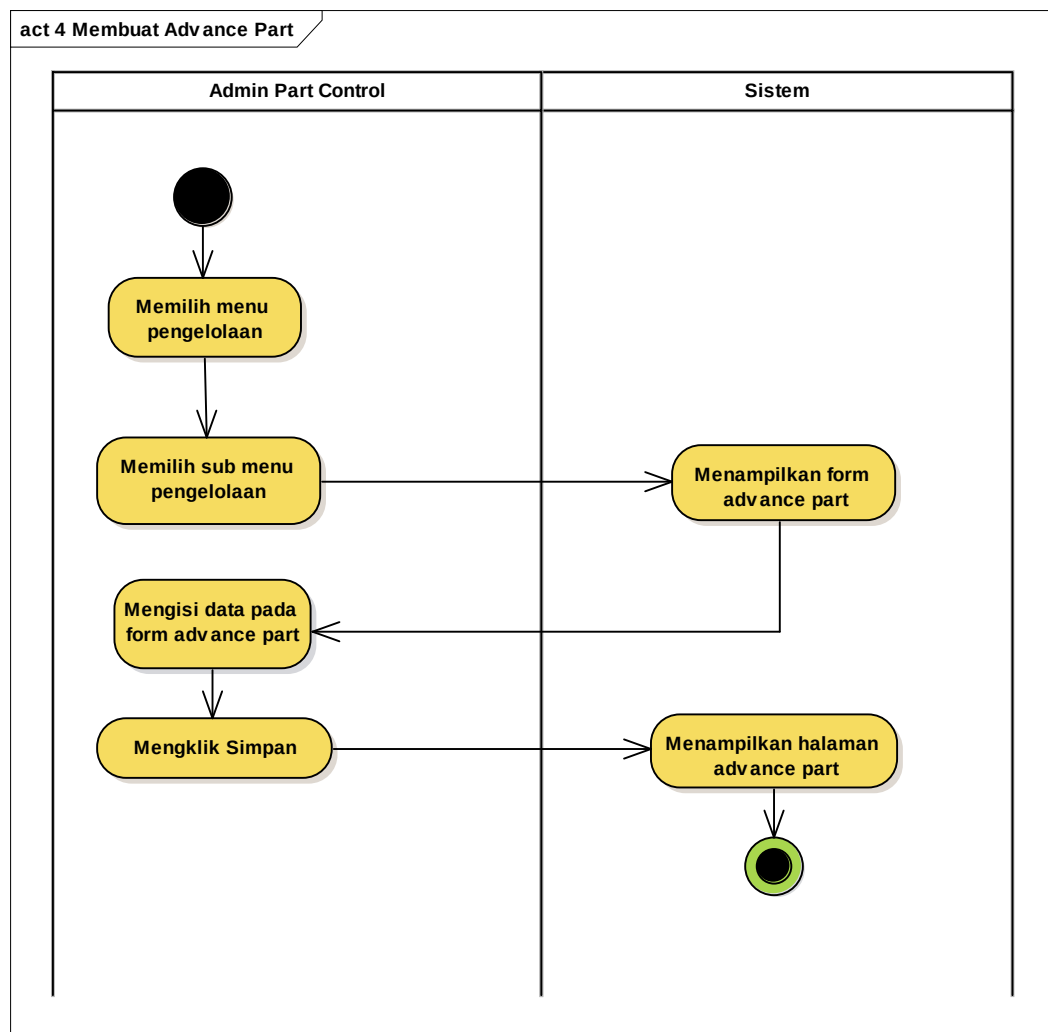
Activity diagram mengecek stok komponen berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada pengecekan stok komponen. Activity diagram mengecek stok komponen dapat dilihat pada Gambar V.5 sebagai berikut:



Gambar V.5 Activity Diagram Mengecek Stok Komponen
(Sumber: Hasil Analisis, 2018)

4. Activity Diagram Membuat Permintaan Advance Part

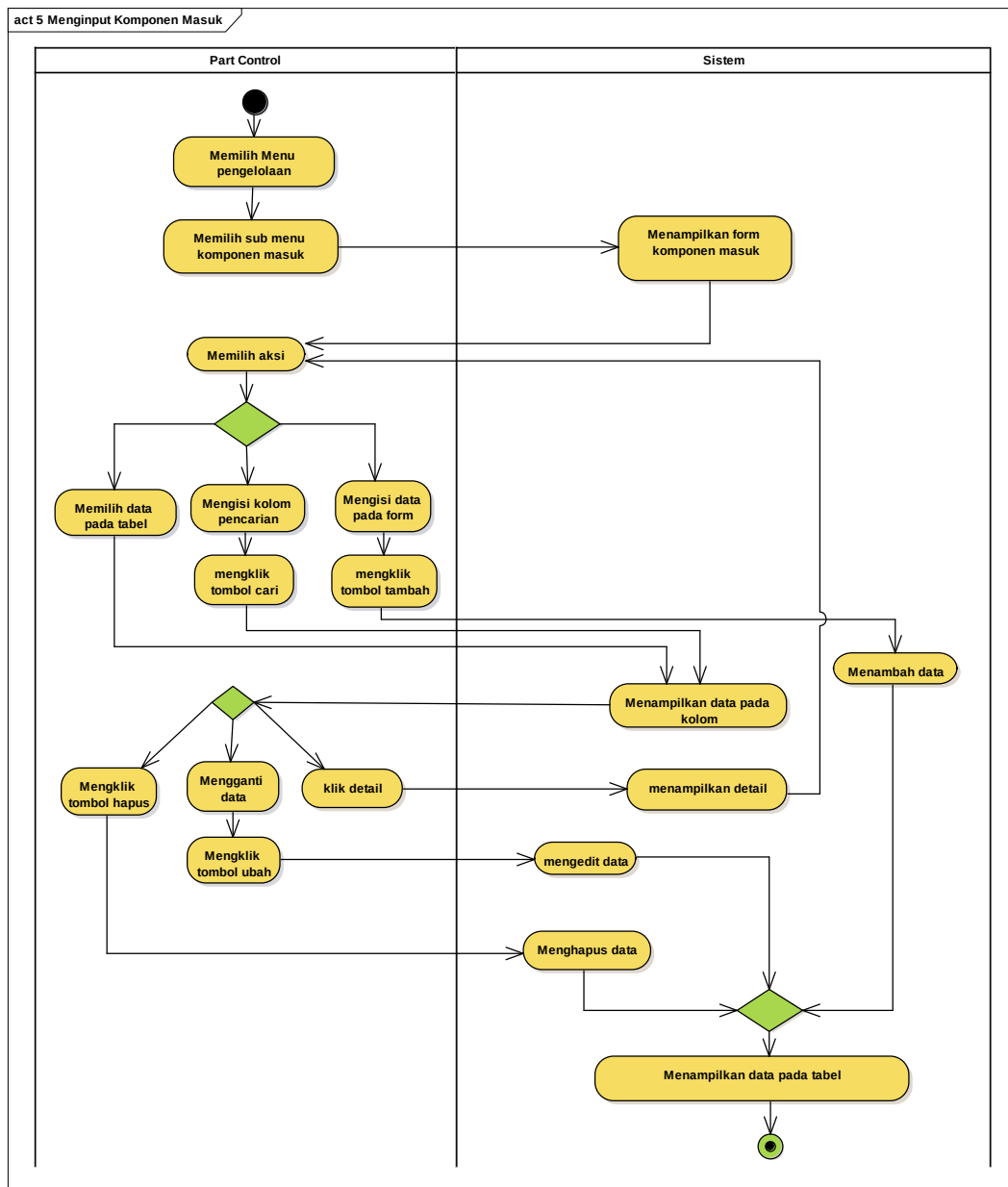
Activity diagram membuat permintaan *advance part* berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada permintaan *advance part*. Activity diagram membuat permintaan *advance part* dapat dilihat pada Gambar V.6 sebagai berikut:



Gambar V.6 Activity Diagram Membuat Permintaan Advance Part
(Sumber: Hasil Analisis, 2018)

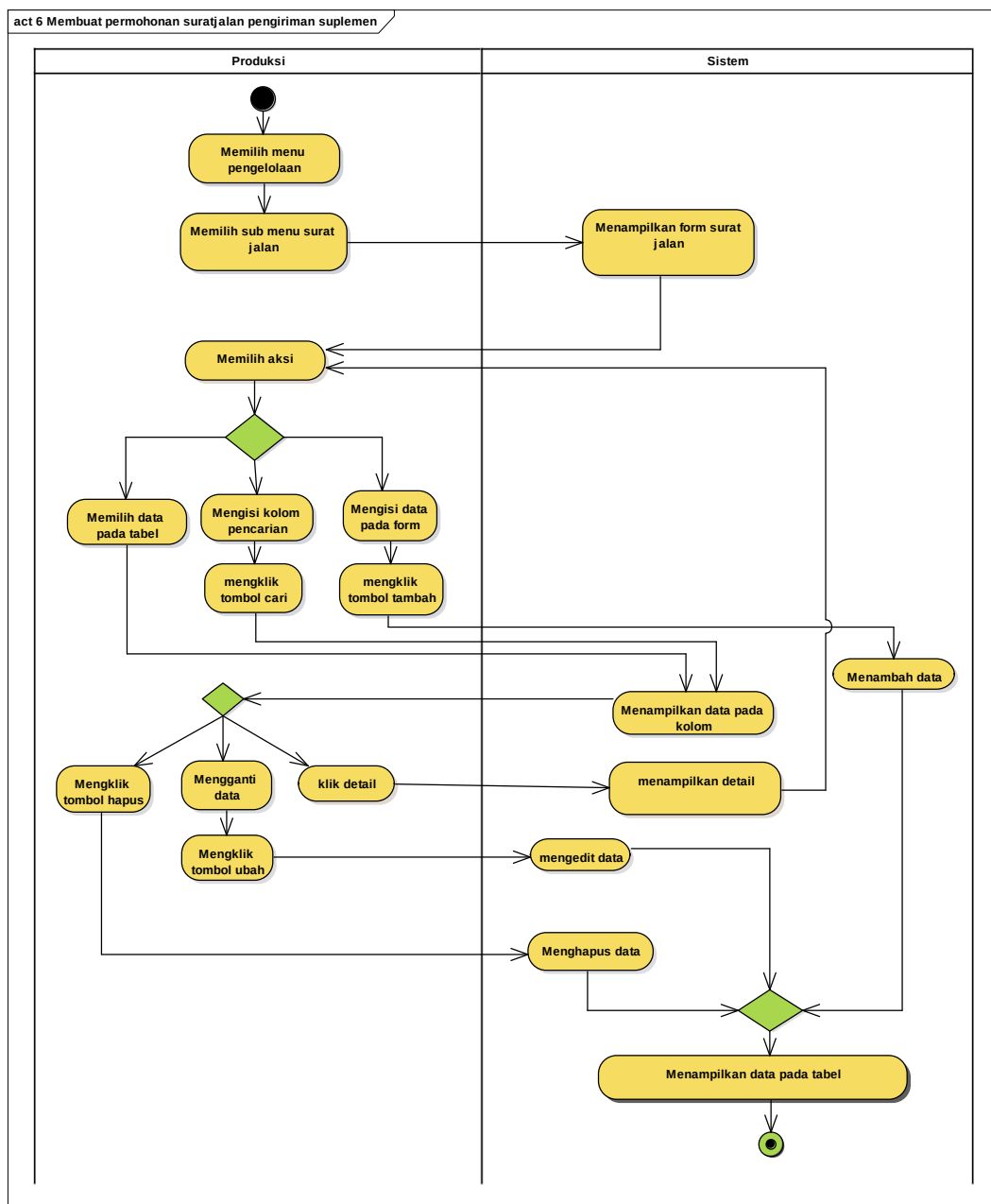
5. Activity Diagram Menginput Komponen Masuk

Activity diagram menginput komponen masuk berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada penginputan komponen masuk. Activity diagram membuat menginput komponen masuk dapat dilihat pada Gambar V.7 sebagai berikut:



Gambar V.7 Activity Diagram Menginput Komponen Masuk
(Sumber: Hasil Analisis, 2018)

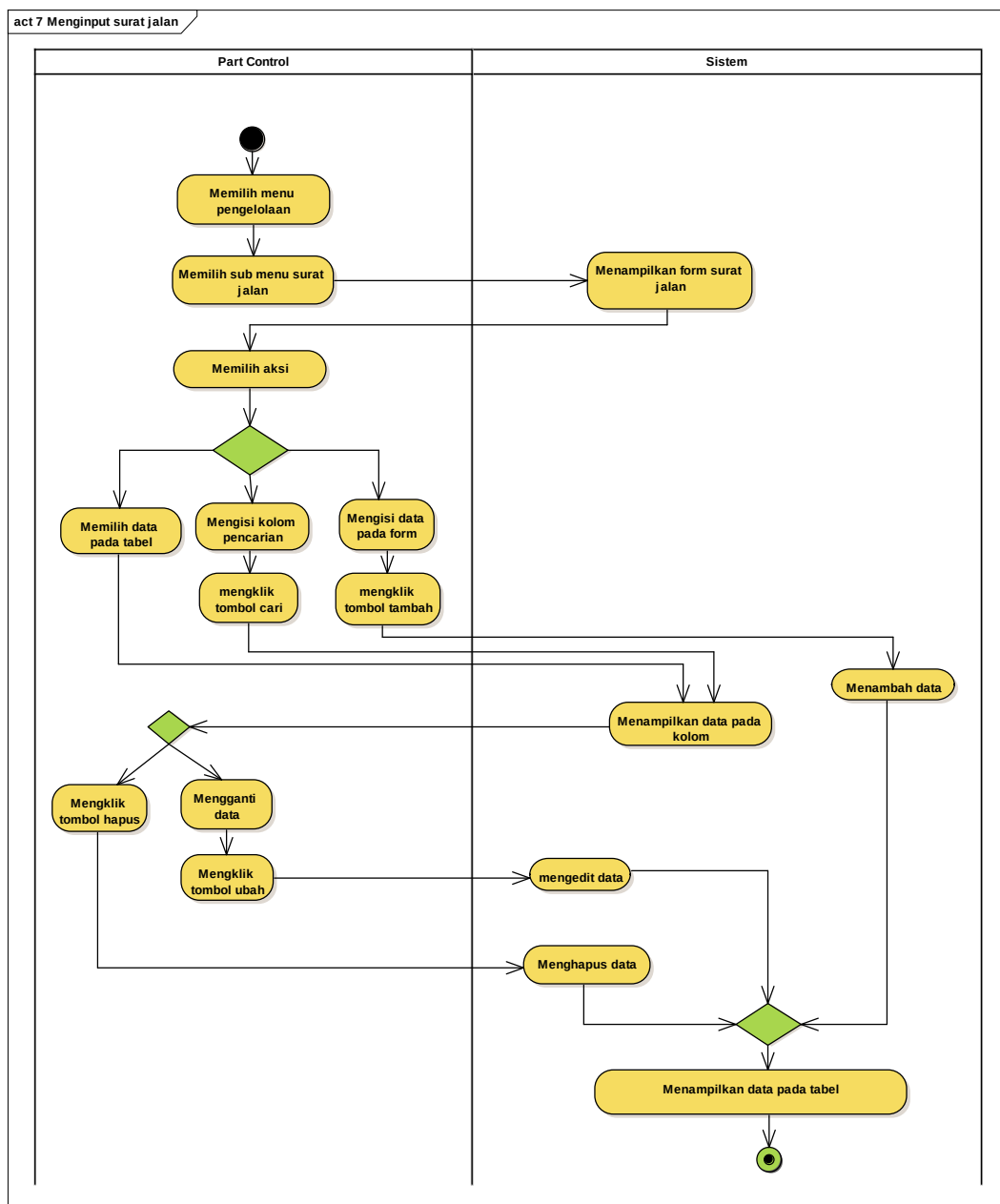
6. Activity Diagram Membuat Permohonan Surat Jalan Pengiriman Suplemen
- Activity diagram membuat permohonan surat jalan pengiriman suplemen berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada pembuatan permohonan surat jalan pengiriman suplemen. Activity diagram membuat permohonan surat jalan pengiriman suplemen dapat dilihat pada Gambar V.8 sebagai berikut:



Gambar V.8 Activity Diagram Membuat Permohonan Surat Jalan Pengiriman Suplemen
(Sumber: Hasil Analisis, 2018)

7. Activity Diagram Membuat Surat Jalan Pengiriman Suplemen

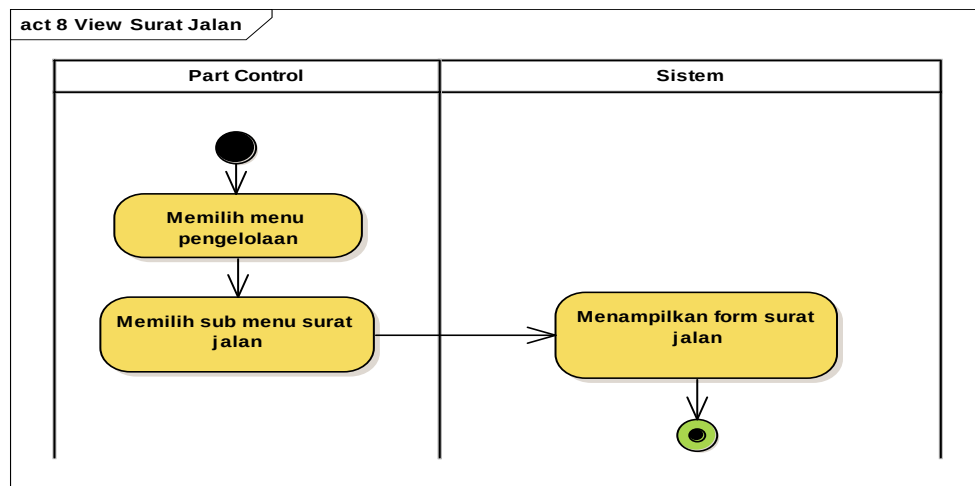
Activity diagram membuat surat jalan pengiriman suplemen berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada pembuatan surat jalan pengiriman suplemen. Activity diagram membuat surat jalan pengiriman suplemen dapat dilihat pada Gambar V.9 sebagai berikut:



Gambar V.9 Activity Diagram Membuat Surat Jalan Pengiriman Suplemen
(Sumber: Hasil Analisis, 2018)

8. Activity Diagram View Surat Jalan

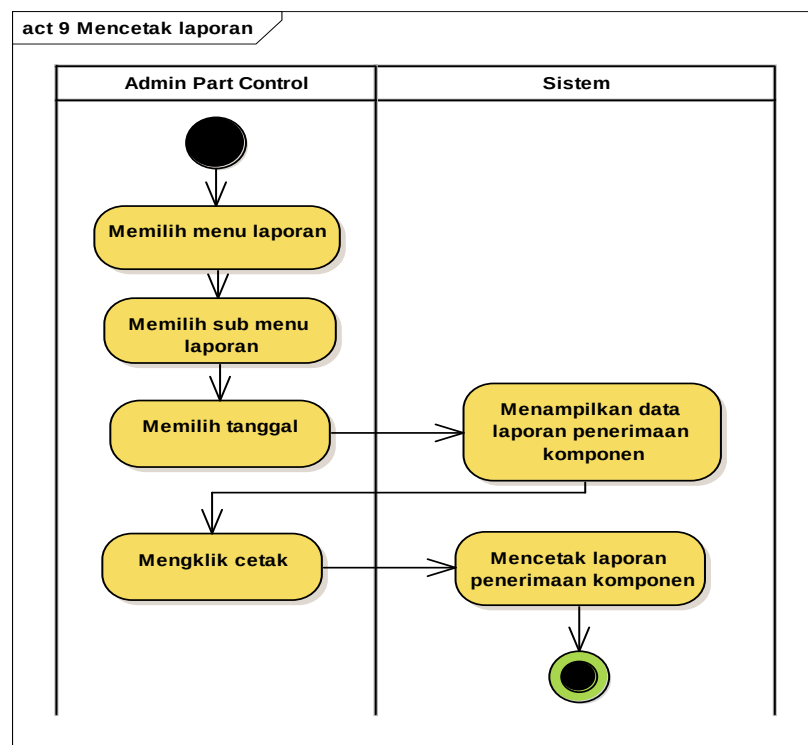
Activity diagram View Surat Jalan berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada proses melihat surat jalan pengiriman suplemen komponen. Activity diagram View Surat Jalan dapat dilihat pada Gambar V.10 sebagai berikut:



Gambar V.10 Activity Diagram View Surat Jalan
(Sumber: Hasil Analisis, 2018)

9. Activity Diagram Mencetak Laporan

Activity diagram mencetak laporan berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada pencetakan laporan penerimaan dan pengiriman komponen. Activity diagram mencetak laporan penerimaan dan pengiriman komponen dapat dilihat pada Gambar V.11 sebagai berikut:



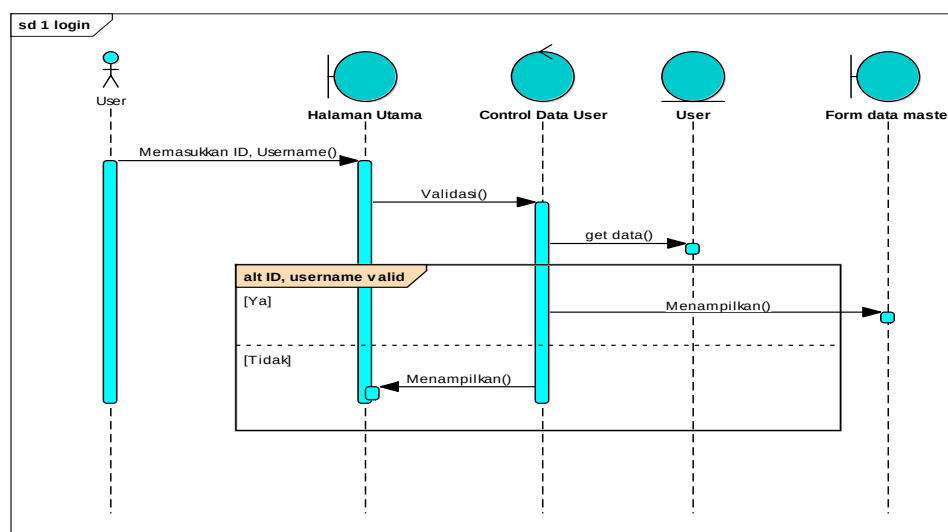
Gambar V.11 Activity Diagram Mencetak Laporan
(Sumber: Hasil Analisis, 2018)

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 di bawah ini adalah proses yang dilakukan oleh sistem ketika melakukan proses yang sesuai dengan suatu objek *use case diagram*, berikut adalah *sequence diagram* pada sistem informasi penerimaan dan pengiriman komponen:

1. Sequence Diagram Login

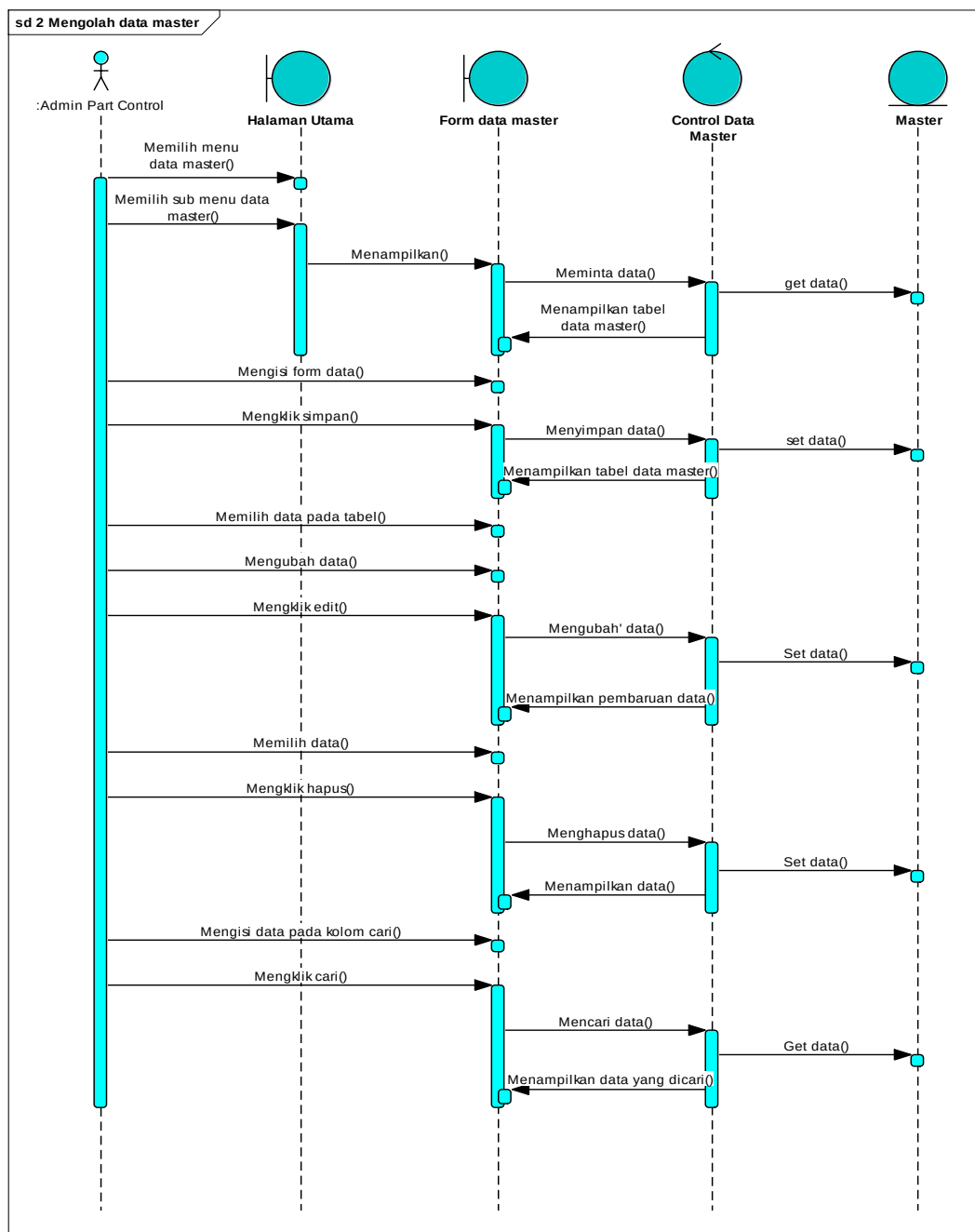
Sequence diagram login menjelaskan sebuah *sequence diagram* dalam proses *login*. Proses ini dilakukan oleh *user* yang mempunyai hak akses pada sistem. Adapun *sequence diagram* dari *use case login* dapat dilihat pada Gambar V.12.



Gambar V.12 Sequence Diagram Login
(Sumber: Hasil Analisis, 2018)

2. Sequence Diagram Mengelola Data Master

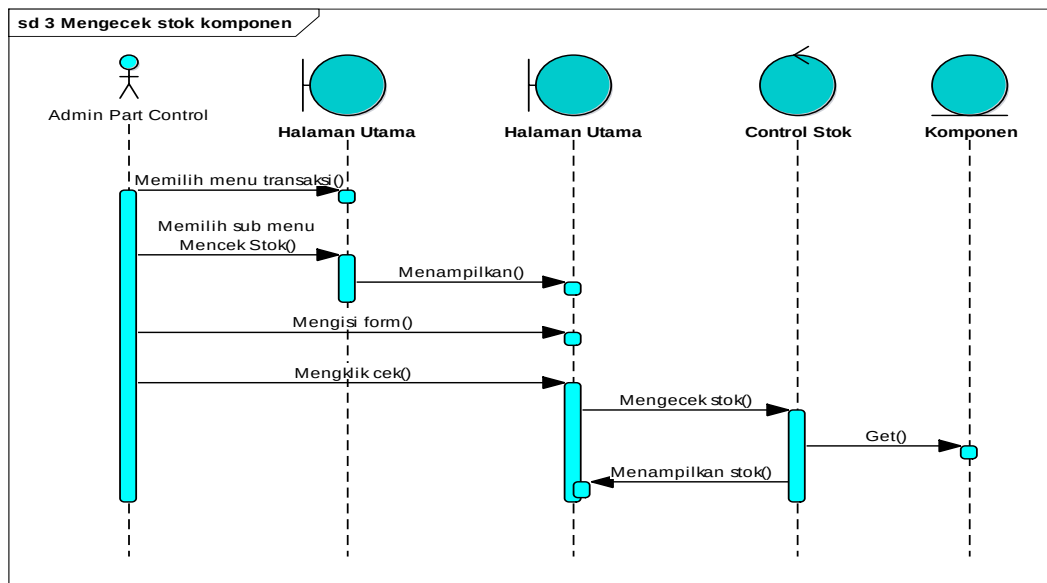
Sequence diagram mengelola data *master* ini menjelaskan interaksi yang dilakukan Admin *Part Control* dengan sistem dalam mengelola data master yaitu data *User*, data *Komponen*, dan data *Vendor* agar data yang ada dalam *database* selalu diperbarui. *Sequence diagram* proses mengelola data *master* dapat dilihat pada Gambar V.13.



Gambar V.13 *Sequence Diagram* Mengelola Data Master
(Sumber: Hasil Analisis, 2018)

3. *Sequence Diagram* Mengecek Stok Komponen

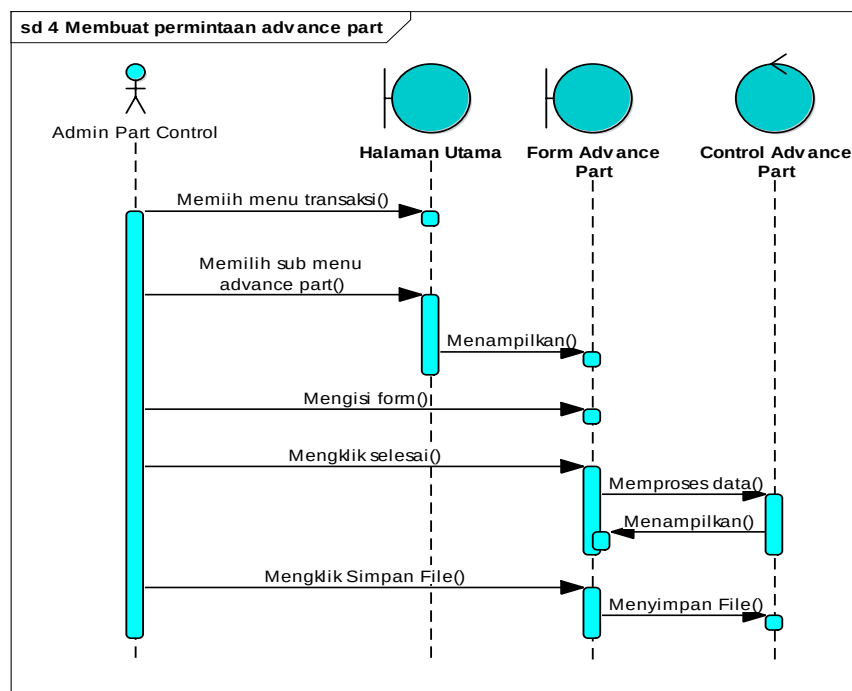
Sequence diagram mengecek stok komponen ini menjelaskan sebuah *sequence diagram* dalam proses mengecek kecukupan stok komponen oleh Admin Part Control. *Sequence diagram* proses mengecek stok komponen dapat dilihat pada Gambar V.14.



Gambar V.14 Sequence Diagram Mengecek Stok Komponen
(Sumber: Hasil Analisis, 2018)

4. Sequence Diagram Membuat Permintaan Advance Part

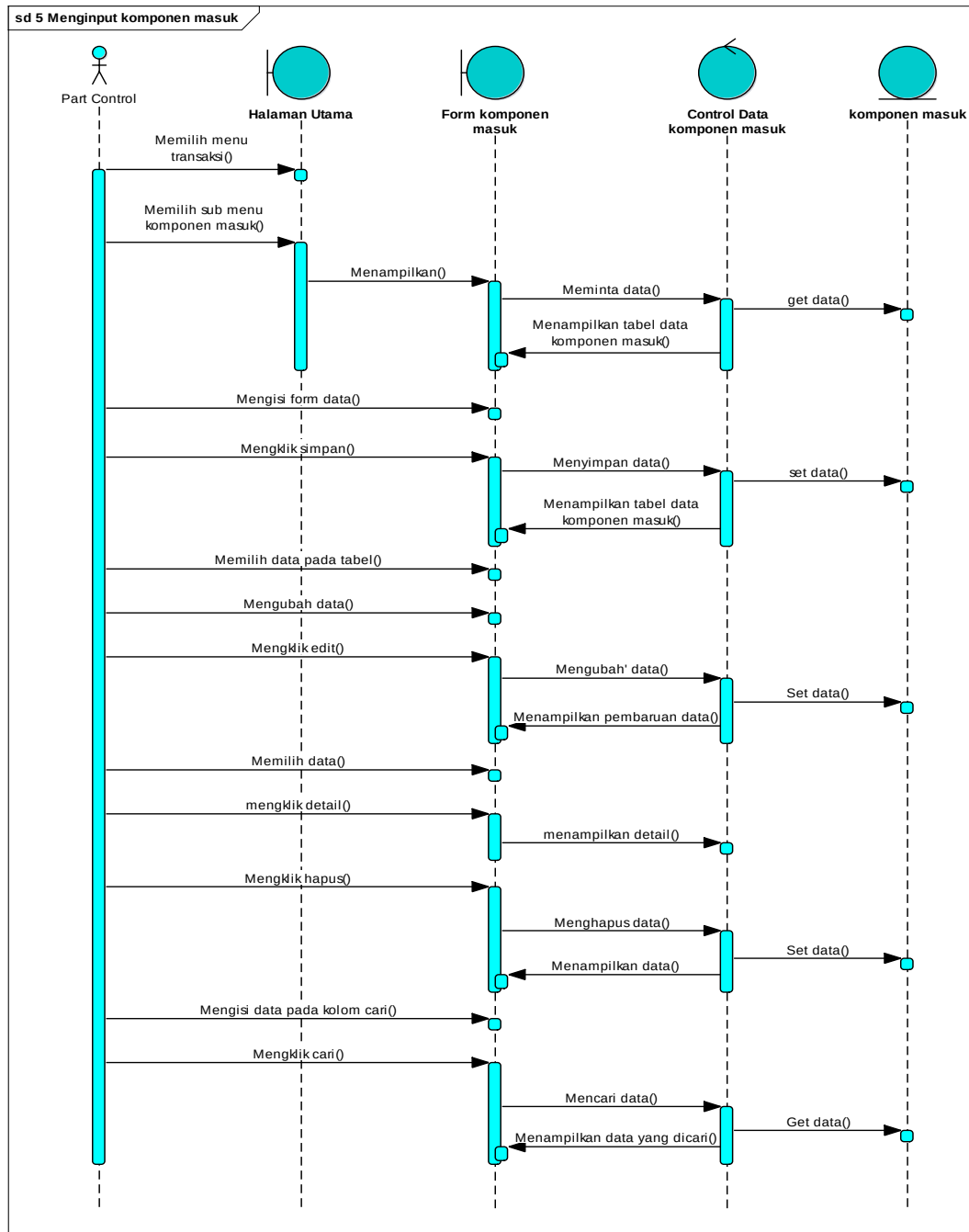
Sequence diagram membuat permintaan *advance part* ini menjelaskan sebuah sequence diagram dalam proses membuat permintaan *advance part* oleh Admin Part Control.



Gambar V.15 Sequence Diagram Membuat Permintaan Advance Part
(Sumber: Hasil Analisis, 2018)

5. Sequence Diagram Menginput komponen masuk

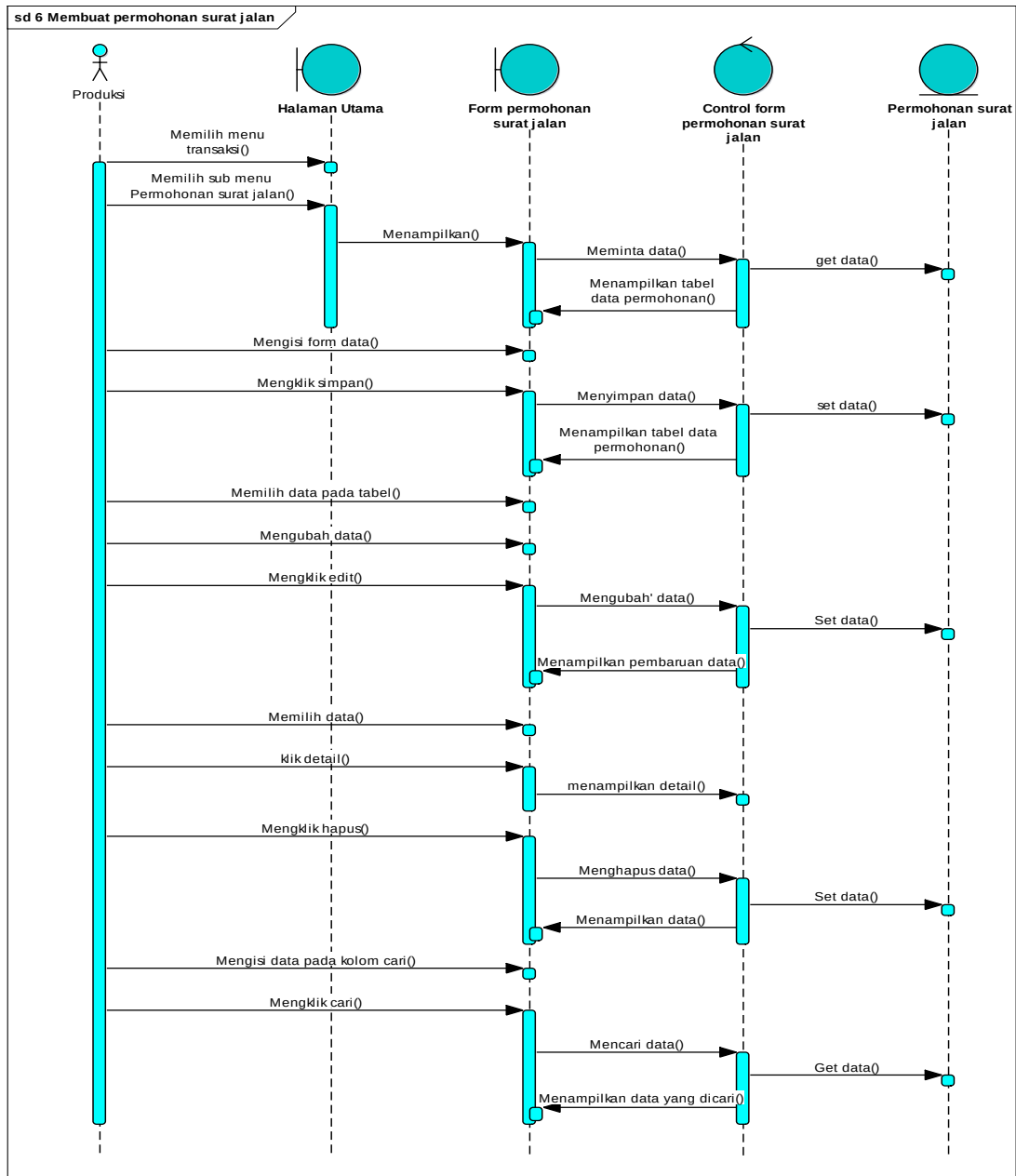
Sequence diagram membuat menginput komponen masuk ini menjelaskan sebuah *sequence diagram* dalam proses menginput komponen masuk oleh *Part Control*. *Sequence diagram* proses menginput komponen masuk dapat dilihat pada Gambar V.16.



Gambar V.16 *Sequence Diagram* Menginput Komponen Masuk
(Sumber: Hasil Analisis, 2018)

6. *Sequence Diagram* Membuat Permohonan Surat Jalan Pengiriman Suplemen

Sequence diagram membuat permohonan surat jalan pengiriman suplemen menjelaskan sebuah *sequence diagram* dalam proses membuat permohonan surat jalan pengiriman suplemen oleh Produksi. *Sequence diagram* proses membuat permohonan surat jalan pengiriman suplemen dapat dilihat pada Gambar V.17.

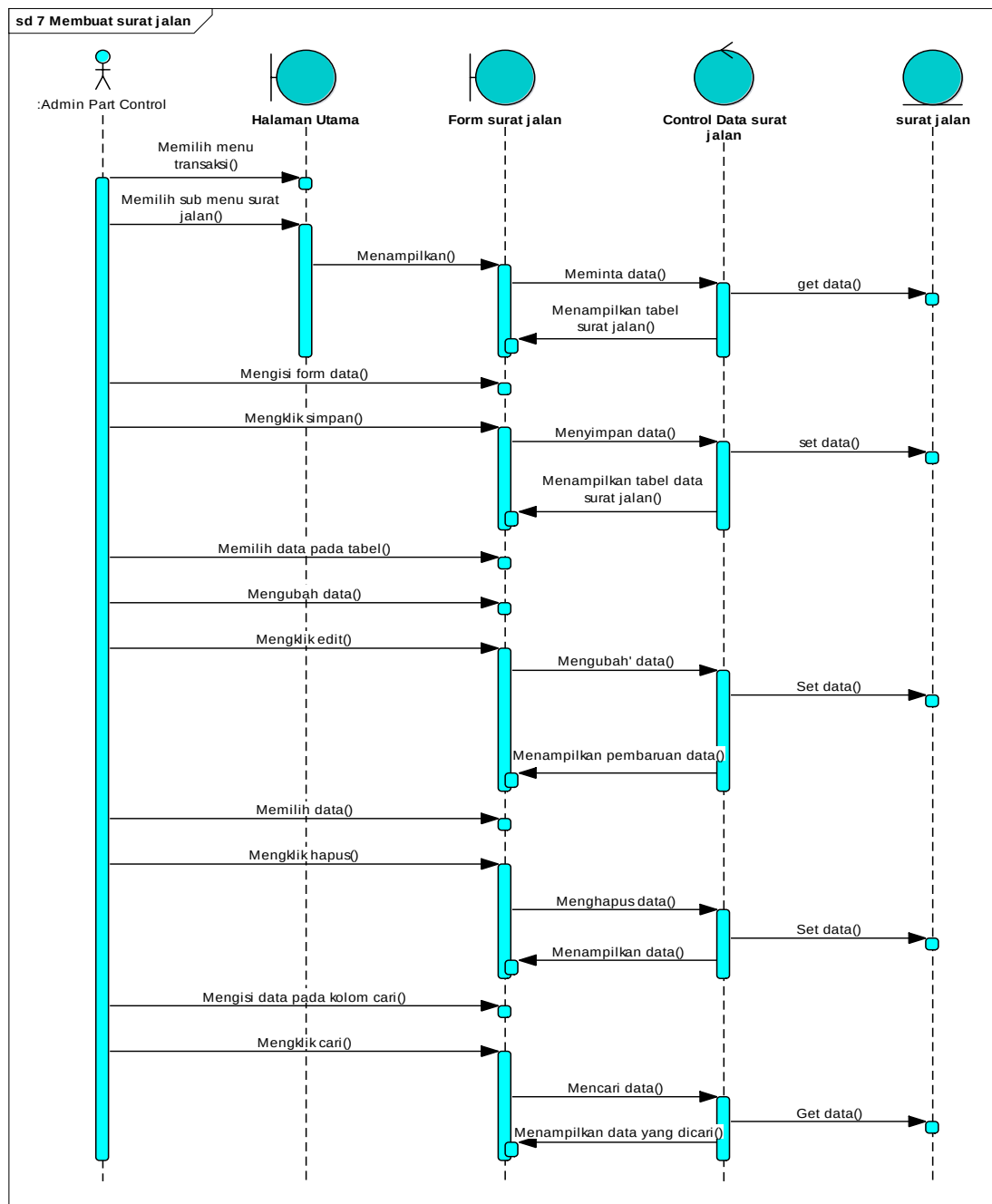


Gambar V.17 *Sequence Diagram* Membuat Permohonan Surat Jalan Pengiriman Suplemen
(Sumber: Hasil Analisis, 2018)

7. *Sequence Diagram* Membuat Surat Jalan Pengiriman Suplemen

Sequence diagram membuat surat jalan pengiriman suplemen menjelaskan sebuah *sequence diagram* dalam proses membuat surat jalan pengiriman suplemen oleh

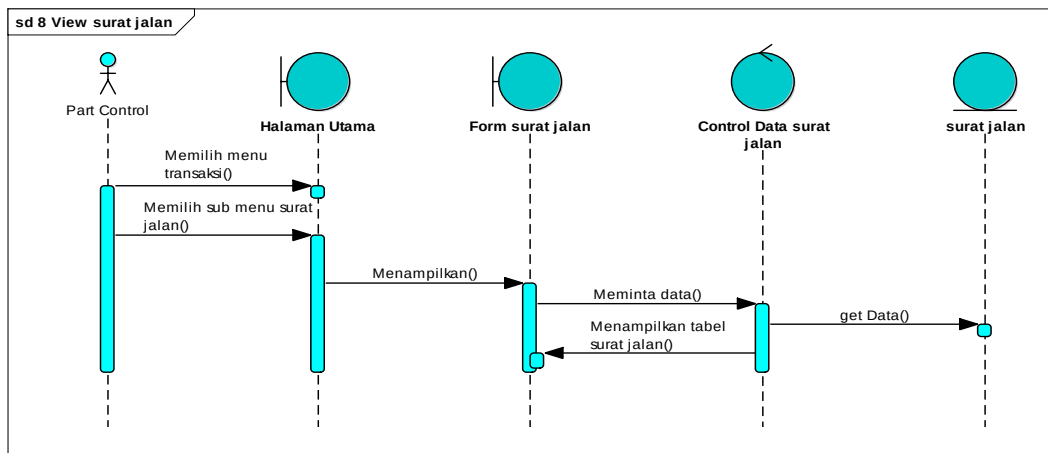
Admin Part Control. Sequence diagram proses membuat surat jalan pengiriman suplemen dapat dilihat pada Gambar V.18.



Gambar V.18 *Sequence Diagram* Membuat Permohonan Surat Jalan Pengiriman Suplemen
(Sumber: Hasil Analisis, 2018)

8. *Sequence Diagram View* Surat Jalan

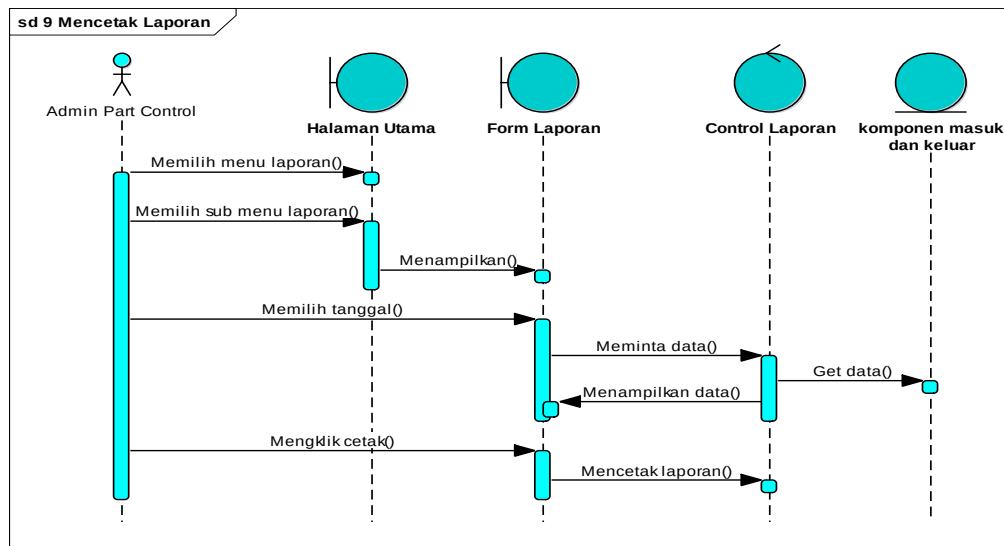
Sequence diagram View Surat Jalan menjelaskan sebuah *sequence diagram* dalam proses melihat surat jalan oleh *Part Control. Sequence diagram* proses *View* Surat Jalan dapat dilihat pada Gambar V.19.



Gambar V.19 Sequence Diagram View Surat Jalan
(Sumber: Hasil Analisis, 2018)

9. Sequence Diagram Mencetak Laporan

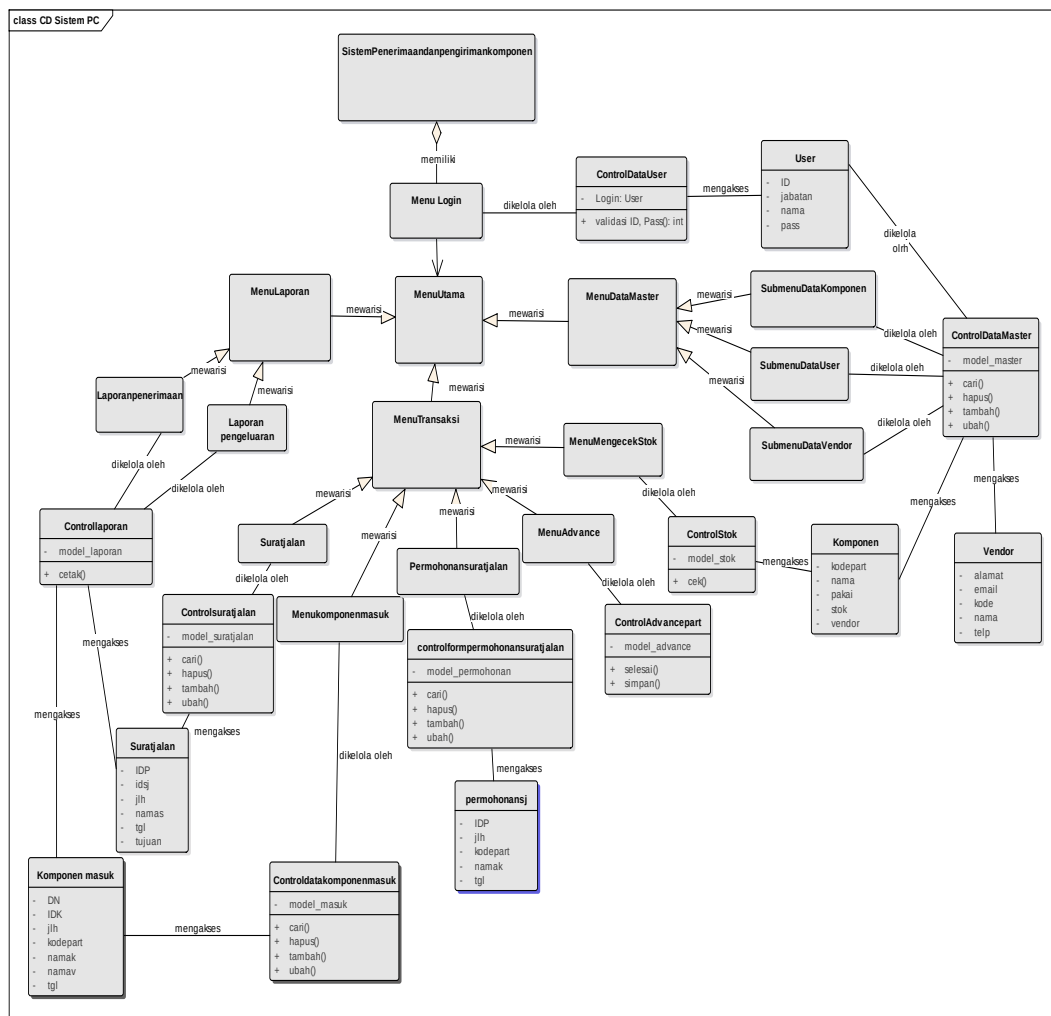
Sequence diagram mencetak laporan menjelaskan sebuah sequence diagram dalam proses mencetak laporan penerimaan dan pengiriman oleh Admin Part Control. Sequence diagram proses mencetak laporan dapat dilihat pada Gambar V.20.



Gambar V.20 Sequence Diagram Mencetak Laporan Komponen Masuk
(Sumber: Hasil Analisis, 2018)

5.3.4 Class Diagram

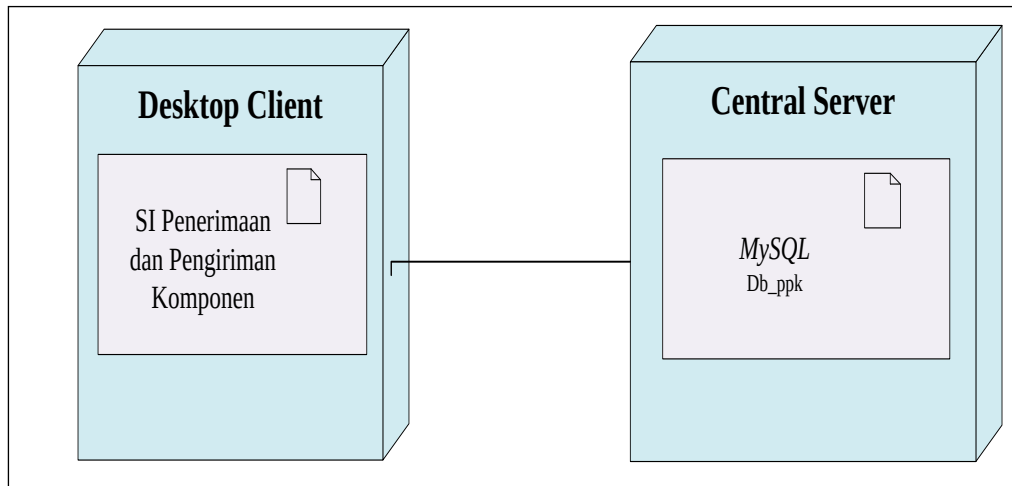
Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Class diagram memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas.



Gambar V.21 *Class Diagram* Sistem Informasi Penerimaan dan Pengiriman Komponen Usulan
(Sumber: Hasil Analisis, 2018)

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 pengiriman komponen yang diusulkan dapat dilihat pada Gambar V.22.



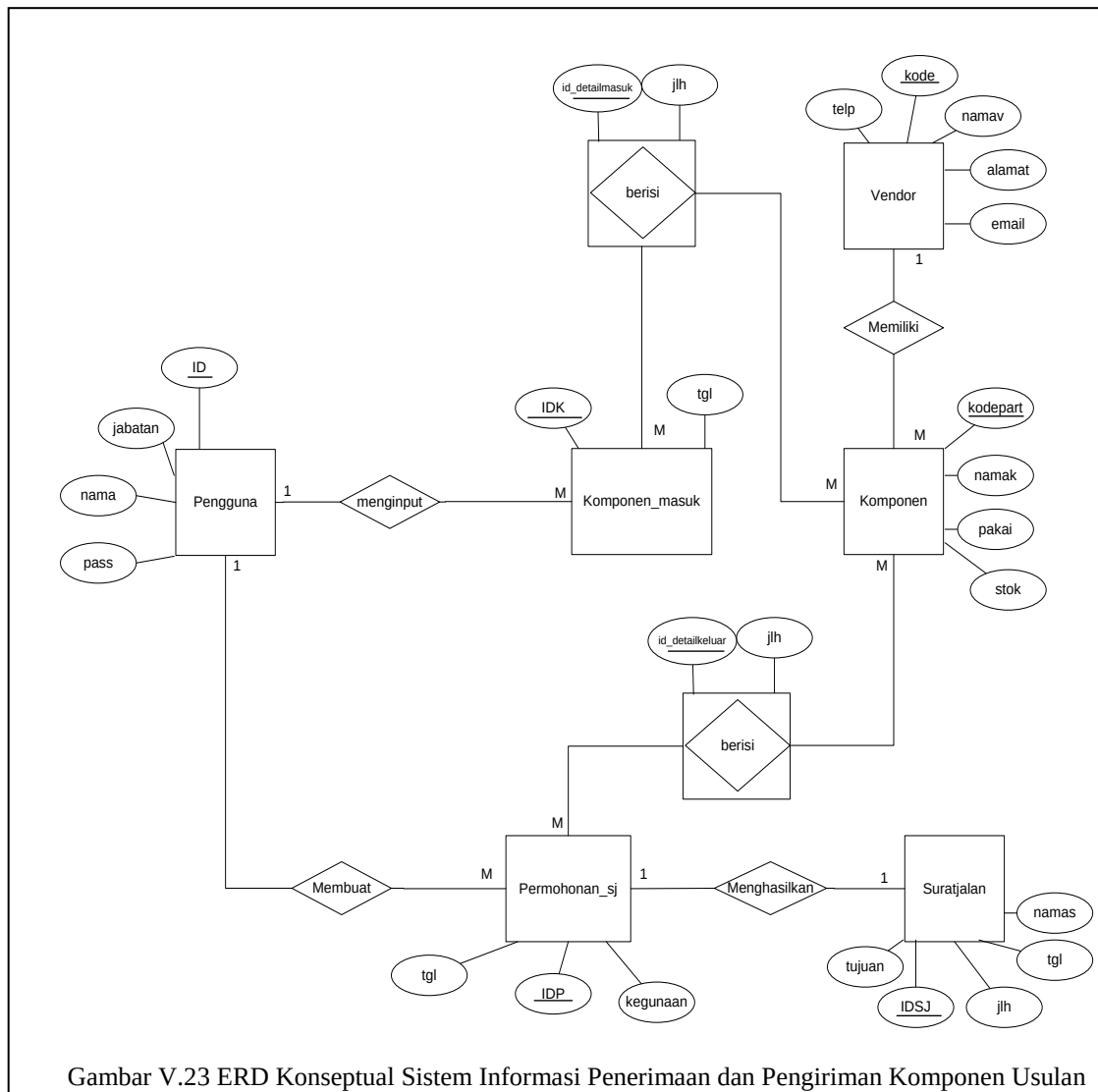
Gambar V.22 *Deployment Diagram* Sistem Informasi Penerimaan dan Pengiriman Komponen Usulan
(Sumber: Hasil Analisis, 2018)

5.4 Pemodelan Data

Pemodelan data merupakan metode yang digunakan untuk menentukan dan menganalisis persyaratan data yang diperlukan untuk mendukung proses bisnis suatu organisasi. Pemodelan data ada 2. Yaitu:

5.4.1. *Entity Relationship Diagram (ERD)*

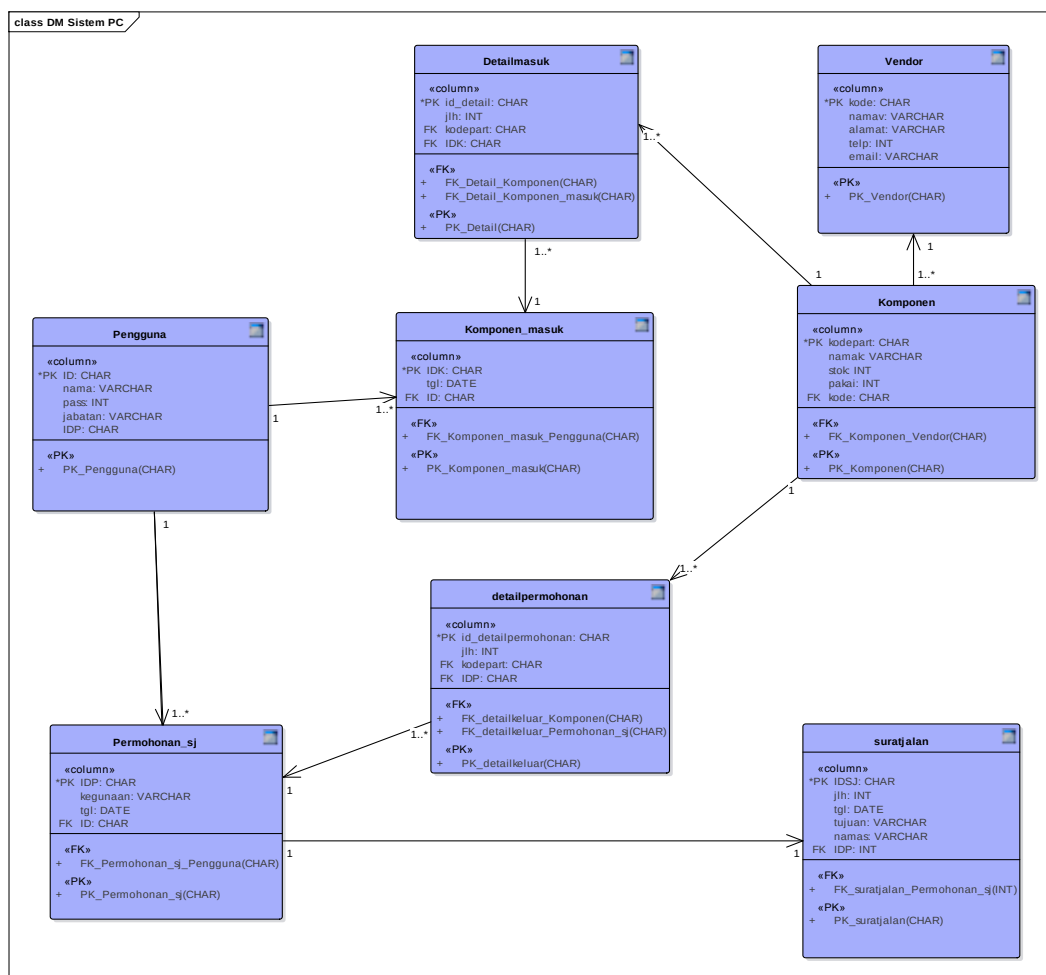
ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarannya digunakan beberapa notasi dan simbol. ERD sistem informasi penerimaan dan pengiriman komponen yang diusulkan dapat dilihat pada Gambar V.23.



Gambar V.23 ERD Konseptual Sistem Informasi Penerimaan dan Pengiriman Komponen Usulan
(Sumber: Hasil Analisis, 2018)

5.4.2. Conceptual Data Model (CDM)

CDM (*Conceptual Data Model*) merupakan penjabaran lebih lanjut dari ERD. Pada gambar V.24 adalah CDM Sistem Informasi Penerimaan dan Pengiriman Komponen.



Gambar V.24 CDM Sistem Informasi Penerimaan dan Pengiriman Komponen Usulan
(Sumber: Hasil Analisis, 2018)

5.5 Kamus Data

Kamus data adalah suatu daftar data elemen 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 Pengiriman Komponen Usulan.

1. Spesifikasi Tabel Pengguna

Nama Tabel : Pengguna
Fungsi : Untuk menyimpan data pengguna
Tipe : File data master

Tabel V.13 Tabel Pengguna

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
----	-------------	---------	------	---------	------------

1.	ID	ID_Pengguna	<i>Char</i>	10	<i>Primary Key</i>
2.	nama	nama_Pengguna	<i>varchar</i>	20	<i>Not Null</i>
3.	pass	Password	<i>Int</i>	7	<i>Not Null</i>
4.	jabatan	jabatan	<i>varchar</i>	10	<i>Not Null</i>

(Sumber: Hasil Analisis, 2018)

2. Spesifikasi Tabel Komponen

Nama Tabel : Komponen

Fungsi : Untuk menyimpan data komponen

Tipe : File data master

Tabel V.14 Tabel Komponen

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	kodepart	Kodekomponen	<i>Char</i>	10	<i>Primary Key</i>
2.	kode	kodeVendor	<i>Char</i>	10	<i>Foreign Key</i>
3.	namak	namakomponen	<i>varchar</i>	20	<i>Not Null</i>
4.	stok	jumlah	<i>Int</i>	3	<i>Not Null</i>
5.	pakai	penggunaan	<i>Int</i>	3	<i>Not Null</i>

(Sumber: Hasil Analisis, 2018)

3. Spesifikasi Tabel Vendor

Nama Tabel : *Vendor*

Fungsi : Untuk menyimpan data *Vendor*

Tipe : File data master

Tabel V.15 Tabel *Vendor*

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	kode	KodeVendor	<i>Char</i>	10	<i>Primary Key</i>
2.	namav	namavendor	<i>varchar</i>	20	<i>Not Null</i>
3.	telp	telepon	<i>Int</i>	12	<i>Not Null</i>

Tabel V.15 Tabel *Vendor*(Lanjutan)

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
4.	alamat	AlamatVendor	<i>varchar</i>	20	<i>Not Null</i>
5.	<i>Email</i>	<i>Email</i>	<i>varchar</i>	20	<i>Not Null</i>

(Sumber: Hasil Analisis, 2018)

4. Spesifikasi Tabel Komponen Masuk

Nama Tabel : *Komponen_masuk*

Fungsi : Untuk menyimpan data komponen masuk

Tipe : File data transaksi

Tabel V.16 Tabel Komponen Masuk

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	IDK	ID_komponenmasuk	Char	10	Primary Key
2.	ID	ID_user	Char	10	Foreign Key
3.	tgl	Tanggal_masuk	Date		Not Null

(Sumber: Hasil Analisis, 2018)

5. Spesifikasi Tabel Detail Komponen Masuk

Nama Tabel : Detailmasuk

Fungsi : Untuk menyimpan data detail komponen masuk

Tipe : File data transaksi

Tabel V.17 Tabel Detail Komponen Masuk

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	Id_detail	ID_detail	Char	10	Primary Key
2.	kodepart	kodekomponen	Char	10	Foreign Key
3.	IDK	ID_komponenmasuk	Char	10	Foreign Key
4.	jlh	jumlah	Int	10	Not Null

(Sumber: Hasil Analisis, 2018)

6. Spesifikasi Tabel Permohonan Surat Jalan Pengiriman Suplemen

Nama Tabel : Permohonan_sj

Fungsi : Untuk menyimpan data Permohonan Surat Jalan

Tipe : File data transaksi

Tabel V.18 Tabel Permohonan Surat Jalan Pengiriman Suplemen

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	IDP	ID_Permohonan	Char	10	Primary Key
2.	ID	ID_User	Char	10	Foreign Key
3.	tgl	Tgl_permohonan	Date		Not Null
4.	Kegunaan	Deskripsi	varchar	20	Not Null

(Sumber: Hasil Analisis, 2018)

7. Spesifikasi Tabel Detail Permohonan

Nama Tabel : Detailpermohonan

Fungsi : Untuk menyimpan data detail permohonan surat jalan

Tipe : File data transaksi

Tabel V.19 Tabel Detail Permohonan

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	Id_detailpermohonan	ID_detailpermohonan	Char	10	Primary Key
2.	kodepart	kodekomponen	Char	10	Foreign Key
3.	IDP	ID_Permohonan	Char	10	Foreign Key
4.	jlh	jumlah	Int	10	Not Null

(Sumber: Hasil Analisis, 2018)

8. Spesifikasi Tabel Surat Jalan

Nama Tabel : suratjalan

Fungsi : Untuk menyimpan data Surat Jalan

Tipe : File data transaksi

Tabel V.20 Tabel Surat Jalan

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	IDSJ	ID_Suratjalan	Char	10	Primary Key
2.	IDP	ID_Permohonan	Char	10	Foreign Key
3.	jlh	jumlah	Int	3	Not Null
4.	tgl	Tanggal	Date		Not Null
5.	tujuan	alamat	varchar	20	Not Null
6.	namas	Nama_suplemen	varchar	20	Not Null

(Sumber: Hasil Analisis, 2018)

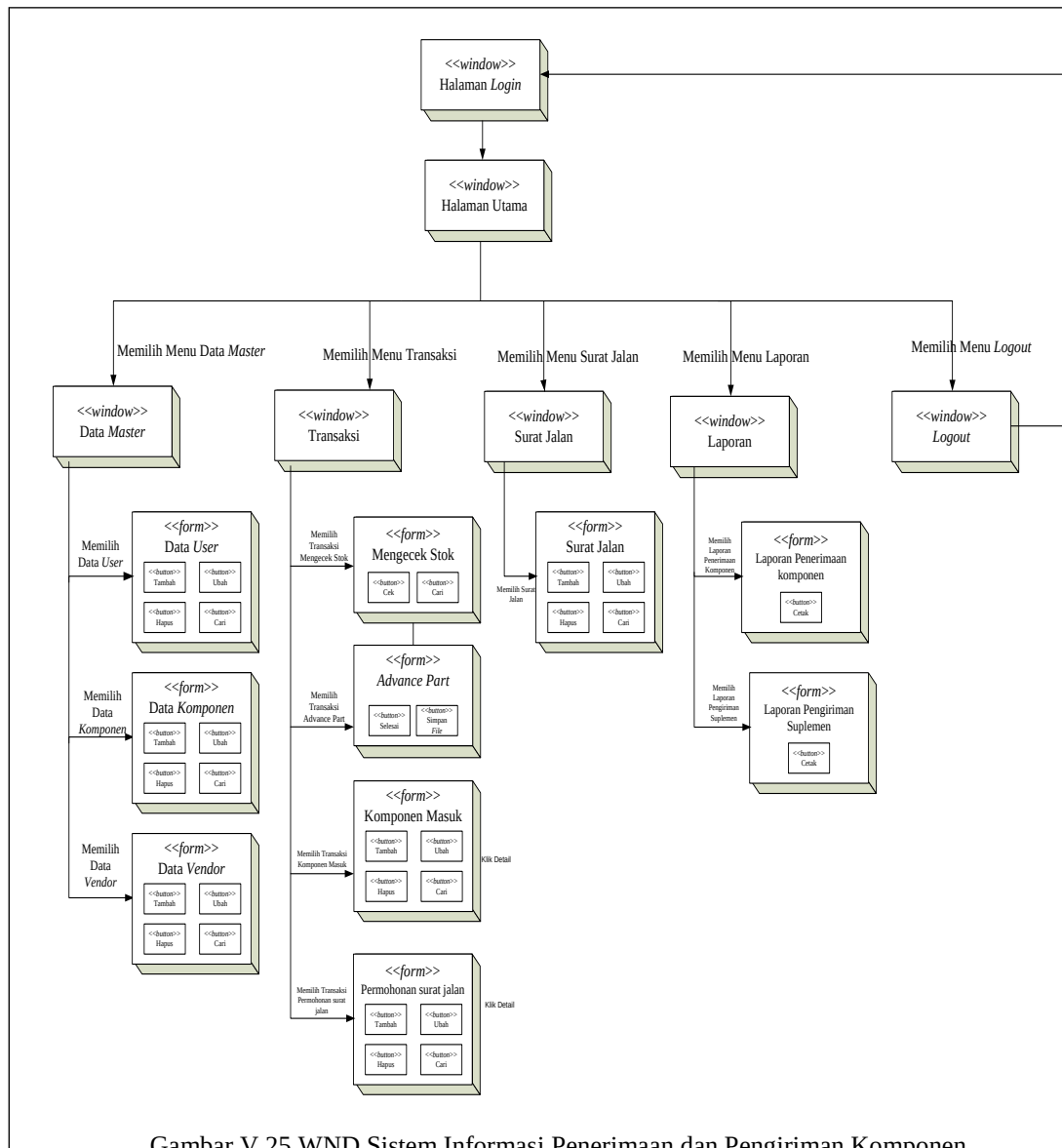
5.6 Analisis Desain Program

Tahap ini merupakan tahap kedua dalam metodologi prototipe evolusioner, yaitu tahap membuat sebuah prototipe dari program atau aplikasi. Dimulai dengan analisis desain usulan meliputi pembuatan struktur menu program dan *interface* program sampai dengan program dapat dijalankan.

5.6.1 WND (Windows Navigation Diagram)

Windows Navigation Diagram (WND) digunakan untuk mendokumentasikan sebuah struktur yang menggambarkan hubungan antar fungsi dalam program secara

hirarkis. Diagram ini memuat semua modul judul yang ada dalam sistem beserta nama dan nomornya. Diagram ini menunjukkan *window-window* tersebut. Sebuah *window* dapat digambarkan sebagai sebuah *state*. *State* ini memiliki nama dan berisi gambar *miniature window*. Transisi antar *state* dipicu oleh ditekannya sebuah tombol yang menghubungkan dua *window*. Perancangan WND aplikasi usulan sistem informasi penerimaan dan pengiriman komponen dapat dilihat pada Gambar V.25



Gambar V.25 WND Sistem Informasi Penerimaan dan Pengiriman Komponen

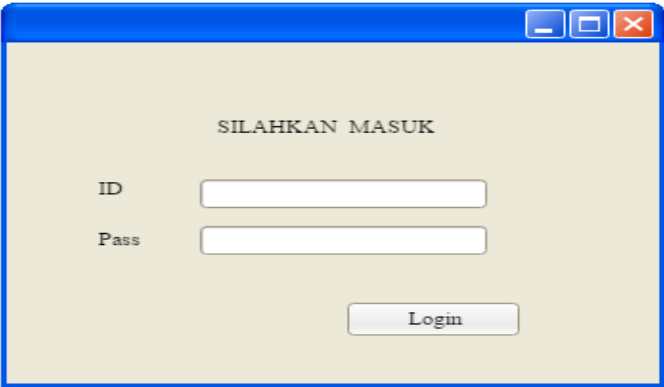
(Sumber: Hasil Analisis, 2018)

5.6.2 Perancangan *Interface* Sistem

Rancangan *interface* dari program penerimaan dan pengiriman komponen ini adalah sebagai berikut:

1. Form Login

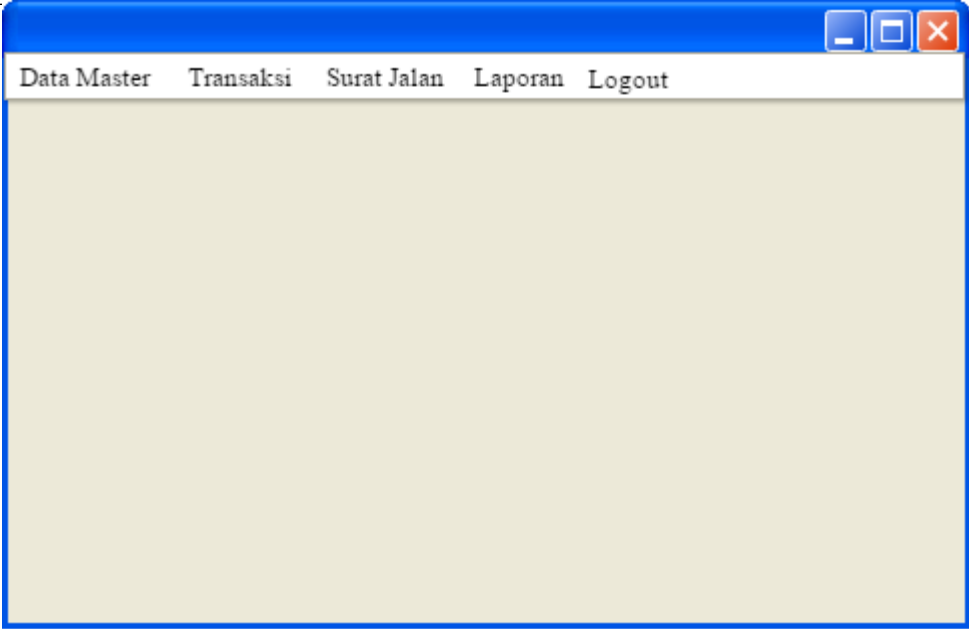
Form login adalah *form* yang digunakan untuk masuk ke dalam program aplikasi. Untuk masuk ke dalam aplikasi, pengguna harus memasukkan ID dan *password* yang benar. Rancangan *form login* dapat dilihat pada Gambar V.26.



Gambar V.26 *Interface Login*
(Sumber: Hasil Analisis, 2018)

2. Form Menu Utama

Form menu utama adalah tampilan awal pada aplikasi yang berisi empat menu. Menu tersebut adalah Data Master, Transaksi, Surat Jalan dan laporan.



Gambar V.27 *Interface Login*
(Sumber: Hasil Analisis, 2018)

3. Form Data Komponen

Form data komponen adalah *form* yang digunakan untuk mengelola data komponen. Rancangan *interface* dari *form* data komponen dapat dilihat pada Gambar V.28 berikut:

The screenshot shows a software window titled "Form Data Master". On the left, under the heading "DATA KOMPONEN", there are five input fields: "Kode Komponen", "Nama Komponen", "Stok", "Kebutuhan", and "Vendor". The "Vendor" field is a dropdown menu currently showing "pilih". To the right of these fields are four buttons: "Simpan", "Hapus", "Edit", and "Cari". Below the buttons is a table with five columns: "Kode Komponen", "Nama Komponen", "Stok", "Kebutuhan", and "Vendor". The table has three rows, with the first row being a header and the next two rows being empty data rows.

Kode Komponen	Nama Komponen	Stok	Kebutuhan	Vendor

Gambar V.28 *Interface Form Data Master Komponen*
(Sumber: Hasil Analisis, 2018)

Pada *interface form data master* komponen terdapat empat tombol, yaitu cari, tambah, ubah dan hapus. Tombol cari digunakan untuk mencari data pelanggan yang diinginkan. Tombol simpan atau ubah digunakan untuk menyimpan data komponen ke dalam database. Sedangkan tombol hapus digunakan untuk menghapus data komponen yang dipilih. Data *master* komponen, dikelola oleh Admin Part Control.

4. *Form Data Vendor*

Form data vendor adalah *form* yang digunakan untuk mengelola data *vendor*. Rancangan *interface* dari *form data vendor* dapat dilihat pada Gambar V.29 berikut:

Form Data Master

DATA VENDOR

Kode

Nama Vendor

No. telp

Email

Alamat

LOGO

Simpan Hapus Edit

Cari

Kode	Nama Vendor	No. Telp	Email	Kode Part	Alamat

Gambar V.29 Interface Form Data Master Vendor
(Sumber: Hasil Analisis, 2018)

Pada *interface form data master vendor* terdapat empat tombol, yaitu cari, tambah, ubah dan hapus. Tombol cari digunakan untuk mencari data pelanggan yang diinginkan. Tombol simpan atau ubah digunakan untuk menyimpan data *vendor* ke dalam database. Sedangkan tombol hapus digunakan untuk menghapus data *vendor* yang dipilih. Data *master vendor*, dikelola oleh Admin *Part Control*.

5. Form Data User

Form data user adalah *form* yang digunakan untuk mengelola data *user*. Rancangan *interface* dari *form data user* dapat dilihat pada Gambar V.30 berikut:

Form Data Master

DATA USER

ID

Nama

Pass

Jabatan ☐ Admin ☐ Staff

LOGO

Simpan Hapus Edit

Cari

ID	Nama	Password	Jabatan

Gambar V.30 *Interface Form Data Master User*
(Sumber: Hasil Analisis, 2018)

Pada *interface form data master user* terdapat empat tombol, yaitu cari, tambah, ubah dan hapus. Tombol cari digunakan untuk mencari data pelanggan yang diinginkan. Tombol simpan atau ubah digunakan untuk menyimpan data *user* ke dalam database. Sedangkan tombol hapus digunakan untuk menghapus data *user* yang dipilih. Data *master user*, dikelola oleh *Admin Part Control*.

6. *Form Data Mengecek Stok*

Form data mengecek stok adalah *form* yang digunakan untuk mengecek stok komponen yang tersedia dengan cara mengisi data target produksi dan mengklik tombol cek. Rancangan *interface* dari *form* mengecek stok dapat dilihat pada Gambar V.31 berikut:

Form Transaksi

LOGO

MENGECEK STOK

Target Produksi

Kode Part	Nama Komponen	Nama Vendor	Stok

Gambar V.31 *Interface Form Mengecek Stok Komponen*
(Sumber: Hasil Analisis, 2018)

7. *Form Data Advance Part*

Form Advance Part adalah *form* yang digunakan untuk membuat surat *Advance Part* kepada *vendor*. *Form* ini dikelola oleh *Admin Part Control*. Rancangan *interface* dari *form data Advance Part* dapat dilihat pada Gambar V.32 berikut:

Form Transaksi

LOGO

ADVANCE PART

Purchasing Doc OLD

Kode Komponen Qty Pesan

Nama Komponen Advance

Vendor Qty Adv

Next Simpan File

Purchasing Doc	Kode Komponen	Nama Komponen	Vendor	OLD	Qty Pesan	Advance	Qty Adv

Gambar V.32Interface Form Advance Part
(Sumber: Hasil Analisis, 2018)

8. Form Komponen Masuk

Form komponen masuk adalah *form* yang digunakan untuk mengelola data komponen yang masuk di bagian *Part Control*. Pada *interface form* komponen masuk terdapat empat tombol, yaitu cari, tambah, ubah dan hapus. Tombol cari digunakan untuk mencari data komponen masuk yang diinginkan. Tombol simpan atau ubah digunakan untuk menyimpan data komponen masuk ke dalam database. Sedangkan tombol hapus digunakan untuk menghapus data komponen masuk yang dipilih. Data komponen masuk, dikelola oleh *Part Control*. Rancangan *interface* dari *form* komponen masuk dapat dilihat pada Gambar V.33 berikut:

Form Transaksi

KOMPONEN MASUK

ID Komponen Masuk

Delivery Note

Kode Komponen

Nama Komponen

Vendor

Jumlah

Tanggal

LOGO

ID Komponen	Kode Komponen	Nama Komponen	Vendor	Jumlah	Tanggal

Gambar V.33 *Interface Form* Komponen Masuk
(Sumber: Hasil Analisis, 2018)

9. *Form* Permohonan Surat Jalan

Form permohonan surat jalan adalah *form* yang digunakan untuk membuat permohonan surat jalan dari bagian produksi ke admin *part control*. Pada *interface form* permohonan surat jalan terdapat empat tombol, yaitu cari, tambah, ubah dan hapus. Tombol cari digunakan untuk mencari data permohonan surat jalan yang diinginkan. Tombol simpan atau ubah digunakan untuk menyimpan data permohonan surat jalan ke dalam database. Sedangkan tombol hapus digunakan untuk menghapus permohonan surat jalan yang dipilih. Data permohonan surat jalan, dikelola oleh Produksi. Rancangan *interface* dari *form* permohonan surat jalan dapat dilihat pada Gambar V.34 berikut:

Form Transaksi

PERMOHONAN SURAT JALAN

ID Permohonan

Kode Komponen

Tanggal

Jumlah

Kegunaan

LOGO

ID Permohonan	Kode Komponen	Tanggal	Jumlah	Kegunaan

Gambar V.34 *Interface Form* Permohonan Surat Jalan
(Sumber: Hasil Analisis, 2018)

10. *Form* Surat Jalan

Form surat jalan adalah *form* yang digunakan untuk membuat surat jalan. Pada *interface form* surat jalan terdapat empat tombol, yaitu cari, tambah, ubah dan hapus. Tombol cari digunakan untuk mencari data surat jalan yang diinginkan. Tombol simpan atau ubah digunakan untuk menyimpan data surat jalan ke dalam database. Sedangkan tombol hapus digunakan untuk menghapus data surat jalan yang dipilih. Data surat jalan, dikelola oleh Admin *Part Control*. Rancangan *interface* dari *form* surat jalan dapat dilihat pada Gambar V.35 berikut:

Form Transaksi

SURAT JALAN

ID Surat Jalan

Jumlah

Tanggal

Tujuan

Nama Suplemen

LOGO

ID Surat Jalan	Jumlah	Tanggal	Tujuan	Nama Suplemen

Gambar V.35 *Interface Form Surat Jalan*
(Sumber: Hasil Analisis, 2018)

11. *Form Laporan Penerimaan Komponen*

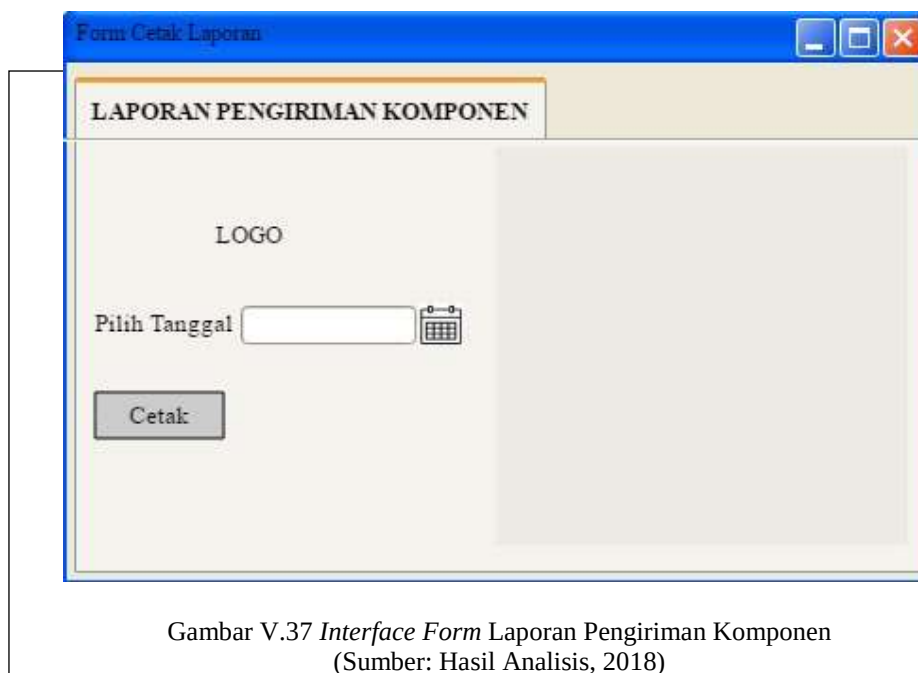
Form Laporan Penerimaan Komponen adalah *form* yang digunakan untuk mencetak laporan penerimaan komponen pada bagian *Part Control* dengan memilih tanggal laporan yang ingin dicetak. *Form* ini dikelola oleh Admin *Part Control* Rancangan *interface* dari *form* laporan penerimaan komponen dapat dilihat pada Gambar V.36 berikut:



Gambar V.36 *Interface Form Laporan Penerimaan Komponen*
(Sumber: Hasil Analisis, 2018)

12. Form Laporan Pengiriman Komponen

Form Laporan Pengiriman Komponen adalah form yang digunakan untuk mencetak laporan pengiriman komponen dengan memilih tanggal laporan yang ingin dicetak. Form ini dikelola oleh Admin *Part Control*. Rancangan *interface* dari form laporan pengiriman komponen dapat dilihat pada Gambar V.37 berikut:



Gambar V.37 *Interface Form Laporan Pengiriman Komponen*
(Sumber: Hasil Analisis, 2018)

5. 7 Implementasi Sistem *Software* dan *Hardware*

Tahap ini adalah tahap pengkodean program menggunakan bahasa pemrograman *Java Netbeans*. Setiap *interface* berisikan kode program agar program

dapat dijalankan sesuai dengan fungsinya. Untuk mendukung kebutuhan implementasi sistem diperlukan suatu spesifikasi perangkat lunak (*software*) maupun perangkat keras (*hardware*). Adapun spesifikasinya sebagai berikut:

1. Analisis Kebutuhan *Software*

- a. Sistem Operasi : *Microsoft Windows 7*
- b. *Database Server* : MySQL versi 5.5.32
- c. Bahasa Pemrograman : Java
- d. *IDE Platform* : *NetBeans IDE 8.0.2*

2. Analisis Kebutuhan *Hardware*

- a. *Processor* : Minimal *Processor Pentium IV*
- b. RAM : Minimal RAM 512 MB
- c. *Harddisk* : Minimal *Harddisk 64 GB*
- d. *Mouse, Keyboard, Monitor* sebagai peralatan antar muka.
- e. *Printer* sebagai media pencetakan dokumen.

Kebutuhan *non functional requirements* adalah secara umum berisi batasan-batasan pada pelayanan atau fungsi yang disediakan oleh sistem. Kebutuhan *non functional requirements* sistem informasi penerimaan dan pengiriman komponen usulan terdapat 4 yaitu:

- 1. Sistem hanya dapat digunakan di sistem operasi *windows*.
- 2. Untuk dapat masuk kedalam sistem ini, Pengguna harus melakukan *login* terlebih dahulu menggunakan ID dan *Password*.
- 3. Sistem hanya membahas penerimaan dan pengiriman komponen di bagian *Part Control*, tidak membahas pembelian komponen.
- 4. Sistem hanya menyediakan dalam satu bahasa saja yaitu bahasa Indonesia.

BAB VI

KESIMPULAN DAN SARAN

6.3 Kesimpulan

Berdasarkan hasil penelitian selama proses pengumpulan dan pengolahan data, analisis sistem, dan implementasi sistem usulan maka dapat diambil kesimpulan sebagai berikut:

1. Dengan adanya aplikasi sistem informasi penerimaan dan pengiriman komponen ini proses pembuatan surat jalan suplemen di bagian *Part Control* menjadi lebih cepat dan *simple* karena sudah terintegrasi dengan bagian produksi serta memudahkan admin *Part Control* dalam membuat laporan penerimaan dan pengiriman komponen.
2. Dengan adanya aplikasi sistem informasi penerimaan dan pengiriman komponen yang terintegrasi dengan database memberikan fasilitas penyimpanan data yang aman sehingga dapat mencegah terjadinya kehilangan data dan mempercepat bagian *Part Control* PT. Krama Yudha Ratu Motor dalam melakukan pencarian data.

6.4 Saran

Saran yang diberikan untuk pengembangan sistem informasi penerimaan dan pengiriman komponen ini selanjutnya adalah sebagai berikut:

1. Untuk penerapan sistem baru terhadap sistem lama, sebaiknya dilakukan secara bertahap dan diperlukan sosialisasi penggunaan sistem ini kepada bagian yang terkait.
2. Pengembangan aplikasi selanjutnya sistem informasi ini bisa diintegrasikan dengan bagian PPC dan dapat menambahkan fitur *notification* apabila ada data transaksi baru.

DAFTAR PUSTAKA

- Amirullah. 2015. *Pengantar Manajemen*. Jakarta: Mitra Wacana Media.
- Assauri, Sofyan. 2016. *Manajemen Operasi Produksi*, Edisi Ketiga. Jakarta: Rajawali Pers.
- Buntarto. 2015. *Pengenalan Bodi Otomotif*. Cetakan Pertama. Yogyakarta: Pustaka Baru Press.
- Dennis, Alan. 2010. *System Analysis and Design with UML*. United States of America: John Wiley and Sons, Inc.
- Dennis, Alan., Wixom, Barbara Haley., Tegarden, David. 2005. *System Analysis and Design an Object Oriented Approach with UML version 2.0*. New Jersey: John Wiley & Sons, inc.
- Fathansyah. 2015. *Basis Data*. Revisi kedua. Bandung: Informatika.
- Gata, Windu. 2013. *Sukses Membangun Aplikasi Penjualan dengan Java*. Jakarta: Elex Media.
- Gelinas, Ulrich., dan Dull, B. Richard. 2012. *Accounting Information Systems*, 9th ed. South Western Cengage Learning. 5191 Natorp Boulevard Mason, USA. P. 19.
- Hoffer, Jeffrey A., Ramesh, V., dan Topi, Heikki. 2010. *Modern Database Management 10th Edition*. New Jersey: Pearson Education.
- Irawan, 2007. *Dasar Pemrograman Java*. Palembang: Maxikom.
- Jogiyanto, HM. 2005. *Analisis dan Desain Sistem Informasi*. Yogyakarta: Andi Offset.
- Kadir, Abdul. 2014. *Pengenalan Sistem Informasi. Edisi Revisi*. Yogyakarta: Andi Offset.
- Kendall, Kenneth E. 2003. *Analisis dan Perancangan Sistem*. Jakarta: Indeks.
- Kotler, Philip dan Armstrong, Gary. 2004. *Dasar-Dasar Pemasaran edisi ke-9*. Jakarta : PT Indeks.
- McLeod, R. dan Scholl, G. 2008. *Sistem Informasi Manajemen Edisi 10*. Jakarta: Salemba Empat.
- Mulyadi. 2010. *Sistem Akuntansi*. Edisi ketiga, Cetakan ke-5. Jakarta: Salemba.
- Mahyudanil, Novri Asyara. 2014. *Rancang Bangun Mobile Commerce Berbasis Android Smartphone dengan Extreme Programming* (Studi Kasus: PT GBI).

- Fakultas Sains dan Teknologi. Universitas Islam Negeri Syarif Hidayatullah. Jakarta.
- Mustakini, Jogiyanto Hartono. 2012. *Analisis dan Disain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi Offset.
- Pratama, I Putu dan Agus Eka. 2014. *Sistem Informasi dan Implementasinya*. Cetakan Pertama. Bandung: Informatika.
- Pressman, Roger. 2002. *Rekayasa Perangkat Lunak Pendekatan Praktisi*. Yogyakarta: Andi Offset.
- Rosa, A.S., Shalahuddin, M. 2011. *Modul Pembelajaran Rekayasa Perangkat Lunak*. Bandung, Modula.
- Sommerville, I. 2003. *Software Engineering RPL*. Edisi Keenam. Jakarta: Erlangga.
- Sugiyono. 2013. *Metodelogi Penelitian Kuantitatif, Kualitatif, dan R&B*. Bandung: Alfabeta.
- Sutabri, Tata. 2004. *Analisa Sistem Informasi*. Yogyakarta: Andi Offset.
- Sutaji, Deni. 2012. *Sistem Inventory Mini Market dengan PHP & JQuery*. Yogyakarta: Lokomedia.
- Tasmawati, Mila. 2008. *Aplikasi Konversi Regular Grammer menjadi Ekspresi Regular dengan Menggunakan Bahasa Pemrograman Java*. Fakultas Teknologi Industry. Universitas Gunadarma. Depok.
- Valacich, George. dan Hoffer. 2012. *Essentials of Systems Analysis and Design, 5th Edition*. U.S.A.: Pearson.
- Welling, L. dan Thomson. 2003. *PHP and MySql Web Development, Second Edition*. Indianapolis: Sams Publishing.
- Wilkinson, Joseph W. 2012. *Sistem Akunting dan Informasi, Edisi Ketiga Jilid Satu*. Jakarta: Binarupa Aksara.
- Zakiyudin, Ais. 2012. *Sistem informasi manajemen edisi 2*. Jakarta: Mitra Wacana Media.
- _____. "NetBeans". <https://en.wikipedia.org/wiki/NetBeans>. (Tanggal Akses: 10 Mei 2018).

LAMPIRAN A

ANTAR MUKA APLIKASI

1. Login



2. Menu Utama



3. Menu Data Master

FORM DATA MASTER

DATA KOMPONEN

Kode Komponen:

Nama Komponen:

Stok:

Vendor:

Kebutuhan:

PT Krama Yudha Ratu Motor

Tambah Hapus Edit

Masukkan data Cari

ID Komponen	Nama Komponen	Stok	Kebutuhan	Vendor
11001	HEADLAMP LH	76	13	INDONESIA KOITO
11002	HEADLAMP RH	89	1	INDONESIA KOITO
11003	LAMP COMB.FR LH	44	3	STANLEY ELECTR...

Form Data Master

DATA VENDOR

Kode:

Nama Vendor:

No. Telp:

Email:

Alamat:

PT Krama Yudha Ratu Motor

Tambah Hapus Edit

Masukkan data Cari

Kode	Nama Vendor	Email	No Telp	Kode Paki	Alamat
12001	INDONESIA KOI...	koto@gmail.com	213384	campaka pulih	
12002	STANLEY ELEC...	istanley@gmail...	82202213	Bekasi	

Form Data Master

DATA USER

ID:

Nama:

Pass:

Jabatan: ☐ Admin ☐ Staff

PT Krama Yudha Ratu Motor

Tambah Hapus Edit

Masukkan data Cari

ID	nama	pass	jabatan
123	khairunnisa	dodo	Admin
42	sth	ara	Staff
pc001	nisa	Admin	Admin
pc002	ivana	haha	Staff

4. Menu Transaksi

Form Transaksi

 **PT Krama Yudha Ratu Motor**

MENGECEK STOK

Target Produksi:

Cek 

Kode Part	Nama Komponen	Nama Vendor	Stok
11001	HEADLAMP LH	INDONESIA KOITO	76
11002	HEADLAMP RH	INDONESIA KOITO	89
11003	LAMP COMB FR LH	STANLEY ELECTRIC	44

Form Transaksi

 **PT Krama Yudha Ratu Motor**

ADVANCE PART

Purchasing Doc: OLD:

Kode Komponen: pilih Qty Pesan:

Nama Komponen: Advance:

Vendor: Qty Advance:

Next Simpan File 

Purchasing Doc	Kode Ka	Nama	Vendor	Jadwal L	Jumlah	Jadwal B	Jumlah
----------------	---------	------	--------	----------	--------	----------	--------

Form Transaksi

KOMPONEN MASUK

ID Komponen Masuk:

Delivery Note:

Kode Komponen: pilih

Nama Komponen:

Vendor:

Jumlah:

Tanggal:

 **PT Krama Yudha Ratu Motor**

Tambah Edit Hapus 

Memukkan data

ID Komponen	Delivery Note	Kode Kampo	Nama Kampo	Vendor	Jumlah	Tanggal
134	3113	11003	LAMP COMB	STANLEY EL	30	2018-09-07
24242	442	11003	LAMP COMB	STANLEY EL	1	2018-09-07
242524	11221	11001	HEADLAMP LH	INDONESIA	55	2018-09-05

Form Transaksi

PERMOHONAN SURAT JALAN

ID Perumahan:
 Kode Part:
 Kegunaan:
 Tanggal:
 Jumlah:

Tambah Hapus Edit
 Masukkan data: Cari

ID Perumahan	Kode Part	Kegunaan	Tanggal	Jumlah
800	11003	mugg	2018-07-05	90
3424	11002	MUM	2018-09-07	2
14001	11001	suplemen SL	2018-07-18	10

5. Menu Surat Jalan

Form Surat Jalan

SURAT JALAN

ID Surat Jalan:
 Jumlah:
 Tanggal:
 Tujuan:
 Nama Suplemen:

Tambah Hapus Edit
 Masukkan data: Cari

ID Komponen	Nama Komponen	Stok	Kebutuhan	Vendor
11001	HEADLAMP LH	76	13	INDONESIA KITO
11002	HEADLAMP RH	89	1	INDONESIA KITO
11003	LAMP COMB FR LH	44	3	STANLEY ELECT.

6. Menu Laporan

Form Laporan

LAPORAN PENERIMAAN KOMPONEN



Pilih Tanggal:

Form Laporan

LAPORAN PENERIMAAN KOMPONEN

 **PT Krama Yudha Ratu Motor**

Pilih Tanggal   

LAMPIRAN B

KODE PROGRAM

1. Login

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;

public class koneksi {
    public static Connection setKoneksi() {
        String konString = "jdbc:mysql://NISA:3306/ppk";
        Connection koneksi = null;
        try {
            Class.forName("com.mysql.jdbc.Driver");
            koneksi = (com.mysql.jdbc.Connection)
DriverManager.getConnection(konString,"root","");
            System.out.println("Koneksi Berhasil");
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(koneksi.class.getName()).log(Level.SEVERE,
null, ex);
            System.out.println("Koneksi Gagal");
        } catch (SQLException ex) {
            Logger.getLogger(koneksi.class.getName()).log(Level.SEVERE,
null, ex);
            System.out.println("Koneksi Gagal");
        }
        return koneksi;
    }

    public static Connection koneksiDB() throws SQLException {
        if(koneksi==null){
            try {
                String DB="jdbc:mysql://NISA:3306/ppk"; // database
                String user="root"; // user database
                String pass=""; // password database
                DriverManager.registerDriver(new com.mysql.jdbc.Driver());
                koneksi = (Connection)
DriverManager.getConnection(DB,user,pass);
            } catch (Exception e) {
                JOptionPane.showMessageDialog(null,"gagal koneksi");
            }
        }
        return koneksi;
    }
}
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;

```

```

import javax.swing.JOptionPane;
public class login extends javax.swing.JFrame {

    Connection con;
    Statement stat;
    ResultSet rs;
    String sql;
    String hakakses;
    String File = "login.jpg";

    public login() {
        initComponents();
        //pemanggilan fungsi koneksi database yang sudah kita buat pada
class koneksi.java
        koneksi DB = new koneksi();
        DB.config();
        con = DB.con;
        stat = DB.stm;

    }

    private void cekuser(){
        koneksi DB = new koneksi();
        DB.config();
        con = DB.con;
        stat = DB.stm;
        try{sql = "SELECT jabatan FROM user WHERE
ID='"+jTextField1.getText()+"' AND pass='"+jTextField2.getText()+"'";
        rs = stat.executeQuery(sql);
        while (rs.next()){
            hakakses = rs.getString(1);}

            if(hakakses.equals("")){
                JOptionPane.showMessageDialog(null, "ID atau password
salah");
            }
            else {
                setmenulogin();
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, e.getMessage());
        }
        // TODO add your handling code here:
    }
    private void setmenulogin(){
        if (hakakses.equals("Admin")){
            Menu a = new Menu(); // memanggil form lain
            a.setVisible(true);
            a.setLocationRelativeTo(null);//lokasi form
            dispose();//menghilangkan form

        }
        else if(hakakses.equals("Staff")){
            Menu a = new Menu(); // memanggil form lain
            a.setVisible(true);
            a.setLocationRelativeTo(null);//lokasi form
            a.jMenu1.setEnabled(false);

```

```

        a.jMenu4.setEnabled(false);
        dispose();//menghilangkan form
    }
}

```

2. Transaksi

```

import java.sql.SQLException;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import java.sql.ResultSet;
import java.util.logging.Level;
import java.util.logging.Logger;

public class Komponen extends javax.swing.JFrame {
    public static int statusSearching = 0;

    public Komponen() {
        initComponents();
        selectData();
        tampil_combo();
    }

    public void tampil_combo()
    {
        try {

            String SQL = "select nama from supplier order by kode asc";
            ResultSet rs = koneksi.executeQuery(SQL);
            while(rs.next()){
                Object[] ob = new Object[3]; //sesuai tabel di form
                ob[0] = rs.getString(1);

                cbV.addItem(ob[0]); // fungsi ini bertugas menampung isi dari
database
            }
            rs.close();

        } catch (Exception e) {
            System.out.println(e.getMessage());
        }

        private void baru()
        {
            txtID.setText("");
            txtNama.setText("");
            txtStok.setText("");
            txtGuna.setText("");
            txtCari.setText("");
            cbV.setSelectedIndex(0);
        }

        public void selectData()
        {
            String kolom[] = {"ID Komponen", "Nama
Komponen", "Stok", "Kebutuhan", "Vendor"};

```

```

DefaultTableModel dtm = new DefaultTableModel(null, kolom);
String SQL = "SELECT * FROM part";
ResultSet rs = koneksi.executeQuery(SQL);
try {
    while(rs.next()) { // sesuai db
        String kodepart = rs.getString(1);
        String stok = rs.getString(2);
        String nama= rs.getString(3);
        String Vendor = rs.getString(4);
        String guna = rs.getString(5);
        String data[] = {kodepart, nama, stok,guna, Vendor};
//sesuai tabel form
        dtm.addRow(data);
    }
} catch (SQLException ex) {
    Logger.getLogger(Pengguna.class.getName()).log(Level.SEVERE,
null, ex);
}
tblData4.setModel(dtm);
}
private void tblData4MouseClicked(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:
int baris = tblData4.getSelectedRow();
if (baris != -1) {
    txtID.setText(tblData4.getValueAt(baris, 0).toString());
    txtNama.setText(tblData4.getValueAt(baris, 1).toString());
    txtStok.setText(tblData4.getValueAt(baris, 2).toString());
    txtGuna.setText(tblData4.getValueAt(baris, 3).toString());
    cbV.setSelectedItem(tblData4.getValueAt(baris, 4).toString());
}
}
private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
if ("".equals(txtID.getText()) ||"".equals(txtNama.getText())) {
    JOptionPane.showMessageDialog(this, "Harap Lengkapi Data",
"Error", JOptionPane.WARNING_MESSAGE);
} else {

        String SQL = "INSERT INTO part (kodepart, stok, nama, Vendor,
pake) "
        + "VALUES('"+txtID.getText()+"', '"+txtStok.getText()+"', '"
+txtNama.getText()+"', '"+cbV.getSelectedItem()+"', '"+txtGuna.getText()+"')";

        int status = koneksi.execute(SQL);

        if (status == 1) {
            JOptionPane.showMessageDialog(this, "Data berhasil
ditambahkan", "Sukses", JOptionPane.INFORMATION_MESSAGE);
            selectData();
            baru();
        } else {
            JOptionPane.showMessageDialog(this, "Data gagal
ditambahkan", "Sukses", JOptionPane.WARNING_MESSAGE);
        }
}
}

```

```

    }

    }

    private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        int baris = tblData4.getSelectedRow();
        if (baris != -1) {
            String no = tblData4.getValueAt(baris, 0).toString();
            String SQL = "DELETE FROM part WHERE kodepart='"+no+"'";
            int status = koneksi.execute(SQL);
            if (status==1) {
                JOptionPane.showMessageDialog(this, "Data berhasil dihapus",
                "Sukses", JOptionPane.INFORMATION_MESSAGE);
                selectData();
                baru();
            } else {
                JOptionPane.showMessageDialog(this, "Data gagal dihapus",
                "Gagal", JOptionPane.WARNING_MESSAGE);
            }
        } else {
            JOptionPane.showMessageDialog(this, "Pilih Baris Data Terlebih
            dahulu", "Error", JOptionPane.WARNING_MESSAGE);
        }
    }

    private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        if ("".equals(txtID.getText()) || "".equals(txtNama.getText()) ||
            "".equals(txtStok.getText())){
            JOptionPane.showMessageDialog(this, "Harap Lengkapi Data",
            "Error", JOptionPane.WARNING_MESSAGE);
        } else {

            String SQL = "UPDATE part SET stok='"+txtStok.getText()+"', "
            + "nama='"+txtNama.getText()+"', "
            + "Vendor='"+cbV.getSelectedItem()+"', "
            + "pake='"+txtGuna.getText()+"' WHERE
            kodepart='"+txtID.getText()+"'";
            int status = koneksi.execute(SQL);
            if (status == 1) {
                JOptionPane.showMessageDialog(this, "Data berhasil
            diupdate", "Sukses", JOptionPane.INFORMATION_MESSAGE);
                selectData();
                baru();
            } else {
                JOptionPane.showMessageDialog(this, "Data gagal diupdate",
                "Gagal", JOptionPane.WARNING_MESSAGE);
                baru();
            }
        }
    }

    private void cariActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        statusSearching=1;
        if(txtCari.getText().isEmpty())

```

```

        { statusSearching = 0;}
        else if(statusSearching==1){
            DefaultTableModel Tabelbrg = new DefaultTableModel();
            Tabelbrg.addColumn("ID Komponen");
            Tabelbrg.addColumn("Nama Komponen");
            Tabelbrg.addColumn("Stok");
            Tabelbrg.addColumn("Kebutuhan");
            Tabelbrg.addColumn("Vendor");
            String cari = txtCari.getText();
            try{
                String SQL = "Select * From part Where kodepart LIKE
'" +cari+"'"
                + " OR nama LIKE '" +cari+"'" OR stok LIKE '" +cari+"'" OR pake
LIKE '" +cari+"'" OR Vendor LIKE '" +cari+"'" ORDER BY kodepart";

                ResultSet rs = koneksi.executeQuery(SQL);
                while(rs.next()){
                    Tabelbrg.addRow(new Object[]{
                        rs.getString(1),
                        rs.getString(3),
                        rs.getString(2),
                        rs.getString(5),
                        rs.getString(4),
                    });
                    //nomortabel++;
                }

                tblData4.setModel(Tabelbrg);
                baru();
            }catch(SQLException ex){
                JOptionPane.showMessageDialog(rootPane, "Data yang dicari
tidak ada !!!!");}}}

        private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
            selectData();
            baru();// }

```