

**RANCANG BANGUN SISTEM INFORMASI PENJADWALAN
PRODUKSI MENGGUNAKAN FRAMEWORK
CODEIGNITER 3.1.8 DENGAN PHP 5.6.30 DAN MARIADB
10.1.21 PADA PT BANEEX INDONESIA**

TUGAS AKHIR

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

OLEH
FADHLA FANINI
1314024



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
PENJADWALAN PRODUKSI MENGGUNAKAN
FRAMEWORK CODEIGNITER 3.1.8 DENGAN PHP
5.6.30 DAN MARIADB 10.1.21 PADA PT BANEEX
INDONESIA

Disusun Oleh :
Nama : Fadhla Fanini
Nim : 1314024
Program Studi : Sistem Informasi Industri Otomotif

Telah Diuji Oleh Tim Penguji Sidang Tugas Akhir Program Studi Sistem
Informasi Industri Otomotif Politeknik STMI Jakarta Kementerian Perindustrian
Republik Indonesia pada hari Jumat tanggal 14 September 2018.

Jakarta, 19 September 2018

Dosen Pembimbing



Drs. Jacob Saragih, MM.
NIP: 195404281986031002

Ketua Penguji



Noveriza Yuliasari, ST, MT.
NIP: 197811212009012003

Dosen Penguji



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

Dosen Penguji



Triana Fatmawati, ST, MT.
NIP: 198005142005022001

POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

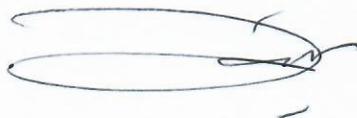
LEMBAR PERSETUJUAN DOSEN PEMBIMBING

Judul Tugas Akhir : RANCANG BANGUN SISTEM INFORMASI
PENJADWALAN PRODUKSI MENGGUNAKAN
FRAMEWORK CODEIGNITER 3.1.8 DENGAN PHP
5.6.30 DAN MARIADB 10.1.21 PADA PT BANEEX
INDONESIA

Disusun Oleh :
Nama : Fadhla Fanini
Nim : 1314024
Program Studi : Sistem Informasi Industri Otomotif
Tanggal Seminar : 23 Agustus 2018
Tanggal Sidang : 14 September 2018
Tanggal Lulus : 14 September 2018

Jakarta, 19 September 2018

Menyetujui
Dosen Pembimbing



Drs. Jacob Saragih, MM.
NIP: 195404281986031002

POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

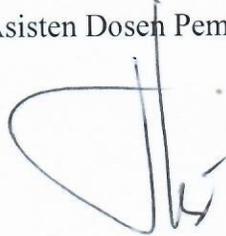
LEMBAR PERSETUJUAN ASISTEN DOSEN PEMBIMBING

Judul Tugas Akhir : RANCANG BANGUN SISTEM INFORMASI
PENJADWALAN PRODUKSI MENGGUNAKAN
FRAMEWORK CODEIGNITER 3.1.8 DENGAN PHP
5.6.30 DAN MARIADB 10.1.21 PADA PT BANEEX
INDONESIA

Disusun Oleh :
Nama : Fadhla Fanini
Nim : 1314024
Program Studi : Sistem Informasi Industri Otomotif
Tanggal Seminar : 23 Agustus 2018
Tanggal Sidang : 14 September 2018
Tanggal Lulus : 14 September 2018

Jakarta, 19 September 2018

Menyetujui
Asisten Dosen Pembimbing



Ulil Hamida, ST.MT
NIP: 198103272005022001

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Fadhla Fanini
NIM : 1314024
Judul Tugas Akhir : Rancang Bangun Sistem Informasi Penjadwalan Produksi Menggunakan Framework Codeigniter 3.1.8 Dengan PHP 5.6.30 Dan MariaDB 10.1.21 Pada PT Baneex Indonesia
Pembimbing : Drs. Jacob Saragih, MM.

Tanggal	Keterangan	Paraf
30-07-2018	Bimbingan BAB I	
31-07-2018	Revisi BAB I dan Bimbingan BAB II	
03-08-2018	Revisi BAB II	
06-08-2018	Bimbingan BAB III	
07-08-2018	Revisi BAB III	
09-08-2018	Bimbingan BAB IV	
10-08-2018	Revisi BAB IV	
13-08-2018	Bimbingan BAB V	
15-08-2018	Revisi BAB V	
16-08-2018	Revisi BAB V	
17-08-2018	Demo Program	

Mengetahui,
Ketua Program Studi
Sistem Informasi Industri Otomotif

Dosen Pembimbing

(Drs. Jacob Saragih, MM)
NIP. 195404281986031002

(Drs. Jacob Saragih, MM)
NIP. 195404281986031002

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Fadhla Fanini
NIM : 1314024
Judul Tugas Akhir : Rancang Bangun Sistem Informasi Penjadwalan Produksi Menggunakan Framework Codeigniter 3.1.8 Dengan PHP 5.6.30 Dan MariaDB 10.1.21 Pada PT Baneex Indonesia
Asisten Pembimbing : Ulil Hamida, ST.MT

Tanggal	Keterangan	Paraf
30-07-2018	Bimbingan BAB I	
31-07-2018	Revisi BAB I dan Bimbingan BAB II	
03-08-2018	Revisi BAB II	
06-08-2018	Bimbingan BAB III	
07-08-2018	Revisi BAB III	
09-08-2018	Bimbingan BAB IV	
10-08-2018	Revisi BAB IV	
13-08-2018	Bimbingan BAB V	
15-08-2018	Revisi BAB V	
16-08-2018	Revisi BAB V	
17-08-2018	Demo Program	

Mengetahui,
Ketua Program Studi
Sistem Informasi Industri Otomotif

Asisten Dosen Pembimbing

(Drs. Jacob Saragih, MM)
NIP. 195404281986031002

(Ulil Hamida, ST.MT)
NIP. 198103272005022001

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : Fadhla Fanini

NIM :1314024

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

“RANCANG BANGUN SISTEM INFORMASI PENJADWALAN PRODUKSI MENGGUNAKAN FRAMEWORK CODEIGNITER 3.1.8 DENGAN PHP 5.6.30 DAN MARIADB 10.1.21 PADA PT BANEEX INDONESIA”

- **Dibuat** dan diselesaikan sendiri, dengan menggunakan literatur hasil kuliah, *survey* lapangan, dibantu oleh dosen pembimbing, dan menggunakan buku-buku maupun jurnal-jurnal ilmiah yang menjadi bahan acuan yang tertera dalam daftar pustaka Tugas Akhir ini.
- **Bukan** merupakan hasil duplikasi karya tulis yang sudah dipublikasikan atau yang pernah dipakai sebelumnya untuk mendapatkan gelar sarjana di Universitas/Perguruan Tinggi lain, kecuali yang telah disebutkan sumbernya dan dicantumkan pada referensi karya Tugas Akhir ini.
- **Bukan** merupakan karya tulis hasil terjemahan dari kumpulan buku atau jurnal acuan yang tertera dalam referensi karya Tugas Akhir ini.

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

Jakarta, 20 Agustus 2018

Fadhla Fanini

ABSTRAK

PT Baneex Indonesia merupakan sebuah perusahaan yang bergerak di bidang otomotif yang memproduksi memproduksi *spring*, *wire* dan *bending*. PPC (*Production Planing and Control*) pada PT Baneex Indonesia merupakan salah satu divisi yang bertanggung jawab terhadap semua proses produksi baik dari mengelola pesanan sampai penjadwalan produksi. Dalam prosesnya, PT Baneex Indonesia melakukan perencanaan produksi secara manual dengan memperkirakan rencana produksi dari *Schedule Delivery* atau jadwal pengiriman yang sudah dijadwalkan oleh pelanggan. Dengan sistem seperti ini, jumlah barang jadi yang diproduksi sering terlambat dari jadwal pengiriman seharusnya. Maka dari itu, perlu ditambahkan sebuah metode untuk membantu PT Baneex Indonesia khususnya pada divisi PPC untuk melakukan penjadwalan produksi menggunakan metode *Master Production Schedule*. Pengembangan sistem yang dilakukan adalah menggunakan metode *waterfall*. Pemodelan sistem menggunakan *Unified Modelling Language* (UML). Pemodelan data menggunakan *Entity Relationship Diagram* (ERD), dan kamus data. Perancangan sistem menggunakan *Windows Navigation Diagram* (WND) dan desain antar muka. Pengujian sistem menggunakan *Black Box Testing*. Pembuatan aplikasi menggunakan PHP 5.6.30 dengan *framework* Codeigniter 3.1.8 dan MariaDB 10.1.21 sebagai *database* dan menerapkan metode *Master Production Schedule* (MPS). Aplikasi sistem informasi penjadwalan produksi yang dibuat berdasarkan proses bisnis yang berjalan dan permasalahan yang ditemukan di PT Baneex Indonesia. Diharapkan dapat digunakan untuk melakukan penjadwalan produksi pada divisi *Production Planning and Control*.

Kata Kunci : *Sistem Informasi, Penjadwalan Produksi, Master Production Schedule, Codeigniter, Schedule Delivery.*

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya, penulis dapat menyelesaikan Tugas Akhir dengan judul **“Rancang Bangun Sistem Informasi Penjadwalan Produksi Menggunakan Framework Codeigniter 3.1.8 Dengan PHP 5.6.30 Dan MariaDB 10.1.21 Pada Pt Baneex Indonesia”**.

Tugas akhir ini disusun guna memenuhi sebagian syarat yang harus dipenuhi dalam menempuh program sarjana terapan jurusan Sistem Informasi Industri Otomotif pada Politeknik STMI Jakarta.

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

1. Allah SWT atas rahmat serta kemudahan yang diberikan.
2. Kedua orang tua serta keluarga tercinta yang telah memberikan do'a, dukungan, pengorbanan, semangat dan kasih sayang hingga saat ini.
3. Bapak Dr. Mustofa, S.T., M.T., selaku Direktur Politeknik STMI Jakarta.
4. Bapak Drs. Jacob Saragih, MM., selaku Ketua Program Studi Sistem Informasi Industri Otomotif sekaligus dosen pembimbing yang telah memberikan arahan dan masukan dalam penyusunan Tugas Akhir.
5. Ibu Ulil Hamida, ST, MT., selaku asisten dosen pembimbing yang telah memberikan arahan dan masukan dalam penyusunan Tugas Akhir.
6. Bapak Ir. H Tri Yono Sudarno,MM., selaku Direktur Utama PT Baneex Indonesia.
7. Bapak Genki Ananda selaku pembimbing yang telah membantu mengarahkan dan membimbing selama Kerja Lapangan di PT Baneex Indonesia.
8. Ibu Risa, Ibu Etha dan seluruh pegawai di PT Baneex Indonesia yang telah membantu memberi arahan serta informasi untuk pengumpulan data.
9. Seluruh dosen Politeknik STMI Jakarta yang telah memberikan ilmunya guna menambah pengetahuan dan pengalaman yang sangat bermanfaat.

10. Teman-teman mahasiswa/i Politeknik STMI Jakarta Program Studi Sistem Informasi angkatan 2014 yang telah membantu penulis.
11. Teman-teman HEM Community Indri Fitriana Alfia Nisa, Rusdiansyah, Gumelar Pinanggih Mulyo, Roni Sudrajat, Ainalisa Alya Nabella, Dimas Pangestu, Fakhri Tri, Dimas Abrianno, Sofyan Sauri dan Akhmad Febrianto yang telah membantu penulis baik secara langsung maupun tidak langsung dan mewarnai hari-hari penulis..

Semoga Allah SWT membalas kebaikan dan ketulusan semua pihak yang telah membantu menyelesaikan Tugas Akhir ini dengan melimpahkan rahmat dan karunia-Nya. Besar harapan Penulis bahwa Tugas Akhir ini dapat memberikan manfaat serta pengetahuan bagi pembacanya. Terima kasih.

Jakarta, 20 Agustus 2018

Penulis,
Fadhla Fanini

DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN	ii
LEMBAR PERSETUJUAN DOSEN PEMBIMBING.....	iii
LEMBAR PERSETUJUAN ASISTEN DOSEN PEMBIMBING	iv
LEMBAR BIMBINGAN DOSEN PEMBIMBING	v
LEMBAR BIMBINGAN ASISTEN DOSEN PEMBIMBING.....	vi
LEMBAR PERNYATAAN KEASLIAN	vii
ABSTRAK	viii
KATA PENGANTAR	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL.....	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Pokok Permasalahan.....	3
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 Sistem.....	6
2.1.1 Karakteristik Sistem	7
2.1.2 Klasifikasi Sistem.....	8
2.2 Konsep Dasar Informasi.....	9
2.2.1 Kualitas Informasi	10
2.2.2 Siklus Informasi	11
2.3 Konsep Dasar Sistem Informasi.....	13

2.3.1	Komponen Sistem Informasi.....	13
2.4	Pengertian <i>Production Planning & Inventory Control</i>	14
2.5	Konsep Penjadwalan Produksi.....	14
2.5.1	Pengertian Penjadwalan Produksi	14
2.5.2	Tujuan Penjadwalan Produksi	15
2.5.3	Manfaat Penjadwalan Produksi	15
2.6	Konsep Dasar <i>Master Production Schedule</i> (MPS).....	16
2.6.1	Pengertian <i>Master Production Schedule</i> (MPS)	16
2.6.2	Perhitungan <i>Master Production Schedule</i> (MPS)	17
2.7	Telaah Jurnal	19
2.8	<i>System Development Life Cycle</i> (SDLC)	20
2.9	<i>Waterfall</i>	24
2.10	<i>Flowmap</i>	25
2.11	<i>Unified Modelling Language</i> (UML).....	26
2.11.1	<i>Use Case Diagram</i>	28
2.11.2	<i>Use Case Description</i>	30
2.11.3	<i>Activity Diagram</i>	31
2.11.4	<i>Sequence Diagram</i>	33
2.11.5	<i>Class Diagram</i>	35
2.12	<i>Entity Relationship Diagram</i> (ERD).....	36
2.13	Sistem Manajemen Basis Data.....	37
2.13.1	<i>Database</i>	37
2.13.2	<i>Conceptual Data Modelling</i> (CDM)	38
2.13.3	Kamus Data	41
2.13.4	MariaDB	42
2.13.5	Tipe Data	43
2.14	<i>Windows Navigation Diagram</i> (WND).....	45
2.15	PHP (<i>Hypertext Preprocessor</i>)	45
2.16	Framework	46
2.17	<i>Codeigniter</i>	47
2.18	<i>Black Box Testing</i>	50

BAB III METODOLOGI PENELITIAN	51
3.1 Metodologi Penelitian	51
3.2 Jenis dan Sumber Data	51
3.3 Metode Pengumpulan Data	52
3.4 Metode Pengembangan Sistem	53
3.5 Kerangka Penelitian	54
BAB IV PENGUMPULAN DAN PENGOLAHAN DATA.....	58
4.1 Tentang Perusahaan	58
4.2 Profil Perusahaan	59
4.3 Visi dan Misi Perusahaan.....	60
4.4 Struktur Organisasi Perusahaan	60
4.5 <i>International Organization For Standardization</i>	70
4.6 Perhitungan <i>Safety Stock</i> dan <i>Lot Size</i>	71
4.7 Analisis Dokumen Penjadwalan Produksi	72
4.7.1 Dokumen Masuk	72
4.8 Analisis Proses (<i>Flowmap</i> Sistem yang Sedang Berjalan)	74
4.8.1 <i>Flowmap</i> Proses Penjadwalan Produksi.....	75
4.9 <i>Use Case Diagram</i> Sistem Berjalan.....	76
4.10 Analisis Permasalahan	79
BAB V ANALISIS DAN PEMBAHASAN.....	81
5.1 Analisis Kebutuhan Sistem	81
5.1.1 <i>Functional Requirement</i>	82
5.1.2 <i>Non Functional Requirement</i>	83
5.2 <i>Flowmap</i> Sistem Penjadwalan Produksi Usulan.....	83
5.3 <i>Use Case Diagram</i> (Usulan)	85
5.3.1 <i>Use Case Description</i>	86
5.4 <i>Activity Diagram</i>	89
5.5 <i>Sequence Diagram</i>	95
5.6 <i>Class Diagram</i>	102
5.7 <i>Deployment Diagram</i>	103
5.8 Pemodelan Data Sistem Usulan	104

4.9.1	<i>Entity Relationship Diagram (ERD)</i>	104
4.9.2	<i>Conceptual Data Model</i>	105
4.9.3	Kamus Data.....	106
5.9	<i>Windows Navigation Diagram (WND)</i>	110
5.10	Desain Antar Muka	111
5.11	<i>Black Box Testing</i>	115
BAB IV KESIMPULAN DAN SARAN		123
6.1	Kesimpulan	123
6.2	Saran.....	123
DAFTAR PUSTAKA		124

DAFTAR GAMBAR

	Halaman
Gambar II.1 Pilar Kualitas Informasi.....	11
Gambar II.2 Siklus Informasi dalam Angraeni dan Irviana.....	12
Gambar II.3 Rumus <i>Forecast</i> MPS.....	17
Gambar II.4 Rumus ATP MPS 1	18
Gambar II.5 Rumus ATP MPS 2	18
Gambar II.6 Contoh Tabel MPS	18
Gambar II.7 Metode <i>Waterfall</i>	25
Gambar II.8 Contoh Windows Navigation Diagram	45
Gambar II.9 <i>Codeigniter</i>	49
Gambar III.1 Kerangka Penelitian	56
Gambar IV.1 Logo PT Baneex Indonesia	59
Gambar IV.2 Struktur Organisasi PT Baneex Indonesia	61
Gambar IV.3 Sertifikat ISO 9001:2015	71
Gambar IV.4 <i>Purchase Order</i>	73
Gambar IV.5 <i>Schedule Delivery</i>	74
Gambar IV.6 <i>Flowmap</i> Proses Penjadwalan Produksi	75
Gambar IV.7 <i>Use Case Diagram</i> Proses Penjadwalan Produksi.....	77
Gambar V.1 <i>Flowmap</i> usulan	84
Gambar V.2 <i>Use case</i> usulan	85
Gambar V.3 <i>Activity Diagram Login</i>	89
Gambar V.4 <i>Activity Diagram</i> Mengelola Data <i>Customer</i>	90
Gambar V.5 <i>Activity Diagram</i> Mengelola Data <i>User</i>	91
Gambar V.6 <i>Activity Diagram</i> Mengelola Data Produk	92
Gambar V.7 <i>Activity Diagram</i> Mengelola Data <i>Purchase Order</i>	93
Gambar V.8 <i>Activity Diagram</i> Mengelola Data <i>Schedule Delivery</i>	94
Gambar V.9 <i>Activity Diagram</i> Pembuatan MPS	95

Gambar V.10 <i>Sequence Diagram Login</i>	96
Gambar V.11 <i>Sequence Diagram Mengelola Data Customer</i>	97
Gambar V.12 <i>Sequence Diagram Mengelola Data User</i>	98
Gambar V.13 <i>Sequence Diagram Mengelola Data Produk</i>	99
Gambar V.14 <i>Sequence Diagram Mengelola Data Purchase Order</i>	100
Gambar V.15 <i>Sequence Diagram Mengelola Data Schedule Delivery</i>	101
Gambar V.16 <i>Sequence Diagram Pembuatan MPS</i>	102
Gambar V.17 <i>Class Diagram Sistem Informasi Penjadwalan Produksi</i>	103
Gambar V.18 <i>Deployment Diagram Sistem Informasi Penjadwalan Produksi</i> ..	104
Gambar V.19 ERD Sistem Informasi Penjadwalan Produksi	105
Gambar V.20 CDM Sistem Informasi Penjadwalan Produksi	106
Gambar V.21 WND Sistem Informasi Penjadwalan Produksi	110
Gambar V.22 Desain Antar Muka Login	111
Gambar V.23 Desain Antar Muka Data Customer	111
Gambar V.24 Desain Antar Muka Produk	112
Gambar V.25 Desain Antar Muka <i>Purchase Order</i>	112
Gambar V.26 Desain Antar Muka <i>Detail Purchase Order</i>	113
Gambar V.27 Desain Antar Muka <i>Schedule Delivery</i>	113
Gambar V.28 Desain Antar Muka <i>Detail Schedule Delivery</i>	114
Gambar V.29 Desain Antar Muka <i>User</i>	114
Gambar V.30 Desain Antar Muka Data MPS	115

DAFTAR TABEL

	Halaman
Tabel II.1 Simbol-simbol <i>Flowmap</i>	26
Tabel II.2 Jenis <i>Diagram</i> Pada UML dan Fungsi-fungsinya	27
Tabel II.3 Simbol-simbol <i>Use Case Diagram</i>	29
Tabel II.4 Simbol-simbol <i>Activity Diagram</i>	32
Tabel II.5 Simbol-simbol <i>Sequence Diagram</i>	34
Tabel II.6 Simbol-simbol <i>Class Diagram</i>	35
Tabel II.7 Simbol-simbol ERD	36
Tabel II.8 Contoh Tabel Barang.....	38
Tabel II.9 Simbol-Simbol <i>Conceptual Data Model (CDM)</i>	38
Tabel II.10 Simbol-Simbol Aturan <i>Conceptual Data Model (CDM)</i>	39
Tabel IV.1 Definisi Aktor	77
Tabel IV.2 Definisi <i>Use Case</i>	78
Tabel IV.3 Analisis Permasalahan	79
Tabel V.1 Analisis Kebutuhan Sistem	81
Tabel V.2 <i>Use Case Description Login</i>	86
Tabel V.3 <i>Use Case Description Mengelola Data Customer</i>	86
Tabel V.4 <i>Use Case Description Mengelola Data Produk</i>	87
Tabel V.5 <i>Use Case Description Mengelola Data Purchase Order</i>	87
Tabel V.6 <i>Use Case Description Input Schedule Delivery</i>	88
Tabel V.7 <i>Use Case Description Pembautan MPS</i>	89
Tabel V.8 <i>Use Case Description Mengelola Data User</i>	88
Tabel V.9 Tabel Kamus Data <i>User</i>	107
Tabel V.10 Tabel Kamus Data <i>data_customer</i>	107
Tabel V.11 Tabel Kamus Data <i>data_produk</i>	108
Tabel V.12 Tabel Kamus Data <i>purchase_order</i>	108
Tabel V.13 Tabel Kamus Data <i>purchase_order_detil</i>	109
Tabel V.14 Tabel Kamus Data <i>purchase_order_detil_delivery</i>	109

Tabel V.15 Table Kamus Data mps	110
Tabel V.16 <i>Black Box Testing Use Case Login</i>	115
Tabel V.17 <i>Black Box Testing Use Case Mengelola Data Customer</i>	116
Tabel V.18 <i>Black Box Testing Use Case Mengelola Data Produk</i>	117
Tabel V.19 <i>Black Box Testing Use Case Mengelola Data User</i>	118
Tabel V.20 <i>Black Box Testing Use Case Data Purchase Order</i>	119
Tabel V.21 <i>Black Box Testing Use Case Mengelola Data Schedule Delivery</i> ..	121
Tabel V.22 <i>Black Box Testing Use Case Pembuatan MPS</i>	122

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dalam dunia industri, selain kualitas produk yang dihasilkan penilaian suatu perusahaan juga tergantung dari kemampuan perusahaan untuk memenuhi pesanan yang dijanjikan baik itu dari segi waktu maupun dari segi jumlah. Saat ini persaingan dalam dunia industri semakin ketat. Persaingan tersebut sering terjadi dalam hal kualitas produk maupun pelayanan yang diberikan perusahaan kepada pelanggannya, dimana kepuasan pelanggan menjadi salah satu parameter dalam mengukur kesuksesan suatu perusahaan. Semakin cepat perusahaan memenuhi kebutuhan konsumennya, semakin puas pula konsumen karena dapat memenuhi kebutuhannya dengan cepat. Maka dari itu untuk memenuhi pesanan dibutuhkan sebuah proses produksi yang baik dan penjadwalan produksi yang baik.

Penjadwalan merupakan penentuan waktu dan kuantitas atau sumber daya produktif, meliputi kapasitas, peralatan dan fasilitas produksi, bahan baku, dan tenaga kerja yang dibutuhkan dalam menghasilkan suatu produk atau jasa, agar produksi dapat berlangsung dengan lancar, tepat jumlah, tepat waktu, dan tepat mutu. Penjadwalan produksi ini sangat penting bagi sebuah perusahaan dalam melakukan proses produksi. Hal ini dikarenakan pabrik harus mampu dalam memenuhi kebutuhan pelanggan secara tepat waktu. Fungsi dari penjadwalan produksi ini salah satunya adalah meminimalisasi keterlambatan yang terjadi pada proses produksi di perusahaan.

Dalam era globalisasi, persaingan dalam hal teknologi semakin maju dan berkembang dengan pesat. Oleh karena itu setiap perusahaan harus memiliki sistem yang dapat berjalan dengan perkembangan tersebut. Salah satu proses yang membutuhkan sistem yaitu sistem penjadwalan produksi pada suatu perusahaan. Dengan adanya sistem penjadwalan produksi yang dapat membantu perusahaan maka perusahaan dapat memenuhi kebutuhan pelanggan dengan lancar, tepat jumlah dan tepat waktu.

Dalam penjadwalan produksi terdapat metode salah satunya *Master Production Schedule*. *Master Production Schedule* merupakan rencana produksi jangka pendek perusahaan dalam menghasikan produk jadi atau produk akhir. *Master Production Schedule* ini juga bermanfaat dalam merencanakan kapasitas produksi. Hal ini yang membuat metode MPS banyak digunakan pada suatu perusahaan sebagai metode penjadwalan.

PT Baneex Indonesia merupakan sebuah perusahaan yang bergerak di bidang otomotif yang memproduksi *spring*, *wire* dan *bending*. PPC (*Production Planing and Control*) pada PT Baneex Indonesia merupakan salah satu divisi yang bertanggung jawab terhadap proses produksi baik dari mengelola pesanan sampai penjadwalan produksi.

Dalam prosesnya, PT Baneex Indonesia melakukan perencanaan produksi secara manual dengan memperkirakan rencana produksi dari *Schedule Delivery* atau jadwal pengiriman yang sudah dijadwalkan oleh pelanggan. Dengan sistem seperti ini, jumlah barang jadi yang diproduksi sering terlambat dari jadwal pengiriman seharusnya. Maka dari itu, perlu ditambahkan sebuah metode untuk membantu PT Baneex Indonesia khususnya pada divisi PPC untuk melakukan penjadwalan produksi menggunakan metode *Master Production Schedule*.

Permasalahan lain, data yang berhubungan dengan proses penjadwalan produksi masih belum terkomputerisasi, sehingga menimbulkan masalah dalam pencarian data, penginputan, dan pengolahan data. Data penjadwalan produksi yang disimpan secara bertumpuk, adakalanya tumpukan tersebut hilang atau rusak sehingga data tidak dapat diperoleh.

Berdasarkan latar belakang dan permasalahan tersebut, maka disusun tugas akhir yang membahas tentang penjadwalan produksi. Tugas akhir ini memiliki judul “RANCANG BANGUN SISTEM INFORMASI PENJADWALAN PRODUKSI MENGGUNAKAN FRAMEWORK CODEIGNITER 3.1.8 DENGAN PHP 5.6.30 DAN MARIADB 10.1.21 PADA PT BANEEX INDONESIA”. Tugas akhir ini berfungsi untuk membantu perusahaan dalam proses penjadwalan produksi sehingga proses tersebut dapat ditingkatkan.

1.2 Pokok Permasalahan

Berdasarkan latar belakang yang dipaparkan sebelumnya, maka dapat diidentifikasi permasalahan-permasalahan yang terjadi pada proses penjadwalan produksi di PT Baneex Indonesia adalah sebagai berikut:

1. Belum adanya metode yang digunakan dalam penjadwalan produksi pada PT Baneex Indonesia sehingga penjadwalan tidak dapat berjalan dengan baik. Penjadwalan dengan kondisi saat ini, menyebabkan hasil perhitungan menjadi kurang sesuai yang berakibat terlambatnya pengiriman barang jadi.
2. Belum adanya sistem penjadwalan secara terkomputerisasi yang menyebabkan penyimpanan data dan pencarian data menjadi kurang baik. Hal tersebut menyebabkan, data yang ditumpuk dapat rusak atau hilang.

1.3 Tujuan Penelitian

Tujuan penelitian dari tugas akhir ini yang dilakukan terhadap divisi *Production Planning and Control* PT Baneex Indonesia dalam melakukan penjadwalan produksi adalah untuk merancang dan membangun sistem penjadwalan produksi yang mampu:

1. Melakukan penjadwalan produksi dengan metode *Master Production Schedule* yang dapat digunakan untuk melakukan penjadwalan produksi pada divisi *Production Planning and Control* PT Baneex Indonesia.
2. Melakukan penyimpanan di satu lokasi sehingga mempercepat proses pencarian data jika dibutuhkan serta mengurangi resiko data rusak atau data hilang.

1.4 Batasan Masalah

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

1. Penelitian ini hanya membahas mengenai sistem penjadwalan produksi untuk produksi tanpa *Bill Of Material* untuk produk *Wire*.
2. Sistem dirancang belum diimplementasikan.

1.5 Manfaat Tugas Akhir

Adapun manfaat bagi perusahaan yang dapat diambil dari penelitian Tugas Akhir ini untuk Perusahaan yaitu memiliki aplikasi penjadwalan produksi yang dapat digunakan oleh semua *staff* PPC. Hal ini dapat membantu dalam melakukan penjadwalan produksi.

1.6 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 terurai 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 pada laporan ini adalah seputar sistem informasi, penjadwalan produksi, *Master Production Schedule*, *Unified Modelling Language* (UML), dan teori-teori lain yang berhubungan dengan perancangan sistem informasi.

BAB III METODOLOGI PENELITIAN

Pada bab ini menjelaskan metode ilmiah dalam mencari, mengembangkan, dan menguji kebenaran tentang suatu pengetahuan. Selain itu dijelaskan pula kerangka pemecahan masalah yang menguraikan tahap-tahap untuk mengatasi permasalahan yang telah diidentifikasi.

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Pada bab ini akan membahas tentang data yang telah diperoleh berdasarkan penelitian di PT Baneex Indonesia, meliputi profil perusahaan, struktur organisasi, proses bisnis sistem penjadwalan produksi yang berjalan, dan dokumen yang terlibat pada proses bisnis tersebut.

BAB V ANALISIS DAN PEMBAHASAN

Pada bab ini berisi analisis rinci dari pengolahan data, yakni mulai dari analisis sistem yang meliputi diagram alir sistem usulan, pemodelan sistem dengan UML, pemodelan data dengan *Entity Relationship Diagram* (ERD), *Conceptual Data Model* (CDM), dan Kamus Data, perancangan *Windows Navigation Diagram*, perancangan tampilan layar, dan pembuatan spesifikasi sistem yang diperlukan.

BAB VI KESIMPULAN DAN SARAN

Bab ini berisikan kesimpulan dari hasil penelitian yang dilakukan, serta mengemukakan saran-saran dalam penerapan sistem informasi penjadwalan produksi untuk perusahaan dalam penelitian dan pengembangan selanjutnya.

BAB II

LANDASAN TEORI

2.1 Pengertian Sistem

Berdasarkan Djahir dan Pratita (2014), sistem adalah kumpulan dari komponen-komponen yang saling berkaitan yang bekerja bersama untuk mencapai beberapa tujuan. Sistem apapun dapat mempunyai subsistem. Subsistem adalah sistem yang lebih kecil dari sebuah sistem. Maka, subsistem dapat menjadi satu cara untuk berpikir tentang komponen dari sebuah sistem. Pemahaman sistem sebagai gabungan antara subsistem sangat berguna untuk para analis karena memungkinkan para analis untuk memfokuskan diri pada suatu area tertentu. Supersistem adalah sistem yang lebih besar dan sangat kompleks. Supersistem mengacu kepada sistem apapun yang memiliki sistem-sistem yang lebih kecil.

Sementara menurut Rusdiana dan Irfan (2014), pendekatan sistem berusaha menjelaskan sesuatu berdasarkan sudut pandang tujuan, proses, dan struktur. Dengan memahami struktur, sistem dan proses sistem, seseorang dapat menjelaskan tujuan suatu sistem tidak tercapai. Pendekatan sistem merupakan jumlah keseluruhan dari bagian-bagian yang saling bekerja sama untuk mencapai hasil yang diharapkan berdasarkan kebutuhan tertentu.

Kata “sistem” berasal dari bahasa Yunani yaitu, *systema*, yang artinya himpunan bagian atau komponen yang saling berhubungan secara teratur dan merupakan suatu keseluruhan. Selain itu, bisa diartikan sekelompok elemen yang independen, namun saling berkaitan sebagai satu kesatuan. Berikut adalah definisi dari sistem menurut para ahli :

1. Menurut Sutabri (2012) suatu sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan lain yang berfungsi bersama-sama untuk mencapai tujuan tertentu.
2. Menurut Gordon B. Davis (dalam Rusdiana, 2014) sistem merupakan bagian-bagian yang beroperasi secara bersama-sama untuk mencapai beberapa tujuan.

3. Menurut Raymond Mcleod (dalam Yakub, 2012) sistem adalah sekelompok elemen yang terintegrasi untuk mencapai suatu tujuan.

Menurut Jogiyanto, 2010 Sistem adalah kumpulan dari elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Sistem ini menggambarkan suatu kejadian dan kesatuan yang nyata, seperti tempat, benda, serta orang-orang yang ada dan terjadi.

2.1.1 Karakteristik Sistem

Berikut ini adalah pandangan tentang karakteristik sistem yang dikemukakan oleh Sutanta (2003), dalam Rusdiana dan Irfan (2014) yaitu sebagai berikut :

- a. **Komponen (*components*)**
Komponen sistem adalah segala sesuatu yang menjadi bagian penyusunan sistem. Komponen sistem dapat berupa benda nyata ataupun abstrak. Komponen sistem disebut sebagai subsistem.
- b. **Batas (*boundary*)**
Batas sistem diperlukan untuk membedakan satu sistem dengan sistem yang lain. Tanpa adanya batas sistem, sangat sulit untuk memberikan batasan tinjauan terhadap sistem.
- c. **Lingkungan (*environments*)**
Lingkungan sistem adalah segala sesuatu yang berada diluar sistem yang dapat menguntungkan ataupun merugikan. Pengaruh yang menguntungkan akan dipertahankan untuk menjaga keberlangsungan sistem, sedangkan yang merugikan akan diupayakan mempunyai pengaruh seminimal mungkin atau ditiadakan.
- d. **Penghubung/Antarmuka (*interface*)**
Penghubung/antarmuka adalah sarana memungkinkan setiap komponen sistem, yaitu segala sesuatu yang bertugas menjembatani hubungan antar komponen dalam suatu sistem. Atau dapat dikatakan, suatu sarana tiap komponen untuk saling berinteraksi dan berkomunikasi.

- e. *Masukan (input)*
Masukan merupakan komponen sistem, yaitu segala sesuatu yang perlu dimasukkan ke dalam sistem sebagai bahan yang akan diolah lebih lanjut untuk menghasilkan keluaran yang berguna.
- f. *Pengolahan (processing)*
Pengolahan merupakan komponen sistem yang mempunyai peran utama mengolah masukan agar menghasilkan keluaran yang berguna bagi para pemakainya.
- g. *Keluaran (output)*
Keluaran merupakan komponen sistem yang berupa berbagai macam bentuk keluaran yang dihasilkan oleh komponen pengolahan.
- h. *Sasaran dan Tujuan (objectives and goal)*
Setiap komponen dalam sistem perlu dijaga agar saling bekerja sama agar mampu mencapai sasaran dan tujuan sistem.
- i. *Kendali (control)*
Setiap komponen dalam sistem perlu dijaga agar saling bekerja sesuai dengan peran dan fungsinya masing-masing.
- j. *Umpan Balik (feed back)*
Umpan balik diperlukan oleh bagian kendali untuk mengecek terjadinya penyimpangan proses dalam sistem dan mengembalikannya pada kondisi normal.

Menurut Jogiyanto (2005:3) dalam Rusdiana dan Irfan (2014), mengemukakan bahwa sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu *input*, proses dan *output*. Hal ini merupakan konsep sistem yang sangat sederhana sebab sebuah sistem dapat mempunyai beberapa masukan dan keluaran.

2.1.2 Klasifikasi Sistem

Menurut Yakub (2012) sistem dapat di klasifikasikan dari beberapa sudut pandang. Klasifikasi tersebut diantaranya:

1. Sistem Abstrak

Sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sistem berisi gagasan tentang hubungan manusia dengan Tuhan.

2. Sistem Fisik

Sistem yang ada secara fisik. Contohnya, sistem komputerisasi, sistem produksi, sistem akuntansi, sistem pendidikan dan lain sebagainya.

3. Sistem Tertentu

Sistem dengan operasi tingkah laku yang dapat diprediksi, interaksi antara bagian dapat dideteksi dengan pasti sehingga keluarannya dapat diramalkan.

4. Sistem Tertutup

Sistem yang tidak dapat bertukar materi, informasi atau energi dengan lingkungan. Sistem ini tidak berinteraksi dan tidak dipengaruhi oleh lingkungan.

5. Sistem Terbuka

Sistem yang berhubungan dengan lingkungan dan dipengaruhi oleh lingkungan. Contohnya sistem perdagangan.

2.2 Konsep Dasar Informasi

Informasi atau dalam bahasa Inggrisnya adalah *information*, berasal dari kata *informacion* dalam 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 yang bermanfaat (Rusdiana dan Moch Irfan, 2014).

Puspitawati dan Anggadini (2014) mengatakan bahwa informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Maka bisa dikatakan bahwa informasi merupakan hasil dari pengolahan data, tetapi tidak semua hasil dari pengolahan tersebut bisa menjadi informasi. Sedangkan menurut Gordon B. Davis dikutip oleh Rusdiana dan Moch Irfan (2014), informasi dapat mengenai data mentah, data tersusun, kapasitas sebuah saluran komunikasi dan sebagainya.

Dalam definisi umum informasi adalah data yang diolah menjadi bentuk yang berguna untuk membuat keputusan. Informasi berguna untuk pembuat keputusan karena informasi menurunkan ketidakpastian (meningkatkan pengetahuan). Berdasarkan informasi, para pengelola dapat mengetahui kondisi objektif perusahaannya. Informasi tersebut merupakan hasil pengolahan data atau fakta yang dikumpulkan dengan metode ataupun cara-cara tertentu (Rusdiana dan Moch Irfan, 2014).

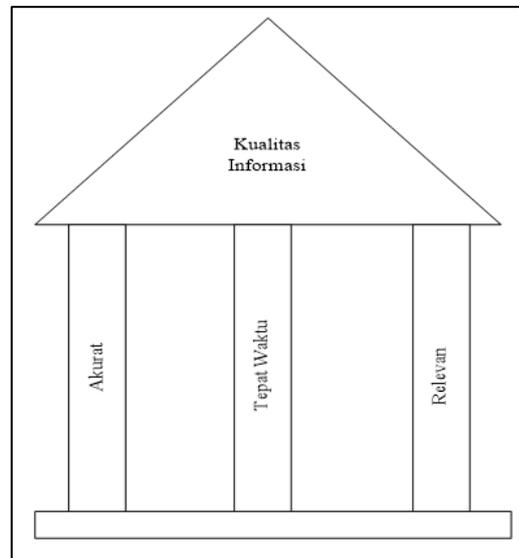
2.2.1 Kualitas Informasi

Suatu informasi yang berkualitas mempunyai ciri-ciri, yaitu (Puspitawati dan Anggadini, 2014):

1. Akurat, artinya informasi harus mencerminkan keadaan yang sebenarnya, jadi informasi bebas dari kesalahan tidak bias ataupun menyesatkan. Akurat dapat diartikan bahwa informasi itu dapat dengan jelas mencerminkan maksudnya.
2. Tepat waktu, artinya informasi harus tersedia pada saat informasi tersebut diperlukan. Informasi yang datang pada penerima tidak boleh terlambat, informasi yang sudah usang tidak lagi ada nilainya. Apabila informasi datang terlambat datang sehingga pengambilan keputusan terlambat dilakukan. Hal tersebut dapat berakibat fatal bagi perusahaan.
3. Relevan, artinya, informasi yang diberikan harus sesuai dengan yang dibutuhkan. Informasi yang disampaikan harus mempunyai keterkaitan dengan masalah yang akan dibahas. Informasi yang disampaikan harus dapat bermanfaat bagi pemakainya.
4. Lengkap, artinya informasi yang diberikan harus lengkap secara keseluruhan dalam arti tidak ada hal-hal yang dikurangi dalam hal menyampaikan informasi tersebut.

Sementara berdasarkan Burch dan Grudnitski (1989) dalam Anggareni dan Irviani (2017), berpendapat bahwa kualitas informasi dapat dianalogikan sebagai pilar-pilar dalam bangunan dan menentukan baik tidaknya pengambil keputusan. Berikut adalah pilar-pilar kualitas informasi dapat dilihat pada Gambar II.1 Kualitas informasi sering kali diukur berdasarkan:

- a. Relevansi
- b. Ketepatan waktu
- c. Keakurasian

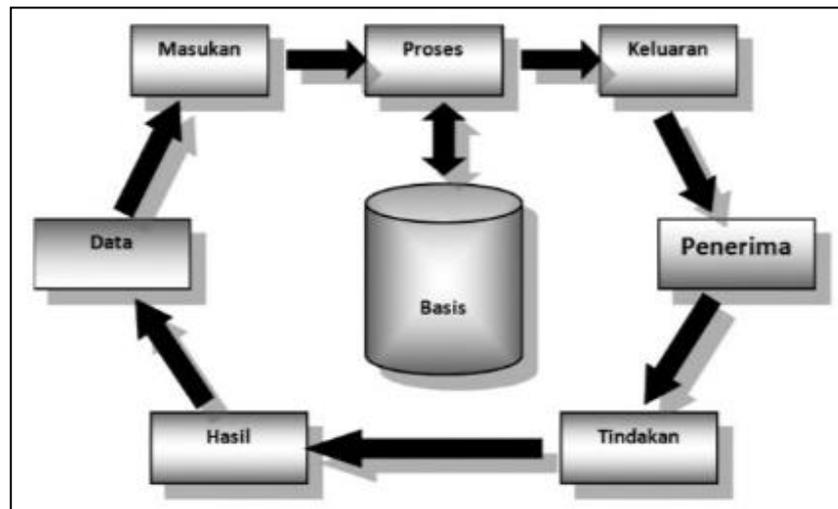


Gambar II.1 Pilar Kualitas Informasi

(Sumber: Anggraeni dan Irviani, 2017)

2.2.2 Siklus Informasi

Berdasarkan Burch dan Grudnitski (dalam Anggraeni dan Irviana, 2017), siklus informasi menggambarkan pengolahan data menjadi informasi dan pemakaian informasi untuk mengambil keputusan, hingga akhirnya dari tindakan hasil pengambilan keputusan tersebut dihasilkan data kembali. Berikut adalah gambar dari siklus informasi dalam Anggraeni dan Irviana (2017) dapat dilihat pada Gambar II.2.



Gambar II.2 Siklus Informasi dalam Angraeni dan Irviana

(Sumber: Angraeni dan Irviana, 2017)

Sementara berdasarkan Sutabri (2012), secara umum informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan. Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian adalah sesuatu yang terjadi pada saat tertentu. Di dalam dunia bisnis, kejadian-kejadian yang sering terjadi adalah transaksi perubahan dari suatu nilai yang disebut transaksi. Kesatuan nyata adalah berupa suatu obyek nyata seperti tempat, benda dan orang yang betul-betul ada dan terjadi.

Data merupakan bentuk yang masih mentah, belum dapat bercerita banyak sehingga perlu diolah lebih lanjut. Data diolah melalui suatu metode untuk menghasilkan informasi. Data dapat berbentuk simbol-simbol semacam huruf, angka, bentuk suara, sinyal, gambar, dan lain sebagainya. Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus.

2.3 Konsep Dasar Sistem Informasi

Menurut Henry C. Lucas dalam Fauzi (2017), sistem informasi adalah suatu kegiatan dari prosedur-prosedur yang diorganisasikan, bilamana dieksekusi, akan menyediakan informasi untuk mendukung pengambilan keputusan dan pengendalian di dalam organisasi. Dan berikut ini adalah pendapat para ahli mengenai definisi dari sistem informasi yang dikutip dalam Rukun dan Hayadi (2018):

- a. Menurut Turban, McLean dan Wetherbe, sistem informasi adalah sebuah sistem informasi yang mempunyai fungsi mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan yang spesifik.
- b. Menurut Bodnar dan Hopwood, sistem informasi adalah kumpulan perangkat keras dan lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna.
- c. Menurut Alter, sistem informasi adalah kombinasi antara prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah perusahaan.

2.3.1 Komponen Sistem Informasi

Berdasarkan Anggraeni dan Irviani (2017), komponen-komponen dari sistem informasi adalah sebagai berikut:

1. Komponen *input*, adalah data yang masuk ke dalam sistem informasi.
2. Komponen model, adalah kombinasi prosedur, logika dan model matematika yang memproses data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.
3. Komponen *output*, adalah hasil informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.
4. Komponen teknologi, adalah alat dalam sistem informasi, teknologi digunakan dalam menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan *output* dan memantau pengendalian sistem.

5. Komponen basis data, adalah kumpulan data yang saling terhubung yang tersimpan di dalam komputer dengan menggunakan *software database*.
6. Komponen kontrol, adalah komponen yang mengendalikan gangguan terhadap sistem informasi.

2.4 Pengertian *Production Planning & Inventory Control*

Production Planning & Inventory Control (PPIC) yaitu suatu department dalam suatu organisasi perusahaan yang berfungsi merencanakan dan mengendalikan rangkaian proses produksi agar berjalan sesuai dengan rencana yang sudah ditetapkan serta mengendalikan jumlah *inventory* agar sesuai dengan kebutuhan yang ada. Berdasarkan beberapa ahli pengertian *Production Planning & Inventory Control* (PPIC), yaitu:

Menurut Nasution (2008), perencanaan produksi dilakukan dengan menentukan arah awal dari tindakan-tindakan yang harus dilakukan, berapa banyak melakukannya, dan kapan harus melakukan. Perencanaan ini berkaitan dengan masa yang akan datang, maka perencanaan disusun atas dasar perkiraan yang dibuat berdasarkan data masa lalu dengan menggunakan asumsi-asumsi.

Menurut Assauri (2008), pengawasan persediaan merupakan salah satu kegiatan dan urutan kegiatan-kegiatan yang berkaitan erat satu sama lain dari seluruh operasi produksi tersebut sesuatu dengan apa yang telah direncanakan terlebih dahulu baik waktu, jumlah, kualitas maupun biayanya.

2.5 Konsep Penjadwalan Produksi

Konsep penjadwalan produksi meliputi pengertian produksi, tujuan penjadwalan produksi, manfaat penjadwalan produksi.

2.5.1 Pengertian Penjadwalan Produksi

Berikut menjelaskan mengenai definisi *scheduling* (penjadwalan) produksi menurut beberapa ahli:

1. *Scheduling* adalah penjadwalan kegiatan. Suatu kegiatan dijadwalkan kapan memulainya, berapa lama mengerjakan setiap tahap kegiatannya dan akhirnya

kan selesainya (Subagyo, 2005).

2. Penjadwalan (*scheduling*) produksi merupakan koordinasi waktu dalam kegiatan produksi untuk mengalokasikan bahan baku, bahan pembantu dan perlengkapan produksi pada fasilitas atau bagian pengolahan dalam pabrik sesuai dengan waktu yang telah ditetapkan, sehingga proses produksi dapat berjalan dengan lancar (Assauri, 2008).
3. Penjadwalan produksi adalah aktifitas pengalokasian sumber daya perusahaan untuk memproduksi suatu barang atau jasa dengan biaya dan tingkat persediaan yang rendah (Heizer dan Render, 2008).
4. Penjadwalan produksi adalah pengalokasian sumber daya yang terbatas untuk mengerjakan sejumlah pekerjaan (Ginting, 2009).

2.5.2 Tujuan Penjadwalan Produksi

Mengidentifikasi beberapa tujuan dari aktivitas *scheduling* sebagai berikut (Ginting, 2009):

1. Meningkatkan penggunaan sumber daya atau mengurangi waktu tunggu, sehingga total waktu proses dapat berkurang dan produktivitasnya dapat meningkat.
2. Mengurangi persediaan barang setengah jadi atau mengurangi sejumlah pekerjaan yang menunggu dalam antrean ketika sumber daya yang ada masih mengerjakan tugas yang lain. Teori Baker mengatakan jika aliran kerja suatu jadwal konstan, maka aliran antrean yang mengurangi rata-rata waktu alir akan mengurangi rata-rata persediaan barang setengah jadi.
3. Mengurangi beberapa keterlambatan pada pekerjaan yang mempunyai batas waktu penyelesaian sehingga akan meminimalisir biaya keterlambatan.
4. Membantu pengambilan keputusan mengenai perencanaan kapasitas pabrik dan jenis kapasitas yang dibutuhkan sehingga penambahan biaya dapat dihindarkan.

2.5.3 Manfaat Penjadwalan Produksi

Menurut Heizer dan Render (2008) akan pentingnya penjadwalan adalah sebagai berikut:

1. Dengan *scheduling* yang efektif, perusahaan menggunakan asetnya dengan efektif dan menghasilkan kapasitas modal yang diinvestasikan menjadi lebih besar, yang sebaliknya akan mengurangi biaya.
2. *Scheduling* menambah kapasitas dan fleksibilitas yang terkait memberikan waktu pengiriman yang lebih cepat dan dengan demikian pelayanan kepada pelanggan menjadi baik.
3. Keuntungan yang ketiga dari bagusnya penjadwalan adalah keunggulan kompetitif dengan pengiriman yang bisa diandalkan.

Fungsi pokok dari *scheduling* adalah untuk membuat arus barang dapat berjalan lancar sesuai dengan waktu yang direncanakan melalui produksi. Hal ini membuat penyelesaian dan pengiriman produk akan dilakukan tepat waktu, dan secara tidak langsung menghindari keterlambatan penerimaan kepada konsumen.

2.6 Konsep Dasar *Master Production Schedule* (MPS)

Master Production Schedule atau yang biasa juga disebut jadwal induk produksi merupakan salah satu fungsi manajemen. MPS merupakan pernyataan akhir mengenai "berapa" banyak item-item akhir yang harus diproduksi dan "kapan" harus diproduksi.

2.6.1 Pengertian *Master Production Schedule* (MPS)

Master Production Schedule merupakan suatu pernyataan tentang produksi akhir dari suatu perusahaan yang merencanakan memproduksi output berkaitan dengan kuantitas dan periode waktu. Aktifitas penjadwalan produksi induk pada dasarnya berkaitan dengan bagaimana menyusun dan memperbaharui jadwal produksi induk, memproses transaksi dari MPS, memelihara catatan-catatan MPS, mengevaluasi efektifitas dari MPS dan memberikan laporan evaluasi dalam periode waktu yang teratur untuk keperluan umpan balik dan tinjauan ulang (Gasperz,2011).

Master Production Schedule dapat dikatakan sebagai modul sentral di dalam perencanaan dan kontrol sistem manufaktur. Sebuah *Master Production Schedule* yang efektif menghasilkan sebuah basis untuk memanfaatkan sumber daya

manufaktur sebaik mungkin, memastikan pesanan pembeli terpenuhi, menyelesaikan masalah penjualan antara *sales* dan kegiatan produksi, dan mencapai tujuan strategis perusahaan sebagaimana yang ditampilkan di dalam perencanaan operasi dan penjualan. *Master Production Schedule* adalah pernyataan tentang keluaran pada masa mendatang yang direncanakan dengan membuat spesifikasi produk-produk (atau produk yang dipilih) yang akan dibuat. Sebuah MPS menghasilkan spesifikasi tentang bagaimana produk akan disimpan di persediaan untuk memenuhi permintaan pada periode mendatang.

2.6.2 Perhitungan *Master Production Schedule* (MPS)

Menurut Hartono dan Haryanti (2014), sebuah *Master Production Schedule* terdapat beberapa perhitungan, antara lain:

a. *Forecast*

Forecast merupakan perhitungan untuk meramalkan permintaan pada periode mendatang. Menggunakan rumus Moving Average Forecasting pada pesanan 6 minggu sebelumnya. Rumusnya adalah sebagai berikut:

$$6 - \text{period MAF} = \sum_{i=t-6+1}^t \text{Actual demand}_i / n$$

Gambar II.3 Rumus *Forecast* MPS

(Sumber: Hartono dan Haryanti, 2014)

b. *Master Production Schedule* (MPS)

MPS adalah jumlah komponen yang diproduksi pada satu periode. Nilai MPS berbeda tergantung dari kebijakan perusahaan. Pada PT Baneex Indonesia dapat dilihat pada jumlah pesanan setiap bulannya.

c. *Projected Available Balance* (PAB)

Projected Available Balance (PAB) digunakan untuk menghitung perkiraan kondisi persediaan setelah dikurangi oleh peramalan permintaan (*forecast*).

d. *Available to Promise* (ATP)

ATP adalah perhitungan jumlah persediaan yang bisa dijanjikan untuk permintaan dihitung dari jumlah produksi (MPS) dikurangi jumlah terbesar

dari permintaan (demand) atau peramalan (forecast). Rumusnya adalah sebagai berikut:

- Untuk periode pertama

$$ATP = On\ Hand + MPS - Sum\ Of\ the\ orders\ until\ the\ next\ MPS$$

Gambar II.4 Rumus ATP MPS 1

(Sumber: Hartono dan Haryanti, 2014)

- Untuk setiap periode MPS berikutnya

$$ATP = MPS - Sum\ of\ the\ orders\ until\ the\ next\ MPS$$

Gambar II.5 Rumus ATP MPS 2

(Sumber: Hartono dan Haryanti, 2014)

- Contoh Tabel MPS

	Period		
	1	2	3
Forecast	10	10	10
Orders	15	10	20
Projected Available Balance	35		
Available-to-Promise	25		
Master Production Schedule	30		30
On Hand = 20			
Lot Size = 30			
Safety Stock = 5			

Gambar II.6 Contoh Tabel MPS

(Sumber: Hartono dan Haryanti, 2014)

e. *Safety Stock*

Safety Stock adalah stok tambahan dari item yang direncanakan untuk berada dalam inventori yang dijadikan sebagai stok pengaman guna mengatasi fluktuasi dalam ramalan penjualan, pesanan-pesanan pelanggan dalam waktu singkat, penyerahan item untuk pengisian kembali inventori, dan lain-lain.

f. *Lot Size*

Lot Size merupakan suatu Teknik yang digunakan untuk menentukan ukuran kuantitas pemesanan.

2.7 Telaah Jurnal

Berdasarkan telaah jurnal yang dilakukan terhadap 2 penelitian yaitu Hartono dan Haryanti (2014) dengan judul “Aplikasi Penjadwalan Produksi Manufaktur Berbasis Web Dengan Metode *Master Production Scheduling*”, Supriyadi dan Riskiyadi (2016) dengan judul “Penjadwalan Produksi IKS-FILLER Pada Proses *Ground Calcium Carbonate* Menggunakan Metode MPS Di Perusahaan Kertas”. Kedua jurnal tersebut membahas tentang penjadwalan produksi dengan metode MPS.

Ruang lingkup yang dibahas oleh Hartono dan Haryanti (2014), Dari hasil pembahasan tentang aplikasi penjadwalan produksi manufaktur berbasis web dengan metode MPS maka dapat diambil kesimpulan:

1. Aplikasi yang dibuat sudah dapat merencanakan kegiatan produksi berdasarkan jumlah stok persediaan dan pesanan yang datanya selalu berubah-ubah setiap waktu.
2. Metode MPS belum dapat diimplementasikan ke dalam aplikasi penjadwalan produksi manufaktur berbasis web yang dibuat.

Sedangkan ruang lingkup yang dibahas oleh Supriyadi dan Riskiyadi (2016). Menurut yang ada dalam jurnal tersebut, metode MPS atau biasa disebut dengan *Master Production Scheduling* adalah suatu metode perencanaan produksi jangka pendek pada suatu perusahaan yang berisi tentang rencana menyeluruh serta perincian dalam menghasilkan produk akhir. Metode MPS yang sesuai untuk diimplementasikan perusahaan berdasarkan data historis adalah metode peramalan.

Perbedaan antara penelitian tersebut adalah pada ruang lingkup penelitian. Penelitian yang dibahas oleh Iwan Hartono & Haryanti (2014) adalah membuat rancangan sistem informasi material produk berbasis *website* dengan metode MPS. Adapun beberapa perancangan tersebut menggunakan pengelolaan stok, yaitu *demand, forecast, on hand, mps, available to promise, dan projected available balance*. Sementara jurnal yang dibahas oleh Supriyadi dan Riskiyadi (2016) adalah membahas tentang teori dan penyelesaian permasalahan penjadwalan menggunakan metode MPS.

Penelitian yang sedang dilakukan dalam Tugas Akhir ini memiliki ruang lingkup yang sama dengan kedua jurnal tersebut yakni membahas penjadwalan barang dengan menggunakan metode MPS dan menerapkannya untuk merancang sistem informasi penjadwalan produksi dengan menggunakan diagram *use case*, diagram *sequence*, diagram *class* serta menggunakan model *waterfall*.

Pada Tugas Akhir ini aplikasi dikembangkan dengan CodeIgniter dan MariaDB. Berbeda dengan Supriyadi dan Riskiyadi (2016) yang membahas penjadwalan produksi pada perusahaan kertas, pada penelitian ini, metode MPS diterapkan untuk penjadwalan produksi pada perusahaan otomotif.

2.8 System Development Life Cycle (SDLC)

Menurut Dennis et al (2015) *System Development Life Cycle (SDLC)* memiliki 4 perangkat fase dasar yaitu *planning*, *analysis*, *design*, dan *implementation*. Beberapa proyek dalam membangun sebuah sistem informasi mungkin dapat berbeda-beda sesuai dengan caranya masing-masing, tetapi hampir semua proyek memiliki elemen dari 4 fase tersebut. Masing-masing fase tersebut tersusun dari beberapa langkah-langkah yang menghasilkan *deliverable* / hasil kegiatan contohnya beberapa dokumen spesifik dan file yang menjelaskan pemahaman tentang proyek.

Dalam beberapa proyek, tahapan dan langkah-langkah SDLC diproses dalam suatu alur tahapan, dimulai dari awal hingga akhir. Dalam proyek lain, tim proyek terus berpindah dari fase ke fase tersebut secara teratur, bertahap, iteratif, atau dalam bentuk pola lainnya. Dalam SDLC, dijelaskan fase-fase, tindakan, serta beberapa teknik yang digunakan untuk menyempurnakan langkah-langkah yang dijelaskan secara umum. Ada dua hal penting untuk dipahami tentang SDLC:

1. Mendapatkan dan memahami pengertian umum dari tahapan dan langkah-langkah yang digunakan dalam dalam proyek Sistem Informasi serta beberapa teknik yang menghasilkan dokumen yang pasti.
2. Kedua hal penting untuk dipahami bahwa SDLC merupakan proses penyempurnaan yang dilakukan secara bertahap.

Dokumen yang dihasilkan dalam tahap analisis, memberikan ide umum dari suatu bagian dari sistem baru. Dokumen dari *deliverable* ini digunakan sebagai *input* pada tahap *design*, yang kemudian disempurnakan untuk menghasilkan dokumen yang menjelaskan secara detail dari sistem yang dibuat. Dokumen ini, akan digunakan dalam tahap implementasi untuk menghasilkan sistem yang sebenarnya. Berikut dibawah ini merupakan penjabaran dari setiap fase dalam SDLC:

1. *Planning* (Perencanaan)

Tahap *planning* / perencanaan adalah proses dasar yang menjelaskan mengapa sebuah sistem informasi harus dibangun dan menentukan bagaimana sebuah tim proyek akan membangunnya. Terdapat dua langkah dalam *planning* (perencanaan) yaitu:

- a. Inisiasi proyek, pada tahap ini nilai bisnis suatu sistem terhadap organisasi diidentifikasi: Apakah sistem tersebut akan menurunkan biaya atau meningkatkan pendapatan? Sebagian besar gagasan untuk sistem baru berasal dari luar area Departemen IT (misalnya dari Departemen Pemasaran, Departemen Akuntansi) dalam suatu bentuk *system request* (permintaan sistem). Permintaan sistem menyajikan ringkasan singkat tentang kebutuhan sistem dalam suatu bisnis dan menjelaskan bagaimana suatu sistem akan menciptakan nilai bisnis. Departemen IT bekerja sama dengan departemen lain yang menghasilkan permintaan (disebut sponsor proyek) untuk melakukan analisis kelayakan sistem informasi. *System request* (permintaan sistem) dan analisis kelayakan disajikan dan dipresentasikan kepada pihak komite persetujuan *Information System* (komite pengarah), yang menentukan apakah proyek tersebut harus diambil atau tidak.
- b. Setelah proyek disetujui, proyek memasuki manajemen proyek. Selama manajemen proyek, manajer proyek membuat sebuah rencana kerja, membentuk anggota staff proyek, dan anggota teknisi lainnya di tempat, untuk membantu tim proyek mengendalikan dan mengarahkan proyek pada keseluruhan tahap SDLC.

2. *Analysis* (Analisis)

Analisis menurut Muslihudin dan Oktafianto (2016) merupakan tahapan awal dalam pengembangan sistem dan merupakan tahap fundamental yang sangat menentukan kualitas sistem yang akan dikembangkan. Tahap analisis dalam SDLC menjelaskan siapa yang akan memakai sistem, apa yang sistem kerjakan serta kapan dan dimana sistem akan digunakan. Selama pada tahap ini, tim proyek menyelidiki sistem yang ada saat ini, mengidentifikasi peluang untuk melakukan perbaikan, dan mengembangkan sistem baru. Tahap ini memiliki tiga langkah:

- a. Langkah yang pertama adalah Strategi Analisis. Langkah ini dikembangkan untuk mengarahkan usaha dari tim proyek untuk menganalisis sistem yang telah ada (disebut sebagai *as-is system*) beserta masalah-masalah yang ada dan untuk merancang suatu sistem yang baru (disebut sebagai *to-be system*).
 - b. Langkah selanjutnya adalah mengumpulkan kebutuhan-kebutuhan sistem (melalui wawancara atau kuesioner). Pada tahap ini terdapat juga masukan analisis informasi dari sponsor proyek dan pihak lainnya yang mengarah pada pengembangan konsep untuk sistem baru. Konsep sistem ini kemudian digunakan sebagai dasar untuk mengembangkan seperangkat model analisis bisnis, yang menggambarkan bagaimana bisnis akan beroperasi jika sistem baru dikembangkan.
 - c. Kemudian Analisis, konsep sistem, dan model digabungkan menjadi dokumen yang disebut proposal sistem, yang dipresentasikan ke sponsor proyek dan pengambil keputusan utama lainnya (misalnya, anggota komite persetujuan) yang memutuskan apakah proyek tetap terus dilanjutkan.
3. *Design* (Perancangan)
- Tahap *design* atau perancangan memutuskan bagaimana sistem akan beroperasi, mulai dari hal yang berkaitan dengan perangkat keras (*hardware*), perangkat lunak (*software*) dan infrastruktur jaringan seperti:
- a. Antarmuka pengguna (*interface*),
 - b. Formulir,
 - c. Laporan,
 - d. Program spesifik: *database*, dan *file-file* lainnya yang akan dibutuhkan.

Meskipun sebagian besar keputusan strategis mengenai sistem dibuat dalam pengembangan konsep sistem selama tahap analisis, langkah-langkah dalam tahap desain menentukan secara tepat bagaimana sistem akan beroperasi. Fase desain ini memiliki empat langkah:

- a. Strategi perancangan (*Design Strategy*) adalah hal yang pertama kali yang dikembangkan. Strategi perancangan menjelaskan apakah sistem akan dikembangkan oleh programmer perusahaan sendiri, atau akan di-*outsourcing* ke perusahaan lain (biasanya perusahaan konsultan), atau apakah perusahaan akan membeli paket perangkat lunak yang sudah ada.
- b. Pengembangan desain arsitektur (*architecture design*) yang mengarah pada dasar untuk sistem, yang menggambarkan perangkat keras, perangkat lunak, dan infrastruktur jaringan yang akan digunakan. Dalam banyak kasus, sistem akan menambah atau mengubah infrastruktur yang sudah ada dalam suatu organisasi. Desain antarmuka menentukan bagaimana pengguna akan berpindah ke sistem yang baru (misalnya, metode navigasi seperti menu dan tombol di layar) serta laporan yang akan digunakan oleh sistem.
- c. Pengembangan *database* dan spesifikasi *file*. Pengembangan ini menentukan dengan tepat data apa yang akan disimpan dan di mana data tersebut akan disimpan.
- d. Tim analis mengembangkan desain program (*Program Design*), yang mendefinisikan program yang perlu dikembangkan dan apa yang akan dilakukan masing-masing program.

Kumpulan dari hasil kegiatan ini seperti desain arsitektur, desain antarmuka, spesifikasi *database* dan *file*, dan perancangan program merupakan spesifikasi sistem yang diserahkan ke tim pemrograman untuk diimplementasi. Pada akhir tahap desain, analisis kelayakan dan rencana proyek diperiksa ulang dan diperbaiki, dan keputusan lain dibuat oleh sponsor proyek dan komite persetujuan mengenai apakah proyek dihentikan atau dilanjutkan.

4. *Implementation* (Implementasi)

Tahap akhir dalam SDLC adalah tahap implementasi, di mana sistem benar-benar dibangun (atau dibeli, dalam kasus paket *design software*). Fase ini

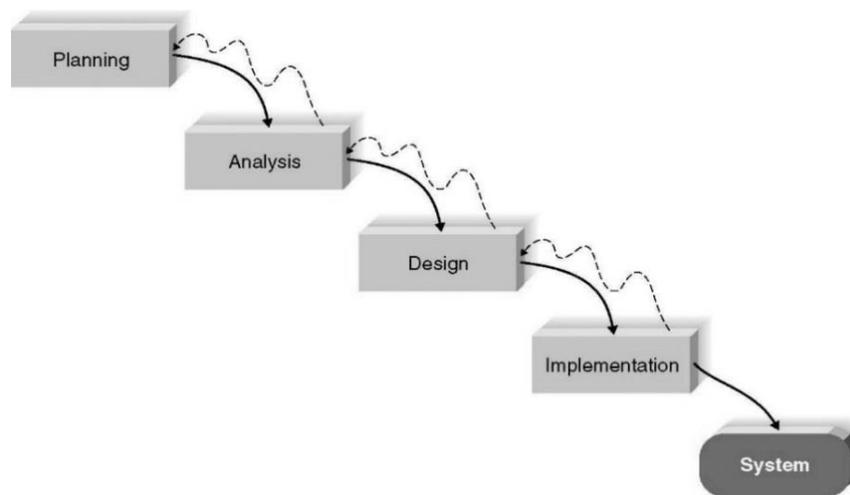
sangat penting, karena kebanyakan sistem merupakan bagian terlama dan paling mahal dari proses pengembangan. Fase implementasi ini memiliki tiga langkah:

- a. Langkah awal adalah Konstruksi Sistem (*System Construction*). Sistem ini dibangun dan diuji untuk memastikan kinerja dari sistem, bekerja sebagaimana yang telah dirancang. Karena biaya *bugs* bisa menjadi sangat besar, pengujian merupakan salah satu langkah paling kritis dalam tahap implementasi. Sebagian besar organisasi memberi lebih banyak waktu dan perhatian untuk menguji daripada mengubah program di tempat penerapan pertama.
- b. Instalasi Sistem, instalasi adalah proses dimana sistem lama dimatikan dan yang baru dinyalakan atau dihidupkan. Salah satu aspek terpenting dari konversi (perubahan dari suatu sistem ke sistem yang baru) adalah pengembangan rencana pelatihan (*training plan*) untuk mengajarkan pengguna cara menggunakan sistem baru dan membantu mengelola perubahan-perubahan yang disebabkan oleh sistem yang baru.
- c. Tim analis menetapkan sebuah rencana dukungan (*support plan*) untuk sistem. Rencana ini biasanya mencakup tinjauan pasca-pelaksanaan formal atau informal serta cara sistematis untuk mengidentifikasi perubahan besar dan kecil yang diperlukan untuk sistem tersebut.

2.9 Waterfall

Berdasarkan Dennis *et al* (2015), contoh model yang masih digunakan hingga saat ini adalah *waterfall development* atau disebut juga dengan model pengembangan “air mancur”. Dengan metode pengembangan *waterfall*, analis dan pengguna atau *user* memproses secara berurutan dari satu fase ke fase berikutnya. Biasanya proses dalam satu fase sangat panjang dan ditampilkan dalam sponsor proyek atau *project sponsor* untuk disetujui sehingga dapat berlanjut ke fase selanjutnya. Suatu fase atau tahapan akan berlanjut jika sponsor proyek telah menyetujui tahapan tersebut. Model ini disebut dengan *waterfall development*, karena tahap demi tahapan bergerak maju dengan cara yang sama seperti air mancur. Tetapi, dalam model ini suatu tahapan sulit untuk bergerak mundur, (contohnya dari

tahap desain mundur ke tahap analisis). Terdapat 5 langkah tahapan dalam metode *waterfall* adalah sebagai berikut:



Gambar II.7 Metode *Waterfall*

(Sumber: Dennis et al, 2015)

2.10 *Flowmap*

Flowmap adalah campuran peta dan *flowchart* yang menunjukkan pergerakan benda dari suatu lokasi ke lokasi lain, seperti jumlah orang dalam migrasi, jumlah barang yang diperdagangkan, atau jumlah paket dalam jaringan. *Flowmap* menolong analis dan *programmer* untuk memecahkan masalah kedalam segmen yang lebih kecil dan menolong dalam menganalisis alternatif lain dalam pengoperasian (Jogiyanto, 2010). Adapun pedoman-pedoman dalam pembuatan *flowmap* adalah sebagai berikut:

1. *Flowmap* sebaiknya digambarkan dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
2. Kegiatan di dalam *flowmap* harus ditunjukkan dengan jelas.
3. Harus ditunjukkan dari mana kegiatan akan dimulai dan dimana akan berakhir.
4. Masing-masing kegiatan didalam *flowmap* sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan.

5. Masing-masing kegiatan didalam *flowmap* harus didalam urutan yang semestinya.
6. Kegiatan yang terpotong dan akan disambung ditempat lain harus ditunjukkan dengan jelas menggunakan simbol penghubung.
7. Gunakan simbol simbol *flowmap* yang standar.

Adapun simbol-simbol yang sering digunakan dalam *flowmap* dapat dilihat pada Tabel II.1 berikut ini:

Tabel II.1 Simbol-simbol *Flowmap*

No	Gambar	Nama	Keterangan
1.		Mulai/Akhir	Menunjukkan dimulainya/akhir dari sebuah proses
2.		Proses	Menunjukkan kegiatan proses dari operasi program komputer
3.		Kegiatan Manual	Menunjukkan kegiatan manual
4.		Garis Alir	Menunjukkan arus dari proses
5.		Keputusan	Keputusan dalam suatu program
6.		Dokumen	Menunjukkan dokumen <i>input/output</i> baik untuk proses manual, mekanik, atau komputer
7.		<i>Keyboard</i>	Menunjukkan <i>input</i> yang menggunakan <i>on-line keyboard</i>

(Sumber: Jogiyanto, 2010)

2.11 *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 et al (2015), diagram-diagram yang ada dalam UML terbagi menjadi dua bagian utama yaitu, *Structure diagram* dan *Behavior diagram*. *Structure diagram* biasanya digunakan untuk mepresentasikan 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. Berikut adalah tabel dari jenis diagram yang ada dan fungsi-fungsinya:

Tabel II.2 Jenis Diagram Pada UML dan Fungsi-fungsinya

Nama Diagram	Fungsi
<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>
<i>Composite Diagram</i>	Menggambarkan struktur internal dari suatu <i>class</i> dan hubungan diantara bagian-bagian dari suatu <i>class</i>
<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

Tabel II.2 Jenis Diagram Pada UML dan Fungsi-fungsinya (Lanjutan)

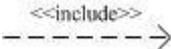
Nama Diagram	Fungsi
<i>Communication Diagram</i>	Memodelkan kebiasaan dari objek-objek dalam suatu <i>use case</i> . Fokus dalam komunikasi antara satu set dari kolaborasi objek dari suatu objek
<i>Interaction Overview Diagram</i>	Menggambarkan suatu ikhtisar alur dari kontrol suatu proses
<i>Timing Diagram</i>	Menggambarkan interaksi yang terjadi diantara suatu set dari objek-objek dan perubahan keadaan selama perjalanan waktu
<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 et al, 2015)

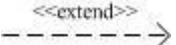
2.11.1 Use Case Diagram

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. Simbol-simbol yang terdapat didalam *use case diagram* dijelaskan pada Tabel II.3 dibawah ini:

Tabel II.3 Simbol-simbol *Use case Diagram*

No	Simbol	Nama	Deskripsi
1		<i>Use Case</i>	<ol style="list-style-type: none"> 1. Merepresentasikan bagian utama dari sistem secara fungsional. 2. Diletakkan di dalam <i>system boundary</i>. 3. Dilabelkan dengan frasa kata kerja deskriptif.
2		<i>Association</i>	Menghubungkan suatu aktor dengan <i>use case</i> .
3	 <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;"> <p><<actor>> Actor/Role</p> </div>	<i>Actor</i>	<ol style="list-style-type: none"> 1. Seseorang atau sistem yang mendapatkan keuntungan dari sistem. 2. Diletakkan di luar batas sistem.
4		<i>System Boundary</i>	<ol style="list-style-type: none"> 1. Nama dari sistem. 2. Merepresentasikan ruang lingkup dari sistem.
5		<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.

Tabel II.3 Simbol-simbol *Use case Diagram* (Lanjutan)

No	Simbol	Nama	Deskripsi
6		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan ini.
7		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.

(Sumber: Dennis et al, 2015)

2.11.2 *Use Case Description*

Use Case Description merupakan deskripsi yang mencatat mengenai detail pemrosesan dari suatu *use case* (Satzinger et al, 2010). *Use Case Description* memiliki urutan yang lengkap dari tahapan-tahapan untuk menyelesaikan suatu proses bisnis. Menurut Dennis et al (2010:168), elemen yang dimiliki *Use Case Description* adalah sebagai berikut:

1. *Basic Information*

a. Nama

Nama digunakan untuk menjelaskan *use case* dan membedakan *use case* satu dengan yang lainnya. Nama yang digunakan harus sederhana namun mampu menjelaskan kondisi yang memungkinkan.

b. *Description*

Description berarti uraian singkat untuk menyampaikan tujuan dari *use case*.

c. Aktor

Aktor merujuk kepada orang, sistem, perangkat lunak ataupun perangkat keras yang berinteraksi langsung dengan sistem untuk mencapai tujuan.

2. *Normal Courses*

Bagian utama dari *use case* berikutnya adalah *normal courses*. Pada *normal courses* menjelaskan mengenai langkah-langkah yang dilakukan untuk mengeksekusi respon terhadap masukan dan keluaran.

3. *Alternative Courses*

Untuk beberapa *use case* memiliki jalur alternatif pada setiap kasusnya. Pada penggambaran jalur alternatif ini terdapat cabang yang juga akan menghasilkan kesimpulan dari *use case* tersebut.

2.11.3 *Activity Diagram*

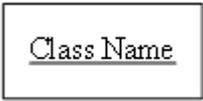
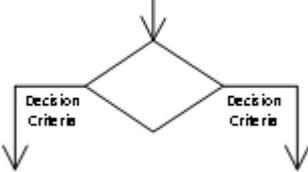
Activity diagram digunakan untuk memodelkan perilaku dalam proses bisnis yang independen terhadap objek. Dalam banyak hal, *activity diagram* dapat dipandang sebagai data flow diagram canggih yang digunakan pada konjungsi dengan analisis terstruktur. Walaupun begitu, *activity diagram* tidak seperti data *flow diagram*, *activity diagram* menggunakan notasi yang mengalamatkan pemodelan paralel. (Dennis et al, 2015)

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

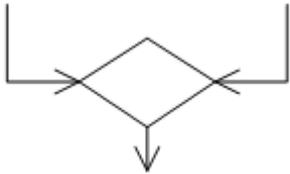
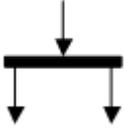
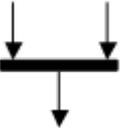
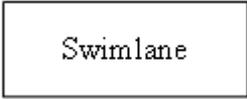
1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan
3. Rancangan pengujian dimana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

Berikut adalah simbol-simbol yang ada pada Activity Diagram:

Tabel II.4 Simbol-simbol *Activity Diagram*

No.	Simbol	Nama	Deskripsi
1.		<i>Action</i>	Merupakan notasi yang <i>simple</i> , dan tidak untuk dianalisis perilakunya. Dinamakan sesuai aksi yg dilakukan.
2.		<i>Activity</i>	Digunakan untuk mewakili satu <i>set</i> aksi yang dilakukan.
3.		<i>Object Node</i>	Digunakan untuk mewakili objek yang terhubung ke satu set arus objek.
4.		<i>Control Flow</i>	Menampilkan urutan eksekusi.
5.		<i>Object Flow</i>	Menunjukkan aliran dari sebuah objek dari satu aktivitas ke aktivitas lainnya.
6.		<i>Initial Node</i>	Menggambarkan permulaan dari suatu set aktivitas atau aksi.
7.		<i>Final-activity Node</i>	Menggambarkan akhir dari setiap aktivitas.
8.		<i>Final-flow Node</i>	Menggambarkan akhir aliran control spesifik atau aliran objek.
9.		<i>Decision Node</i>	Menggambarkan suatu kondisi untuk memastikan bahwa arah panah atau arah objek hanya memiliki satu jalur ke bawah.

Tabel II.4 Simbol-simbol *Activity Diagram* (Lanjutan)

No.	Simbol	Nama	Deskripsi
10.		<i>Merge Node</i>	Digunakan untuk membawa aliran keputusan yang berbeda ke satu <i>decision node</i> .
11.		<i>Fork Node</i>	Menggambarkan percabangan dari satu aliran aktivitas.
12.		<i>Join Node</i>	Menyatukan beberapa percabangan dari aliran aktivitas.
13.		<i>Swimlane</i>	Menggambarkan akhir aliran kontrol spesifik atau aliran objek.

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

2.11.4 *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 et al, 2015). Berikut simbol yang ada pada *Sequence Diagram*:

Tabel II.5 Simbol-simbol *Sequence Diagram*

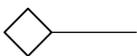
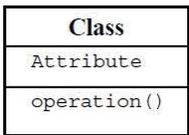
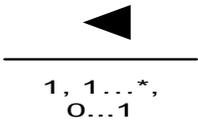
No	Simbol	Nama	Deskripsi
1.	 <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;"> <<actor>> Actor/Role </div>	Aktor	Orang atau sistem yang berasal dari manfaat dan eksternal ke sistem yang berpartisipasi secara berurutan dengan mengirim dan atau menerima pesan
2.		Garis Hidup/ <i>lifeline</i>	Menyatakan kehidupan suatu objek.
3.	<div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;"> nama_objek : nama_kelas </div>	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>	Pesan yang menggambarkan komunikasi yang terjadi antar objek.
6.		<i>Message (return)</i>	Pesan yang dikirim untuk diri sendiri secara langsung.
7.		<i>Message (return)</i>	Pesan yang dikirim untuk diri sendiri.
8.	X	<i>Object Destruction</i>	Sebuah x ditempatkan di ujung suatu objek garis hidup untuk menunjukkan bahwa akan keluar dari eksistensi.
9.	<div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;"> <i>Context</i> </div>	<i>Frame</i>	Menunjukkan konteks dari <i>sequence diagram</i> .

(Sumber: Dennis et al, 2015)

2.11.5 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 (metode/fungsi) (Dennis et al, 2015). Berikut simbol-simbol yang ada pada diagram kelas:

Tabel II.6 Simbol-simbol *Class Diagram*

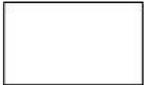
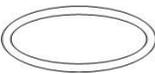
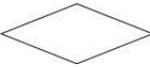
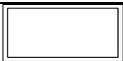
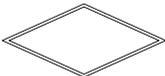
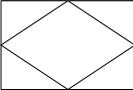
No	Simbol	Nama	Deskripsi
1.		<i>Generalization</i>	Merupakan sebuah <i>taxonomic relationship</i> antara <i>class</i> yang lebih umum dengan <i>class</i> yang lebih khusus.
2.		<i>Aggregation</i>	Mewakili hubungan <i>logic</i> antara <i>class</i> dan beberapa <i>class</i> .
3.		<i>Class</i>	Kelas pada struktur sistem.
4.		<i>Association</i>	Asosiasi yang menghubungkan <i>class</i> dengan beberapa <i>class</i> . Bisa ada di antara satu atau lebih kelas.
5.		<i>Composition</i>	Mewakili hubungan fisik antara <i>class</i> dan beberapa <i>class</i> .

(Sumber: Dennis et al, 2015)

2.12 Entity Relationship Diagram (ERD)

Terdapat sebuah model perancangan hubungan antar entitas (tabel) dari sebuah basis data (*database*). Istilah untuk frase ini biasa dikenal dengan nama *Entity Relationship Model*. Model hubungan ini seterusnya akan berlanjut menjadi sebuah Diagram Hubungan Antar Entitas yang biasa dikenal dengan nama *Entity Relationship Diagram* (ERD) (Jeffrey A. Hoffer, 2007). Simbol-simbol ERD dapat dilihat pada Tabel II.7.

Tabel II.7 Simbol-Simbol ERD

Simbol	Nama	Deskripsi
	<i>Strong Entity</i>	Karakteristik atau tipe entitas yang tidak bergantung pada entitas lain.
	<i>Attribute</i>	Properti atau karakteristik tipe entitas.
	<i>Multivalued Attribute</i>	Karakteristik tipe entitas yang dapat memiliki nilai lebih dari satu.
	<i>Relationship</i>	Hubungan antar entitas.
	<i>Weak Entity</i>	Entitas yang bergantung pada entitas lain.
	<i>Identifying Entity</i>	Entitas yang merupakan induk dari <i>weak entity</i> .
	<i>Associative Entity</i>	Hanya bisa terdapat antara 2 entitas dan menggambarkan hubungan diantara 2 entitas tersebut.

Tabel II.7 Simbol-Simbol ERD (Lanjutan)

Simbol	Nama	Deskripsi
	<i>Derived Attribute</i>	Suatu atribut yang dihasilkan dari atribut yang lain.

(Sumber: Jeffrey A. Hoffer, 2007)

2.13 Sistem Manajemen Basis Data

Berdasarkan Pamungkas (2017), sistem manajemen basis data merupakan perangkat lunak yang didesain untuk melakukan penyimpanan dan pengaturan basis data. Sementara basis data, merupakan suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, dan dengan *software* untuk memanipulasi dengan tujuan tertentu.

2.13.1 Database

Berikut adalah pengertian dari *database* menurut para ahli yang tercantum dalam Jalinus dan Ambiyar (2016):

1. Berdasarkan Connolly dan Begg (2010: 65), *database* adalah sekumpulan data tersebar yang berhubungan secara logis, dan penjelasan dari data ini dirancang untuk memenuhi kebutuhan informasi dari suatu organisasi.
2. Menurut Gottschalk dan Saether dalam jurnal (2010:41), *database* adalah sekumpulan data yang terorganisasi untuk mendukung banyak aplikasi secara efisien dengan memusatkan data dan mengontrol *data redundant*.

Maka dapat disimpulkan *database* adalah sekumpulan data yang saling terhubung dan terorganisasi yang disimpan berdasarkan skema dengan memusatkan data dan mengontrol *data redundant* untuk memenuhi kebutuhan informasi dari suatu organisasi.

Berdasarkan Suryo (2012), *database* adalah kumpulan informasi yang disimpan dalam computer secara sistematis untuk memperoleh informasi dari basis data tersebut. Konsep dasar dari *database* adalah kumpulan dari catatan, atau

potongan dari pengetahuan. Sebuah *database* memiliki penjelasan terstruktur dari jenis fakta yang disimpan didalamnya, penjelasan itu disebut dengan skema. Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur *database*, yang dikenal dengan *database* model atau model data. Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah yaitu mewakili semua informasi dalam bentuk tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom. Contoh dari tabel barang dapat dilihat pada Tabel II.8.

Tabel II.8 Contoh Tabel Barang

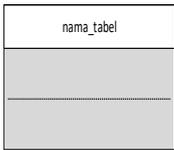
<i>Field</i>	<i>Field Name</i>	<i>Type</i>	<i>Width</i>	<i>Description</i>
1	Kode_Barang	Varchar	7	Kode Barang*
2	Nama_Barang	Varchar	35	Nama Barang
3	Satuan	Varchar	15	Satuan
4	Harga	Varchar	9	Harga

(Sumber: Suryo, 2012)

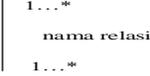
2.13.2 Conceptual Data Modelling (CDM)

Menurut Rosa dan Shalahuddin (2015) 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.9 Simbol-Simbol *Conceptual Data Model* (CDM)

Simbol	Deskripsi
Entitas/tabel 	Entitas atau tabel yang menyimpan data dalam basis data

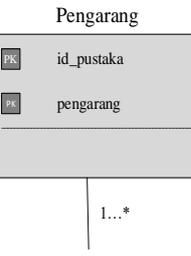
Tabel II.9 Simbol-Simbol *Conceptual Data Model* (CDM) (Lanjutan)

Simbol	Deskripsi
<p>Relasi</p> 	Relasi antar tabel yang terdiri atas nama relasi dan <i>multiplicity</i>

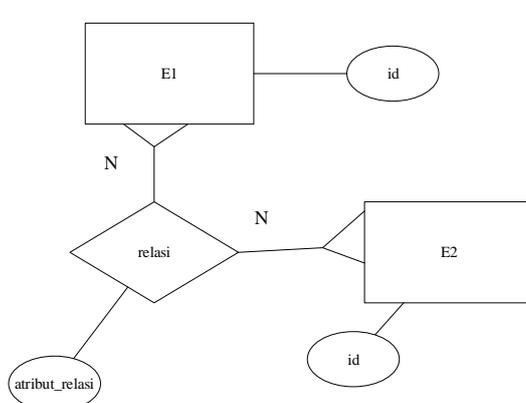
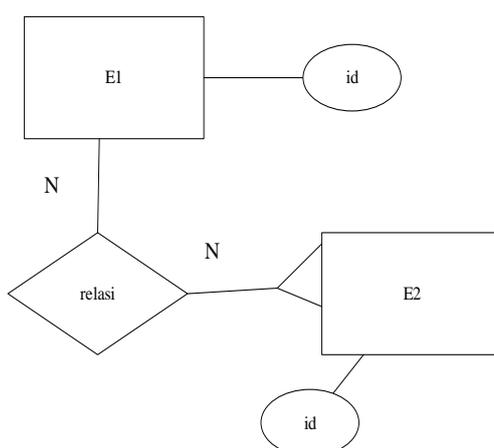
(Sumber: Rosa dan Shalahuddin, 2015)

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

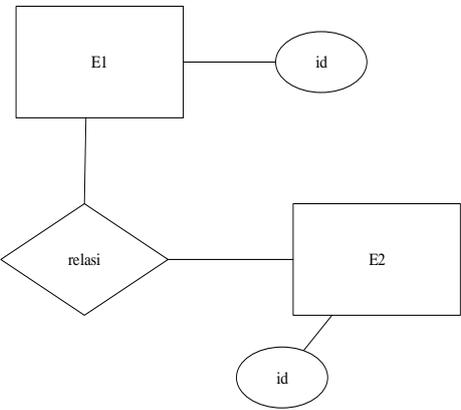
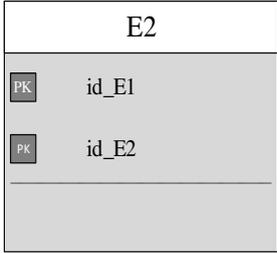
Tabel II.10 Simbol-Simbol Aturan *Conceptual Data Model* (CDM)

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

Tabel II.10 Simbol-Symbol Aturan *Conceptual Data Model* (CDM) (Lanjutan)

ERD	CDM								
 <p>Relasi dengan kardinalitas <i>many to many</i></p>	<table border="1" data-bbox="973 470 1236 683"> <thead> <tr> <th colspan="2">Relasi</th> </tr> </thead> <tbody> <tr> <td>PK</td> <td>id_E1</td> </tr> <tr> <td>PK</td> <td>id_E2</td> </tr> <tr> <td colspan="2">atribut_relasi</td> </tr> </tbody> </table> <p>Menjadi sebuah tabel tersendiri dengan kunci primer adalah atribut yang menjadi kunci primer di kedua entitas yang direlasikannya.</p>	Relasi		PK	id_E1	PK	id_E2	atribut_relasi	
Relasi									
PK	id_E1								
PK	id_E2								
atribut_relasi									
 <p>Relasi dengan kardinalitas <i>one to many</i></p>	<table border="1" data-bbox="957 996 1252 1254"> <thead> <tr> <th colspan="2">E2</th> </tr> </thead> <tbody> <tr> <td>PK</td> <td>id_E1</td> </tr> <tr> <td>PK</td> <td>id_E2</td> </tr> </tbody> </table> <p>Kunci primer entitas yang memiliki hubungan <i>one</i> akan dijadikan kunci primer di entitas yang memiliki hubungan <i>many</i> dengan kata lain, relasi tidak menjadi tabel sendiri</p>	E2		PK	id_E1	PK	id_E2		
E2									
PK	id_E1								
PK	id_E2								

Tabel II.10 Simbol-Symbol Aturan *Conceptual Data Model* (CDM) (Lanjutan)

ERD	CDM
 <p data-bbox="316 1014 759 1043">Relasi dengan kardinalitas <i>one to one</i></p>	 <p data-bbox="885 808 1337 1088">Kunci primer salah satu entitas akan dijadikan kunci asing (<i>foreign key</i>) pada tabel yang lain dan kunci asing itu dijadikan kunci primer juga, dengan kata lain, relasi tidak menjadi tabel sendiri</p>

(Sumber: Rosa dan Shalahuddin, 2015)

2.13.3 Kamus Data

Menurut Jogiyanto (2010) dalam buku Analisis dan Desain, menjelaskan bahwa: "Kamus Data merupakan katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. dalam kamus data harus memuat hal-hal berikut:

1. Nama Arus Data. Nama arus data dicatat pada kamus data, sehingga mereka yang membaca DAD memerlukan penjelasan lebih lanjut tentang suatu arus data tertentu.
2. Alias. Untuk menyatakan nama lain dari element atau data store yang sebenarnya sama dengan data element atau data *store* yang telah ada.
3. Bentuk Data. Dipergunakan untuk mengelompokkan kamus data ke dalam kegunaanya sewaktu perancangan sistem.

4. Arus Data. Menunjukkan dari mana data mengalir dan kemana data akan menuju, keterangan arus data ini perlu dicatat di kamus data supaya memudahkan mencari arus data ini di DAD.
5. Struktur data. Struktur data menunjukkan harus data yang dicatat di kamus data terdiri dari item - item data apa saja.

Berdasarkan definisi di atas maka dapat disimpulkan bahwa kamus data merupakan suatu bantuan yang berguna untuk kebutuhan informasi dari suatu sistem informasi.

2.13.4 MariaDB

Berdasarkan mariadb.com (2018), MariaDB adalah *database* sumber terbuka yang paling cepat berkembang. Pertumbuhan MariaDB dipercepat oleh 12 juta pengguna di 45 negara yang tidak hanya menjalankan aplikasi *mission-critical* tetapi juga terus berinovasi dan secara aktif menyumbangkan teknologi baru untuk proyek-proyek MariaDB. Selain itu, MariaDB adalah *database default* untuk distribusi Linux seperti Red Hat dan SUSE, dan tersedia melalui penyedia *cloud* besar seperti Amazon AWS dan Microsoft Azure.

MariaDB merupakan versi pengembangan terbuka dan mandiri dari MySQL. Sejak diakuisisinya MySQL oleh Oracle pada September 2010, Monty Program sebagai penulis awal kode sumber MySQL memisahkan diri dari pengembangan dan membuat versi yang lebih mandiri yakni MariaDB.

MariaDB adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Setiap pengguna dapat secara bebas menggunakan MariaDB, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial.

MariaDB dibangun di atas arsitektur modern yang dapat diperluas di setiap lapisan: klien, kluster, kernel, dan penyimpanan. Perpanjangan ini memberikan dua keuntungan besar. Hal ini memungkinkan inovasi komunitas berkelanjutan melalui *plugin*, yang berarti bahwa berbagai mesin penyimpanan - seperti MariaDB ColumnStore atau Facebook MyRocks - dapat diintegrasikan melalui arsitektur

diperluas milik MariaDB. Selain itu, memudahkan pelanggan untuk mengkonfigurasi MariaDB untuk mendukung berbagai macam kasus penggunaan dari OLTP ke OLAP.

2.13.5 Tipe Data

Berdasarkan Enterprise Jubilee (2015), Tipe data dibagi menjadi tiga kategori: numerik, tanggal, dan tipe string.

1. Tipe Data Numerik

Berikut ini daftar data numerik yang digunakan dalam MySQL:

- a. **INT** – nilai integer dengan ukuran normal. *Signed* berkisar -2147483648 sampai 2147483647. *Unsigned* berkisar dari 0 sampai 4294967295. *Unsigned* dimaksudkan, hanya untuk kisaran non minus.
- b. **TINYINT** – nilai integer yang sangat kecil. *Signed* berkisar dari -128 sampai 127. *Unsigned* berkisar dari 0 sampai 255.
- c. **SMALLINT** – nilai integer yang kecil. *Signed* berkisar dari -32768 sampai 32767. *Unsigned* berkisar dari 0 sampai 65535.
- d. **MEDIUMINT** – nilai integer medium. *Signed* berkisar dari -8388608 sampai 8388607. *Unsigned* berkisar 0 sampai 16777215.
- e. **BIGINT** – nilai integer dengan ukuran besar. *Signed* berkisar dari -9223372036854775808 sampai 9223372036854775807. *Unsigned* berkisar dari 0 sampai 18446744073709551615.
- f. **FLOAT** – berdasarkan Sutaji (2012), bilangan *floating point* yang kecil (presisi tunggal). Jangkauan nilai yang diizinkan adalah -3.402823466E+38 s/d -1.175494351e-38,0 dan 1.175494351E-38 s/d 3.402823466E+38.
- g. **DOUBLE** – berdasarkan Sutaji (2012), bilangan *floating point* 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.

2. Tipe Data Tanggal dan Waktu

Berikut ini daftar tipe data dan waktu dalam MySQL:

- a. **DATE** – tanggal dengan format YYYY-MM-DD diantara 1000-01-01 sampai 9999-12-31. Sebagai contoh, tanggal 30 Desember 1973 akan disimpan sebagai 1973-12-30.
 - b. **DATETIME** – kombinasi tanggal dan waktu dengan format YYYY-MM-DD HH:MM:SS, diantara 1000-01-01 00:00:00 sampai 9999-12-31 23:59:59. Sebagai contoh, pukul 15:30 sore pada tanggal 30 Desember 1973 akan disimpan sebagai 1973-12-30 15:30:00.
 - c. **TIME** – menyimpan nilai waktu dengan format HH:MM:SS. Contohnya, 15:30:00.
 - d. **YEAR** – menyimpan tahun dengan format 2 digit atau 4 digit. YEAR(2) Memiliki rentang nilai dari tahun 1970 sampai 2069 (70 sampai 69). YEAR(4) memiliki rentang nilai dari tahun 1901 sampai 2155.
3. Tipe Data String
- Berikut ini daftar tipe data string dalam MySQL:
- a. **CHAR** – string dengan panjang tetap antara 1 sampai 255 karakter.
 - b. **VARCHAR** – string dengan panjang bervariasi antara 1 sampai 255 karakter.
 - c. **BLOB/TEXT** – sebuah *field* dengan panjang maksimum 65535 karakter. BLOB (*Binary Large Objects*) digunakan untuk menyimpan data binary yang besar, seperti gambar atau tipe file lainnya. *Field* yang didefinisikan sebagai TEXT juga menyimpan jumlah data yang besar; perbedaannya, BLOB bersifat *case sensitive* yang artinya huruf besar dan kecil akan diartikan secara berbeda. Sebagai contoh penulisan alert () huruf kecil semua. Jika ditulis dengan ALERT () maka kode tidak bekerja. Sedangkan TEXT tidak.
 - d. **TINYBLOB/TINYTEXT** – sebuah kolom BLOB atau TEXT dengan panjang maksimum 255 karakter.
 - e. **MEDIUMBLOB/MEDIUMTEXT** – sebuah kolom BLOB atau TEXT dengan panjang maksimum 16777215 karakter.
 - f. **LONGLOB/LONGTEXT** – sebuah kolom BLOB atau TEXT dengan panjang maksimum 4294967295.

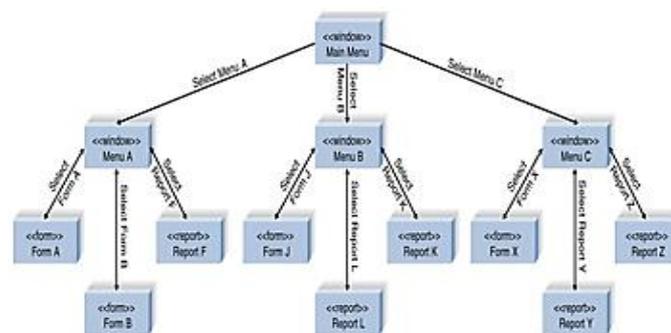
- g. **ENUM** – berdasarkan Sutaji (2012), enum adalah sebuah *enumeration*. Sebuah obyek string yang hanya boleh memiliki satu nilai, yang terambil dari ‘value1’, ‘value2’, ‘...’, NULL atau nilai spesial “”*error*. Sebuah *enum* dapat menampung 65535 pilihan nilai.

2.14 Windows Navigation Diagram (WND)

Windows Navigation Diagram merupakan *statechart* diagram khusus yang berfokus pada *user interface*. *Windows Navigation Diagram* (WND) menurut Dennis et al (2015) adalah sebagai berikut:

1. Menunjukkan bagaimana semua layar, formulir, dan laporan terkait.
2. Menunjukkan bagaimana pengguna bergerak dari satu ke yang lain.
3. Seperti diagram keadaan untuk *user interface*
 - a. Kotak mewakili komponen.
 - b. Panah mewakili transisi.
 - c. Stereotipe menunjukkan tipe antarmuka.

Contoh dari *Windows Navigation Diagram* dapat dilihat pada Gambar II.9.



Gambar II.8 Contoh *Windows Navigation Diagram*

(Sumber: Dennis et al, 2015)

2.15 PHP (*Hypertext Preprocessor*)

PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1994. PHP (PHP: Hypertext Preprocessor) adalah bahasa *server-sidescripting* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis. Karena PHP merupakan *server-side scripting* maka sintaks dan perintah perintah PHP akan dieksekusi di server kemudian hasilnya dikirimkan ke browser dalam format HTML. Dengan demikian

kode program yang ditulis dalam PHP tidak akan terlihat oleh *user* sehingga keamanan dalam *web* lebih terjamin (Sutaji, 2012). Untuk menjalankan sistem PHP dibutuhkan 3 komponen, diantaranya adalah sebagai berikut (Sutaji, 2012):

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

Adapun kelebihan-kelebihan dari bahasa PHP diantaranya adalah sebagai berikut:

1. PHP mudah dibuat dan kecepatan akses tinggi.
2. PHP dapat berjalan dalam *webservers* yang berbeda dalam sistem operasi yang berbeda pula.
3. PHP diterbitkan secara gratis.
4. PHP merupakan bahasa yang dapat diletakkan dalam tag HTML.
5. Sistem *database* yang didukung PHP cukup banyak.
6. PHP termasuk *server side programming*.

2.16 Framework

Berdasarkan Supono dan Virdiandry Putratama (2016), *framework* secara sederhana dapat diartikan kumpulan dari fungsi-fungsi atau prosedur dan *class-class* untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang pemrograman, tanpa harus membuat fungsi atau *class* dari awal. Seorang *programmer* tidak perlu membuat dari awal fungsi-fungsi seperti fungsi koneksi ke *database*, fungsi *string*, dan lainnya.

Berdasarkan Wikipedia Indonesia, PHP *framework* adalah sebagai sebuah kerangka kerja yang disusun oleh berbagai komunitas pengembang web diseluruh dunia. Kerangka ini bertujuan untuk mempermudah dalam membuat sebuah aplikasi web bagi yang sering menulis *script* PHP secara keseluruhan dan itu pun akan diulang pada halaman yang lain. *Framework* secara sederhana dapat diartikan kumpulan dari fungsi-fungsi atau prosedur-prosedur dari *class-class* untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan

mempercepat pekerjaan seorang *programmer*, tanpa harus membuat fungsi atau *class* dari awal. Contoh *framework* PHP seperti *CakePHP*, *Codeigniter*, *Laravel*, *PRADO*, *Symfony*, *Zend Framework*, *Yii*, *Akelos*, *QPHP*, *ZooP*. Ada beberapa alasan mengapa menggunakan *framework* yaitu:

- a. Mempercepat dan mempermudah pembangunan sebuah aplikasi web.
- b. Relatif memudahkan dalam proses *maintenance* karena sudah ada pola tertentu dalam sebuah *framework* (dengan syarat *programmer* mengikuti pola standar yang ada)
- c. Umumnya *framework* menyediakan fasilitas-fasilitas umum yang dipakai sehingga *programmer* tidak perlu membangun dari awal (misalnya validasi, ORM, pagination, *multiple database*, *scaffolding*, pengaturan *session*, *error handling*, dan lain-lain).

Berdasarkan Jubilee (2015), *framework* merupakan banyak kode, yang disimpan dalam beberapa file yang terpisah, dan memudahkan dalam penggunaan kode yang digunakan secara berulang-ulang. Dengan *framework*, *programmer* tidak perlu menulis baris kode yang panjang untuk fungsi tertentu. *Programmer* hanya perlu menggunakan fungsi yang sudah dituliskan ke dalam *framework*.

2.17 Codeigniter

Berdasarkan Supono dan Putratama (2016), *codeigniter* adalah aplikasi *open source* berupa *framework* dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan PHP. Ada 3 jenis komponen yang membangun suatu MVC *pattern* dalam suatu aplikasi yaitu:

1. *View*, merupakan bagian yang menangani *presentation logic*. *View* berfungsi untuk menerima dan mempresentasikan data kepada *user*. Bagian ini tidak memiliki akses langsung terhadap bagian *model*.
2. *Model*, biasanya berhubungan langsung dengan *database* untuk memanipulasi data (*insert, update, delete, search*) menangani validasi dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.

Controller, merupakan bagian yang mengatur hubungan antara bagian *model* dan bagian *view*, *controller* berfungsi untuk menerima *request* dan data dari *user* kemudian menentukan apa yang diproses oleh aplikasi.

Berdasarkan Jubilee (2015), CI merupakan *framework* yang dibuat oleh Rick Ellis, CEO pMachine. CI dibuat dan terus dikembangkan, dan dapat digunakan secara gratis. CI juga mempunyai forum yang dapat digunakan untuk berbagi tips, mencari, dan mengatasi *bug* dalam kode. Tersedia pada situs www.codeigniter.com.

Berdasarkan Komputer (2014), *codeigniter* adalah *framework* bahasa pemrograman PHP. Sebagai contoh, situs kompas.com dan okezone.com mengadopsi sistem ini sebagai *framework* websitenya. *Codeigniter* menawarkan kemudahan serta standarisasi dalam proses pengembangan aplikasi berbasis website. Dengan *codeigniter*, proses pengembangan menjadi lebih cepat dan terstandar. Standard ini sangat berguna dalam pengembangan yang dikerjakan oleh tim. *Codeigniter* juga menyediakan *library*, dan *helper* yang berguna dan mempermudah proses. Terdapat beberapa kelebihan penggunaan *codeigniter* dalam pengembangan suatu proyek, yaitu:

a. Menghemat Waktu

Dengan struktur dan *library* yang telah disediakan, membuat *programmer* hanya fokus pada logika pemrograman yang sedang dikerjakan.

b. *Code Reuse*

Dengan *codeigniter*, suatu pekerjaan akan memiliki standard yang baku. Sehingga *programmer* dapat menggunakan kembali pada proyek-proyek yang akan datang.

c. Dukungan Komunitas

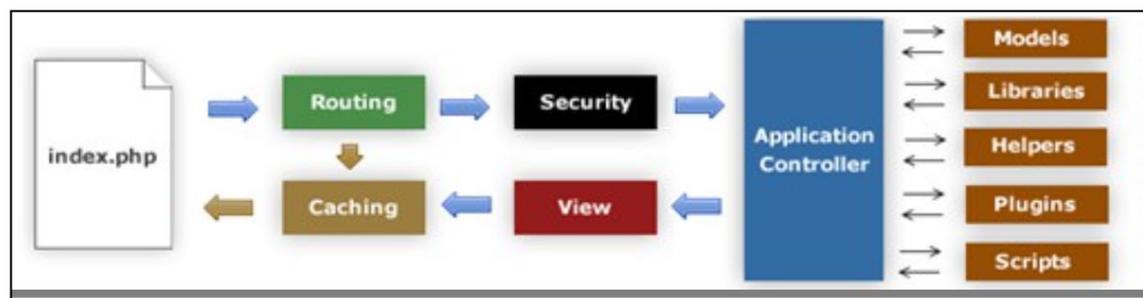
CI mempunyai banyak komunitas yang tersebar untuk membantu masalah *programmer*. Dengan adanya komunitas, *programmer* dapat saling berkonsultasi mengenai masalah yang dihadapi.

Berdasarkan Wikipedia Indonesia, *codeigniter* merupakan aplikasi sumber terbuka yang berupa *framework* PHP dengan model MVC (*Model*, *View*, *Controller*) untuk membangun website dinamis dengan menggunakan PHP.

Codeigniter memudahkan *developer* untuk membuat aplikasi web dengan cepat dan mudah dibandingkan dengan membuatnya dari awal. *Codeigniter* pertama kali dirilis pada 28 Februari 2006. Model *view controller* merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, berawal pada bahasa pemrograman *small talk*, MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, *user interface*, dan bagian yang menjadi kontrol aplikasi. 3 jenis komponen MVC adalah:

1. *View*, bagian ini biasanya berupa file *template* HTML, yang diatur oleh *controller*. *View* berfungsi untuk menerima dan merepresentasikan data kepada *user*. Bagian ini tidak memiliki akses langsung terhadap bagian model.
2. *Model*, biasanya berhubungan langsung dengan *database* untuk memanipulasi data (*insert, update, delete, search*), menangani validasi dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.
3. *Controller*, merupakan bagian yang mengatur hubungan antara bagian *model* dan bagian *view*, *controller* berfungsi untuk menerima *request* dan data dari *user* kemudian menentukan apa yang akan diproses oleh aplikasi.

Cara kerja *codeigniter* berdasarkan situs internet Ipteknet dapat dilihat pada Gambar II.10.



Gambar II.9 *CodeIgniter*

(Sumber: Situs Internet, 2018)

Berikut ini adalah penjelasan cara kerja *codeigniter*:

1. *Index.php* bertindak sebagai *controller* terdepan, dan menganalisis *resource* yang diperlukan untuk menjalankan *codeigniter*.
2. *Router* memeriksa HTTP *request* untuk menentukan apa yang harus dikerjakan.

3. Jika file *cache file* ada, maka akan ditampilkan langsung, dengan melewati eksekusi normal sistem.
4. Sebelum memuat *controller*, *HTTP request* akan memeriksa apa yang disubmit *user* dan mem-*filter*nya untuk keamanan.
5. *Controller* membuat *model*, *core libraries*, *plugin*, *helper*, dan *resource* lainnya untuk memproses permintaan tertentu.
6. *View* ditampilkan di browser sesuai proses yang dikerjakan *controller*. Jika *caching* dijalankan, *view* akan *dicache* terlebih dahulu agar dapat ditampilkan di *request* selanjutnya.

2.18 *Black Box Testing*

Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black box* memungkinkan merekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black box* bukan merupakan alternatif dari teknik *white box*, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan dari pada metode *white box* (Pressman dalam Subri, 2011). Pengujian *black box* berusaha menemukan kesalahan dengan kategori sebagai berikut:

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau akses *database eksternal*
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi

Tidak seperti *white box*, yang dilakukan pada saat awal proses pengujian, pengujian *black box* cenderung diaplikasikan selama tahap akhir pengujian. Karena pengujian *black box* memperhatikan struktur kontrol, maka perhatian berfokus pada domain informasi (Pressman, 2011).

BAB III

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metodologi penelitian adalah tata cara bagaimana suatu penelitian dilaksanakan atau cara yang ditempuh sehubungan dengan penelitian yang dilakukan, 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. Berikut ini adalah pengertian dari metodologi penelitian menurut para ahli:

- a. Menurut KBBI (Kamus Besar Bahasa Indonesia), metode merupakan cara kerja yang mempunyai sistem dalam memudahkan pelaksanaan dari suatu kegiatan untuk mencapai sebuah tujuan tertentu.
- b. Menurut Sugiyono, metode penelitian adalah cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu.

Dari beberapa pengertian dan definisi yang tersebut diatas maka dapat diperoleh kesimpulan pengertian metodologi penelitian adalah ilmu atau cara yang digunakan dalam memperoleh suatu kebenaran dengan menggunakan penelusuran dengan urutan dan tata cara tertentu sesuai dengan apa yang akan dikaji atau yang diteliti secara ilmiah.

3.2 Jenis dan Sumber Data

Data yang diperlukan dalam penelitian ini ada dua jenis, yaitu data primer dan data sekunder. Sumber dari data-data ini berasal dari tempat yang diamati pada praktek kerja lapangan di PT Baneex Indonesia:

1. Data Primer

Data Primer adalah data yang diperoleh secara langsung dari PT Baneex Indonesia, dimana pengumpulan data atau informasi dilakukan langsung dari objek yang diteliti. Data-data tersebut adalah data yang digunakan dalam

proses penyimpanan barang jadi diantaranya, analisis dokumen yang berjalan, proses bisnis sistem saat ini dan yang akan diusulkan, dan kebutuhan pengguna sistem.

2. Data Sekunder

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

3.3 Metode Pengumpulan Data

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

a. Studi Lapangan

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

1. Pengamatan, tahap ini dilakukan secara langsung pada di divisi *Production Planning and Control* di PT Baneex Indonesia dengan mengamati sistem penjadwalan produksi, dimulai dari penerimaan *forecasting order* dari *costumer*, sampai melakukan laporan produksi bulanan. Melalui teknik ini, data yang dibutuhkan diamati, dikumpulkan lalu diolah sebagai bahan di dalam penelitian.
2. Wawancara, yaitu mencari data yang dibutuhkan secara langsung melalui memberikan pertanyaan yang ingin ditanyakan terhadap segala hal yang diperlukan pada penyusunan Tugas Akhir ini. Wawancara ini dilakukan kepada staf yang bekerja di divisi *Production Planning and Control* di PT Baneex Indonesia.

b. Studi Kepustakaan

Studi kepustakaan dilakukan dengan membaca buku dan literatur dalam lingkup perkuliahan maupun di luar lingkup perkuliahan yang berhubungan dengan judul dan permasalahan, sehingga dapat menunjang dalam penulisan tugas akhir ini. Studi kepustakaan yang dilakukan adalah dengan menggunakan buku yang dimiliki, buku yang dipinjam dari perpustakaan dan mencari data yang diperlukan melalui internet.

3.4 Metode Pengembangan Sistem

Metode pengembangan sistem informasi penyimpanan barang jadi ini menggunakan metode pengembangan sistem *waterfall*. Metode *waterfall* ini terdiri dari tahap *planning*, *analysis*, *design*, dan *implementation*. Berikut ini akan dijelaskan secara singkat mengenai tahapan-tahapan yang dilakukan dalam pengembangan sistem dengan metode *waterfall*.

1. Tahap Perencanaan (*Planning*)

Pada tahap ini akan dilakukan perencanaan untuk membangun suatu sistem informasi dengan membuat sebuah *system request*.

2. Tahap Analisis (*Analysis*)

Menganalisis kebutuhan sistem dengan wawancara, observasi, dan membuat analisis permasalahan yang didapat dari tahap identifikasi masalah sebagai bahan pengembangan sistem.

3. Tahap Desain (*Design*)

Membuat pemodelan sistem dengan *Unified Modeling Language* (UML) yaitu menggunakan *usecase* diagram, *activity* diagram, *sequence* diagram dan *class* diagram dan membuat rancangan antarmuka. Data-data yang didapat dari tahap analisis diterapkan dalam tahap desain.

4. Tahap Implementasi (*Implementation*)

Pada tahap ini, pengembang akan membangun sebuah sistem dengan rancangan yang sudah ada diterjemahkan ke dalam sebuah *code* atau aktivitas *coding*. Bahasa pemrograman yang digunakan adalah PHP dengan

menggunakan *framework CodeIgniter* dan *MariaDB* sebagai *database* yang digunakan.

3.5 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

Studi Pendahuluan merupakan langkah awal yang dilakukan untuk memulai penelitian. Studi pendahuluan dilakukan dengan turun langsung ke lokasi untuk mengetahui gambaran yang jelas mengenai sistem yang sedang berjalan di divisi *Production Planning and Control*. Studi dapat dilakukan dengan melakukan observasi, wawancara, dan studi pustaka.

2. Identifikasi Masalah

Pada tahap ini dilakukan cara dalam mengidentifikasi suatu masalah yang terjadi pada divisi *Production Planning and Control* yang berfokus pada penerimaan dan penyimpanan bahan baku, dalam mengidentifikasi suatu masalah dilakukan beberapa cara seperti:

- a. Analisis terhadap hasil dalam bentuk transkrip wawancara dan hasil observasi yang dilakukan pada pihak-pihak yang terkait dalam proses penjadwalan produksi.
- b. Analisis dokumen masuk dan keluar yang terkait pada proses penjadwalan produksi.

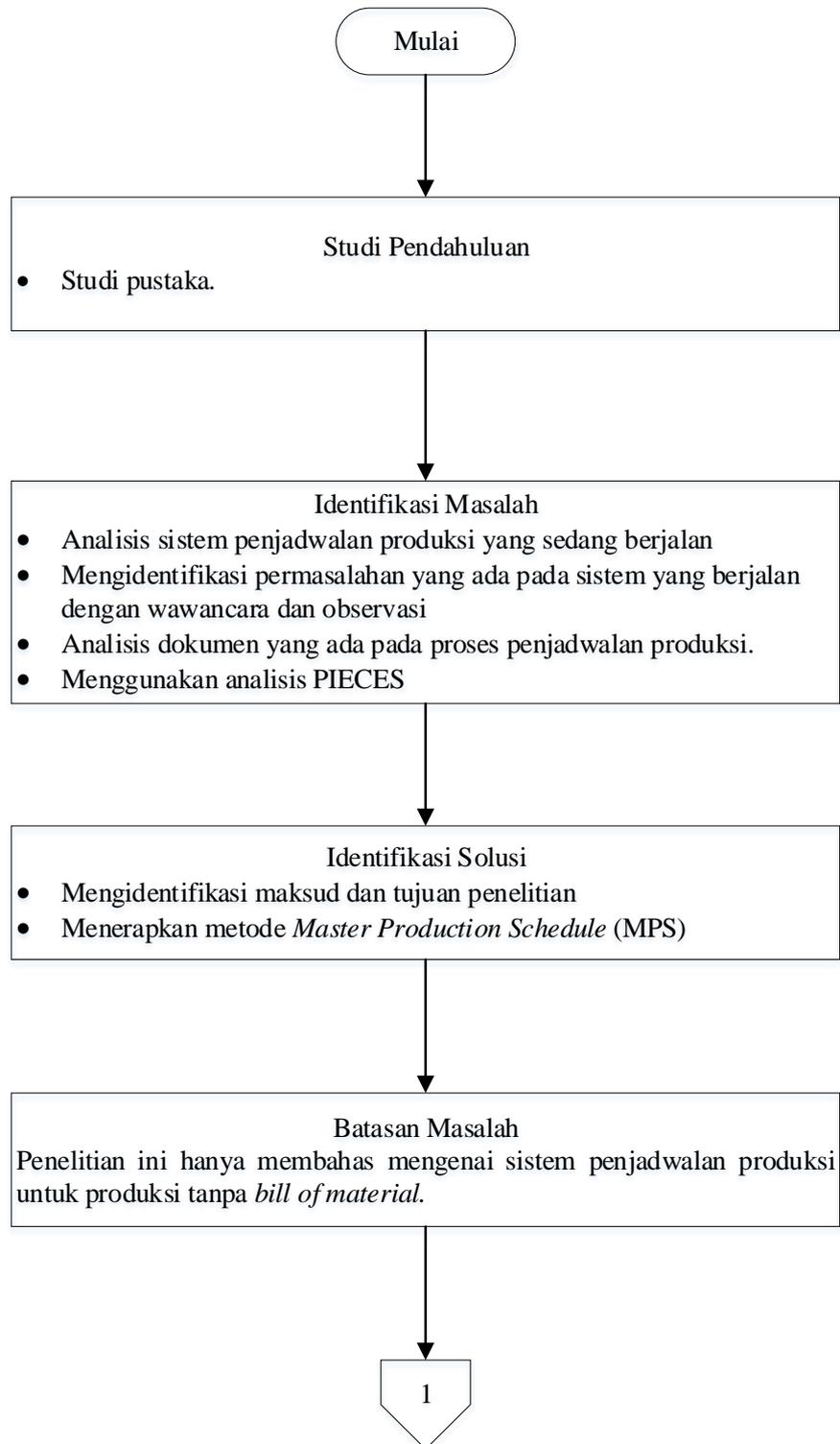
3. Identifikasi Solusi

Setelah mengidentifikasi suatu masalah, maka penulis melakukan pemecahan suatu masalah berdasarkan data-data yang telah di dapat pada tahap sebelumnya. Terdapat beberapa pemecahan masalah yang penulis lakukan seperti:

- a. Membangun sistem informasi yang terintegrasi antara divisi Produksi dengan PPC

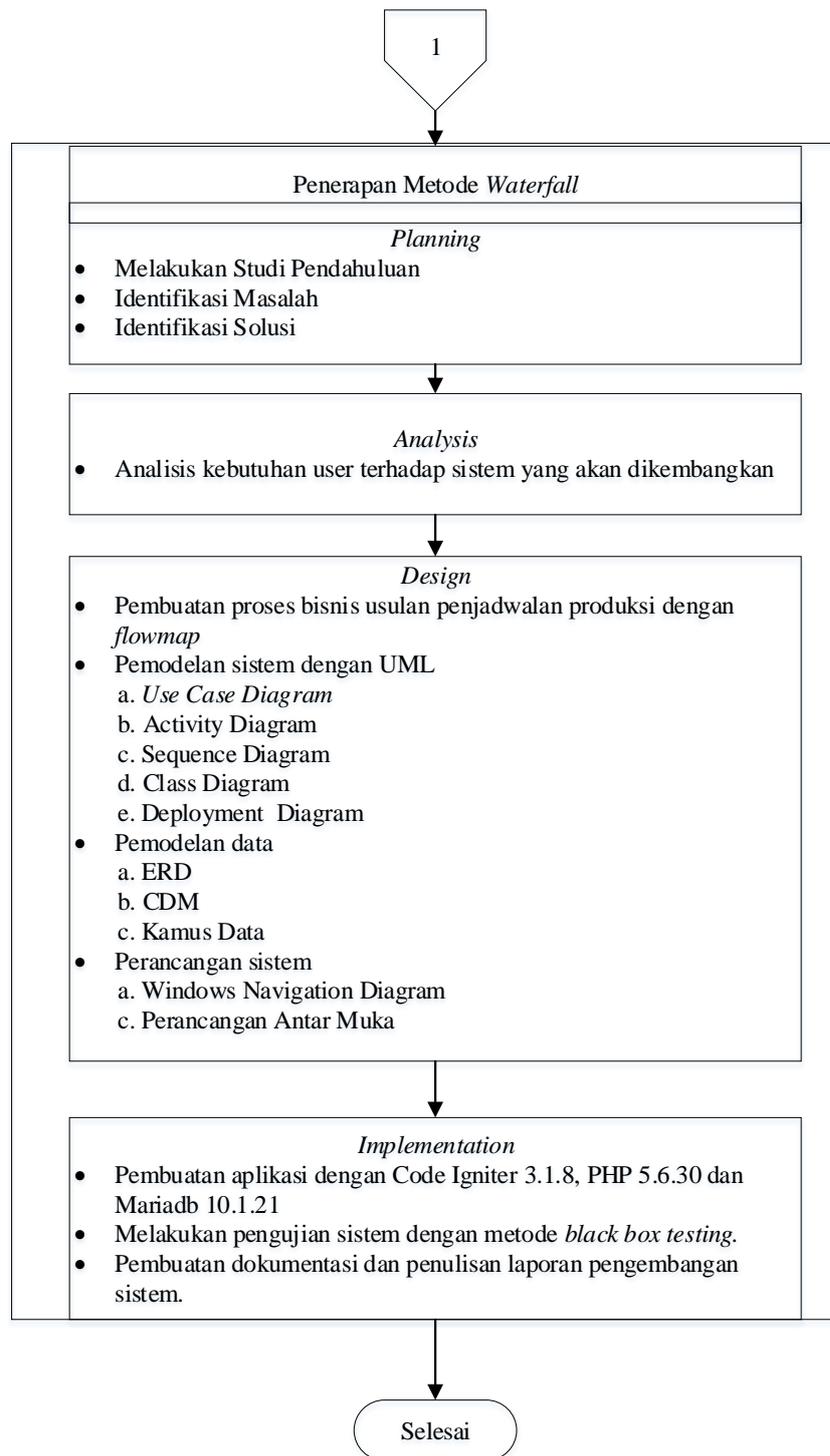
- b. Menerapkan metode *Master Production Schedule* (MPS) untuk menjadwalkan produksi sesuai dengan teknik yang ada dalam penjadwalan produksi
 - c. Mengembangkan sistem dengan menggunakan metode *waterfall*, dengan alasan data dokumen, data kebutuhan *user* hingga batasan waktu pengerjaan sistem
4. Penerapan Metode *Waterfall*
- a. Perencanaan (*Planning*)
Penulis melakukan perencanaan dengan beberapa tahap yaitu melakukan studi pendahuluan, identifikasi masalah dan identifikasi solusi.
 - b. Melakukan Analisis (*Analysis*)
Menganalisis kebutuhan sistem dengan wawancara, observasi, dan membuat analisis permasalahan yang didapat dari tahap identifikasi masalah sebagai bahan pengembangan sistem.
 - c. Membuat Desain (*Design*)
Membuat perancangan sistem dengan menggunakan *Unified Modeling Language* (UML) diantaranya *usecase* diagram, *activity* diagram, *sequence* diagram dan *class* diagram dan membuat rancangan antarmuka. Data-data yang didapat dari tahap analisis diterapkan dalam tahap desain.
 - d. Implementasi (*Implementation*)
Pada tahap ini, pengembang akan membangun sebuah sistem dengan rancangan yang sudah ada diterjemahkan ke dalam sebuah *code* atau aktivitas *coding*. Bahasa pemrograman yang digunakan adalah PHP dengan menggunakan *framework CodeIgniter* dan *MariaDB* sebagai *database* yang digunakan.

Berikut merupakan diagram alir kerangka penelitian yang digunakan dalam laporan tugas akhir ini dapat dilihat pada Gambar III.1.



Gambar III.1 Kerangka Penelitian

(Sumber: Pengolahan Data, 2018)



Gambar III.1 Kerangka Penelitian (Lanjutan)

(Sumber: Pengolahan Data, 2018)

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Tentang Perusahaan

PT Baneex Indonesia adalah perusahaan yang bergerak di bidang Manufaktur Otomotif yang memproduksi komponen *Spring*, *Bending* dan *Stamping*. PT Baneex Indonesia didirikan pada bulan Januari 2007 yang berlokasi di Jalan Express Blok WW No 6 RT 002 RW 024 Bojong Rawa Lumbu, Bekasi. Perusahaan ini sebagai produsen komponen Otomotif dan Elektronik mempunyai komitmen untuk selalu memberikan produk dan layanan terbaik yang mampu memenuhi dan menterjemahkan keinginan pelanggan. Komitmen ini dipegang oleh seluruh jajaran manajemen, mulai dari *Top Management*, *Middle Management*, dan seluruh karyawan yang bekerja di PT Baneex Indonesia.

Agar dapat memenuhi komitmen tersebut, maka dibuat suatu program pedoman sistem manajemen mutu yang sistematis dan terkendali untuk semua aktivitas, operasi kerja dan fungsi-fungsi yang berpengaruh terhadap mutu produk dan layanan. Program pedoman sistem manajemen mutu ini diuraikan didalam manual mutu yang mengacu kepada ISO 9001:2000 dan sebagai panduan dan penerapannya, maka perusahaan mencanangkan kebijakan mutu sebagai berikut:

“Kepuasan pelanggan adalah tujuan yang paling utama di dalam tanggung jawab sosial perusahaan dan merupakan dasar untuk mengembangkan perusahaan”. Untuk mewujudkan kebijakan mutu ini, dibuat pedoman aktivitas untuk seluruh karyawan PT Baneex Indonesia sebagai berikut:

1. Meningkatkan Efektifitas Kerja Sistem Mutu
2. Meningkatkan Produktivitas yang Berkualitas
3. Meningkatkan Kepuasan dan Kepercayaan Pelanggan

Sasaran mutu PT Baneex Indonesia ditetapkan secara berkala sesuai dengan target manajemen serta selaras dengan kebijakan mutu perusahaan. Pencapaian kebijakan mutu menjadi tanggung jawab seluruh jajaran manajemen dan karyawan PT Baneex Indonesia.



Gambar IV.1 Logo PT Baneex Indonesia
(Sumber: PT Baneex Indonesia, 2018)

4.2 Profil Perusahaan

PT Baneex Indonesia merupakan salah satu perusahaan yang bergerak dalam bidang Manufaktur Otomotif yang memproduksi komponen untuk kendaraan roda empat khususnya bagian jok mobil yang memproduksi komponen *Spring*, *Bending* dan *Stamping*, berdiri sejak 2007 oleh Bapak Eka Mulyana, Bapak Triono Sudarno, dan Bapak Wiji, Ibu Laura S.

Nama Perusahaan	: PT Baneex Indonesia
Alamat <i>Head Office</i>	: Jalan Express Blok WW No 6 RT 002 RW 024 Bojong Rawa Lumbu – Rawa Lumbu Bekasi
Alamat Pabrik 1	: Jalan Toyogiri RT 002 RW 03 – Kampung Legon Kelurahan Jatimulya Kecamatan Tambun Selatan Kabupaten Bekasi
Alamat Pabrik 2	: Jalan Raya Setu Kampung Burangkeng RT 007 RW 009 Desa Cileduk Kecamatan Setu Kabupaten Bekasi
Tahun Berdiri	: 2007
Produk	: - <i>Spring</i> - <i>Bending</i> - <i>Stamping</i>

4.3 Visi dan Misi Perusahaan

PT Baneex Indonesia merupakan bisnis usaha yang yang memproduksi komponen *Spring*, *Bending* dan *Stamping*. Guna mendukung pencapaian permintaan produksi dan produk baru, PT Baneex Indonesia mempunyai visi dan misi sebagai berikut:

Visi:

“Menjadi mitra utama Perusahaan Otomotif dan elektronik yang kompetitif dan handal dalam mutu dan layanan.”

Misi:

1. *Innovations*

Selalu melakukan inovasi secara berkesinambungan baik proses teknik maupun kinerja karyawan.

2. Jaminan Mutu

Menetapkan kerangka kerja proses peningkatan mutu yang sistematis dan terorganisir dengan membakukan proses guna memastikan konsistensi dan mampu telusur serta meningkatkan hubungan antar fungsi yang dapat mempengaruhi mutu.

3. Kepuasan pelanggan

Selalu memberikan produk dan layanan terbaik bagi pelanggan.

4. Kompetitif

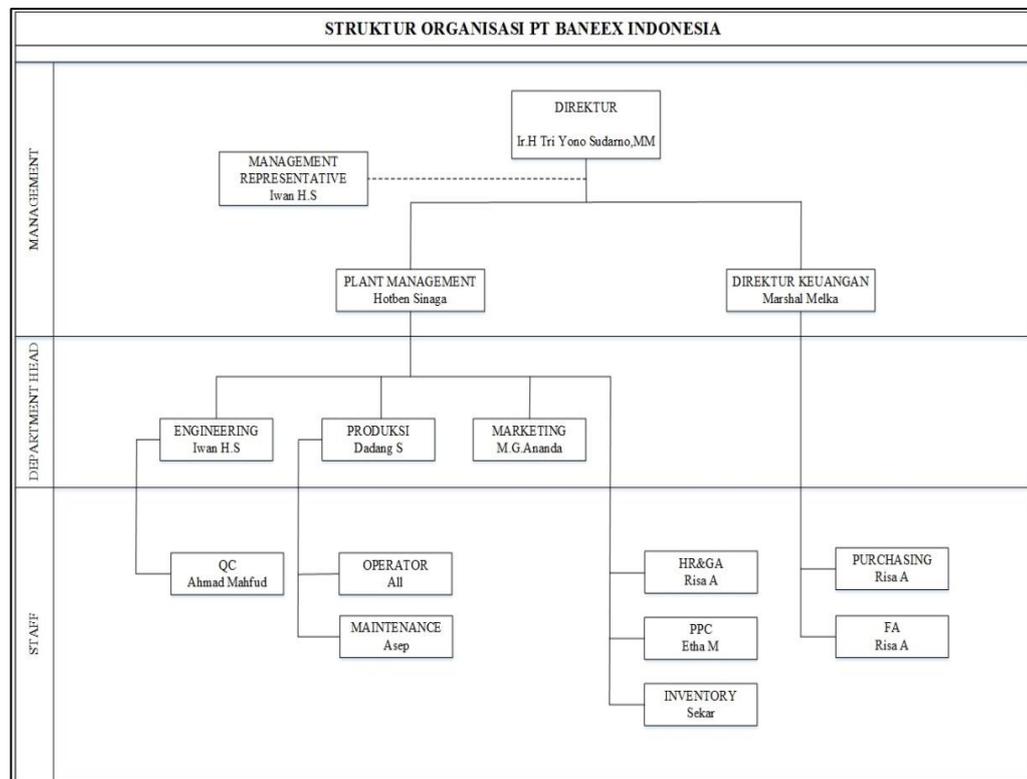
Meningkatkan produktivitas dan mutu produk serta layanan dengan memenuhi persyaratan pelanggan melalui kerjasama dan komunikasi yang baik, pengendalian bisnis dan penurunan pemborosan yang diakibatkan dari mutu kerja yang buruk.

4.4 Struktur Organisasi Perusahaan

Struktur Organisasi adalah bagaimana pekerjaan dibagi, dikelompokkan dan dikoordinasikan secara formal (Wikipedia Indonesia). Pengertian lain dari struktur organisasi yang dikemukakan oleh beberapa ahli sebagai berikut:

1. Struktur organisasi dapat diartikan sebagai kerangka formal organisasi yang dengan kerangka kerja itu tugas-tugas pekerja dibagi-bagi, dikelompokkan dan dikoordinasikan (*Robbins dan Coulter, 2007:284*).
2. Struktur organisasi didefinisikan sebagai mekanisme-mekanisme formal dengan mana organisasi dikelola (*Handoko, 2003:169*).
3. Struktur organisasi adalah pola formal mengelompokkan orang dan pekerjaan (*Gibson dkk, 2002:9*).

PT Baneex Indonesia memiliki struktur organisasi, agar terorganisasi jalannya kegiatan dalam perusahaan. Struktur Organisasi PT Baneex Indonesia dapat dilihat pada Gambar IV.2.



Gambar IV.2 Struktur Organisasi PT Baneex Indonesia

(Sumber: PT Baneex Indonesia, 2018)

Untuk lebih jelasnya berikut merupakan penjelasan atau fungsi dari struktur organisasi PT Baneex Indonesia:

1. Direktur

- a. Tugas dan Tanggung Jawab
 - Menjalankan bisnis perusahaan.

- Memimpin seluruh karyawan dalam menjalankan bisnis perusahaan.
- Menetapkan kebijakan-kebijakan perusahaan.
- Menetapkan dan merumuskan strategi bisnis perusahaan.
- Memilih staf–staf yang membantu kinerja perusahaan.
- Menyetujui anggaran tahunan perusahaan.
- Menyampaikan laporan kepada pemegang saham.
- Meningkatkan performa perusahaan.

b. **Wewenang**

- Mengkoordinasikan dan mengendalikan kegiatan-kegiatan dibidang administrasi keuangan, kepegaiwaan, dan kesekretariatan.
- Mengkoordinasi dan mengendalikan kegiatan pengadaan dan peralatan perubahan.
- Merencanakan dan mengembangkan sumber-sumber pendapatan serta pembelanjaan dan kekayaan perusahaan.
- Memimpin seluruh dewan atau komite eksekutif.
- Menawarkan visi dan imajinasi tingkat tinggi.
- Memimpin rapat umum perusahaan dalam hal meningkatkan perusahaan.
- Bertindak sebagai perwakilan organisasi dalam hal hubungan dengan luar.
- Mengambil keputusan sebagaimana didelegasikan dalam situasi tertentu dianggap perlu, yang diputuskan dalam pertemuan-pertemuan.
- Menetapkan peraturan perusahaan, merencanakan, menetapkan sistem operasional perusahaan, menetapkan strategi pencapaian visi dan misi perusahaan.
- Menetapkan kebijakan-kebijakan tentang ketentuan pelaksanaan operasional perusahaan dengan pembagian tugas yang jelas.

2. *Management Representative*

Bertanggung jawab atas koordinasi dengan pihak-pihak di luar perusahaan dalam kaitannya dengan sistem manajemen mutu perusahaan.

3. *Plant Manager*

a. Tugas dan Tanggung Jawab

- Mengontrol kinerja bawahan.
- Bertanggung jawab atas keseluruhan kegiatan di lingkup kerja perusahaan.
- Mengontrol bisnis *plan* yang telah dibuat terhadap kondisi real yang ada di lapangan.
- Secara berkala melakukan pertemuan guna melakukan peninjauan ulang terhadap semua kegiatan yang telah dan sedang berjalan.
- Memeriksa pencapaian program terhadap persoalan yang dihadapi serta memberikan ide-ide perbaikan.
- Memeriksa pelaksanaan kegiatan di lapangan secara langsung pelaksanaan kegiatan di lapangan.
- Membuat laporan kinerja perusahaan yang akan dilaporkan kepada atasan.
- Mengikuti aturan yang sudah ditetapkan oleh atasan atau perusahaan.

b. Wewenang

- Menegur bawahan jika terjadi ketidaksesuaian dengan aturan yang ditetapkan di perusahaan.
- Mengatur dalam pengoperasian suatu kegiatan guna mencapai tujuan yang sudah ditetapkan.
- Mengambil keputusan yang berhubungan dengan kelancaran dalam perusahaan untuk tujuan perusahaan.
- Member masukan atau ide-ide yang dapat memajukan kinerja dalam perusahaan.

4. *Direktur Keuangan*

a. Tugas dan Tanggung Jawab

- Membuat draft Rencana Anggaran dan Belanja Perusahaan serta membuat rencana pengeluaran kas untuk kegiatan operasional perusahaan minggu berikutnya.

- Membuat jadwal pembayaran hutang kepada penagih piutang perusahaan.
- Melakukan pengawasan terhadap transaksi keuangan yang berupa kas masuk dan kas keluar untuk menjamin tersedianya minimal kas yang likuid.
- Membuat laporan Realisasi Anggaran, Belanja Perusahaan, membuat laporan mengenai posisi kas, hutang dan piutang perusahaan.

b. Wewenang

Mengkses data keuangan terkait perusahaan.

5. *Engineering*

a. Tugas dan Tanggung Jawab

- Pengambil keputusan tertinggi dalam internal bagian *engineering* dan *QC*.
- Senantiasa mengawasi serta mengadakan peninjauan kembali terhadap prosedur dan tatacara pelaksanaan tugas para bawahan, terutama mengenai tingkat keefektifan sistem kerja di bagian *engineering* dan *QC*.
- Mengadakan evaluasi terhadap mesin-mesin dan perlengkapan pendukungnya guna peningkatan efisiensi penggunaannya.
- Membuat jadwal '*overhaul*' mesin secara periodik dan mengkoordinir pelaksanaannya.
- Mengadakan evaluasi terhadap prestasi kerja para bawahan di bagian *engineering* dan *QC* secara periodik sesuai standar yang telah ditentukan.
- Mencegah tindakan-tindakan yang dapat merusak atau menghambat kelancaran/keutuhan sistem kerja di bagian *engineering* dan *QC*, baik langsung maupun tidak langsung.
- Mengadakan hubungan kerja yang baik dengan bawahan hingga memungkinkan bagian *engineering* dan *QC* mencapai tujuan.
- Mengadakan hubungan kerja yang baik dengan bagian lain untuk kelancaran kerja di perusahaan.

- Melaksanakan tugas yang diberikan oleh atasan.

b. Wewenang

- Memberikan saran-saran perbaikan kepada divisi lain guna peningkatan produktifitas dan kualitas perusahaan.
- Menghentikan proses produksi jika mengetahui adanya ketidaknormalan mesin atau perlengkapan pendukungnya atau karena kualitas hasil produksi yang tidak baik.
- Menegur secara langsung bawahan dari bagian lain, jika diketahui melakukan penyimpangan kerja.

6. Produksi

a. Tugas dan Tanggung Jawab

- Pengambilan keputusan tertinggi dalam *internal* bagian produksi.
- Memimpin dan mengkoordinir para bawahan untuk seluruh tugas operasional.
- Membuat jadwal produksi sejalan dengan jadwal produksi yang diterima dari bagian PPC
- Memeriksa ketersediaan bahan baku yang akan dipakai sesuai jadwal dan menganalisis jika terjadi ketidak sesuaian
- Memeriksa laporan, berita acara dan catatan yang dibuat bawahannya dan menganalisis jika terjadi penyimpangan
- Mengkoordinasi antar *shift* agar terjalin komunikasi yang baik dan informasi-informasi yang dibutuhkan sampai pada sasaran.
- Memantau dan mengadakan evaluasi mesin, sistem kerja, administrasi dan para bawahannya secara periodik sesuai stantar yang telah ditentukan, serta secara langsung mengatasi masalah-masalah yang timbul.
- Mengusahakan peningkatan, keahlian dan keterampilan bawahannya dengan memberikan pengarahan mengenai cara-cara yang lebih praktis dan efisien.
- Membina, memelihara dan menjaga disiplin kerja bawahannya agar tercapai efisiensi dan produktivitas kerja yang tinggi.

- Senantiasa mengawasi serta mengadakan peninjauan kembali terhadap prosedur dan tatacara pelaksanaan tugas para bawahan, terutama mengenai tingkat efisiensi dan efektifitas sistem kerja.
 - Mencegah tindakan-tindakan yang dapat merusak atau menghambat kelancaran/keutuhan sistem kerja, baik langsung maupun tidak langsung.
 - Mengadakan hubungan kerja yang baik dengan bawahan hingga mencapai tujuan yang telah ditetapkan.
 - Mengadakan hubungan kerja yang baik dengan semua bagian untuk kelancaran kerja perusahaan.
 - Melaksanakan tugas lain yang diberikan oleh atasan.
- b. Wewenang
- Memberikan saran-saran perbaikan kepada divisi lain guna peningkatan produktivitas perusahaan.
 - Menegur secara langsung bawahan dari bagian lain, jika diketahui melakukan penyimpangan kerja.

7. Marketing

Bertugas untuk memasarkan produk, guna meningkatkan penjualan perusahaan dan juga mengelola data *customer* serta mengelola data *purchase order* dari *customer*.

8. Quality Control

- a. Tugas dan Tanggung Jawab
- Melakukan pekerjaan sesuai standar kerja yang telah ditentukan
 - Memeriksa material yang datang dari *supplier*.
 - Memeriksa hasil produksi selama proses produksi secara periodik dan sebelum dikirim ke gudang jadi secara acak, juga barang pengembalian dari *customer*.
 - Membuat laporan tertulis terhadap hasil pemeriksaan.
 - Melaksanakan pengarsipan semua data yang berhubungan dengan bagian *Quality*.

- Melapor kepada atasan jika menemukan suatu proses atau barang menyimpang dari spesifikasi.
- Menjaga ketertiban dan memperhatikan keselamatan kerja selama bekerja.
- Menjaga kebersihan alat-alat yang digunakan dan lingkungan kerja.
- Senantiasa bekerjasama dengan petugas *Quality Control* yang lain.
- Melaksanakan tugas lain yang diberikan oleh atasan.

b. Wewenang

- Memberikan saran-saran perbaikan kepada divisi lain guna peningkatan produktivitas perusahaan.
- Menghentikan atau merubah produksi yang sedang berjalan sesuai informasi terbaru dari *marketing*.
- Menegur secara langsung bawahan dari bagian lain, jika diketahui melakukan penyimpangan kerja.

9. *Maintenance*

Bertugas untuk perbaikan dan perawatan mesin sebagai penunjang aktivitas produksi.

10. Operator Produksi

a. Tugas dan Tanggung Jawab

- Melakukan pekerjaan sesuai standar kerja yang telah ditentukan.
- Melapor kepada atasan jika menemukan sesuatu yang menyimpang.
- Menjaga, memeriksa serta merawat mesin dan peralatan pendukung yang digunakan agar selalu dalam kondisi baik, bersih dan siap pakai.
- Senantiasa bekerjasama dan menjalin komunikasi yang baik dengan *QC Inspector*.
- Mengonfirmasikan ke atasan jika proses produksi tidak sesuai dengan spesifikasi produk yang telah ditentukan.
- Menjaga ketertiban dan memperlihatkan keselamatan kerja selama bekerja.
- Menjaga kebersihan alat-alat yang digunakan dan lingkungan kerja.
- Melaksanakan tugas lain yang telah diberikan oleh atasan.

b. Wewenang

Memberikan saran-saran atau alternatif kepada atasan produksi guna peningkatan efektivitas dan efisiensi di perusahaan.

11. *Human Resources & General Affairs*

Bertanggung jawab mengelola dan mengembangkan sumber daya manusia. Dalam hal ini termasuk perencanaan, pelaksanaan dan pengawasan sumber daya manusia dan pengembangan kualitas sumber daya manusia.

12. *PPC*

a. Tugas dan Tanggung Jawab

- Pengambil keputusan tertinggi dalam internal bagian PPC.
- Senantiasa mengawasi serta mengadakan peninjauan kembali terhadap prosedur dan tatacara pelaksanaan tugas para bawahan, terutama mengenai tingkat efektivitas dan efisiensi sistem kerja di bagian PPC.
- Mengadakan evaluasi terhadap prestasi kerja para bawahan bagian PPC secara periodik sesuai standar yang telah ditentukan.
- Mencegah tindakan-tindakan yang dapat merusak atau menghambat kelancaran/keutuhan sistem kerja di bagian PPC baik langsung maupun tidak langsung.
- Mengadakan hubungan kerja yang baik dengan bawahan hingga memungkinkan bagian PPC mencapai tujuan.
- Mengadakan hubungan kerja yang baik antar semua bagian lain untuk kelancaran kerja perusahaan.
- Menjadwalkan dan mengkoordinir *stock opname* inventaris pabrik secara periodik.
- Membuat jadwal produksi.

b. Wewenang

- Memberikan saran-saran perbaikan kepada divisi lain guna peningkatan priduktivitas perusahaan.
- Menghentikan atau merubah produksi yang sedang berjalan atau sesuai informasi terbaru dari *marketing*.

- Menegur secara langsung bawahan dari bagian lain, jika diketahui melakukan penyimpangan kerja.

13. *Inventory*

a. Tugas dan Tanggung Jawab

- Melakukan pekerjaan sesuai standar yang telah ditentukan.
- Memeriksa fisik barang, menerima barang dan menyimpan sesuai tempat yang disediakan
- Menyiapkan dan mengirimkan barang baik untuk kebutuhan produksi maupun ke *customer*.
- Melakukan *update* kartu stock baik untuk material, *spare part* dan produk.
- Melapor ke pimpinan langsung jika menemukan sesuatu yang menyimpang.
- Menjaga ketertiban dan memperhatikan keselamatan kerja selama bekerja.
- Menjaga kebersihan alat-alat yang digunakan dan lingkungan kerja.
- Senantiasa bekerjasama dengan karyawan di bagian lainnya.
- Melaksanakan tugas lain yang diberikan oleh atasan.

b. Wewenang

Memberikan saran-saran perbaikan kepada divisi lain guna peningkatan produktivitas perusahaan.

14. *Purchasing*

a. Tugas dan Tanggung Jawab

- Melakukan seleksi *supplier* untuk didaftar ke daftar *supplier* terpilih
- Membuat *Purchase Order* dan pembelian langsung (*cash*) dari permintaan pembelian yang disetujui oleh bagian terkait.
- Melakukan *monitoring* pemenuhan pengadaan barang dan jasa dari *supplier*.
- Lakukan *monitoring* kinerja *supplier* berkala kepada *supplier* yang bertransaksi dengan perusahaan di periode penilaian.

- Menginformasikan pembayaran dari tagihan kepada *supplier* ke bagian keuangan.
 - Melakukan *monitoring* hasil perbaikan dari *supplier* terkait permintaan perbaikan dan klaim yang diterbitkan perusahaan.
 - Menjaga hubungan baik dengan bagian lain untuk menjaga budaya perusahaan yang saling mendukung untuk peningkatan bisnis perusahaan.
- b. Wewenang
- Menetapkan *supplier* menjadi rekanan perusahaan berdasarkan seleksi *supplier*.
 - Melakukan negosiasi kepada *supplier* untuk pemenuhan pengadaan barang atau jasa.

15. Finance and Accounting

Melakukan semua proses penerimaan dan pengeluaran uang. Dan mencatat, memeriksa dan melaporkan semua transaksi yang berkaitan dengan keuangan perusahaan.

4.5 International Organization For Standardization

International Organization for Standardization atau yang biasa disingkat dengan ISO merupakan badan penetap standar internasional yang terdiri dari wakil-wakil dari badan standarisasi nasional setiap negara. Pada PT Baneex Indonesia telah mendapatkan akreditasi dari badan ISO yaitu ISO 9001 : 2015. ISO 9001 merupakan standar internasional di bidang sistem manajemen mutu. Suatu lembaga/organisasi yang telah mendapatkan akreditasi (pengakuan dari pihak lain yang independen) ISO tersebut, dapat dikatakan telah memenuhi persyaratan internasional dalam hal manajemen penjaminan mutu produk/jasa yang dihasilkannya.



Gambar IV.3 Sertifikat ISO 9001:2015

(Sumber: PT Baneex Indonesia, 2018)

4.6 Perhitungan *Safety Stock* dan *Lot Size*

Menurut wikipedia *safety stock* adalah istilah yang digunakan oleh ahli logistik untuk menggambarkan tingkat stok ekstra yang dipertahankan untuk mengurangi risiko kehabisan stok yang disebabkan oleh ketidakpastian pasokan dan permintaan. Sementara *lot size* merupakan suatu teknik yang digunakan untuk menentukan ukuran kuantitas pemesanan.

Berikut adalah perhitungan dalam menentukan *safety stock* dan *lot size* pada PT Baneex Indonesia:

1. *Safety Stock*

Safety stock = Jumlah keseluruhan produksi / Banyaknya Hari * 0.5

2. *Lot Size*

Lot size pada PT Baneex Indonesia sebesar 2000/hari untuk produk *wire* dan 120/hari untuk produk stamping.

4.7 Analisis Dokumen Penjadwalan Produksi

Analisis dokumen merupakan kegiatan pengumpulan informasi mengenai dokumen-dokumen yang digunakan dalam suatu sistem. Tujuan dari analisis dokumen adalah mengetahui dan memahami dokumen-dokumen apa saja yang terlibat dan mengalir dalam suatu sistem yang sedang berjalan. Setelah dokumen dianalisis nantinya dokumen ini akan digunakan sebagai pendukung dalam sistem informasi penjadwalan produksi usulan.

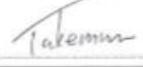
Dalam penjadwalan produksi ini dibutuhkan agar proses dapat berjalan sesuai dengan target yang ada. Dokumen-dokumen yang terlibat dalam sistem penjadwalan produksi ini adalah sebagai berikut :

4.7.1 Dokumen Masuk

Berikut dokumen masuk yang diterima PT Baneex Indonesia untuk melakukan penjadwalan produksi diantaranya :

1. Dokumen *Purchase Order*

Purchase Order (PO) adalah dokumen yang dibuat oleh pembeli untuk menunjukkan barang yang ingin mereka beli dari pihak penjual. *Purchase order* juga merupakan sebuah kontrak yang membentuk kesepakatan antara pembeli dan penjual mengenai barang yang ingin dibeli oleh pihak pembeli.

PT. TS TECH INDONESIA KAW. INDUSTRI KOTA BUKIT INDAH BLOK A-II / 3 BUNGURSARI PURWAKARTA 41181 JAWA BARAT INDONESIA TELP : (62)264-350 560 - 63 FAX : (62)264-350 559		TO: PT BANEEX INDONESIA Jl. Kemang Pratama Raya A4 Bekasi T. 8201162/32970284, F. 82400274 Jl. Raya Pasar Rebo No.28 Kp.Utan TLP 82407169 FAX : 82400813 Marcel 081281004145, Gengki (081213102500)				
FRM/MC/02/00		PURCHASE ORDER P027423/TS/18 21-Mar-18 REV. : 1 (QTY)				
PART No.	PART NAME	COLOR	QTY	DUE DATE	PRICE	AMOUNT
	PART FOR 2TQ					
81136-TG11-T010-24-0000	TRIM WIRE A R,FR CUSH		6,000	BY SCHED	2,163.06	12,978,360.00
81536-TG11-T010-24-0000	TRIM WIRE A L,FR CUSH		6,000		2,163.06	12,978,360.00
81136-TG11-T010-25-0000	TRIM WIRE C R,FR CUSH		6,000		1,233.00	7,398,000.00
81536-TG11-T010-25-0000	TRIM WIRE C L,FR CUSH		6,000		1,233.00	7,398,000.00
81136-TG11-T010-25-0000	TRIM WIRE B R,FR CUSH		6,000		1,047.06	6,282,360.00
81536-TG11-T010-25-0000	TRIM WIRE B L,FR CUSH		6,000		1,047.06	6,282,360.00
T201020-0600-0959-00	WIRE 01.6.0X359		18,000		4,998.00	89,964,000.00
SUB TOTAL						143,281,440.00
VAT						14,328,144.00
TOTAL						157,609,584.00
DELIVERY TO: PT.TS-TECH INDONESIA PAYMENT : 1 MONTH REMARKS : APR '18 PROD.						
SUPPLIER	ISSUED BY	CHECKED BY	APPROVED BY			
						

Gambar IV.4 Purchase Order

(Sumber: PT Baneex Indonesia, 2018)

Deskripsi : Dokumen yang ada pada gambar IV.4 merupakan *Purchase Order* yang dikirim oleh *customer* ke PT Baneex Indonesia.

Asal : PT TS Tech Indonesia (*Customer*)

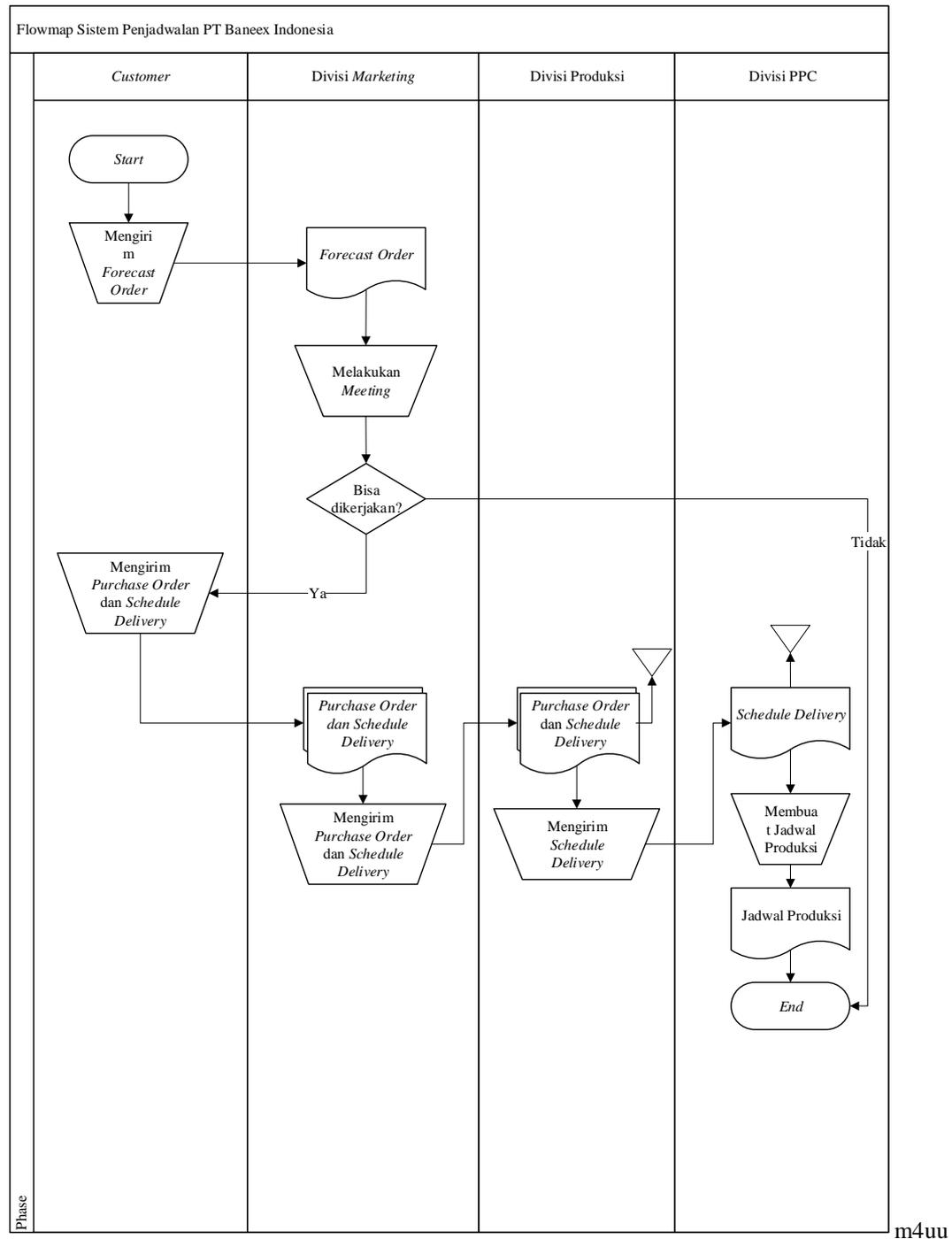
Tujuan : PT Baneex Indonesia

Kolom-kolom :

- Part No* : Berisikan nomor *part*/produk.
- Part Name* : Berisikan nama *part*/produk.
- QTY* : Jumlah pemesanan.

4.8.1 Flowmap Proses Penjadwalan Produksi

Flowmap pada Gambar IV.6 menjelaskan tentang proses penjadwalan produksi yang berjalan pada PT Baneex Indonesia:



Gambar IV.6 Flowmap Proses Penjadwalan Produksi

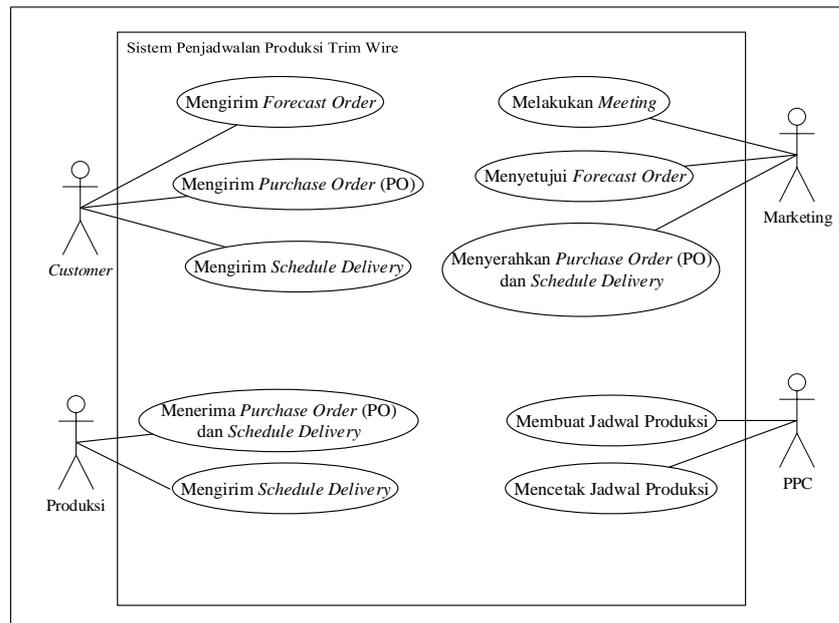
(Sumber: Hasil Analisis, 2018)

Keterangan Gambar IV.6:

1. *Customer* mengirim *Forecast Order* kepada PT Baneex Indonesia melalui divisi Marketing atau melalui E-mail.
2. Bila *Forecast Order* disetujui maka *Customer* akan dikirimkan oleh PT Baneex Indonesia *Forecast* selama 1 bulan.
3. Setelah itu *Customer* mengirimkan *Purchase Order* dan *Schedule Delivery* ke Divisi *Marketing* PT Baneex Indonesia.
4. Divisi PPC Menerima *Schedule Delivery* dari *Customer* yang dikirim melalui Divisi *Marketing* PT Baneex Indonesia lalu menyerahkan ke PO dan *schedule delivery* ke divisi produksi.
5. Divisi produksi menerima PO dan *schedule delivery* lalu menyerahkan *schedule delivery* ke divisi PPC.
6. Divisi PPC mengelola *schedule delivery* menjadi jadwal produksi sesuai dengan *Schedule Delivery* yang diberikan.
7. Divisi PPC Mencetak jadwal produksi tersebut.

4.9 Use Case Diagram Sistem Berjalan

Pemodelan sistem penjadwalan produksi berjalan ini akan direpresentasikan dengan UML melalui *Use Case Diagram*. Pertimbangan menggunakan diagram ini karena dapat menggambarkan sistem yang berjalan dan mudah dipahami karena menggambarkan interaksi antara aktor dengan sistem. Berikut merupakan *Use Case Diagram* dari sistem penjadwalan produksi pada PT Baneex Indonesia.



Gambar IV.7 Use Case Diagram Proses Penjadwalan Produksi

(Sumber: Hasil Analisis, 2018)

Penjelasan *Use Case Diagram* sistem informasi penjadwalan produksi di PT Baneex Indonesia adalah sebagai berikut:

1. Definisi Aktor

Pendefinisian aktor pada *Use Case Diagram* berdasarkan sistem informasi penjadwalan produksi di PT Baneex Indonesia dapat dilihat pada Tabel IV.1.

Tabel IV.1 Definisi Aktor

No	Aktor	Definisi
1	<i>Customer</i>	Melakukan <i>forecast order</i> , <i>purchase order</i> dan <i>schedule delivery</i> .
2	<i>Marketing</i>	Menerima segala <i>order</i> mulai dari <i>forecast order</i> sampai <i>schedule delivery</i> .
3	PPC	Membuat penjadwalan produksi berdasarkan <i>schedule delivery</i> dari <i>customer</i> .
4	Produksi	Menerima <i>purchase order</i> dan <i>schedule delivery</i> dari divisi marketing, lalu mengirimkan <i>schedule delivery</i> ke divisi PPC.

(Sumber: Hasil Analisis, 2018)

2. Definisi *Use Case*

Pendefinisian *use case* pada *use case diagram* berdasarkan sistem informasi penjadwalan produksi di PT Baneex Indonesia dapat dilihat pada Tabel IV.2.

Tabel IV.2 Definisi *Use Case*

No	Nama <i>Use case</i>	Definisi
1.	Mengirimkan <i>forecast order</i>	Mengirimkan <i>forecast order</i> ke PT Baneex Indonesia melalui divisi <i>marketing</i> .
2.	Melakukan <i>meeting</i>	Melakukan <i>meeting</i> terkait <i>forecast order</i> yang diterima dari <i>customer</i> .
3.	Menyetujui <i>forecast order</i>	Menyetujui atau tidak menyetujui <i>forecast order</i>
4.	Mengirimkan <i>purchase order</i> (PO)	Jika <i>forecast order</i> disetujui maka <i>customer</i> mengirimkan <i>purchase order</i> ke PT Baneex Indonesia melalui divisi <i>marketing</i> ..
5.	Mengirimkan <i>schedule delivery</i>	Customer mengirimkan <i>schedule delivery</i> ke divisi <i>marketing</i> PT Baneex Indonesia.
6.	Menyerahkan PO dan <i>Schedule Delivery</i>	Divisi <i>marketing</i> PT Baneex Indonesia menerima <i>purchase order</i> dan <i>schedule delivery</i> dari <i>customer</i> lalu menyerahkannya kepada divisi produksi.
7.	Menerima PO dan <i>schedule delivery</i>	Divisi produksi menerima PO dan <i>schedule delivery</i> dari <i>marketing</i> .
8.	Mengirim <i>schedule delivery</i>	Divisi produksi mengirim <i>schedule delivery</i> ke divisi PPC.
9.	Membuat jadwal produksi	Divisi PPC membuat jadwal produksi berdasarkan <i>schedule delivery</i> dari <i>customer</i> .
10.	Mencetak jadwal produksi	Jadwal produksi yang telah dibuat kemudian dicetak oleh divisi PPC.

(Sumber: Hasil Analisis, 2018)

4.10 Analisis Permasalahan

Permasalahan yang ada pada sistem penjadwalan produksi yang berjalan pada PT Baneex Indonesia dapat dilihat pada Tabel IV.3 berikut:

Tabel IV.3 Analisis Permasalahan

Analisis	Uraian Masalah	Sebab	Akibat	Solusi
<i>Performance</i> (Kinerja)	Untuk memproses penjadwalan produksi masih dilakukan secara manual.	Belum adanya sistem yang dapat membuat penjadwalan produksi pada PT Baneex Indonesia.	Penjadwalan produksi tidak dapat berjalan dengan baik.	Merancang dan membangun suatu sistem yang membantu proses penjadwalan produksi.
<i>Information</i> (Informasi)	Data penjadwalan produksi yang tidak terorganisir dengan baik contohnya redundansi dan kehilangan data.	Masih menggunakan <i>Microsoft Excel</i> .	Terjadinya redundansi data dan data dapat hilang.	Membuat proses penginputan data penjadwalan produksi menjadi terkomputerisasi.
<i>Economics</i> (Biaya)	-	-	-	-
<i>Control</i> (Pengendalian)	-	-	-	-

Tabel IV.3 Analisis Permasalahan (Lanjutan)

Analisis	Uraian Masalah	Sebab	Akibat	Solusi
<i>Efficiency</i> (Efisiensi)	-	-	-	-
<i>Service</i> (Layanan)	-	-	-	-

(Sumber: Hasil Analisis, 2018)

BAB V

ANALISIS DAN PEMBAHASAN

5.1 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem untuk sistem informasi penjadwalan produksi dapat dilihat pada Tabel V.1 sebagai berikut:

Tabel V.1 Analisis Kebutuhan Sistem

No.	Permasalahan	Kebutuhan <i>User</i>	Solusi	Kebutuhan Sistem
				<i>Functional Requirement</i>
1.	Untuk memproses penjadwalan produksi masih dilakukan secara manual.	Sistem terkomputerisasi yang dapat membantu proses penjadwalan produksi.	Merancang dan membangun suatu sistem yang membantu proses penjadwalan produksi.	Sistem mampu mengelola <i>purchase order</i> dan <i>schedule delivery</i> untuk menghasilkan penjadwalan produksi.
2.	Data terkait penjadwalan produksi yang tidak terorganisir dengan baik contohnya redundansi dan kehilangan data.	Sistem dengan <i>database</i> seperti MariaDB untuk menghindari kemungkinan redundansi data dan data dapat hilang.	Membuat proses penginputan data <i>customer</i> dan data produk menjadi terkomputerisasi.	Sistem dapat mengelola data <i>customer</i> dan data produk.

Tabel V.1 Analisis Kebutuhan Sistem (Lanjutan)

No.	Permasalahan	Kebutuhan <i>User</i>	Solusi	Kebutuhan Sistem
				<i>Functional Requirement</i>
3.		Penjadwalan produksi yang terkomputerisasi sehingga mempermudah departemen PPC dalam membuat jadwal produksi.	Menerapkan metode MPS (<i>Master Production Schedule</i>) untuk membuat penjadwalan produksi.	Sistem dapat memproses penjadwalan produksi berdasarkan metode MPS (<i>Master Production Schedule</i>)
4.		Sistem yang aman dan nyaman digunakan	Membuat sistem dengan menu <i>login</i> untuk pengguna.	Sistem dengan hak akses khusus untuk pengguna agar bisa masuk ke sistem.

(Sumber: Hasil Analisis, 2018)

5.1.1 *Functional Requirement*

Functional Requirement merupakan penjelasan tentang layanan yang perlu disediakan oleh sistem, bagaimana sistem menerima dan mengolah masukan, dan bagaimana sistem mengatasi situasi-situasi tertentu. Berikut adalah *functional requirement* dari sistem informasi penjadwalan produksi yang diusulkan:

1. Staf bagian produksi mengelola data *purchase order*, *customer* dan data produk.
2. Staf PPC mengelola *schedule delivery*.
3. Sistem penjadwalan produksi dapat memproses penjadwalan produksi berdasarkan metode MPS (*Master Production Schedule*).

4. *Login* dengan *username* dan *password* untuk masuk kedalam sistem.

5.1.2 Non Functional Requirement

Non Functional Requirement secara umum berisi batasan-batasan pada pelayanan atau fungsi yang disediakan oleh sistem. Berikut adalah *non functional requirement* dari sistem informasi penjadwalan produksi yang diusulkan:

1. Operasional

- Membutuhkan *web server* dan *database*. Seperti Apache dan MariaDB.

2. Keamanan

- Menggunakan *username* dan *password* untuk masuk ke dalam sistem.
- *Password* di enkripsi menggunakan md5.

3. Informasi

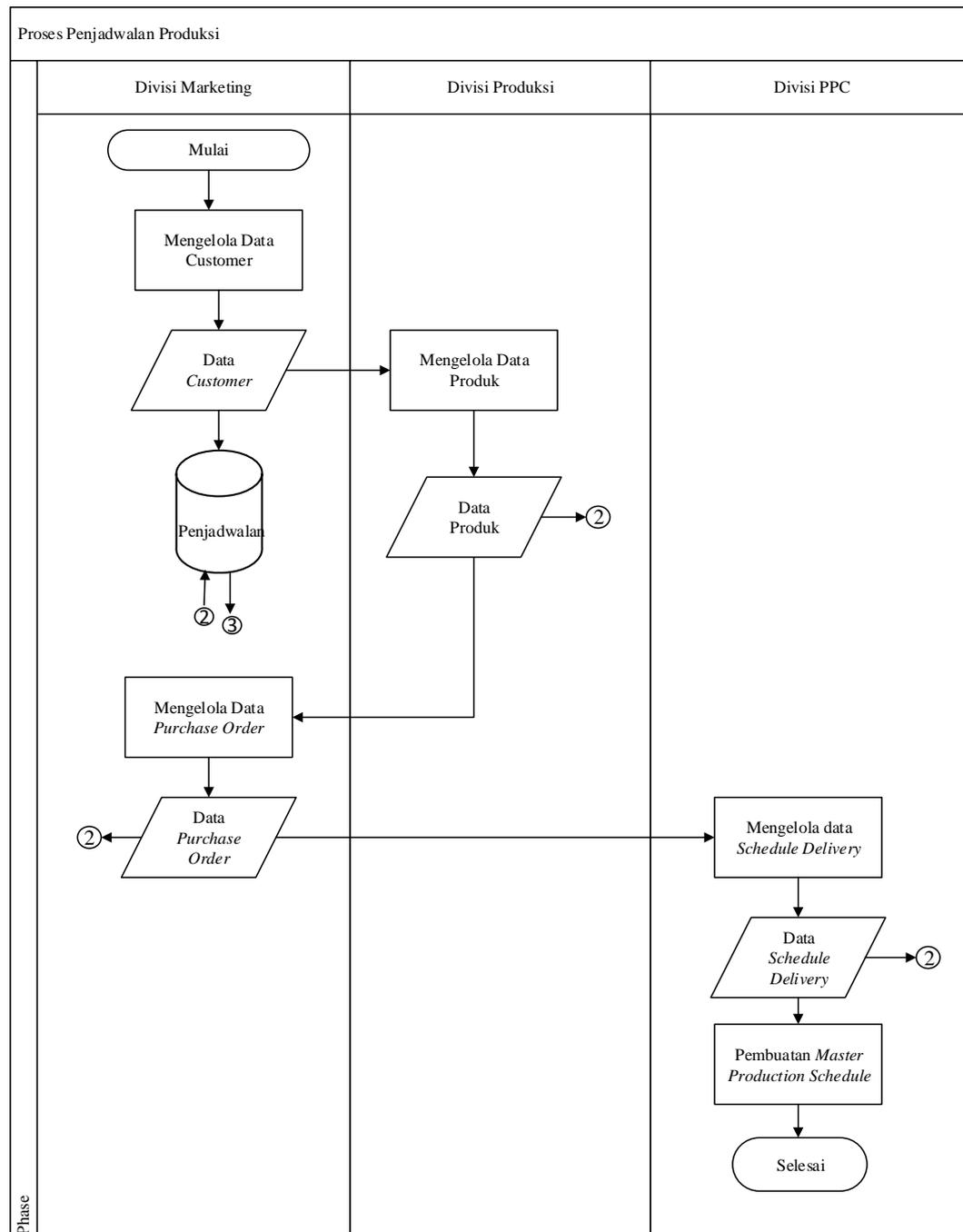
- Data *master production schedule* ditampilkan data produksi perhari selama satu bulan.

4. *Supportability*

- Minimal RAM 1 GB.
- Minimal *hardisk* 64 GB.
- Menggunakan *mouse* dan *keyboard*.
- Menggunakan *printer* untuk mencetak laporan *Master Production Schedule*.

5.2 Flowmap Sistem Penjadwalan Produksi Usulan

Berdasarkan analisis sistem yang sedang berjalan maka diusulkan proses perencanaan dan pengendalian produksi sebagai berikut:

Gambar V.1 *Flowmap Usulan*

(Sumber: Hasil Analisis, 2018)

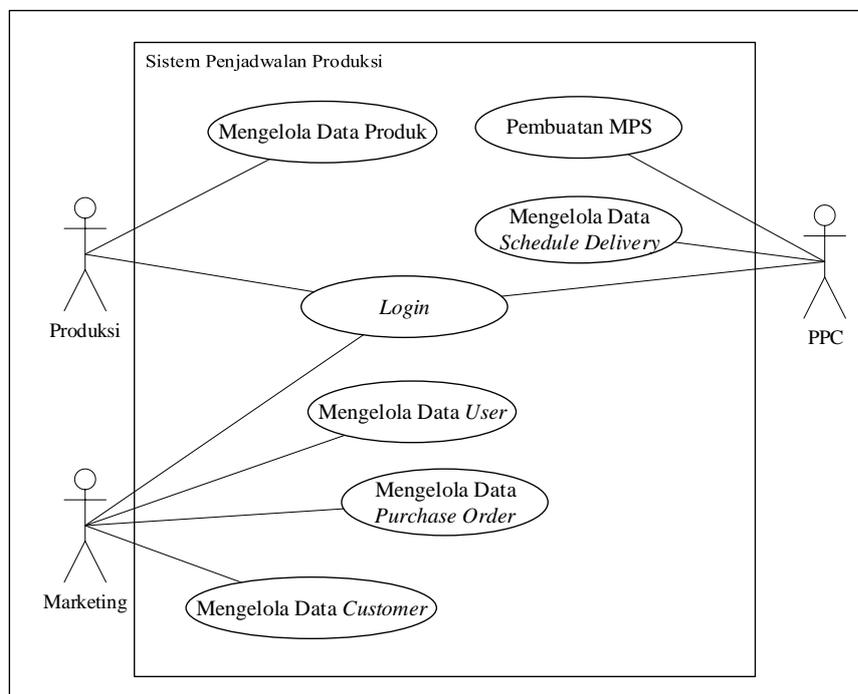
Keterangan Gambar V.1:

1. Bagian *marketing* menerima *purchase order* (PO) dan *schedule delivery* dari bagian *customer* lalu mengelola data *customer*.

2. Kemudian divisi produksi mengelola data produk.
3. Lalu divisi marketing mengelola data *purchase order* tersebut ke dalam aplikasi penjadwalan produksi.
4. Setelah divisi marketing mengelola *purchase order* ke dalam sistem, data *schedule delivery* dikirim ke bagian *production planning and control* (PPC) untuk diinputkan ke dalam aplikasi perencanaan produksi untuk menghasilkan *output* berupa data *planning* produksi dan juga *master production schedule* (MPS), dan hasil *output* tersebut dapat disimpan secara langsung ke dalam *database*.

5.3 Use Case Diagram (Usulan)

Use case diagram merupakan sebuah diagram yang menggambarkan fungsionalitas yang diharapkan dari sebuah sistem dan apa yang diperbuat sistem. Berikut adalah gambar *use case diagram* usulan berdasarkan analisis kebutuhan sistem pada tabel V.1:



Gambar V.2 *Use Case* Usulan

(Sumber: Hasil Analisis, 2018)

5.3.1 Use Case Description

Berikut adalah *use case description* dari *use case* usulan diantaranya:

Tabel V.2 *Use Case Description Login*

Nama Use Case	<i>Login</i>
Aktor	PPC, Produksi, dan Marketing
Deskripsi	<i>Use case</i> ini menggambarkan setiap aktor dapat masuk kedalam sistem
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>User</i> membuka <i>form login</i> 2. <i>User</i> memasukkan <i>username</i> dan <i>password</i> pada <i>form login</i> 3. Sistem akan memvalidasi data, apakah data yang telah dimasukkan telah benar atau tidak 4. Jika data benar, maka <i>user</i> akan masuk kedalam halaman utama sistem 5. Jika data salah, maka <i>user</i> akan kembali kembali ke halaman <i>form login</i>
<i>Alternate Flow</i>	

(Sumber: Hasil Analisis, 2018)

Tabel V.3 *Use Case Description Mengelola Data Customer*

Nama Use Case	Mengelola Data <i>Customer</i>
Aktor	<i>Marketing</i>
Deskripsi	<i>Use case</i> ini menggambarkan aktor yang mengelola data <i>customer</i> .
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. Staf <i>marketing</i> memilih menu <i>customer</i>. 2. Sistem akan mengambil semua informasi data <i>customer</i> yang tersimpan pada <i>database</i> dan menampilkan informasi tersebut kedalam tabel data <i>customer</i>. 3. Staf <i>marketing</i> dapat melihat <i>detail database</i> yang tersimpan dan dapat juga menambahkan, mengedit, menghapus, dan mencari data <i>customer</i>.
<i>Alternate Flow</i>	

(Sumber: Hasil Analisis, 2018)

Tabel V.4 *Use Case Description* Mengelola Data Produk

Nama <i>Use Case</i>	Mengelola Data Produk
Aktor	Produksi
Deskripsi	<i>Use case</i> ini menggambarkan aktor yang mengelola data produk.
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. Staf produksi memilih menu produk. Lalu sistem akan mengambil semua informasi data produk yang tersimpan pada <i>database</i> dan menampilkan informasi tersebut kedalam tabel data produk 2. Staf produksi dapat melihat <i>detail database</i> yang tersimpan dan dapat juga menambahkan, mengedit, menghapus, dan mencari data produk.
<i>Alternate Flow</i>	

(Sumber: Hasil Analisis, 2018)

Tabel V.5 *Use Case Description* Mengelola Data *Purchase Order*

Nama <i>Use Case</i>	Mengelola Data <i>Purchase Order</i> (PO)
Aktor	<i>Marketing</i>
Deskripsi	<i>Use case</i> ini menggambarkan aktor yang mengelola data PO.
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>User</i> membuka <i>form Purchase Order</i> lalu <i>user</i> menekan tombol tambah <i>Purchase Order</i>. 2. <i>User</i> memasukkan nomor PO, tanggal dan nama <i>customer</i> 3. Sistem akan mevalidasi data yang akan dimasukkan. Jika data sudah ada maka sistem akan mengirim pesan maaf data sudah ada. Jika data berhasil masuk maka sistem akan mengirim pesan data berhasil dimasukkan dan sistem akan membuka <i>form Purchase Order Detail</i> 4. <i>User</i> memasukkan nama barang dan Qty/jumlah pemesanan 5. Jika data sudah ada maka sistem akan mengirim pesan maaf data sudah ada. Jika data berhasil masuk maka sistem akan mengirim pesan data berhasil dimasukkan. 6. Staf produksi dapat juga mengedit, menghapus, dan mencari data <i>purchase order</i>.
<i>Alternate Flow</i>	

(Sumber: Hasil Analisis, 2018)

Tabel V.6 *Use Case Description* Mengelola Data *Schedule Delivery*

Nama <i>Use Case</i>	Mengelola Data <i>Schedule Delivery</i>
Aktor	PPC
Deskripsi	<i>Use case</i> ini menggambarkan aktor yang memasukkan data <i>Schedule Delivery</i>
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>User</i> membuka <i>form Schedule Delivery</i> 2. <i>User</i> memilih data <i>Purchase Order</i> 3. <i>User</i> memasukan data <i>Schedule Delivery</i> 4. Sistem akan menyimpan data <i>Schedule Delivery</i>
<i>Alternate Flow</i>	

(Sumber: Hasil Analisis, 2018)

Tabel V.7 *Use Case Description* Pembuatan MPS

Nama <i>Use Case</i>	Pembuatan MPS
Aktor	PPC
Deskripsi	<i>Use case</i> ini menggambarkan aktor yang akan membuat MPS
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>User</i> membuka <i>form</i> MPS 2. <i>User</i> memilih data yang ingin dibuat
<i>Alternate Flow</i>	

Sumber: Hasil Analisis, 2018

Tabel V.8 *Use Case Description* Mengelola Data *User*

Nama <i>Use Case</i>	Mengelola Data <i>User</i>
Aktor	<i>Marketing</i>
Deskripsi	<i>Use case</i> ini menggambarkan aktor yang mengelola data <i>user</i> .
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. Staf produksi memilih menu <i>user</i>. 2. Sistem akan mengambil semua informasi data <i>user</i> yang tersimpan pada <i>database</i> dan menampilkan informasi tersebut kedalam tabel data produk 3. Staf <i>marketing</i> dapat melihat <i>detail database</i> yang tersimpan dan dapat juga menambahkan, mengedit, menghapus, dan mencari data <i>User</i>.
<i>Alternate Flow</i>	

(Sumber: Hasil Analisis, 2018)

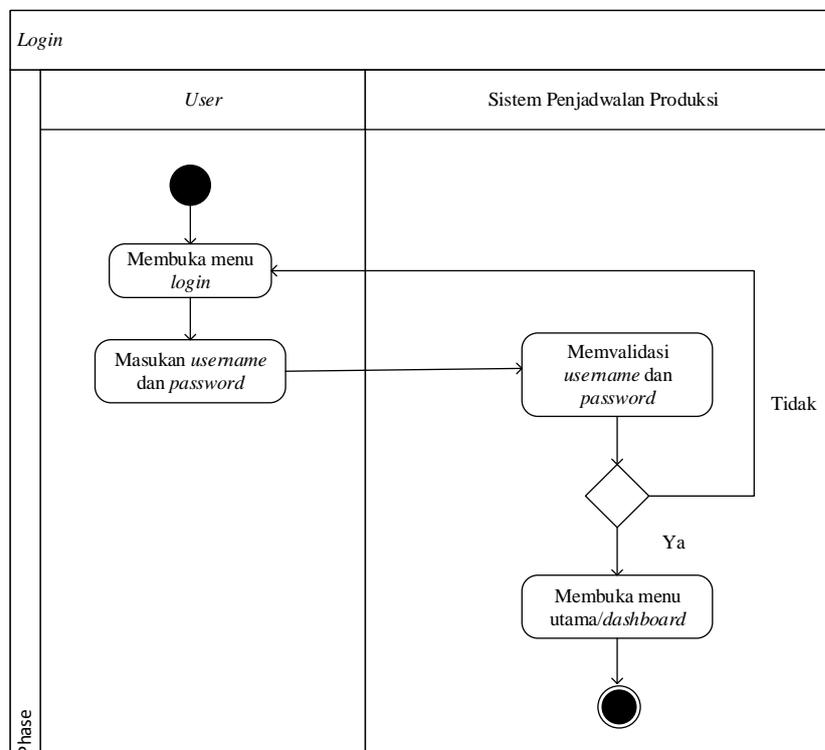
5.4 Activity Diagram

Setelah melihat *Use Description* diatas, penggambaran dengan menggunakan *Activity Diagram* dari masing-masing *Use Case* diperlukan. *Activity Diagram* digunakan untuk model perilaku dalam independen proses bisnis benda. Dalam banyak hal, *Activity Diagram* dapat dipandang sebagai diagram aliran data yang canggih yang digunakan dalam hubungannya dengan analisis terstruktur. Namun, tidak seperti aliran data diagram, *Activity Diagram* termasuk notasi yang membahas pemodelan paralel, kegiatan bersamaan dan proses.

Maka dari itu *Activity Diagram* pada sistem penjadwalan produksi pada PT Baneex Indonesia yang diusulkan di antaranya adalah sebagai berikut:

1. Activity diagram login

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu staf produksi, staf PPC dan staf marketing untuk mendapatkan akses kedalam sistem informasi penjadwalan produksi. Berikut merupakan *activity diagram login* sistem informasi penjadwalan produksi:

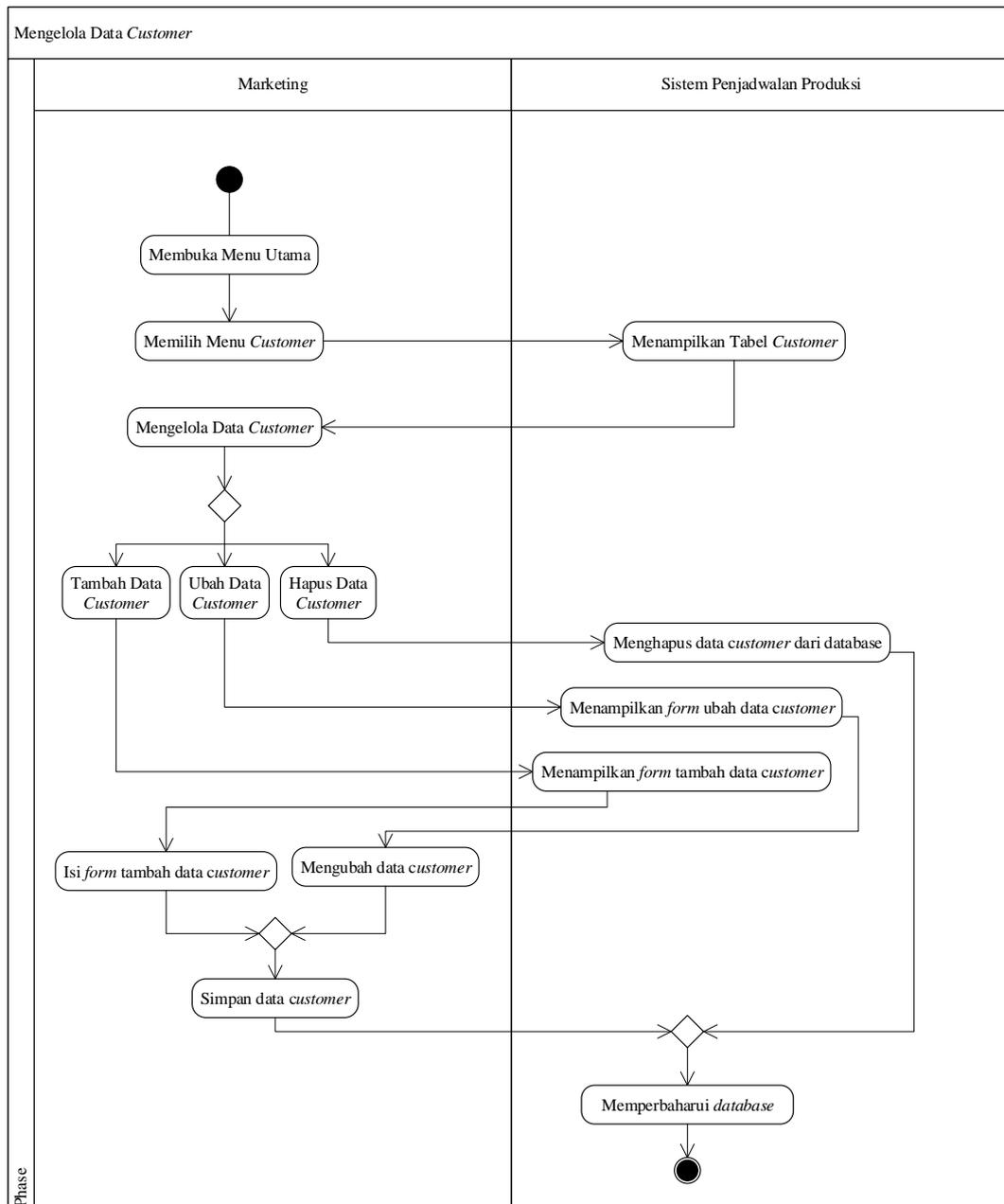


Gambar V.3 Activity Diagram Login

(Sumber: Hasil Analisis, 2018)

2. Activity Diagram Mengelola Data Customer

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu staf marketing untuk mengelola data *customer* mulai dari menambah data, mengubah data, sampai hapus data.

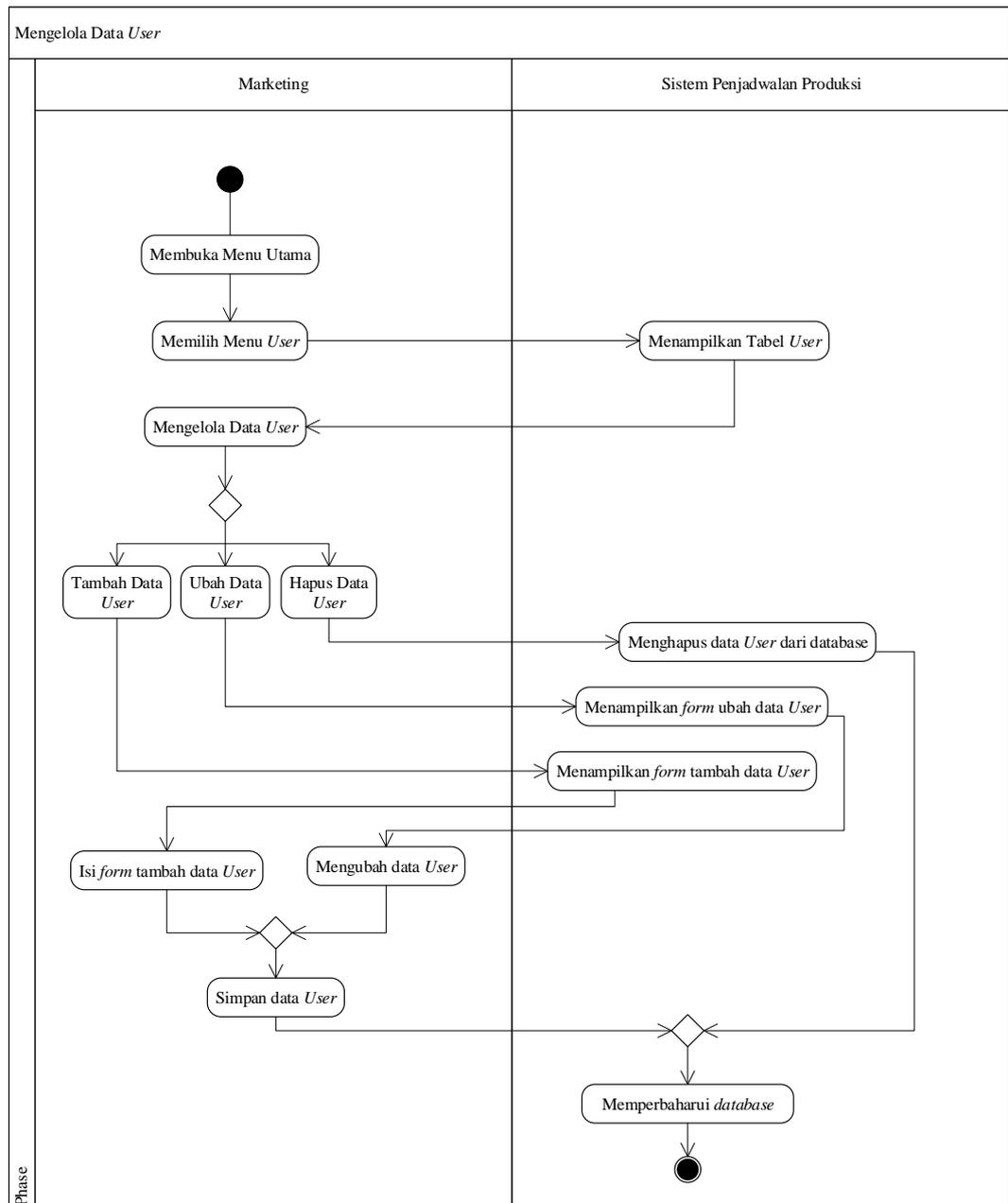


Gambar V.4 Activity Diagram Mengelola Data Customer

(Sumber: Hasil Analisis, 2018)

3. Activity Diagram Mengelola Data *User*

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu kepala produksi untuk mengelola data *user* mulai dari menambah data, mengubah data, sampai hapus data.

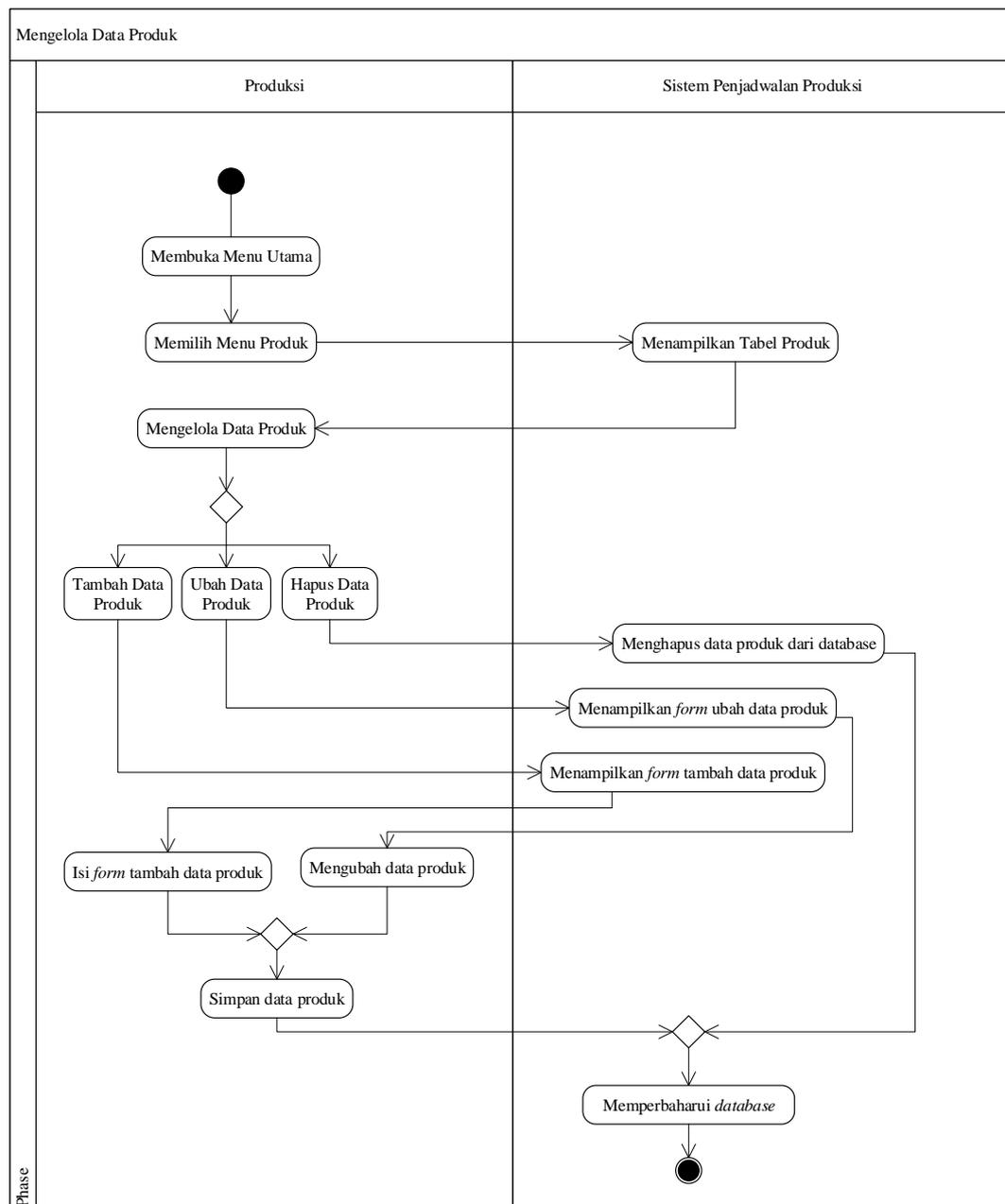


Gambar V.5 Activity Diagram Mengelola Data *User*

(Sumber: Hasil Analisis, 2018)

4. Activity Diagram Mengelola Data Produk

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu staf produksi untuk mengelola data produk mulai dari menambah data, mengubah data, sampai hapus data.

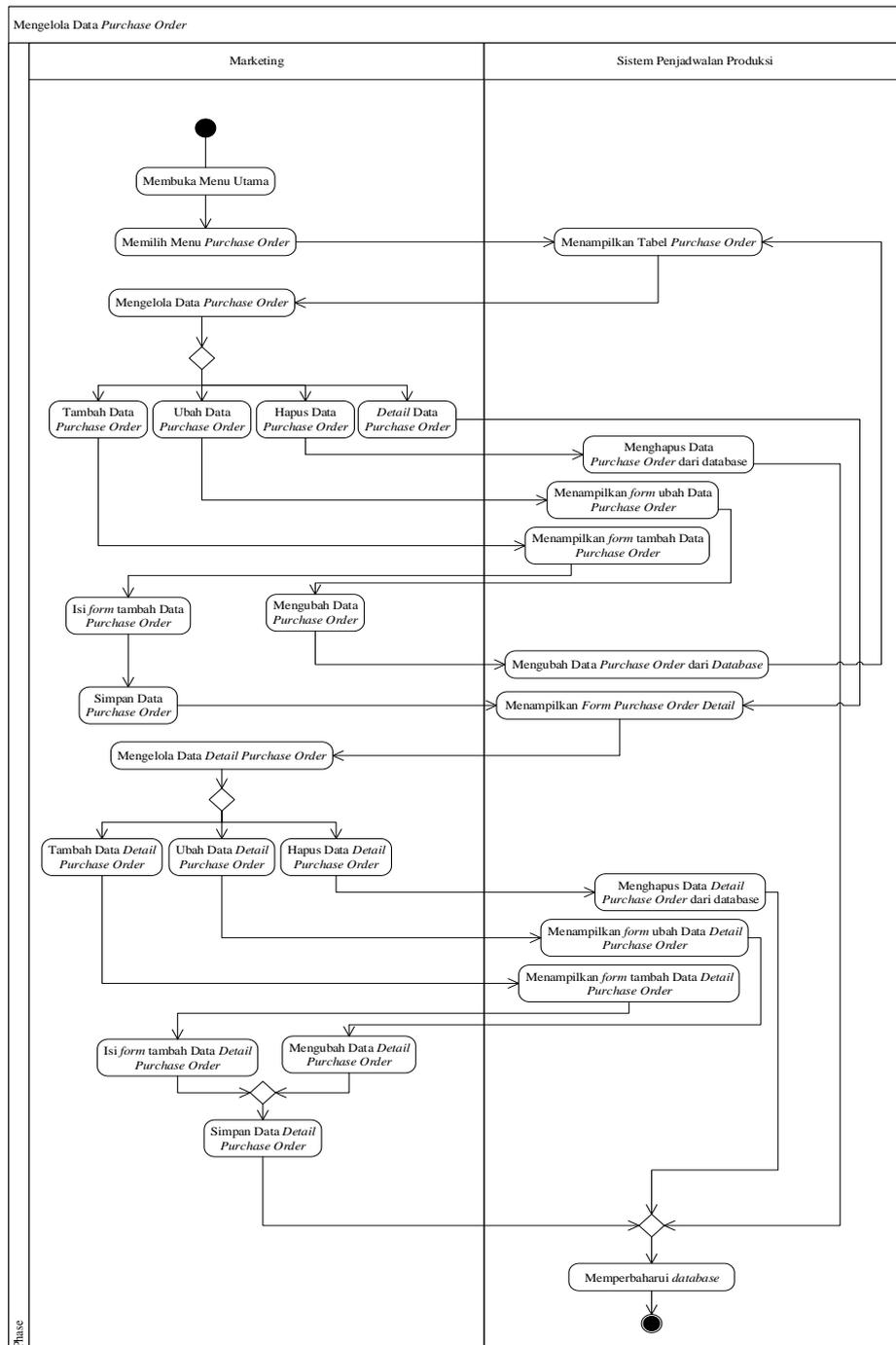


Gambar V.6 Activity Diagram Mengelola Data Produk

(Sumber: Hasil Analisis, 2018)

5. *Activity Diagram Mengelola Data Purchase Order*

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu staf marketing untuk mengelola data *purchase order* mulai dari menambah data, mengubah data, sampai hapus data.

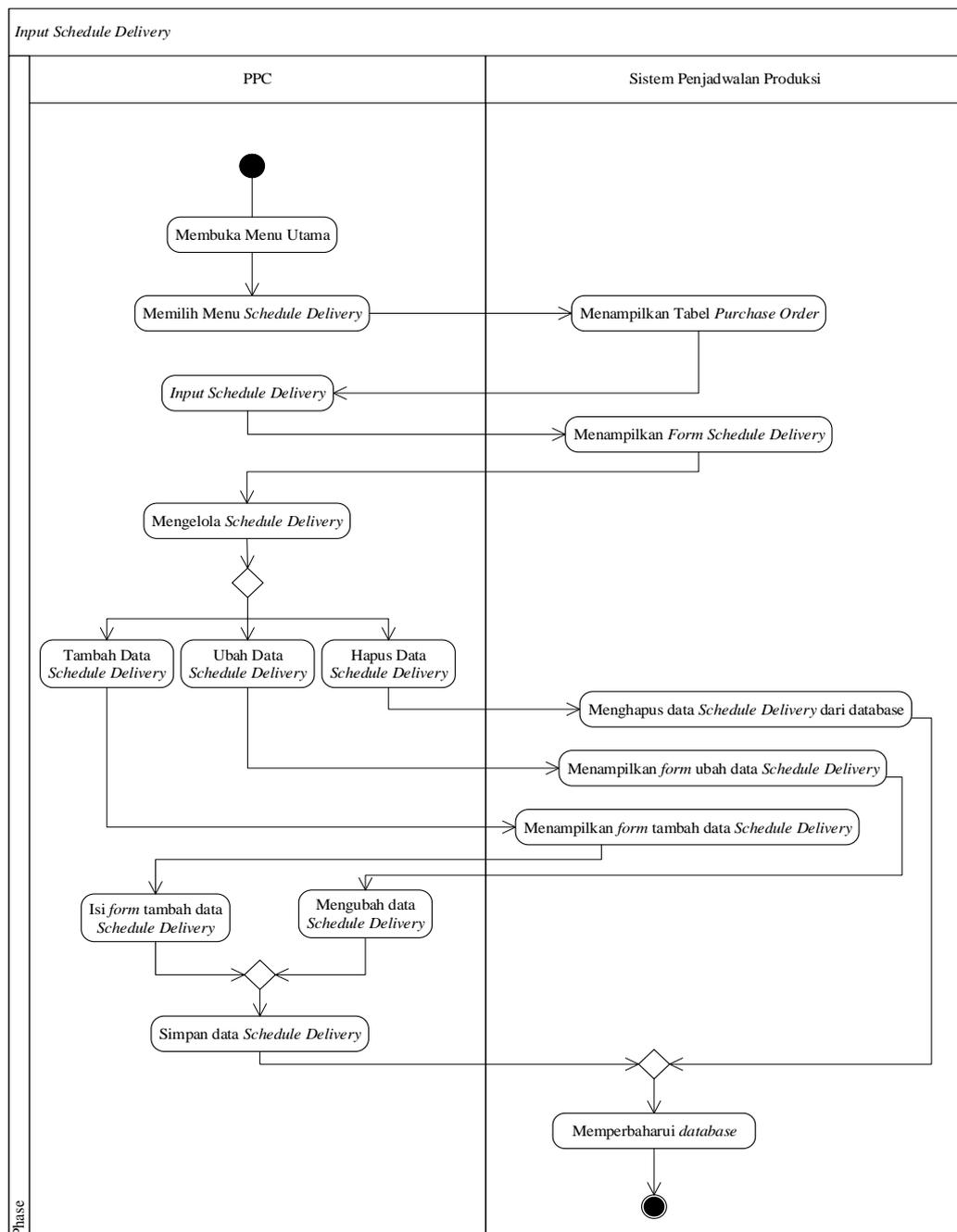


Gambar V.7 *Activity Diagram Mengelola Data Purchase Order*

(Sumber: Hasil Analisis, 2018)

6. *Activity Diagram Mengelola Data Schedule Delivery*

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu staf PPC untuk *input* dan mengelola *schedule delivery* mulai dari menambah data, mengubah data, sampai hapus data.

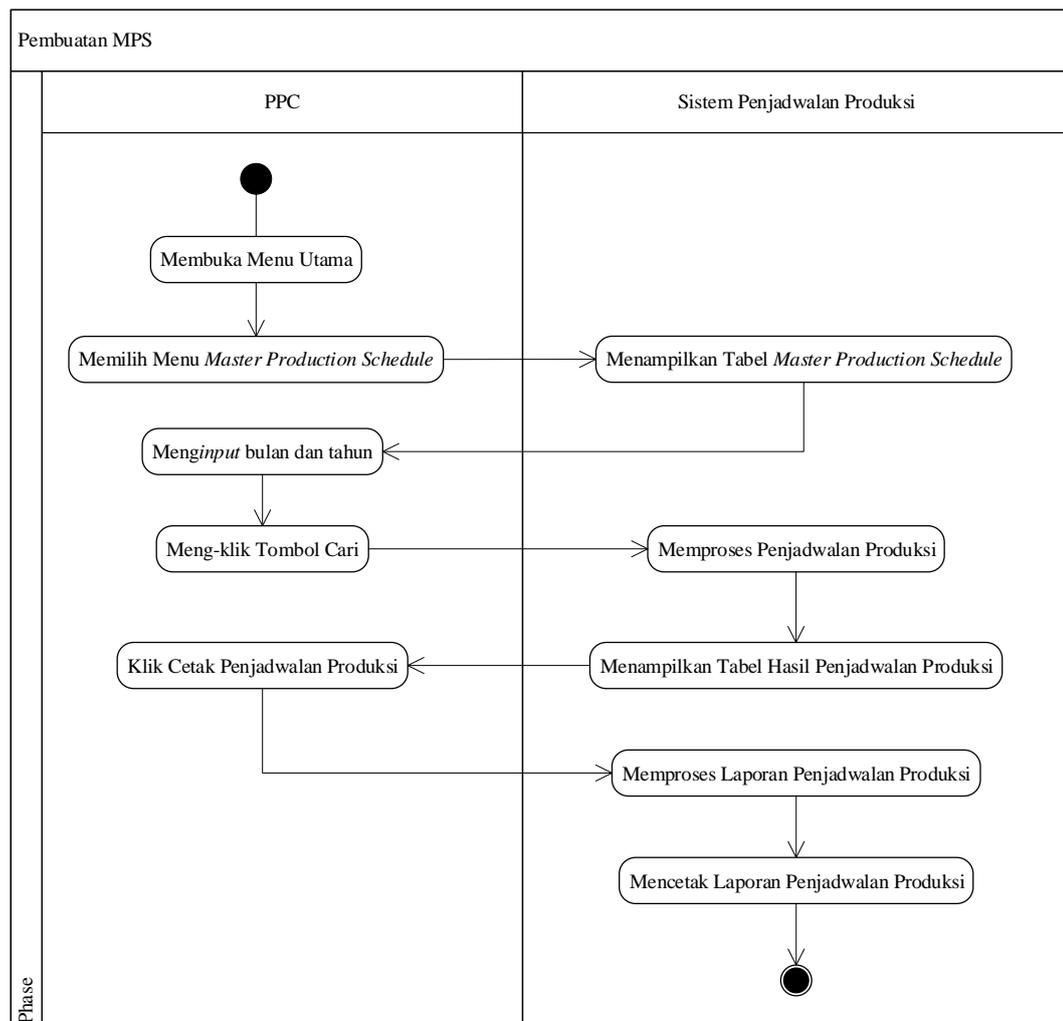


Gambar V.8 *Activity Diagram Mengelola Data Schedule Delivery*

(Sumber: Hasil Analisis, 2018)

7. Activity Diagram Pembuatan MPS

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu staf PPC untuk memproses penjadwalan produksi dengan metode MPS (*Master Production Schedule*) dan mencetak laporan penjadwalan produksi.



Gambar V.9 Activity Diagram Cetak MPS

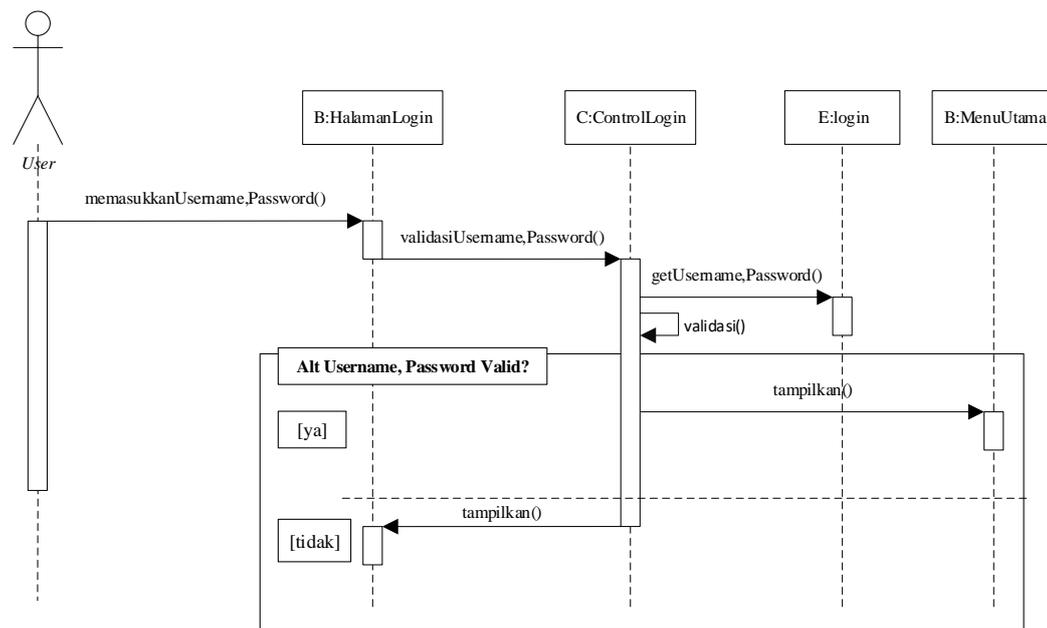
(Sumber: Hasil Analisis, 2018)

5.5 Sequence Diagram

Sequence diagram merupakan diagram untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Berikut merupakan beberapa *sequence diagram* usulan dari sistem informasi penjadwalan produksi:

1. *Sequence Diagram Login*

Sequence diagram login merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses *login*. Proses *login* dalam sistem informasi penjadwalan produksi ini dilakukan oleh *user* agar dapat mengakses sistem. Berikut merupakan *sequence diagram* dari *use case login*:

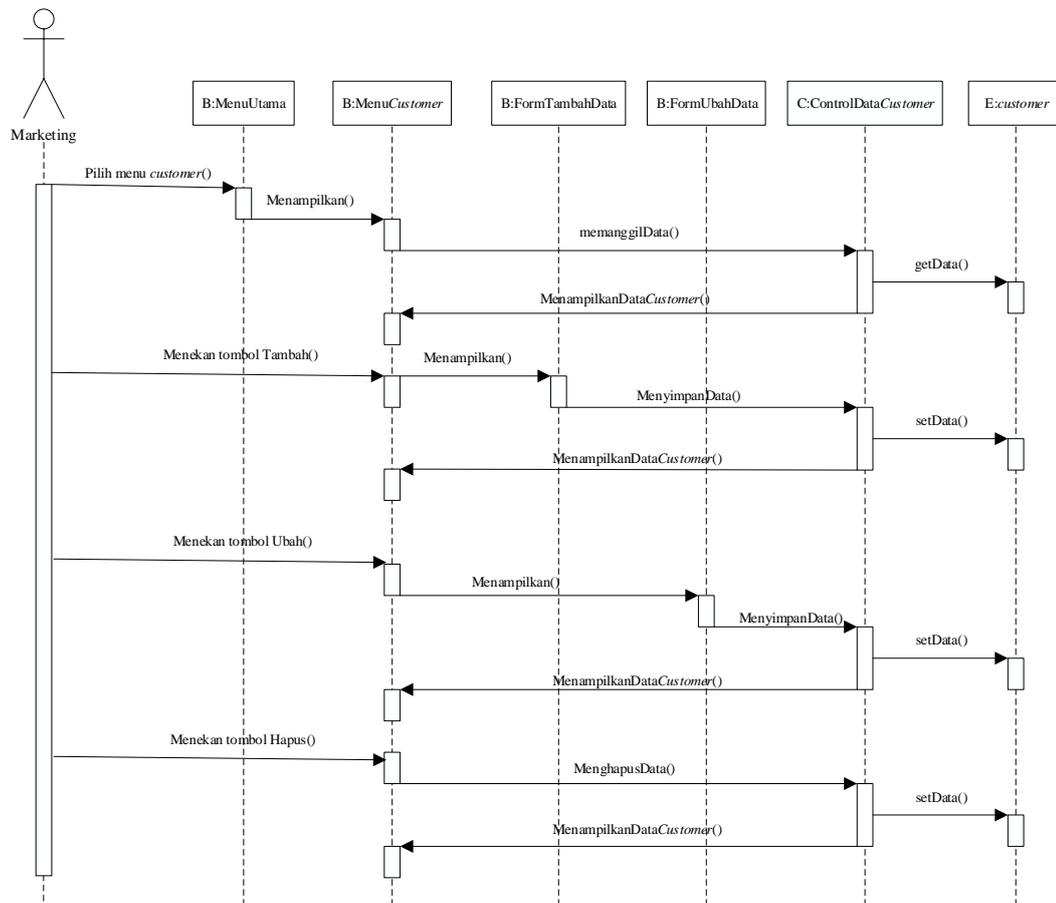


Gambar V.10 *Sequence Diagram Login*

(Sumber: Hasil Analisis, 2018)

2. *Sequence Diagram Mengelola Data Customer*

Sequence diagram mengelola data *customer* merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses mengelola data *customer*. Proses mengelola data *customer* dalam sistem informasi penjadwalan produksi ini dilakukan oleh user yaitu staf marketing agar dapat menyimpan, menambah, dan menghapus data. Berikut merupakan *sequence diagram* dari *use case* mengelola data *customer*:

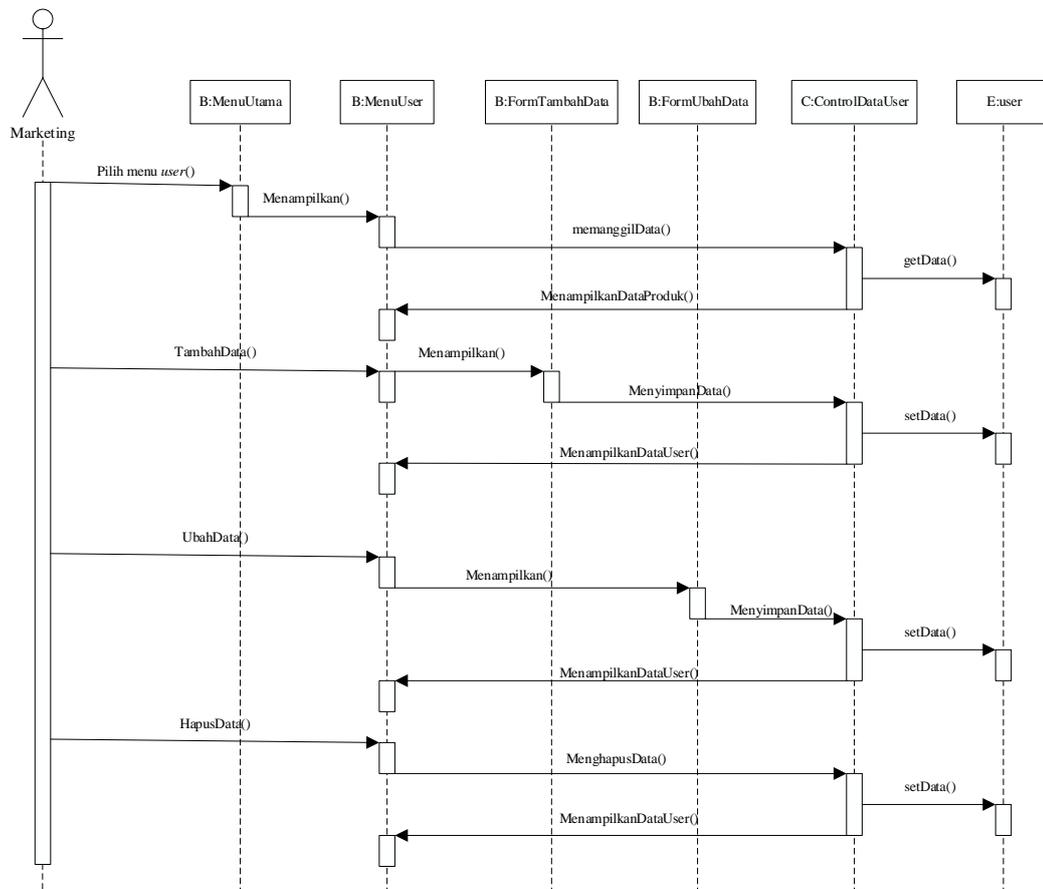


Gambar V.11 *Sequence Diagram* Mengelola Data Customer

(Sumber: Hasil Analisis, 2018)

3. *Sequence Diagram* Mengelola Data User

Sequence diagram mengelola data produk merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses mengelola data *user*. Proses mengelola data produk dalam sistem informasi penjadwalan produksi ini dilakukan oleh user yaitu kepala produksi agar dapat menyimpan, menambah, dan menghapus data. Berikut merupakan *sequence diagram* dari *use case* mengelola data *user*:

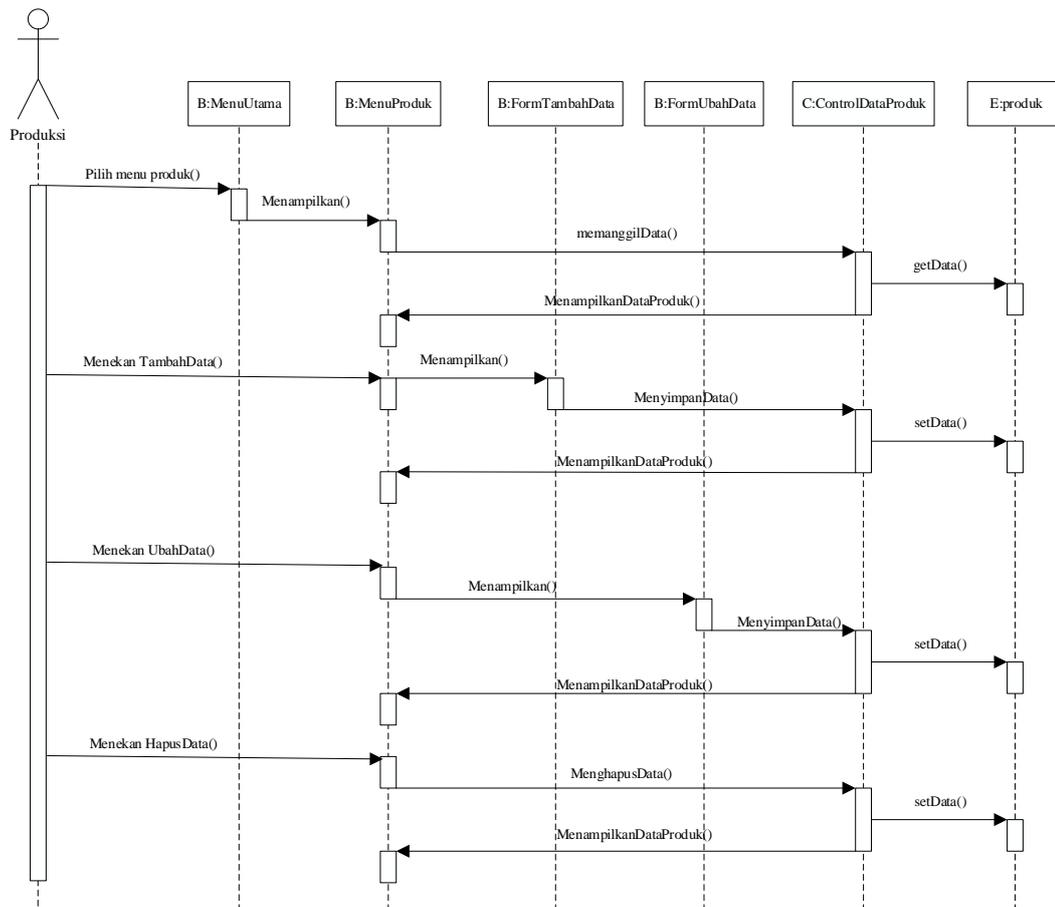


Gambar V.12 *Sequence Diagram* Mengelola Data User

(Sumber: Hasil Analisis, 2018)

4. *Sequence Diagram* Mengelola Data Produk

Sequence diagram mengelola data produk merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses mengelola data produk. Proses mengelola data produk dalam sistem informasi penjadwalan produksi ini dilakukan oleh user yaitu staf produksi agar dapat menyimpan, menambah, dan menghapus data. Berikut merupakan *sequence diagram* dari *use case* mengelola data produk:

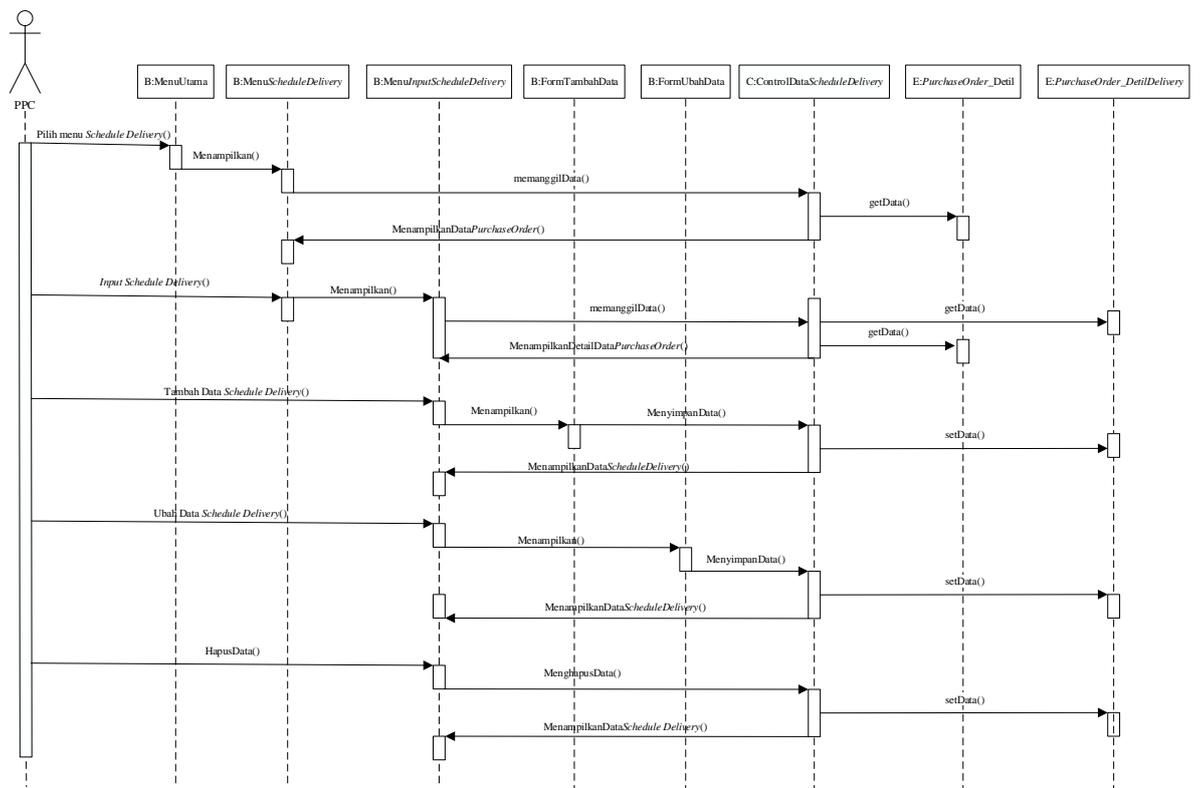


Gambar V.13 *Sequence Diagram* Mengelola Data Produk

(Sumber: Hasil Analisis, 2018)

5. *Sequence Diagram* Mengelola Data *Purchase Order*

Sequence diagram mengelola data *purchase order* merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses mengelola data *purchase order*. Proses mengelola data *purchase order* dalam sistem informasi penjadwalan produksi ini dilakukan oleh *user* agar dapat menyimpan, menambah, dan menghapus data. Berikut merupakan *sequence diagram* dari *use case* mengelola data *purchase order*:

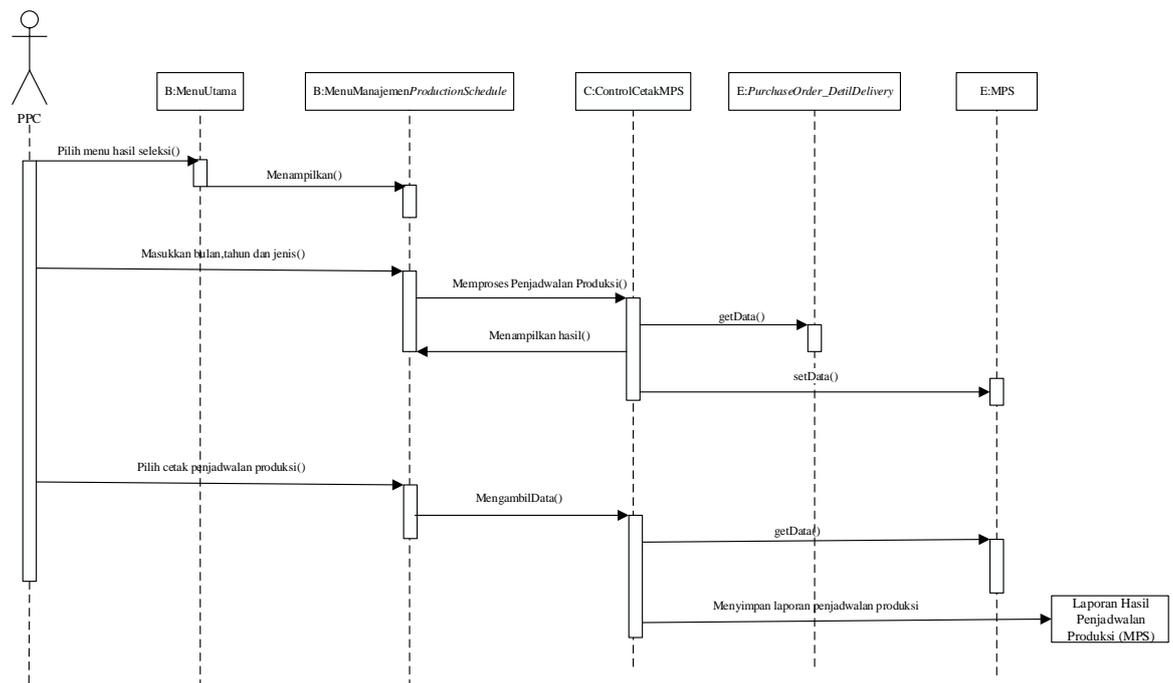


Gambar V.15 *Sequence Diagram* Mengelola Data *Schedule Delivery*

(Sumber: Hasil Analisis, 2018)

7. *Sequence Diagram* Pembuatan MPS

Sequence diagram cetak MPS merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses cetak MPS. Proses cetak MPS dalam sistem informasi penjadwalan produksi ini dilakukan oleh user yaitu staf PPC untuk memproses penjadwalan produksi dengan metode MPS dan mencetak laporan penjadwalan produksi. Berikut merupakan *sequence diagram* dari *use case* pembuatan MPS:

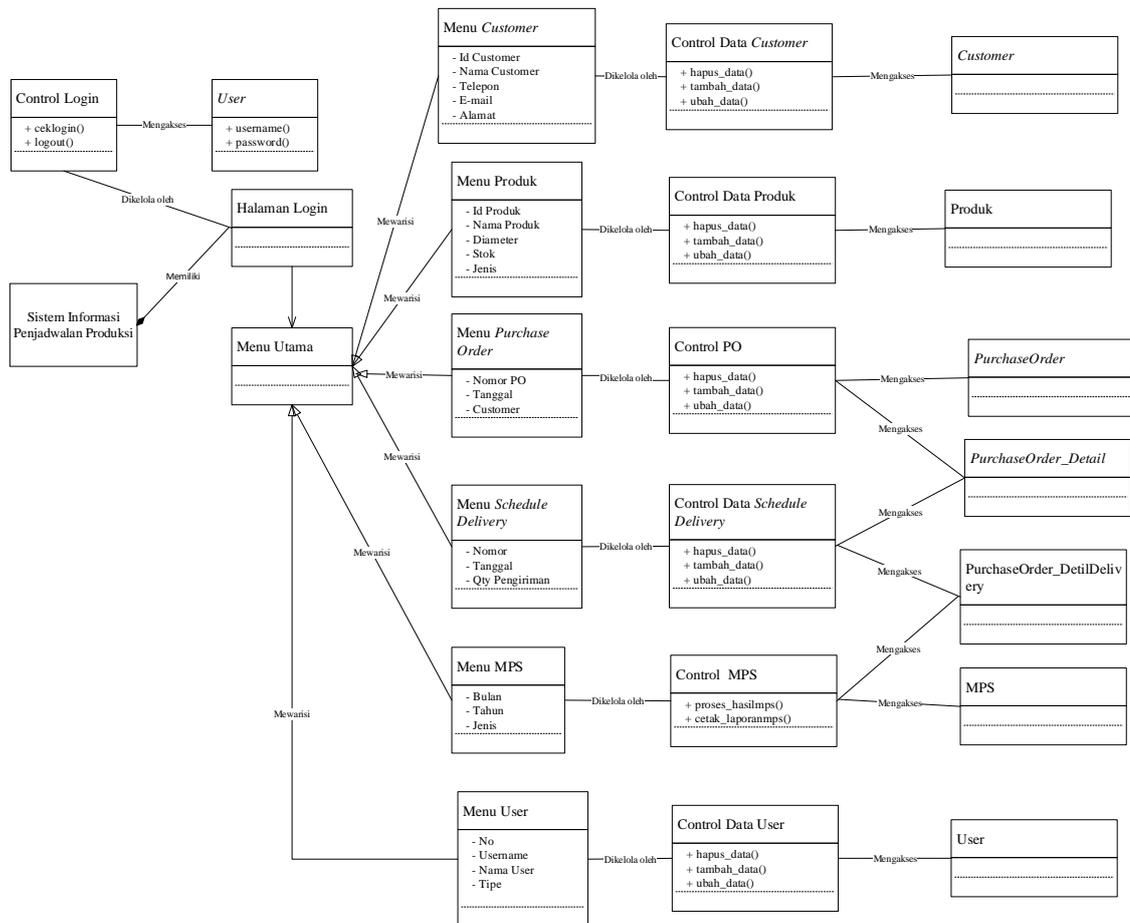


Gambar V.16 *Sequence Diagram* Pembuatan MPS

(Sumber: Hasil Analisis, 2018)

5.6 *Class Diagram*

Class diagram pada usulan sistem informasi penjadwalan produksi digunakan untuk menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat, untuk membangun sistem *class diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem. Berikut adalah *class diagram* usulan sistem informasi penjadwalan produksi:

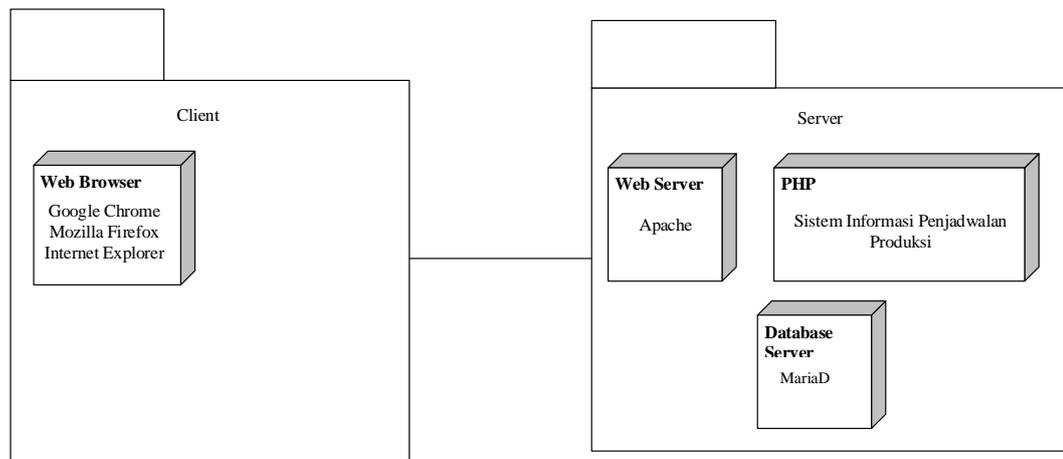


Gambar V.17 Class Diagram Sistem Informasi Penjadwalan Produksi

(Sumber: Hasil Analisis, 2018)

5.7 Deployment Diagram

Deployment diagram pada usulan sistem informasi penjadwalan produksi digunakan untuk mewakili komponen-komponen *software* dan cara *software* ditempatkan pada arsitektur fisik atau infrastruktur sistem informasi. Berikut merupakan *deployment diagram* sistem informasi penjadwalan produksi.



Gambar V.18 *Deployment Diagram* Sistem Informasi Penjadwalan Produksi

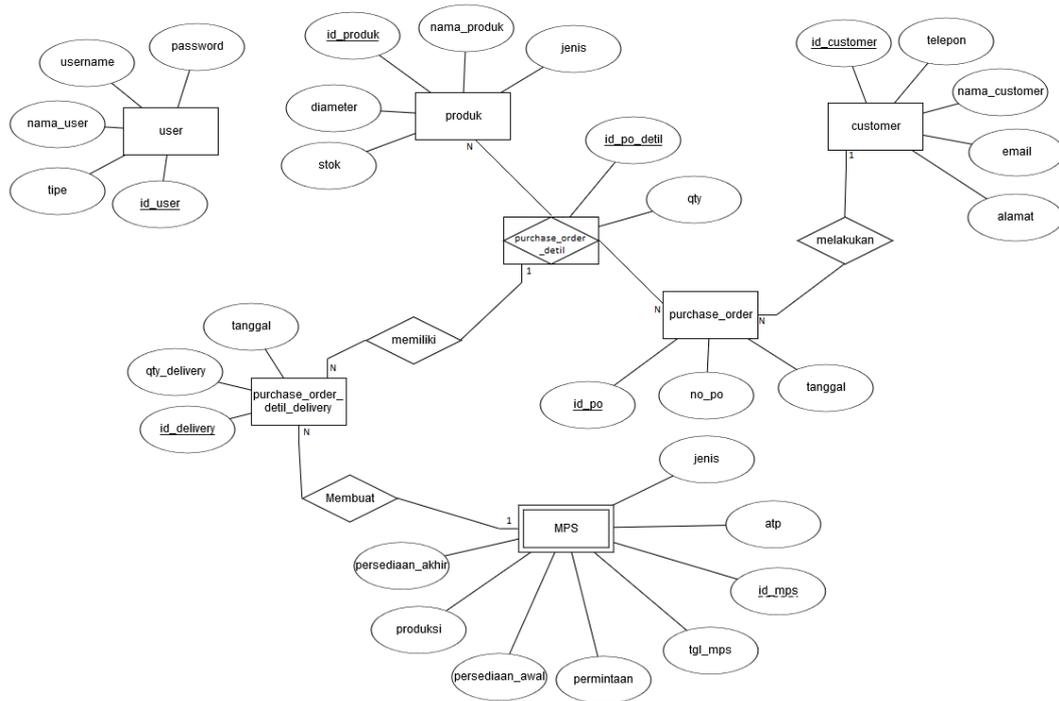
(Sumber: Hasil Analisis, 2018)

5.8 Pemodelan Data Sistem Usulan

Pemodelan data pada sistem informasi rekrutmen karyawan di PT Nandya Karya Perkasa menggunakan dua cara yaitu *Entity Relationship Diagram* (ERD) yang digunakan untuk menjelaskan hubungan antar data dalam basis data dan kamus data yang digunakan untuk menjelaskan isi dari *database* yang digunakan dalam sistem usulan.

5.8.1 *Entity Relationship Diagram* (ERD)

ERD digunakan sebagai suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi dalam sistem usulan. ERD digambarkan dengan simbol dan notasi yang berbeda, berikut adalah ERD sistem informasi penjadwalan produksi usulan:

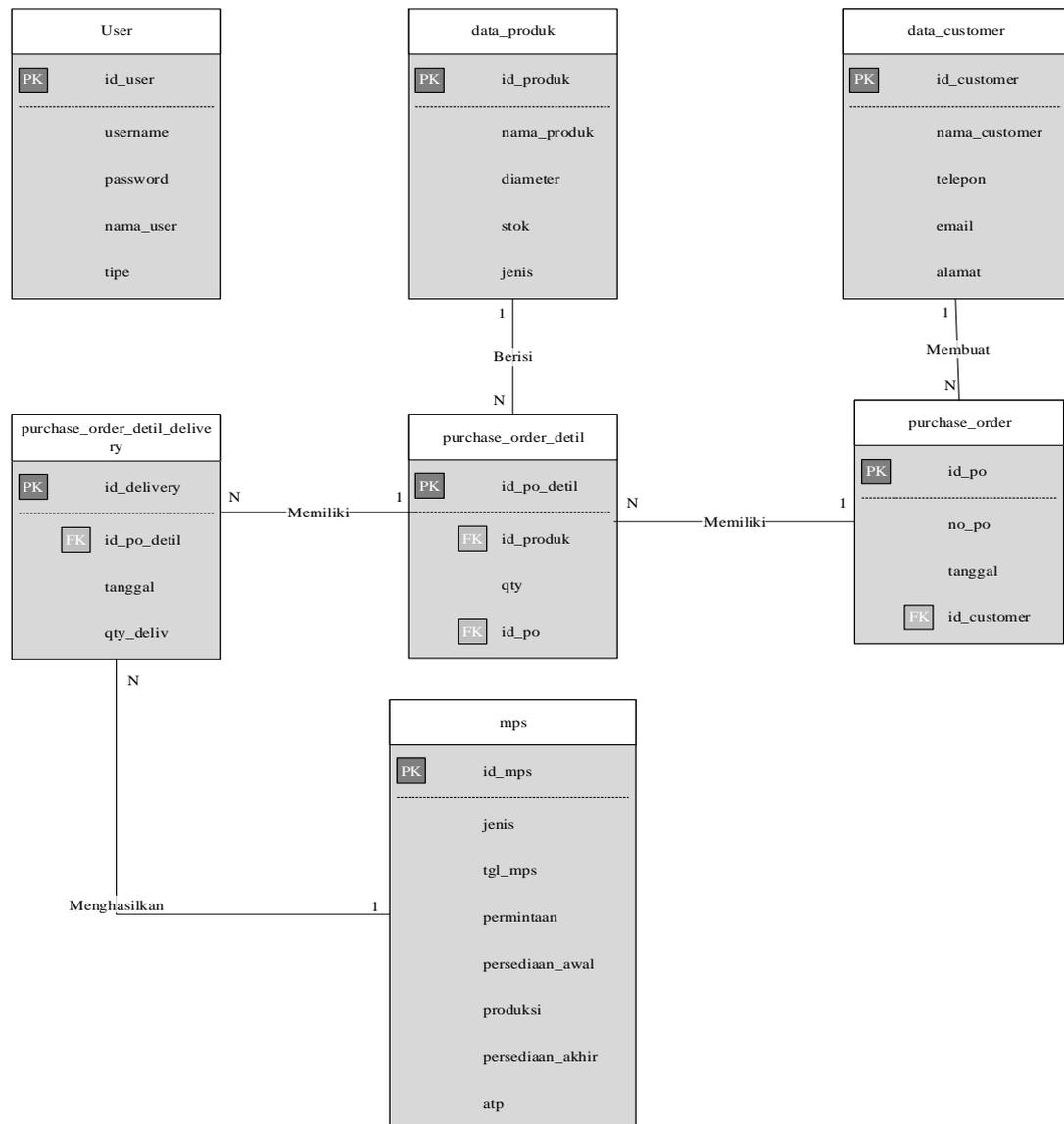


Gambar V.19 ERD Sistem Informasi Penjadwalan Produksi

(Sumber: Hasil Analisis, 2018)

5.8.2 Conceptual Data Model

Conceptual Data Model digunakan untuk menggambarkan atau merepresentasikan seluruh muatan informasi yang dikandung oleh basis data, berikut merupakan *Conceptual Data Model* dari sistem usulan:



Gambar V.21 CDM Sistem Informasi Penjadwalan Produksi

(Sumber: Hasil Analisis, 2018)

5.8.3 Kamus Data

Kamus menggambarkan sebuah katalog data yang terdapat didalam sistem dengan maksud untuk mendefinisikan data yang mengalir di dalam sistem dengan lengkap sehingga user dan analis sistem mempunyai pengertian yang sama tentang input, output, dan komponen data store. Pada tahap perancangan sistem, kamus data

digunakan untuk merancang input, laporan dan basis data. Berikut adalah kamus data sistem informasi penjadwalan produksi:

1. Spesifikasi tabel *login*

- a. Nama tabel : *user*
- b. Fungsi : Untuk menyimpan data login pengguna
- c. Tipe : File data *master*

Tabel V.9 Tabel Kamus Data *user*

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1.	id_user	Id user	Integer	11	<i>Primary Key</i>
2.	<i>Username</i>	<i>username</i>	Varchar	50	
2.	<i>Password</i>	<i>password</i>	Varchar	50	
3	Nama_user	Nama	Varchar	50	
4	Tipe	Tipe	Int	2	

(Sumber: Hasil Analisis, 2018)

2. Spesifikasi tabel data customer

- a. Nama tabel : data_customer
- b. Fungsi : Untuk menyimpan data *customer*
- c. Tipe : File data *master*

Tabel V.10 Tabel Kamus Data data_customer

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1.	Id_customer	ID <i>Customer</i>	Varchar	8	<i>Primary Key</i>
2.	Nama_customer	Nama <i>Customer</i>	Varchar	50	
3.	Telepon	Telp	Varchar	13	
4	Email	Email	Varchar	50	
5	Alamat	Alamat	Varchar	50	

(Sumber: Hasil Analisis, 2018)

3. Spesifikasi tabel data produk

- a. Nama tabel : data_produk
- b. Fungsi : Untuk menyimpan data produk
- c. Tipe : File data *master*

Tabel V.11 Tabel Kamus Data data_produk

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1.	Id_produk	Id produk	Varchar	25	<i>Primary Key</i>
2.	Nama_produk	Nama produk	Varchar	50	
3.	Diameter	Diameter	Int	2	
4.	Stok	Stok	Int	5	
5.	Jenis	Jenis	Int	1	

(Sumber: Hasil Analisis, 2018)

4. Spesifikasi tabel data *purchase order*

- a. Nama tabel : purchase_order
- b. Fungsi : Untuk menyimpan data *Purchase Order*
- c. Tipe : File data *transaksi*

Tabel V.12 Tabel Kamus Data purchase_order

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1.	Id_po	Id_po	Int	11	<i>Primary Key</i>
2.	No_po	Nomor Po	Varchar	20	
3.	Tanggal	Tanggal	Date		
4.	Id_customer	Id customer	Varchar	8	<i>Foreign Key</i>

(Sumber: Hasil Analisis, 2018)

5. Spesifikasi tabel data *purchase order* detail

- a. Nama tabel : purchase_order_detail
- b. Fungsi : Untuk menyimpan data *Purchase Order* detail
- c. Tipe : File data transaksi

Tabel V.13 Tabel Kamus Data purchase_order_detail

No	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1.	Id_po_detail	Id po detail	Int	11	<i>Primary Key</i>
2.	Id_produk	Id Produk	Varchar	25	<i>Foreign Key</i>
3.	Qty	Qty	Int	5	
4.	Id_po	Id Po	Int	11	<i>Foreign Key</i>

(Sumber: Hasil Analisis, 2018)

6. Spesifikasi tabel data *purchase order* detail *delivery*

- a. Nama tabel : purchase_order_detail_delivery
- b. Fungsi : Untuk menyimpan data *Schedule Delivery*
- c. Tipe : File data transaksi

Tabel V.14 Tabel Kamus Data purchase_order_detail_delivery

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1.	Id_delivery	Id deliv	Int	11	<i>Primary Key</i>
2.	Id_po_detail	Id po detail	Int	11	<i>Foreign Key</i>
3.	Tanggal	Tanggal	Date		
4.	Qty_deliv	Qty deliv	Int	11	

(Sumber: Hasil Analisis, 2018)

7. Spesifikasi tabel data MPS

- a. Nama tabel : mps
- b. Fungsi : Untuk menyimpan data hasil MPS
- c. Tipe : File data transaksi

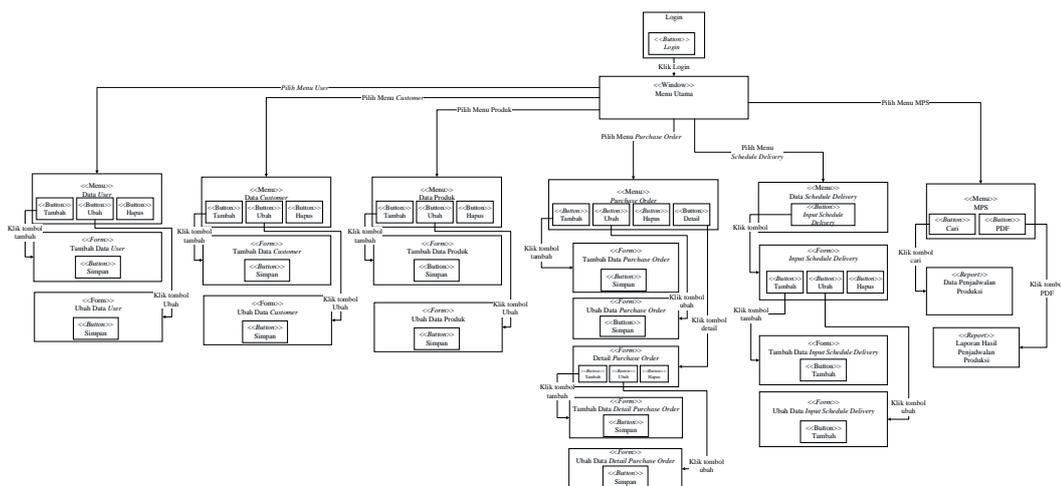
Tabel V.15 Tabel Kamus Data mps

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1.	Id_mps	Id mos	Int	11	<i>Primary Key</i>
2.	Jenis	Jenis	Int	1	
3.	Tgl_mps	Tgl mps	Date		
4.	Permintaan	Permintaan	Int	11	
5.	Persediaan_a wal	Persediaan awal	Int	11	
6.	Produksi	Produksi	Int	11	
7.	Persediaan_ak hir	Persediaan akhir	Int	11	
8.	Atp	Atp	Int	11	

(Sumber: Hasil Analisis, 2018)

5.9 Windows Navigation Diagram (WND)

Windows Navigation Diagram pada sistem usulan digunakan untuk menunjukkan bagaimana navigasi dari halaman-halaman yang terdapat di dalam aplikasi, berikut merupakan Windows Navigation Diagram sistem informasi penjadwalan produksi.



Gambar V.19 WND Sistem Informasi Penjadwalan Produksi

(Sumber: Hasil Analisis, 2018)

5.10 Desain Antar Muka

Menurut Wikipedia desain antarmuka pengguna (bahasa Inggris: *User Interface Design*) atau rekayasa antarmuka pengguna adalah desain untuk komputer, peralatan, mesin, perangkat komunikasi mobile, aplikasi perangkat lunak, dan situs web yang berfokus pada pengalaman pengguna (bahasa Inggris: *User Experience*) dan interaksi. Berikut adalah desain antar muka pada sistem informasi penjadwalan produksi:

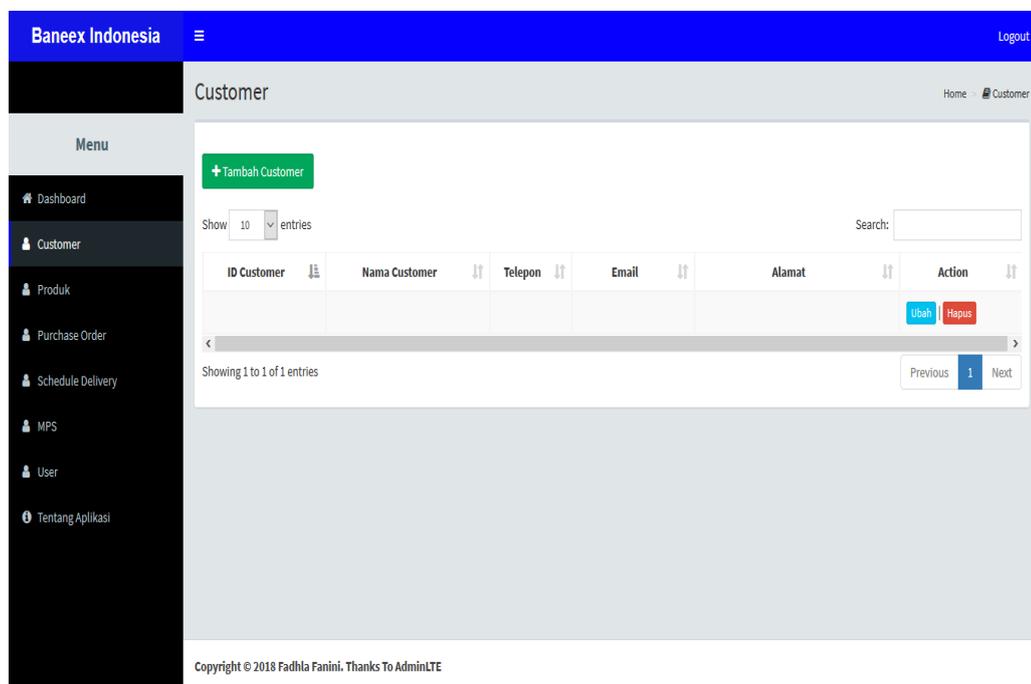
1. Login



Gambar V.22 Desain Antar Muka Login

(Sumber: Hasil Analisis, 2018)

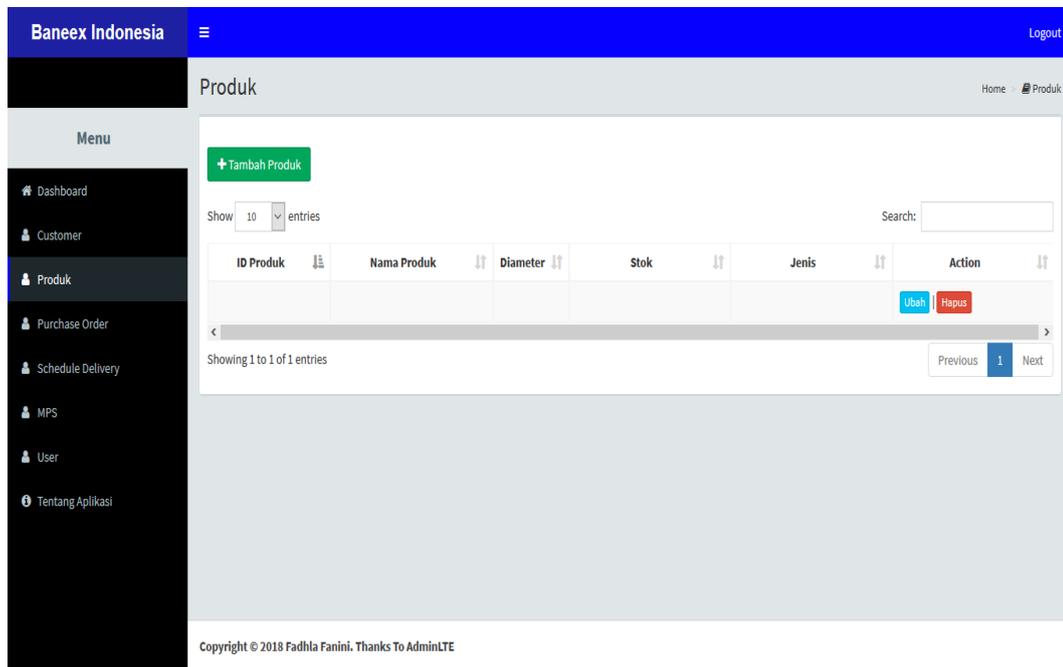
2. Data Customer



Gambar V.23 Desain Antar Muka Data Customer

(Sumber: Hasil Analisis, 2018)

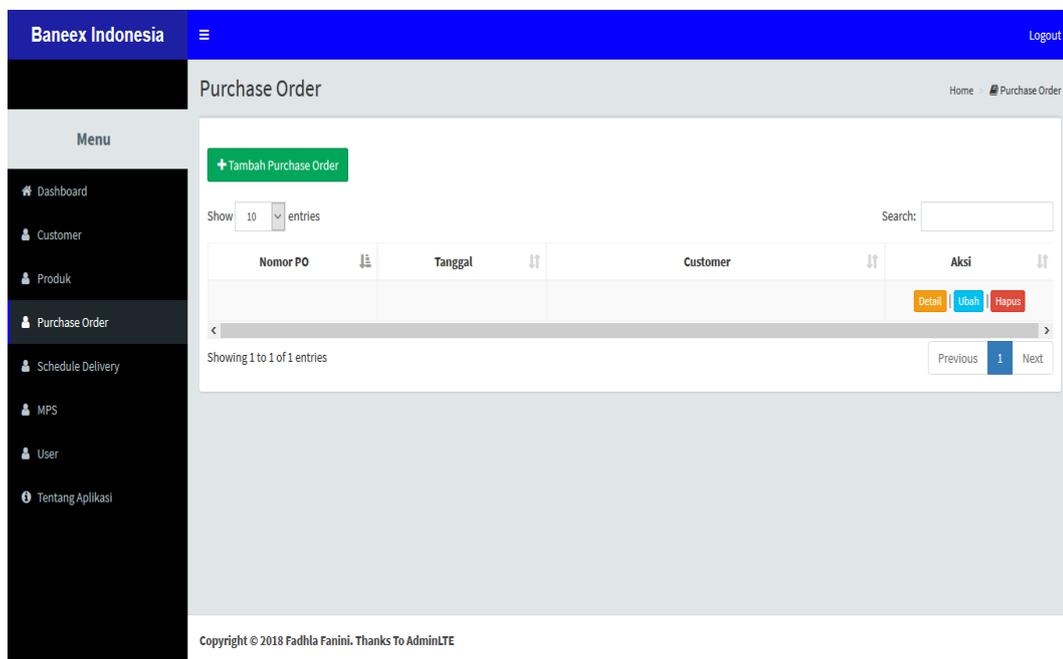
3. Data Produk



Gambar V.24 Desain Antar Muka Data Produk

(Sumber: Hasil Analisis, 2018)

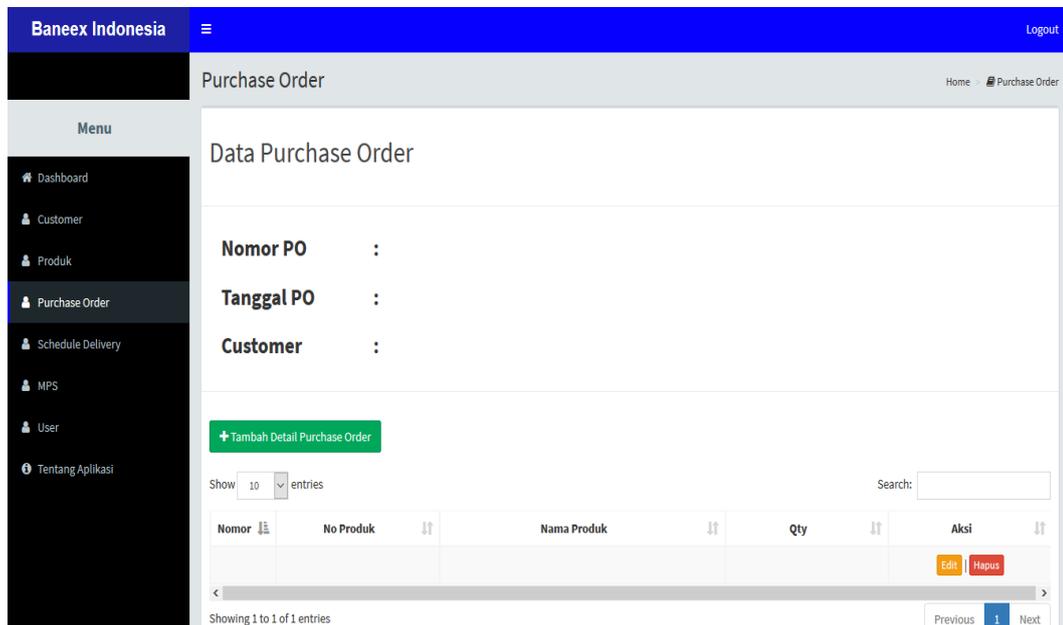
4. Data *Purchase Order*



Gambar V.25 Desain Antar Muka Data *Purchase Order*

(Sumber: Hasil Analisis, 2018)

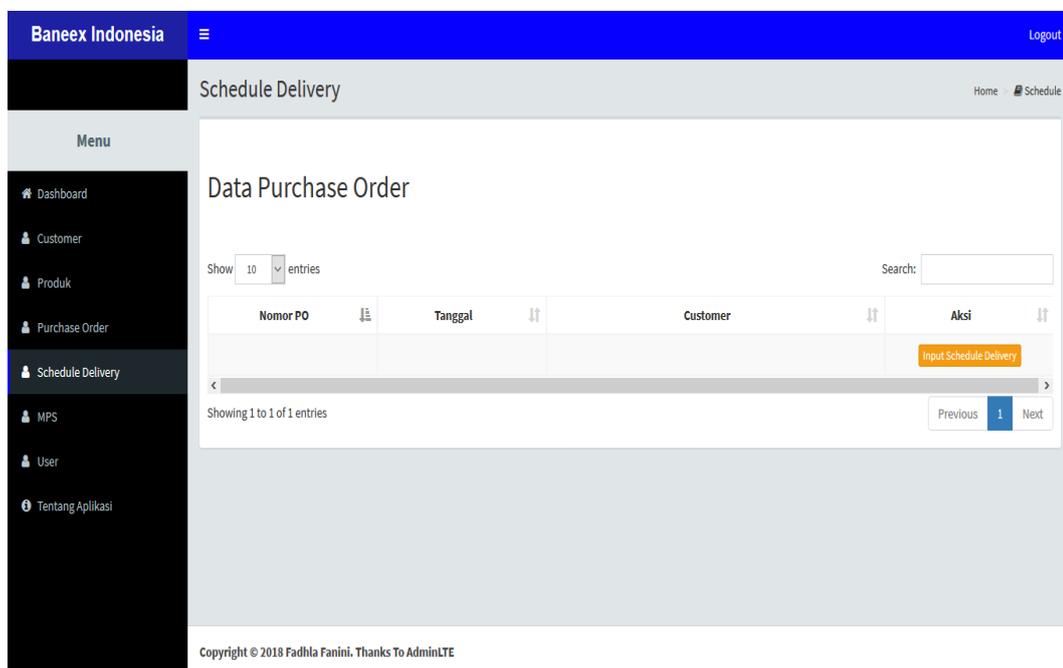
5. Data *Detail Purchase Order*



The screenshot shows the 'Purchase Order' page in the Baneex Indonesia application. The page has a blue header with the logo and a 'Logout' link. A dark sidebar on the left contains a 'Menu' with options like Dashboard, Customer, Produk, Purchase Order (selected), Schedule Delivery, MPS, User, and Tentang Aplikasi. The main content area is titled 'Purchase Order' and 'Data Purchase Order'. It features a form with fields for 'Nomor PO', 'Tanggal PO', and 'Customer', each followed by a colon. Below the form is a green button labeled '+ Tambah Detail Purchase Order'. Underneath is a search bar and a table with columns: 'Nomor', 'No Produk', 'Nama Produk', 'Qty', and 'Aksi'. The 'Aksi' column contains 'Edit' and 'Hapus' buttons. The table shows one entry. At the bottom, it says 'Showing 1 to 1 of 1 entries' and has 'Previous', '1', and 'Next' navigation buttons.

Gambar V.26 Desain Antar Muka Data *Detail Purchase Order*
(Sumber: Hasil Analisis, 2018)

6. Data *Schedule Delivery*



The screenshot shows the 'Schedule Delivery' page in the Baneex Indonesia application. The page has a blue header with the logo and a 'Logout' link. A dark sidebar on the left contains a 'Menu' with options like Dashboard, Customer, Produk, Purchase Order, Schedule Delivery (selected), MPS, User, and Tentang Aplikasi. The main content area is titled 'Schedule Delivery' and 'Data Purchase Order'. It features a search bar and a table with columns: 'Nomor PO', 'Tanggal', 'Customer', and 'Aksi'. The 'Aksi' column contains an 'Input Schedule Delivery' button. The table shows one entry. At the bottom, it says 'Showing 1 to 1 of 1 entries' and has 'Previous', '1', and 'Next' navigation buttons. The footer contains the text 'Copyright © 2018 Fadhla Fanini. Thanks To AdminLTE'.

Gambar V.27 Desain Antar Muka Data *Schedule Delivery*
(Sumber: Hasil Analisis, 2018)

7. Data Detail Schedule Delivery

The screenshot shows the 'Schedule Delivery' page in the Baneex Indonesia application. The page title is 'Schedule Delivery' and the breadcrumb is 'Home > Schedule Delivery > Detail'. The main content area is titled 'Data Purchase Order' and contains the following fields:

- Nomor PO :
- Tanggal PO :
- Customer :

Below the fields, there are additional fields:

- Nama Barang :
- Qty :
- Belum di Schedule :

A blue button labeled 'Tambah Schedule Delivery' is located below these fields. Below the button, there is a search bar and a table with the following columns: Nomor, Tanggal, Qty Pengiriman, and Aksi. The table contains one entry. The 'Aksi' column has 'Edit' and 'Hapus' buttons. Below the table, there is a pagination control showing 'Showing 1 to 1 of 1 entries' and 'Previous 1 Next'.

Gambar V.28 Desain Antar Muka Data *Detail Schedule Delivery*

(Sumber: Hasil Analisis, 2018)

8. Data User

The screenshot shows the 'User' page in the Baneex Indonesia application. The page title is 'User' and the breadcrumb is 'Home > User'. The main content area is titled 'Data User' and contains the following elements:

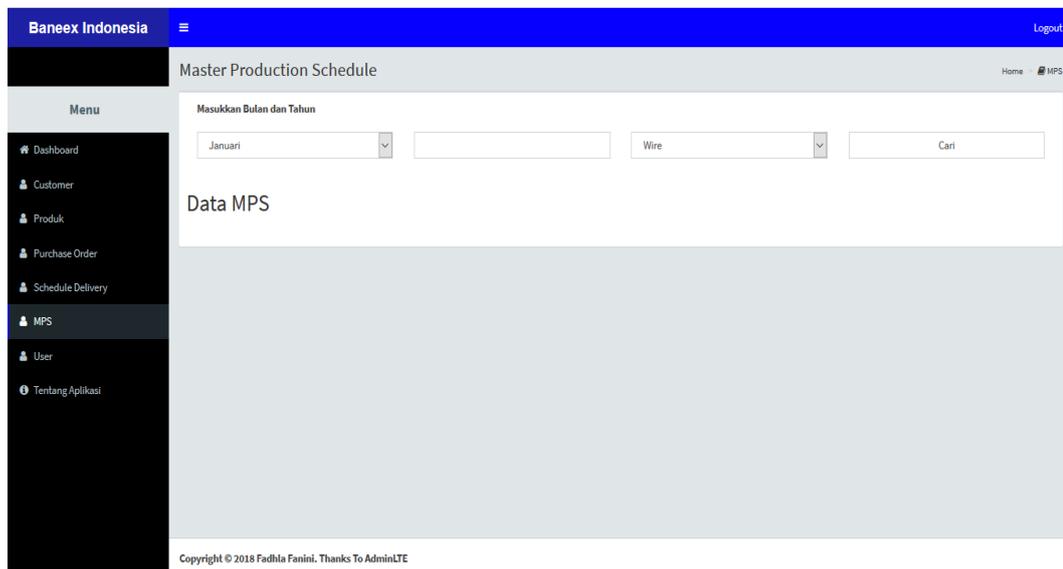
- A green button labeled '+ Tambah User'.
- A search bar.
- A table with the following columns: No, Username, Nama User, Tipe, and Aksi. The table contains one entry. The 'Aksi' column has 'Edit' and 'Hapus' buttons.
- A pagination control showing 'Showing 1 to 1 of 1 entries' and 'Previous 1 Next'.

At the bottom of the page, there is a copyright notice: 'Copyright © 2018 Fadhma Faniini. Thanks To AdminLTE'.

Gambar V.29 Desain Antar Muka Data *User*

(Sumber: Hasil Analisis, 2018)

9. Data MPS



Gambar V.30 Desain Antar Muka Data MPS

(Sumber: Hasil Analisis, 2018)

5.11 Black Box Testing

Untuk tahap pengujian penyusun menggunakan metode *black box*, untuk memastikan bahwa program atau sistem yang dibuat masih terdapat bug (kesalahan) atau tidak. Dari setiap tes pengujian tidak menutup kemungkinan masih terdapat bug (kesalahan) dari sistem yang telah dites, namun pengujian ini setidaknya dapat meminimalisir bug (kesalahan) yang terdapat di dalam sistem.

1. Black Box Testing Use Case Login

Berikut merupakan *Black Box Testing* dari *use case login*

Penguji : Indri Fitriana Alfia Nisa (1314042)

Tabel V.16 *Black Box Testing Use Case Login*

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Gagal Melakukan Login	Memasukan <i>username</i> dan <i>password</i> yang tidak sesuai	Tidak berhasil masuk ke dalam sistem	Tidak berhasil masuk ke dalam sistem	Valid
2	Berhasil Melakukan Login	Memasukan <i>username</i> dan <i>password</i> yang sesuai	Berhasil masuk ke dalam sistem	Berhasil masuk ke dalam sistem	Valid

(Sumber: Hasil Analisis, 2018)

2. *Black Box Testing Use Case Mengelola Data Customer*

Berikut merupakan *Black Box Testing* dari *use case* Mengelola Data *customer*

Penguji : Indri Fitriana Alfia Nisa (1314042)

Tabel V.17 *Black Box Testing Use Case Mengelola Data Customer*

No	Skenario Pengujian	<i>Test Case</i>	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Menambah data <i>customer</i>	Menambah data <i>customer</i> dengan mengisi semua kolom pada <i>form</i> tambah data.	Berhasil tambah data	Berhasil tambah data	Valid
2	Menambah data <i>customer</i>	Menambah data <i>customer</i> dengan mengosongkan salah satu kolom selain email pada <i>form</i> tambah data.	Menampilkan pesan “ <i>Please fill out this field</i> ”	Menampilkan pesan “ <i>Please fill out this field</i> ”	Valid
3	Mengubah data <i>customer</i>	Melakukan perubahan pada kolom telepon	Data <i>customer</i> berubah	Data <i>customer</i> berubah	Valid
4	Menghapus data <i>customer</i> yang dipilih	Menghapus salah satu data <i>customer</i>	Menghapus data <i>customer</i> dari tabel <i>customer</i>	Menghapus data <i>customer</i> dari tabel <i>customer</i>	Valid
5	Mencari data <i>customer</i>	Mencari data <i>customer</i> yang sesuai data yang ada di dalam <i>database</i>	Menampilkan data <i>customer</i> yang dicari	Menampilkan data <i>customer</i> yang dicari	Valid
6	Mencari data <i>customer</i>	Mencari data <i>customer</i> yang sesuai data yang ada tidak ada di dalam <i>database</i>	Menampilkan pesan “ <i>No matching records found</i> ”	Menampilkan pesan “ <i>No matching records found</i> ”	Valid

(Sumber: Hasil Analisis, 2018)

3. *Black Box Testing Use Case Mengelola Data Produk*

Berikut merupakan *Black Box Testing* dari *use case* mengelola data produk

Penguji : Indri Fitriana Alfia Nisa (1314042)

Tabel V.18 *Black Box Testing Use Case* Mengelola Data Produk

No	Skenario Pengujian	<i>Test Case</i>	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Menambah data produk	Menambah data produk dengan mengisi semua kolom pada <i>form</i> tambah data	Berhasil tambah data	Berhasil tambah data	Valid
2	Menambah data produk	Menambah data produk dengan mengosongkan salah satu kolom pada <i>form</i> tambah data	Menampilkan pesan “ <i>Please fill out this field</i> ”	Menampilkan pesan “ <i>Please fill out this field</i> ”	Valid
3	Mengubah data produk	Melakukan perubahan pada kolom stok	Data produk berubah	Data produk berubah	Valid
4	Menghapus data produk yang dipilih	Menghapus salah satu data produk	Menghapus data produk dari tabel produk	Menghapus data produk dari tabel produk	Valid
5	Mencari data produk	Mencari data produk yang sesuai data yang ada di dalam <i>database</i>	Menampilkan data produk yang dicari	Menampilkan data produk yang dicari	Valid
6	Mencari data produk	Mencari data produk yang sesuai data yang ada tidak ada di dalam <i>database</i>	Menampilkan pesan “ <i>No matching records found</i> ”	Menampilkan pesan “ <i>No matching records found</i> ”	Valid

(Sumber: Hasil Analisis, 2018)

4. *Black Box Testing Use Case* Mengelola Data User

Berikut merupakan *Black Box Testing* dari *use case* mengelola data user

Penguji : Indri Fitriana Alfia Nisa (1314042)

Tabel V.19 *Black Box Testing Use Case Mengelola Data User*

No	Skenario Pengujian	<i>Test Case</i>	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Menambah data <i>user</i>	Menambah data <i>user</i> dengan mengisi semua kolom pada <i>form</i> tambah data	Berhasil tambah data	Berhasil tambah data	Valid
2	Menambah data <i>user</i>	Menambah data <i>user</i> dengan mengosongkan salah satu kolom pada <i>form</i> tambah data	Menampilkan pesan “ <i>Please fill out this field</i> ”	Menampilkan pesan “ <i>Please fill out this field</i> ”	Valid
3	Menambah data <i>user</i>	Menambah data <i>user</i> dengan <i>username</i> yang telah ada	Menampilkan pesan “Maaf Username Sudah Ada”	Menampilakn pesan “Maaf Username Sudah Ada”	Valid
4	Mengubah data <i>user</i>	Melakukan perubahan pada kolom Nama <i>User</i>	Data <i>user</i> berubah	Data <i>user</i> berubah	Valid
5	Mengubah data <i>user</i>	Melakukan perubahan kolom <i>username</i> dengan <i>username</i> yang telah ada	Menampilkan pesan “Maaf Username Sudah Ada”	Manampilkan pesan “Maaf Username Sudah Ada”	Valid
6	Menghapus data <i>user</i> yang dipilih	Menghapus salah satu data <i>user</i> yang dipilih	Menghapus data <i>user</i> dari tabel <i>user</i>	Menghapus data <i>user</i> dari tabel <i>user</i>	Valid
7	Mencari data <i>user</i>	Mencari data <i>user</i> yang sesuai data yang ada di dalam <i>database</i>	Menampilkan data <i>user</i> yang dicari	Menampilkan data <i>user</i> yang dicari	Valid

(Sumber: Hasil Analisis, 2018)

5. *Black Box Testing Use Case Data Purchase Order*

Berikut merupakan *Black Box Testing* dari *use case* Mengelola Data *purchase order*.

Penguji : Indri Fitriana Alfia Nisa (1314042)

Tabel V.20 *Black Box Testing Use Case Data Purchase Order*

No	Skenario Pengujian	<i>Test Case</i>	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Menambah data <i>purchase order</i>	Menambah data <i>purchase order</i> dengan mengisi semua kolom pada <i>form</i> tambah data	Berhasil tambah data	Berhasil tambah data	Valid
2	Menambah data <i>purchase order</i>	Menambah data <i>purchase order</i> dengan mengosongkan salah satu kolom pada <i>form</i> tambah data	Menampilkan pesan “ <i>Please fill out this field</i> ”	Menampilkan pesan “ <i>Please fill out this field</i> ”	Valid
3	Mengubah data <i>purchase order</i>	Melakukan perubahan pada kolom Tanggal	Data <i>purchase order</i> berubah	Data <i>purchase order</i> berubah	Valid
4	Menghapus data <i>purchase order</i> yang dipilih	Menghapus salah satu data <i>purchase order</i> yang dipilih	Menghapus data <i>purchase order</i> dari tabel <i>purchase order</i>	Menghapus data <i>purchase order</i> dari tabel <i>purchase order</i>	Valid
5	Mencari data <i>purchase order</i>	Mencari data <i>purchase order</i> yang sesuai data yang ada di dalam <i>database</i>	Menampilkan data <i>purchase order</i> yang dicari	Menampilkan data <i>purchase order</i> yang dicari	Valid
6	Menampilkan <i>detail purchase order</i> yang dipilih	Menekan tombol <i>detail</i> pada <i>purchase order</i> yang dipilih	Menampilkan data	Berhasil tambah data	Valid

Tabel V.20 *Black Box Testing Use Case Data Purchase Order* (Lanjutan)

No	Skenario Pengujian	<i>Test Case</i>	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
7	Menambah data <i>detail purchase order</i>	Menambah data <i>detail purchase order</i> dengan mengisi semua kolom pada <i>form</i> tambah data	Berhasil tambah data	Berhasil tambah data	Valid
8	Menambah data <i>detail purchase order</i>	Menambah data <i>detail purchase order</i> dengan mengosongkan salah satu kolom pada <i>form</i> tambah data	Menampilkan pesan “ <i>Please fill out this field</i> ”	Menampilkan pesan “ <i>Please fill out this field</i> ”	Valid
9	Mengubah data <i>detail purchase order</i>	Melakukan perubahan pada kolom Nama Qty	Data <i>detail purchase order</i> berubah	Data <i>detail purchase order</i> berubah	Valid
10	Menghapus data <i>detail purchase order</i> yang dipilih	Menghapus salah satu data <i>detail purchase order</i> yang dipilih	Menghapus data <i>detail purchase order</i> dari tabel <i>detail purchase order</i>	Menghapus data <i>detail purchase order</i> dari tabel <i>detail purchase order</i>	Valid
11	Mencari data <i>detail purchase order</i>	Mencari data <i>detail purchase order</i> yang sesuai data yang ada di dalam <i>database</i>	Menampilkan data <i>detail purchase order</i> yang dicari	Menampilkan data <i>detail purchase order</i> yang dicari	Valid

(Sumber: Hasil Analisis, 2018)

6. *Black Box Testing Use Case* Mengelola Data *Schedule Delivery*

Berikut merupakan *Black Box Testing use case* mengelola data *schedule delivery*.

Penguji : Indri Fitriana Alfia Nisa (1314042)

Tabel V.21 *Black Box Testing Use Case* Mengelola Data *Schedule Delivery*

No	Skenario Pengujian	<i>Test Case</i>	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Menambah data <i>schedule delivery</i>	Menambah data <i>schedule delivery</i> dengan mengisi semua kolom pada <i>form</i> tambah data	Berhasil tambah data	Berhasil tambah data	Valid
2	Menambah data <i>schedule delivery</i>	Menambah data <i>schedule delivery</i> dengan mengosongkan salah satu kolom pada <i>form</i> tambah data.	Menampilkan pesan “ <i>Please fill out this field</i> ”	Menampilkan pesan “ <i>Please fill out this field</i> ”	Valid
3	Mengubah data <i>schedule delivery</i>	Melakukan perubahan pada kolom tanggal	Data <i>schedule delivery</i> berubah	Data <i>schedule delivery</i> berubah	Valid
4	Menghapus data <i>schedule delivery</i> yang dipilih	Menghapus salah satu data <i>schedule delivery</i> yang dipilih	Menghapus data <i>schedule delivery</i> dari tabel <i>schedule delivery</i>	Menghapus data <i>schedule delivery</i> dari tabel <i>schedule delivery</i>	Valid
5	Mencari data <i>schedule delivery</i>	Mencari data <i>schedule delivery</i> yang sesuai data yang ada di dalam <i>database</i>	Menampilkan data <i>schedule delivery</i> yang dicari	Menampilkan data <i>schedule delivery</i> yang dicari	Valid

(Sumber: Hasil Analisis, 2018)

7. *Black Box Testing Use Case* Pembuatan MPS

Berikut merupakan *Black Box Testing use case* pembuatan MPS

Penguji : Indri Fitriana Alfia Nisa (1314042)

Tabel V.22 *Black Box Testing Use Case* Pembuatan MPS

No	Skenario Pengujian	<i>Test Case</i>	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Memproses MPS	Memproses MPS dengan mengisi semua kolom	Menampilkan data MPS	Menampilkan data MPS	Valid
2	Memproses MPS	Memproses MPS dengan mengosongkan kolom tahun	Tidak menampilkan data MPS	Tidak menampilkan data MPS	Valid

(Sumber: Hasil Analisis, 2018)

5.12 Spesifikasi Pengembangan Sistem

Untuk mengimplementasi sistem informasi penjadwalan produksi ini dibutuhkan minimum spesifikasi untuk menjalankan sistem informasi penjadwalan produksi diantaranya:

1. Spesifikasi Kebutuhan *Hardware*

Kebutuhan perangkat keras atau *hardware* dari sistem ini meliputi:

- a. Untuk *server*, berupa unit komputer yang diinstall dan dikonfigurasi sesuai standar minimal yaitu *Apache*, *PHP*, dan *MariaDB/MySQL*
- b. Untuk *client*, berupa komputer atau laptop yang terdapat aplikasi *web browser*

2. Spesifikasi Kebutuhan *Software*

Kebutuhan perangkat lunak atau *software* dari sistem ini meliputi:

- a. Sistem operasi *Windows 7*
- b. *XAMPP Control Panel V.3.2.2*
- c. *Framework Codeigniter*
- d. *Web browser* seperti *Mozilla Firefox* dan sejenisnya

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan pengumpulan, pengolahan, dan analisis data yang telah dilakukan pada bab-bab sebelumnya pada penelitian ini maka dapat disimpulkan beberapa poin kesimpulan sebagai berikut:

1. Sistem informasi penjadwalan produksi yang dibuat berdasarkan proses bisnis yang berjalan dan permasalahan yang ditemukan di PT Baneex Indonesia ini diharapkan dapat melakukan penjadwalan produksi dengan metode *Master Production Schedule (MPS)* pada divisi *production planning and control (PPC)* dan diharapkan dapat menunjang proses bisnis perusahaan dan menyelesaikan permasalahan yang ada pada proses penjadwalan produksi pada sistem yang berjalan.
2. Sistem informasi penjadwalan produksi yang mempunyai penyimpanan dalam satu lokasi diharapkan dapat mempercepat proses pencarian data serta dapat mengurangi resiko data rusak atau data hilang.

6.2 Saran

Saran yang dapat diberikan penulis untuk pengembangan dan penerapan sistem informasi penjadwalan produksi selanjutnya adalah sebagai berikut:

1. Melakukan sosialisasi untuk penggunaan aplikasi secara intensif kepada divisi-divisi terkait agar penggunaan aplikasi ini dapat berjalan dengan baik dan membantu jalannya proses penjadwalan produksi.
2. Melakukan pengembangan aplikasi lebih lanjut seperti pengembangan pengelolaan laporan produksi harian.

DAFTAR PUSTAKA

- Anggraeni, E. Y., & Irviani, R. (2017). *Pengantar Sistem Informasi*. Yogyakarta: ANDI.
- Ariawan. 2009. *Buku Ajar Sistem Informasi Manajemen*. Gorontalo: Universitas Ichsan STMIK Ichsan.
- Assauri, Sofjan. 2008. *Manajemen Produksi dan Operasi*. Jakarta : Fakultas Ekonomi Universitas Indonesia.
- Binarto, Suryo, 2012. *Trik dan Tips Membuat program Aplikasi Penjualan Menggunakan Visual basis 6.0*. Jakarta : PT.Transmedia.
- Dennis, Alan., Wixom, Barbara Haley., dan Tegarden, David. 2010. *System Analysis & Design. An Object-Oriented Approach with UML*. USA: Wiley.
- Dennis, Alan., Wixom, Barbara Haley., dan Tegarden, David. 2015. *System Analysis & Design. An Object-Oriented Approach with UML (Fifth Edition)*. USA: Wiley.
- Djahir, Yulia., dan Pratita, Dewi. 2014. *Sistem Informasi Manajemen*. Yogyakarta: Deepublish.
- Enterprise Jubilee. 2015. *Membuat Website PHP dengan Codeigniter*. Jakarta : Elex Media Komputindo.
- Fauzi, R. A. (2017). *Sistem informasi akuntansi (berbasis akuntansi)*. Yogyakarta: Deepublish.
- Gaspersz, Vincent. 2004. *Production Planning and Inventory Control*. Jakarta: Gramedia Pustaka.
- H.M, Jogyanto. 2010. *Analisis dan Desain Sistem Informasi*. Yogyakarta: Andi Offset.
- Heizer, Jay & Barry Render, 2008. *Manajemen Operasi (Buku 1 Edisi 9)*. Jakarta: Salemba Empat.

- Hoffer, Jeffrey A., Prescott Mary B., Mcfadden, Fred R. 2010. *Modern Database Management*. New Jersey: Pearson Prentice Hall.
- Jalinus N dan Ambiyar. 2016. *Media dan Sumber Pembelajaran*. Jakarta: Penerbit Kencana
- McLeod, Raymond Jordan George P. Schell. 2011. *Sistem Informasi Manajemen*. Jakarta: Penerbit Salemba Empat.
- Muslihudin, Muhammad., dan Oktafianto. 2016. *Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML*. Yogyakarta: Andi.
- Nasution, Arman Hakim. 2008. *Perencanaan dan Pengendalian Produksi*. Yogyakarta: Graha Ilmu.
- Pressman, Roger S. 2010. *Software Engineering : a practitioner's approach 7th ed.* USA: New York.
- Puspitawati, Lilis., dan Anggadini, Sri Dewi. 2011. *Sistem Informasi Akuntansi*. Yogyakarta: Graha Ilmu.
- Robbins, Stephen P.; Judge, Timothy A. (2008). *Perilaku Organisasi Buku 2*, Jakarta: Salemba Empat.
- Rosa, A.S, dan Shalahuddin, M. 2015. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.
- Rosnani, Ginting. (2009). *Penjadwalan Mesin. Edisi Pertama*. Yogyakarta: Graha Ilmu.
- Rusdiana., dan Irfan, Moch. 2014. *Sistem Informasi Manajemen*. Bandung: Pustaka Setia.
- Supono, dan Viridiandry Putratama. 2016. *Pemrograman Web Dengan Menggunakan PHP dan Framework Codeigniter*. Yogyakarta: Deepublish.
- Sutabri, Tata. 2012. *Konsep Sistem Informasi*. Yogyakarta: Andi.
- Sutaji, Deni, 2012, *Sistem Inventory Mini Market dengan PHP & JQuery*, Yogyakarta: Lokomedia

Yakub.2012.*Pengantar Sistem Informasi*. Yogyakarta: Graha Ilmu

Situs:

Wikipedia, *Desain Antar Muka*,

https://id.wikipedia.org/wiki/Desain_antarmuka_pengguna, (Tanggal Akses:
6 Agustus 2018)

Jurnal Ilmiah:

Hartono, Haryanti. 2014. *Aplikasi Penjadwalan Produksi Manufaktur Berbasis Web Dengan Metode Master Production Schedule*. Jurnal Teknologi Informasi, Vol 00, Nomor 0.

Reynaldo, Sudirwa, Soebandrija. 2011. *Perencanaan Dan Optimasi Produk Dan Penyediaan Pada PT. Goodyear Indonesia*.

Supriyadi, Riskiyadi. 2016. *Penjadwalan Produksi IKS-FILLER Pada Proses Ground Calcium Carbonate Menggunakan Metode MPS Di Perusahaan Kertas*. Jurnal SINERGI Vol 20, No 2 Hal 157-164

Segerstedt, A. (2006). *International Journal of Production Research*,44(18-19),3585-3606.

LAMPIRAN

1. Source Code

a. Customer.php

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Customer extends CI_Controller {

    function __construct(){
        parent::__construct();
        /*
            if($this->session->userdata('status') != "login"){
                redirect(base_url("login"));
            }
        */
        if($this->session->userdata('status') != "login"){
            redirect(base_url("login"));
        }
        $this->load->model('m_perencanaan');
    }

    public function index()
    {
        $this->load->view('home/header');
        $y['tipe'] = $this->session->userdata('tipe');
        $y['tab'] = '2';
        $this->load->view('home/sidebar',$y);
        $x['data']=$this->m_perencanaan->show_customer();
        $this->load->view('konten_customer',$x);
        $this->load->view('home/footer');
    }

    public function tambah_customer()
    {
        $sid_customer = $this->input->post('id_customer');
        $nama_customer = $this->input->post('nama_customer');
        $alamat = $this->input->post('alamat');
        $telepon = $this->input->post('telepon');
        $email = $this->input->post('email');

        $data = array(
            'id_customer' => $sid_customer,
```

```

        'nama_customer' => $nama_customer,
        'alamat' => $alamat,
        'telepon' => $telepon,
        'email' => $email
    );
    //cek dulu
    $query = $this->db->query("select * from data_customer where
id_customer = '$id_customer'")->num_rows();
    if($query > 0){
        $this->session->set_flashdata('pesan', '<div class="alert
alert-danger alert-dismissible">
        <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
        <strong>Maaf</strong> Data Sudah Ada.
        </div>');
    }else{
        $this->m_perencanaan-
>tambah_customer($data,'data_customer');
        $this->session->set_flashdata('pesan', '<div class="alert
alert-success alert-dismissible">
        <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
        <strong>Berhasil</strong> Menambah customer.
        </div>');
    }
    redirect('customer');
}

public function ubah_customer($id_customer)
{
    $this->load->view('home/header');
    $y['tipe'] = $this->session->userdata('tipe');
    $y['tab'] = '2';
    $this->load->view('home/sidebar',$y);
    $where = array('id_customer' => $id_customer);
    $data['user'] = $this->m_perencanaan-
>tampil_customer_by_id($where,'data_customer')->result();
    $this->load->view('konten_customer_ubah',$data);
    $this->load->view('home/footer');
}

public function hapus_customer($id_customer)
{
    $this->load->view('home/header');
    $y['tipe'] = $this->session->userdata('tipe');
    $y['tab'] = '2';

```

```

        $this->load->view('home/sidebar',$y);
        $where = array('id_customer' => $id_customer);
        $data['user'] = $this->m_perencanaan-
>tampil_customer_by_id($where,'data_customer')->result();
        $this->load->view('konten_customer_hapus',$data);
        $this->load->view('home/footer');
    }

    public function ubah_customer_save()
    {
        $id_customer = $this->input->post('id_customer');
        $nama_customer = $this->input->post('nama_customer');
        $alamat = $this->input->post('alamat');
        $telepon = $this->input->post('telepon');
        $email = $this->input->post('email');

        $data = array(
            'nama_customer' => $nama_customer,
            'alamat' => $alamat,
            'telepon' => $telepon,
            'email' => $email
        );

        $where = array(
            'id_customer' => $id_customer
        );

        $this->m_perencanaan-
>ubah_customer($where,$data,'data_customer');
        $this->session->set_flashdata('pesan', '<div class="alert alert-
success alert-dismissible">
        <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
        Data Customer Berhasil Di Ubah.
        </div>');
        redirect('customer');
    }

    public function hapus_customer_save()
    {
        $id_customer = $this->input->post('id_customer');
        $where = array(
            'id_customer' => $id_customer
        );

        $this->m_perencanaan->hapus_customer($where,'data_customer');
    }

```

```

        $this->session->set_flashdata('pesan', '<div class="alert alert-
warning alert-dismissible">
        <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
        Data Customer Berhasil Di Hapus.
        </div>');
        redirect('customer');
    }
}

```

b. Login.php
<?php

```

class Login extends CI_Controller{

    function __construct(){
        parent::__construct();
        $this->load->model('m_perencanaan');

    }

    function index(){
        $data['error'] = $this->uri->segment(3);
        if(isset($data['error'])){
            $this->session->set_flashdata('pesan', '<div class="alert
alert-danger alert-dismissible">
            <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
            <center>Maaf, username atau password salah.</center>
            </div>');
        }
        $this->load->view('login',$data);
    }

    function aksi_login(){
        $username = $this->input->post('username');
        $password = $this->input->post('password');
        $where = array(
            'username' => $username,
            'password' => md5($password)
        );
        $cek = $this->m_perencanaan->cek_login("user",$where)-
>num_rows();
    }
}

```

```

$cek1 = $this->m_perencanaan->cek_login("user",$where)-
>row();
if($cek > 0){

    $data_session = array(
        'nama' => $cek1->nama_user,
        'tipe' => $cek1->tipe,
        'status' => "login"
    );

    $this->session->set_userdata($data_session);

    redirect(base_url("home"));

}else{
    redirect(base_url("login/index/1"));
}

}

function logout(){
    $this->session->sess_destroy();
    redirect(base_url('login'));
}
}

```

c. Mps.php

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

```

```

class Mps extends CI_Controller {

    function __construct(){
        parent::__construct();
        /*
        if($this->session->userdata('status') != "login"){
            redirect(base_url("login"));
        }
        */
        if($this->session->userdata('status') != "login"){
            redirect(base_url("login"));
        }
        $this->load->model('m_perencanaan');
    }

    public function index()
    {
        $this->load->view("home/header");
    }
}

```

```

        $y['tab'] = '7';
        $y['tipe'] = $this->session->userdata('tipe');
        $this->load->view('home/sidebar',$y);
        $x['data']=$this->m_perencanaan->show_po();
        $this->load->view('mps/mps',$x);
        $this->load->view('home/footer');
    }

```

```

    }
d. Produk.php
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

```

```

class Produk extends CI_Controller {

```

```

    function __construct(){
        parent::__construct();
        /*
        if($this->session->userdata('status') != "login"){
            redirect(base_url("login"));
        }
        */
        if($this->session->userdata('status') != "login"){
            redirect(base_url("login"));
        }
        $this->load->model('m_perencanaan');
    }

```

```

    public function index()
    {
        $this->load->view('home/header');
        $y['tab'] = '3';
        $y['tipe'] = $this->session->userdata('tipe');
        $this->load->view('home/sidebar',$y);
        $x['data']=$this->m_perencanaan->show_produk();
        $this->load->view('data_produk/produk',$x);
        $this->load->view('home/footer');
    }

```

```

    public function tambah_produk()
    {
        $id_produk = $this->input->post('id_produk');
        $nama_produk = $this->input->post('nama_produk');
        $diameter = $this->input->post('diameter');
        $stok = $this->input->post('stok');
    }

```

```

$jenis = $this->input->post('jenis');

$data = array(
    'id_produk' => $id_produk,
    'nama_produk' => $nama_produk,
    'diameter' => $diameter,
    'jenis' => $jenis,
    'stok' => $stok
);
//cek dulu
$query = $this->db->query("select * from data_produk where
id_produk = '$id_produk'")->num_rows();
if($query > 0){
    $this->session->set_flashdata('pesan', '<div class="alert
alert-danger alert-dismissible">
    <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
    <strong>Maaf</strong> Data Sudah Ada.
</div>');
} else {
    $this->m_perencanaan-
>tambah_produk($data,'data_produk');
    $this->session->set_flashdata('pesan', '<div class="alert
alert-success alert-dismissible">
    <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
    <strong>Berhasil</strong> Menambah produk.
</div>');
}
redirect('produk');
}

public function ubah_produk($id_produk)
{
    $this->load->view('home/header');
    $y['tab'] = '3';
    $y['tipe'] = $this->session->userdata('tipe');
    $this->load->view('home/sidebar',$y);
    $where = array('id_produk' => $id_produk);
    $data['produk'] = $this->m_perencanaan-
>tampil_produk_by_id($where,'data_produk')->result();
    $this->load->view('data_produk/produk_ubah',$data);
    $this->load->view('home/footer');
}

```

```

public function ubah_produk_save()
{
    $id_produk = $this->input->post('id_produk');
    $nama_produk = $this->input->post('nama_produk');
    $diameter = $this->input->post('diameter');
    $stok = $this->input->post('stok');
    $jenis = $this->input->post('jenis');

    $data = array(
        'nama_produk' => $nama_produk,
        'diameter' => $diameter,
        'jenis' => $jenis,
        'stok' => $stok
    );

    $where = array(
        'id_produk' => $id_produk
    );

    $this->m_perencanaan-
>ubah_produk($where,$data,'data_produk');
    $this->session->set_flashdata('pesan', '<div class="alert alert-
success alert-dismissible">
        <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
        Data Produk Berhasil Di Ubah.
    </div>');
    redirect('produk');
}

public function hapus_produk($id_produk)
{
    $this->load->view('home/header');
    $y['tab'] = '3';
    $y['tipe'] = $this->session->userdata('tipe');
    $this->load->view('home/sidebar',$y);
    $where = array('id_produk' => $id_produk);
    $data['produk'] = $this->m_perencanaan-
>tampil_produk_by_id($where,'data_produk')->result();
    $this->load->view('data_produk/produk_hapus',$data);
    $this->load->view('home/footer');
}

public function hapus_produk_save()
{
    $id_produk = $this->input->post('id_produk');

```

```

        $where = array(
            'id_produk' => $id_produk
        );

        $this->m_perencanaan->hapus_produk($where,'data_produk');
        $this->session->set_flashdata('pesan', '<div class="alert alert-
warning alert-dismissible">
        <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
        Data Produk Berhasil Di Hapus.
        </div>');
        redirect('produk');
    }
}

```

e. User.php

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

```

```

class User extends CI_Controller {

```

```

    function __construct(){
        parent::__construct();
        /*
        if($this->session->userdata('status') != "login"){
            redirect(base_url("login"));
        }
        */
        if($this->session->userdata('status') != "login"){
            redirect(base_url("login"));
        }
        $this->load->model('m_perencanaan');
    }

    public function index()
    {
        $this->load->view('home/header');
        $y['tab'] = '8';
        $y['tipe'] = $this->session->userdata('tipe');
        $this->load->view('home/sidebar',$y);
        $x['detail_user']=$this->m_perencanaan->show_user();
        $this->load->view('user/user_detail',$x);
        $this->load->view('home/footer');
    }

    public function tambah_user()

```

```

{
    $username = $this->input->post('username');
    $password = $this->input->post('password');
    $nama_user = $this->input->post('nama_user');
    $tipe = $this->input->post('tipe');

    $data = array(
        'id_user' => "",
        'username' => $username,
        'password' => md5($password),
        'nama_user' => $nama_user,
        'tipe' => $tipe
    );
    //cek dulu
    $query = $this->db->query("select * from user where username =
'Susername'")->num_rows();
    if($query > 0){
        $this->session->set_flashdata('pesan', '<div class="alert
alert-danger alert-dismissible">
        <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
        <strong>Maaf</strong> Username Sudah Ada.
        </div>');
    }else{
        $this->m_perencanaan->tambah($data,'user');
        $this->session->set_flashdata('pesan', '<div class="alert
alert-success alert-dismissible">
        <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
        <strong>Berhasil</strong> Menambah customer.
        </div>');
    }
    redirect('user');
}

function search_user($id)
{
    $data = $this->db->query("SELECT * from user where
id_user='$id'")->row();

    echo json_encode($data);
}

public function ubah_user_save()
{
    $username = $this->input->post('username');

```

```

$username1 = $this->input->post('username1');
$password = $this->input->post('password');
$password1 = $this->input->post('password1');
$name_user = $this->input->post('nama_user');
$type = $this->input->post('tipe');
$id_user = $this->input->post('id_user');

if($password1 == $password){
    $data = array(
        'username' => $username,
        'nama_user' => $name_user,
        'tipe' => $type
    );
} else {
    $data = array(
        'username' => $username,
        'password' => md5($password),
        'nama_user' => $name_user,
        'tipe' => $type
    );
}

$where = array(
    'id_user' => $id_user
);

//cek dulu
$query = 0;
if($username <> $username1){
    $query = $this->db->query("select * from user where
username = '$username'")->num_rows();
}

if($query > 0){
    $this->session->set_flashdata('pesan', '<div class="alert
alert-danger alert-dismissible">
    <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
    <strong>Maaf</strong> Username Sudah Ada.
</div>');
} else {
    $this->m_perencanaan->ubah($where,$data,'user');
    $this->session->set_flashdata('pesan', '<div class="alert
alert-success alert-dismissible">
    <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>

```

```

        <strong>Berhasil</strong>Mengubah.
    </div>');
    }
    redirect('user');
}

public function delete_user($id)
{
    $where = array(
        'id_user' => $id
    );

    $this->m_perencanaan->hapus($where,'user');
    $this->session->set_flashdata('pesan', '<div class="alert alert-
warning alert-dismissible">
    <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
    Data User Berhasil Di Hapus.
    </div>');
    echo json_encode(array("status" => TRUE));
    //redirect('user');
}
}

```

f. Schedule_delivery.php

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Schedule_delivery extends CI_Controller {

    function __construct(){
        parent::__construct();
        /*
        if($this->session->userdata('status') != "login"){
            redirect(base_url("login"));
        }
        */
        if($this->session->userdata('status') != "login"){
            redirect(base_url("login"));
        }
        $this->load->model('m_perencanaan');
    }

    public function index()
    {
        $this->load->view('home/header');
    }
}

```

```

        $y['tab'] = '6';
        $y['tipe'] = $this->session->userdata('tipe');
        $this->load->view('home/sidebar',$y);
        $x['data']=$this->m_perencanaan->show_po();
        $this->load->view('schedule_delivery/schedule_delivery',$x);
        $this->load->view('home/footer');
    }

    public function detail_sd()
    {
        $this->load->helper('form_helper');
        $no_po = $this->uri->segment(3);
        $this->load->view('home/header');
        $y['tab'] = '6';
        $y['tipe'] = $this->session->userdata('tipe');
        $this->load->view('home/sidebar',$y);
        $x['data_po']=$this->m_perencanaan->cari_po($no_po);
        $x['detail_po']=$this->m_perencanaan->cari_detail_po($no_po);
        $x['dd_produk'] = $this->m_perencanaan->search_produk();
        $x['no_po'] = $no_po;
        $this->load->view('schedule_delivery/sd_detail',$x);
        $this->load->view('home/footer');
    }

    function search_sd_detail($id)
    {
        $query = $this->db->query("SELECT a.id_po_detil,b.id_produk,
        b.nama_produk as nama_produk, a.qty as qty FROM purchase_order_detil a,
        data_produk b where a.id_produk = b.id_produk and a.id_po_detil='$id')-
        >row();
        $data['qty'] = $query->qty;
        $data['nama_produk'] = $query->nama_produk;
        $query = $this->db->query("Select (CASE WHEN qty_deliv IS
        NOT NULL THEN sum(qty_deliv) ELSE 0 END) as jumlah
        from
        purchase_order_detil_delivery where id_po_detil='$id')->row();
        $data['sisal'] = $data['qty'] - $query->jumlah ;
        $data['id_po_detil'] = $id ;
        echo json_encode($data);
    }

    function search_sd_detail_edit($id,$id2)
    {

```

```

        $query = $this->db->query("SELECT a.id_po_detil,b.id_produk,
b.nama_produk as nama_produk, a.qty as qty FROM purchase_order_detil a,
data_produk b where a.id_produk = b.id_produk and a.id_po_detil='$id2'")-
>row();
        $data['qty'] = $query->qty;
        $data['nama_produk'] = $query->nama_produk;
        $query = $this->db->query("Select (CASE WHEN qty_deliv IS
NOT NULL THEN sum(qty_deliv) ELSE 0 END) as jumlah
                                from
purchase_order_detil_delivery where id_po_detil='$id2'")->row();
        $data['sisal'] = $data['qty'] - $query->jumlah ;
        $query = $this->db->query("Select id_delivery,qty_deliv,tanggal
from purchase_order_detil_delivery where id_delivery='$id'")->row();
        $data['qty_deliv'] = $query->qty_deliv;
        $data['tanggal'] = $query->tanggal;
        $data['id_delivery'] = $query->id_delivery;
        echo json_encode($data);
    }

public function tambah_schedule()
{
    $id_detil_po = $this->input->post('id_po_detil');
    $tanggal = $this->input->post('tanggal');
    $qty_deliv = $this->input->post('qty_deliv');
    $nama_produk = $this->input->post('nama_produk');
    $sisal = $this->input->post('sisal');
    $no_po = $this->input->post('no_po');

    $data = array(
        'id_delivery' => "",
        'id_po_detil' => $id_detil_po,
        'tanggal' => $tanggal,
        'qty_deliv' => $qty_deliv
    );
    //cek dulu
    $query = $this->db->query("select * from
purchase_order_detil_delivery where tanggal = '$tanggal' and id_po_detil=
'$id_detil_po' ")>num_rows();
    if($query > 0){
        $this->session->set_flashdata('pesan', '<div class="alert
alert-danger alert-dismissible">
        <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>

```

```

                <strong>Maaf</strong> Pada tanggal <b>'.$tanggal.'</b> Data
Produk <b>'.$nama_produk.'</b> Sudah Dijadwalkan. Mohon diperiksa
kembali
                </div>');
                redirect('schedule_delivery/detail_sd/'.$sno_po);
            }
            if($qty_deliv > $sisasisa){
                $this->session->set_flashdata('pesan', '<div class="alert
alert-danger alert-dismissible">
                <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
                <strong>Maaf</strong> Data Pengiriman Produk
<b>'.$nama_produk.'</b> Melebihi sisa pengiriman. Mohon diperiksa
kembali
                </div>');
                redirect('schedule_delivery/detail_sd/'.$sno_po);
            }else{
                $this->m_perencanaan-
>tambah_po($data,'purchase_order_detil_delivery');
                $this->session->set_flashdata('pesan', '<div class="alert
alert-success alert-dismissible">
                <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
                <strong>Berhasil</strong> Menambah pengiriman untuk produk
<b>'.$nama_produk.'</b> pada tanggal <b>'.$tanggal.'</b>.
                </div>');
                redirect('schedule_delivery/detail_sd/'.$sno_po);
            }
        }
    }

    public function edit_schedule()
    {

        $sid_detil_po = $this->input->post('id_po_detil');
        $tanggal = $this->input->post('tanggal');
        $qty_deliv = $this->input->post('qty_deliv');
        $nama_produk = $this->input->post('nama_produk');
        $sisasisa = $this->input->post('sisasisa');
        $sno_po = $this->input->post('no_po');
        $qty_deliv_lama = $this->input->post('qty_deliv_lama');
        $sid_delivery = $this->input->post('id_delivery');

        $data = array(
            'tanggal' => $tanggal,
            'qty_deliv' => $qty_deliv
        );
    }

```

```

        $where = array(
            'id_delivery' => $id_delivery
        );
        //cek dulu
        if(($qty_deliv - $qty_deliv_lama) > $sisa){
            $this->session->set_flashdata('pesan', '<div class="alert
alert-danger alert-dismissible">
            <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
            <strong>Maaf</strong> Data Pengiriman Produk
<b>'. $nama_produk.'</b> Melebihi sisa pengiriman. Mohon diperiksa
kembali
            </div>');
            redirect('schedule_delivery/detail_sd/'. $no_po);
        }else{
            $this->session->set_flashdata('pesan', '<div class="alert
alert-success alert-dismissible">
            <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
            <strong>Berhasil</strong> Mengubah pengiriman untuk produk
<b>'. $nama_produk.'</b> pada tanggal <b>'. $tanggal.'</b>.
            </div>');

            $this->m_perencanaan-
>ubah_customer($where,$data,'purchase_order_detil_delivery');
            redirect('schedule_delivery/detail_sd/'. $no_po);
        }
    }

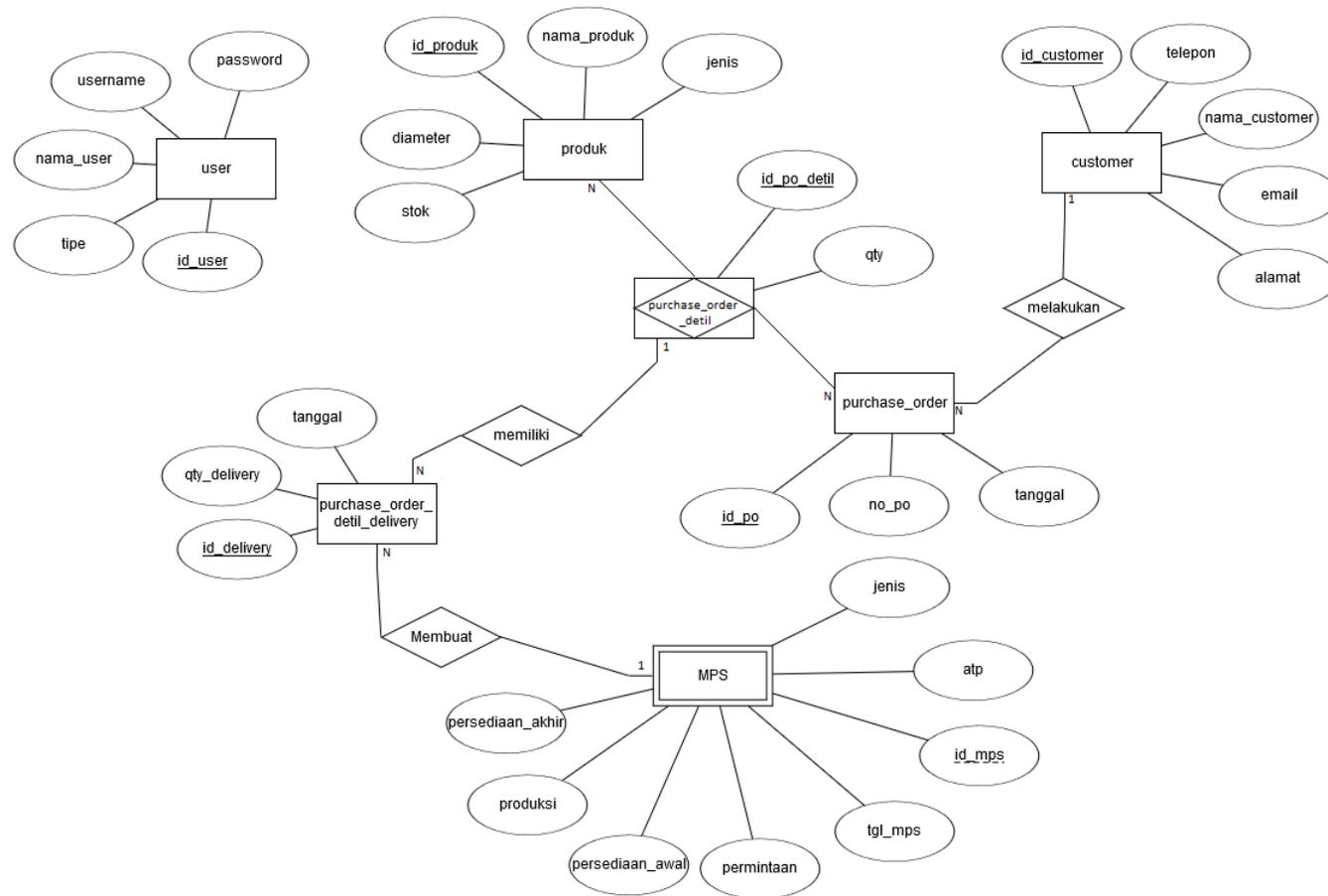
    public function delete_schedule_detail($id,$no_po)
    {
        $where = array(
            'id_delivery' => $id
        );

        $this->m_perencanaan-
>hapus_customer($where,'purchase_order_detil_delivery');
        $this->session->set_flashdata('pesan', '<div class="alert alert-
warning alert-dismissible">
            <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
            Data Delivery Berhasil Di Hapus.
            </div>');
        echo json_encode(array("status" => TRUE));
    }

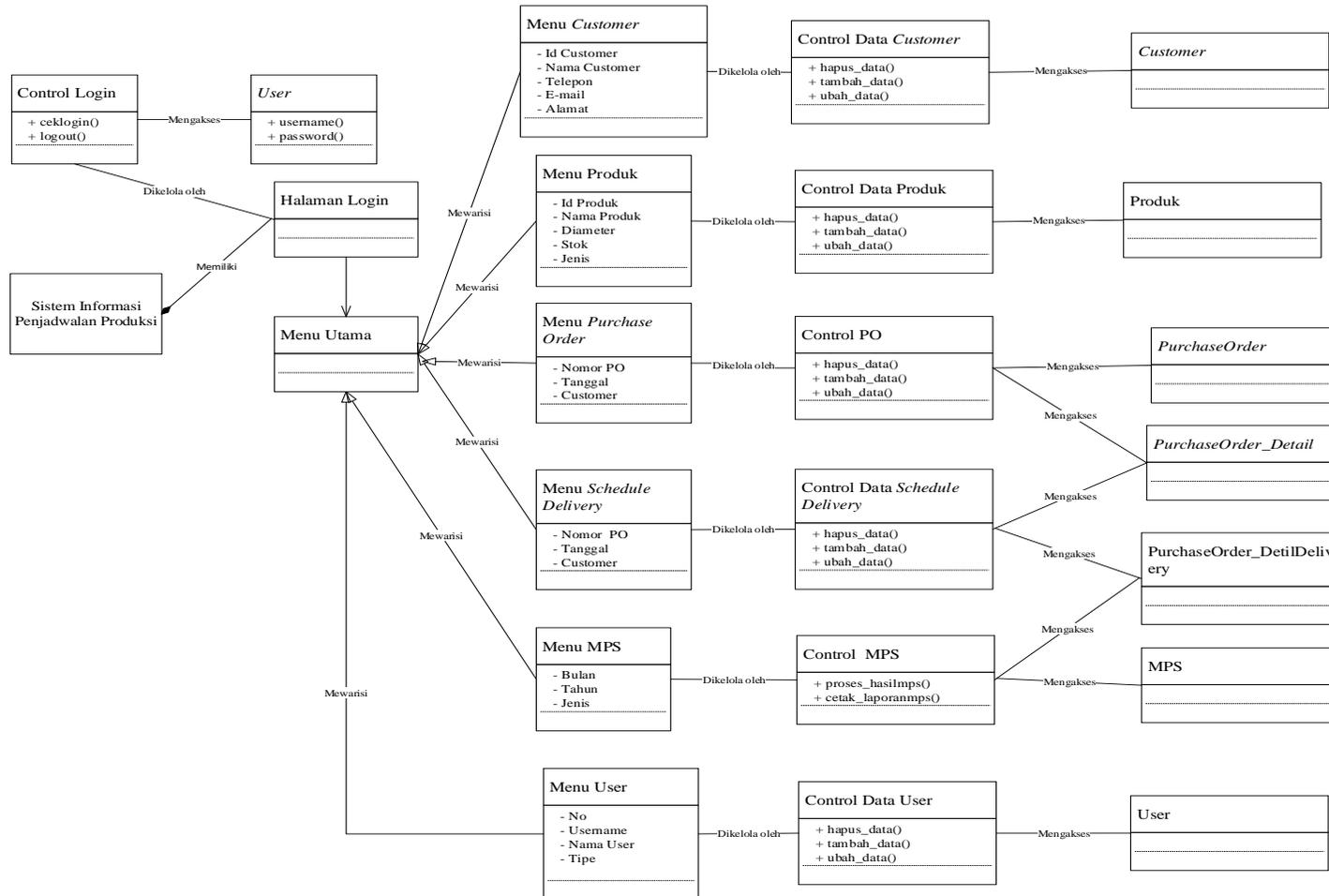
```

```
        //redirect('schedule_delivery/detail_sd/.$no_po);  
    }  
}
```

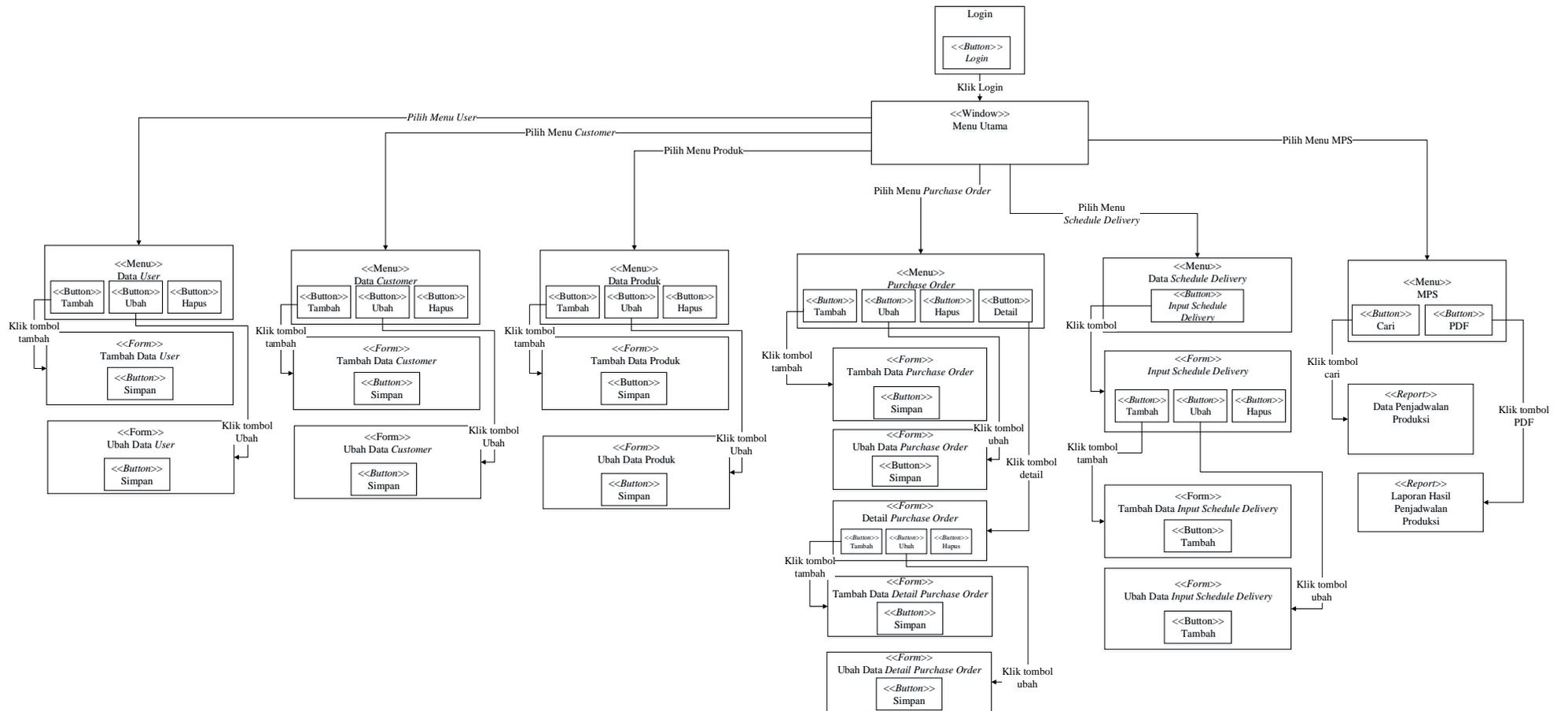
2. ERD



3. Class Diagram



4. Windows Navigation Diagram



5. Sequence Diagram Mengelola Data Purchase Order

