No Pok : 7036

D3 658.787
Abr
S

SISTEM INFORMASI PERSEDIAAN MENGGUNAKAN METODE *BUFFER STOCK* UNTUK MENGENDALIKAN PERSEDIAAN BARANG JADI PADA PT BANEEX INDONESIA

SUMBANGAN ALUMNI

TUGAS AKHIR

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

OLEH
DIMAS ABRIANNO
1314038

DATA BUKU PERPUSTAKAAN

Tgl Terima	21/10/22
No Induk Buku	986/SIS/SB/TA/22

POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA
JAKARTA
2019

No Pok : 7036

D3 658.787
Abr
S

SISTEM INFORMASI PERSEDIAAN MENGGUNAKAN METODE *BUFFER STOCK* UNTUK MENGENDALIKAN PERSEDIAAN BARANG JADI PADA PT BANEEX INDONESIA

SUMBANGAN ALUMNI

TUGAS AKHIR

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

OLEH
DIMAS ABRIANNO
1314038

DATA BUKU PERPUSTAKAAN

Tgl Terima	21/10/22
No Induk Buku	986/SIS/SB/TA/22

POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA
JAKARTA
2019

POLITEKNIK STMI JAKARTA

KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

LEMBAR PERSETUJUAN DOSEN PEMBIMBING

JUDUL TUGAS AKHIR:

SISTEM INFORMASI PERSEDIAAN MENGGUNAKAN METODE *BUFFER STOCK* UNTUK MENGENDALIKAN PERSEDIAAN BARANG JADI PADA PT BANEEX INDONESIA

DISUSUN OLEH:

Nama	: Dimas Abrianno
NIM	: 1314038
Program Studi	: Sistem Informasi Industri Otomotif
Tanggal Seminar	: 20 September 2019
Tanggal Sidang	: 23 September 2019
Tanggal Lulus	: 24 September 2019

Jakarta, 25 September 2019

Dosen Pembimbing

Lucky Heriyanto, S.T., M.T.I

NIP : 197908202009011009

POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

LEMBAR PENGESAHAN

JUDUL TUGAS AKHIR:

SISTEM INFORMASI PERSEDIAAN MENGGUNAKAN METODE *BUFFER STOCK* UNTUK MENGENDALIKAN PERSEDIAAN BARANG JADI PADA PT BANEEX INDONESIA

Disusun Oleh:

Nama	:	Dimas Abrianno
NIM	:	1310438
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 Senin Tanggal 23 September 2019.

Jakarta, 25 September 2019

Dosen Pembimbing	Ketua Penguji
Lucky Heriyanto, S.T., M.T.I	**Dr.Ridzky Kramanandita,S.Kom,M.T.**
NIP : 197908202009011009	NIP : 197403022002121001
Dosen Penguji	Dosen Penguji
Yulius Jatmiko Nuryanto, S.E, MM.	**Ahmad Juniar, S.Kom, M.T**
NIP : 198607262014021001	NIP : 197906052006041002

BADAN PENGEMBANGAN SUMBER DAYA MANUSIA INDUSTRI
POLITEKNIK STMI JAKARTA
Jl. Letjen Suprapto No. 26 Cempaka Putih. Jakarta 10510
Telp: (021) 42886064 Fax: (021) 42888206
www.stmi.ac.id

LEMBAR BIMBINGAN PENYUSUNAN TUGAS AKHIR

Nama : Dimas Abrianno
NIM : 1314038
Judul Tugas Akhir : Sistem Informasi Persediaan Menggunakan Metode *Buffer Stock* Untuk Mengendalikan Persediaan Barang Jadi Pada PT. Baneex Indonesia
Pembimbing : Lucky Heriyanto, ST, MTI

Tanggal	Keterangan	Paraf
19 April 2018	Bimbingan BAB I	
20 April 2018	Revisi BAB I dan Bimbingan BAB II	
26 Juni 2018	Revisi BAB II dan Bimbingan BAB III	
05 Agustus 2018	Revisi BAB III dan Bimbingan BAB IV	
29 Agustus 2018	Revisi BAB IV	
29 Desember 2018	Revisi BAB IV dan Bimbingan BAB V	
07 Mei 2019	Revisi BAB V Flow Usulan	
03 Juni 2019	Revisi BAB V Use Case	
19 Juni 2019	Revisi BAB V	
07 Agustus 2019	Revisi BAB V dan Demo Program	

Mengetahui,
Ketua Program Studi
Sistem Informasi Industri Otomotif

an

Noveriza Yuliasari, M.T
NIP : 197811212009012003

Dosen Pembimbing

Lucky Heriyanto, ST, MTI
NIP : 197908202009011009

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : Dimas Abrianno

NIM : 1314038

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

"SISTEM INFORMASI PERSEDIAAN MENGGUNAKAN METODE *BUFFER STOCK* DAN UNTUK MENGENDALIKAN PERSEDIAAN BARANG JADI PADA PT BANEEX INDONESIA"

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

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

Jakarta, 25 September 2019



Dimas Abrianno

ABSTRAK

PT Baneex Indonesia merupakan sebuah industri manufaktur yang bergerak dibidang otomotif dengan melakukan produksi komponen *Spring, Bending* dan *Stamping*. Divisi Inventori pada PT Baneex Indonesia bertanggung jawab melakukan berbagai macam pekerjaan terkait mengelola barang jadi, pembuatan kartu stok barang jadi dan melakukan perubahan pada stok barang jadi keseluruhan. Dalam melakukan kegiatanya, Divisi Inventori PT Baneex Indonesia masih menggunakan sistem manual baik dalam pembuatan kartu stok barang jadi dan perubahan stok keseluruhah barang jadi. Sistem informasi persediaan ditujukan untuk membantu mengelola data barang jadi agar setiap data dapat terorganisir dengan baik, dan membantu dalam mengendalikan stok barang jadi agar mengurangi terjadinya kehabisan stok serta kesalahan dalam penghitungan stok barang jadi dapat lebih dihindari. 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 antarmuka pengguna. Pengujian sistem menggunakan *Black Box Testing*. Pembuatan aplikasi menggunakan PHP 7.3.7 dan MariaDB 10.3.16 sebagai *database* dan menerapkan metode *Buffer Stok* dan *Reorder Point*, untuk menjaga ketersediaan stok. Aplikasi sistem informasi persediaan barang jadi dibuat berdasarkan proses bisnis yang berjalan dan permasalahan yang ditemukan di PT Baneex Indonesia ini dapat mengorganisir dan mengelola barang jadi dengan baik, mempermudah dalam menjaga stok barang jadi menyelesaikan permasalahan yang ada.

Kata Kunci : Sistem Informasi, persediaan dan mengendalikan, barang jadi, *Waterfall, Buffer Stock,*

KATA PENGANTAR

Puji syukur kepada Allah SWT yang telah memberikan rahmat dan karunia – Nya sehingga Penulis dapat menyelesaikan laporan Tugas Akhir ini dengan judul "**Sistem Informasi Persediaan Menggunakan Metode *Buffer Stock* Untuk Mengendalikan Persediaan Barang Jadi Pada PT. Baneex Indonesia**".

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

Dalam menyusun Tugas Akhir ini penulis telah banyak mendapat bimbingan dan informasi dari seluruh pihak yang terkait. Untuk itu, pada kesempatan kali ini penulis ingin menyampaikan terima kasih kepada:

1. Allah SWT atas rahmat serta kemudahan yang diberikan.
2. Kedua orang tua dan Kakak tercinta yang selalu mendoakan dan selalu memberikan dukungan baik moril maupun materil selama menyusun laporan Tugas Akhir.
3. Bapak Dr. Mustofa, S.T., M.T., selaku Direktur Politeknik STMI Jakarta.
4. Ibu Noveriza Yuliasari, Ssi, MT selaku Ketua Program Studi Sistem Informasi Industri Otomotif.
5. Bapak Lucky Heriyanto, S.T., M.T.I., selaku dosen pembimbing yang telah memberikan arahan dan masukan dengan sabar dalam penyusunan Tugas Akhir.
6. Bapak Ir. H. Tri Yono Sudarno, MM,. selaku Presiden Direktur PT Baneex Indonesia
7. Ibu Sekar selaku pembimbing yang telah membantu mengarahkan dan membimbing selama Kerja Lapangan.
8. Bapak Nanda, Bapak Iwan, Bapak Hotben, Ibu Etha, Ibu Risa 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 kampus STMI yang telah membantu penulis.
11. Teman-Teman HEM *Community* Ainalisa Alya Nabella, Akhmad Febrianto, Dimas Pangestu Surwendo, Fadhla Fanini, Gumelar Pinanggih Mulyo, Indri Fitriana Alfia Nisa , M. Fakhri Tri Febriyanto, Roni Sudrajat, Rusdiansyah, Sofyan Sauri yang telah membantu penulis menyelesaikan Tugas Akhir.
12. Anak Agung Aditya Kusuma Patera S.E yang telah meminjamkan laptop dan Atifah Esti Pratiwi S.E yang bersedia menerima keluh kesah penulis dalam mengerjakan Tugas Akhir ini.
13. Teman-teman SA01 2014 yang selalu memberi motivasi dan semangat dalam menyusun Tugas Akhir.

Demikianlah penulis berharap semoga laporan ini dapat dijadikan bahan kajian, walaupun dari pemikiran ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran guna perbaikan yang bersifat membangun sangat penulis harapkan. Akhir kata, penulis berharap semoga tulisan ini dapat bermanfaat.

Jakarta, 8 Agustus 2019



Penulis

DAFTAR ISI

Halaman

HALAMAN JUDUL i
LEMBAR PERSETUJUAN DOSEN PEMBIMBING ii
LEMBAR PENGESAHAN TUGAS AKHIR iii
LEMBAR BIMBINGAN PRAKTEK TUGAS AKHR iv
LEMBAR PERNYATAAN KEASLIAN v
ABSTRAK vi
KATA PENGANTAR vii
DAFTAR ISI ix
DAFTAR GAMBAR xiii
DAFTAR TABEL xv
DAFTAR LAMPIRAN xvii

BAB I PENDAHULUAN 1
1.1. Latar Belakang 1
1.2. Pokok Permasalahn 3
1.3. Tujuan Penelitian 3
1.4. Batasan Masalah 3
1.5. Manfaat Tugas Akhir 4
1.6. Sistematika Penulisan 3

BAB II LANDASAN TEORI 6
2.1 Pengertian Sistem 6
2.1.1. Komponen Sistem 7
2.1.2. Klasifikasi Sistem 9
2.2 Konsep Dasar Informasi 9
2.2.1. Kualitas Informasi 10
2.2.2. Siklus Informasi 12
2.3 Konsep Dasar Sistem Informasi 14

2.3.1 Komponen Sistem Informasi 15
2.4 Pengertian Persediaan (*Inventory)* 16
2.4.1 Tujuan Persediaan 16
2.4.2 Jenis – Jenis Persediaan 16
2.4.3 Metode *Buffer Stock* 17
2.5 Pengertian Gudang 16
2.5.1. Alasan Diperlukannya Gudang 16
2.6 Pengertian Barang Jadi 19
2.7 *System Development Life Cycle* (SDLC) 20
2.7.1. *Planning* (Perencanaan) 21
2.7.2 *Analysis* (Analisis) 22
2.7.3 *Design* (Desain) 23
2.7.4 *Implementation*(Implementasi) 24
2.8 Metode Pengembangan Sistem *Waterfall* 25
2.9 Flowchart 26
2.10 *Unified Modelling Language* 28
2.10.1. *Use Case Diagram* 28
2.10.2. *Activity Diagram* 29
2.10.3. *Sequence Diagram* 32
2.10.4. *Class Diagram* 33
2.11 Sistem Manajemen Basis Data 35
2.11.1 Database 35
2.11.2 *Entity Relationship Diagram* (ERD) 36
2.11.3 Kamus Data 37
2.12 MariaDB 38
2.12.1 Tipe Data 39
2.13 *Windows Navigation Diagram* 41
2.14 *Hypertext Preprocessor* (PHP) 42
2.15 *Framework* 43
2.16 *Codeigniter* 44
2.17 *Black-Box Testing* 46

2.17.1 Metode *Black-Box Testing* 47
2.17.2 Perbedaan *White Box* dan *Black Box* 48
2.18 Analisis *Pieces* 49

BAB III METODOLOGI PENELITIAN 50
3.1 Metodologi Penelitian 50
3.2 Identifikasi Masalah 50
3.3 Pengumpulan Data 50
3.4 Metode Pengumpulan Data 51
3.5 Metode Pengembangan Sistem 51
3.6 *Literture Review* 53
3.7 Kerangka Penelitian 55

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA 61
4.1. Tentang Perusahaan 61
4.2. Profil Perusahaan 62
4.3. Visi dan Misi Perusahaan 63
4.4. Struktur Organisasi 63
4.5. *International Organization For Standardization* 73
4.6. Analisis Proses (*Flowmap* Sistem Berjalan) 74
4.6.1. Prosedur Sistem Persediaan Barang Jadi 74
4.6.2. *Flowmap* Proses Persediaan Barang Jadi 75
4.7. Analisis Dokumen Persediaan Barang Jadi 77
4.7.1 Dokumen Persediaan Barang Jadi 77
4.8. Analisis Permasalahan 82

BAB V ANALISIS DAN PEMBAHASAN 86
5.1. Analisis Kebutuhan Sistem 86
5.1.1. *Functional Requirement* 87
5.1.2. *Non Functional Requirement* 88
5.2. Flowmap Sistem Informasi Persediaan Barang Jadi Usulan. 89
5.3. *Use Case* Diagram (Usulan) 90

5.3.1. *Use Case Description* 90
5.4. *Activity Diagram* 96
5.5. *Sequence Diagram* 106
5.6. *Class Diagram* 115
5.7. *Deployment Diagram* 116
5.8. Pemodelan Sistem Usulan 117
5.9.1. *Entity Relationship Diagram* 117
5.9.2. Kamus Data 125
5.9. *Windows Navigation Diagram* 123
5.10. Perancangan Antar Muka 123

BAB VI KESIMPULAN DAN SARAN 129
6.1 Kesimpulan 129
6.2 Saran 129
DAFTAR PUSTAKA 130
LAMPIRAN

5.3.1. *Use Case Description* 90
5.4. *Activity Diagram* 96
5.5. *Sequence Diagram* 106
5.6. *Class Diagram* 115
5.7. *Deployment Diagram* 116
5.8. Pemodelan Sistem Usulan 117
5.9.1. *Entity Relationship Diagram* 117
5.9.2. Kamus Data 125
5.9. *Windows Navigation Diagram* 123
5.10. Perancangan Antar Muka 123

BAB VI KESIMPULAN DAN SARAN 129
6.1 Kesimpulan 129
6.2 Saran 129
DAFTAR PUSTAKA 130
LAMPIRAN

DAFTAR TABEL

Halaman

Tabel II.1	Simbol-simbol *Flowchart*	27
Tabel II.2	Simbol *Use Case Diagram*	28
Tabel II.3	Simbol *Activity Diagram*	30
Tabel II.4	*Sequence Diagram*	32
Tabel II.5	*Class Diagram*	34
Tabel II.6	Contoh Tabel Barang	36
Tabel II.7	Simbol-simbol ERD	36
Tabel II.8	Notasi-Notasi Kamus Data	38
Tabel III.1	*Literature Review*	54
Tabel IV.1	Analisis Permasalahan	82
Tabel V.1	Analisis Kebutuhan Sistem	86
Tabel V.2	*Use Case Description Login*	91
Tabel V.3	*Use Case Description* Mengelola Data Barang Jadi	91
Tabel V.4	*Use Case Description* Mengelola Data Hasil Produksi	92
Tabel V.5	*Use Case Description* Mengelola Data Surat Jalan	92
Tabel V.6	*Use Case Description* Cetak Surat Jalan	93
Tabel V.7	*Use Case Description* Validasi Barang Jadi Keluar	93
Tabel V.8	*Use Case Description* Mengelola Data *User*	94
Tabel V.9	*Use Case Description* Mengelola Data *Customer*	94
Tabel V.10	*Use Case Description* Pembuatan Rekap Stok	95
Tabel V.11	*Use Case Description* Pembuatan Kartu Stok	95
Tabel V.12	*Use Case Description* Melihat Laporan Hasil Produksi	95
Tabel V.13	Spesifikasi Tabel *User*	118
Tabel V.14	Spesifikasi Tabel Data Barang	119
Tabel V.15	Spesifikasi Tabel Hasil Produksi	119
Tabel V.16	Spesifikasi Tabel Detil Hasil Produksi	120
Tabel V.17	Spesifikasi Tabel Data *Customer*	120

Tabel V.18 Spesifikasi Tabel Surat Jalan..... 121
Tabel V.19 Spesifikasi Tabel Detil Surat Jalan..... 122
Tabel V.20 Spesifikasi Tabel Barang Keluar..... 122
Tabel V.21 Spesifikasi Tabel Rekap Stok..... 122

DAFTAR GAMBAR

Halaman

Gambar II.1	Pilar Kualitas Informasi	12
Gambar II.2	Siklus Informasi dalam Anggraeni dan Irviana	13
Gambar II.3	Siklus Informasi dalam Hutahaean	14
Gambar II.4	Rumus Perhitungan *Buffer Stock*	18
Gambar II.5	Metode *Waterfall* Berdasarkan Dennis *et al*	26
Gambar II.6	*Windows Navigation Diagram*	42
Gambar II.7	*Codeigniter*	46
Gambar III.1	Metode *Waterfall*	53
Gambar III.2	Kerangka Penelitian	59
Gambar IV.1	Logo PT Baneex Indonesia	62
Gambar IV.2	Struktur Organisasi PT Baneex Indonesia	64
Gambar IV.3	ISO PT. Baneex Indonesia	74
Gambar IV.4	*Flowmap* Proses Persediaan Barang Jadi	76
Gambar IV.5	Form Hasil Produksi	78
Gambar IV.6	Rekap Stok	79
Gambar IV.8	Kartu Stok	80
Gambar IV.9	Surat Jalan	81
Gambar V.1	*Flowmap* Usulan Sistem Persediaan Barang Jadi	89
Gambar V.2	*Use Case Diagram* Usulan	90
Gambar V.3	*Activity Diagram Login*	96
Gambar V.4	*Activity Diagram* Mengelola Data User	97
Gambar V.5	*Activity Diagram* Mengelola Data Barang Jadi	98
Gambar V.6	*Activity Diagram* Mengelola Data Hasil Produksi	99
Gambar V.7	*Activity Diagram* Mengelola Data Surat Jalan	100
Gambar V.8	*Activity Diagram* Validasi Barang Jadi Keluar	101
Gambar V.9	*Activity Diagram* Pembuatan Kartu Stok	102
Gambar V.10	*Activity Diagram Customer*	103

Gambar V.11 *Activity Diagram* Pembuatan Rekap Stok 104
Gambar V.12 *Activity Diagram* Cetak Surat Jalan 105
Gambar V.13 *Activity Diagram* Melihat Laporan Hasil Produksi 106
Gambar V.14 *Sequence Diagram Login* 107
Gambar V.15 *Sequence Diagram Customer* 108
Gambar V.16 *Sequence Diagram* Data *User* 109
Gambar V.17 *Sequence Diagram* Mengelola Data Barang Jadi 110
Gambar V.18 *Sequence Diagram* Mengelola Data Hasil Produksi 111
Gambar V.19 *Sequence Diagram* Data Surat Jalan 112
Gambar V.20 *Sequence Diagram* Data Barang Keluar 113
Gambar V.21 *Sequence Diagram* Cetak Surat Jalan 113
Gambar V.22 *Sequence Diagram* Pembuatan Rekap Stok 114
Gambar V.23 *Sequence Diagram* Pembuatan Kartu Stok 115
Gambar V.24 *Class Diagram* 116
Gambar V.25 *Deployment Diagram* 117
Gambar V.26 *Entitiy Relationship Diagram* 118
Gambar V.27 *Windows Navigation Diagram* 123
Gambar V.28 *Form Login* 124
Gambar V.29 Data User 124
Gambar V.30 Data Barang Jadi 124
Gambar V.31 Data Hasil Produksi 125
Gambar V.32 Data Detil Hasil Produksi 125
Gambar V.33 Data Surat Jalan 125
Gambar V.34 Data Detil Surat Jalan 126
Gambar V.35 Data Barang Keluar 126
Gambar V.36 Data Detil Barang Keluar 126
Gambar V.37 Data Rekap Stok 127
Gambar V.38 Data Kartu Stok 127

DAFTAR LAMPIRAN

Halaman

LAMPIRAN A WAWANCARA **L-1**
Tabel Wawancara L-1
Transkrip Wawancara L-2
Topik Pengiriman Barang Jadi Ke Bagian Inventori L-2
Topik Proses Masuk dan Keluar Barang Jadi Di Bagian Inventori ... L-3
LAMPIRAN B OBSERVASI **L-5**
Dokumen Hasil Observasi L-5
Data Hasil Produksi L-5
Surat Jalan L-6
Kartu Stok Barang Jadi L-7
Rekap Stok Barang Jadi L-8

LAMPIRAN D *SOURCE CODING* **L-9**
LAMPIRAN E BLACK-BOX TESTING **L-23**

MILIK PERPUSTAKAAN STMI
Membaca : Ibadah, Mengambil : Dosa

BAB I
PENDAHULUAN

1.1 Latar Belakang

Secara umum pengertian inventori adalah suatu aset yang ada dalam bentuk barang - barang yang dimiliki perusahaan. Inventori meliputi semua barang yang dimiliki perusahaan dengan maksud untuk dikonsumsi dalam siklus operasi dalam perusahaan sebagai barang yang dimiliki. Inventori diklasifikasikan menurut jenis perusahaan, perusahaan yang penulis amati adalah perusahaan yang mengolah barang mentah menjadi barang jadi. Beberapa macam klasifikasi dalam jenis perusuhaan ini, antara lain : Persediaan Produksi, Persediaan MRO (*Maintenance, Repair,* dan *operating supplies*), Persediaan Proses Produksi, dan Persediaan Barang Jadi. Dan penulis hanya berfokus pada Persediaan Barang Jadi.

Perkembangan teknologi informasi yang sangat cepat dan maju pada abad ini, memberikan dampak besar bagi segala aktivitas industri dan terutama pada persediaan barang jadi. Kemudahan dalam pertukaran data dan informasi adalah salah satu bukti yang dapat dilihat dan dirasakan oleh pekerja industri. Oleh karena itu setiap perusahaan industri mempunyai tujuan dalam mencapai hasil yang maksimal dan waktu yang cepat dalam memberikan kepuasan bagi pelanggan. Maka dari itu diperlukan adanya suatu keharmonisasian dalam hubungan antar tiap divisi, agar dapat mencapai suatu tujuan yang ingin dicapai.

PT Baneex Indonesia merupakan perusahaan industri manufaktur yang membuat komponen mobil khusus jok mobil. Dalam kegiatan produksinya PT Baneex Indonesia mempunyai divisi Inventori yang mempunyai tugas dalam hal mengontrol persediaan bahan baku dan barang jadi yang tersimpan agar produksi dapat terus berjalan.. PT Baneex Indonesia memiliki satu karyawan di Bagian Inventori yang kegiatannya mencakup ruang lingkup bahan baku dan barang jadi. Data persediaan pada Bagian Inventori mudah rusak dan sulit dalam pencarian data di komputer dan data tidak terintegrasi serta tidak adanya laporan.

Dalam menjalankan aktivitasnya, divisi Inventori PT Baneex Indonesia menggunakan *Microsoft Excel* dalam pencatatan penyimpanan barang jadi, yang menyebabkan data yang tersimpan tidak terorganisir dengan rapi. Selain itu beberapa permasalahan terjadi seperti tidak adanya metode yang digunakan dalam pengendalian barang jadi, setiap terjadinya keterlambatan proses produksi yang menghasilkan barang jadi, membuat karyawan harus lembur untuk memenuhi penjualan. Oleh karena itu penggunaan metode persediaan barang jadi *Buffer Stock* sangat tepat untuk diterapkan pada PT Baneex Indonesia agar data persediaan barang jadi dapat lebih terorganisir dengan rapi.

Berdasarkan latar belakang dan permasalahan tersebut, maka disusun tugas akhir dengan judul "SISTEM INFORMASI PERSEDIAAN MENGGUNAKAN METODE *BUFFER STOCK* UNTUK MENGENDALIKAN PERSEDIAAN BARANG JADI PADA PT BANEEX INDONESIA" untuk membantu perusahaan dalam memproses penyimpanan barang jadi sehingga proses penympanan barang jadi bisa dilakukan dengan maksimal.

1.2 Pokok Permasalahan

Berdasarkan latar belakang di atas, maka dapat di identifikasikan kendala yang terdapat di Bagian Inventori di PT Baneex Indonesia yakni sebagai berikut:

1. Proses persediaan barang jadi yang masih menggunakan cara manual, dikarenakan data barang jadi yang di masukkan ke dalam *Microsoft Excel* akan ditulis ulang di kartu stok dan rekap stok.
2. Belum adanya metode perhitungan yang dapat menjaga persediaan barang jadi.

1.3 Tujuan Penelitian

Tujuan penelitian dari tugas akhir ini yang dilakukan di departemen inventori PT Baneex Indonesia adalah:

1. Membangun suatu sistem informasi persediaan menggunakan basis data sehingga membantu dalam melakukan pembaruan informasi stok barang jadi

yang masuk, proses penyimpanan barang jadi menjadi lebih terorganisir, membantu dalam pencarian data, serta mencegah terjadinya kehilangan data.

2. Menerapkan metode *Buffer Stock* pada sistem informasi persediaan barang jadi untuk membantu staf inventori dalam menjaga persediaan barang jadi.

1.4 Batasan Masalah

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

1. Tempat melakukan penelitian adalah di PT. Baneex Indonesia.
2. Penelitian dilakukan pada divisi inventori selama satu bulan terhitung sejak 22 Januari 2018 s.d. 22 Februari 2018.
3. Ruang lingkup yang diamati sebatas pada proses persediaan barang jadi dari selesai produksi hingga pembuatan surat jalan.
4. Lingkup barang jadi yang diamati adalah *Trim Wire Ar/L Fr Cush, Trim Wire Br Fr Cush, Trim Wire Cr Fr Cush, Support Wire Mid Back, Trim Wire Ar\L Mid Cush*, Trim Wire Bl Mid Cush, Wire R Mid Cush, Bracket Wire Mid Cush.
5. Pengembangan Sistem Informasi menggunakan *Framework Code Igniter* versi 3.1.10 dengan bahasa pemrograman PHP 7.5.4 dan *database MariaDB* versi 2.6.26

1.5 Manfaat Tugas Akhir

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

1. Memberikan data stok barang jadi yang informatif dalam bentuk tabel untuk perusahaan.
2. Membantu perusahaan dalam pembuatan laporan kartu stok barang jadi

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, *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 persediaan barang jadi 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 kamus data, perancangan tampilan layar, perancangan hierarki menu, 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 persediaan barang jadi untuk perusahaan dalam penelitian dan pengembangan selanjutnya.

MILIK PERPUSTAKAAN STMI
Membaca : Ibadah, Mengambil : Dosa

BAB II
LANDASAN TEORI

2.1 Pengertian Sistem

Berdasarkan Djahir dan Pratita (2014), sistem adalah kumpulan dari komponen-kompenen 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 memampukan 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.

Berdasarkan Rukun dan Hayadi (2018), kata "sistem" mengandung arti kumpulan dari komponen-komponen yang memiliki unsur keterkaitan antara satu dan lainnya.

Berdasarkan Anggraeni dan Irviani (2017), sistem adalah kumpulan orang yang saling bekerja sama dengan ketentuan-ketentuan aturan yang sistematis dan terstruktur untuk membentuk satu kesatuan yang melaksanakan suatu fungsi untuk mencapai tujuan.

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 Ludwig, 1991 sistem adalah seperangkat unsur yang saling berhubungan dan saling memengaruhi dalam satu lingkungan tertentu.
2. Menurut Gordon B. Davis, 1995 sistem merupakan bagian-bagian yang beroperasi secara bersama-sama untuk mencapai beberapa tujuan.
3. Menurut Raymond Mcleod, 2001 sistem adalah sekelompok elemen yang terintegrasi untuk mencapai suatu tujuan.
4. Menurut Jogiyanto, 2005 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 :

1. 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.
2. 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.
3. 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.
4. 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.

5. Masukan (*input)*

   Masukan merupakan komponen sistem, yaitu segala sesuatu yang perlu dimasukan ke dalam sistem sebagai bahan yang akan diolah lebih lanjut untuk menghasilkan keluaran yang berguna.

6. Pengolahan (*processing)*

   Pengolahan merupakan komponen sistem yang mempunyai peran utama mengolah masukan agar menghasilkan keluaran yang berguna bagi para pemakainya.

7. Keluaran (*output)*

   Keluaran merupakan komponen sistem yang berupa berbagai macam bentuk keluaran yang dihasilkan oleh komponen pengolahan.

8. Sasaran dan Tujuan (*objectives and goal)*

   Setiap komponen dalam sistem perlu dijaga agar saling bekerja sama agar mampu mencapai sasaran dan tujuan sistem.

9. Kendali (*control)*

   Setiap komponen dalam sistem perlu dijaga agar saling bekerja sesuai dengan peran dan fungsinya masing-masing.

10. Umpan Balik (*feed back)*

    Umpan balik diperlukan oleh bagian kendali untuk mengecek terjadinya penyimpanan proses dalam sistem dan mengembalikannya pada kondisi normal.

Menurut Jogianto (2005) 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 menurut Darmawan dan Kunkun (2016).

2.2 Konsep Dasar Informasi

Berikut ini adalah pendapat-pendapat yang dikemukakan para ahli mengenai definisi informasi yang dikutip dalam Muslihudin dan Oktafianto (2016):

1. Jogiyanto (2008), informasi adalah data yang diolah menjadi bentuk yang berguna bagi para pemakainya.
2. Sutarman (2008), informasi merupakan hasil pengolahan data, sehingga menjadi bentuk yang penting bagi penerimanya dan mempunyai kegunaan sebagai dasar dalam pengambilan keputusan yang dapat dirasakan

akibatnya secara langsung saat itu juga atau secara tidak langsung pada saat mendatang.

3. Krismiaji dalam Zabaer (2005), informasi adalah data yang telah diorganisasikan, dan telah memiliki kegunaan dan manfaat.
4. Raymond Mc. Leod, informasi adalah data yang telah diolah menjadi bentuk yang memiliki arti bagi si penerima dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang.

Menurut Muslihudin dan Oktafianto (2016), dapat disimpulkan bahwa informasi merupakan data yang diolah menjadi bentuk yang berguna untuk membuat keputusan. Informasi tersebut merupakan hasil pengolahan data atau fakta yang dikumpulkan dengan metode ataupun cara-cara tertentu.

Sedangkan berdasarkan Hutahaean (2015), informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya. Sumber informasi adalah data. Data kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian adalah kejadian yang terjadi pada saat tertentu. Menurut Gordon B. Davis dalam Hutahaen, Jeperson (2015) informasi adalah data yang telah diolah menjadi suatu bentuk yang penting bagi si penerima dan mempunyai nilai nyata atau yang dapat dirasakan dalam keputusan-keputusan yang sekarang atau keputusan-keputusan yang akan datang.

Berdasarkan Anggraeni dan Irviani (2017), informasi adalah sekumpulan data atau fakta yang diorganisasi atau diolah dengan cara tertentu sehingga mempunyai arti bagi penerima. Data yang diolah menjadi sesuatu yang berguna bagi si penerima. Maksudnya, yaitu dapat memberikan keterangan dan pengetahuan.

2.2.1 Kualitas Informasi

Berdasarkan Parker (1989) dalam Tyoso (2016), informasi yang berkualitas harus memenuhi beberapa persyaratan sebagai berikut:

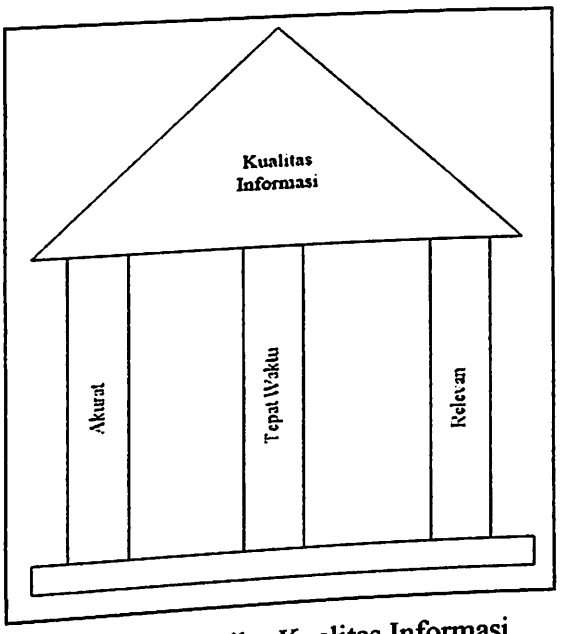
1. Ketersediaan (*Availability*)

2. Informasi harus dapat diakses oleh orang yang membutuhkannya, maka dari itu informasi harus tersedia setiap saat pada "gudang data" yang terorganisasi rapi.
3. Mudah dipahami (*Comprehensibility*)
   Informasi yang berbelit-belit atau tidak jelas koneksinya bahkan bersifat rumit, maka berakibat keputusan yang akan diambil tertunda, karena lebih banyak waktu yang digunakan untuk membahasnya.
4. Akurat (*Accuracy*)
   Informasi harus bersih dari kesalahan dan kekeliruan. Artinya, informasi harus jelas dan tepat dalam mencerminkan makna yang terkandung dari data.
5. Relevan (*Relevant*)
   Berkaitan dengan pengoperasian suatu organisasi, informasi yang dibutuhkan ialah informasi yang benar-benar relevan dengan permasalahan, misi dan tujuan organisasi yang bersangkutan.
6. Bermanfaat (*Benefits*)
   Informasi sebaiknya dapat disajikan dalam bentuk-bentuk yang mudah dilihat dan dipelajari sehingga kepemanfaatannya terlihat jelas.
7. Tepat Waktu (*Being On/In Time*)
   Informasi harus tersedia tepat pada waktunya sehingga saat organisasi membutuhkannya informasi sudah tersedia. Juga harus diperhatikan kapan informasi itu diperoleh pada peristiwa apa saat itu.
8. Keterandalan (*Reliability*)
   Informasi harus diperoleh dari sumber daya yang jelas dan dapat dipertanggungjawabkan kebenarannya. Demikian juga dengan pengelola atau pemberi informasi juga merupakan pihak-pihak yang dapat dipercaya.
9. Konsisten (*Consistent*)
   Informasi tidak bermuatan hal-hal yang kontradiktif, sehingga bahasa yang digunakan harus sesuai dengan yang disajikan.

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:

1. Relevansi
2. Ketepatan waktu
3. Keakurasian

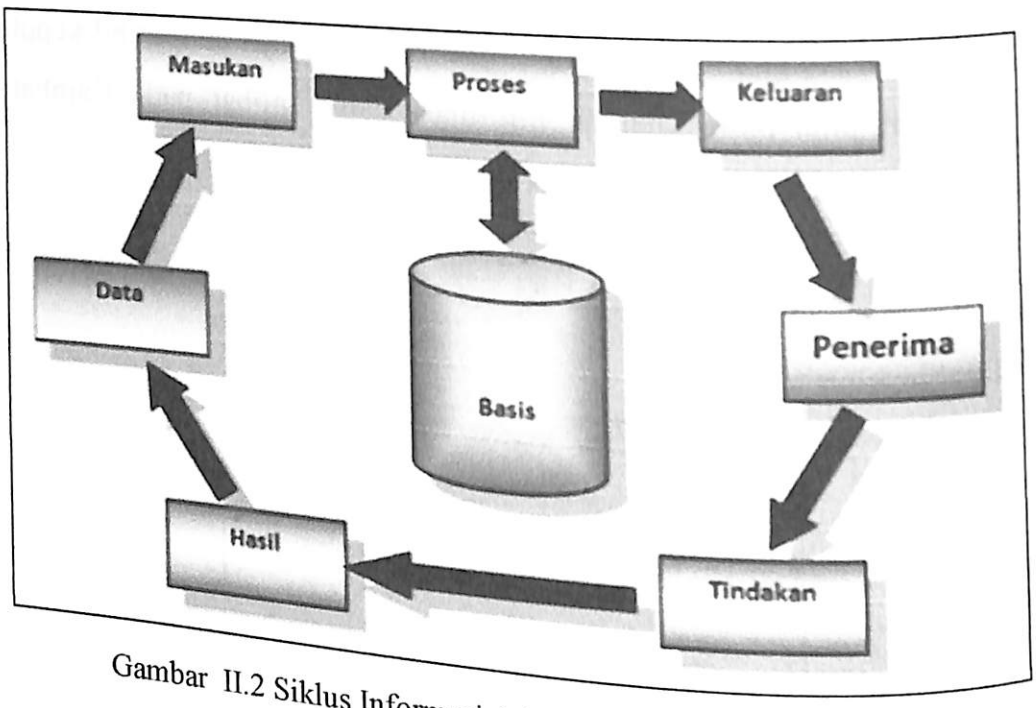


Gambar II.1 Pilar Kualitas Informasi

(Sumber: Anggraeni dan Irviani, 2017)

2.2.2 Siklus Informasi

Berdasarkan Burch dan Grudnitski (1989) 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 Anggraeni dan Irviana

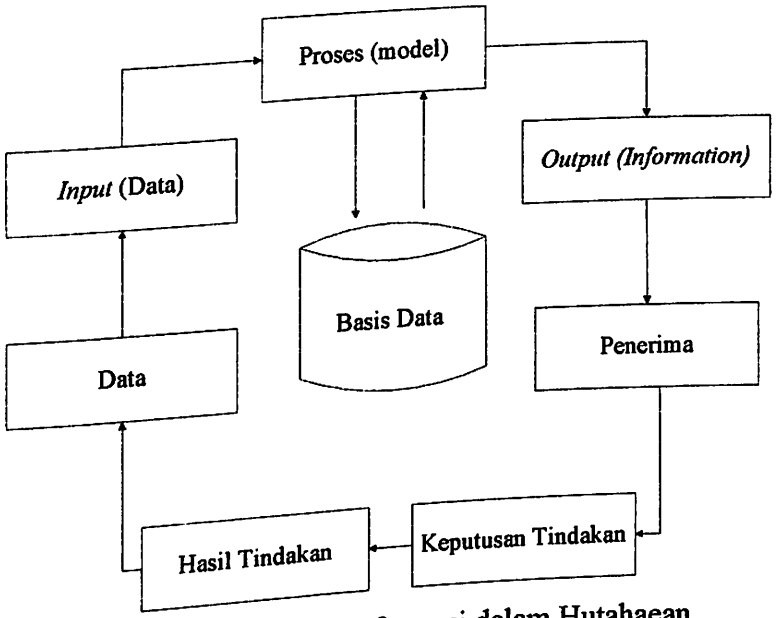
(Sumber: Anggareni dan Irviana, 2017)

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 sabagai *input*, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus.

Sedangkan berdasarkan Hutahaean (2015), data yang diolah untuk menghasilkan informasi menggunakan model proses yang tertentu. Misalkan suhu dalam *Fahrenheit* diubah ke dalam *Celcius*. Dalam hal ini digunakan model matematik berupa rumus konversi dari derajat *Fahrenheit* menjadi satuan derajat *Celcius*. Data yang diolah melalui suatu model menjadi informasi, kemudian penerima menerima informasi tersebut, yang berarti menghasilkan keputusan dan melakukan tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai *input*, diproses kembali lewat suatu model dan seterusnya yang disebut dengan siklus informasi (*information cycle*). Siklus ini juga disebut dengan siklus pengolahan data (*data processing cycles*). Siklus informasi ini dapat digambarkan sebagai berikut pada Gambar II.3.



Gambar II.3 Siklus Informasi dalam Hutahaean
(Sumber: Hutahaean, 2015)

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):

1. 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.
2. Menurut Bodnar dan Hopwood, sistem informasi adalah kumpulan perangkat keras dan lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna.
3. 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 Persediaan (*Inventory*)

Persediaan (*inventory*) adalah stock bahan yang digunakan untuk memudahkan produksi atau untuk memuaskan pelanggan yang meliputi bahan baku (*raw materials*), barang dalam proses (*in-process goods*), dan barang jadi (*finished goods*).

Menurut Assauri (2016) persediaan *(inventory)* adalah stok dari suatu *item* atau sumber daya yang digunakan dalam suatu organisasi perusahaan. Sistem inventori adalah sekumpulan kebijakan dan pengendalian, yang memonitor tingkat inventory dan menentukan tingkat mana yang harus dijaga, bila stok harus diisi kembali dan berapa banyak yang harus dipesan.

2.4.1 Tujuan Persediaan

Menurut Assauri (2016), persediaan memiliki beberapa tujuan, antara lain:

1. Untuk menjaga independensi dari operasi, dimana pasokan material pada *work center* dimungkinkan untuk dapat fleksibel dalam operasi.
2. Untuk dapat memenuhi variasi dari permintaan produk, dimana permintaan produk tidak dapat diketahui secara tepat, sehingga terdapat kesulitan untuk menghasilkan produk secara tepat dalam memenuhi permintaan.
3. Untuk memungkinkan dapat dilakukannya fleksibilitas dalam *scheduling* produksi, dimana disediakannya stok dari *inventory* guna menghilangkan tekanan terhadap sistem operasi produksi.
4. Untuk memberikan usaha perlindungan atau penjagaan terhadap perbedaan waktu *delivery* bahan baku, dimana terdapatnya keterlambatan atas kedatangan material yang dipesan dari *vendor*.

2.4.2 Jenis-jenis Persediaan

Assauri (2016) dalam bukunya untuk menjelaskan fungsi *inventory*, perusahaan-perusahaan umumnya menjaga adanya empat jenis *inventory*. Keempat jenis *inventory* itu adalah:

1. *Inventory* bahan baku (*Raw Material*) merupakan masukan awal proses transformasi produksi yang selanjutnya akan diolah menjadi produk jadi.

Ketersediaan bahan baku akan sangat menentukan kelancaran produksi sehingga perlu dikelola secara saksama. Inventori jenis ini didatangkan dari luar sistem dan keberadaannya secara fisik biasanya disimpan di gudang penerimaan (*receiving storage*).

2. *Inventory* barang dalam proses atau *Work-in-Process* (WIP) merupakan bentuk peralihan dari bahan baku menjadi produk jadi. Dalam sistem manufaktur yang bersifat pesanan (*job order*), adanya inventori barang setengah jadi biasanya tidak dapat dihindari sebab proses transformasi produksinya memerlukan waktu yang cukup lama. Sementara dalam sistem manufaktur yang bersifat produksi massa (*mass production*), adanya inventori barang setengah
3. *Maintenance/Repair/Operating supplies* (MROs) adalah mencurahkan untuk perlengkapan *maintenance/repair/operating* yang dibutuhkan, agar terjaga mesin-mesin dan proses dapat produktif
4. *Inventory* Barang Jadi (*finished goods*) adalah produk yang sudah selesai dalam proses produksi dan menunggu pengiriman. Barang jadi diinventorikan, karena untuk mengantisipasi permintaan yang melonjak dari para *customer* yang tidak dapat perkirakan.

2.4.3 Metode *Buffer Stock*

Menurut Herjanto (2008), *buffer stock* adalah persediaan pengamanan yang berfungsi untuk melindungi atau menjaga kemungkinan terjadinya kekurangan barang, misalnya karena penggunaan barang yang lebih besar dari perkiraan semula atau keterlambatan dalam penerimaan barang yang dipesan. Pentingnya menghitung *buffer stock* karena sering terlambanta pengiriman bahan baku, keadaan ini akan mengganggu proses produksi yang selanjutntya akan berakibat terjadinya *stock out*.

Menurut Cognizant perbedaan antara *buffer stock* dan *safety stock* adalah *buffer stock* mencegah produsen dari kemungkinan tidak dapat memenuhi permintaan pelanggan sedangkan *safety stock* membantu produsen untuk memenuhi variasi dalam penyediaan bahan baku

bahan baku (*raw material*), barang setengah jadi (*work-in process*) ataupun barang jadi (*finished good*).

Berdasarkan Martono (2015) gudang adalah tempat penyimpanan sementara dan pengambilan *inventory* untuk mendukung kegiatan operasi bagi proses operasi berikutnya, ke lokasi distribusi, atau kepada konsumen akhir.

Berdasarkan Martono (2015), gudang memiliki keuntungan tersendiri. Keuntungan adanya gudang bagi *inventory* adalah:

1. Menyediakan tempat untuk meletakan dan melindungi.
2. Menyediakan tepat waktu sesuai pesanan (menjamin *service level*).
3. Memonitor status.
4. Sebagai alat komunikasi dengan konsumen.
5. Mengurangi biaya transportasi.

2.5.1 Alasan Diperlukannya Gudang

Berdasarkan Warman (2012), dianalogikan seperti warung kue donat yang sedang ditunggui oleh banyak konsumen. Konsumen menunggu donat yang sedang disiapkan untuk digoreng. Setelah donat selesai dibuat, donat segera diserahkan ke konsumen. Dalam hal ini, tidak ada hambatan waktu atau gerak barang dari ujung garis produksi ketika donat digoreng, sampai kepada konsumen. Analogi sederhana ini menunjukkan bahwa gudang tidak diperlukan. Apabila gudang tidak diperlukan atau di mana penyaluran sangat minimal, maka biaya tambahan boleh dikatakan tidak ada. Semakin baik mengatur ruang gerak barang untuk mengurangi proses penyaluran, semakin konsumen dapat memperoleh barang itu dengan harga terjangkau. Jadi, gudang diperlukan untuk menyimpan barang jadi di mana barang tersebut mempunyai jangka waktu untuk sampai ke tangan pelanggan atau konsumen.

2.6 Pengertian Barang Jadi

Menurut Rusdi dan Suyuti (2017), barang jadi merupakan hasil akhir proses transformasi produksi yang siap dipasarkan kepada pemakai. Sebelum diangkut kepada pemakai yang membutuhkan, byarang jadi ini disimpan di gudang barang jadi. Dalam sistem manufaktur yang bersifat produksi massa,

biasanya barang jadi disimpan untuk beberapa waktu sampai dengan datangnya pembeli, sedangkan dalam sistem manufaktur yang bersifat pesanan, begitu barang tersebut selesai diproduksi akan segera diambil oleh pemakai yang memesannya. Dengan demikian, dalam sistem manufaktur berdasarkan pesanan sangat jarang ditemui *inventory* barang jadi di gudang.

Berdasarkan Assauri (2016), menjelaskan persediaan barang jadi adalah produk yang sudah selesai diproses dan menunggu pengiriman. Barang jadi disimpan, karena permintaan dari pelanggan pada masa depan adalah tidak dapat diketahui.

2.7 System Development Life Cycle (SDLC)

Menurut Dennis *et al* (2015) *System Development Life Cycle* (SDLC) memiliki 4 langkah dasar atau fase untuk membentuk suatu sistem informasi. Fase tersebut yaitu *planning*, *analysis*, *design*, dan *implementation* atau dapat disebut dengan perencanaan, analisis, perancangan, dan penerapan. Perbedaan proyek dalam membangun sebuah sistem infromasi mungkin dapat berbeda-beda sesuai dengan caranya masing-masing. Akan tetapi, semua proyek pasti memiliki elemen dari 4 fase tersebut. Masing-masing fase tersebut tersusun dari beberapa langkah yang mengandalkan pada teknik dan menghasilkan suatu hasil berupa dokumen yang spesifik atau file yang menjelaskan tentang proyek.

Tahapan SDLC diproses dalam suatu alur yang dimulai dari awal hingga akhir. Dalam suatu proyek, tim proyek memindahkan fase tersebut dengan teratur, secara bertahap, secara iteratif, atau dalam pola lainnya. Dalam SDLC, fase tersebut dijelaskan dalam beberapa teknik yang digunakan untuk menjelaskannya secara umum.

Ada dua hal penting untuk dipahami tentang SDLC:

1. Pertama, yang harus dipahami adalah mendapatkan dan memahami pengertian umum dari tahapan dan langkah-langkah yang digunakan dalam proyek sistem informasi serta beberapa teknik yang menghasilkan dokumen.

2. Kedua, hal yang penting untuk dipahami bahwa SDLC adalah proses penyempurnaan yang dilakukan secara bertahap.

Hasil yang diperoleh dalam tahap analisis adalah suatu ide umum dari bagian sistem yang baru. Hasil ini lalu digunakan sebagai *input* pada tahap *design*, yang kemudian disempurnakan atau diperbaiki untuk menghasilkan dokumen yang menjelaskan detail dari bentuk sistem yang dibuat. Dokumen ini, akan digunakan dalam tahap implementasi untuk menghasilkan sistem yang sebenarnya. Setiap tahapan menyempurnakan dan menguraikan pekerjaan yang telah dilakukan sebelumnya. Berikut dibawah ini merupakan setiap tahapan dalam SDLC:

2.7.1. *Planning* (Perencanaan)

Tahap *planning* / perencanaan merupakan 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:

1. 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.
2. *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. Setelah proyek disetujui, 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.

2.7.2. *Analisys* (Analisis)

Berdasarkan Wikipedia Indonesia (2018), dalam linguistik, analisis adalah kajian yang dilaksanakan terhadap sebuah bahasa guna meneliti struktur bahasa tersebut secara mendalam. Sedangkan pada kegiatan laboratorium, kata analisis dapat juga berarti kegiatan yang dilakukan di laboratorium untuk memeriksa kandungan suatu zat dalam cuplikan.

Sedangkan berdasarkan Kamus Besar Bahasa Indonesia (KBBI) (2018), analisis adalah penyelidikan terhadap suatu peristiwa (karangan, perbuatan, dan sebagainya) untuk mengetahui keadaan yang sebenarnya (sebab-musabab, duduk perkaranya, dan sebagainya).

Berdasarkan Muslihudin dan Oktafianto (2016), analisis sistem adalah sebuah istilah yang secara kolektif mendeskripsikan fase-fase awal pengembangan sistem. Analisis sistem adalah teknik pemecahan masalah yang menguraikan bagian-bagian komponen dengan mempelajari seberapa bagus bagian-bagian komponen tersebut bekerja dan berinteraksi untuk mencapai tujuan mereka. Analisis sistem merupakan tahapan paling awal dari pengembangan sistem yang menjadi fondasi dalam menentukan keberhasilan sistem informasi yang dihasilkan nantinya. Tahapan ini merupakan tahap yang mudah jika klien sangat paham dengan masalah yang dihadapi organisasinya dan tahu betul fungsionalitas dari sistem informasi yang akan dibuat. Tetapi tahap ini bisa menjadi tahap paling sulit jika klien tidak bisa mengidentifikasi kebutuhannya dan tertutup terhadap pihak luar yang ingin mengetahui detail proses-proses bisnisnya.

Tahap analisis dalam SDLC akan mengidentifikasi siapa yang akan memakai sistem, apa yang dikerjakan oleh sistem, kapan dan dimana sistem akan digunakan. Selama pada tahap ini, tim proyek menganalisis sistem yang digunakan, mengidentifikasi peluang untuk melakukan perbaikan sistem tersebut, dan mengembangkan konsep untuk sistem baru. Tahap ini memiliki tiga langkah:

1. Menyusun strategi analisis untuk mengarahkan usaha tim proyek. Seperti misalnya strategi untuk menganalisis sistem yang telah ada (disebut sebagai *as-is system*) beserta masalah-masalah yang ada dan untuk merancang suatu sistem baru (disebut sebagai *to-be system*).
2. Langkah selanjutnya adalah pengumpulan dokumen (misalnya, melalui wawancara atau kuesioner). Analisis informasi ini bersamaan dengan masukan dari sponsor proyek dan banyak orang lainnya yang mengarah pada pengembangan konsep untuk sistem baru. Konsep sistem baru ini kemudian digunakan sebagai dasar untuk mengembangkan seperangkat model analisis bisnis, yang menggambarkan bagaimana bisnis akan beroperasi jika sistem baru dikembangkan.
3. Analisis, konsep sistem, dan model digabungkan menjadi dokumen yang disebut sebagai proposal sistem, yang dipresentasikan ke sponsor proyek dan divisi yang mengambil keputusan lainnya yang memutuskan apakah proyek dilanjutkan atau tidak.

Proposal sistem adalah dokumen awal yang menjelaskan apa kebutuhan yang harus dipenuhi untuk sistem baru. Karena ini benar-benar langkah pertama dalam perancangan sistem baru, beberapa ahli berpendapat bahwa tidak tepat menggunakan istilah "analisis" sebagai nama untuk fase ini. Beberapa ahli berpendapat lebih baik menggunakan nama "analisis dan desain awal." Tetapi sebagian besar organisasi menggunakan "analisis" untuk fase ini.

2.7.3. *Design* (Desain)

Tahap perancangan / *design* untuk mengidentifikasi bagaimana sistem akan beroperasi dalam perangkat keras (*hardware)*, perangkat lunak (*software*) dan infrastruktur jaringan. Seperti *interface* pengguna, formulir, dan laporan. Serta program yang spesifik seperti *database*, dan file yang akan dibutuhkan. Langkah-langkah dalam tahap perancangan atau *design* menentukan secara tepat bagaimana sistem akan beroperasi. Berikut merupakan fase *design* yang memiliki empat langkah:

1. Strategi perancangan / *design.* Menjelaskan apakah sistem akan dikembangkan oleh *programmer* perusahaan itu sendiri, apakah sistem akan diberikan ke perusahaan lain (biasanya perusahaan konsultan), atau apakah perusahaan akan membeli paket perangkat lunak yang ada.
2. Pengembangan desain yang menggambarkan perangkat keras, perangkat lunak, dan infrastruktur jaringan yang akan digunakan. Dalam kebanyakan kasus, sistem akan menambah atau mengubah infrastruktur yang sudah ada dalam suatu organisasi. Desain antarmuka menentukan bagaimana pengguna akan bergerak melalui sistem (misalnya, metode navigasi seperti menu dan tombol di layar) dan formulir dan laporan yang akan digunakan sistem.
3. Pengembangan *database* dan spesifikasi file. Pengembangan ini menentukan dengan tepat data apa yang akan disimpan dan di mana mereka akan disimpan.
4. Tim analis mengembangkan desain program, mendefinisikan program tersebut dan apa yang akan dilakukan 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 perancangan, analisis kelayakan dan rencana proyek diperiksa ulang dan diperbaiki, dan keputusan lain dibuat oleh sponsor proyek dan divisi yang melakukan persetujuan untuk memutuskan apakah proyek akan dihentikan atau dilanjutkan.

2.7.4. *Implementation* (Implementasi)

Tahap akhir dalam SDLC adalah tahap implementasi, di mana sistem benar-benar dibangun (atau dibeli, dalam kasus paket *design software*). Implementasi adalah fase yang paling mendapat perhatian, karena tahapan ini merupakan bagian terlama dan paling mahal dari proses pengembangan. Fase implementasi ini memiliki tiga langkah:

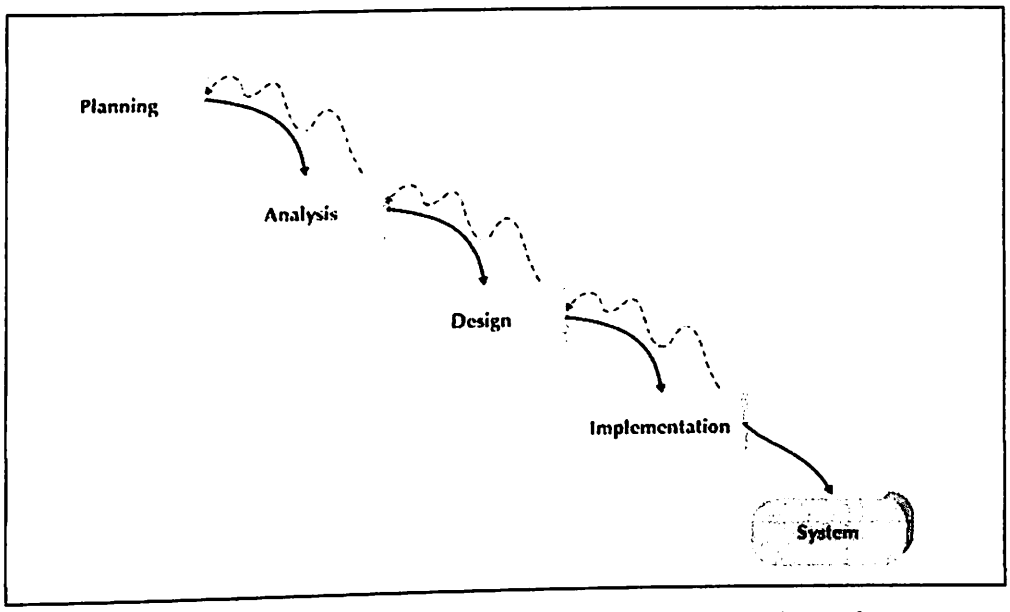
1. Pertama merekonstruksi sistem. Sistem ini dibangun dan diuji untuk memastikan kinerjanya sebagaimana yang telah dirancang. Karena biaya *bugs* bisa sangat besar, pengujian adalah salah satu langkah paling kritis

dalam implementasi. Sebagian besar organisasi memberi lebih banyak waktu dan perhatian untuk menguji dari pada mengubah program di tempat pertama.

2. Instalasi sistem. Instalasi adalah proses dimana sistem lama dimatikan dan yang baru dinyalakan atau dihidupkan. Salah satu aspek terpenting adalah pengembangan rencana pelatihan untuk mengajarkan pengguna cara menggunakan sistem baru dan membantu mengelola perubahan-perubahan yang disebabkan oleh sistem yang baru.
3. Tim analisis menetapkan rencana dukungan untuk sistem. Rencana ini biasanya mencakup tinjauan formal atau informal serta cara sistematis untuk mengidentifikasi perubahan besar dan kecil yang diperlukan untuk sistem tersebut.

2.8 Metode Pengembangan Sistem *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 seperi air mancur. Tetapi, dalam model ini suatu tahapan sulit untuk bergerak mundur, (contohnya dari tahap desain mundur ke tahap analis). Tahapan-tahapan metode *waterfall* menurut Dennis *et al* (2015) dapat dilihat pada Gambar II.5.



Gambar II.5 Metode *Waterfall* Berdasarkan Dennis *et al*

(Sumber: Dennis *et al*, 2015)

2.9 *Flowchart*

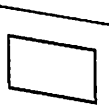
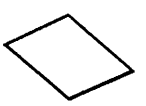
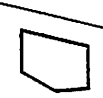
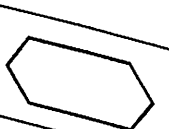
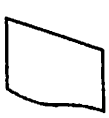
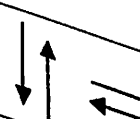
Berdasarkan Sitorus (2015), untuk menggambarkan sebuah proses agar mudah dipahami oleh orang lain maka dibutuhkan alat bantu yang berbentuk diagram alir (*flowchart*). *Flowchart* menggambarkan urutan logika dari suatu prosedur pemecahan masalah, sehingga *flowchart* merupakan langkah-langkah penyelesaian masalah yang dituliskan dalam simbol-simbol tertentu. Diagram alir ini selain dibutuhkan sebagai alat komunikasi, juga diperlukan sebagai dokumentasi.

Berdasarkan Indrajani (2015), *flowchart* merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program. Biasanya mempermudah penyelesaian masalah, khususnya yang perlu dipelajari dan dievaluasi lebih lanjut.

Berdasarkan Jalinus dan Ambiyar (2016), ada dua model *flowchart* yaitu *system flowchart* dan *program flowchart*. *System flowchart* tidak digunakan untuk menggambarkan urutan langkah pemecahan masalah serta hanya untuk menggambarkan prosedur dalam sistem yang dibentuk. Sementara *program flowchart* menggambarkan urutan logika dari suatu prosedur pemecahan masalah.

Tujuan dari *flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapi dan jelas menggunakan simbol-simbol. Simbol-simbol yang terdapat dalam sebuah *flowchart* dapat dilihat pada Tabel II.1.

Tabel II.1 Simbol-simbol *Flowchart*

Simbol	Nama	Keterangan
	Terminal	Menyatakan permulaan atau akhir suatu pogram.
	*Input-Output*	Meyatakan proses *input* dan *output* tanpa tergantung dengan jenis peralatannya.
	*Process*	Menyatkan suatu tindakan (proses) yang dilakukan oleh komputer.
	*Decision*	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya atau tidak.
	*Connector*	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.
	*Off-line Connector*	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.
	*Predafined Process*	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
	*Punched Card*	Menyatakan input berasal dari kartu atau output ditulis ke kartu.
	*Punch Tape*	Menunjukan *input/output* menggunakan pita kertas berlubang.
	*Document*	Simbol yang menyatakan *input* berasal dari dokumen dalam bentuk kertas atau *output* dicetak di kertas.
	*Flow*	Menyatakan jalannya arus suatu proses.

(Sumber: Sitorus, 2015)

2.10 *Unifed Modeling Language* (UML)

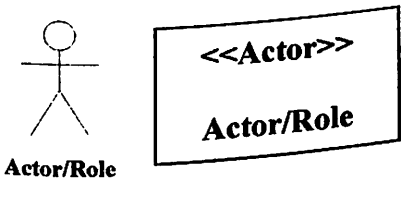
Berdasarkan Mulyani (2016), *Unified Modelling Language* (UML) adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem.

Berdasarkan 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 mempresentasikan data dan hubungan statik yang ada di dalam sebuah sistem informasi. Sedangkan *Behavior diagram* untuk menyediakan para analis dengan sebuah gambaran hubungan yang dinamis antara instansi atau objek yang merepresentasikan sistem informasi bisnis.

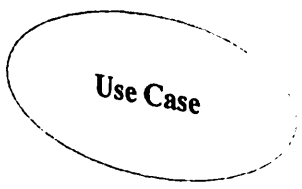
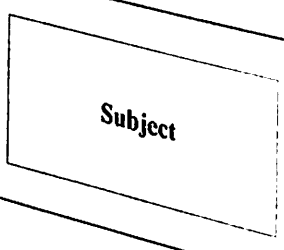
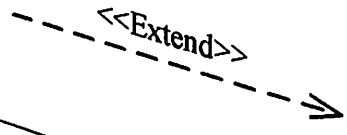
2.10.1 *Use Case Diagram*

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 *use case* dijelaskan pada Tabel II.2

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

Simbol	Fungsi
	▪ Seseorang atau sistem yang mendapatkakn keuntungan dari sistem. ▪ Dilabelkan dengan peran. ▪ Bisa diasosiasikan dengan *actor* lainnya berdasarkan spesialisasi. ▪ Diletakan diluar batas sistem.

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

Simbol	Fungsi
	▪ Merepresentasisan bagian utama dari sistem secara fungsional. ▪ Dapat memperpanjang *use case* lain. ▪ Dapat termasuk *use case* lain. ▪ Diletakan didalam batas sistem. ▪ Dilabelkan dengan frasa kata kerja deskriptif.
	▪ Menyertakan nama subjek didalam maupun diatas. ▪ Merepresentasikan ruang lingkup dari sistem.
	▪ Merepresentasikan eksistensi dari use case untuk menyertakan perilaku optional. ▪ Memiliki anak panah yang digambarkan dari eksistensi *use case* ke *use case* dasar.
	▪ Merepresentasikan *use case* khusus ke yang lebih umum ▪ Memiliki anak panah yang digambarkan dari *use case* khusus ke *use case*.

(Sumber: Dennis *et al.* 2015)

2.10.2 *Activity Diagram*

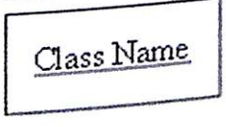
Berdasarkan Dennis *et al* (2015), *activity diagram* digunakan untuk memodelkan perilaku dalam proses bisnis yang independen terhadap objek. Dalam banyak hal, *activity diagram* dapat dipandang sebagai data flow diagram yang digunakan pada konjungsi dengan analisis terstruktur. Walaupun begitu, *activity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

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

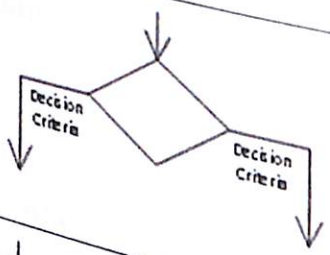
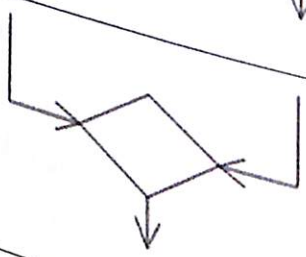
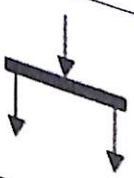
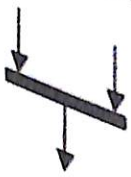
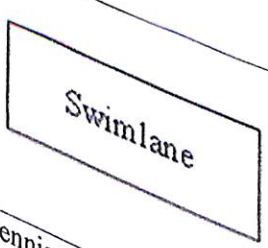
1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana 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* dapat dilihat pada Tabel II.3.

Tabel II.3 Simbol-Simbol *Activity Diagram*

No	Simbol	Nama	Deskripsi
1.		*Action*	Merupakan notasi yang *simple*, dan tidak untuk dianalisis perilakunya. Dinamakan sesuai aksi yg dilakukan.
2.		*Activity*	Digunakan untuk mewakili satu *set* aksi yang dilakukan.
3.		*Object Node*	Digunakan untuk mewakili objek yang terhubung ke satu set arus objek.
4.		*Control Flow*	Menampilkan urutan eksekusi.
5.		*Object Flow*	Menunjukan aliran dari sebuah objek dari satu aktivitas ke aktivitas lainnya.

Tabel II.3 Simbol-Simbol *Activity Diagram*(Lanjutan)

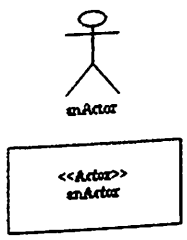
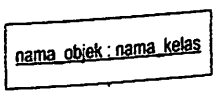
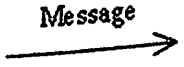
No	Simbol	Nama	Deskripsi
6.		*Initial Node*	Menggambarkan permulaan dari suatu set aktivitas atau aksi.
7.		*Final-activity Node*	Menggambarkan akhir dari setiap aktivitas.
8.		*Final-flow Node*	Menggambarkan akhir aliran control spesifik atau aliran objek.
9.		*Decision Node*	Menggambarkan suatu kondisi untuk memastikan bahwa arah panah atau arah objek hanya memiliki satu jalur ke bawah.
10.		*Merge Node*	Digunakan untuk membawa aliran keputusan yang berbeda ke satu *decision node*.
11.		*Fork Node*	Menggambarkan percabangan dari satu aliran aktivitas.
12.		*Join Node*	Menyatukan beberapa percabangan dari aliran aktivitas.
13.		*Swimlane*	Menggambarkan akhir aliran kontrol spesifik atau aliran objek.

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

2.10.3 *Sequence Diagram*

Berdasarkan Dennis *et al* (2015), *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 menunjukan 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. Berikut simbol yang ada pada *Sequence Diagram* dapat dilihat pada Tabel II.4.

Tabel II.4 Simbol-Simbol *Sequence Diagram*

No	Simbol	Nama	Deskripsi
1.		Aktor	Orang atau sistem yang berasal dari manfaat dan eksternal ke sistem yang berpartisipasi secara berurutan dengan mengirim dan atau menerima pesan.
2.		*Lifeline*	Menyatakan kehidupan suatu objek.
3.		Objek	Berpartisipasi secara berurutan dengan mengirim dan atau menerima pesan yang ditempatkan diatas diagram.
4.		*Execcution Occurence*	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.
5.		*Message*	Pesan yang menggambarkan komunikasi yang terjadi antar objek.

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

No	Simbol	Nama	Deskripsi
6.	Return Value <------	*Message* (*return*)	Pesan yang dikirim untuk diri sendiri.
7.	[GuardCondition] (a Message) ——▶	*Guard Condition*	Menggambarkan tes yang harus dipenuhi untuk pesan yang akan dikirim.
8.	X	*Object Destruction*	Ditempatkan di akhir *lifeline* suatu objek untuk menunjukan akhir dari eksitensi.
9.		*Frame*	Mengindikasikan konteks dari *sequence diagram.*

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

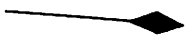
2.10.4 *Class Diagram*

Berdasarkan Dennis *et al* (2015), sebuah *class diagram* digunakan untuk mengatur dan menyimpan informasi yang ada dalam sistem. Selama analisis, *class* mengarah kepada orang, tempat, atau sesuatu yang akan ditangkap oleh sistem. Sebuah *class* digambarkan dengan tiga buah persegi panjang, dimana di bagian atas diisikan dengan nama *class*, di bagian tengah diisi dengan *attribute*, dan di bagian bawah diisikan dengan *operation*. *Sebuah class* merupakan deskripsi dari sekumpulan objek yang memiliki properti *(attribute)*, operasi *(method)*, relasi *(association)*, dan tingkah laku *(behavior)* yang sama. Sebuah *class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda /fungsi). Berikut simbol-simbol yang ada pada diagram kelas dapat dilihat pada Tabel II.5.

Tabel II.5 Simbol-Simbol *Class Diagram*

No.	Simbol	Deskripsi
1.	*Class* 	*Class* berfungsi untuk menggambarkan orang/aktor, tempat, atau sesuatu yang dibutuhkan oleh sistem untuk mendokumentasikan dan menyimpan informasi. Judul *class* ditulis dengan huruf tebal dan berada di tengah atas. *Attribute* ditempatkan di tengah kotak. *Operation* ditempatkan di bawah kotak.
2.	Nama *attribute* 	Mewakili suatu properti untuk mendeskripsikan keadaan dari suatu objek. Dapat diturunkan dari *attribute* lain dengan menempatkan garis miring sebelum nama *attribut.*
3.	Nama operasi/*operation* ( )	Mewakili suatu aksi atau fungsi dari suatu *class.* Dapat diklasifikasikan sebagai konstruktor (*method* yang secara otomatis dipanggil/dijalankan pada sebuah *class*), *query,* atau perubahan operasi.
4	*AssociatedWith* ( )...* 1	Mewakili sebuah hubungan antara beberapa *class* atau *class* dengan *class.* Digambarkan dengan kata kerja atau nama, yang mewakili sebuah hubungan. Tersedia diantara satu atau lebih *class.*
5	*Generalization* 	Merupakan sebuah *taxonomic relationship* antara *class* yang lebih umum dengan *class* yang kebih khusus Mewakili hubungan antara beberapa *class* (*class* banyak).
6	*Aggregation* 0..* *IsPartOf* ▶ 1 	Mewakili bagian hubungan yang logis antara beberapa *class* atau *class* dengan *class.* Merupakan bagian dari *association.*

Tabel II.5 Simbol-Simbol *Class Diagram*

7	*Composition* 1..* *IsPartOf* ▶ 1 	Mewakili bagian hubungan yang fisik antara beberapa *class* atau *class* dengan *class*. Merupakan bagian dari *association*.

(Sumber: Dennis *et al.* 2015)

2.11 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.

2.11.1 *Database*

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

1. Berdasarkan Connolly dan Begg (2010), *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 sistematik 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.6.

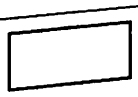
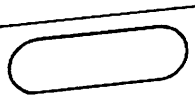
Tabel II.6 Contoh Tabel Barang

Field	*Field Name*	*Type*	*Width*	*Description*
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)

2.11.2 *Entity Relationship Diagram* (ERD)

Berdasarkan Yanto (2016), ERD adalah suatu diagram untuk menggambarkan desain konseptual dari model konseptual suatu basis data relasional. ERD juga merupakan gambaran yang merelasikan antara objek yang satu dengan objek yang lain dari objek dunia nyata yang sering dikenal dengan hubungan antar entitas. Sebagai contoh jika membuat ERD dari system perpustakaan maka bahan sebagai objek ERD bias berupa anggota, buku, peminjaman, pengembalian, dan lain sebagainya. ERD terdiri dari tiga komponen utama yaitu:

	Entitas (*entitiy*)
	Atribut (*attribute*)
	Relasi (*Relationship*)

Berikut ini adalah symbol-simbol dari *Entity Relationship Diagram* (ERD), dapat dilihat pada Tabel II.7.

Tabel II.7 Simbol-simbol *Entity Relationship Diagram*

No.	Simbol	Deskripsi
1.	Entitas Nama_entitas	Entitas merupakan data inti yang akan disimpan
2.	Atribut	*Field* atau kolom data yang butuh disimpan dalam suatu entitas
3.	Atribut kunci primer Nama_atribut	*Field* atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses *record* yang diinginkan.
5.	Relasi Nama_relasi	Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja
6.	Asosiasi	Penghubung antara relasi dan entitas

(Sumber: Rosa dan Shalahuddin, 2015)

2.11.3 Kamus Data

Berdasarkan Indrajani (2015), kamus data adalah katalog fakta tentang data dan kebutuhan informasi suatu sistem informasi. Kamus data terdapat pada tahap analisis dan perancangan. Pada tahap analisis, kamus data berfungsi untuk mendefinisikan data yang mengalir pada sistem. Sedangkan pada tahap perancangan, kamus data digunakan untuk merancang masukan dan keluaran seperti laporan serta basis data. Sumber kamus data yaitu Data *Store* (file-file), Data *Flow* (aliran data), dan Data *Element* yang dinyatakan dalam spesifikasi data dan berasal dari file. Berikut adalah notasi-notasi yang digunakan dalam kamus data dapat dilihat pada Tabel II.8.

Tabel II.8 Notasi-notasi Kamus Data

Notasi	Keterangan
=	*Is composed of*
+	*And*
( )	*Optional (may be present or absent)*
{ }	*Iteration*
[ ]	*Select one of several alternative choices*
**	*Comment*
@	*Identifier (key field) for a store*
|	*Separates alternative choices*

(Sumber: Indrajani, 2015)

Contoh kamus data antara lain:

Name = courtesy-title + first-name + (middle-name) + last-name

Courtesy-title = [Mrs. | Mr. | Miss | Ms. | Dr. | Professor]

First-name = {legal-character}

Middle-name= {legal-character}

Last-name= {legal-character}

Legal-character= [A-Z | a-z |0-9]

2.12 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 basisdata 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.

2.12.1 Tipe Data

Berdasarkan Enterprise Jubilee (2014), Tipe data dibagi menjadi tiga kategori: numerik, tanggal, dan tipe string.

1. Tipe Data Numerik

   Berikut ini daftar data numerik yang digunakan:

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:

a. **DATE** – tanggal dengan format YYY-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 YYY-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:

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.

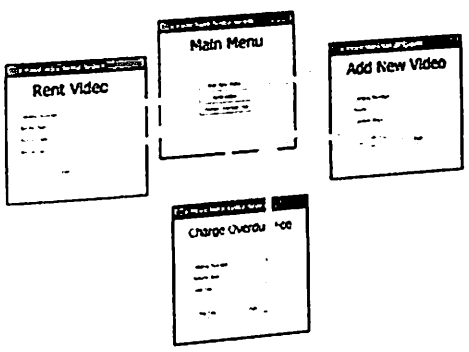
g. **ENUM** – berdasarkan Sutaji (2012), enum adalah sebuah *enumeration*. Sebuah obyek string yang hanya boleh memiliki satu nilai, yang terambil dari 'value1','value2,'...,NULL atau nilai spesial ""*error*. Sebuah *enum* dapat menampung 65535 pilihan nilai.

2.13 *Windows Navigation Diagram* (WND)

Berdasarkan Dennis *et al* (2011, *Windows Navigation Diagram* merupakan *statechart* diagram khusus yang berfokus pada *user interface*.

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.

Berdasarkan Dennis *et al* (2015), *Windows Layout Diagram* adalah suatu gambar yang menyerupai antar-muka pengguna. Biasanya dibuat dalam *Microsoft Visio*. Dengan *visio*, perancang dapat menarik komponen antar-muka dengan cepat ke dalam kanvas. Contoh *Windows Navigation Diagram* dapat dilihat pada Gambar II. 6.



Gambar II.6 *Windows Navigation Diagram*

(Sumber: Dennis *et al*, 2015)

2.14 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 *webserver* 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*.

2.15 *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 *scipt* 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:

1. Mempercepat dan mempermudah pembangunan sebuah aplikasi web.
2. Relatif memudahkan dalam proses *maintenance* karena sudah ada pola tertentu dalam sebuah *framework* (dengan syarat *programmer* mengikuti pola standar yang ada)
3. 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.*

2.16 *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.*
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 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:

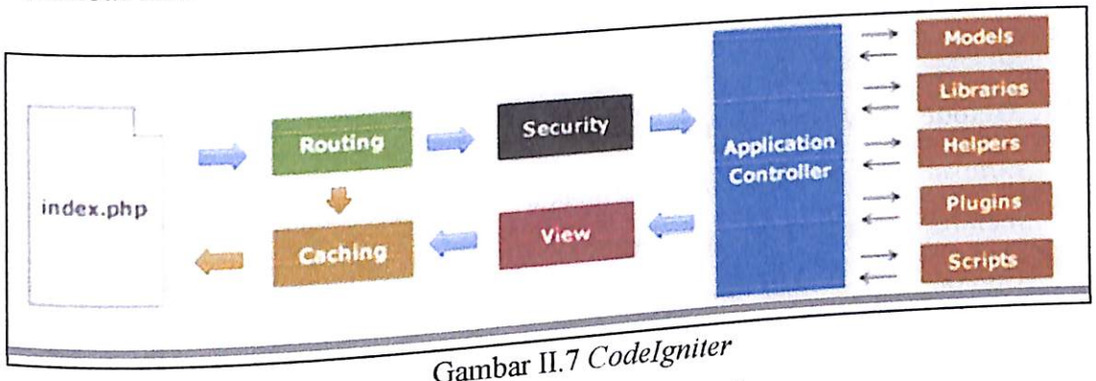
1. Menghemat Waktu
2. Dengan struktur dan *library* yang telah disediakan, membuat *programmer* hanya fokus pada logika pemrograman yang sedang dikerjakan.
3. *Code Reuse*
4. Dengan *codeigniter*, suatu pekerjaan akan memiliki standard yang baku. Sehingga *programmer* dapat menggunakan kembali pada proyek-proyek yang akan datang.
5. Dukungan Komunitas
6. 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.7.



Gambar II.7 *CodeIgniter*

(Sumber: Situs Internet, 2018)

Berikut ini adalah penjelasan cara kerja *codeigniter*:

1. Index.php bertindak sebagai *controller* terdepan, dan menganalisasi *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 di*cache* terlebih dahulu agar dapat ditampilakan di *request* selanjutnya.

2.17 *Black Box Testing*

Berdasarkan Jurnal Mustaqbal dan Firdaus (2015), *black box testing* berfokus pada spesifikasi fungsional dari perangkat lunak. *Tester* dapat mendefinisikan kumpulan *input* dan melakukan pengetesan pada spesifikasi

fungsional program. *Black box testing* cenderung untuk menemukan hal-hal berikut:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).

2.17.1. Metode *Black Box Testing*

1. *Equivalence partitioning* merupakan metode *blackbox testing* yang membagi domain masukan dari suatu program ke dalam kelas-kelas data. Contoh *black box testing* dengan *equivalence partitioning* adalah pemeliharaan data untuk aplikasi bank yang sudah diotomatiskan. Pemakai dapat memutar nomor telepon bank dengan menggunakan mikro komputer yang terhubung dengan *password* yang telah ditentukan dan diikuti dengan perintah-perintah. Data yang diterima adalah:

   Kode area : kosong atau 3 digit
   Prefix : 3 digit atau tidak diawali 0 atau 1
   Suffix : 4 digit
   Password : 6 digit alfanumerik
   Perintah : *check, deposit*, dll,

   Selanjutnya kondisi *input* digabungkan dengan masing-masing *data* elemen dapat ditentukan sebagai berikut:

   Kode area : kondisi *input, boolean*-kode area mungkin ada atau tidak
   Prefix : kondisi *input*, range –nilai ditentukan antara 200 dan 999.
   Suffix : kondisi *input range* > 200 atau tidak diawali 0 atau 1
   Password : kondisi *input* nilai 4 digit
   Perintah : kondisi *input boolean- password* mungkin diperlukan *atau* tidak kondisi *input* nilai dengan 6 karakter *string*
   : kondisi *input set* berisi perintah-perintah yang telah didefinisikan

2.17.2 Perbedaan *White Box* dan *Black Box*

1. *White Box* (*Struktural*)

a. Dilakukan oleh penguji yang mengetahui tentang QA.

b. Melakukan *testing* pada *software*/program aplikasi menyangkut *security* dan *performance* program tersebut (meliputi tes *code*, desain implementasi, *security, data flow, software failure*).

c. Dilakukan seiring dengan tahapan pengembangan *software* atau pada tahap *testing*.

1. *Black Box* (Fungsional)

a. Dilakukan oleh penguji *independent*.

b. Melakukan pengujian berdasarkan apa yang dilihat, hanya fokus terhadap fungsionalitas dan *output*. Pengujian lebih ditunjukan pada desain *software* sesuai standar dan reaksi apabila terhadap celah-celah *bug/vulnerabilitas* program aplikasi tersebut setelah dilakukan *white box testing*.

2.18 Analisis *Pieces*

Menurut Wukil Ragil (2010), metode PIECES adalah metode analisis sebagai dasar untuk memperoleh pokok-pokok permasalahan yang lebih spesifik. Dalam menganalisis sebuah sistem, biasanya akan dilakukan terhadap beberapa aspek antara lain adalah kinerja, informasi, ekonomi, keamanan aplikasi, efisiensi, dan pelayanan pelanggan. Analisis ini disebut dengan PIECES *Analysis* (*Performance, Information, Economy, Control, Efficiency,* and *Service*).

Analisis PIECES ini sangat penting untuk dilakukan sebelum mengembangkan sebuah sistem informasi karena dalam analisis ini biasanya akan ditemukan beberapa masalah utama maupun masalah yang bersifat gejala dari masalah utama. Metode ini menggunakan enam variabel evaluasi, yaitu:

1. *Performance* (kinerja)

   Kinerja merupakan variable pertama dalam metode analisis PIECES. Dimana memiliki peran penting untuk menilai apakah proses atau prosedur yang ada masih mungkin ditingkatkan kinerjanya, dan melihat sejauh mana dan seberapa handalkah suatu sistem informasi dalam berproses untuk

menghasilkan tujuan yang diinginkan. Dalam hal ini kinerja diukur dari *throughput*, yaitu jumlah pekerjaan/*output deliverables* yang dapat dilakukan/ dihasilkan pada saat tertentu.response time, yaitu waktu yang dibutuhkan untuk menyelesaikan serangkaian kegiatan untuk menghasilkan *output/deliverables* tertentu.

2. *Information* (informasi)

   Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki sehingga kualitas informasi yang dihasilkan menjadi semakin baik. Informasi yang disajikan haruslah benar–benar mempunyai nilai yang berguna. Hal ini dapat diukur dengan :

   Keluaran (*outputs*): Suatu sistem dalam memproduksi keluaran.

   Masukan (*inputs*): Dalam memasukkan suatu data sehingga kemudian diolah untuk menjadi informasi yang berguna.

3. *Economic* (ekonomi)

   Menilai apakah prosedur yang ada saat ini masih dapat ditingkatkan manfaatnya (nilai gunanya) atau diturunkan biaya penyelenggaraannya.

4. *Control* (pengendalian)

   Menilai apakah prosedur yang ada saat ini masih dapat ditingkatkan sehingga kualitas pengendalian menjadi semakin baik, dan kemampuannya untuk mendeteksi kesalahan/ kecurangan menjadi semakin baik pula.

5. *Efficiency* (efisiensi)

   Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki, sehingga tercapai peningkatan efisiensi operasi, dan harus lebih unggul dari pada sistem manual.

6. *Service* (layanan)

   Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki kemampuannya untuk mencapai peningkatan kualitas layanan. Buatlah kualitas layanan yang sangat user friendly untuk *end–user* (pengguna) sehingga pengguna mendapatkan kualitas layanan yang baik.

MILIK PERPUSTAKAAN STMI
Membaca : Ibadah, Mengambil : Dosa

BAB III
METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metodologi penelitian adalah proses atau cara ilmiah untuk mendapatkan data yang akan digunakan untuk keperluan penelitian. Metodologi penelitian juga dikenal sebagai metode ilmiah dalam mencari, mengembangkan dan menguji suatu kebenaran pengetahuan. Metodologi penelitian juga bertujuan untuk membuat penelitian menjadi lebih terarah.

3.2 Identifikasi Masalah

Identifikasi masalah adalah salah satu proses penelitian yang boleh dikatakan paling penting diantara proses lain. Masalah penelitian akan menentukan kualitas dari penelitian, bahkan juga menentukan apakah sebuah kegiatan bisa disebut penelitian atau tidak. Masalah penelitian secara umum bisa ditemukan lewat studi literatur atau lewat pengamatan lapangan (observasi, survei dan sebagainya).

3.3 Pengumpulan Data

Dalam melakukan pengumpulan data terdapat 2 jenis data, yaitu data primer dan data sekunder. Sumber dari data-data ini berasal dari tempat yang diamati oleh penulis pada praktek kerja lapangan pada PT Baneex Indonesia.

1. Data Primer

   Data primer adalah data yang diperoleh secara langsung dari perusahaan dimana pengumpulan data atau informasi dilakukan langsung dari objek yang diteliti. Data-data tersebut adalah data yang digunakan dalam proses berlangsungnya sistem persediaan barang jadi diantaranya, wawancara dan hasil pengamatan

2. Data Sekunder

   Data sekunder adalah data yang diperoleh dari data yang tersedia dan telah terlebih dahulu dikumpulkan dan dilaporkan oleh pihak 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.

3.4 Metode Pengumpulan Data

Metode pengumpulan data merupakan sebuah cara yang dapat digunakan oleh peneliti untuk mengumpulkan data dari berbagai sumber, baik sumber eksternal maupun internal. Dalam pengumpulan data penelitian ini, digunakan beberapa teknik yaitu:

1. Studi lapangan

   Studi lapangan adalah usaha melakukan pengumpulan data secara langsung pada objek yang dilakukan dengan cara wawancara, yaitu mencari data yang dibutuhkan secara langsung dengan memberikan pertanyaan terhadap segala hal yang diperlukan pada penyusunan tugas akhir ini. Wawancara ini dilakukan kepada pegawai pada divisi inventori dan beberapa pegawai pada bagian *Office* PT Baneex Indonesia. Data yang didapatkan melalui wawancara diantaranya mengenai proses pengendalian barang jadi yang terjadi pada PT. Baneex Indonesia. Dan untuk memvalidasi hasil wawancara dilakukan pengamatan
2. Studi kepustakaan

   Studi kepustakaan dilakukan dengan membaca buku dan jurnal 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, artikel ilmiah dalam bentuk *hardcopy* maupun artikel yang didapatkan melalui internet.

3.5 Metode Pengembangan Sistem

Menurut Dennis (2015) *System Development Life Cycle* (SDLC) memiliki empat perangkat fase dasar yaitu *planning*, *analysis*, *design*, dan *implementation*. Beberapa cara dalam membangun sebuah sistem mungkin dapat mempunyai perbedaan, tetapi hampir semua proyek memiliki elemen dari empat fase tersebut. Masing-masing fase tersebut tersusun dari beberapa langkah-langkah yang menghasilkan *deliverable* atau hasil kegiatan seperti beberapa dokumen spesifik

dan file yang menjelaskan pemahaman tentang proyek. Pada penelitian ini untuk mengatasi masalah yang ada pada sistem, diputuskan untuk merancang dan mengembangkan sistem dengan menggunakan metode *waterfall*.

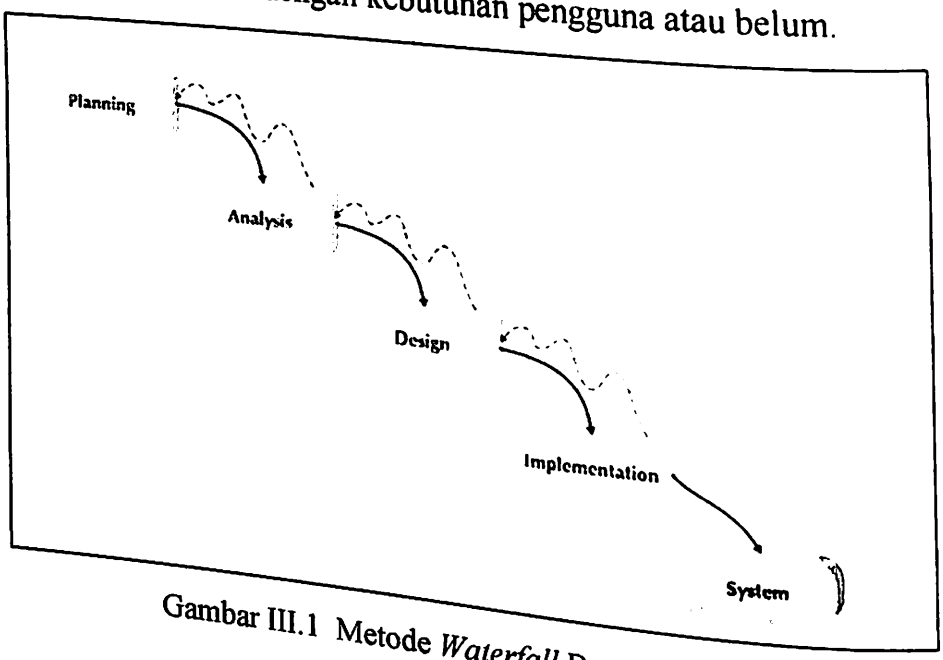
Metode *waterfall* sering juga disebut model sekuensial linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*). Metode *waterfall* menyediakan pendekatan alur hidup perangkat lunak dengan terurut dari fase awal (*planning*) hingga fase akhir (*implementation*), dan pada setiap fase pihak pengembang dan sponsor akan selau ada kontak untuk mendapatkan persetujuan sebelum masuk pada tahap selanjunya (Dennis, 2015).

Tahapan-tahapan dalam *waterfall* adalah sebagai berikut:

a. Identifikasi kebutuhan pengguna (*planning)*
   Pengembang memahami sistem yang berjalan yang ada pada divisi Inventori kemudian pengembang menarik permasalahan yang terdapat pada sistem tersebut dengan dilakukan perbaikan lebih lanjut.
b. Analisis kebutuhan perangkat lunak (*Analysis*)
   Pengembang berdiskusi dengan kepala divisi inventori untuk mempelajari alasan dan menganalisis kelayakan pada sistem tersebut yang nanti akan diimplementasikan pada perusahaan dan mengalokasikan sumber daya untuk mengembangkan sistem tersebut.
c. Desain (*Design*)
   Pengembang membuat sebuah desain program perangkat lunak seperti struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean serta dokumentasinya dari sistem yang telah dijelaskan oleh *manager* dan staf divisi Inventori
4. Pembuatan Kode Program (*Implementaion*)
   Pengembang merealisasikan desain yang sudah dirancang sebelumnya ke dalam bahasa pemrograman dan membuat database untuk mendukung pemrosesan data berikut dengan pengujian sistem tersebut.

5. Sistem (*System*)

Pada tahap ini pengembang melakukan migrasi data dan pemeliharaan, yang dapat digunakan untuk menentukan apakah sistem atau perangkat lunak yang sudah dibuat sesuai dengan kebutuhan pengguna atau belum.



Gambar III.1 Metode *Waterfall* Berdasarkan Dennis *et al*
(Sumber: Dennis *et al*, 2015)

3.6 *Literature Review*

Literature review tidak hanya bermakna membaca literatur, tapi lebih ke arah evaluasi yang mendalam dan kritis tentang penelitian sebelumnya pada suatu topik. *Literature review* memiliki peran penting dalam membuat suatu tulisan ataupun karangan ilmiah, karena dapat memberikan ide dan tujuan tentang topik penelitian yang akan dilakukan. Pada umumnya berisi ulasan, rangkuman & pemikiran penulis tentang beberapa pustaka (buku, jurnal, majalah) yang berkaitan dengan topik yang dibahas. Menurut (romisatriowahono.net), yang disebut dengan literatur ilmiah dapat berupa:

1. Paper dari *Journal* Ilmiah
2. Paper dari *Conference* (*Proceedings*).
3. *Thesis* dan Disertasi.
4. *Report* (laporan) dari organisasi yang terpercaya.
5. Buku *Textbook*.

Berikut adalah *literature review* dari beberapa jurnal ilmiah:

Tabel III.1 Literature Review

No	Pengarang/ Tahun	Topik Penelitian	Permasalahan	Metode yang digunakan	Kesimpulan Penelitian	Hasil Review (Perbandingan)
1.	Agus Wantoro, Imam Alkarim	Aplikasi pengendalian persediaan *spare part* traktor dengan metode *buffer stock* dan *reorder point* (ROP) di gudang cabang tanjung karang	Di gudang cabang Tanjung Karang hanya meramalkan beberapa jumlah kebutuhan *spare part* dari rata-rata penjualan dan maksimal penjualan, belum adanya persediaan pengaman dan tidak dapat diketahui kapan saatnya pemesanan harus diadakan kembali, sehingga menimbulkan terjadinya kelebihan dan kehabisan stok	Metode *Buffer stock* dan ROP (*reorder point)*	Menerapkan persediaan pengaman (*Buffer Stock*) yang berfungsi untuk melindungi atau menjaga kemungkinan terjadinya kekurangan stok. Dan titik atau batas (*Reorder Point* / ROP) dari jumlah persediaan yang ada pada suatu saat pemesanan harus diadakan kembali	Dalam melakukan *review* pada jurnal dan tugas akhir, penulis menemukan beberapa persamaan yaitu persamaan dalam penggunaan metode *Buffer Stock* dan *Reorder Point* yang bertujuan untuk melindungi dari kekurangan stok. Perbedaan yang terdapat dalam jurnal adalah penggunaan program dalam pengembangan sistem yang mana jurnal menggunakan VB.net sedangkan penulis menggunakan *Python* dan MySQL. dan perbedaan selanjutnya adalah penulis berfokus pada produk *collar*. sedangkan dalam jurnal berfokus pada *spare part* traktor

(Sumber, Hasil Analisis 2018)

Tabel III.1 *Literature Review* (lanjutan)

No	Pengarang/ Tahun	Topik Penelitian	Permasalahan	Metode yang digunakan	Kesimpulan Penelitian	Hasil Review (Perbandingan)
2.	*Rahmad Harjono, Yudha Prasetyawan. 2015*	*Perancangan tata letak gudang untuk meminimumkan jumlah produk yang tidak tertampung dalam blok dan efisiensi aktivitas perpindahan barang di divisi penyimpanan produk jadi PT. ISM BOGASARI FLOUR MILLS SURABAYA*	*Rendahnya rasio luas blok terhadap luas gudang, ukuran blok-blok penyimpanan yang terlalu besar, serta kebijakan penyimpanan secara randomized menyebabkan banyaknya produk-produk yang tidak tertampung dalam blok-blok penyimpanan dan biaya material handling yang cukup tinggi.*	*Metode SS (Safety Stock) dan ROP (reorder point)*	a. *Rasio luas blok terhadap luas gudang existing masih cukup rendah yaitu sebesar 49,1925% dari luas gudang keseluruhan* b. *Besarnya utilitas blok pada kondisi existing sebesar 88,46%* c. *Jumlah produk out of block pada kondisi existingcukup tinggi yaitu sebesar 11.12 % dari jumlah keseluruhan stok barang dalam gudang* d. Terdapat dua hal yang dilakukan untuk mengurangi jumlah produk *out of block* yaitu menambah kapasitas ruang penyimpanan dan menentukan besarnya ukuran blok . Penambahan kapasitas dilakukan untuk meminimumkan jumlah produk yang tidak tertampung dalam blok sedangkan penentuan ukuran blok dilakukan untuk meminimumkan jumlah produk sisa dalam blok	Persamaan penulisan jurnal dengan laporan penulis adalah penggunaan metode yang sama yaitu dengan menggunakan *Reorder Point*. Persamaan kedua adalah berfokus pada pengendalian persediaan barang jadi. Perbedaan yang terdapat dalam jurnal adalah pembahasan masalah, yang mana jurnal berfokus pada tata letak penyimpanan di gudang sedangkan penulis pada pengendalian persediaan barang jadi perbedaan kedua adalah lokasi penelitian.

(Sumber, Hasil Analisis 2018)

3.7 Kerangka Penelitian

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

1. Studi Pendahuluan

   Pada tahap awal dilakukan sebuah studi pendahuluan yaitu dengan membaca buku literatur, *browsing internet,* membaca artikel serta sumber-sumber lain dalam lingkup perkuliahan maupun di luar lingkup perkuliahan yang semuanya berhubungan dengan judul dan permasalah tugas akhir yang di ambil oleh penulis selain itu, penulis juga melakukan observasi dan wawancara pada divisi Inventori PT Baneex Indonesia dengan jangka waktu 1 bulan penuh.

2. Analisis Kebutuhan Sistem Informasi

   a. Analisis Sistem Berjalan

   Pada tahap ini dilakukan pengamatan bagaimana cara sistem pada pengendalian persediaan barang jadi berjalan.

   - Analisis dokumen masuk dan keluar yang terlibat pada proses pengendalian persediaan barang jadi.
   - Melakukan masalah yang terjadi pada divisi Inventori yang berfokus pada pengendalian persediaan barang jadi dalam mengidentifikasi suatu masalah penulis melakukan beberapa cara seperti:Analisis proses bisnis untuk identifiasi sistem yang berjalan

   b. Identifikasi Masalah

   Pada tahap ini dilakukan cara dalam mengidentifikasi suatu masalah

   - Analisis terhadap hasil dalam bentuk transkip wawancara dan hasil observasi yang dilakukan pada pihak-pihak yang terkait dalam proses pengendalian persediaan barang jadi.

- Menggunakan analisis P.I.E.C.E.S untuk identifikasi permasalahan.

3. Desain

Pada tahap ini dilakukan perancangan dalam pembuatan sistem dengan menggunakan UML diagram yang bertujuan untuk memodelkan sistem. Dalam pembuatan UML terdapat beberapa diagram seperti:

a. Perancangan sistem *flowmap* usulan

b. Modelling sistem dengan UML :

- *Use case Diagram* yang bertujuan untuk mendeskripsikan sebuah interaksi antara *user* (actor) dengan sistem yang akan di rancang.
- *Activity Diagram* yang bertujuan untuk menggambarkan *workflow* atau aliran kerja pada proses bisnis.
- *Class Diagam* yang bertujuan untuk mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat di antara objek-objek tersebut.
- *Sequence Diagram* bertujuan untuk menggambarkan kelakukan atau prilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang diterima dan dikeluarkan antar objek.
- *Deployment Diagram* yang bertujuan untuk memvisualisasikan, menspesifikasikan dan mendokumentasi proses yang terjadi pada suatu sistem perangkat lunak yang akan dibangun.

c. Pemodelan Data

- Pemodelan data dengan ERD yang bertujuan untuk memodelkan suatu data dan hubungan data
- Rancangan basis data dengan kamus data dengan tujuan untuk memvalidasi diagram alir data dalam hal kelengkapan dan keakuratan data.

d. Perancangan User Interface

- *Windows Navigation Diagram* (WND) yang bertujuan dalam memberikan suatu gambaran navigasi yang terdapat dalam sistem

yang akan dibuat, pembuatan *Entity Relationship Diagram* (ERD) serta pembuatan kamus data

4. Implementasi Sistem
   a. Melakukan kodifikasi dengan bahasa pemrograman PHP versi 7.3 dengan menggunakan *framework Code Igniter* versi 3.1.10 dan *database MariaDB* 10.3.16
   b. Melakukan pengujian sistem dengan metode *black box testing* yang bertujuan untuk menguji fungsionalitas dan output yang dihasilkan aplikasi.
   c. Membuat laporan pengembangan Sistem Informasi Persediaan Dengan Metode *Buffer Stock* Untuk Mengendalikan Persediaan Barang Jadi Pada PT. Baneex Indonesia

Berikut adalah gambar III.1 prosedur pembuatan program untuk menyelesaikan masalah dalam Tugas Akhir ini:

MILIK PERPUSTAKAAN STMI
Membaca : Ibadah, Mengambil : Dosa

BAB IV
PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Tentang Perusahaan

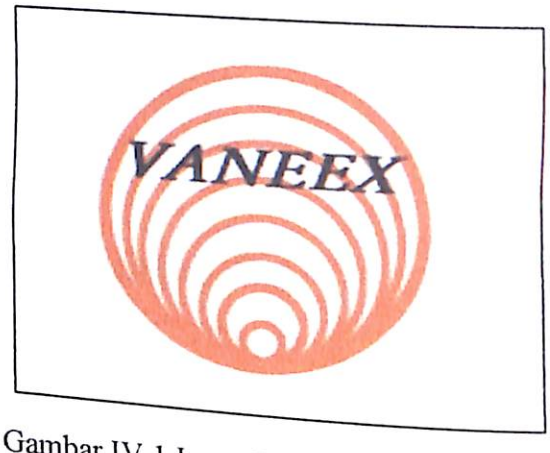
PT Baneex Indonesia adalah perusahaan yang bergerak di bidang Manufaktur Otomotif yang memproduksi komponen *Spring, Bending* dan *Stamping*. PT Baneex Indonesia didirikan pada bulan Januari 2007 yang berlokasi di Jalan Express Blok WW No 6 RT 002 RW 024 Bojong Rawa Lumbu, Bekasi. Perusahaan ini sebagai 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 di uraikan 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 *Head Office*	:	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	:	- *Spring* - *Bending* - *Stamping*

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.

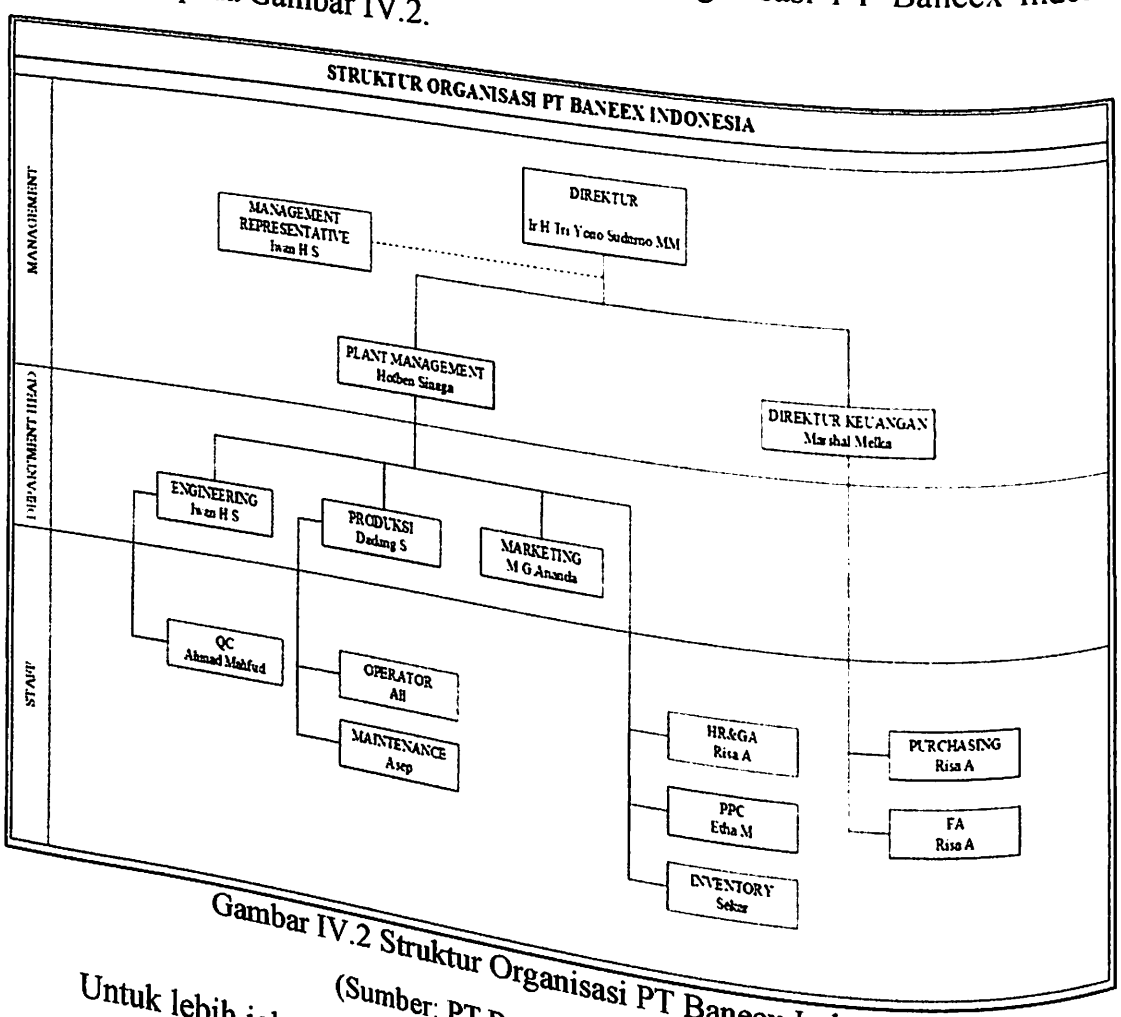
4.4 Struktur Organisasi Perusahaan

Struktur Organisasi adalah bagaimana pekerjaan dibagi, dikelompokan 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, dikelompokan dan dikoordinasikan (Robbins dan Coulter, 2007).
2. Struktur organisasi didefinisikan sebagai mekanisme-mekanisme formal dengan mana organisasi dikelolah (Handoko, 2003).
3. Struktur organisasi adalah pola formal mengelompokan orang dan pekerjaan (Gibson dkk, 2002).

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, kepegaiwaian, dan kesekertariatan.
- 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 mengenasi 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 penggunanya.
- 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 juka mengetahui adanya ketidaknormalan mesin atau perlengkapan pendukungnya atau karena kualitas form 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 memberikkan 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

8. ***Quality Control***

a. Tugas dan Tanggung Jawab

- Melakukan pekerjaan sesuai standar kerja yang telah ditentukan
- Memeriksa material yang datang dari *supplier*.
- Memeriksa form 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.
- Mengirimkan barang jadi ke *Customer*
- Membuat jadwal produksi.

b. Wewenang

- Memberikan saran-saran perbaikan kepada divisi lain guna peningkatan priduktifitas 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 untuk kebutuhan produksi
- Melakukan *update* kartu stock baik untuk material, *spare part* dan barang jadi.
- 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.


Certificate of Registration

This is to Certify That
Quality Management System of

PT. BANEEX INDONESIA

HEAD OFFICE: JL. EXPRESS 5 BLOK WW NO. 6, RT.002, RW. 024,KEL. BOJONG RAWA LUMBU, KEC. RAWA LUMBU,KAB. BEKASI, INDONESIA.

FACTORY ADDRESS: JL. TOYOGIRI RT.002, RW.03, KP. LEGON, KEL. JATIMULYA, KEC. TAMBUN SELATAN, KAB. BEKASI, INDONESIA

has been assessed and found to conform
to the requirements of

ISO 9001:2015

for the following scope :

MANUFACTURING OF SPRING, WIRE BENDING AND STAMPING PARTS FOR AUTOMOTIVE INDUSTRY

Certificate No.: 17IQBU32
Initial Registration Date: 27.10.2017
Date of Expiry* : 26.10.2020
Ist Surveillance Due : 27.09.2018
Issuance Date : 27.10.2017
2nd Surveillance Due : 27.09.2019

DIRECTOR

ROHS

ROHS Certification Pvt. Ltd.

ACCREDITED

IAF

ISO 9001:2015

Gambar IV.3 Sertifikat ISO 9001:2015
(Sumber: PT Baneex Indonesia, 2018)

4.6 Analisis Proses (*Flowmap* Sistem yang Sedang Berjalan)

Bagan alir (*flowmap)* menunjukkan arus dari pekerjaan secara keseluruhan dari sistem. Bagan alir ini digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. *Flowmap* ini menjelaskan urut-urutan dari prosedur-prosedur yang ada di dalam sistem. Bagan alir sistem menunjukkan apa yang dikerjakan di sistem. Adapun proses-proses tersebut dapat dijelaskan sebagai berikut:

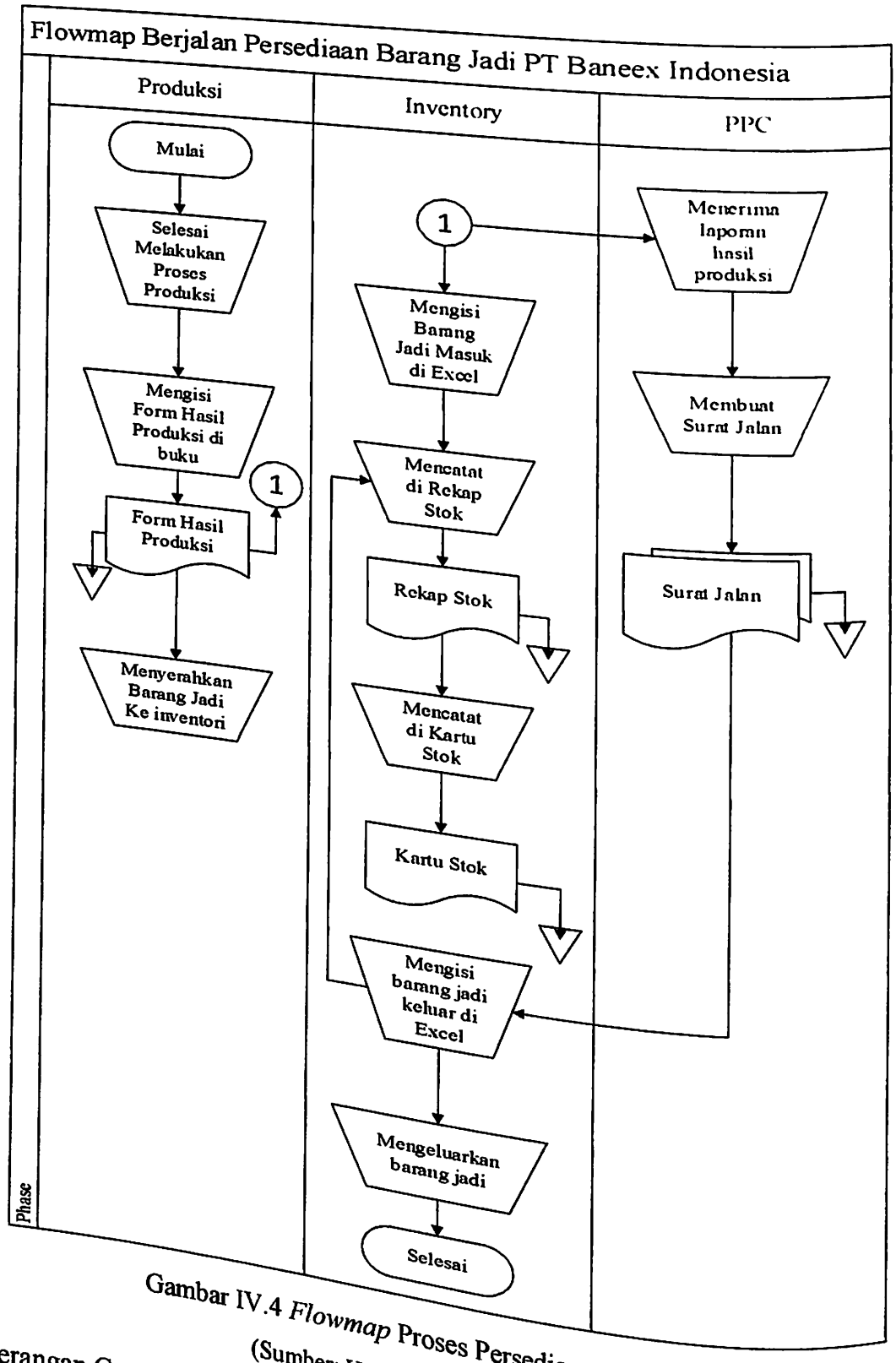
4.6.1. Prosedur Sistem Persediaan Barang Jadi

Berikut merupakan penjelasan dari sistem informasi persediaan barang jadi yang sesuai dengan prosedur pada PT Baneex Indonesia:

1. Setelah selesai proses produksi di bagian produksi akan mencatat hasil produksi di buku hasil produksi yang lalu diberikan ke divisi inventori.
2. Divisi produksi menginformasikan hasil produksi ke bagian PPC dan menyimpan barang jadi ke gudang.
3. Divisi Inventori mencatat barang yang masuk ke rekap kartu stok berdasarkan form hasil produksi lalu staf inventori melakukan pencatatan kembali di kartu stok berdasarkan nama barang.
4. Divisi PPC lalu membuat surat jalan untuk pengiriman barang ke customer.
5. Divisi inventori mencatat dan mengeluarkan barang jadi dari gudang dan memperbarui rekap kartu stok dan kartu stok berdasarkan nama barang yang keluar.

4.6.2. *Flowmap* Proses Persediaan Barang Jadi

Flowmap pada Gambar IV.4 menjelaskan tentang Proses sistem informasi persediaan barang jadi yang berjalan pada PT Baneex Indonesia:



Gambar IV.4 *Flowmap* Proses Persediaan Barang Jadi
(Sumber: Hasil Analisis, 2019)

Keterangan Gambar IV.4:

1. Setelah selesai proses produksi di bagian produksi akan mencatat hasil produksi di buku hasil produksi yang lalu diberikan ke divisi inventori.

2. *Divisi produksi menginformasikan hasil produksi ke bagian PPC dan menyimpan barang jadi ke gudang.*
3. *Divisi Inventori mencatat barang yang masuk ke rekap kartu stok* berdasarkan form hasil produksi lalu staf inventori melakukan pencatatan kembali di kartu stok berdasarkan nama barang.
4. Divisi PPC lalu membuat surat jalan untuk pengiriman barang ke customer.
5. Divisi inventori mencatat dan mengeluarkan barang jadi dari gudang dan memperbarui rekap kartu stok dan kartu stok berdasarkan nama barang yang keluar.

4.7 Analisis Dokumen Persediaan Barang Jadi

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 persediaan barang jadi usulan.

Dalam sistem informasi persediaan barang jadi ini dibutuhkan agar proses dapat berjalan sesuai dengan target yang ada. Dokumen-dokumen yang terlibat dalam sistem informasi persediaan barang jadi ini adalah sebagai berikut :

4.7.1 Dokumen Persediaan Barang Jadi

Dalam persediaan barang jadi, tentunya terdapat dokumen-dokumen yang berperan untuk membantu staff inventori. Dokumen yang terlibat pada divisi inventori ini tidak hanya satu dokumen. Tetapi ada beberapa macam dokumen. Berikut ini adalah dokumen-dokumen yang terlibat dalam sistem penyimpanan barang jadi:

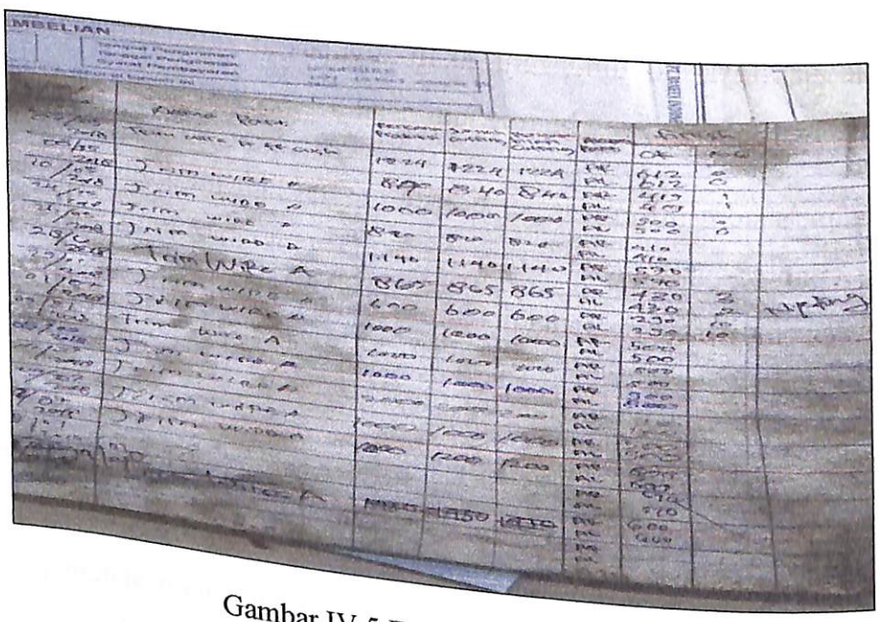
1. Form Hasil Produksi

   Form hasil produksi ini diisi oleh staff produksi untuk mencatat arus barang yang keluar maupun masuk ke dalam produksi. Seperti untuk mencatat berapa jumlah barang jadi yang keluar, berapa jumlah bahan baku yang

masuk. Form hasil produksi diisi di buku dan tidak rangkap. Berikut adalah keterangan yang terdapat dalam form hasil produksi yang digunakan oleh Operator produksi untuk mencatat barang yang masuk maupun keluar:

Nama Barang	: Untuk mencatat nama barang yang disimpan dalam gudang.
Rencana Produksi	: Untuk mengetahui berapa barang yang akan dibuat.
Tanggal	: Kolom tanggal untuk mencatat kapan tanggal produksi.
Jumlah *Cutting*	: Untuk mengetahui barang yang sudah jadi.
Pengambilan *Cutting*	: Untuk mengetahui berapa barang jadi yang diambil untuk dikirim.
Alokasi Part	: Untuk mengetahui jenis part yang jadi terdiri dari AR dan AL.
Finish	: Untuk mengetahui barang *Good* atau *Not Good.*

Untuk lebih jelasnya, gambar dari form hasil produksi dapat dilihat pada Gambar IV.5.



Gambar IV.5 Form Hasil Produksi
(Sumber: PT Baneex Indonesia, 2017)

2. Rekap Stok

Rekap kartu stok berfungsi sebagai dokumentasi persediaan barang jadi, rekap kartu stok diisi berdasarkan tanggal. Rekap stok barang jadi harus selalu diperbarui bila ada barang masuk atau keluar. Kartu stok PT Baneex

Indonesia tidak rangkap dan diisi setidaknya satu kali sehari oleh staf inventori. Berikut adalah keterangan yang terdapat dalam rekap kartu stok.

Data *update* per tgl : Untuk mencatat tanggal kapan rekap kartu stok dibuat

Model : Untuk mengetahui berapa barang yang akan dibuat.

Part Name : Untuk mencatat nama part yang di simpan

Part Number : Untuk mencatat nomor part sesuai nama part sebelumnya.

*Min Stock** : Untuk mengetahui berapa minimal stok barang jadi yang ada di gudang.

Jumlah Stok : Untuk mengetahui jumlah stok yang ada di gudang.

Keterangan : Untuk tambahan keterangan jika diperlukan.

Dilaporkan Oleh : Untuk mengetahui siapa yang membuat rekap stok

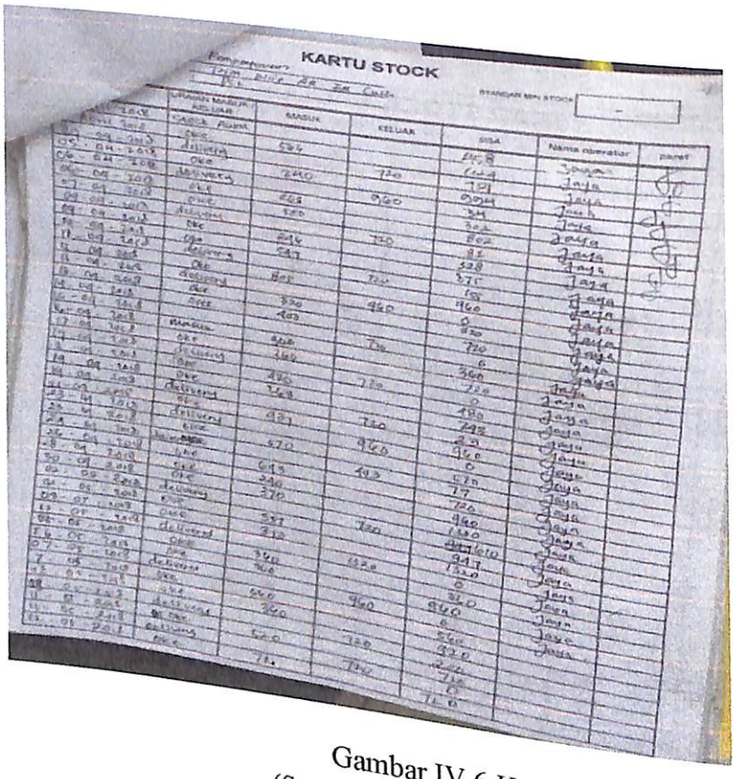
REKAP STOCK PRODUK

Data Up date per tgl : .....

Model	Part Name	Part Number	Min Stock*	Jml Stok	KETERANGAN

Tanggal : untuk mencatat waktu barang masuk maupun keluar
Uraian : kolom untuk penjelasan barang keluar atau barang masuk
Masuk : jumlah barang jadi yang masuk
Keluar : jumlah barang jadi yang keluar
Sisa : stok yang tersedia di gudang
Nama Operator : nama operator yang mengisi form hasil produksi
Paraf : validator dalam pengecekan stok
Gudang : untuk mengetahui letak gudang
Nama Barang : untuk mengetahui nama barang yang disimpan
Satuan Barang : untuk mengetahui satuan barang yang disimpan

Berikut ini adalah gambar dari kartu stok yang dapat dilihat pada Gambar IV.6



Gambar IV.6 Kartu Stok
(Sumber: PT Baneex Indonesia, 2018)

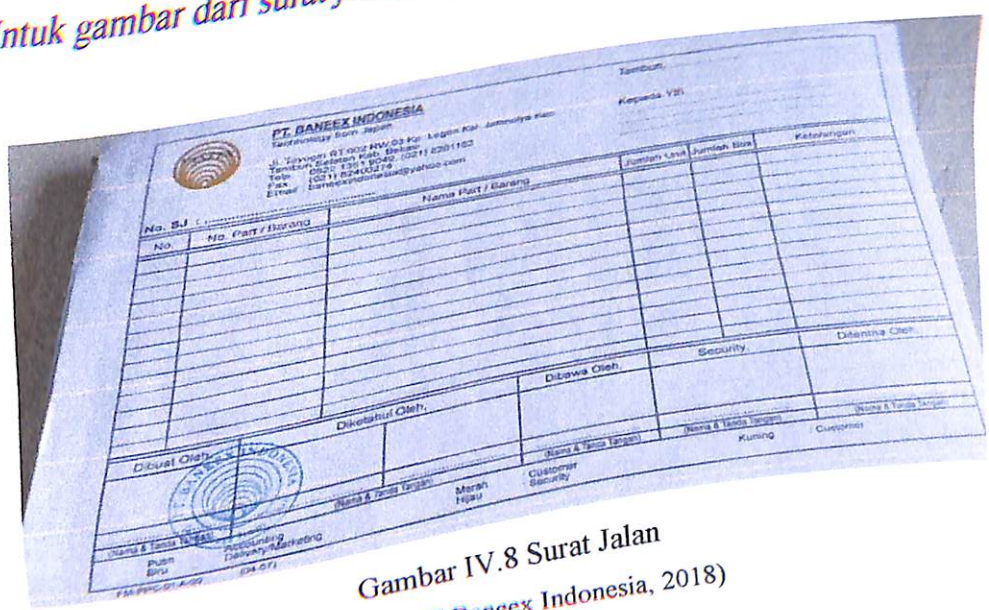
4. Form Surat Jalan

Surat Jalan berfungsi sebagai dokumentasi pengiriman barang ke pelanggan. Surat jaln dibuat oleh PPC. Kartu stok PT Baneex Indonesia tidak rangkap dan diisi dan dikeluarkan sesuai dengan rencana pengiriman barang jadi.

Surat jalan dibuat setidaknya satu kali sehari oleh staf PPC. Berikut adalah keterangan yang terdapat dalam surat jalan:

Kepada Yth	:untuk pelanggan dan alamat pelanggan
No. SJ	:untuk mencatat no surat jalan yang diterbitkan
Nama Barang	:untuk mencatat nama barang yang disimpan akan dikirim
No. Part Barang	: untuk mencatat nomor part yang akan dikirim
Jumlah unit	:untuk mengetahui berapa jumlah unit barang yang akan dikirim
Jumlah box	: untuk mengetahui berapa jumlah box yang akan dikirim
Keterangan	: untuk menambahkan keterangan yang dibutuhkan
Tanggal	:kolom tanggal untuk mencatat kapan tanggal surat jalan dibuat
Dibuat oleh	:untuk mengetahui staff yang membuat surat jalan
Diketahui oleh	:untuk atasan yang mengetahui pengiriman
Dibawa oleh	:untuk mengetahui siapa pengantar yang mengirim barang jadi
Keterangan	:untuk menambahkan .
Security	: untuk mengetahui petugas yang berjaga
Diterima oleh	: untuk pelanggan yang menerima pengiriman

Untuk gambar dari surat jalan, dapat dilihat pada Gambar IV.8



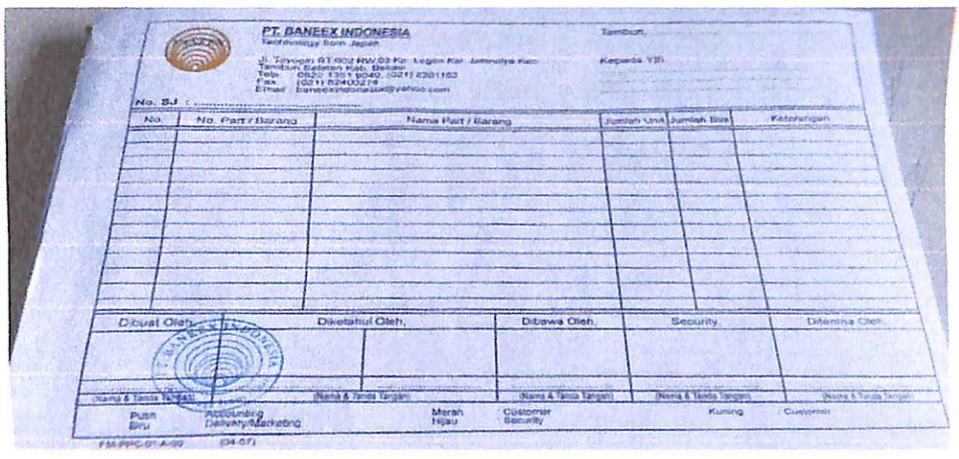
Gambar IV.8 Surat Jalan

(Sumber: PT Baneex Indonesia, 2018)

Surat jalan dibuat setidaknya satu kali sehari oleh staf PPC. Berikut adalah keterangan yang terdapat dalam surat jalan:

Kepada Yth	:untuk pelanggan dan alamat pelanggan
No. SJ	:untuk mencatat no surat jalan yang diterbitkan
Nama Barang	:untuk mencatat nama barang yang disimpan akan dikirim
No. Part Barang	: untuk mencatat nomor part yang akan dikirim
Jumlah unit	:untuk mengetahui berapa jumlah unit barang yang akan dikirim
Jumlah box	: untuk mengetahui berapa jumlah box yang akan dikirim
Keterangan	: untuk menambahkan keterangan yang dibutuhkan
Tanggal	:kolom tanggal untuk mencatat kapan tanggal surat jalan dibuat
Dibuat oleh	:untuk mengetahui staff yang membuat surat jalan
Diketahui oleh	:untuk atasan yang mengetahui pengiriman
Dibawa oleh	:untuk mengetahui siapa pengantar yang mengirim barang jadi
Keterangan	:untuk menambahkan .
Security	: untuk mengetahui petugas yang berjaga
Diterima oleh	: untuk pelanggan yang menerima pengiriman

Untuk gambar dari surat jalan, dapat dilihat pada Gambar IV.8


PT. BANEEX INDONESIA

Tambun,

Kepada Yth

No. SJ :

No.	No. Part / Barang	Nama Part / Barang	Jumlah Unit	Jumlah Box	Keterangan

Dibuat Oleh,	Diketahui Oleh,	Dibawa Oleh,	Security	Diterima Oleh,
(Nama & Tanda Tangan)	(Nama & Tanda Tangan)	(Nama & Tanda Tangan)	(Nama & Tanda Tangan)	(Nama & Tanda Tangan)

Putih Accounting
Biru Delivery/Marketing
Merah Customer Service
Hijau
Kuning
Customer

FM-PPC-01.A-09 (04-07)

Gambar IV.8 Surat Jalan

(Sumber: PT Baneex Indonesia, 2018)

4.8 Analisis Permasalahan

Permasalahan yang ada pada sistem persediaan barang jadi yang berjalan pada PT Baneex Indonesia dapat dilihat pada Tabel IV.3 berikut:

Tabel IV.1 Analisis Permasalahan

Analisis	**Uraian**	**Sebab**	**Akibat**	**Solusi**
Performance (Kinerja)	Untuk melakukan pengawasan persediaan barang jadi masih dilakukan secara manual.	Belum ada sistem yang terkomputer-isasi untuk dapat membantu pengawasan proses persediaan barang jadi pada PT Baneex Indonesia	Sistem informasi persediaan barang tidak ter-*monitor* dengan baik.	Merancang dan membangun suatu sistem informasi yang membantu proses *monitoring* persediaan barang jadi

Tabel IV.1 Analisis Permasalahan (Lanjutan)

Information (Informasi)	Data sistem informasi persediaan barang jadi yang tidak terorganisir dengan baik seperti data yang tersimpan ganda dan data yang tertulis pada form namun data sulit dibaca.	Masih menggunakan *Microsoft Excel* dan kertas.	Terjadinya redudansi data dan data dapat hilang.	Membuat pangkalan data dimana data tersebut bisa diinputkan ke dalam pangkalan data yang terkompute-risasi
Economics (Biaya)	Proses sistem informasi persediaan barang jadi masih menggunakan banyak kertas untuk mencetak laporan dan pencatatan barang masuk dan keluar	Proses sistem informasi persediaan barang jadi masih dilakukan secara manual dimana proses tersebut menggunakan kertas.	Bertambahnya biaya untuk tempat pengarsipan seperti membeli lemari, rak arsip dan ordner.	Membuat sistem yang berbasis komputer dimana proses tersebut meminimalisir pemakaian kertas

Tabel IV.1 Analisis Permasalahan (Lanjutan)

Control (Pengendalian)	Kurangnya monitor terhadap proses sistem informasi persediaan barang jadi sehingga stok mengalami kekurangan.	Belum adanya metode yang dapat memenuhi kebutuhan persediaan untuk tetap tersedia	Persediaan barang jadi yang sulit dimonitor mengakibatkan *stock out*	Menggunakan metode *buffer stock* dan ROP yang dapat mengamankan atau memenuhi ketersediaan barang jadi
Efficiency *(Efisiensi)*	Waktu yang digunakan staf antar divisi untuk mengantarkan dokumen ke divisi lain cukup memakan waktu produktif antara 5 – 10 menit.Ji	Belum adanya sistem terkomputeriasa-si yang dapat mengirimkan dokumen ke antar divisi.	Penggunaan waktu menjadi tidak optimal karena digunakan untuk mengirimkan dokumen ke divisi lain.	Membuat sistem yang terkomputerisasi untuk mempercepat proses pengiriman dokumen/arsip dari satu divisi ke divisi lain menjadi <= 1 menit

MILIK PERPUSTAKAAN STMI
Membaca : Ibadah, Mengambil : Dosa

BAB V
ANALISIS DAN PEMBAHASAN

5.1 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem untuk sistem informasi pengendalian persediaan dapat dilihat pada Tabel V.1 sebagai berikut:

Tabel V.1 Analisis Kebutuhan Sistem

No.	Permasalahan	Kebutuhan Pengguna	Solusi	Kebutuhan Sistem
				Functional Requirement
1.	Untuk pembuatan kartu stok dan rekap stok persediaan barang jadi masih dilakukan secara manual	Sistem terkomputerisasi yang dapat membantu proses pembuatan rekap stok dan kartu stok	Merancang dan membangun suatu sistem yang membantu pembuatan rekap stok dan kartu stok barang jadi	Sistem mampu mengelola data hasil produksi dan melakukan validasi barang jadi keluar untuk menghasilkan rekap stok dan kartu stok persediaan barang jadi.
2.	Data terkait persediaan barang jadi yang tidak terorganisir dengan baik seperti data yang tersimpan ganda dan data yang tertulis pada form namun data sulit dibaca.	Sistem dengan *database* seperti MariaDB untuk menghindari kemungkinan redudansi data, data tak terbaca dan data hilang.	Membuat proses penginputan data persediaan barang jadi menjadi terkomputerisasi.	Sistem dapat mengelola data persediaan barang jadi.

Tabel V.1 Analisis Kebutuhan Sistem (Lanjutan)

No.	Permasalahan	Kebutuhan *User*	Solusi	Kebutuhan Sistem
				Functional Requirement
3.		Sistem informasi persediaan yang terkomputerisasi sehingga mempermudah divisi Inventori dalam menjaga persediaan barang jadi.	Menerapkan metode dan *Buffer Stock* untuk membantu staff Inventori dalam menjaga persediaan barang jadi	Sistem dapat membantu menjaga persediaan barang jadi menggunakan metode *Buffer Stock.*
4.		Sistem yang aman dan nyaman digunakan	Membuat sistem dengan menu *login* 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 pengendalian persediaan yang diusulkan:

1. Staf bagian produksi mengelola data hasil produksi dan data barang jadi seperti *input*, ubah dan menghapus data,.
2. Inventori melakukan validasi barang jadi keluar dan mencetak rekap stok dan kartu stok.

3. Staf PPC dapat membuat dan mencetak surat jalan.
4. *Login* dengan *username* dan *password* untuk masuk kedalam sistem.

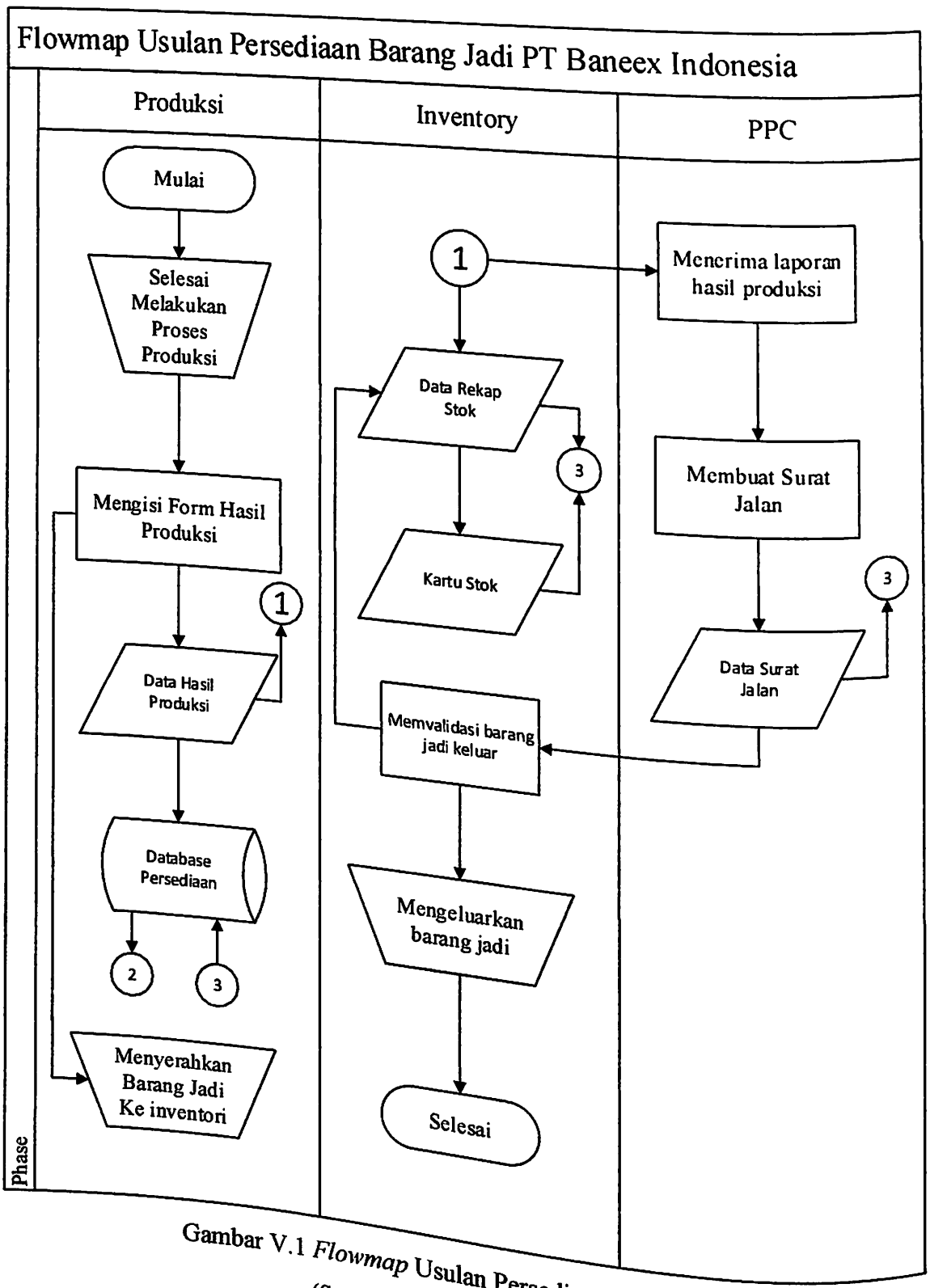
5.1.2 *Non Functional Requirement*

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

1. Operasional
   - Membutuhkan *framework* dan *database*. Seperti *Code Igniter* 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 kartu stok

5.2 *Flowmap* Sistem Informasi Persediaan Barang Jadi Usulan

Berdasarkan analisis sistem yang sedang berjalan maka diusulkan sistem informasi persediaan sebagai berikut:



Gambar V.1 *Flowmap* Usulan Persediaan Barang Jadi
(Sumber: Hasil Analisis, 2018)

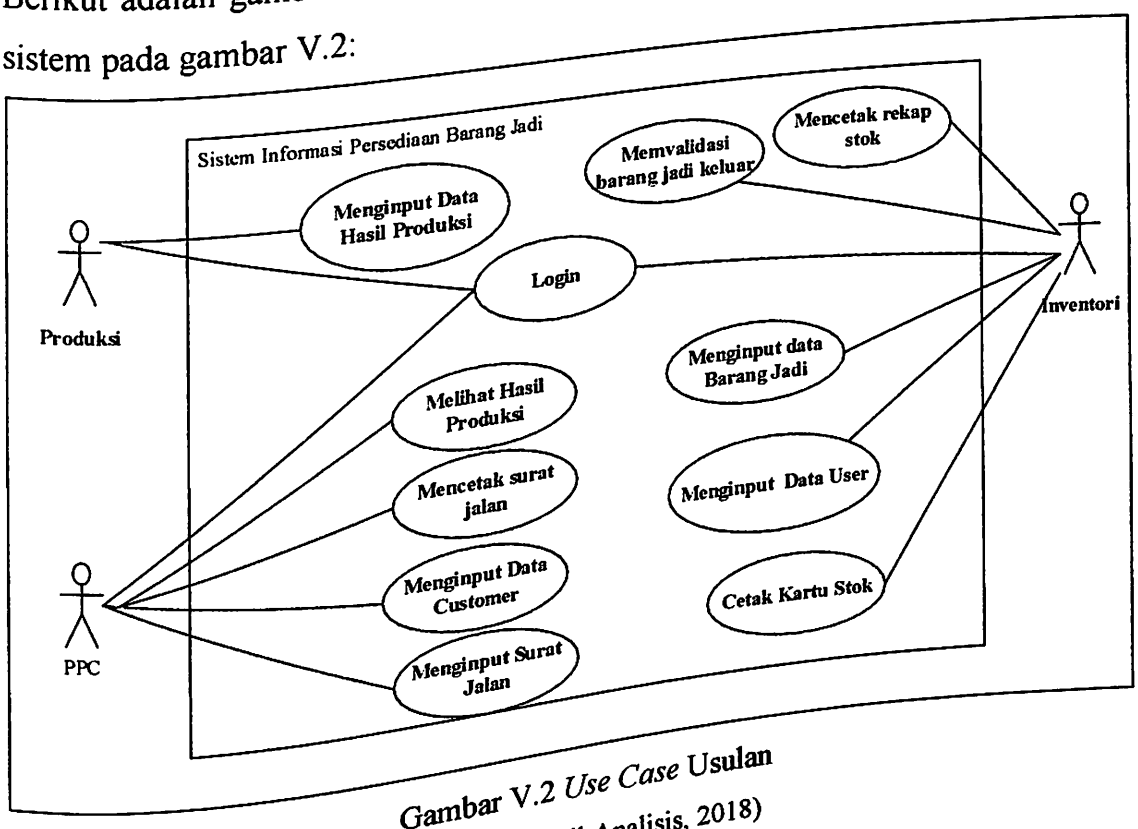
Keterangan Gambar V.1:

1. Setelah selesai proses produksi di bagian produksi akan menginput hasil produksi di form hasil produksi yang nanti otomatis masuk dalam data rekap stok.

2. Setelah menerima laporan hasil produksi, PPC membuat surat jalan berdasarkan rencana pengiriman yang telah dibuat lalu.
3. Inventori melakukan validasi pengeluaran barang sesuai surat jalan yang dibuat. Dan menyiapkan barang yang akan dikirim.
4. Divisi Inventori dapat mencetak rekap kartu stok berdasarkan tanggal dan kartu stok berdasarkan nama barang yang mana keduanya akan digunakan sebagai bukti laporan.

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 gambar V.2:



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*	*Login*
Aktor	PPC, Inventori dan Produksi
Deskripsi	*Use case* ini menggambarkan setiap aktor dapat masuk kedalam system
Normal Flow of Events	1. *User* membuka *form login* 2. *User* memasukkan *username* dan *password* pada *form login* 3. Sistem akan memvalidasi data, apakah data yang telah dimasukkan telah benar atau tidak 4. A. Jika data benar, maka *user* akan masuk kedalam halaman utama system
Alternate Flow	B. Jika data salah, maka *user* akan kembali kembali ke halaman *form login*

(Sumber: Hasil Analisis, 2018)

Tabel V.3 *Use Case Description* Mengelola Data Barang Jadi

Nama *Use Case*	Mengelola Data Barang Jadi
Aktor	Inventori
Deskripsi	*Use case* ini menggambarkan aktor yang mengelola data barang jadi.
Normal Flow of Events	1. Staf inventori memilih menu barang jadi. Lalu sistem akan mengambil semua informasi data barang jadi yang tersimpan pada *database* dan menampilkan informasi tersebut kedalam tabel data barang jadi 2. Staf inventori dapat melihat *detail database* yang tersimpan dan dapat juga menambahkan, mengedit, menghapus, dan mencari data barang jadi.
Alternate Flow	

(Sumber: Hasil Analisis, 2018)

Tabel V.4 *Use Case Description* Mengelola Data Hasil Produksi

Nama *Use Case*	Mengelola Data Hasil Produksi
Aktor	Produksi
Deskripsi	*Use case* ini menggambarkan aktor mengelola data hasil produksi.
Normal Flow of Events	1. *User* membuka *form* Hasil produksi lalu *user* menekan tombol tambah Hasil produksi. 2. *User* memasukkan nomor Hasil produksi dan tanggal 3. Sistem akan mevalidasi data yang akan dimasukkan. a. Jika data berhasil masuk maka sistem akan mengirim pesan data berhasil dimasukkan dan sistem akan membuka *form* Hasil produksi Detil 4. *User* memasukkan nama barang dan Qty/jumlah barang jadi 5. Jika data berhasil masuk maka sistem akan mengirim pesan data berhasil dimasukkan. 6. Staf produksi dapat juga mengedit, menghapus, dan mencari data hasil produksi.
Alternate Flow	1. Jika data sudah ada maka sistem akan mengirim pesan maaf data sudah ada. 2. Jika data sudah ada maka sistem akan mengirim pesan maaf data sudah ada

(Sumber: Hasil Analisis, 2018)

Tabel V.5 *Use Case Description* Mengelola Data Surat Jalan

Nama *Use Case*	Mengelola Data Surat Jalan
Aktor	PPC
Deskripsi	*Use case* ini menggambarkan aktor yang mengelola data surat jalan
Normal Flow of Events	1. *User* membuka *form* Surat Jalan lalu *user* menekan tombol tambah Surat Jalan 2. *User* memasukkan nomor surat jalan , nomor part, nama barang dan jumlah barang 3. Sistem akan mevalidasi data yang akan dimasukkan. a. Jika data berhasil masuk maka sistem akan mengirim pesan data berhasil dimasukkan dan sistem akan

Tabel V.5 *Use Case Description* Mengelola Data Surat Jalan(Lanjutan)

	membuka *form* Bukti Detil 4. a .Jika data berhasil masuk maka sistem akan mengirim pesan data berhasil dimasukkan. 5. Staf *PPC* dapat juga mengedit, menghapus, dan mencari data Surat Jalan
Alternate Flow	1. Jika data sudah ada maka sistem akan mengirim pesan maaf data sudah ada. 2. Jika data sudah ada maka sistem akan mengirim pesan maaf data sudah ada

(Sumber: Hasil Analisis, 2018)

Tabel V.6 *Use Case Description* Cetak Surat Jalan

Nama *Use Case*	Cetak Surat Jalan
Aktor	PPC
Deskripsi	*Use case* ini menggambarkan aktor yang mencetak surat jalan
Normal Flow of Events	1. *User* memilih menu surat jalan. 2. *User* memilih data yang akan dibuat 3. Sistem akan mengambil semua informasi data transaksi barang jadi yang tersimpan pada *database* dan menampilkan informasi tersebut kedalam tabel data surat jalan.
Alternate Flow	

(Sumber: Hasil Analisis, 2018)

Tabel V.7 *Use Case Description* Validasi Barang Jadi Keluar

Nama *Use Case*	Mengelola Data Kartu Stok Barang Jadi Keluar
Aktor	Inventori
Deskripsi	*Use case* ini menggambarkan aktor yang mengelola kartu stok data barang jadi keluar.
Normal Flow of Events	1. Staf inventori memilih menu kartu stok barang jadi masuk. Lalu sistem akan mengambil semua informasi daftar stok yang tersimpan pada *database* dan menampilkan informasi tersebut kedalam tabel data kartu stok barang jadi masuk

Tabel V.7 *Use Case Description* Validasi Barang Jadi Keluar(Lanjutan)

	2. Staf inventori dapat juga mengedit, menghapus, dan mencari data barang jadi keluar.
Alternate Flow	

Sumber: Hasil Analisis, 2018

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

Nama *Use Case*	Mengelola Data *User*
Aktor	Inventori
Deskripsi	*Use case* ini menggambarkan aktor yang mengelola data *user.*
Normal Flow of Events	1. Staf inventori memilih menu *user.* 2. Sistem akan mengambil semua informasi data *user* yang tersimpan pada *database* dan menampilkan informasi tersebut kedalam tabel data produk 3. Staf inventori dapat melihat *detail database* yang tersimpan dan dapat juga menambahkan, mengedit, menghapus, dan mencari data *User.*
Alternate Flow	

(Sumber: Hasil Analisis, 2018)

Tabel V.9 *Use Case Description* Mengelola Data *Customer*

Nama *Use Case*	Mengelola Data *User*
Aktor	PPC
Deskripsi	*Use case* ini menggambarkan aktor yang mengelola data *user.*
Normal Flow of Events	1. Staf PPC memilih menu *customer.* 2. Sistem akan mengambil semua informasi data *user* yang tersimpan pada *database* dan menampilkan informasi tersebut kedalam tabel data produk 3. Staf PPC dapat melihat *detail database* yang tersimpan dan dapat juga menambahkan, mengedit, menghapus, dan mencari data *Customer.*
Alternate Flow	

(Sumber: Hasil Analisis, 2018)

Tabel V.10 *Use Case Description* Pembuatan Rekap Stok

Nama *Use Case*	Pembuatan Rekap Stok
Aktor	Inventori
Deskripsi	*Use case* ini menggambarkan aktor membuat rekap kartu stok
Normal Flow of Events	1. *User* memilih menu rekap stok. 2. *User* memilih data yang akan dibuat 3. Sistem akan mengambil semua informasi data transaksi barang jadi yang tersimpan pada *database* dan menampilkan informasi tersebut kedalam tabel data rekap stok.
Alternate Flow	

(Sumber: Hasil Analisis, 2018)

Tabel V.11 *Use Case Description* Pembuatan Kartu Stok

Nama *Use Case*	Pembuatan Kartu Stok
Aktor	Inventori
Deskripsi	*Use case* ini menggambarkan aktor yang membuat kartu stok
Normal Flow of Events	4. *User* memilih menu kartu stok. 5. *User* memilih data yang akan dibuat 6. Sistem akan mengambil semua informasi data transaksi barang jadi yang tersimpan pada *database* dan menampilkan informasi tersebut kedalam tabel data kartu stok.
Alternate Flow	

(Sumber: Hasil Analisis, 2018)

Tabel V.11 *Use Case Description* Melihat Laporan Hasil Produksi

Nama *Use Case*	Melihat Laporan Hasil Produksi
Aktor	PPC
Deskripsi	*Use case* ini menggambarkan aktor dapat melihat hasil produksi
Normal Flow of Events	1. *User* memilih laporan hasil produksi. 2. Sistem akan mengambil semua informasi data transaksi barang jadi yang tersimpan pada *database* dan menampilkan informasi tersebut kedalam tabel hasil produksi.
Alternate Flow	

(Sumber: Hasil Analisis, 2018)

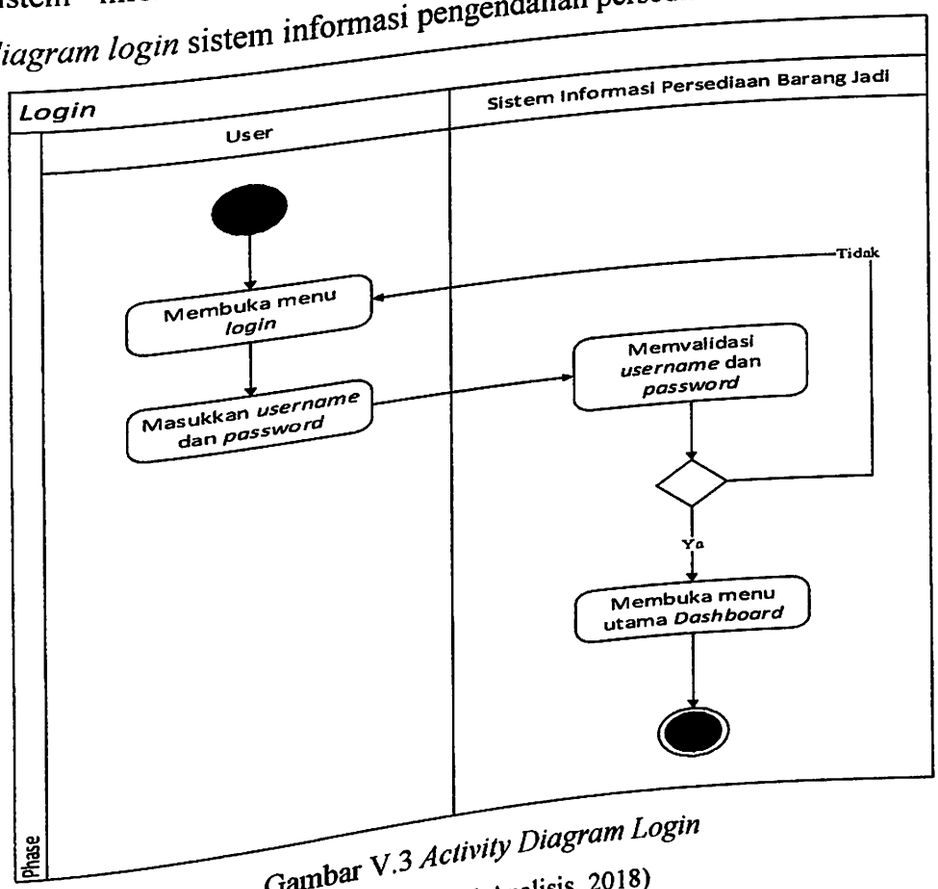
5.4 *Activity Diagram*

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 hubunganya 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 pengendalian persediaan 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 inventori untuk mendapatkan akses kedalam sistem informasi pengendalian persediaan. Berikut merupakan *activity diagram login* sistem informasi pengendalian persediaan

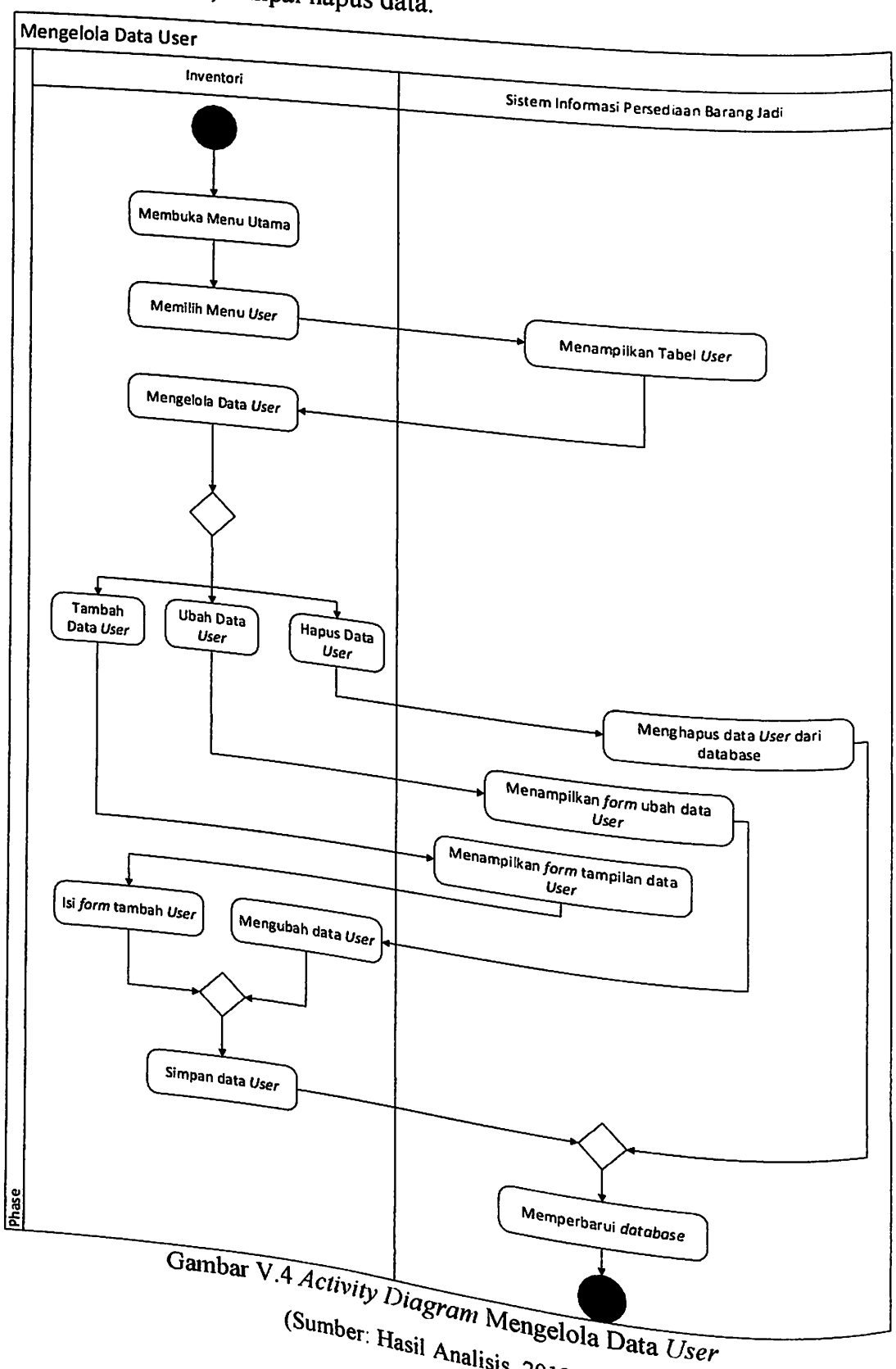


Gambar V.3 *Activity Diagram Login*

(Sumber: Hasil Analisis, 2018)

2. *Activity Diagram* Mengelola Data *User*

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

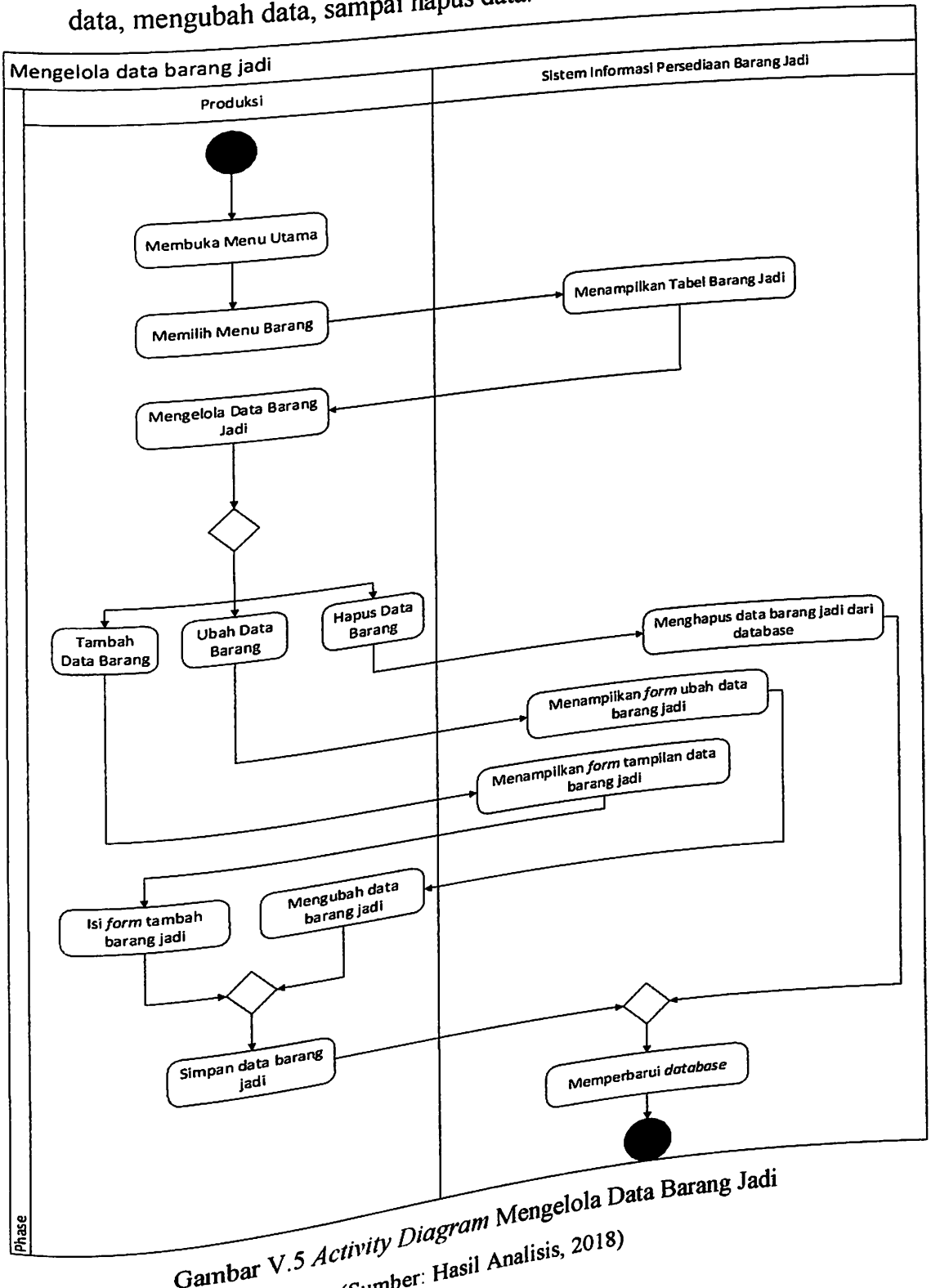


Gambar V.4 *Activity Diagram* Mengelola Data *User*

(Sumber: Hasil Analisis, 2018)

3. *Activity Diagram* Mengelola Data Barang Jadi

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

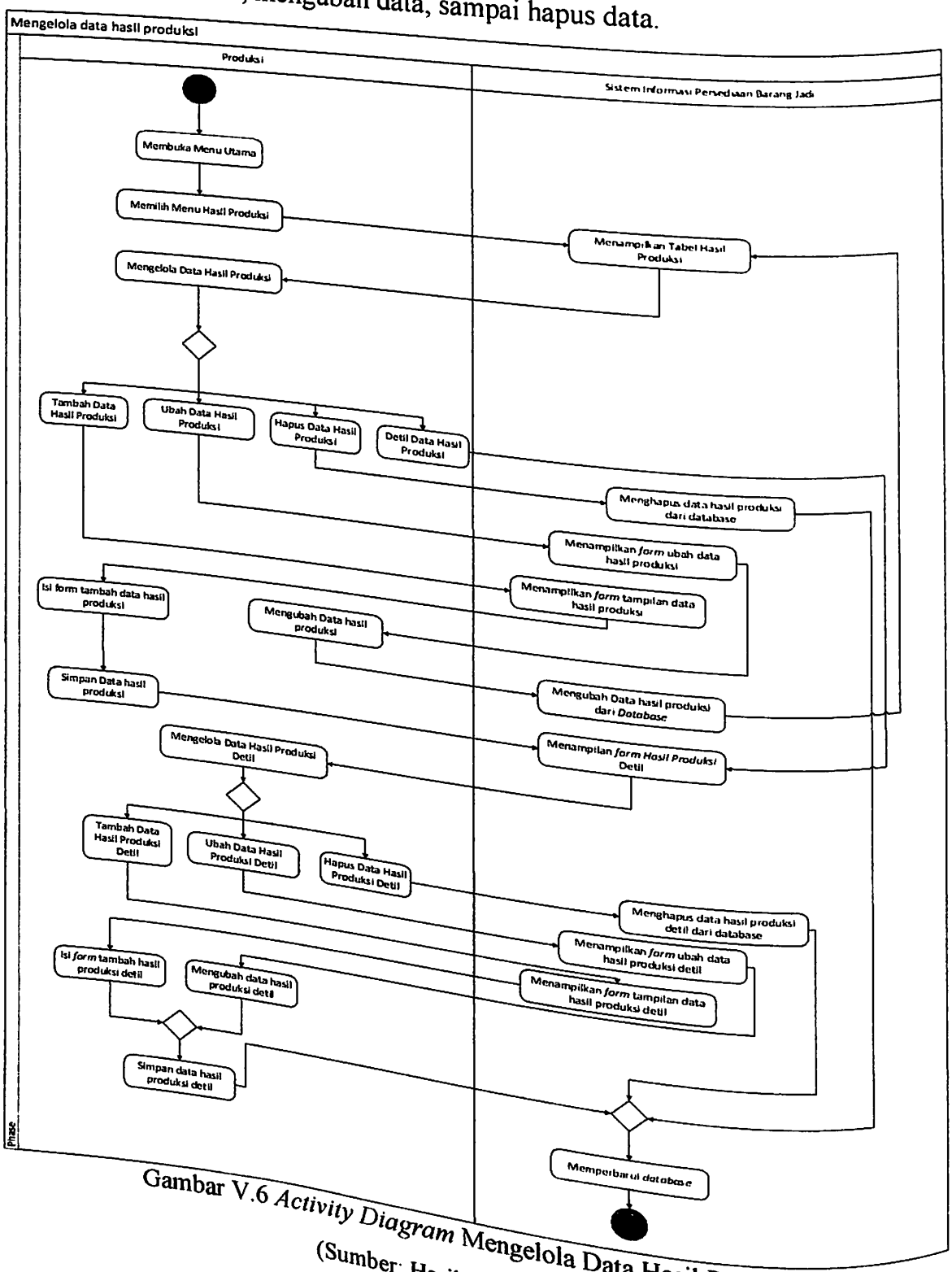


Gambar V.5 *Activity Diagram* Mengelola Data Barang Jadi

(Sumber: Hasil Analisis, 2018)

4. *Activity Diagram* Mengelola Data Hasil Produksi

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

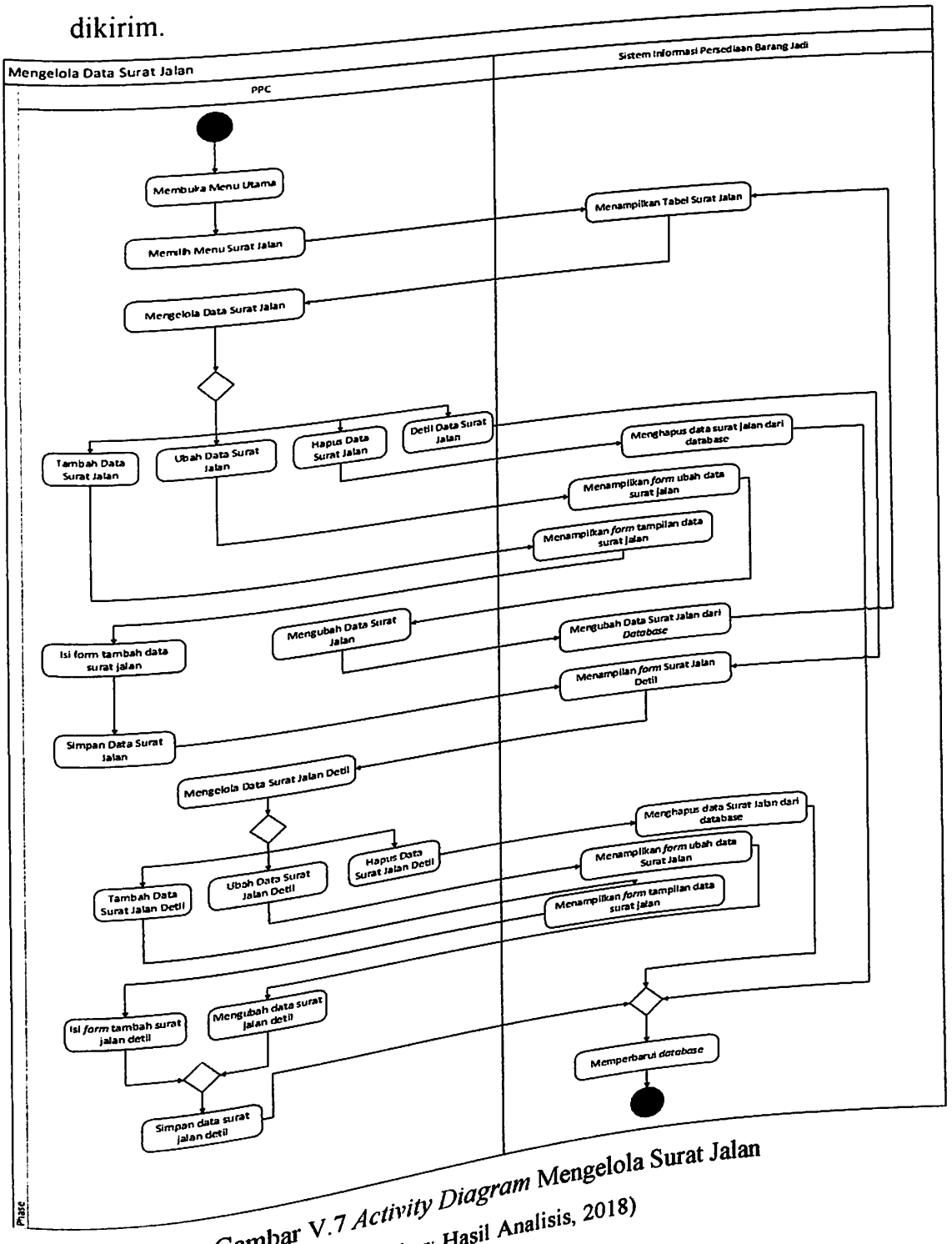


Gambar V.6 *Activity Diagram* Mengelola Data Hasil Produksi

(Sumber: Hasil Analisis, 2018)

5. *Activity Diagram* Mengelola Surat Jalan

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu staf PPC untuk meminta Inventori melakukan surat jalan untuk dikirim.

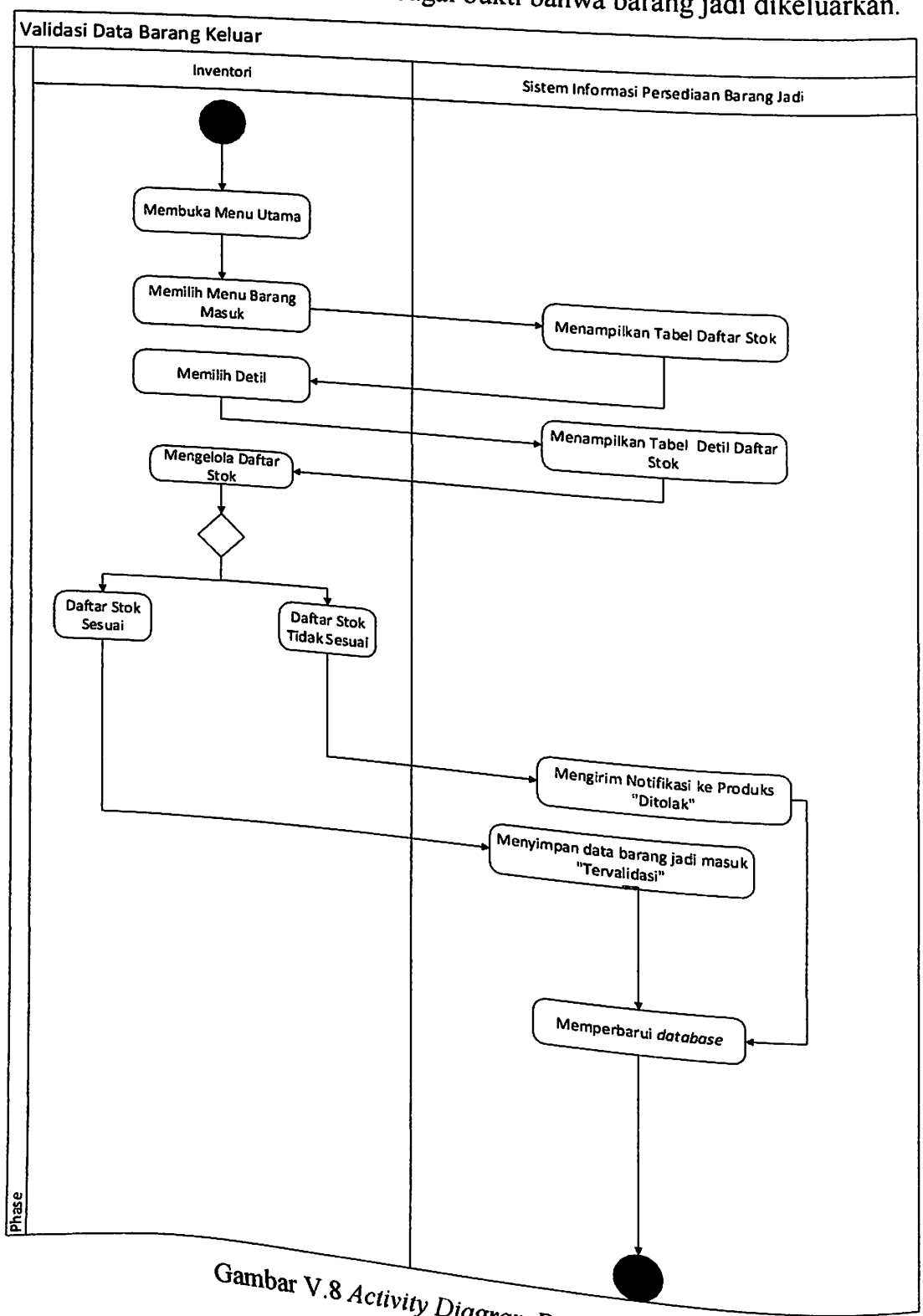


Gambar V.7 *Activity Diagram* Mengelola Surat Jalan

(Sumber: Hasil Analisis, 2018)

6. *Activity Diagram* Data Barang Keluar

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu staf Inventori untuk sebagai bukti bahwa barang jadi dikeluarkan.

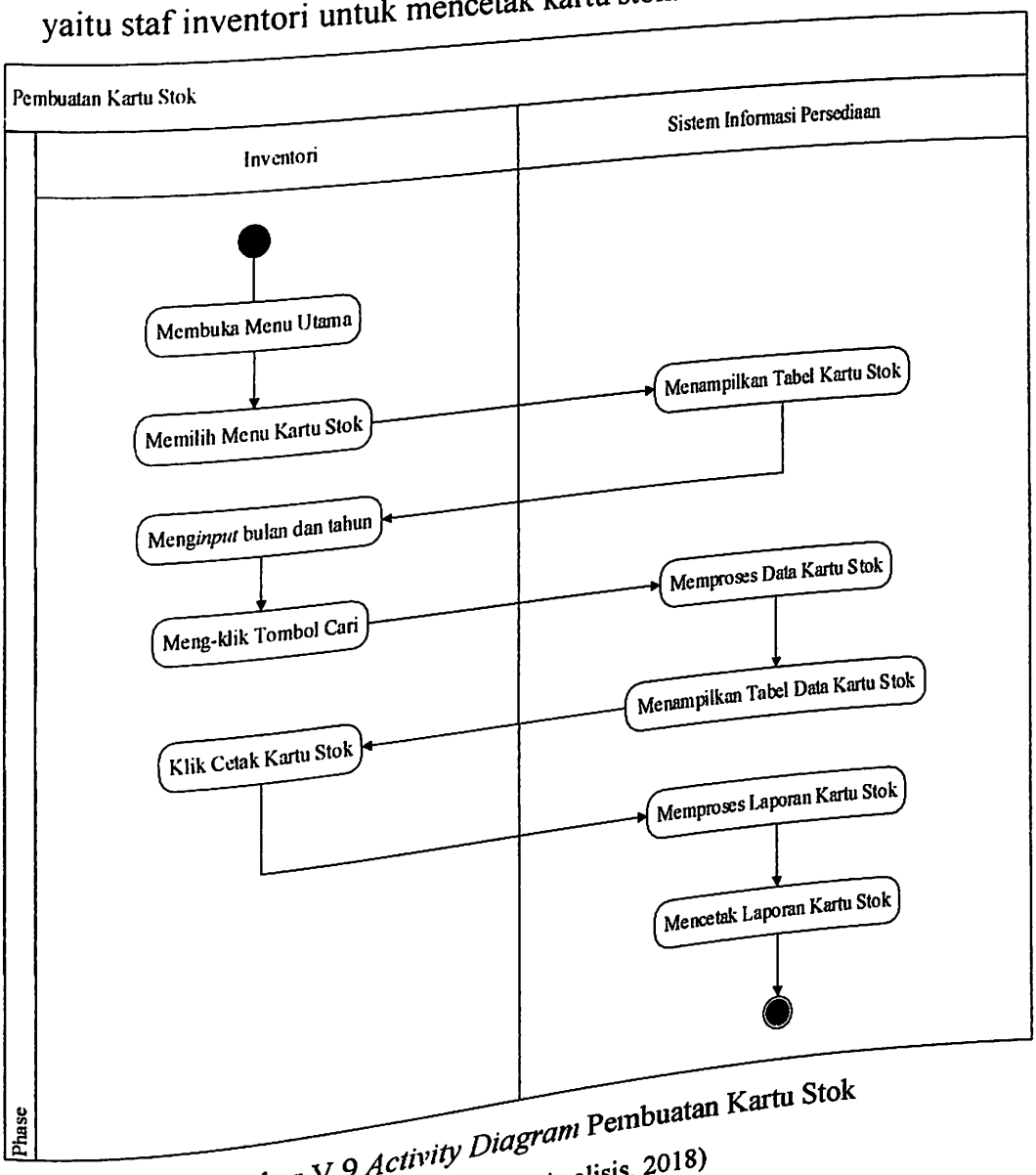


Gambar V.8 *Activity Diagram* Barang Jadi Keluar

(Sumber: Hasil Analisis, 2018)

7. *Activity Diagram* Pembuatan Kartu Stok

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu staf inventori untuk mencetak kartu stok.

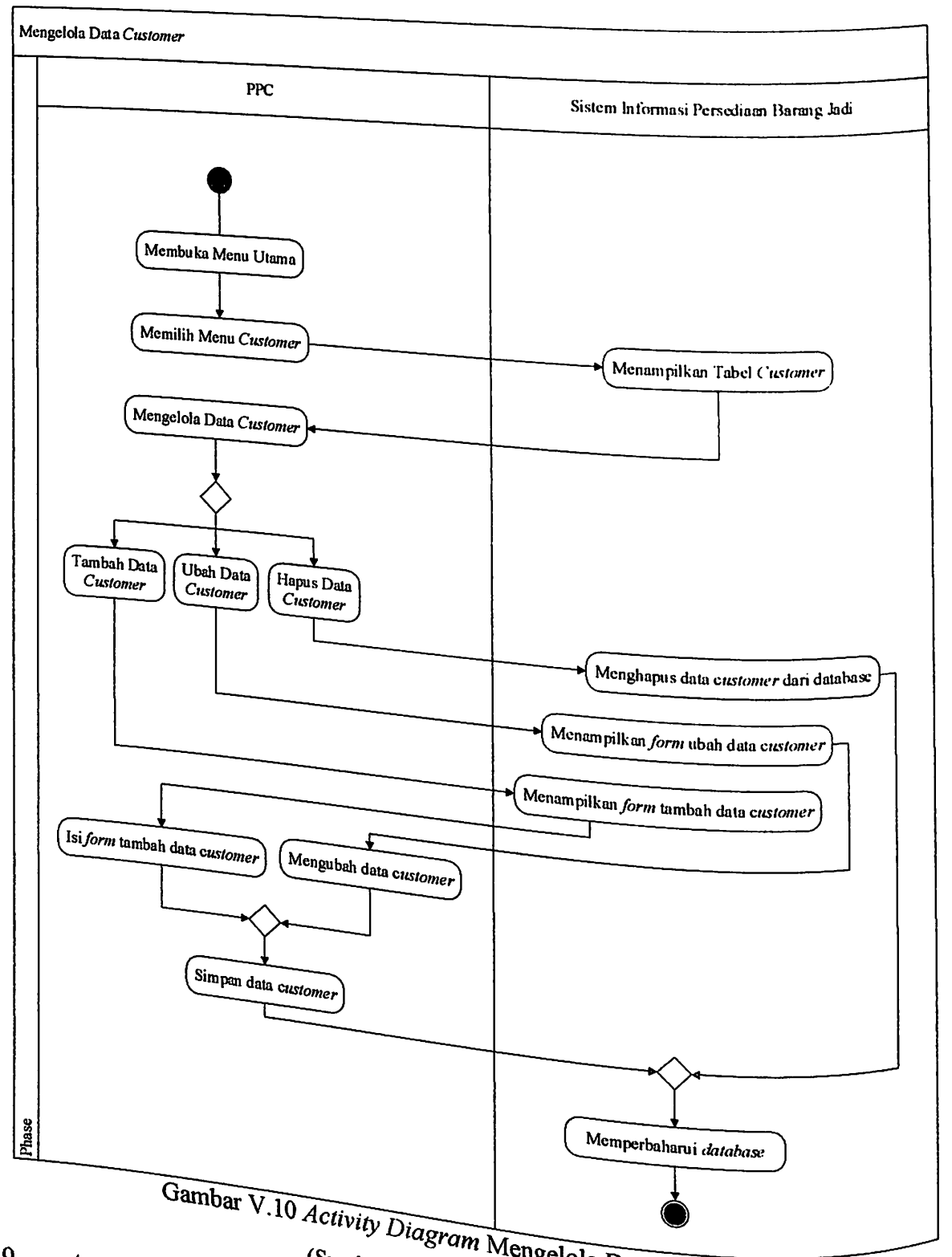


Gambar V.9 *Activity Diagram* Pembuatan Kartu Stok

(Sumber: Hasil Analisis, 2018)

8. *Activity Diagram* Mengelola Data *Customer*

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

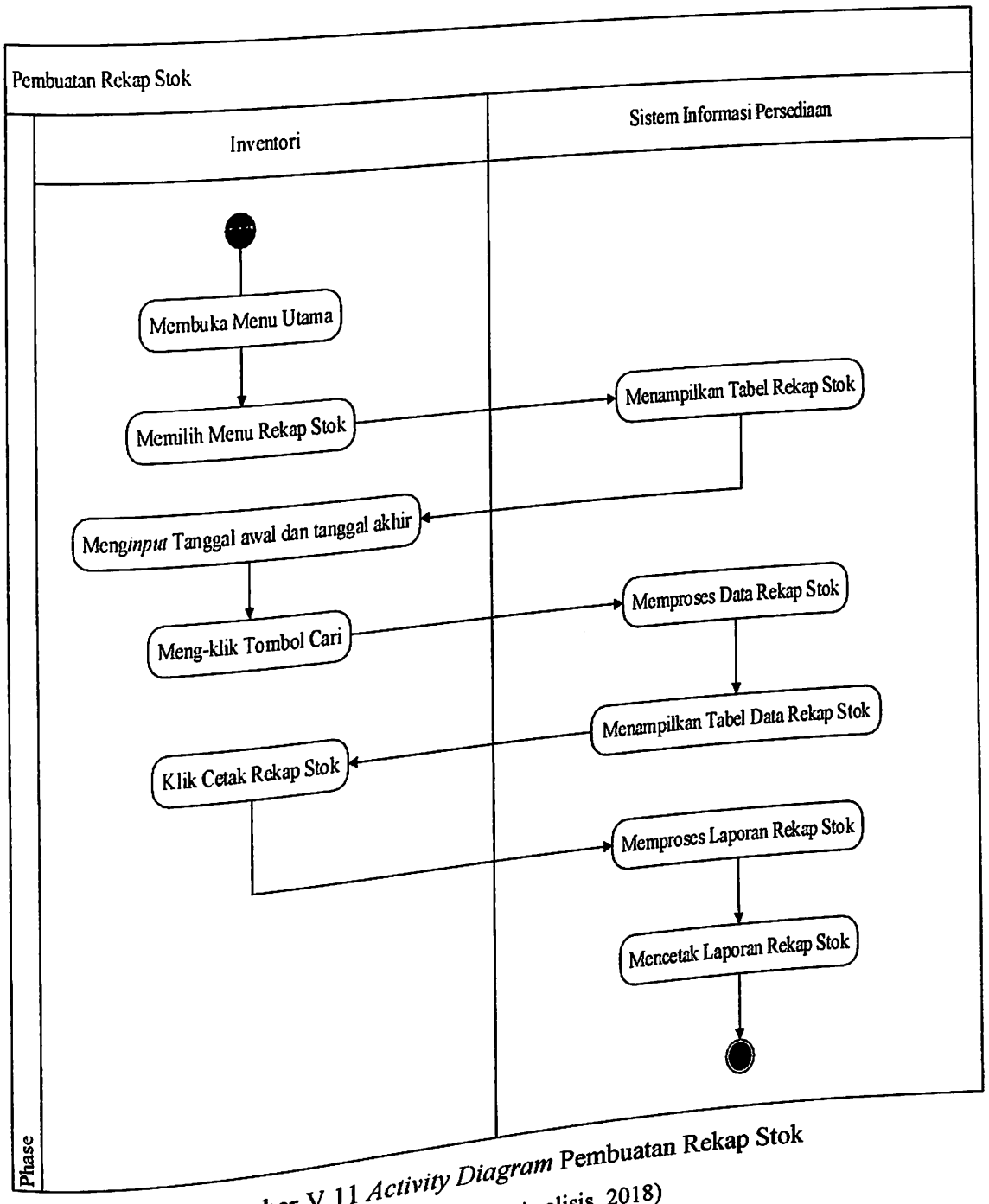


Gambar V.10 *Activity Diagram* Mengelola Data *Customer*

(Sumber: Hasil Analisis, 2018)

9. *Activity Diagram* Pembuatan Rekap Stok

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu staf inventori untuk mencetak Rekap stok.

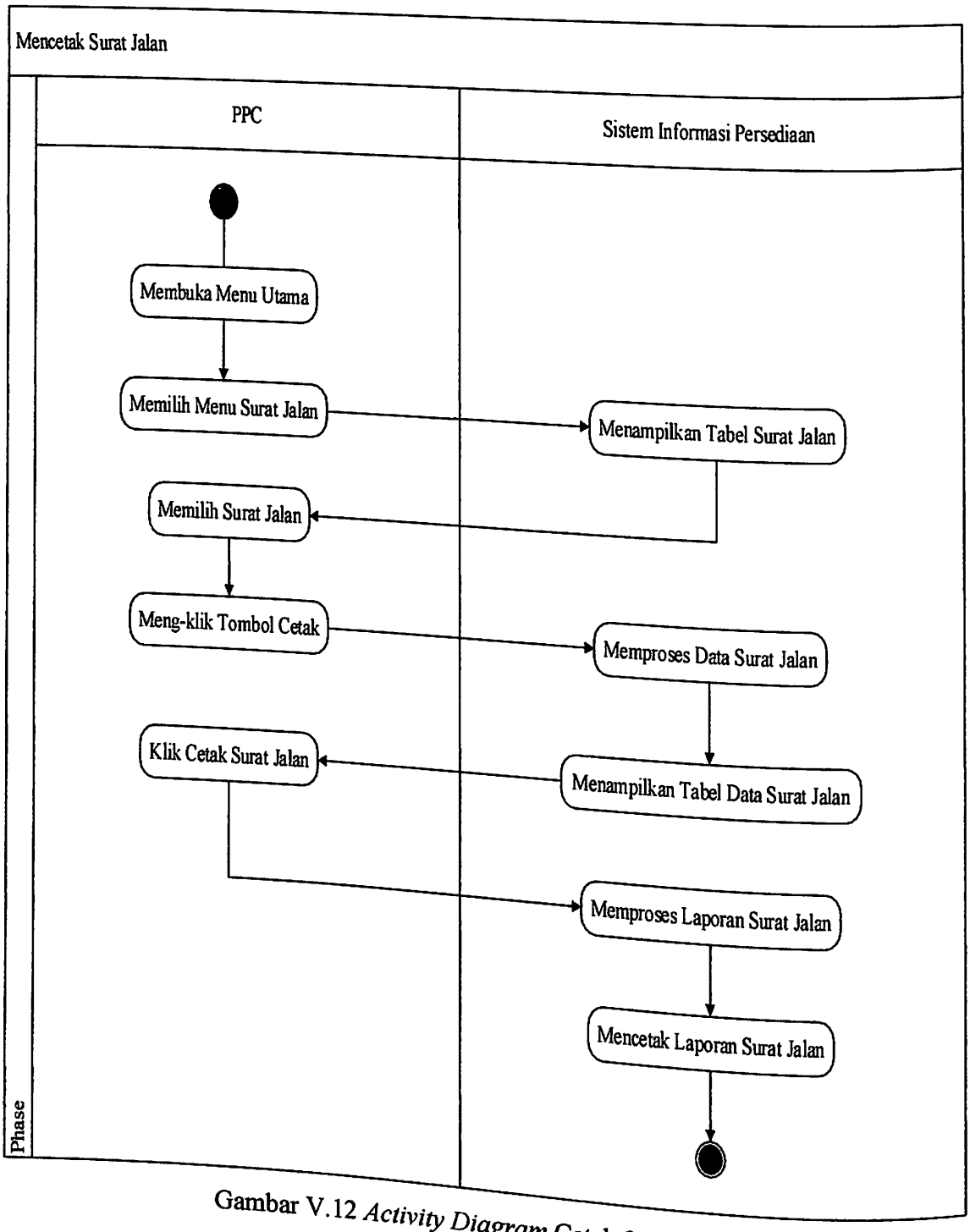


Gambar V.11 *Activity Diagram* Pembuatan Rekap Stok

(Sumber: Hasil Analisis, 2018)

10. *Activity Diagram* Cetak Surat Jalan

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu staf inventori untuk mencetak surat jalan.

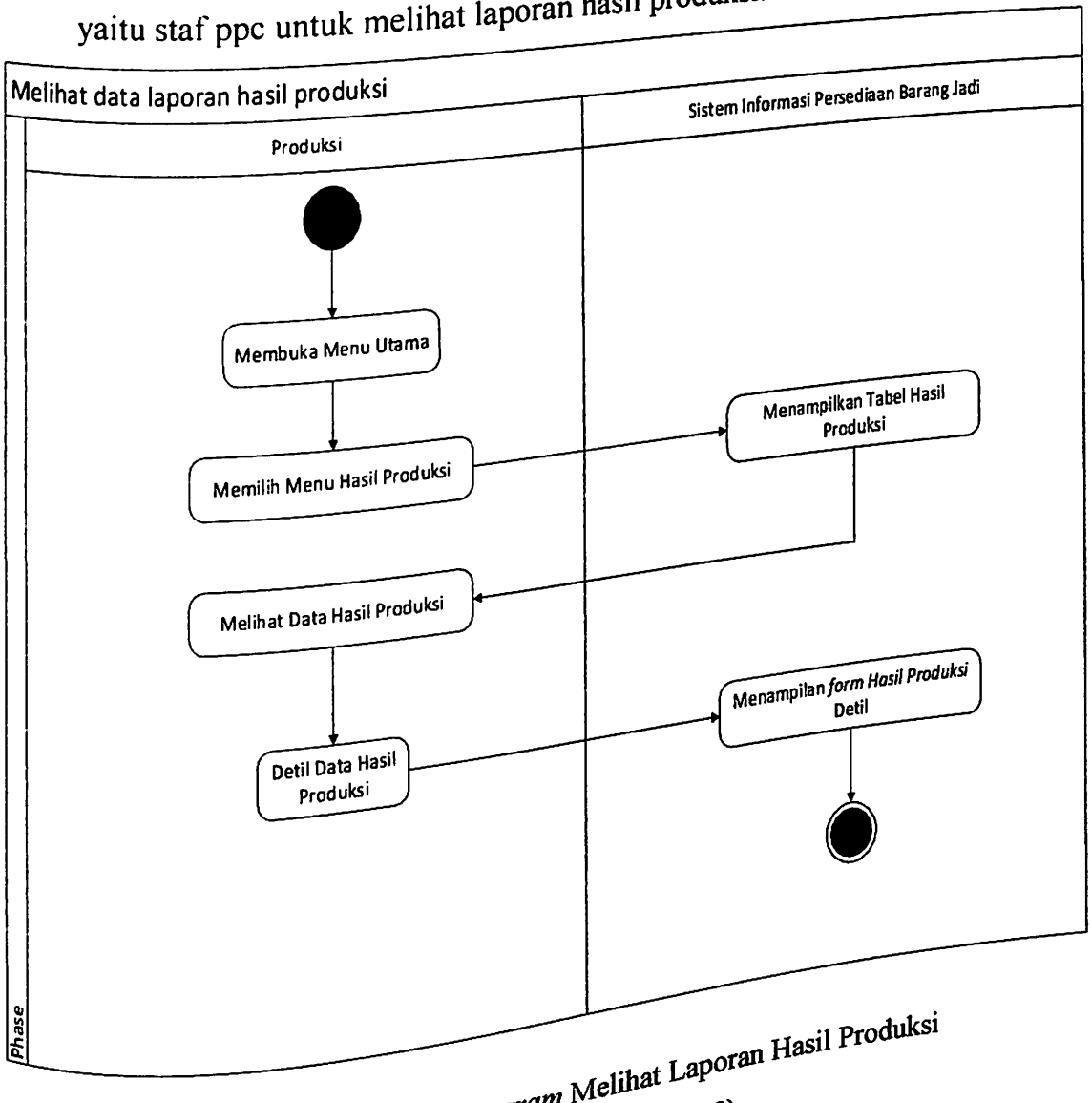


Gambar V.12 *Activity Diagram* Cetak Surat Jalan

(Sumber: Hasil Analisis, 2018)

11. *Activity Diagram* Melihat Laporan Hasil Produksi

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu staf ppc untuk melihat laporan hasil produksi.



Gambar V.13 *Activity Diagram* Melihat Laporan Hasil Produksi

(Sumber: Hasil Analisis, 2018)

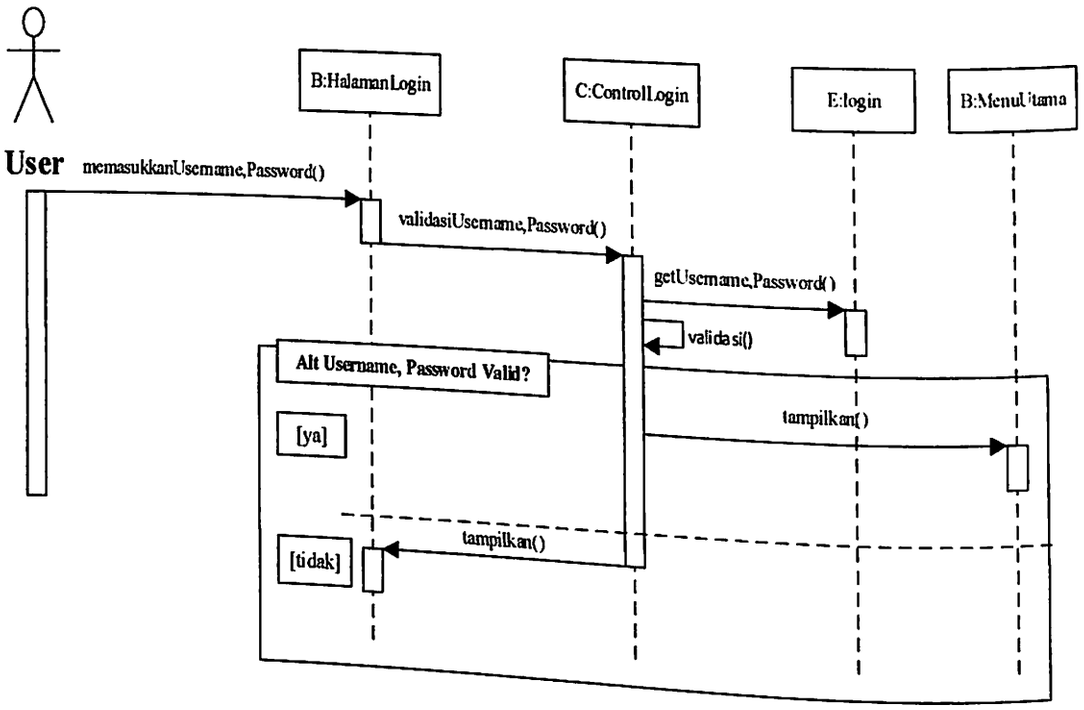
5.5 *Sequence Diagram*

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 pengendalian persediaan:

1. *Sequence Diagram Login*

Sequence diagram login merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses *login*. Proses *login*

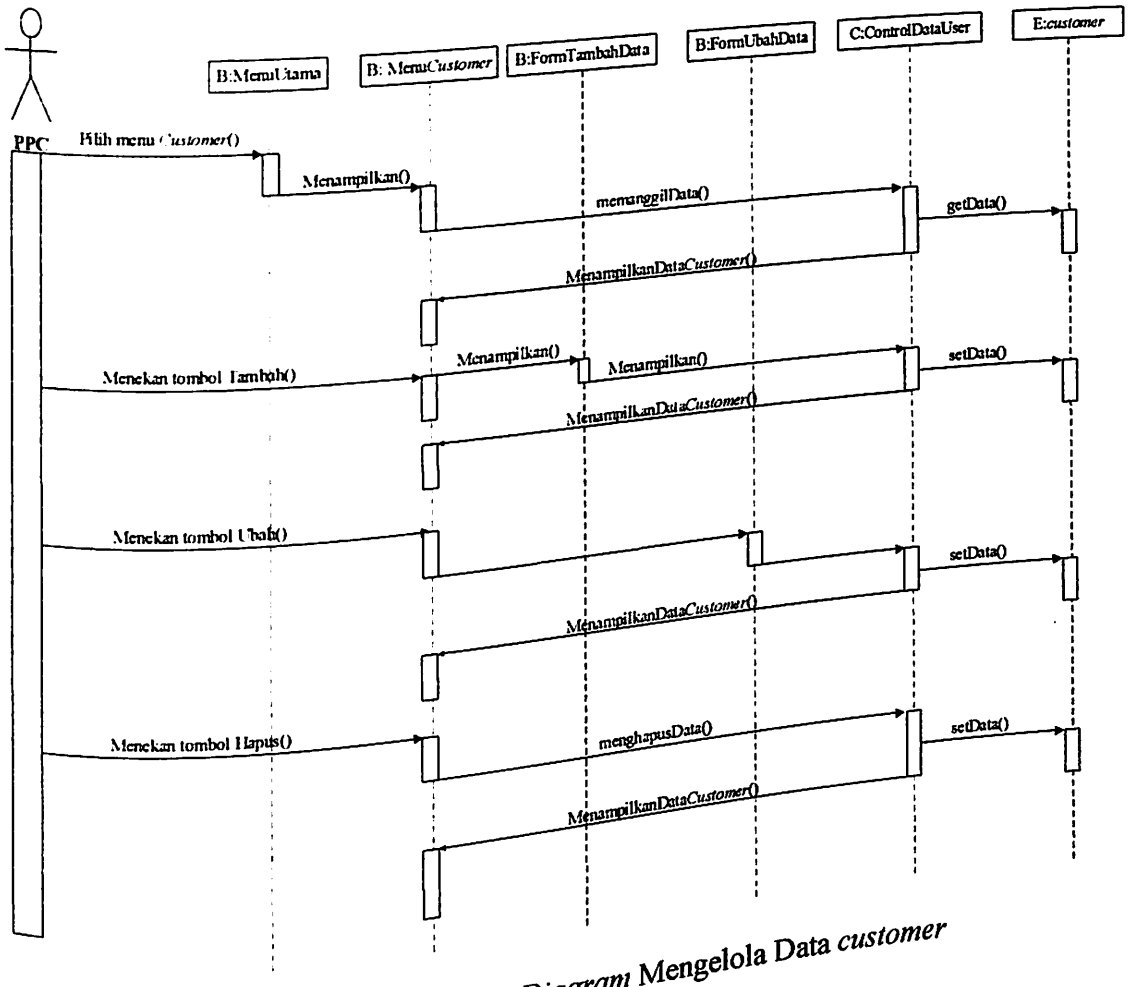
dalam sistem informasi pengendalian persediaan ini dilakukan oleh *user* agar dapat mengakses sistem. Berikut merupakan *sequence diagram* dari *use case login*:



Gambar V.14 *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 persediaan ini dilakukan oleh user yaitu staff PPC agar dapat menyimpan, menambah, dan menghapus data. Berikut merupakan *sequence diagram* dari *use case* mengelola data *customer*:

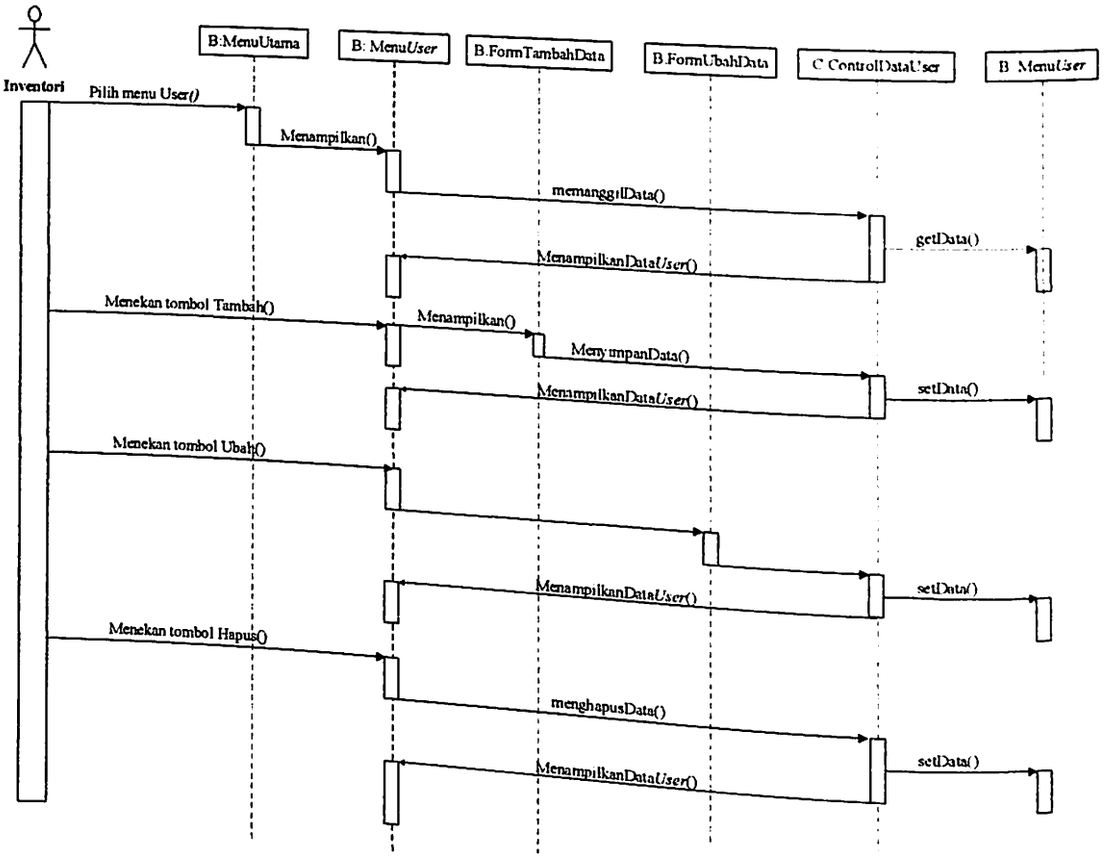


Gambar V.15*Sequence Diagram* Mengelola Data *customer*

(Sumber: Hasil Analisis, 2018)

3. *Sequence Diagram* Mengelola Data *User*

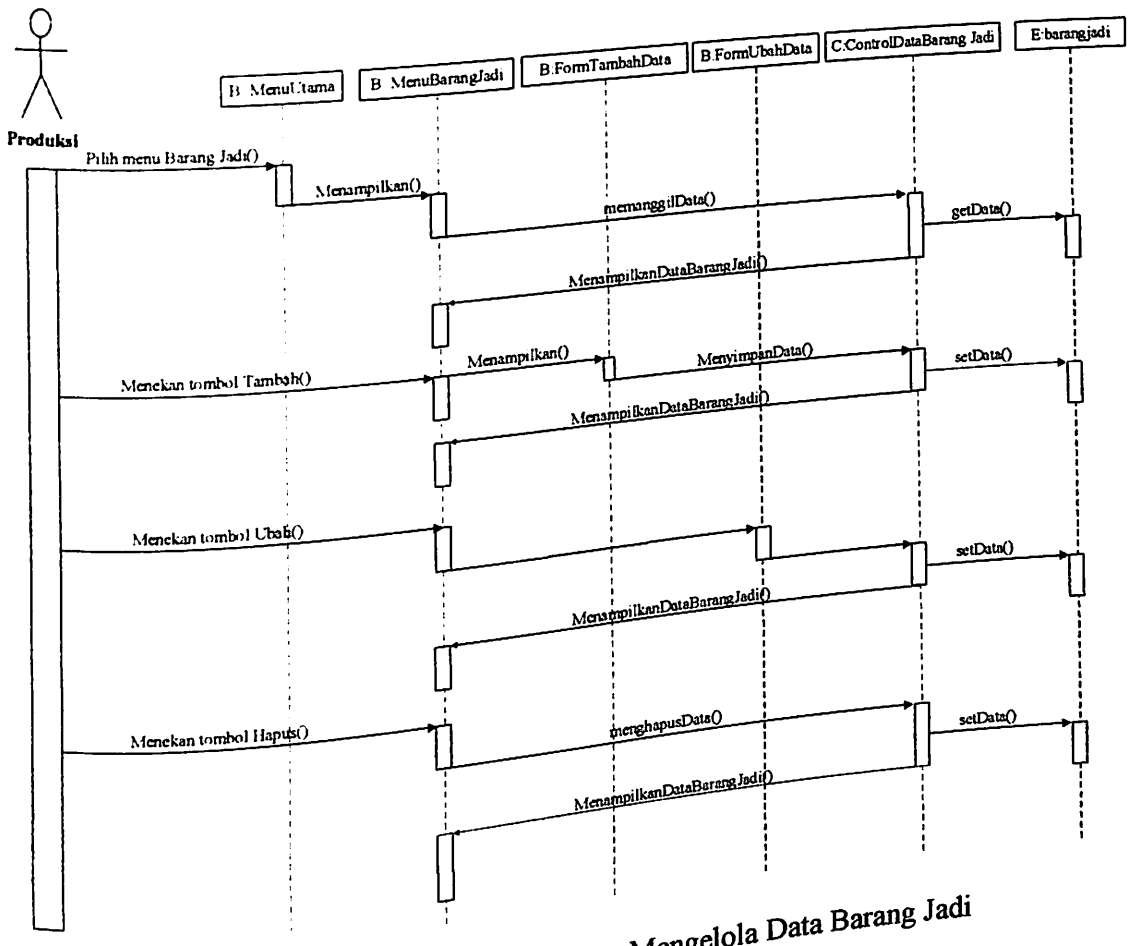
Sequence diagram mengelola data *user* merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses mengelola data *user*. Proses mengelola data produk dalam sistem informasi pengendalian persediaan ini dilakukan oleh user yaitu kepala produksi agar dapat menyimpan,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



Gambar V.16 *Sequence Diagram* Mengelola Data *User*
(Sumber: Hasil Analisis, 2018)

4. *Sequence Diagram* Mengelola Data Barang Jadi

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

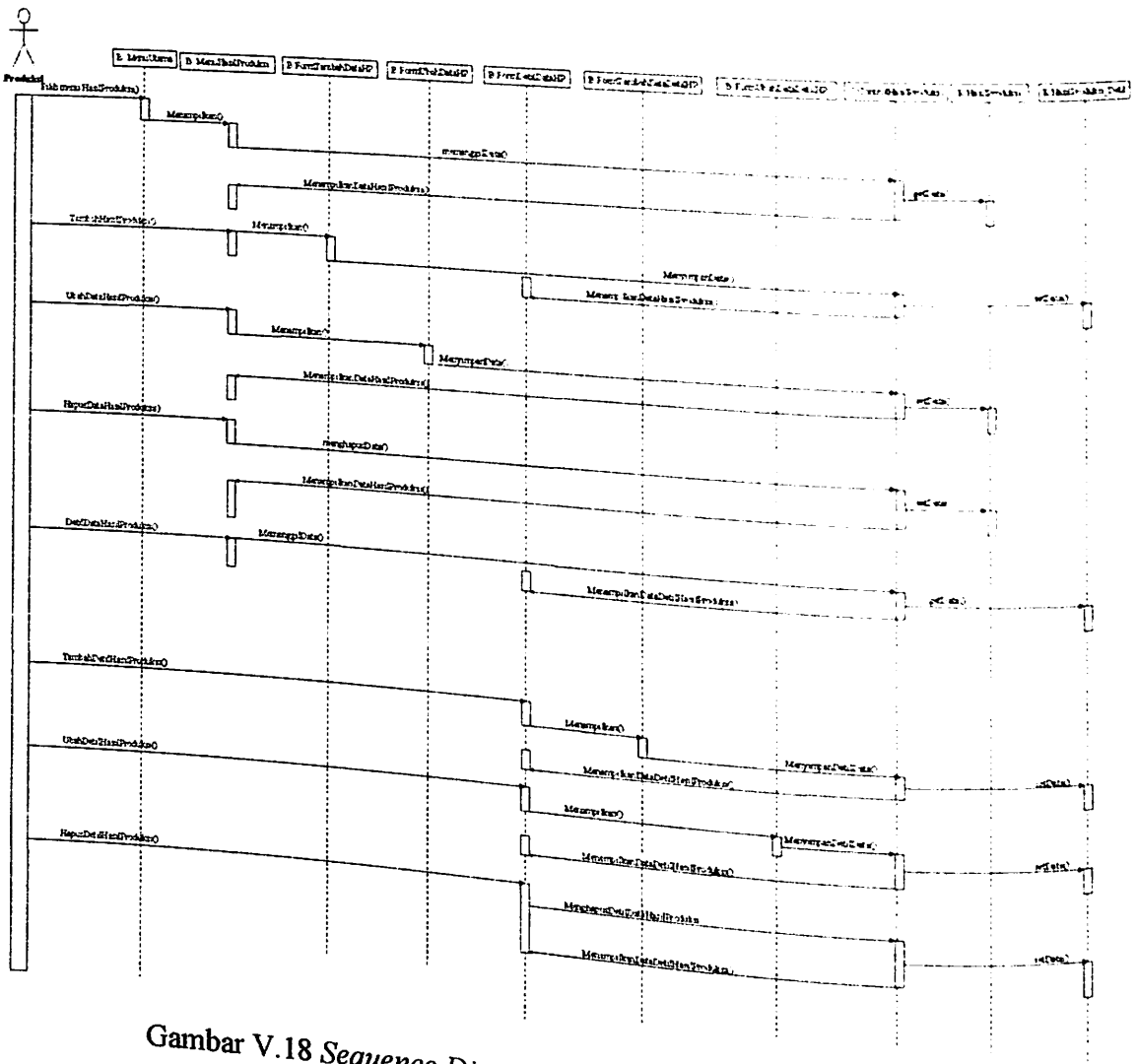


Gambar V.17 *Sequence Diagram* Mengelola Data Barang Jadi

(Sumber: Hasil Analisis, 2018)

5. *Sequence Diagram* Mengelola Data Hasil Produksi

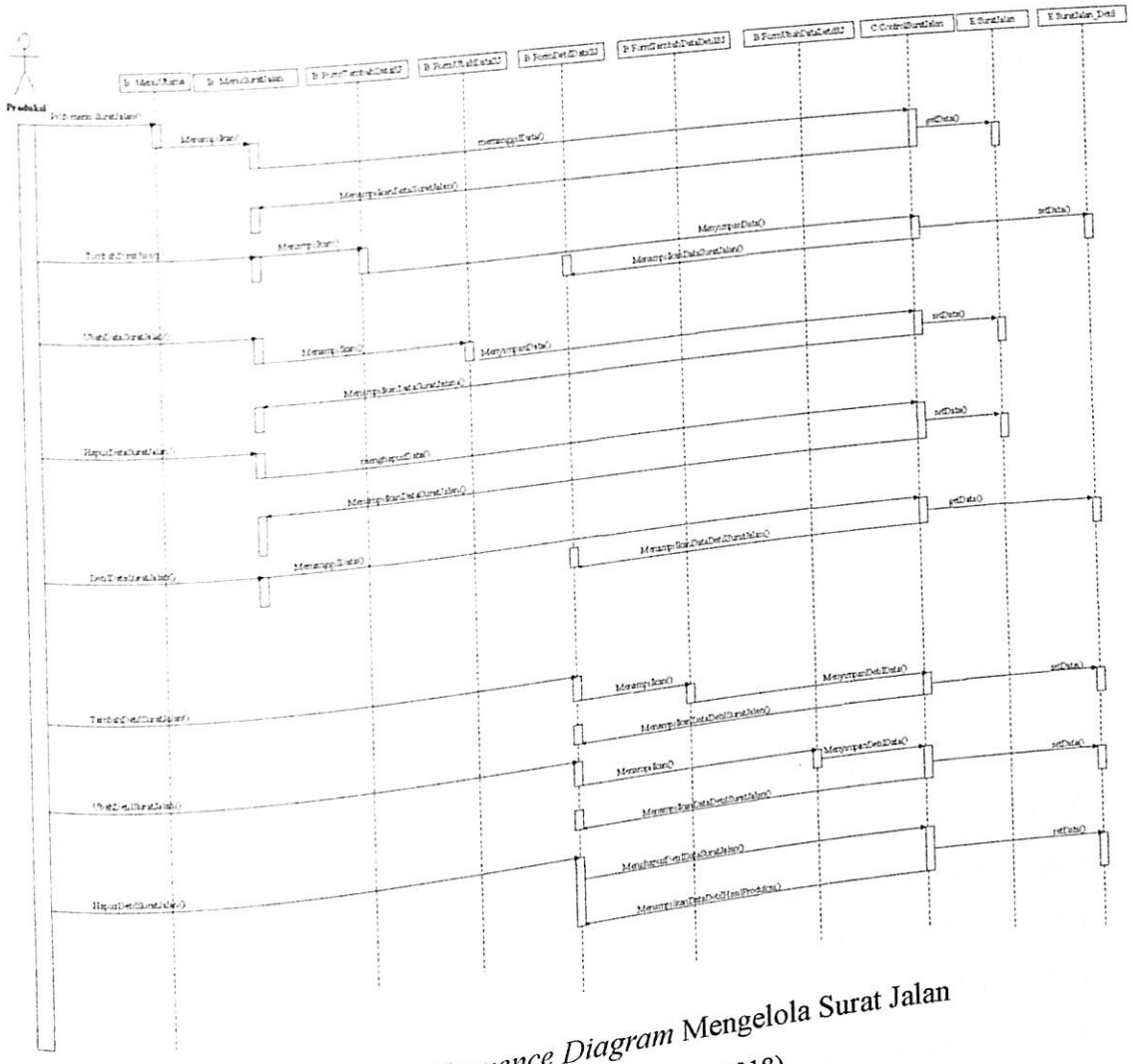
Sequence diagram mengelola data hasil produksi merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses mengelola data hasil produksi. Proses mengelola hasil produksi dalam sistem informasi pengendalian persediaan ini dilakukan oleh *user* agar dapat menyimpan, menambah,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



Gambar V.18 *Sequence Diagram* Mengelola Data Hasil Produksi
(Sumber: Hasil Analisis, 2018)

6. *Sequence Diagram* Mengelola Data Surat Jalan

Sequence diagram mengelola data surat jalan merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses pembuatan surat jalan. Proses mengelola data surat jalan barang dalam sistem informasi persediaan ini dilakukan oleh user yaitu staf PPC untuk membuat surat jalan. Berikut merupakan *sequence diagram* dari *use case* pembuatan surat jalan:

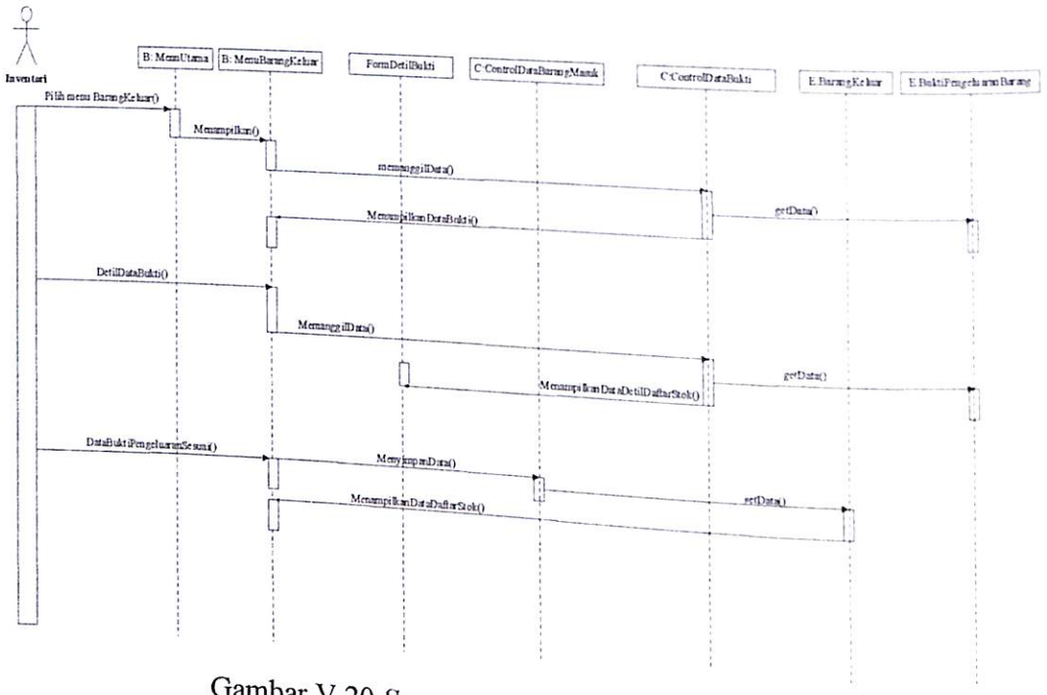


Gambar V.19 *Sequence Diagram* Mengelola Surat Jalan

(Sumber: Hasil Analisis, 2018)

7. *Sequence Diagram* Data Barang Keluar

Sequence diagram data barang keluar merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses pengeluaran barang dari Inventori. Proses pengeluaran barang dalam sistem informasi pengendalian persediaan ini dilakukan oleh user yaitu staf Inventori untuk memproses pengendalian persediaan. Berikut merupakan *sequence diagram* dari *use case* pembuatan bukti pengeluaran barang:

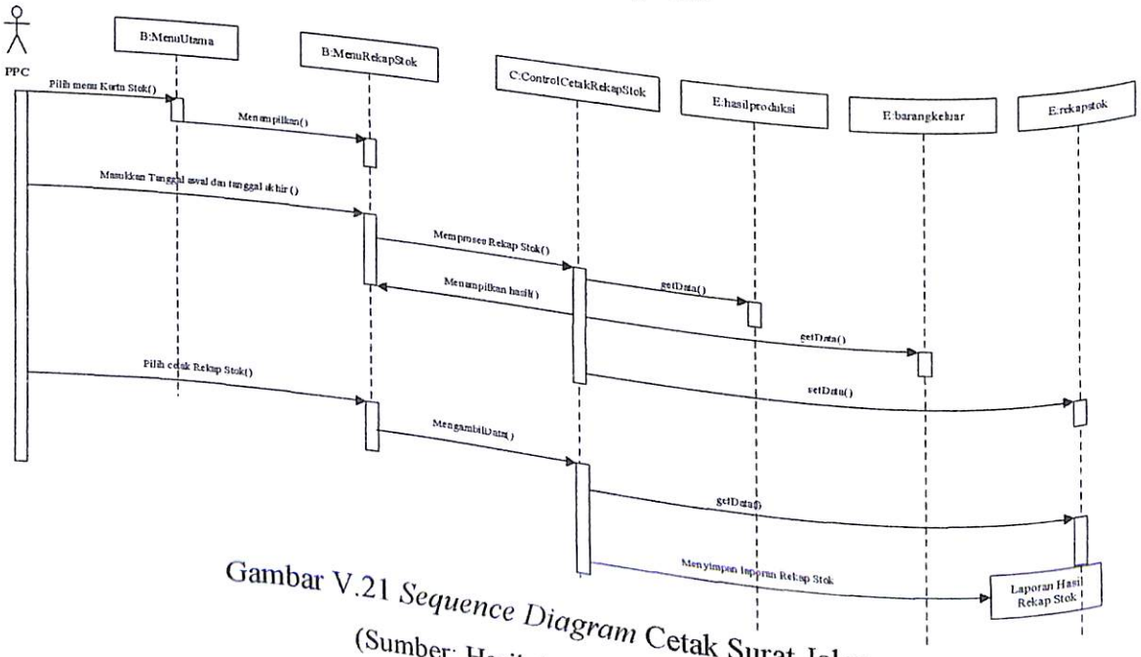


Gambar V.20 *Sequence Diagram* Data Barang Keluar

(Sumber: Hasil Analisis, 2018)

8. *Sequence Diagram* Mencetak Surat Jalan

Sequence diagram cetak surat jalan merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses cetak surat jalan di PPC. Proses pembuatan surat jalan dalam sistem informasi persediaan ini dilakukan oleh user yaitu staf PPC. Berikut merupakan *sequence diagram* dari *use case* cetak surat jalan:

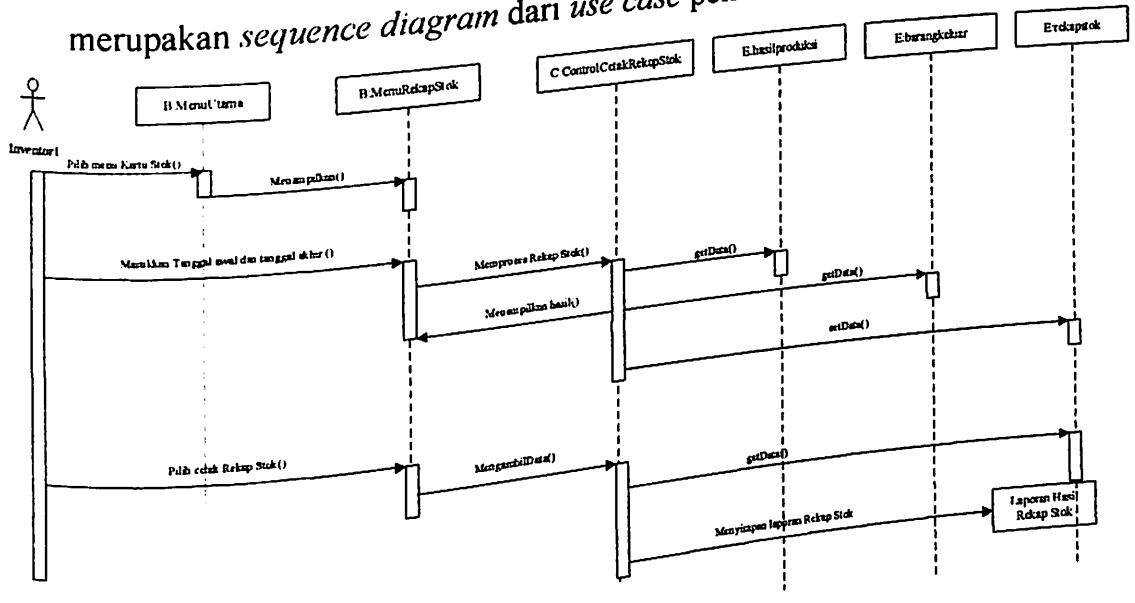


Gambar V.21 *Sequence Diagram* Cetak Surat Jalan

(Sumber: Hasil Analisis, 2018)

9. *Sequence Diagram* Pembuatan Rekap Stok

Sequence diagram pembuatan rekap stok merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses cetak rekap stok di Inventori. Proses pembuatan rekap stok dalam sistem informasi pengendalian persediaan ini dilakukan oleh user yaitu staf Inventori. Berikut merupakan *sequence diagram* dari *use case* pembuatan rekap stok:

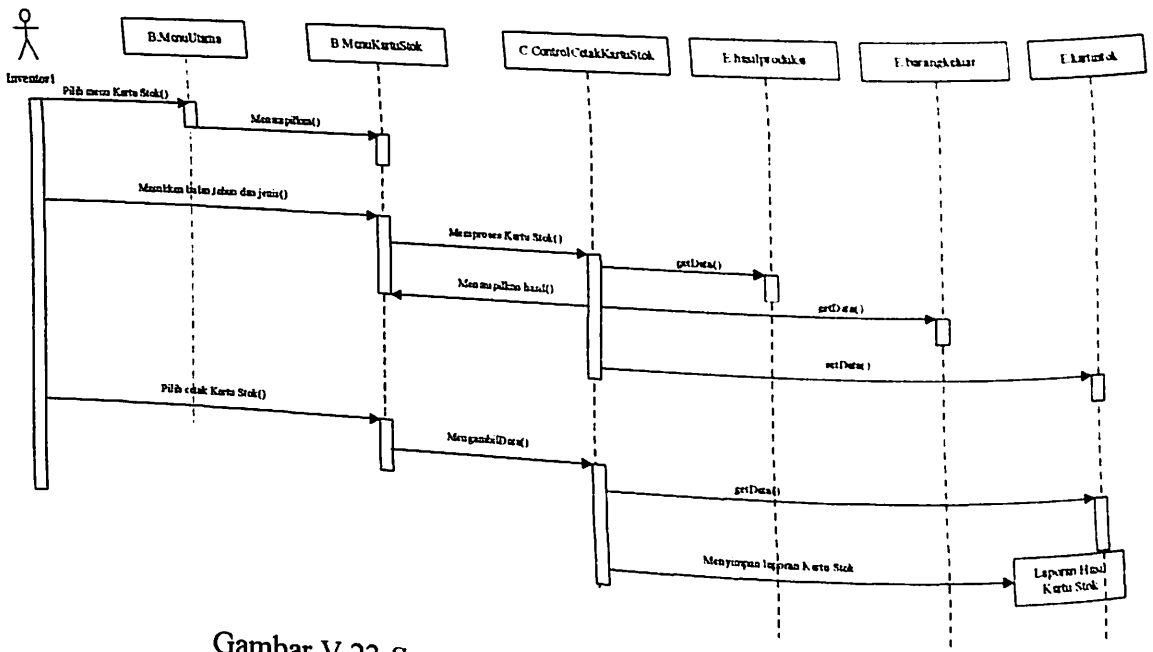


Gambar V.22 *Sequence Diagram* Pembuatan Rekap Stok

(Sumber: Hasil Analisis, 2018)

10. *Sequence Diagram* Pembuatan Kartu Stok

Sequence diagram pembuatan kartu stok merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses pembuatan kartu stok di Inventori. Proses pembuatan kartu stok dalam sistem informasi persediaan ini dilakukan oleh user yaitu staf Inventori. Berikut merupakan *sequence diagram* dari *use case* pembuatan kartu stok:

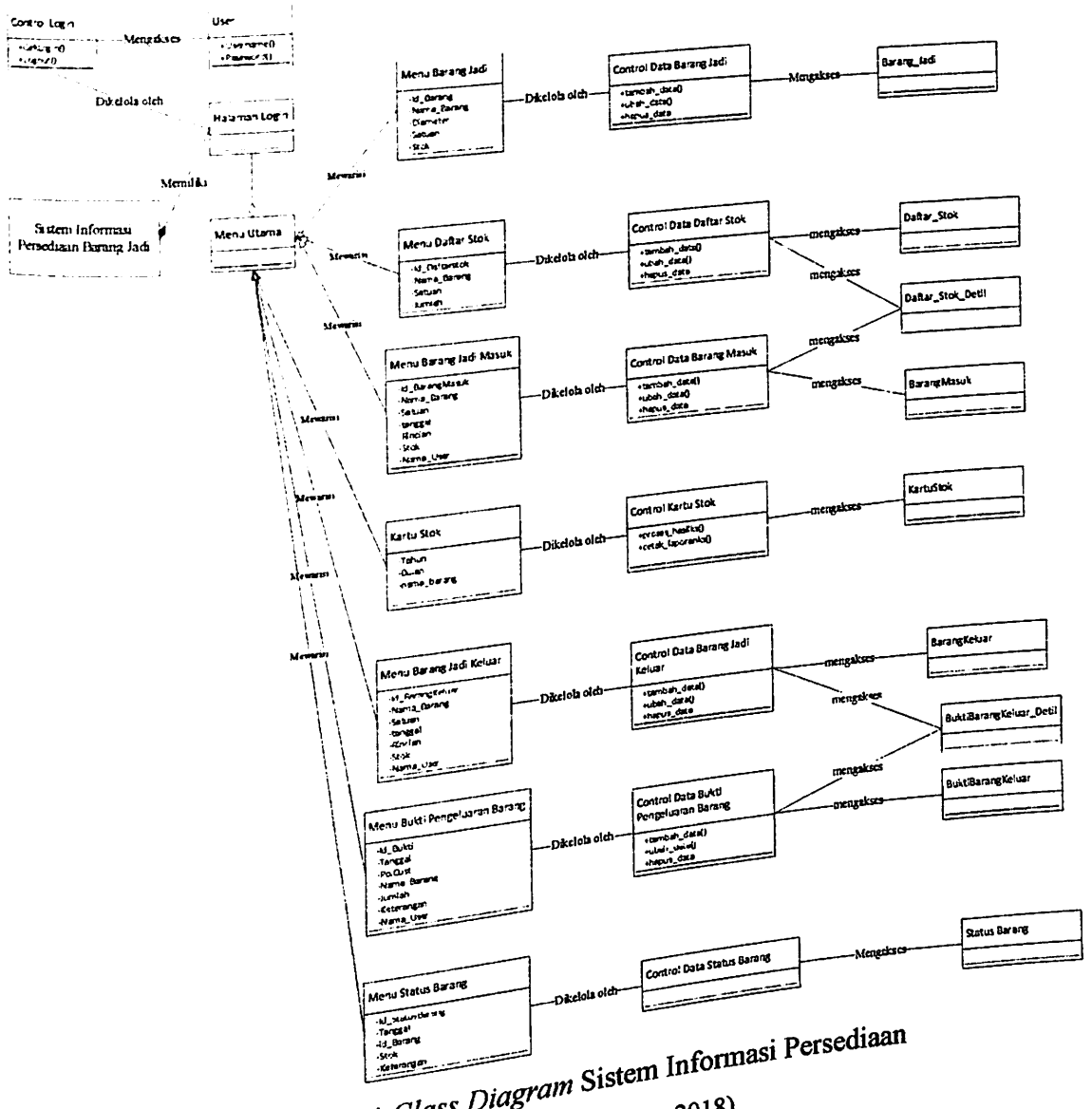


Gambar V.23 *Sequence Diagram* Pembuatan Kartu Stok

(Sumber: Hasil Analisis, 2018)

5.6 *Class Diagram*

Class diagram pada usulan sistem informasi pengendalian persediaan 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 pengendalian persediaan:

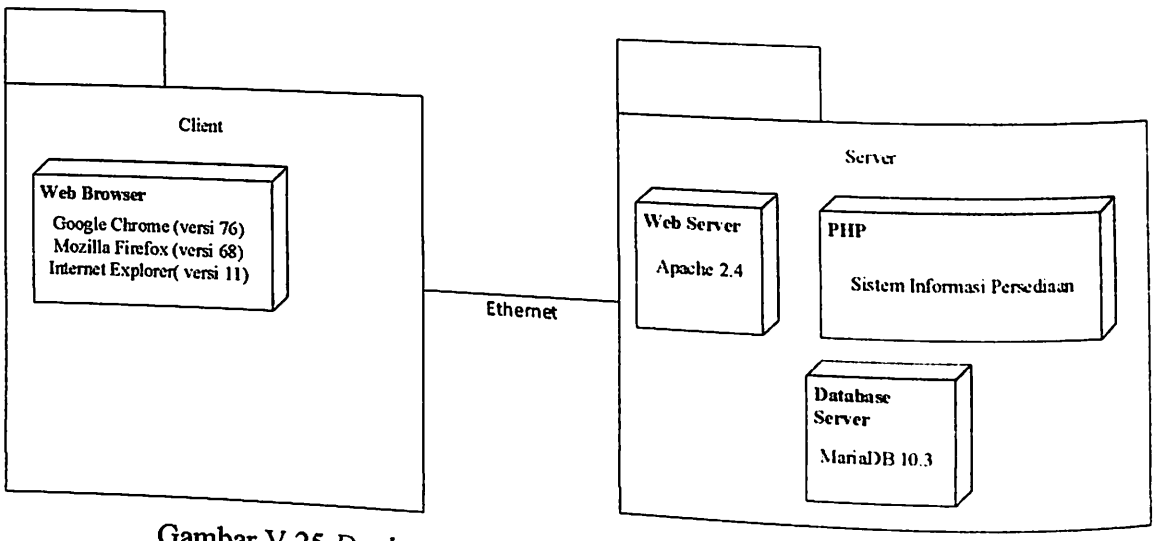


Gambar V.24 *Class Diagram* Sistem Informasi Persediaan

(Sumber: Hasil Analisis, 2018)

5.7 *Deployment Diagram*

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



Gambar V.25 *Deployment Diagram* Sistem Informasi Persediaan

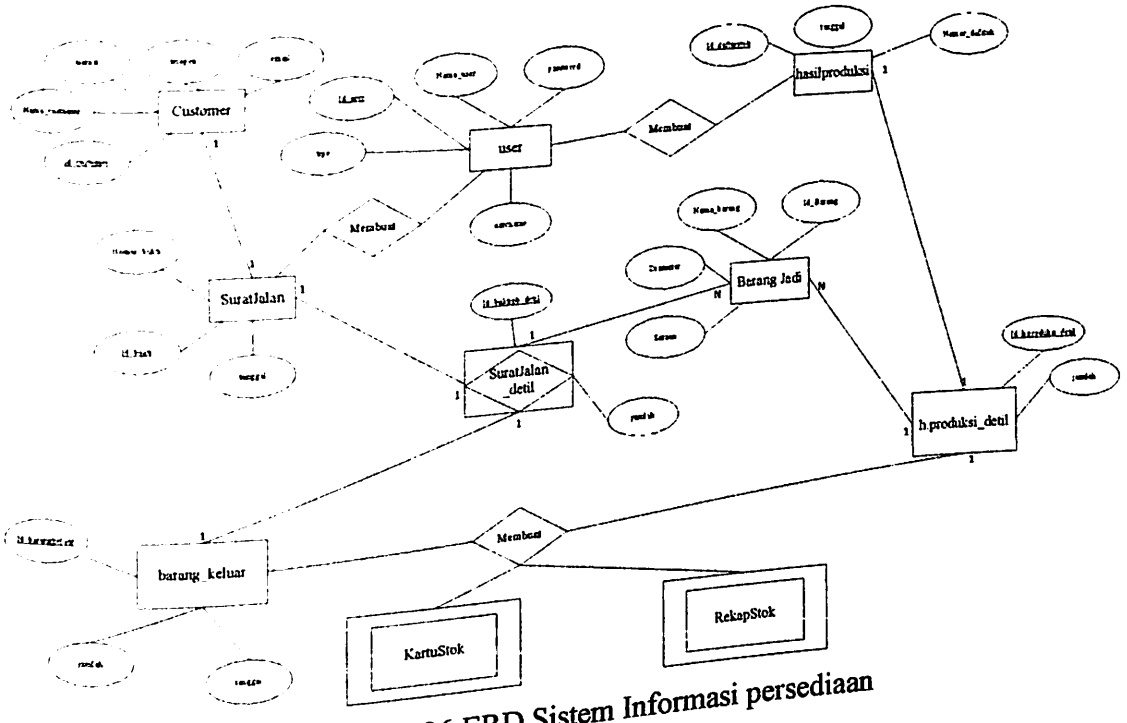
(Sumber: Hasil Analisis, 2018)

5.8 Pemodelan Data Sistem Usulan

Pemodelan data pada sistem informasi rekrutmen karyawan di PT Baneex Indonesia 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 pengendalian persediaan usulan:



Gambar V.26 ERD Sistem Informasi persediaan

(Sumber: Hasil Analisis, 2018)

5.8.2 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 pengendalian persediaan:

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

Tabel V.12 Tabel Kamus Data *user*

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1.	id_user	Id user	Varchar	10	*Primary Key*
2.	*Username*	*username*	Varchar	35	
2.	*Password*	*password*	Varchar	35	

Tabel V.12 Tabel Kamus Data *user* (Lanjutan)

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
3	Nama_user	Nama	Varchar	50	
4	Tipe	Tipe	Int	1	

(Sumber: Hasil Analisis, 2018)

2. Spesifikasi tabel data barang
   a. Nama tabel : data_barang
   b. Fungsi : Untuk menyimpan data barang
   c. Tipe : File data *master*

Tabel V.12 Tabel Kamus Data data_barang

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1.	Id_barang	Id barang	Varchar	10	*Primary Key*
2.	Nama_barang	Nama barang	Varchar	50	
3.	Diameter	Diameter	Float	5	
4.	Satuan	Satuan	Varchar	15	
5	Stok	Stok	Int	15	

(Sumber: Hasil Analisis, 2018)

3. Spesifikasi table data hasil produksi
   a. Nama table : hasilproduksi
   b. Fungsi : Untuk menyimpan data hasil produksi
   c. Tipe : File data *transaksi*

Tabel V.13 Tabel Kamus Data hasilproduksi

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1.	Id_produksi	Id Hasil Produksi	Int	11	*Primary Key*
2.	No_produksi	Nomor Hasil Produksi	Varchar	20	

Tabel V.13 Tabel Kamus Data hasilproduksi (Lanjutan)

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
3.	Tanggal	Tanggal	Date		
4.	Id_User	Id User	Varchar	10	*Foreign Key*
5.	status	status	Int	1	

(Sumber: Hasil Analisis, 2018)

4. Spesifikasi tabel data hasil produksi detil
   a. Nama tabel : h.produksi_detil
   b. Fungsi : Untuk menyimpan data daftarstok detil
   c. Tipe : File data transaksi

Tabel V.14 Tabel Kamus Data h.produksi_detil

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1.	Id_prod_detil	Id produksi detil	Int	11	*Primary Key*
2.	Id_barang	Id Barang Jadi	Varchar	10	*Foreign Key*
3.	Jumlah	Jumlah	Int	10	
4.	Id_produksi	Id Hasil Produksi	Int	10	*Foreign Key*

(Sumber: Hasil Analisis, 2018

5. Spesifikasi tabel data *customer*
   d. Nama tabel : data_*customer*
   e. Fungsi : Untuk menyimpan data *customer*
   f. Tipe : File data *master*

Tabel V.12 Tabel Kamus Data *customer*

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1.	id_customer	Id user	Varchar	10	*Primary Key*
2.	nama_*customer*	Nama *customer*	Varchar	35	

Tabel V.12 Tabel Kamus Data *customer* (Lanjutan)

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
2.	alamat	alamat	Varchar	50	
3	telepon	Telepon	Varchar	13	
4	*email*	*email*	Int	1	

(Sumber: Hasil Analisis, 2018)

6. Spesifikasi tabel data surat jalan

   d. Nama tabel : suratjalan

   e. Fungsi : Untuk menyimpan data suratjalan

   f. Tipe : File data *transaksi*

Tabel V.16 Tabel Kamus Data Surat Jalan

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1.	Id_surat	Id Surat Jalan	Int	10	*Primary Key*
2.	No_surat	Nomor Surat Jalan	Varchar	15	
3.	No_po	Nomor PO	Varchar	20	
4.	Tanggal	Tanggal	Date		
5.	Id_Customer	Id Customer	Varchar	10	*Foreign Key*
6.	Status	Status	Int	1	

(Sumber: Hasil Analisis, 2018)

7. Spesifikasi tabel data surat jalan detil

   d. Nama tabel : suratj_detil

   e. Fungsi : Untuk menyimpan data suratj_detil

   f. Tipe : File data transaksi

Tabel V.17 Tabel Kamus Data suratj_detil

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1.	Id_surat_detil	Id surat jalan detil	Int	11	*Primary Key*
2.	Id_barang	Id barang	Varchar	10	*Foreign Key*
3.	Jumlah	Jumlah	Int	10	
4.	Id_surat	Id surat jalan	Int	10	*Foreign Key*

(Sumber: Hasil Analisis, 2018)

8. Spesifikasi tabel data barang keluar
   a. Nama tabel : barangkeluar
   b. Fungsi : Untuk menyimpan data barang keluar
   c. Tipe : File data transaksi

Tabel V.18 Tabel Kamus Data barangkeluar

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1.	Id_barangkeluar	Id barangkeluar	Int	10	*Primary Key*
2.	Id_bukti_detil	Id bukti detil	Varchar	10	*Foreign Key*
3.	Tanggal	Tanggal	Date		
4.	Jumlah	Jumlah	Int	10	

(Sumber: Hasil Analisis, 2018)

9. Spesifikasi tabel data rekap stok
   d. Nama tabel : rekapstok
   e. Fungsi : Untuk menyimpan data rekap stok
   f. Tipe : File data transaksi

Tabel V.18 Tabel Kamus Data rekapstok

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1.	Id_rekap	Id rekap stok	Int	10	*Primary Key*

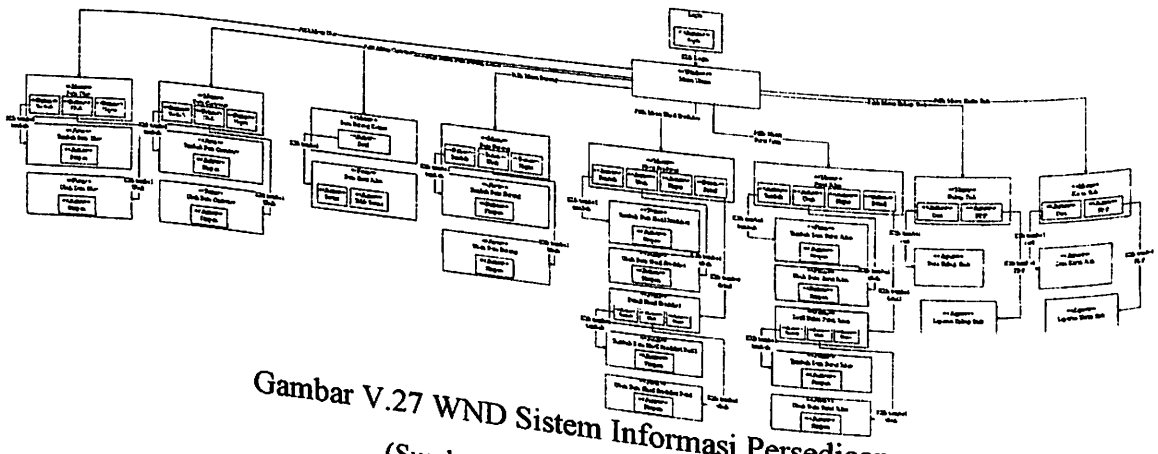
Tabel V.18 Tabel Kamus Data rekapstok (Lanjutan)

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
2.	Id_barang	Id bukti detil	Varchar	10	*Foreign Key*
3.	Tanggal	Tanggal	Date		
4.	stok_awal	Jumlah	Int	15	
5	Masuk	Masuk	Int	10	
6	Keluar	Keluar	Int	10	
7	stok_akhir	Stok Akhir	Int	15	

(Sumber: Hasil Analisis, 2018)

5.9 *Windows Navigation Diagram (WND)*

Windows Navigation Diagram pada sistem usulan digunakan untuk menunjukan bagaimana navigasi dari halaman-halaman yang terdapat di dalam aplikasi, berikut merupakan *Windows Navigation Diagram* sistem informasi pengendalian persediaan.



Gambar V.27 WND Sistem Informasi Persediaan

(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 pengendalian persediaan:

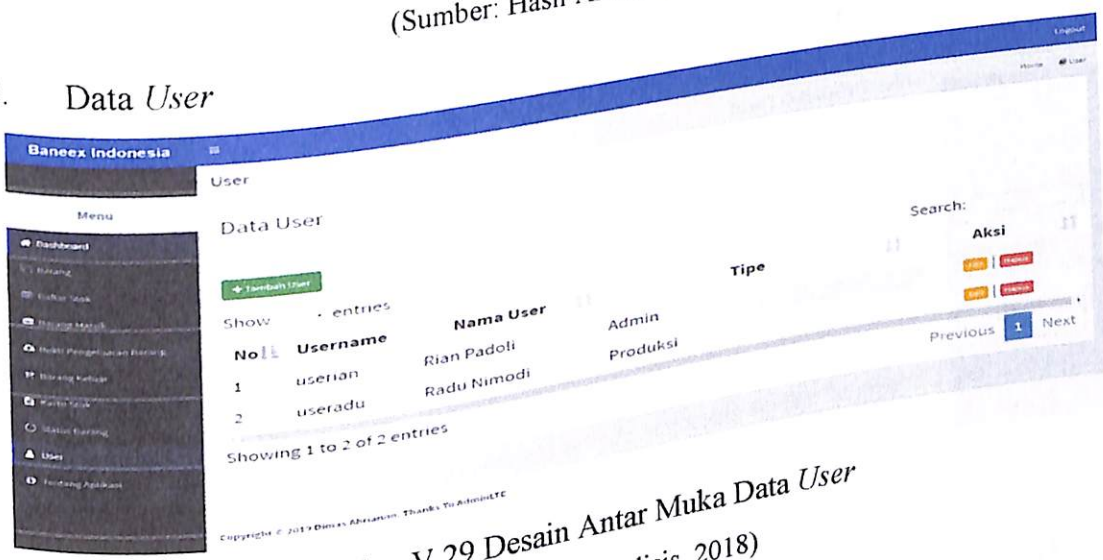
1. *Login*



Gambar V.28 Desain Antar Muka *Login*

(Sumber: Hasil Analisis, 2018)

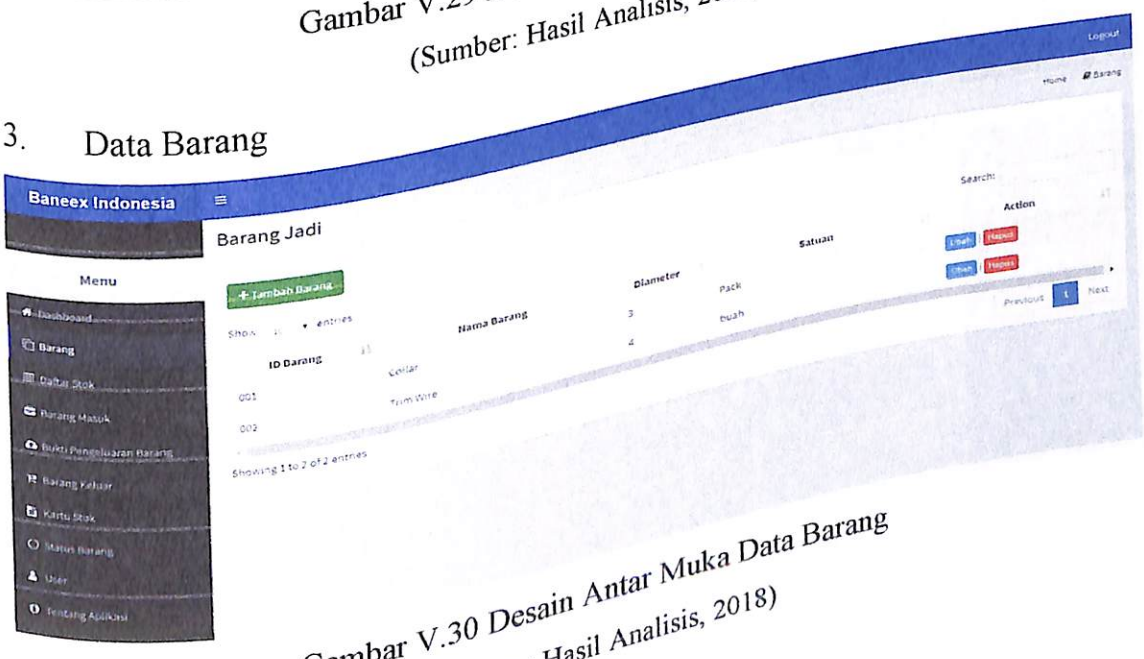
2. Data *User*



Gambar V.29 Desain Antar Muka Data *User*

(Sumber: Hasil Analisis, 2018)

3. Data Barang



Gambar V.30 Desain Antar Muka Data Barang

(Sumber: Hasil Analisis, 2018)

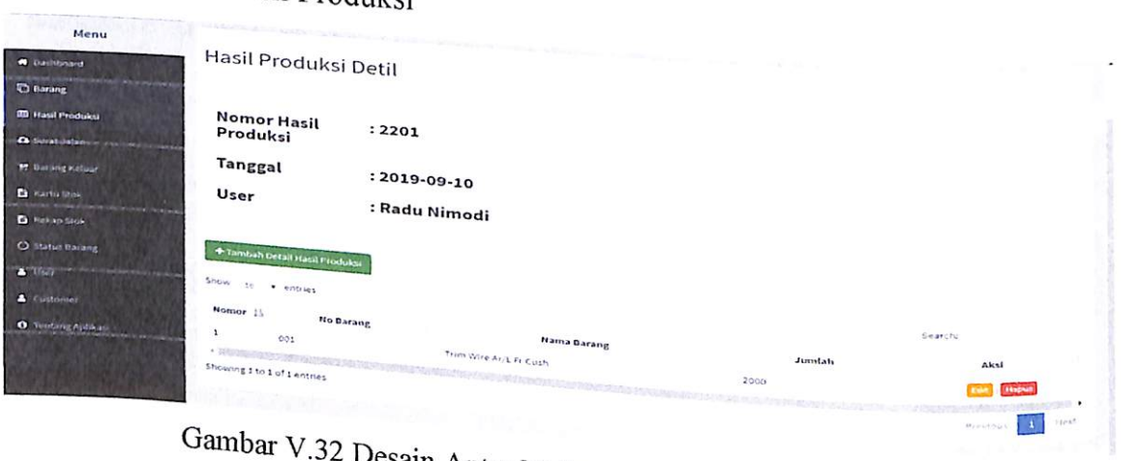
4. Data Hasil Produksi



Gambar V.31 Desain Antar Muka Data Hasil Produksi

(Sumber: Hasil Analisis, 2018)

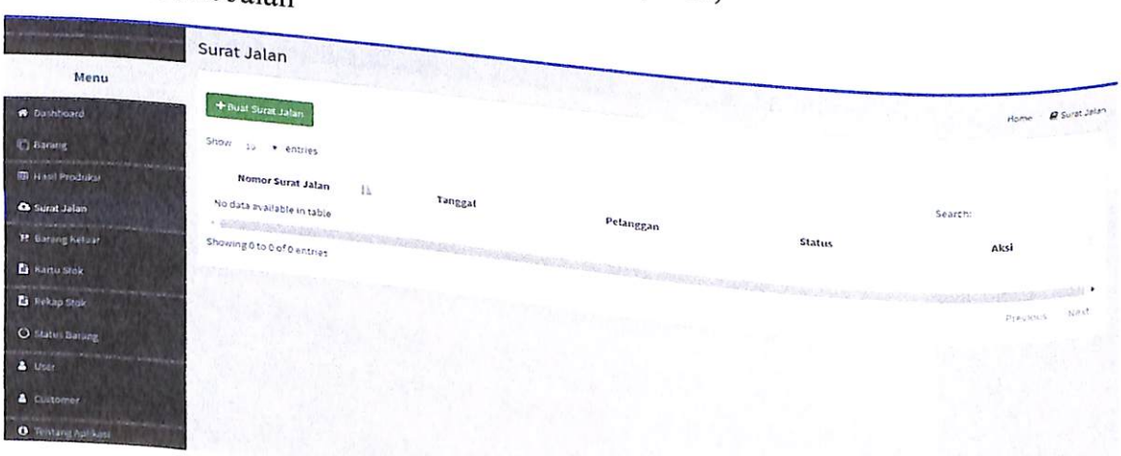
5. Data Detil Hasil Produksi



Gambar V.32 Desain Antar Muka Data Hasil Produksi Detil

(Sumber: Hasil Analisis, 2018)

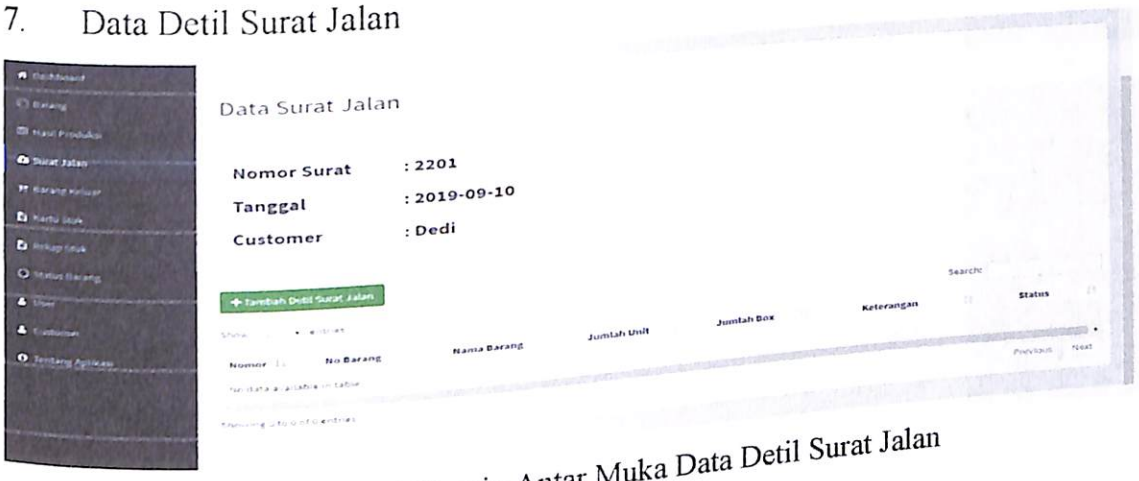
6. Data Surat Jalan



Gambar V.33 Desain Antar Muka Data Surat Jalan

(Sumber: Hasil Analisis, 2018)

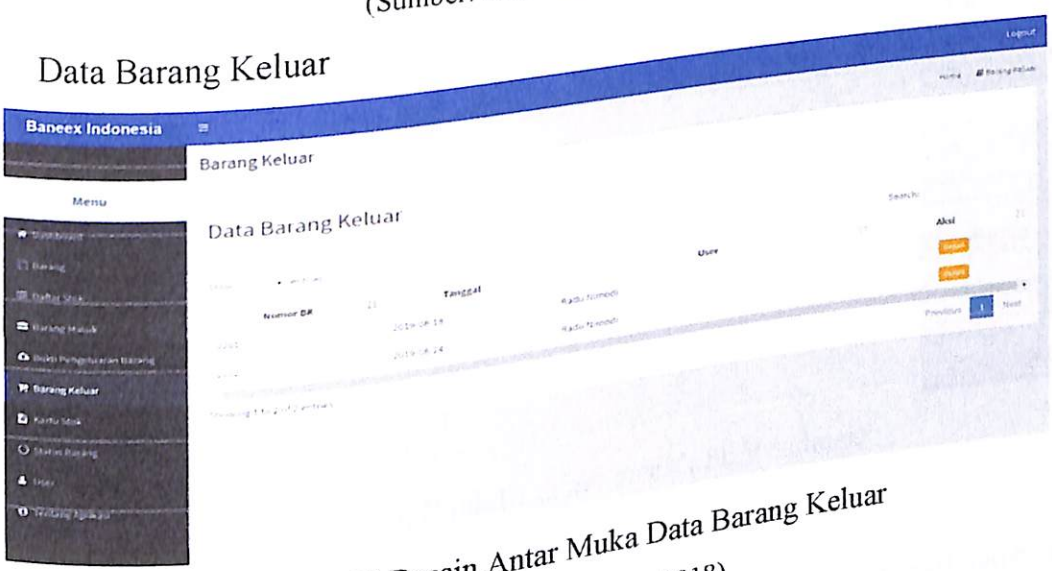
7. Data Detil Surat Jalan



Gambar V.34 Desain Antar Muka Data Detil Surat Jalan

(Sumber: Hasil Analisis, 2018)

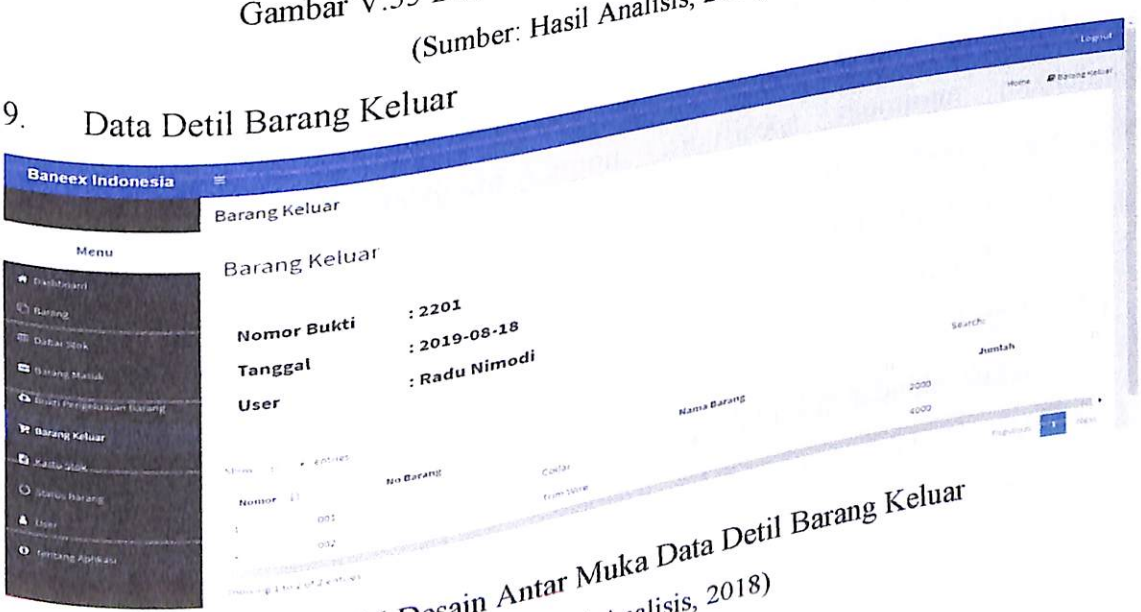
8. Data Barang Keluar



Gambar V.35 Desain Antar Muka Data Barang Keluar

(Sumber: Hasil Analisis, 2018)

9. Data Detil Barang Keluar



Gambar V.36 Desain Antar Muka Data Detil Barang Keluar

(Sumber: Hasil Analisis, 2018)

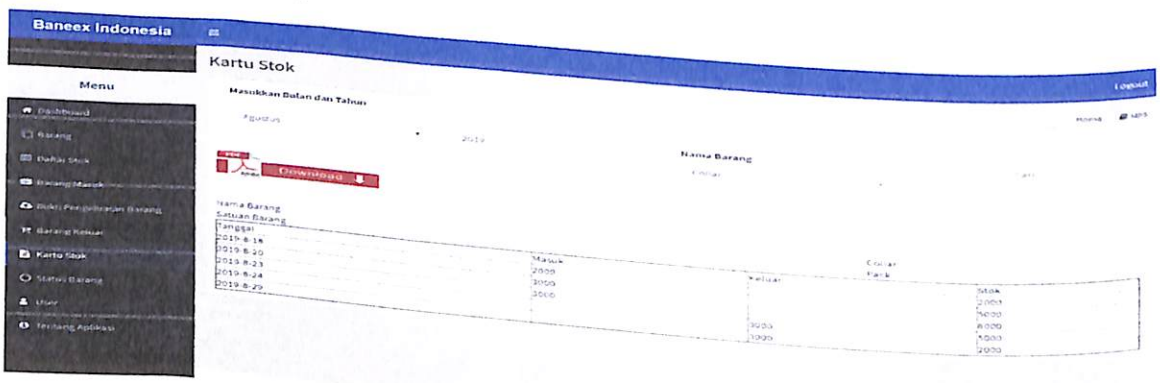
10. Data Rekap Stok



Gambar V.37 Desain Antar Muka Data Rekap Stok

(Sumber: Hasil Analisis, 2018)

11. Data Kartu Stok



Gambar V.38 Desain Antar Muka Data Kartu Stok

(Sumber: Hasil Analisis, 2018)

5.11 Spesifikasi Pengembangan Sistem

Untuk mengimplementasi sistem informasi pengendalian persediaan ini dibutuhkan minimum spesifikasi untuk menjalankan sistem informasi pengendalian persediaan 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

MILIK PERPUSTAKAAN STMI
Membaca : Ibadah, Mengambil : Dosa

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 persediaan barang jadi yang dibuat berdasarkan proses bisnis yang berjalan dan permasalahan yang ditemukan di PT Baneex Indonesia ini diharapkan dapat menjaga persediaan barang jadi dengan metode *Buffer Stock* dan *Reorder Point* pada divisi Inventori dan diharapkan dapat menunjang proses bisnis perusahaan dan menyelesaikan permasalahan yang ada pada proses persediaan barang jadi pada sistem yang berjalan.
2. Sistem informasi persediaan barang jadi 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 persediaan barang jadi.
2. Melakukan pengembangan aplikasi lebih lanjut seperti pengembangan pengelolaan laporan kartu stok bulanan dan rekap stok harian.

DAFTAR PUSTAKA

Assauri, S., 2016, *Manajemen Operasi Produksi Edisi 3,* PT Raja Grafindo Persada, Jakarta.

Dadan & Developers, Kerendi., 2015, *Membuat CMS Multifitur*, PT Elex Media Komputindo, Jakarta.

Anggraeni, E. Y. & Irviani, R., 2017, *Pengantar Sistem Informasi,* CV Andi Offset, Yogyakarta.

Rusdiana & Irfan, M., 2014, *Sistem Informasi Manajemen*, CV Pustaka Setia, Bandung.

Darmawan, Deni dan Kunkun Nur Fauzi. 2016. *Sistem Informasi Manajemen.* Bandung: PT Remaja Rosdakarya

Warman, J., 2012, *Warehouse Management Cetakan Ketujuh*, Pustaka Sinar Harapan, Jakarta.

Nur, R. & Suyuti, M. A., 2017, *Pengantar Sistem Manufaktur*, CV Budi Utama, Yogyakarta.

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.

H.M, Jogiyanto. 2010. *Analisis dan Desain Sistem Informasi.* Yogyakarta: Andi Offset.

Djahir, Y. & Pratita, D., 2014, *Sistem Informasi Manajemen,* Deepublish, Yogyakarta.

Jalinus, N. & Ambiyar., 2016, *Media dan Sumber Pembelajaran*, Kencana, Jakarta

Sitorus, L., 2015, *Algoritma dan Pemrograman*, Andi Offset, Yogyakarta.

Kadir, Abdul. 2014. *Pengenalan Sistem Informasi Edisi Revisi*. Yogyakarta: Andi.

Muslihudin, Muhammad., dan Oktafianto. 2016. *Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML*. Yogyakarta: Andi

Hutahaean, J., 2015, *Konsep Informasi Sistem*, Deepublish, Yogyakarta.

Mulyani, S., 2016, *Sistem Informasi Manajemen Rumah Sakit: Analisis dan Perancangan*, Abdi Sistematika, Bandung.

Indrajani., 2015, *Database Design*, PT Elex Media Komputindo, Jakarta.

Enterprise, J., 2015., *Membuat Website PHP dengan CodeIgniter*, PT Elex Media Komputindo, Jakarta.

Mustaqbal M. Sidi & Firdaus F. Roeri, 2015, 'Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis'.

Pressman, Roger S, (2010), Rekayasa Perangkat Lunak: Pendekatan Praktisi (Buku Satu), Yogyakarta:Andi,

Ragil, Wukil. 2010. *Pedoman Sosialisasi Prosedur Standar*. Jakarta: Mitra Wacana Media

Sutabri, Tata. 2012. *Konsep Sistem Informasi*. Yogyakarta: Andi.

Utami, Feri H. & Asnawati, 2015, *Rekayasa Perangkat Lunak*, Deepublish, Yogyakarta.

Sutaji, Deni, 2012, *Sistem Inventory Mini Market dengan PHP & Jquery*, Lokomedia: Yogyakarta

Tyoso, Jaluanto Sunu Punjul, 2016, *Sistem Informasi Manajemen*, Deepublish, Yogyakarta.

Yanto, R., 2016, *Manajemen Basis Data Menggunakan MySQL*, Deepublish, Yogyakarta.

Martono. R.. 2015. *Manajemen Logistik Terintegrasi*, PPM, Jakarta.

Rukun. Kasman dan Hayadi, B. Herawan, 2018, *Sistem Informasi Berbasis Expert System*. Deepublish. Yogyakarta.

Supono & Putratama. V., 2016. *Pemrograman Web dengan Menggunakan PHP dan Framework Code Igniter*. Deepublish, Yogyakarta.

Wardana., 2016. *Aplikasi Website Profesional dengan PHP dan jQuery*, PT Elex Media Komputindo. Jakarta.

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

Situs:

Menetapkan Tema. Judul. dan Alur Latar Belakang Penelitian http://romisatriawahono.net/2009/04/10/penelitian-tugas-akhir-itu-mudah-3-menetapkan-tema-dan-judul/ (Tanggal Akses: 10 Januari 2019)

http://romisatriawahono.net/2012/06/18/kiat-menyusun-alur-latar-belakang-masalah-penelitian/ (Tanggal Akses: 10 Januari 2019)

Jurnal Ilmiah:

Agus Wantoro. Imam Alkarim. Aplikasi pengendalian persediaan spare part traktor dengan metode *buffer stock* dan *reorder point* (ROP) di gudang cabang tanjung karang

Rahmad Harjono, Yudha Prasetyawan. 2015. Perancangan tata letak gudang untuk meminimumkan jumlah produk yang tidak tertampung dalam blok dan efisiensi aktivitas perpindahan barang di divisi penyimpanan produk jadi PT. ISM BOGASARI FLOUR MILLS SURABAYA

LAMPIRAN A

JADWAL WAWANCARA

Tanggal	Jam	Pewawancara	Yang diwawancarai (Jabatannya)	Lokasi Wawancara	Topik Wawancara
13/02/2018	10:00	Dimas A.	Bpk Dadang (*Dept. Head* Produksi)	Area Produksi	- Proses pengiriman barang jadi dari produksi ke bagian inventori
14/02/2018	10:30	Dimas A	Sekar (Staff Inventori)	Kantor Inventori	- Proses penerimaan bahan baku

TRANSKRIP WAWANCARA

Narasumber : Bpk Dadang (*Dept. Head* Produksi)

Tempat & Waktu Wawancara : Ruang kantor , 13/02/2018

Jam : 10:00 WIB – 10:20 WIB

Topik : Proses pengiriman barang jadi dari produksi ke bagian inventori

Pewawancara	:Selamat pagi pak Dadang, saya ingin menanyakan tentang bagaimana proses pengiriman barang jadi dari bagian produksi ke bagian inventori.
Narasumber	:iya jadi begini, ketika barang jadi telah selesai di produksi, barang jadi di cek sama qc, kalo gak ada cacat, kita isi tuh buku daftar stok sesuai jumlah sama barangnya yg jadi, kalo ada cacat ya di identifikasi aja rusaknya apa, kalo masih bisa di repair ya dikembaliin lagi ke kita bagian produksi. Nah kalo udah selesai diisi tuh buku daftar stoknya, kita kasih bukunya ke inventori buat dia catet di bagiannya nah kalo udah sesuai sih yaudah langsung aja di taruh di gudang.
Pewawancara	:oh jadi begitu Pak, proses selanjutnya jika selesai dari mengisi buku daftar stoknya Pak?
Narasumber	:Nah kalo udah selesai diisi tuh buku daftar stoknya, kita kasih bukunya ke inventori buat dia catet di bagiannya nah kalo udah sesuai sih yaudah langsung aja di taruh di gudang.
Pewawancara	:hmmm, jadi diisinya di buku ya pak?

Narasumber	:iya di buku, soalnya dulu pernah diisi di kertas pisah gitu kan hvs, sering hilang di sini, soalnya banyak angina mesin gitu jadi kalo terbang ga ketahuan, makanya diganti jadi buku.
Pewawancara	:oh begitu pak, lalu pernah ga sih ada salah data yang dibuku sama di bagian inventori?
Narasumber	:hmm disebut sering sih engga, jarang juga engga. Tapi emang suka beda tuh datanya.
Pewawancara	:hmm begitu ya pak, baik terimakasih ya pak atas waktunya. Saya pamit dulu.
Narasumber	:iya sama-sama.

Narasumber : Sekar(Staff Inventori)

Tempat & Waktu Wawancara : Kantor Inventori, 14-02-2018

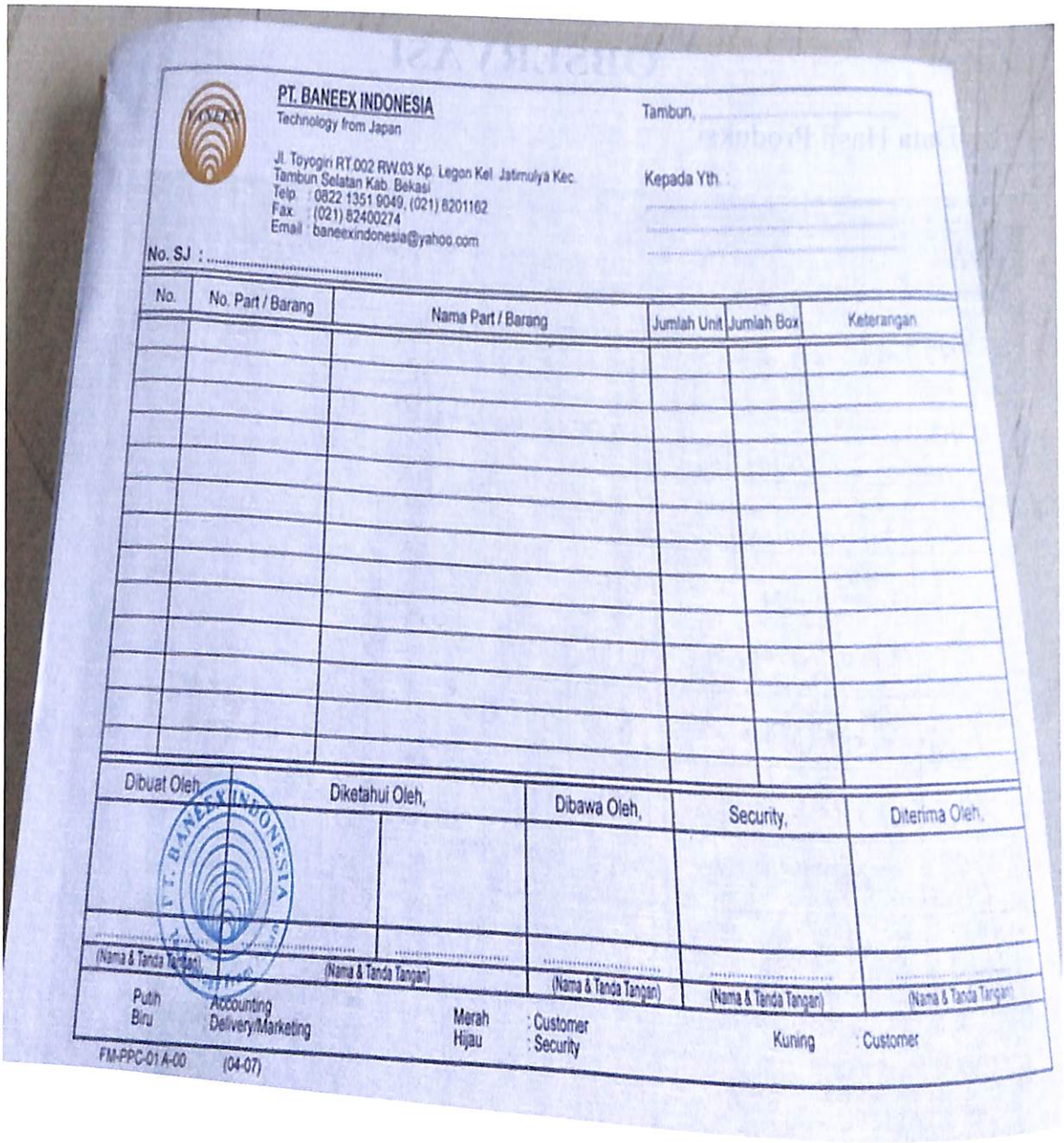
Jam : 10:30 WIB – 10:45 WIB

Topik : Proses masuk dan keluar barang jadi

Pewawancara	:selamat pagi Mba Sekar, saya mau tanya Mba soal bagaimana proses penyimpanan barang jadi dari produksi Mba?
Narasumber	:Oh, barang jadi yang udah selesai produksi tuh dicatet dulu kan di buku, nah bukunya itu kalo selesai nyatet dikasih saya buat saya catat di kartu stok.
Pewawancara	:Oh jadi Mba tulis ulang yang ada di buku ke kartu stok. Lalu mba suka cek ga ke produksi barangnya sesuai apa tidak?

Pewawancara	:Hmmm jadi mba validasi juga kan bener apa engga di lapangan?. Terus kalo ada kesalahan gimana mba?
Narasumber	:iya aku harus cek biar sesuai dengan kartu stok, nah kalo salah ya paling Cuma aku coret atau di tipex aja terus diganti gitu.
Pewawancara	:Hooo, oiya tadi kan soal penyimpanan ya mba, saya mau Tanya kalo keluarnya gimana ya?
Narasumber	:Hmm kalo keluar barang tuh nunggu dari PPC buat bikin Bukti Permohonan Pengeluaran Barang sih untuk dikirim, disitu jelas deskripsi sama jumlah barang yang mau dikirim, jadi nanti aku tinggal validasi aja sesuai apa engga, abis itu *update* kartu stok terus barang dikeluarin deh lalu dikirim ke pelanggan.
Pewawancara	: Oh jadi, Mba Cuma cek aja ya barang yang keluar sesuai ga sama barang yg diminta keluar sama PPC?
Narasumber	: iya cek sama tadi *update* kartu stok.
Pewawancara	: Oke mba, makasih ya penjelasannya saya rasa cukup. Selamat Bekerja.
Narasumber	: Iyaa sama sama juga, sukses ya skripsinya.

2. Surat Jalan



PT. BANEEX INDONESIA
Technology from Japan

Jl. Toyogiri RT.002 RW.03 Kp. Legon Kel. Jatimulya Kec. Tambun Selatan Kab. Bekasi
Telp. : 0822 1351 9049, (021) 8201162
Fax. : (021) 82400274
Email : baneexindonesia@yahoo.com

Tambun, ...

Kepada Yth. :
.....
.....
.....

No. SJ : .....

No.	No. Part / Barang	Nama Part / Barang	Jumlah Unit	Jumlah Box	Keterangan

Dibuat Oleh,	Diketahui Oleh,	Dibawa Oleh,	Security,	Diterima Oleh,
.....	.....	.....	.....	.....
(Nama & Tanda Tangan)	(Nama & Tanda Tangan)	(Nama & Tanda Tangan)	(Nama & Tanda Tangan)	(Nama & Tanda Tangan)

Putih : Accounting
Biru : Delivery/Marketing
Merah : Customer
Hijau : Security
Kuning : Customer

FM-PPC-01 A-00 (04-07)

4. Rekap Stok

REKAP STOCK PRODUK

Data Up date per tgl : .....

Model	Part Name	Part Number	Min Stock*	Jml Stok	KETERANGAN

Keterangan :

Dilaporkan Oleh,

FM-INV-02/Rev00-03.07.17

LAMPIRAN C

Source Code Program

1. Data Barang

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Barang extends CI_Controller {
    function __construct(){
            parent::__construct();
            $this->load->model('pengendaliaan');

    }
    public function index()
    {
            $this->load->view('home/header');
            $y['tab'] = '2';
            $this->load->view('home/sidebar',$y);
            $x['data']=$this->pengendaliaan->show_barang();
            $this->load->view('data_barang/barang',$x);
            $this->load->view('home/footer',);

    }
    public function tambah_barang()
    {
            $id_barang = $this->input->post('id_barang');
            $nama_barang = $this->input->post('nama_barang');
            $diameter = $this->input->post('diameter');
            $satuan = $this->input->post('satuan');
            $data = array(
                    'id_barang' => $id_barang,
                    'nama_barang' => $nama_barang,
                    'diameter' => $diameter,
```

```
                'satuan' => $satuan
                );
        //checking
        $query = $this->db->query("select * from data_barang
where id_barang = 'id_barang'")->num_rows();
        if($query > 0){
                $this->session->set_flashdata('pesan', '<div
class="alert-danger-alert-dimissable">
                <a href="#" class="close"data-
dimiss="alert" aria-label="close">$times;</a><strong>Maaf</strong>
                </div>');
        } else {
                $this->pengendaliaan-
>tambah_barang($data,'data_barang');
                $this->session->set_flashdata('pesan', '<div
class="alert-danger-alert-dimissable">
                <a href="#" class="close"data-
dimiss="alert"
                                                aria-
label="close">$times;</a><strong>Berhasil</strong>
                        Menambah Barang.
                </div>');
    }

                redirect('barang');
    }
    public function ubah_barang($id_barang)
    {
        $this->load->view('home/header');
        $y['tab'] = '2';
        $this->load->view('home/sidebar',$y);
        $where = array('id_barang' => $id_barang);
```

```
        $data['barang']          =          $this->pengendaliaan-
>tampil_barang_by_id($where,'data_barang')->result();
        $this->load->view('data_barang/barang_ubah',$data);
        $this->load->view('home/footer');

    }
    public function ubah_barang_save()
    {       $id_barang = $this->input->post('id_barang');
        $nama_barang = $this->input->post('nama_barang');
        $diamater = $this->input->post('diameter');
        $satuan = $this->input->post('satuan');
        $data = array(
                'nama_barang' => $nama_barang,
                'diameter' => $diameter,
                'satuan' => $satuan


        );

        $where = array(
                'id_barang' => $id_barang

        );
        $this->pengendaliaan-
>ubah_barang($where,$data,'data_barang');
        $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 Barang Berhasil Di Ubah.
        </div>');
        redirect('barang');

    }
    public function hapus_barang($id_barang)
```

```
    {
        $this->load->view('home/header');
        $y['tab'] = '2';
        $this->load->view('home/sidebar',$y);
        $where = array('id_barang' => $id_barang);
        $data['barang'] = $this->pengendaliaan-
>tampil_barang_by_id($where,'data_barang')->result();
     $this->load->view('data_barang/barang_hapus',$data);
        $this->load->view('home/footer');
    }

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

        $this->pengendaliaan-
>hapus_barang($where,'data_barang');
        $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 Barang Berhasil Di Hapus.
        </div>');
        redirect('barang');
    }
    }
```

2. Daftar Stok

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

```
class Daftarstok extends CI_Controller {

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

            }
            */

     $this->load->model('pengendaliaan');
    }

    public function index()
    {
            $this->load->view('home/header');
            $y['tab'] = '3';
            $this->load->view('home/sidebar',$y);
            $x['data']=$this->pengendaliaan->show_dafstok();
     $this->load->view('produksi/daftarstok',$x);
            $this->load->view('home/footer');

    }

    public function detail_dafstok()
    {
            $this->load->helper('form_helper');
            $id_daftarstok = $this->uri->segment(3);
            $this->load->view('home/header');
            $y['tab'] = '3';
            $this->load->view('home/sidebar',$y);
```

```
            $x['data_dafstok']=$this->pengendaliaan-
>cari_dafstok($id_daftarstok);
            $x['detail_dafstok']=$this->pengendaliaan-
>cari_detail_dafstok($id_daftarstok);
            $x['dd_barang'] = $this->pengendaliaan->search_barang();
            $x['id_daftarstok'] = $id_daftarstok;
            $x['barang_selected']  =  $this->input->post('provinsi')  ?
$this->input->post('provinsi') : '';
     $this->load->view('produksi/dafstok_detail',$x);
            $this->load->view('home/footer');
    }

    public function add()
    {
            $this->load->helper('form_helper');
            $this->load->view('home/header');
            $y['tab'] = '3';
            $data = array(
        'dd_user' => $this->pengendaliaan->search_user(),
                    'user_selected'  =>  $this->input->post('provinsi')  ?
$this->input->post('provinsi') : ''
        );

            $this->load->view('home/sidebar',$y);
     $this->load->view('produksi/dafstok_add',$data);
            $this->load->view('home/footer');
    }

    public function edit($id)
    {
            $this->load->helper('form_helper');
```

```
        $this->load->view('home/header');
        $y['tab'] = '3';
        $data = array(
                'dd_user' => $this->pengendaliaan->search_user(),
        'data_dafstok' => $this->pengendaliaan->cari_dafstok($id)
 );

        $this->load->view('home/sidebar',$y);
$this->load->view('produksi/dafstok_ubah', $data);
        $this->load->view('home/footer');

}

public function edit_dafstok()
{
        $id_daftarstok = $this->input->post('id_daftarstok');
        $no_dafstok = $this->input->post('no_dafstok');
        $tanggal = $this->input->post('tanggal');
        $id_user = $this->input->post('id_user');

        $data = array(
                'id_daftarstok' => $id_daftarstok,
                'no_dafstok' => $no_dafstok,
                'tanggal' => $tanggal,
                'id_user' => $user


         );

         $where = array(
                'id_daftarstok' => $id_daftarstok
```

```
            );

            $query = $this->db->query("select * from daftarstok where
no_dafstok = '$no_dafstok'")->num_rows();
            if($query > 0){
                $this->session->set_flashdata('pesan', '<div
class="alert alert-danger alert-dismissible">
                <a href="#" class="close" data-dismiss="alert"
aria-label="close">&times;</a>
                <strong>Maaf</strong> Data Sudah Ada.
                </div>');
                redirect('daftarstok/edit/'.$id_daftarstok);
            }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>
                Data Detail Purchase Berhasil Di Ubah.
                </div>');
                redirect('daftarstok');
            }
    }

    public function delete_dafstok($id)
    {
            $where = array(
                'id_daftarstok' => $id
            );
```

```
            $this->session->set_flashdata('pesan', '<div class="alert
alert-warning alert-dismissible">
                <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
                Data Purchase Berhasil Di Hapus.
            </div>');
            echo json_encode(array("status" => TRUE));
            //redirect('po/detail_po/'.$no_po);

    }

    public function tambah_dafstok()
    {
            $no_dafstok = $this->input->post('no_dafstok');
            $tanggal = $this->input->post('tanggal');
            $id_user = $this->input->post('id_user');


            $data = array(
                    'id_daftarstok' => '',
                    'no_dafstok' =>$no_dafstok,
                    'tanggal' => $tanggal,
                    'id_user' => $id_user,
                    'status' => '0'

            );
            //cek dulu
            $query = $this->db->query("select * from daftarstok where
no_dafstok = '$no_dafstok'")->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>');
                redirect('daftarstok/add');
                }else{
                        $this->pengendaliaan-
>tambah_dafstok($data,'daftarstok');
                        $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 Purchase Order.
                </div>');
                $query = $this->db->query("select * from daftarstok order
by id_daftarstok DESC limit 1")->row();
                redirect('daftarstok/detail_dafstok/'.$query->id_daftarstok);
                }
        }


        public function tambah_dafstok_detail()
        {
                $id_barang = $this->input->post('id_barang');
                $jumlah = $this->input->post('jumlah');
                $id_daftarstok = $this->input->post('id_daftarstok');

                $data = array(
                        'id_dafstok_detil' => '',
                        'id_barang' => $id_barang,
                        'jumlah' => $jumlah,
                        'id_daftarstok' => $id_daftarstok
                );
                //cek dulu
```

```
            $query = $this->db->query("select * from daftarstok_detil
where id_daftarstok = '$id_daftarstok' and id_barang='$id_barang'")-
>num_rows();
            if($query > 0){
                  $this->session->set_flashdata('pesan',          '<div
class="alert alert-danger alert-dismissible">
                  <a  href="#"  class="close"  data-dismiss="alert"  aria-
label="close">&times;</a>
                  <strong>Maaf</strong> Data Detail Purchase Sudah Ada.
                  </div>');
                  redirect('daftarstok/detail_dafstok/'.$id_daftarstok);
                  }else{
                  $this->pengendaliaan-
>tambah_dafstok($data,'daftarstok_detil');
                  $this->session->set_flashdata('pesan',          '<div
class="alert alert-success alert-dismissible">
                  <a  href="#"  class="close"  data-dismiss="alert"  aria-
label="close">&times;</a>
                  <strong>Berhasil</strong> Menambah Detail Purchase.
                  </div>');
                  redirect('daftarstok/detail_dafstok/'.$id_daftarstok);
                  }
      }
      function search_dafstok_detail($id)
      {
            $data                =          $this->db->query("SELECT
a.id_dafstok_detil,b.id_barang,  b.nama_barang  as  nama_barang,
a.jumlah as jumlah FROM  daftarstok_detil a, data_barang b where
a.id_barang = b.id_barang and a.id_dafstok_detil='$id'")->row();

            echo json_encode($data);
```

```
    }

    public function ubah_dafstok_detail_save()
    {
        $id_dafstok_detil = $this->input->post('id_dafstok_detil');
        $jumlah = $this->input->post('jumlah');
        $id_daftarstok = $this->input->post('id_daftarstok');

        $data = array(
            'jumlah' => $jumlah,
            'id_daftarstok' => $id_daftarstok,
            'status' =>'0'
        );

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

        $this->pengendaliaan-
>ubah_user($where,$data,'daftarstok_detil');

            $data = array(
            'status' => '0'
        );

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

        $this->pengendaliaan-
>ubah_user($where,$data,'daftarstok');
```

```
            $this->session->set_flashdata('pesan', '<div class="alert
alert-success alert-dismissible">
                <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
                Data Detail Purchase Berhasil Di Ubah.
            </div>');
            redirect('daftarstok/detail_dafstok/'.$id_daftarstok);
    }

    public function ubah_dafstok_save()
    {
            $no_dafstok = $this->input->post('no_dafstok');
            $tanggal = $this->input->post('tanggal');

                $data = array(
                'tanggal' => $tanggal,
                'status' => '0'

            );

            $where = array(
                    'no_dafstok' => $no_dafstok

            );
            $this->pengendaliaan-
>ubah_user($where,$data,'daftarstok');
            $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 Detail Purchase Berhasil Di Ubah.
            </div>');
```

```
            redirect('daftarstok');
    }

    public function delete_dafstok_detail_save($id,$no_dafstok)
    {
            $where = array(
                    'id_dafstok_detil' => $id
            );

            $this->pengendaliaan-
>hapus_user($where,'daftarstok_detil');
            $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 Detail Purchase Berhasil Di Hapus.
            </div>');
            echo json_encode(array("status" => TRUE));
     }

}
```

LAMPIRAN D

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 L.1 *Black Box Testing Use Case Login*

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

   Berikut merupakan *Black Box Testing* dari *use case* mengelola data barang jadi

   Penguji : Indri Fitriana Alfia Nisa (1314042)

Tabel L.2 *Black Box Testing Use Case* Mengelola Data Barang jadi

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Menambah data barang jadi	Menambah data barang jadi dengan mengisi semua kolom pada *form* tambah data	Berhasil tambah data	Berhasil tambah data	Valid

Tabel L.2 *Black Box Testing Use Case* Mengelola Data Barang jadi (lanjutan)

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
2	Menambah data barang jadi	Menambah data barang jadi dengan mengosongkan salah satu kolom pada *form* tambah data	Menampilkan pesan "*Please fill out this field*"	Menampilkan pesan "*Please fill out this field*"	Valid
3	Mengubah data barang jadi	Melakukan perubahan pada kolom stok	Data barang jadi berubah	Data barang jadi berubah	Valid
4	Menghapus data barang jadi yang dipilih	Menghapus salah satu data barang jadi	Mengahapus data barang jadi dari tabel barang jadi	Menghapus data barang jadi dari tabel barang jadi	Valid
5	Mencari data barang jadi	Mencari data barang jadi yang sesuai data yang ada di dalam *database*	Menampilkan data barang jadi yang dicari	Menampilkan data barang jadi yang dicari	Valid
6	Mencari data barang jadi	Mencari data barang jadi yang sesuai data yang ada tidak ada di dalam *database*	Menampilkan pesan "*No matching records found*"	Menampilkan pesan "*No matching records found*"	Valid

(Sumber: Hasil Analisis, 2018)

3. *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 L.3 *Black Box Testing Use Case* Mengelola Data *User*

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Menambah data *user*	Menambah data *user* dengan mengisi semua kolom pada *form* tambah data	Berhasil tambah data	Berhasil tambah data	Valid

Tabel L.3 *Black Box Testing Use Case* Mengelola Data *User* (lanjutan)

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

(Sumber: Hasil Analisis, 2018)

4. *Black Box Testing Use Case* Data Hasil produksi

   Berikut merupakan *Black Box Testing* dari *use case* Mengelola Data hasil produksi.

   Penguji : Indri Fitriana Alfia Nisa (1314042)

Tabel L.4 *Black Box Testing Use Case* Data Hasil Produksi

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

Tabel L.4 *Black Box Testing Use Case* Data Hasil Produksi (lanjutan)

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
8	Menambah data *detail* hasil produksi	Menambah data *detail* hasil produksi dengan mengosongkan salah satu kolom pada *form* tambah data	Menampilkan pesan "*Please fill out this field*"	Menampilkan pesan "*Please fill out this field*"	Valid
9	Mengubah data *detail* hasil produksi	Melakukan perubahan pada kolom Nama Qty	Data *detail* hasil produksi berubah	Data *detail* hasil produksi berubah	Valid
10	Menghapus data *detail* hasil produksi yang dipilih	Menghapus salah satu data *detail* hasil produksi yang dipilih	Mengahapus data *detail* hasil produksi dari tabel *detail* hasil produksi	Menghapus data *detail* hasil produksi dari tabel *detail* hasil produksi	Valid
11	Mencari data *detail* hasil produksi	Mencari data *detail* hasil produksi yang sesuai data yang ada di dalam *database*	Menampilkan data *detail* hasil produksi yang dicari	Menampilkan data *detail* hasil produksi yang dicari	Valid

(Sumber: Hasil Analisis, 2018)

5. *Black Box Testing Use Case* Data Surat jalan

Berikut merupakan *Black Box Testing* dari *use case* Mengelola Data surat jalan.

Penguji : Indri Fitriana Alfia Nisa (1314042)

Tabel L.5 *Black Box Testing Use Case* Data Surat Jalan

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Menambah data surat jalan	Menambah data surat jalan dengan mengisi semua kolom pada *form* tambah data	Berhasil tambah data	Berhasil tambah data	Valid

Tabel L.5 *Black Box Testing Use Case* Data Surat Jalan (lanjutan)

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
2	Menambah data surat jalan	Menambah data surat jalan dengan mengosongkan salah satu kolom pada *form* tambah data	Menampilkan pesan "*Please fill out this field*"	Menampilkan pesan "*Please fill out this field*"	Valid
3	Mengubah data surat jalan	Melakukan perubahan pada kolom Tanggal	Data surat jalan berubah	Data surat jalan berubah	Valid
4	Menghapus data surat jalan yang dipilih	Menghapus salah satu data surat jalan yang dipilih	Mengahapus data surat jalan dari tabel surat jalan	Menghapus data surat jalan dari tabel surat jalan	Valid
5	Mencari data surat jalan	Mencari data surat jalan yang sesuai data yang ada di dalam *database*	Menampilkan data surat jalan yang dicari	Menampilkan data surat jalan yang dicari	Valid
6	Menampilkan *detail* surat jalan yang dipilih	Menekan tombol *detail* pada surat jalan yang dipilih	Menampilkan data	Berhasil tambah data	Valid
7	Menambah data *detail* surat jalan	Menambah data *detail* surat jalan dengan mengisi semua kolom pada *form* tambah data	Berhasil tambah data	Berhasil tambah data	Valid
8	Menambah data *detail* surat jalan	Menambah data *detail* surat jalan dengan mengosongkan salah satu kolom pada *form* tambah data	Menampilkan pesan "*Please fill out this field*"	Menampilkan pesan "*Please fill out this field*"	Valid

Tabel L.5 *Black Box Testing Use Case* Data Surat Jalan (lanjutan)

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
9	Mengubah data *detail* surat jalan	Melakukan perubahan pada kolom Nama Qty	Data *detail* surat jalan berubah	Data *detail* surat jalan berubah	Valid
10	Menghapus data *detail* surat jalan yang dipilih	Menghapus salah satu data *detail* surat jalan yang dipilih	Mengahapus data *detail* surat jalan dari tabel *detail* surat jalan	Menghapus data *detail* surat jalan dari tabel *detail* surat jalan	Valid
11	Mencari data *detail* surat jalan	Mencari data *detail* surat jalan yang sesuai data yang ada di dalam *database*	Menampilkan data *detail* surat jalan yang dicari	Menampilkan data *detail* surat jalan yang dicari	Valid

(Sumber: Hasil Analisis, 2018)

6. *Black Box Testing Use Case* Validasi Barang Keluar

Berikut merupakan *Black Box Testing use case* validasi barang keluar

Penguji : Indri Fitriana Alfia Nisa (1314042)

Tabel L.6 *Black Box Testing Use Case* Validasi Barang Keluar

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Memvalidasi Barang Keluar	Memvalidasi barang keluar dengan mengklik *button* sesuai	Data Barang Masuk tersimpan di database	Data Barang Masuk tersimpan di database	Valid
2	Menolak Barang Keluar	Memvalidasi barang keluar dengan mengklik *button* tidak sesuai	Data tidak tersimpan dan status surat jalan menjadi "tidak sesuai"	Data tidak tersimpan dan status surat jalan menjadi "tidak sesuai"	Valid

(Sumber: Hasil Analisis, 2018)

7. *Black Box Testing Use Case* Validasi Barang Masuk

Berikut merupakan *Black Box Testing use case* validasi barang masuk

Penguji : Indri Fitriana Alfia Nisa (1314042)

Tabel V.22 *Black Box Testing Use Case* Validasi Barang Masuk

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Memvalidasi Barang Masuk	Memvalidasi barang masuk dengan mengklik *button* sesuai	Data Barang Masuk tersimpan di database	Data Barang Masuk tersimpan di database	Valid
2	Menolak Barang Masuk	Memvalidasi barang masuk dengan mengklik *button* tidak sesuai	Data tidak tersimpan dan status hasil produksi menjadi "tidak sesuai"	Data tidak tersimpan dan status hasil produksi menjadi "tidak sesuai"	Valid

(Sumber: Hasil Analisis, 2018)

8. *Black Box Testing Use Case* Pembuatan Kartu Stok

Berikut merupakan *Black Box Testing use case* pembuatan Kartu Stok

Penguji : Indri Fitriana Alfia Nisa (1314042)

Tabel V.22 *Black Box Testing Use Case* Pembuatan Kartu Stok

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Memproses Kartu Stok	Memproses Kartu Stok dengan mengisi semua kolom	Menampilkan data Kartu Stok	Menampilkan data Kartu Stok	Valid
2	Memproses Kartu Stok	Memproses Kartu Stok dengan mengosongkan kolom tahun	Tidak menampilkan data Kartu Stok	Tidak menampilkan data Kartu Stok	Valid

(Sumber: Hasil Analisis, 2018)