RANCANG BANGUN SISTEM INFORMASI PENYIMPANAN BARANG JADI PADA DEPARTEMEN *WAREHOUSE* DI PT TEKNIKATAMA KARYA MANDIRI MENGGUNAKAN *CODEIGNITER* 3.1.4 DAN MARIADB 10.1.31

TUGAS AKHIR

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

OLEH

AINALISA ALYA NABELLA

1314014

POLITEKNIK STMI JAKARTA

KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

JAKARTA

2018

POLITEKNIK STMI JAKARTA

KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

LEMBAR PERSETUJUAN DOSEN PEMBIMBING

Judul Tugas Akhir : RANCANG BANGUN SISTEM INFORMASI PENYIMPANAN BARANG JADI PADA DEPARTEMEN *WAREHOUSE* DI PT TEKNIKATAMA KARYA MANDIRI MENGGUNAKAN *CODEIGNITER* 3.1.4 DAN MARIADB 10.1.31

Disusun Oleh

Nama : Ainalisa Alya Nabella

NIM : 1314014

Pogram Studi : Sistem Informasi Industri Otomotif

Telah diperiksa dan disetujui untuk memenuhi sebagian syarat penyelesaian jenjang Diploma Empat (D-4) Program Studi Sistem Informasi Industri Otomotif pada Politeknik STMI Jakarta.

Jakarta, 13 Agustus 2018

Dosen Pembimbing

(Ulil Hamida, ST, MT)

NIP: 19810327 200502 2 001

LEMBAR BIMBINGAN PENYUSUNAN TUGAS AKHIR

Nama : Ainalisa Alya Nabella

NIM : 1314014

Judul Tugas Akhir :Rancang Bangun Sistem Informasi Penyimpanan Barang Jadi Pada Departemen *Warehouse* di PT Teknikatama Karya Mandiri Menggunakan *Codeigniter* 3.1.4 dan MariaDB 10.1.31

Pembimbing : Ulil Hamida, ST, MT.

Tanggal	Keterangan	Paraf
20 April 2018	Pengajuan Proposal Tugas Akhir dan BAB I	
27 April 2018	Revisi BAB I dan Pengajuan BAB II	
9 Mei 2018	Revisi BAB II dan BAB I	
16 Mei 2018	Revisi BAB I, II, dan Pengajuan BAB III	
22 Mei 2018	Bimbingan Program	
28 Mei 2018	Revisi Bab I,II, III dan Pengajuan BAB IV	
31 Mei 2018	Bimbingan Program	
26 Juni 2018	Pengajuan BAB V	
09 Juli 2018	Revisi BAB V (*Activity Diagram*)	
23 Juli 2018	Revisi BAB V dan Bimbingan Program	
31 Juli 2018	Revisi BAB V dan Bimbingan Program	
1 Juli 2018	Revisi BAB IV,V,VI	
7 Juli 2018	Pengajuan Keseluruhan BAB dan Demo Program	
13 Agustus 2018	Revisi Keseluruhan BAB dan Revisi Program	

Mengetahui,
Ketua Program Studi
Sistem Informasi Industri Otomotif
Pembimbing

Drs. Jacob Saragih, MM

NIP. 195404281986031002

Ulil Hamida, ST, MT

NIP. 198103272005022001

ABSTRAK

PT Teknikatama Karya Mandiri merupakan perusahaan manufaktur yang bergerak di bidang industri komponen otomotif khususnya kepada komponen *spare part* kendaraan roda empat. Dalam kegiatan penyimpanan barang jadi di PT Teknikatama Karya Mandiri ini, ditemukan suatu permasalahan dalam sistem penyimpanan barang jadi yang sedang berjalan. Seperti pada pencatatan aktivitas keluar/masuknya barang jadi pada daftar stok sering ditemukannya daftar stok yang tidak dicatat dengan rutin. Belum adanya laporan bulanan dikarenakan data yang diperlukan dalam daftar stok tidak tercatat dengan baik. Pengambilan barang jadi dilakukan secara acak sehingga barang jadi yang pertama kali masuk ke dalam gudang tidak terambil ketika waktu pengiriman tiba hal tersebut dikhawatirkan dapat membuat barang menjadi rusak karena terlalu lama berada di gudang. Maka dari itu, sistem informasi diperlukan untuk pencatatan barang jadi sehingga stok barang dapat terkontrol dengan baik. Membuat suatu sistem informasi dan basis data (*database*) untuk aktivitas keluar/masuknya barang jadi yang mana data tersebut dapat membantu Operator *Warehouse* untuk membuat sebuah laporan dengan cepat. Menerapkan cara pengambilan barang jadi dengan teknik *First In First Out*. Sehingga Operator *Warehouse* dapat mengetahui barang jadi mana saja yang harus dikeluarkan dari dalam gudang. Metodologi pengembangan sistem yang digunakan pada penelitian ini adalah *waterfall*. Adapun pemodelan sistem menggunakan diagram UML antara lain *usecase diagram, activity diagram, sequence diagram, class diagram,* dan *deployment diagram*. Pemodelan data yang digunakan yaitu *diagram* ERD dan kamus data. Sistem informasi ini dibangun menggunakan *CodeIgniter* 3.1.4 dan MariaDB 10.1.31. Sistem ini diharapkan dapat membantu proses pencatatan penyimpanan barang jadi dan sebagai tempat penyimpanan data untuk mencegah terjadinya kehilangan atau kerusakan data.

Kata kunci: Sistem informasi, Penyimpanan Barang Jadi, Penyimpanan, *Waterfall*, *Unified Modeling Language*, *CodeIgniter*, MariaDB , Barang Jadi.

KATA PENGANTAR

Puji Syukur dipanjatkan kepada Allah SWT yang telah melimpahkan rahmat dan karunia-Nya, sehingga dapat diselesaikan Tugas Akhir dengan judul "**Rancang Bangun Sistem Informasi Penyimpanan Barang Jadi Pada Departemen *Warehouse* di PT Teknikatama Karya Mandiri Menggunakan *Codeigniter* 3.1.4 dan MariaDB 10.1.31**".

Penulisan Tugas Akhir ini bertujuan untuk memenuhi sebagian syarat penyelesaian Jenjang Diploma Empat (D-4) pada Program Studi Sistem Informasi Industri Otomotif Politeknik STMI Jakarta.

Telah disadari bahwa Tugas Akhir ini tidak akan terwujud tanpa adanya bantuan, bimbingan serta dukungan dari berbagai pihak. Oleh karena itu, saya mengucapkan terima kasih kepada:

1. Allah SWT yang telah memberikan segala karunia-Nya, berupa kesehatan dan kelancaran dalam penyusunan Tugas Akhir ini.
2. Kedua orang tua saya Bapak Agung Widjaya dan Ibu Suwarti yang selalu memberikan dukungan, atas ridhonya saya dapat menjalankan pendidikan di Politeknik STMI.
3. Bapak Dr. Mustofa, S.T, M.T. Selaku Direktur Politeknik STMI Jakarta.
4. Bapak Drs. Jacob Saragih, M.M. Selaku Ketua Prodi Sistem Informasi Industri Otomotif.
5. Ibu Ulil Hamida, S.T, M.T. Selaku Dosen Pembimbing yang telah membantu, meluangkan waktu untuk membimbing jalannya penulisan dan penyusunan Tugas Akhir.
6. Bapak Anton Saputra selaku Direktur PT Teknikatama Karya Mandiri yang telah mengizinkan penulis untuk melakukan Tugas Lapangan di PT Teknikatama Karya Mandiri.
7. Ibu Irna dan Ibu Cici selaku pembimbing yang telah membantu mengarahkan dan membimbing selama Tugas Lapangan.
8. Bapak Agung. Bapak Indra, Bapak Giri, Bapak Koko dan seluruh pegawai di PT Teknikatama Karya Mandiri yang telah membantu memberi arahan serta informasi untuk pengumpulan data.
9. Seluruh dosen Politeknik STMI Jakarta yang telah memberikan ilmunya untuk menambah pengetahuan dan pengalaman yang sangat bermanfaat.

10. Teman-teman mahasiswa/i Politeknik STMI Jakarta Program Studi Sistem Informasi terutama SA01 2014 atas kebersamaan dan motivasinya selama ini. Khususnya Roni Sudrajat, Indri Fitriana Alfia Nisa, Dimas Pangestu Surwendo, Rusdiansyah, Fadhla Fanini, Fakhri Tri, Dimas Abrianno, Sofyan Sauri, Gumelar Pinanggih, Akhmad Febrianto.
11. Dan kepada kelompok belajar yang tiada hentinya untuk menuntut ilmu bersama khususnya M. Hardiansyah, Indria Aprilyana, Octavirzi Afni, Rizkiana Kunastiti, Yeyen Maryani, Titin Mardiyah, Yulia Sahadah, Fakhri Malik.
12. Kepada Riza Agustin, dan Nurul Rizal atas waktunya untuk konsultasi.
13. Kepada Risna Damayanti, Deli Cahya Illahi yang telah menampung keluh kesah penulis selama penyusunan Tugas Akhir.
14. Kepada Hanifa Dian Adila Vega, Bella Sagita, Mia Amalia, dan Alissa Qudroni yang selalu memberikan semangat dan motivasinya untuk menyusun Tugas Akhir.
15. Kepada teman-teman 7 tahun yang memberikan motivasi dalam penyusunan Tugas Akhir ini khususnya kepada Indri Fitriana Alfia Nisa, Muthia Ratuzzahrah, Tiara Alincia Fitri, Khadigah Padma Bilgis, Siti Aisyah, dan Andi Rachman.

Semoga Allah SWT membalas segala kebaikan dan ketulusan serta memberikan kemudahan dalam segala urusan kepada pihak-pihak yang telah membantu.

Jakarta, 13 Agustus 2018

Penulis

DAFTAR ISI

Halaman

HALAMAN JUDUL
LEMBAR PERSETUJUAN DOSEN PEMBIMBING
LEMBAR BIMBINGAN DOSEN PEMBIMBING
LEMBAR PERNYATAAN KEASLIAN
ABSTRAK iv
KATA PENGANTAR v
DAFTAR ISI vii
DAFTAR GAMBAR xii
DAFTAR TABEL xv

BAB I PENDAHULUAN 1

1.1 Latar Belakang 1

1.2 Pokok Permasalahan 3

1.3 Tujuan Penelitian 4

1.4 Batasan Masalah 4

1.5 Manfaat Tugas Akhir 4

1.6 Sistematika Penulisan 5

BAB II LANDASAN TEORI 7

2.1 Pengertian Sistem 7

2.1.1 Karakteristik Sistem 8

2.1.2 Klasifikasi Sistem 10

2.2 Konsep Dasar Informasi 10

2.2.1 Kualitas Informasi 11

2.2.2 Siklus Informasi 13

2.3 Konsep Dasar Sistem Informasi 15

2.3.1 Komponen Sistem Informasi 16

2.4 Konsep Dasar Penyimpanan.....17

2.4.1 Metode Penyimpanan.....17

2.4.2 Metode Penyimpanan FIFO (*First In First Out*).....18

2.5 Pengertian *Warehouse* atau Gudang.....20

2.5.1 Alasan Diperlukannya Gudang 21

2.6 Pengertian Barang Jadi 21

2.7 Telaah Jurnal.....22

2.8 *System Development Life Cycle* (SDLC).....23

2.9 Metode Pengembangan Sistem *Waterfall*.....28

2.10 *Flowchart*.....31

2.11 *Unifed Modeling Language* (UML) 32

2.11.1 *Use Case Diagram* 34

2.11.2 *Activity Diagram* 36
2.11.3 *Sequence Diagram* 38
2.11.4 *Class Diagram* 39
2.11.5 *Deployment Diagram* 41
2.12 Sistem Manajemen Basis Data 42
2.12.1 *Database* 43
2.12.2 *Entity Relationship Diagram* (ERD)..... 44
2.12.3 Kamus Data..... 45
2.13 MariaDB 46
2.13.1 Tipe Data..... 47
2.14 *Windows Navigation Diagram* (WND) 49
2.15 PHP..... 50
2.16 *Framework*..... 52
2.17 *CodeIgniter* 53
2.18 *Blackbox Testing*..... 56
2.18.1 Metode *Black Box Testing*..... 57
2.18.2 Perbedaan *White Box* dan *Black Box*..... 58
2.19 Analisis *Pieces*..... 58
BAB III METODOLOGI PENELITIAN 60

3.1 Metodologi Penelitian 60

3.2 Jenis dan Sumber Data 60

3.3 Metode Pengumpulan Data 61

3.4 Metode Pengembangan Sistem 62

3.5 Kerangka Penelitian 63

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA 67

4.1 Sekilas Perusahaan 67

4.2 Profil Umum Perusahaan 68

4.3 Visi dan Misi Perusahaan 68

4.4 Struktur Organisasi Perusahaan 69

4.4.1 Tugas dan Wewenang Jabatan di PT Teknikatama Karya Mandiri 70

4.4.2 Struktur Organisasi pada Divisi *Warehouse* 73

4.5 Barang Jadi pada PT Teknikatama Karya Mandiri 73

4.6 Penyimpanan 79

4.6.1 Siklus Penyimpanan Barang Jadi 80

4.6.2 Sistem Penyimpanan Barang Jadi yang Berjalan 80

4.6.3 Dokumen pada Penyimpanan Barang Jadi yang Berjalan82

4.6.4 Arus Dokumen yang Berjalan pada Penyimpanan Barang Jadi 86

4.7 Penggambaran Penyimpanan Barang Jadi dengan UML (*Unified Modeling Language)* 88

4.8 Analisis *Pieces* 90

BAB V ANALISIS DAN PEMBAHASAN 93

5.1 Analisis Sistem Informasi Usulan 93

5.1.1 Analisis Kebutuhan Sistem 93

5.1.2 Perancangan *Flowmap* Sistem Usulan 95

5.1.3 *Use Case Diagram* Usulan 97

5.1.4 *Activity Diagram* 106

5.1.5 *Sequence Diagram* 116

5.1.6 *Class Diagram* 127

5.1.7 *Deployment Diagram* 127

5.1.8 *Entity Relationship Diagram* (ERD) 128

5.1.9 Kamus Data 129

5.1.10 *Windows Navigation Diagram* (WND) 135

5.1.11 Perancangan Sistem Antarmuka 136

5.1.12 Pengujian *Black Box Testing* 149

5.1.13 Implementasi Sistem 165

BAB VI KESIMPULAN DAN SARAN 167

6.1 Kesimpulan 167

6.2 Saran 167

DAFTAR PUSTAKA 167

LAMPIRAN A (TAMPILAN PROGRAM) L-1

LAMPIRAN B (KODE PROGRAM) L-9

LAMPIRAN C (WAWANCARA) L-17

LAMPIRAN D (OBSERVASI) L-19

DAFTAR GAMBAR

Halaman

Gambar II.1 Pilar Kualitas Informasi 13
Gambar II.2 Siklus Informasi dalam Anggraeni dan Irviana 14
Gambar II.3 Siklus Informasi dalam Hutahaean 15
Gambar II.4 Metode *Waterfall* Berdasarkan Dennis *et al* 29
Gambar II.5 Metode *Waterfall* Berdasarkan Pressman dan Sommerville.....30
Gambar II.6 *Windows Navigation Diagram* 49
Gambar II.7 Alur Kerja *CodeIgniter*.....56
Gambar III.1 Kerangka Penelitian 65
Gambar IV.1 Struktur Organisasi Perusahaan 70
Gambar IV.2 Struktur Organisasi Departemen *Warehouse*.....73
Gambar IV.3 Hasil Proses Produksi 1 *Gusset*.....74
Gambar IV.4 Hasil Proses Produksi 2 *Gusset*.....74
Gambar IV.5 Hasil Proses Produksi 3 *Gusset*.....75
Gambar IV.6 Barang Jadi *Gusset*.....75
Gambar IV.7 Barang Jadi *Reinforcement* 76
Gambar IV.8 Barang Jadi *Pin Shaft*.....77
Gambar IV.9 Hasil Proses Produksi 1 *End Housing BZ 080*.....78
Gambar IV.10 Hasil Proses Produksi 2 *End Housing BZ 080*.....78
Gambar IV.11 Barang Jadi *End Housing*.....79
Gambar IV.12 Tata Letak Barang di Gudang PT Teknikatama Karya Mandiri....81
Gambar IV.13 Lokasi Gudang PT Teknikatama Karya Mandiri.....82
Gambar IV.14 Daftar Stok 83
Gambar IV.15 Kartu Tanda Kotak *Reinforcement* 84
Gambar IV.16 Kartu Tanda Kotak *End Housing*.....84
Gambar IV.17 Kartu Tanda Kotak *Pin Shaft*.....84
Gambar IV.18 Kartu Tanda Kotak *Gusset*.....85

Gambar IV.19 Surat Jalan 86
Gambar IV.20 *Flowmap* Penyimpanan Barang Jadi yang Berjalan 87
Gambar IV.21 *Use Case Diagram* Penyimpanan Barang Jadi yang Berjalan.....90
Gambar V.1 *Flowmap* Sistem Informasi Penyimpanan Barang Jadi Usulan 96
Gambar V.2 *Use Case Diagram* Sistem Informasi Penyimpanan Barang Jadi Usulan 97
Gambar V.3 *Activity Diagram* Melakukan *Login* 106
Gambar V.4 *Activity Diagram* Mengelola Data *Master* Pelanggan.....107
Gambar V.5 *Activity Diagram* Mengelola Data *Master* Barang.....108
Gambar V.6 *Activity Diagram* Mengelola Data *Master* Pengguna 109
Gambar V.7 *Activity Diagram* Memasukkan Data Transaksi Barang Masuk 110
Gambar V.8 *Activity Diagram* Memasukkan Data Transaksi Barang Keluar dan Surat Jalan 111
Gambar V.9 *Activity Diagram* Melihat Data Stok Barang 112
Gambar V.10 *Activity Diagram* Mencetak Laporan Data Stok 112
Gambar V.11 *Activity Diagram* Melihat Data Transaksi Masuk.....113
Gambar V.12 *Activity Diagram* Melihat Data Transaksi Keluar.....113
Gambar V.13 *Activity Diagram* Mencetak Laporan Transaksi Keluar.....114
Gambar V.14 *Activity Diagram* Mencetak Laporan Transaksi Masuk.....115
Gambar V.15 *Activity Diagram* Validasi Surat Jalan 115
Gambar V.16 *Sequence Diagram* Melakukan *Login* 116
Gambar V.17 *Sequence Diagram* Mengelola Data *Master* Barang.....118
Gambar V.18 *Sequence Diagram* Mengelola Data Master Pelanggan.....119
Gambar V.19 *Sequence Diagram* Mengelola Data Master Pengguna.....120
Gambar V.20 *Sequence Diagram* Memasukkan Data Transaksi Barang Masuk 121
Gambar V.21 *Sequence Diagram* Memasukkan Data Transaksi Barang Keluar dan Surat Jalan 122
Gambar V.22 *Sequence Diagram* Mencetak Transaksi Keluar 123
Gambar V.23 *Sequence Diagram* Mencetak Transaksi Masuk 124
Gambar V.24 *Sequence Diagram* Mencetak Laporan Data Stok 124
Gambar V.25 *Sequence Diagram* Melihat Data Stok Barang.....125

Gambar V.26 *Sequence Diagram* Melihat Data Transaksi Masuk 125
Gambar V.27 *Sequence Diagram* Validasi Surat Jalan 126
Gambar V.28 *Sequence Diagram* Melihat Transaksi Barang Keluar 126
Gambar V.29 *Class Diagram* Usulan 127
Gambar V.30 *Deployment Diagram* 128
Gambar V.31 *Entity Relationship Diagram* Usulan 129
Gambar V.32 *Windows Navigation Diagram* Usulan.....136
Gambar V.33 Halaman *Login* 137
Gambar V.34 Halaman Menu Utama.....137
Gambar V.35 Halaman Sub Menu Data *Master* Pelanggan 138
Gambar V.36 *Form* Tambah Pelanggan 138
Gambar V.37 *Form* Ubah Data Pelanggan 139
Gambar V.38 Halaman Data Master Barang Jadi 139
Gambar V.39 *Form* Tambah Data Barang Jadi 140
Gambar V.40 *Form* Ubah Data Barang Jadi 140
Gambar V.41 Halaman Data Master Pelanggan 141
Gambar V.42 *Form* Tambah Data Master Pengguna.....141
Gambar V.43 *Form* Ubah Data Master Pengguna 142
Gambar V.44 Halaman Data Transaksi Masuk.....142
Gambar V.45 *Form* Tambah Detail Data Barang 143
Gambar V.46 *Form* Buat Transaksi Masuk 143
Gambar V.47 Halaman Data Transaksi Keluar.....144
Gambar V.48 *Form* Tambah Detail Barang Keluar.....145
Gambar V.49 *Form* Buat Transaksi Barang Keluar.....145
Gambar V.50 Halaman Detail Transaksi Barang Masuk.....146
Gambar V.51 Halaman Detail Transaksi Barang Keluar.....146
Gambar V.52 Halaman Data Surat Jalan 147
Gambar V.53 Halaman Stok Barang Jadi 147
Gambar V.54 Halaman Data Laporan Transaksi Masuk 148
Gambar V.55 Halaman Data Laporan Transaksi Keluar 148

DAFTAR TABEL

Halaman

Tabel II.1 Simbol-simbol *Flowchart*.....31
Tabel II.2 Ringkasan Diagram UML 33
Tabel II.3 Simbol-Simbol *Use-Case Diagram* 35
Tabel II.4 Simbol-Simbol *Activity Diagram*.....36
Tabel II.5 Simbol-Simbol *Sequence Diagram*.....38
Tabel II.6 Simbol-Simbol *Class Diagram* 40
Tabel II.7 *Deployment Diagram* 41
Tabel II.8 Contoh Tabel Barang.....43
Tabel II.9 Simbol-simbol *Entity Relationship Diagram*.....44
Tabel II.10 Notasi-notasi Kamus Data.....45
Tabel IV.1. Nama Pelanggan Barang Jadi PT Teknikatama Karya Mandiri 81
Tabel IV.2. Definisi Aktor Penyimpanan Barang Jadi 88
Tabel IV.3 Definisi *Use Case* Penyimpanan Barang Jadi.....89
Tabel IV.4 Analisis *Pieces* Penyimpanan Barang Jadi 91
Tabel V.1 Analisis Kebutuhan Sistem 93
Tabel V.2 Definisi Aktor Pada *Use Case* Diagram.....98
Tabel V.3 Definisi *Use Case* Mengelola Data *Master*.....98
Tabel V.4 Definisi *Use Case* Memasukkan Data Transaksi Barang Masuk 99
Tabel V.5 Definisi *Use Case* Melakukan *Login* 100
Tabel V.6 Definisi *Use Case* Memasukkan Data Transaksi Barang Keluar dan Surat Jalan 101
Tabel V.7 Definisi *Use Case* Mencetak Laporan Data Stok.....102
Tabel V.8 Definisi *Use Case* Mencetak Laporan Transaksi Masuk 103
Tabel V.9 Definisi *Use Case* Mencetak Laporan Transaksi Keluar 103
Tabel V.10 Definisi *Use Case* Validasi Surat Jalan.....104
Tabel V.11 Definisi *Use Case* Melihat Data Transaksi Masuk 104

Tabel V.12 Definisi *Use Case* Melihat Data Transaksi Keluar 104
Tabel V.13 Definisi *Use Case* Melihat Data Stok Barang.....105
Tabel V.14 Tabel Data Pelanggan 130
Tabel V.15 Tabel Data Barang.....130
Tabel V.16 Tabel Data Pengguna 131
Tabel V.17 Tabel Data Transaksi Barang Masuk 131
Tabel V.18 Tabel Data Transaksi Barang Masuk Detail 132
Tabel V.19 Tabel Data Transaksi Barang Keluar 133
Tabel V.20 Tabel Data Transaksi Barang Keluar Detail 133
Tabel V.21 Tabel Data *Box* Keluar Detail 134
Tabel V.22 Tabel Data Box Masuk Detail.....135
Tabel V.23 *Black Box Testing*.....149

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dalam era globalisasi sekarang ini, bukan hal asing lagi untuk sebuah perusahaan menerapkan sistem informasi untuk perusahaannya. Salah satu manfaat adanya sistem informasi pada perusahaan adalah untuk mendukung pengambilan keputusan. Selain membantu dalam pengambilan keputusan, sistem informasi juga berguna untuk membuat suatu pekerjaan menjadi lebih mudah. Selain itu, sistem informasi juga dapat membantu penggunanya untuk mengumpulkan data dalam jumlah besar. Salah satu contoh penerapan sistem informasi dalam perusahaan adalah membuat suatu *website* yang dapat membantu perusahaan berinteraksi dengan antar departemen bahkan antara perusahaan dengan pelanggannya.

Gudang digunakan oleh suatu perusahaan sebagai tempat penyimpanan barang baik bahan baku, barang setengah jadi, ataupun barang jadi. Suatu sistem informasi dapat berperan baik untuk Operator *Warehouse* dalam mengelola penyimpanan barang yang ada dalam gudang. Salah satu bentuk dari suatu sistem informasi untuk penyimpanan barang yang ada dalam gudang, adalah sistem informasi berbasis *website.*

Sistem informasi berbasis *website* saat ini sudah banyak digunakan oleh individu, organisasi, maupun perusahaan untuk penyebaran informasi. Dalam Departemen *Warehouse* yang mengelola penyimpanan barang dalam gudang, suatu sistem informasi akan digunakan untuk membantu pekerjaan yang dilakukan oleh Operator *Warehouse* seperti mencatat arus keluar/masuknya barang jadi, dan membuat suatu laporan penyimpanan barang jadi.

Setiap perusahaan semakin mengedepankan kualitas dari produk yang mereka hasilkan untuk memuaskan para konsumennya. Kualitas produk yang dihasilkan oleh suatu perusahaan tak lepas dari bagaimana perusahaan tersebut mengolah bahan baku menjadi barang jadi yang telah diproduksi. Untuk menghasilkan suatu produk atau barang jadi dengan kualitas tinggi, proses-proses

yang ada mulai dari proses penerimaan bahan baku, proses produksi, proses pengemasan, penyimpanan hingga pengiriman harus dijaga dengan baik. Demi menjaga suatu produk mulai dari bahan baku hingga barang jadi, maka peranan gudang tidak dapat dipisahkan dari aktivitas tersebut.

Barang jadi adalah produk yang sudah selesai diproses dan menunggu waktu pengiriman. Proses penyimpanan barang jadi yang baik akan menghasilkan produk dengan kualitas yang baik pula. Dalam mengelola penyimpanan barang jadi dalam gudang, peranan Operator *Warehouse* dalam proses penyimpanan barang jadi adalah menerima dan mencatat berapa barang jadi yang masuk ke gudang, dan mengamati apakah barang yang disimpan dalam gudang dapat terjaga dengan baik atau tidak.

PT Teknikatama Karya Mandiri yang terletak di Jalan Raya Bahkilong nomor 88 Sukadami, Cikarang Selatan adalah salah satu perusahaan menengah yang berjalan pada bidang industri otomotif. Bahan baku yang didapat, dikirim langsung oleh pelanggan (*customer*) sehingga PT Teknikatama Karya Mandiri hanya melakukan proses *finishing* seperti menghaluskan permukaan yang kasar, hingga pemberian lubang pada suatu produk. Produk yang dihasilkan seperti *Pin Shaft, Reinforcement, End Housing* dan *Gusset.*

Sering kali ditemukan kendala atau permasalahan yang ada dalam proses penyimpanan barang jadi dalam gudang. Permasalahan tersebut misalnya seperti tidak rutinnya Operator *Warehouse* mengisi daftar stok. Hal ini dapat menghambat Operator *Warehouse* untuk membuat sebuah laporan bulanan. Ataupun proses pengambilan barang jadi yang kurang tepat sehingga menyebabkan barang jadi terlalu lama disimpan dalam gudang. Selanjutnya, belum adanya pencatatan detail aktivitas keluar/masuknya barang jadi dari gudang. Hal ini menghambat Departemen *Marketing* menganalisis jumlah barang jadi yang keluar dari gudang dan Direktur untuk mengetahui aktivitas keluar/masuknya barang jadi dalam gudang. Laporan bulanan ini nantinya dibuat oleh Operator *Warehouse* lalu diberikan ke Departemen *Marketing* untuk dianalisis apakah bahan baku yang masuk dan barang jadi yang keluar ke pelanggan jumlahnya seimbang atau tidak. Jika ditemukan ketidakseimbangan,

maka akan dilakukan evaluasi kerja. Barang jadi yang telah diproduksi dikirim secara acak. Sehingga mengakibatkan barang jadi yang pertama masuk ke dalam gudang setelah waktu produksi terlihat kotor, atau tergores dikarenakan tertumpuk dalam gudang.

Untuk mengatasi masalah tersebut, alangkah baiknya jika sistem yang ada diperbaiki dengan menggunakan suatu sistem informasi untuk membantu Operator *Warehouse* dalam menyelesaikan pekerjaannya untuk menyimpan barang jadi. Adapun judul Tugas Akhir ini adalah "Rancang Bangun Sistem Informasi Penyimpanan Barang Jadi Pada Departemen *Warehouse* di PT Teknikatama Karya Mandiri Menggunakan *CodeIgniter* 3.1.4 dan MariaDB 10.1.31"

1.2 Pokok Permasalahan

Permasalahan yang terjadi pada Departemen *Warehouse* dalam penyimpanan barang jadi pada PT Teknikatama Karya Mandiri adalah sebagai berikut:

1. Sering ditemukannya daftar stok yang kosong karena tidak dicatat dengan rutin sehingga menghambat informasi aktivitas pencatatan barang jadi yang keluar/masuk ke dalam gudang.
2. Belum adanya laporan bulanan dikarenakan data aktivitas keluar/masuknya barang jadi yang diperlukan dalam daftar stok sering kosong sehingga kurangnya informasi mengenai aktivitas barang jadi dalam gudang.
3. Pengambilan barang jadi dilakukan secara acak sehingga barang jadi yang pertama kali masuk ke dalam gudang tidak terambil ketika waktu pengiriman tiba. Hal tersebut dikhawatirkan dapat membuat barang menjadi rusak karena terlalu lama disimpan di dalam gudang.

1.3 Tujuan Penelitian

Adapun tujuan penelitian Tugas Akhir ini adalah sebagai berikut:

1. Membuat suatu sistem informasi untuk pencatatan barang jadi sehingga aktivitas keluar/masuknya barang jadi dapat terkontrol dengan baik.
2. Membuat suatu sistem informasi dan basis data (*database*) untuk aktivitas keluar/masuknya barang jadi yang mana data tersebut dapat membantu Operator *Warehouse* untuk membuat sebuah laporan bulanan dengan cepat.
3. Menerapkan cara pengambilan barang jadi dengan teknik *First In First Out*. Sehingga Operator *Warehouse* dapat mengetahui barang jadi mana saja yang harus dikeluarkan dari dalam gudang. Barang jadi yang pertama masuk ke dalam gudang, adalah barang jadi yang keluar dari gudang untuk dikirim ke pelanggan. Hal ini untuk meminimalisir adanya barang jadi yang terlalu lama disimpan dalam gudang.

1.4 Batasan Masalah

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

1. Penelitian dilakukan selama satu bulan bertempat di PT Teknikatama Karya Mandiri J*alan Raya Bahkilong* No.88 Cikarang Selatan – Bekasi 17330.
2. *Jangka waktu untuk penelitian ini, mulai dari 31 Juli 2017 sampai dengan 28 Agustus 2017.*
3. Ruang lingkup yang dianalisis adalah menangani proses penyimpanan barang jadi yang mencakup pencatatan barang jadi yang masuk ke gudang dan keluar gudang hingga membuat surat jalan.

1.5 Manfaat Tugas Akhir

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

1. Bagi perusahaan

a. Hasil penelitian ini agar dapat diimplementasikan di perusahaan untuk membantu kinerja perusahaan dalam melakukan pemantauan data barang jadi yang tersedia.

2. Bagi mahasiswa
   a. Memberikan kemampuan dalam mengaplikasikan teori secara jelas terhadap masalah yang diamati.
   b. Memberikan pembelajaran untuk menerapkan ilmu yang diperoleh selama perkuliahan serta menambah wawasan dan pengetahuan penulis, khususnya dalam analisis dan perancangan sistem informasi.
3. Bagi pihak lain

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

1.6 Sistematika Penulisan

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

BAB I PENDAHULUAN

Bab ini memuat latar belakang, batasan masalah, tujuan penelitian, metodologi penelitian, dan sistematika penulisan laporan.

BAB II LANDASAN TEORI

Bab ini menjelaskan tentang berbagai teori yang dirangkum dari buku-buku ataupun berbagai macam referensi yang berkaitan dengan tema yang diambil. Maupun analisis dokumen jurnal yang membahas tentang penyimpanan barang jadi. Teori-teori yang dipaparkan pada laporan ini adalah seputar analisis sistem informasi penyimpanan barang jadi dan metodologi pengembangan sistem.

BAB III **METODOLOGI PENELITIAN**

Pada bab ini membahas tentang jenis dan sumber data, metode pengumpulan data, metode pengembangan sistem yang digunakan, serta kerangka penelitin untuk membantu dalam pemecahan masalah.

BAB IV **PENGUMPULAN DAN PENGOLAHAN DATA**

Pada bab ini akan membahas mengenai data-data yang telah diperoleh berdasarkan hasil pengamatan selama melaksanakan penelitian di PT Teknikatama Karya Mandiri.

BAB V **ANALISIS DAN PEMBAHASAN**

Bab ini berisi analisis dari pengolahan data dengan pemodelan sistem menggunakan UML (*Unified Modeling Language*) meliputi *usecase diagram, activity diagram, sequence diagram, class diagram,* dan *deployment diagram*. Pemodelan data menggunakan ERD dan Kamus Data. Serta membuat perancangan antarmuka. Untuk dokumentasi *testing* menggunakan *blackbox testing*.

BAB VI **KESIMPULAN DAN SARAN**

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

BAB II
LANDASAN TEORI

2.1 Pengertian Sistem

Berdasarkan Djahir dan Pratita (2014), sistem adalah kumpulan dari komponen-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 :

a. Komponen (*components)*

Komponen sistem adalah segala sesuatu yang menjadi bagian penyusunan sistem. Komponen sistem dapat berupa benda nyata ataupun abstrak. Komponen sistem disebut sebagai subsistem.

b. Batas (*boundary)*

Batas sistem diperlukan untuk membedakan satu sistem dengan sistem yang lain. Tanpa adanya batas sistem, sangat sulit untuk memberikan batasan tinjauan terhadap sistem.

c. Lingkungan (*environments)*

Lingkungan sistem adalah segala sesuatu yang berada diluar sistem yang dapat menguntungkan ataupun merugikan. Pengaruh yang menguntungkan akan dipertahankan untuk menjaga keberlangsungan sistem, sedangkan yang merugikan akan diupayakan mempunyai pengaruh seminimal mungkin atau ditiadakan.

d. Penghubung/Antarmuka (*interface)*

Penghubung/antarmuka adalah sarana memungkinkan setiap komponen sistem, yaitu segala sesuatu yang bertugas menjembatani hubungan antar komponen dalam suatu sistem. Atau dapat dikatakan, suatu sarana tiap komponen untuk saling berinteraksi dan berkomunikasi.

e. Masukan (*input)*

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

f. Pengolahan (*processing)*

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

g. Keluaran (*output)*

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

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

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

i. Kendali (*control)*

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

j. Umpan Balik (*feed back)*

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

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

2.1.2 Klasifikasi Sistem

Berdasarkan Rusdiana dan Irfan (2014), 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):

a. Jogiyanto (2008), informasi adalah data yang diolah menjadi bentuk yang berguna bagi para pemakainya.

b. 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.

c. Krismiaji dalam Zabaer (2005), informasi adalah data yang telah diorganisasikan, dan telah memiliki kegunaan dan manfaat.

d. 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: 151) dalam Tyoso (2016), informasi yang berkualitas harus memenuhi beberapa persyaratan sebagai berikut:

a. Ketersediaan (*Availability*)

Informasi harus dapat diakses oleh orang yang membutuhkannya, maka dari itu informasi harus tersedia setiap saat pada “gudang data” yang terorganisasi rapi.

b. 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.

c. Akurat (*Accuracy*)

Informasi harus bersih dari kesalahan dan kekeliruan. Artinya, informasi harus jelas dan tepat dalam mencerminkan makna yang terkandung dari data.

d. Relevan (*Relevant*)

Berkaitan dengan pengoperasian suatu organisasi, informasi yang dibutuhkan ialah informasi yang benar-benar relevan dengan permasalahan, misi dan tujuan organisasi yang bersangkutan.

e. Bermanfaat (*Benefits*)

Informasi sebaiknya dapat disajikan dalam bentuk-bentuk yang mudah dilihat dan dipelajari sehingga kepemanfaatannya terlihat jelas.

f. 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.

g. Keterandalan (*Reliability*)

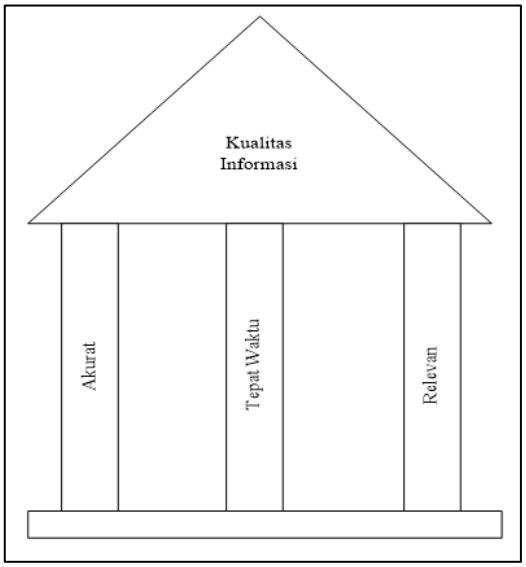
Informasi harus diperoleh dari sumber daya yang jelas dan dapat dipertanggungjawabkan kebenarannya. Demikian juga dengan pengelola atau pemberi informasi juga merupakan pihak-pihak yang dapat dipercaya.

h. 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:

a. Relevansi
b. Ketepatan waktu
c. Keakurasian

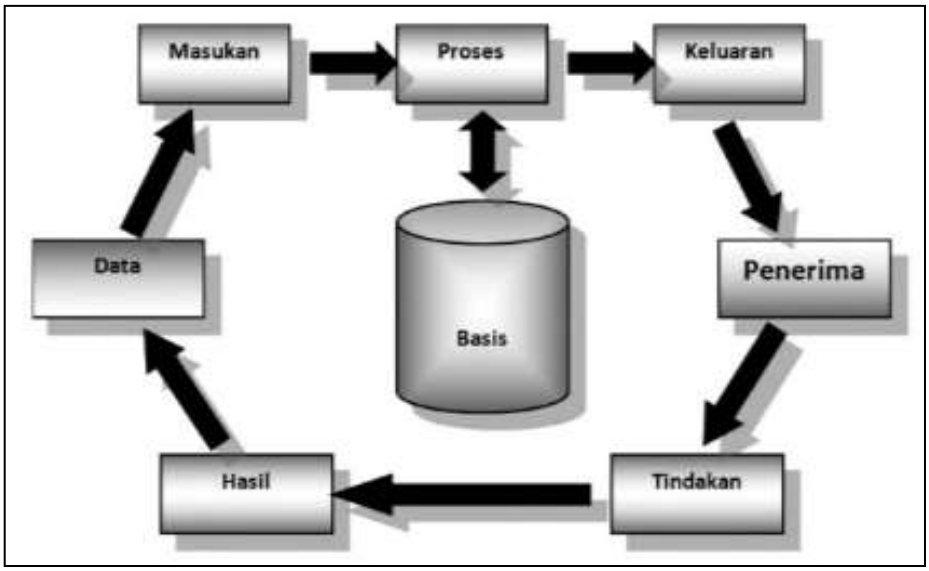


Gambar II.1 Pilar Kualitas Informasi

(Sumber: Anggraeni dan Irviani, 2017)

2.2.2 Siklus Informasi

Berdasarkan Burch dan Grudnitski (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

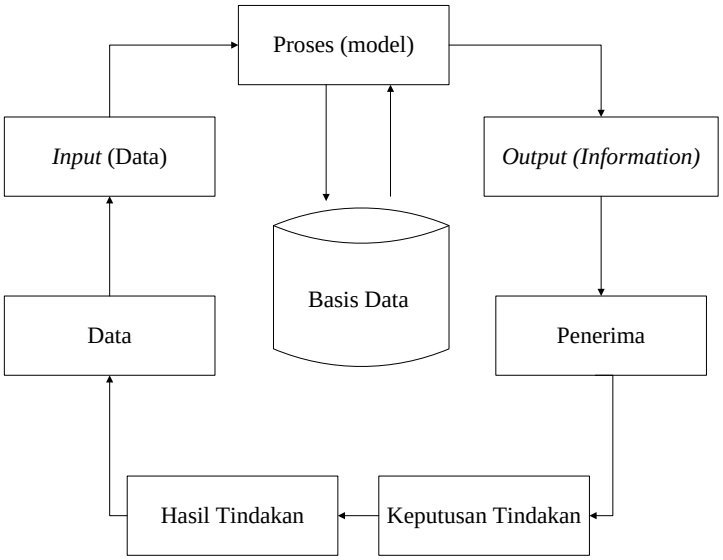
(Sumber: Anggareni dan Irviana, 2017)

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):

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

2.3.1 Komponen Sistem Informasi

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

1. Komponen *input*, adalah data yang masuk ke dalam sistem informasi.
2. Komponen model, adalah kombinasi prosedur, logika dan model matematika yang memproses data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.
3. Komponen *output*, adalah hasil informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.
4. Komponen teknologi, adalah alat dalam sistem informasi, teknologi digunakan dalam menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan *output* dan memantau pengendalian sistem.
5. Komponen basis data, adalah kumpulan data yang saling terhubung yang tersimpan di dalam komputer dengan menggunakan *software database.*
6. Komponen kontrol, adalah komponen yang mengendalikan gangguan terhadap sistem informasi.

2.4 Konsep Dasar Penyimpanan

Berdasarkan Sobir dan Firmansyah (2014), penyimpanan merupakan upaya penundaan pengaturan produk sehingga ketika ada pengiriman bisa dilakukan dengan cepat. Sering kali penyimpanan juga terkait dengan upaya menjaga kualitas produk sebelum sampai di tangan konsumen.

Berdasarkan Sattar (2017), penyimpanan adalah menyimpan barang produksi perusahaan atau barang yang akan dijual untuk sementara waktu sebelum dipasarkan.

Berdasarkan Sriyanto (2008), kegiatan penyimpanan merupakan proses pengaturan dan penyimpanan data input yang ditempatkan pada posisi-posisi tertentu agar data tersebut sewaktu-waktu dapat diproses tanpa mengalami kesulitan.

Sementara berdasarkan halaman KBBI (Kamus Besar Bahasa Indonesia) (2017), penyimpanan diambil dari kata ‘simpan’ dimana kata simpan berarti menaruh barang di tempat yang aman supaya tidak rusak, hilang, dan lain sebagainya. Dan untuk penyimpanan sendiri adalah tempat menyimpan (mengumpulkan) atau proses, cara, perbuatan yang berkaitan dengan aktivitas menyimpan.

2.4.1 Metode Penyimpanan

Berdasarkan Warman (2012) dalam penyimpanan barang di gudang agar nantinya barang yang disimpan tersebut tidak mengalami kerusakan dan memudahkan dalam penanganan sangatlah dibutuhkan suatu metode, cara maupun prosedur tertentu. Setidaknya dalam penyimpanan barang di gudang, pengguna mengenal adanya dua sistem :

1. LIFO (*Last in First Out*)

LIFO adalah suatu sistem atau cara penyimpanan barang dalam gudang yaitu barang yang datang terakhir digunakan terlebih dahulu. Sistem ini biasanya digunakan untuk barang-barang yang dapat bertahan lama atau barang yang jika disimpan lebih lama kualitasnya akan lebih baik.

2. FIFO (*First in First Out*)

FIFO adalah suatu sistem penyimpanan barang yaitu barang yang masuk terlebih dahulu juga dikeluarkan terlebih dahulu. Sehingga keluar masuknya barang secara berurutan atau sesuai kronologis. Sistem ini biasanya digunakan untuk barang barang yang kurang bisa tahan lama.

2.4.2 Metode Penyimpanan FIFO (*First In First Out*)

Berdasarkan Wikipedia Indonesia (2018), FIFO merupakan singkatan dari *First in First out* atau dalam bahasa Indonesia, pertama masuk pertama keluar yang berarti bahwa persediaan yang pertama kali masuk itulah yang pertama kali dicatat sebagai barang yang dijual.

Berdasarkan Jurnal Id (2018), FIFO (*First in First out*) merupakan metode perhitungan biaya *inventory* dengan cara menjual produk atau stok yang terlebih dahulu masuk menjadi stok yang pertama keluar. FIFO banyak digunakan di industri yang menjual produk makanan dan kesehatan karena cenderung memiliki umur yang relatif pendek sehingga produk dengan tanggal kadaluwarsa terdekat yang akan dijual terlebih dahulu.

Berdasarkan Poros Ilmu (2016), sistem FIFO atau *First in First out* adalah suatu sistem penyimpanan barang yang dilakukan dengan sistem barang yang masuk terlebih dahulu, yang juga dikeluarkan terlebih dahulu. Sistem FIFO umumnya digunakan untuk barang-barang yang kurang tahan lama atau bila disimpan dalam waktu lama akan rusak atau berkurang kualitasnya. Contohnya seperti gula, beras, dan sejenisnya. Sebaiknya metode FIFO digunakan apabila barang tersebut kurang mampu bertahan lama dan dikhawatirkan jika barang yang datang disimpan terlalu lama dapat merusak kualitas barang itu sendiri. Kelebihan FIFO itu sendiri adalah sebagai berikut:

1. Barang Lebih Terjaga Kualitasnya

   Dengan FIFO, diharapkan barang yang pertama kali masuklah yang juga pertama kali keluar. Artinya, barang-barang tidak akan terlalu lama tersimpan dalam gudang. Jadi, barang dengan masa kadaluarsa yang paling

awal keluar. Dengan begitu, kualitas barang dapat lebih terjamin serta mengantisipasi terjadinya kerusakan barang secara masal.

2. Pengendalian Harga Lebih Terjamin

   Tidak selamanya harga dapat selalu sama. Ada kalanya harga naik, dan ada pula kalanya harga turun. Dengan sistem FIFO, diharapkan barang yang pertama kali masuk dengan harga tertentu dapat terjual sama harganya pada saat dikeluarkan nanti. Kekhawatiran bahwa harga barang akan anjlok pun dapat dihindari.

3. Pencatatan Lebih Sistematis

   Membuat petugas pencatatan barang masuk dan keluar jadi lebih mudah dalam melakukan control. Hal ini dimungkinkan karena keluarnya barang dilakukan secara berurutan atau sesuai kronologis. Jadi, petugas pencatatan biasanya tidak perlu melakukan pengecekan terhadap semua barang. Petugas biasanya hanya perlu mengecek jumlah barang yang keluar pada saat itu, apakah jumlahnya sudah sesuai dengan jumlah barang pada saat barang tersebut masuk.

Selain memiliki kelebihan, sistem FIFO memiliki kelemahan. Yaitu, sistem FIFO umumnya kurang efektif apabila pihak-pihak di bagian pergudangan tidak mampu menata letak barang-barang secara berurutan sesuai dengan tanggal atau waktu barang tersebut masuk. Sebab, penataan yang kurang tepat akan menyulitkan proses masuk dan keluarnya barang dari gudang. Jika demikian, proses atau waktu yang diperlukan dalam penataan, baik saat masuk atau mengeluarkan barang bisa lebih lama.

Selain itu, kemungkinan bahwa konsumen atau pihak pemakai barang merasa kurang puas dengan kualitas barang yang diterima. Sebab, barang yang diterima tersebut merupakan barang lama sehingga konsumen mungkin beranggapan bahwa barang tersebut kualitasnya kurang baik. Akan tetapi, kelemahan tersebut tentu dapat dihindari. Caranya, pihak pengelola gudang harus mampu mengatur pemindahan barang-barang secara efisien dan efektif sesuai dengan sistem ini. Bila menggunakan sistem FIFO, sebaiknya penataan barang di gudang tidak ditumpuk melainkan dijajar sesuai dengan waktu barang tersebut

masuk. Idealnya, gudang yang menggunakan sistem FIFO memiliki dua pintu. Pintu pertama adalah pintu masuk barang, dan pintu kedua adalah pintu keluarnya barang. penataannya, barang yang masuk dapat diletakkan dekat dengan pintu keluar barang dan begitu seterusnya. Penataan ini dimaksudkan agar barang yang pertama kali masuk dapat dengan mudah keluar karena dekat dengan pintu keluar. Demikian pula dengan barang yang baru saja masuk, dapat dengan mudah masuk tidak terhalangi oleh barang-barang yang sebelumnya sudah masuk.

2.5 Pengertian *Warehouse* atau Gudang

Berdasarkan Warman (2012), *Warehouse* atau gudang (kata benda) adalah bangunan yang dipergunakan untuk menyimpan barang dagangan. Penggudangan (kata kerja) ialah kegiatan menyimpan dalam gudang.

Sementara berdasarkan Wikipedia Indonesia (2017), Gudang adalah sebuah ruangan yang digunakan untuk menyimpan berbagai macam barang. Karena untuk menyimpan berbagai macam barang, biasanya gudang berpotensi untuk menyimpan debu. Karena itu, peletakan gudang perlu diperhatikan agar tidak menganggu aktivitas lain dalam bangunan tersebut. Gudang sebaiknya terletak di lokasi yang tidak lembap agar barang-barang yang disimpan tidak mudah rusak. Agar tidak menghabiskan lahan, gudang bisa diletakkan di bawah tangga atau di loteng. Gudang mencakup banyak jenis barang yang disimpan.

Menurut Sriyanto (2016) menerangkan bahwa gudang adalah suatu tempat atau bangunan yang digunakan untuk menimbun, menyimpan barang baik berupa 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, barang 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 Telaah Jurnal

Berdasarkan telaah jurnal yang dilakukan terhadap 2 jurnal yaitu Prasetyo & Ambarwati (2017) dengan judul "Perancangan Sistem Informasi Material Produk Kemasan Menggunakan Metode FIFO Pada PT. CRS", Manohar & Appaiah (2017) dengan judul "*Stabilization of FIFO system and Inventory Management*". Kedua jurnal tersebut membahas tentang penyimpanan barang dengan metode FIFO.

Ruang lingkup yang dibahas oleh Prasetyo & Ambarwati (2017), sudah menggunakan metode FIFO dalam penyimpanan PT. CRS yang mana *material* yang pertama kali dibeli atau masuk ke dalam gudang adalah yang pertama kali keluar untuk diproduksi. Tujuannya pun, agar setiap produk tidak tertimbun terlalu lama dan menghindari waktu kadaluarsa. Pada jurnal tersebut, untuk membangun sebuah sistem digunakan metodologi penelitian dengan menggunakan metode *waterfall*. Yang mana tahapannya meliputi:

1. Pengumpulan data
   a. Data primer.
   b. Data sekunder.
2. Analisis
   Menganalisis sistem yang berjalan dan permasalahan.
3. Perancangan
   Merancang sistem dengan UML.
4. Dokumentasi
   Membuat sebuah laporan.

Sedangkan ruang lingkup yang dibahas oleh Manohar & Appaiah (2017), gudang yang ada belum menggunakan metode FIFO. Menurut yang ada dalam jurnal tersebut, metode FIFO atau biasa disebut dengan *First In First Out* adalah suatu metode penyimpanan yang mana barang yang datang pertama kali, adalah barang yang pertama keluar. Metode FIFO dapat membantu suatu organisasi dalam menyimpan barang dengan cara memisahkan barang lama dan barang yang sudah tidak terpakai. Permasalahan yang ada dalam observasi jurnal tersebut adalah petugas yang akan menyimpan barang ke dalam area *work in process*

menyimpan barang dimanapun selama ada ruang yang kosong. Hal tersebut mengakibatkan jalannya pengambilan barang terhambat. Seringkali petugas mengambil barang yang terakhir diproduksi untuk dikeluarkan dari gudang. Hal ini mengakibatkan barang lama yang ada dalam gudang mengalami waktu kadaluarsa.

Perbedaan antara jurnal tersebut adalah pada ruang lingkup jurnal. Jurnal yang dibahas oleh Prasetyo & Ambarwati (2017) adalah membuat rancangan sistem informasi material produk berbasis *website* dengan metode FIFO. Adapun perancangan tersebut menggunakan diagram *use case diagram*, *sequence diagram*, dan *class diagram* serta metode pengembangan sistem menggunakan model *waterfall*. Sementara jurnal yang dibahas oleh Manohar & Appaiah (2017) adalah membahas tentang teori dan penyelesaian permasalahan penyimpanan barang yang ada di gudang dengan menggunakan metode FIFO.

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

Perbedaannya pada jenis barang yang disimpan. Pada Tugas Akhir ini membahas penyimpanan barang jadi sementara untuk jurnal Prasetyo & Ambarwati (2017) mengenai penyimpanan bahan baku atau *material product.* Sedangkan pada jurnal Manohar & Appaiah (2017) membahas penyimpanan barang *work in process* atau barang setengah jadi.

2.8 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:

a. 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.
b. 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:

a. *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.

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.

b. *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.

c. *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.

d. *Implementation* (Implementasi)

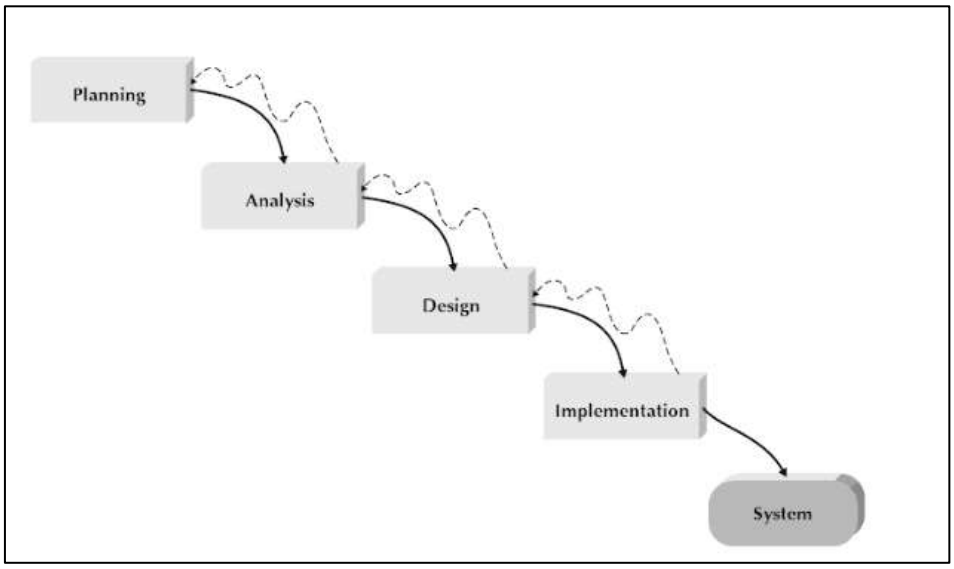
Tahap akhir dalam SDLC adalah tahap implementasi, di mana sistem benar-benar dibangun (atau dibeli, dalam kasus paket *design software*). Implementasi adalah fase yang paling mendapat perhatian, karena tahapan ini merupakan bagian terlama dan paling mahal dari proses pengembangan. Fase implementasi ini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

2.9 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

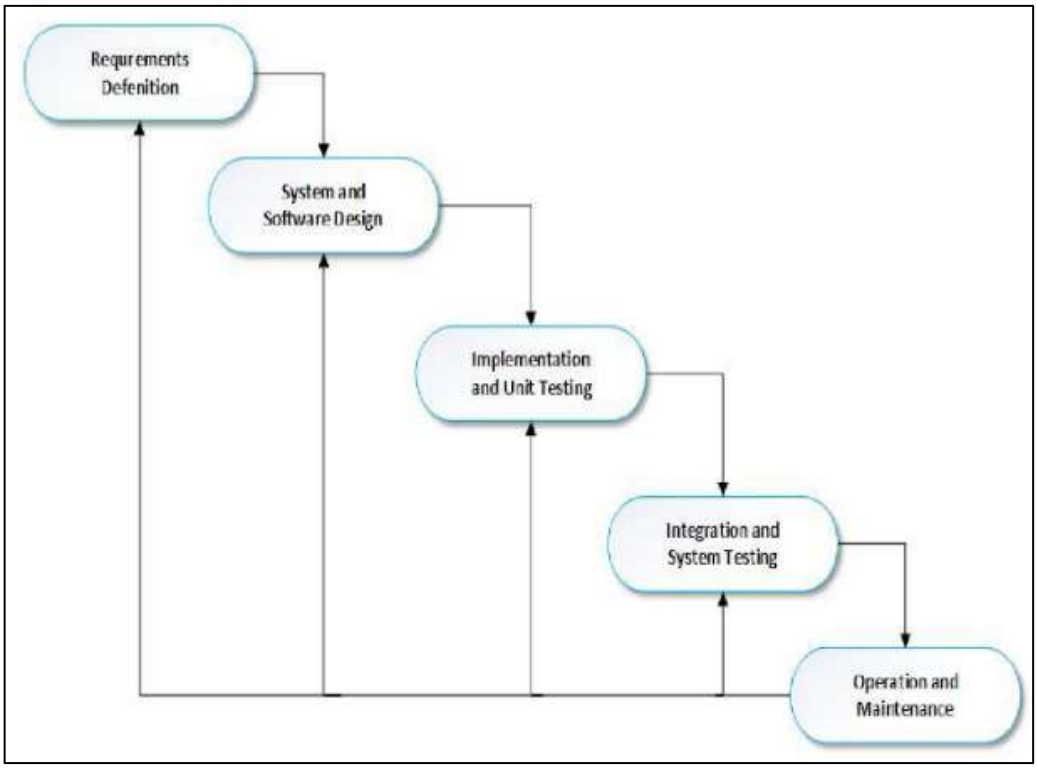
proyek atau *project sponsor* untuk disetujui sehingga dapat berlanjut ke fase selanjutnya. Suatu fase atau tahapan akan berlanjut jika sponsor proyek telah menyetujui tahapan tersebut. Model ini disebut dengan *waterfall development*, karena tahap demi tahapan bergerak maju dengan cara yang sama seperi 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.4.



Gambar II.4 Metode *Waterfall* Berdasarkan Dennis *et al*

(Sumber: Dennis *et al*, 2015)

Berdasarkan Muharto dan Ambarita (2016), model *waterfall* dikenal juga dengan "*Linier Sequential Model*" yang sering juga disebut dengan "*classic life cycle*" atau model *waterfall*. Metode ini muncul pertama kali sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model/metode yang paling banyak dipakai di dalam *Software Engineering* (SE). Metode ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan *system* lalu menuju ke tahap analisis, desain, *coding*, *testing/verification*, dan *maintance*. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Sebagai contoh tahap desain harus menunggu selesainya tahap sebelumnya yaitu tahap *requirement*. Menurut Pressman dan Sommerville (2010) pada Muharto dan Ambarita (2016), metode ini terdiri dari beberapa langkah, seperti ditunjukkan pada Gambar II.5.



Gambar II.5 Metode *Waterfall* Berdasarkan Pressman dan Sommerville

(Sumber: Pressman dan Sommerville, 2010)

Adapun tahap yang ada pada gambar dijelaskan sebagai berikut:

1) Analisis dan definisi kebutuhan. Layanan, batasan, dan tujuan sistem ditentukan melalui konsultasi dengan *user* atau pemakai.
2) Perancangan sistem dan perangkat lunak. Proses perancangan sistem membagi persyaratan dalam sistem perangkat keras atau perangkat lunak. Kegiatan ini menentukan arsitektur sistem secara keseluruhan. Perancangan melibatkan identifikasi dan deskripsi sistem perangkat lunak yang mendasar.
3) Implementasi dan pengujian unit. Pada tahap ini, perancangan perangkat lunak direalisasikan dengan program atau unit program.
4) Integrasi dan pengujian sistem. Unit program diuji sebagai sistem yang lengkap untuk menjamin bahwa kebutuhan sistem telah dipenuhi.
5) Operasi dan pemeliharaan, yaitu mengoperasikan program di lingkungannya dan melakukan pemeliharaan. Biasanya ini merupakan fase siklus hidup yang paling lama. Pemeliharaan mencakup koreksi dari

berbagai *error* yang ditemukan pada tahap-tahap sebelumnya, melakukan perbaikan atas implementasi unit sistem dan pengembangan sistem.

2.10 *Flowchart*

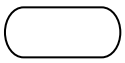
Berdasarkan Sitorus (2015), untuk menggambarkan sebuah proses agar mudah dipahami oleh orang lain maka dibutuhkan alat bantu yang berbentuk diagram alir (*flowchart*). *Flowchart* menggambarkan urutan logika dari suatu prosedur pemecahan masalah, sehingga *flowchart* merupakan langkah-langkah penyelesaian 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 (1) *system flowchart* dan (2) *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.

Tabel II.1 Simbol-simbol *Flowchart* (lanjutan)

Simbol	Nama	Keterangan
	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)

2.11 *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. UML ini mendefinisikan suatu set dari 15 teknik untuk pemodelan sistem, dapat dilihat pada Tabel II.2.

Tabel II.2 Ringkasan Diagram UML

Nama Diagram	**Fungsi**
Structure Diagrams	
Class	Menggambarkan hubungan antara pemodelan-pemodelan *class* di dalam sistem.
Object	Menggambarkan hubungan antara pemodelan-pemodelan objek di dalam sistem.
Package	Mengelompokkan elemen UML lainya menjadi satu untuk membentuk level konstruksi yang lebih tinggi.
Deployment	Menampilkan arsitektur fisik dari suatu sistem. Bisa juga digunakan untuk menunjukkan komponen *software* sedang dijadikan arsitektur fisik suatu sistem.
Component	Menggambarkan hubungan fisik diantara komponen-komponen *software*.
Composite	Menggambarkan struktur internal dari suatu class dan hubungan diantara bagian-bagian dari suatu *class*.
Profile	Digunakan untuk mengembangkan eksistensi UML sendiri.
Behavioral Diagrams	
Activity	Menggambarkan proses bisnis masing-masing *class*. Aliran aktivitas dalam *use case*, atau *detail* desain dari suatu metode.

Tabel II.2 Ringkasan Diagram UML (lanjutan)

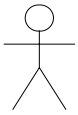
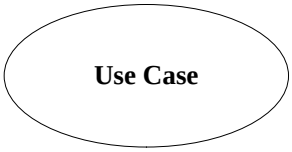
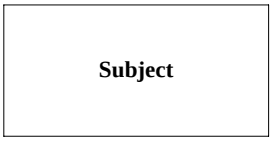
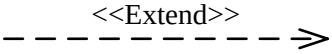
Nama Diagram	Fungsi
Behavioral Diagrams	
Sequence	Memodelkan kebiasaan dari objek-objek dalam suatu *use case*. Fokus dalam aktivitas berdasarkan urutan waktu.
Communication	Memodelkan kebiasaan dari objek-objek dalam suatu *use case*. Fokus dalam komunikasi antara satu set dari kolaborasi objek dari suatu aktivitas.
Interaction Overview	Menggambarkan suatu ikhtisar alur dari kontrol suatu proses.
Timing	Menggambarkan interaksi yang terjadi diantara suatu set dari objek-objek dan perubahan keadaan selama perjalanan waktu.
Behavioral State Machine	Menguji kebiasaan dari suatu *class.*
Protocol State Machine	Menggambarkan ketergantungan diantara perbedaan *interface* dari suatu *class*.
Use Case	Menggambarkan kebutuhan bisnis untuk sistem dan untuk menggambarkan interaksi antara sistem dan lingkungannya.

(Sumber: Dennis *et al.* 2015)

2.11.1 *Use Case Diagram*

Diagram *use case* dibuat untuk menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Suatu landasan pola pikir yang ditekankan dalam diagram *use case* ini adalah “apa” yang dapat diperbuat oleh sistem, dan bukan “bagaimana” sistem melakukannya. Diagram *use case* menggambarkan kebutuhan sistem dari sudut pandang *user*, yaitu proses yang dilakukan oleh sistem dalam melayani *user* yang berinteraksi dengan sistem tersebut. Simbol-simbol *use case* dijelaskan pada Tabel II.

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

Simbol	**Fungsi**
 **Actor/Role** **<<Actor>>** **Actor/Role**	▪ Seseorang atau sistem yang mendapatkakn keuntungan dari sistem. ▪ Dilabelkan dengan peran. ▪ Bisa diasosiasikan dengan *actor* lainnya berdasarkan spesialisasi. ▪ Diletakan diluar batas sistem.
 **Use Case**	▪ 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.
 **Subject**	▪ Menyertakan nama subjek didalam maupun diatas. ▪ Merepresentasikan ruang lingkup dari sistem.
<<Extend>> 	▪ Merepresentasikan eksistensi dari use case untuk menyertakan perilaku optional. ▪ Memiliki anak panah yang digambarkan dari eksistensi *use case* ke *use case* dasar.
Generalisasi	▪ 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)

2.11.2 *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:

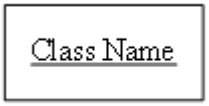
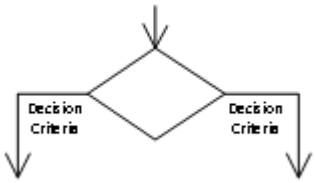
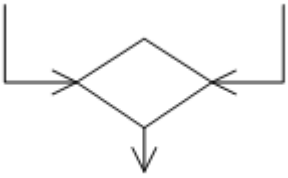
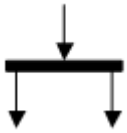
1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana 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.4.

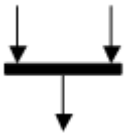
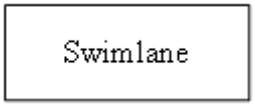
Tabel II.4 Simbol-Simbol *Activity Diagram*

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.

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

No	Simbol	Nama	Deskripsi
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.
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.

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

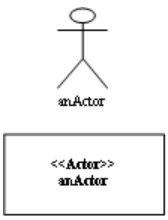
No	Simbol	Nama	Deskripsi
12.		*Join Node*	Menyatukan beberapa percabangan dari aliran aktivitas.
13.		*Swimlane*	Menggambarkan akhir aliran kontrol spesifik atau aliran objek.

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

2.11.3 *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.5.

Tabel II.5 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.

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

No	Simbol	Nama	Deskripsi
2.		*Lifeline*	Menyatakan kehidupan suatu objek.
3.	nama_objek : nama_kelas	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	*Message*	Pesan yang menggambarkan komunikasi yang terjadi antar objek.
6.	Return Value	*Message* (*return*)	Pesan yang dikirim untuk diri sendiri.
7.	[aGuardCondition] (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.	Context	*Frame*	Mengindikasikan konteks dari *sequence diagram.*

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

2.11.4 *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.6.

Tabel II.6 Simbol-Simbol *Class Diagram*

No.	Simbol	Deskripsi
1.	*Class* **Class 1** -attribute1 +operation1()	*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. Tanda kurung mengandung informasi atau parameter yang dibutuhkan untuk menampilkan 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*.

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

No.	Simbol	Deskripsi
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*.
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)

2.11.5 *Deployment Diagram*

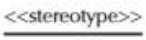
Berdasarkan Dennis *et al* (2015) menyebutkan bahwa *deployment diagram* digunakan untuk merepresentasikan hubungan antara komponen *hardware* yang digunakan dalam infrastruktur fisik dari suatu sistem informasi. *Deployment diagram* juga dapat digunakan untuk mewakili komponen *software* dan bagaimana mereka ditempatkan di atas arsitektur fisik atau infrastruktur dari suatu sistem informasi. Dalam hal ini, *deployment diagram* mewakili lingkungan untuk pelaksanaan *software*.

Tabel II.7 *Deployment Diagram*

No.	Simbol	Istilah dan Definisi
1.		*Node*: – Apakah sumber daya komputasi, misalnya, komputer klien, server, jaringan terpisah, atau perangkat jaringan individu. – Dilabeli dengan namanya.

		– Dapat berisi stereotip untuk secara khusus

Tabel II.7 *Deployment Diagram* (lanjutan)

No.	Simbol	Istilah dan Definisi
		memberi label jenis node yang diwakili, misalnya, perangkat, workstation klien, server aplikasi, perangkat seluler, dll.
2.		*Artifact:* – Adalah spesifikasi perangkat lunak atau basis data, misalnya, basis data atau tabel atau tampilan database, komponen atau lapisan perangkat lunak. – Dilabeli dengan namanya. – Dapat berisi stereotip untuk secara khusus melabeli jenis artefak, misalnya, file sumber, tabel basis data, file yang dapat dieksekusi, dll.
3.		*Node* dengan *Deployed artifact*: – Menampilkan artefak yang ditempatkan pada simpul fisik.
4.		*Communication path:* – Mewakili hubungan antara dua *node*. – Memungkinkan *node* untuk bertukar pesan. – Dapat berisi stereotip untuk secara khusus melabeli jenis jalur komunikasi yang diwakili, (misalnya, LAN, Internet, serial, paralel).

(Sumber: Dennis *et al*, 2015)

2.12 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.

2.12.1 *Database*

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

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

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

Berdasarkan Suryo (2012), *database* adalah kumpulan informasi yang disimpan dalam computer secara 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.8.

Tabel II.8 Contoh Tabel Barang

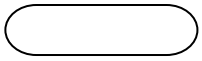
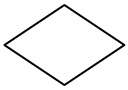
Field	***Field Name***	***Type***	***Width***	***Description***
1	Kode_Barang	Varchar	7	Kode Barang*
2	Nama_Barang	Varchar	35	Nama Barang

3	Satuan	Varchar	15	Satuan
4	Harga	Varchar	9	Harga

(Sumber: Suryo, 2012)

2.12.2 *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.9.

Tabel II.9 Simbol-simbol *Entity Relationship Diagram*

No.	Simbol	Deskripsi
1.	Entitas Nama_entitas	Entitas merupakan data inti yang akan disimpan
2.	Atribut Nama_at	*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.

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

No.	Simbol	Deskripsi
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)

2.12.3 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.10.

Tabel II.10 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]

2.13 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.

2.13.1 Tipe Data

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

a. Tipe Data Numerik

Berikut ini daftar data numerik yang digunakan:

1. **INT** – nilai integer dengan ukuran normal. *Signed* berkisar -2147483648 sampai 2147483647. *Unsigned* berkisar dari 0 sampai 4294967295. *Unsigned* dimaksudkan, hanya untuk kisaran non minus.
2. **TINYINT** – nilai integer yang sangat kecil. *Signed* berkisar dari -128 sampai 127. *Unsigned* berkisar dari 0 sampai 255.
3. **SMALLINT** – nilai integer yang kecil. *Signed* berkisar dari -32768 sampai 32767. *Unsigned* berkisar dari 0 sampai 65535.
4. **MEDIUMINT** – nilai integer medium. *Signed* berkisar dari -8388608 sampai 8388607. *Unsigned* berkisar 0 sampai 16777215.
5. **BIGINT** – nilai integer dengan ukuran besar. *Signed* berkisar dari -9223372036854775808 sampai 9223372036854775807. *Unsigned* berkisar dari 0 sampai 18446744073709551615.
6. **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.
7. **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.

b. Tipe Data Tanggal dan Waktu

Berikut ini daftar tipe data dan waktu:

1. **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.
2. **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.
3. **TIME** – menyimpan nilai waktu dengan format HH:MM:SS. Contohnya, 15:30:00.
4. **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.

c. Tipe Data String

Berikut ini daftar tipe data string:

1. **CHAR** – string dengan panjang tetap antara 1 sampai 255 karakter.
2. **VARCHAR –** string dengan panjang bervariasi antara 1 sampai 255 karakter.
3. **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.
4. **TINYBLOB/TINYTEXT** – sebuah kolom BLOB atau TEXT dengan panjang maksimum 255 karakter.

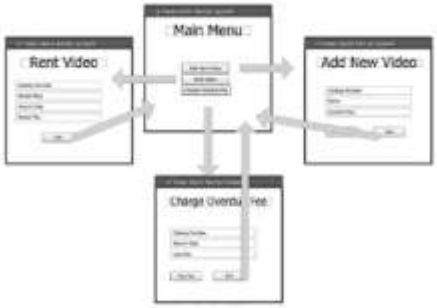
5. **MEDIUMBLOB/MEDIUMTEXT** – sebuah kolom BLOB atau TEXT dengan panjang maksimum 16777215 karakter.
6. **LONGLOB/LONGTEXT** – sebuah kolom BLOB atau TEXT dengan panjang maksimum 4294967295.
7. **ENUM** – berdasarkan Sutaji (2012), enum adalah sebuah *enumeration*. Sebuah obyek string yang hanya boleh memiliki satu nilai, yang terambil dari 'value1','value2,'...,NULL atau nilai spesial ""*error*. Sebuah *enum* dapat menampung 65535 pilihan nilai.

2.14 *Windows Navigation Diagram* (WND)

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)

2.15 PHP

Berdasarkan Tim EMS (2016), PHP adalah singkatan dari *Hypertext Prepocessing*. Merupakan bahasa *scripting* untuk web yang cukup popular. PHP adalah bahasa *script*, artinya ditanamkan atau disisipkan ke dalam HTML (*Hypertext Markup Language*). Salah satu kelebihan PHP dibandingkan dengan bahasa pemrograman web lainnya adalah banyak web server yang mendukung PHP mulai dari Apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah. Ada juga paket web server yang memudahkan melakukan *instalasi* sekali klik, seperti XAMPP yang tersedia dalam berbagai system operasi.

Berdasarkan Winarno dan Zaki (2013). PHP adalah bahasa pemrograman yang memungkinkan *programmer* menggenerate kode HTML secara dinamis, artinya *programmer* bisa membuat tampilan halaman web yang dinamis, bisa berubah-ubah sesuai dengan keinginan *programmer*nya.

Berdasarkan Jubilee (2017), PHP merupakan bahasa pemrograman yang digunakan untuk membuat aplikasi berbasis website. Sebagai sebuah aplikasi, website tersebut hendaknya memiliki sifat dinamis dan interaktif. Memiliki sifat dinamis artinya, website tersebut bisa berubah tampilan kontennya sesuai kondisi tertentu (misalnya menampilkan produk berbeda-beda untuk setiap pengunjung). Interaktif artinya, website tersebut dapat member *feedback* bagi *user* (misalnya, menampilkan hasil pencarian produk). Kode standar untuk mendesain website sebenarnya adalah HTML. Namun, tanpa memanfaatkan PHP, kode HTML tidak dapat digunakan untuk membuat website dengan dua sifat diatas. Dengan kata lain jika mengandalkan HTML saja, tampilan situs yang dilihat para pengunjung satu dan lainnya relative sama dan statis. Dengan memanfaatkan PHP, *programmer* dapat membuat sistem *username login, shopping card, database,* dan sebagainya.

Berdasarkan Adelheid dan Khairil (2012), PHP atau *Hypertext Prepocessor* merupakan bahasa berbentuk *script* yang ditempatkan dalam server dan dieksekusi di dalam server untuk selanjutnya ditransfer dan dibaca oleh *client*. PHP juga bisa disisipkan dalam bahasa HTML. PHP pertama kali diciptakan oleh

seorang pria berkewarganegaraan Denmark yang bernama Rasmus Lerdorf pada tahun 1995.

Berdasarkan Wikipedia Indonesia, PHP atau *Hypertext Prepocessor* adalah bahasa skrip yang dapat disisipkan ke dalam HTML. PHP banyak digunakan untuk membuat program web dinamis. Dalam PHP, mempunyai berbagai sintaks dasar. Sintaks tersebut yaitu antara lain:

1. Pembatas

   PHP hanya mengeksekusi kode yang ditulis dalam pembatas sebagaimana ditentukan oleh dasar sintaks PHP. Apapun yang di luar pembatas tidak diproses oleh PHP. Pembatas yang paling umum adalah "<?php" untuk membuka dan "?>" untuk menutup kode PHP. Tujuan dari pembatas ini adalah untuk memisahkan kode PHP dari kode di luar PHP, seperti HTML, Javascript.

2. Variabel

   Variabel diawali dengan simbol *dollar* $. Pada versi PHP 5, diperkenalkan jenis isyarat yang memungkinkan fungsi untuk memaksa $ menjadi parameter objek dari *class* tertentu seperti array, atau fungsi. Contoh variabel dapat ditulis sebagai $nama_variabel. Penulisan fungsi, penamaan kelas, nama variabel adalah peka akan huruf besar dan huruf kecil.

3. Komentar

   PHP memiliki 3 jenis sintaks sebagai komentar pada kode yaitu tanda blok / * * /, komentar 2 baris // serta tanda pagar # digunakan untuk komentar satu baris. Komentar bertujuan untuk meninggalkan catatan pada kode PHP dan tidak akan diterjemahkan ke program.

4. Fungsi

   Berikut adalah contoh dari penulisan fungsi atau *method* yang ada di PHP.

```
Function tampilkan ($data=" ") // mendefinisikan fungsi, "tampilkan" adalah nama sebuah fungsi
        If ($data) return $data;
Else return 'tidak ada data'; // melakukan proses pengolahan data
Echo tampilkan("isi halaman") // menjalankan fungsi
```

PHP memiliki 8 tipe data, yaitu: *Boolean, Integer, Float/Double, String, Array, Object, Resource, NULL.*

2.16 *Framework*

Berdasarkan Supono dan 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:

b. Mempercepat dan mempermudah pembangunan sebuah aplikasi web.

c. Relatif memudahkan dalam proses *maintenance* karena sudah ada pola tertentu dalam sebuah *framework* (dengan syarat *programmer* mengikuti pola standar yang ada)

d. Umumnya *framework* menyediakan fasilitas-fasilitas umum yang dipakai sehingga *programmer* tidak perlu membangun dari awal (misalnya validasi, ORM, pagination, *multiple database*, *scaffolding*, pengaturan *session*, *error handling*, dan lain-lain.

Berdasarkan Jubilee (2015), *framework* merupakan banyak kode, yang disimpan dalam beberapa file yang terpisah, dan memudahkan dalam penggunaan

kode yang digunakan secara berulang-ulang. Dengan *framework*, *programmer* tidak perlu menulis baris kode yang panjang untuk fungsi tertentu. *Programmer* hanya perlu menggunakan fungsi yang sudah dituliskan ke dalam *framework*.

2.17 *CodeIgniter*

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

2 *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*.

3 *Model,* biasanya berhubungan langsung dengan *database* untuk memanipulasi data (*insert, update, delete, search*) menangani validasi dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.

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

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

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

mempermudah proses. Terdapat beberapa kelebihan penggunaan *codeigniter* dalam pengembangan suatu proyek, yaitu:

a. Menghemat Waktu

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

b. *Code Reuse*

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

c. Dukungan Komunitas

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

Berdasarkan Wikipedia Indonesia, *codeigniter* merupakan aplikasi sumber terbuka yang berupa *framework* PHP dengan model MVC (*Model*, *View*, *Controller*) untuk membangun website dinamis dengan menggunakan PHP. *Codeigniter* memudahkan *developer* untuk membuat aplikasi web dengan cepat dan mudah dibandingkan dengan membuatnya dari awal. *Codeigniter* pertama kali dirilis pada 28 Februari 2006. Model *view controller* merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, berawal pada bahasa pemrograman *small talk*¸ MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, *user interface*, dan bagian yang menjadi kontrol aplikasi. 3 jenis komponen MVC adalah:

1. *View*, bagian ini biasanya berupa file *template* HTML, yang diatur oleh *controller*. *View* berfungsi untuk menerima dan merepresentasikan data kepada *user*. Bagian ini tidak memiliki akses langsung terhadap bagian model.
2. *Model*, biasanya berhubungan langsung dengan *database* untuk memanipulasi data (*insert, update, delete, search),* menangani validasi dari

bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.

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

CodeIgniter adalah Kerangka Pengembangan aplikasi (sebuah toolkit) untuk orang-orang yang membangun situs *web* dengan menggunakan bahasa PHP (*Hypertext Prepocessing*). Tujuannya adalah untuk memungkinkan *programmer* mengembangkan proyek lebih cepat dari pada menulis kode dari awal dengan menyediakan kumpulan pustaka untuk tugas-tugas yang biasanya dibutuhkan, serta antarmuka yang sederhana dan struktur logis untuk mengakses pustaka-pustaka ini. *CodeIgniter* memungkinkan *programmer* fokus pada proyek dengan meminimalkan jumlah kode yang diperlukan untuk tugas yang diberikan.

CodeIgniter didasarkan pada pola pengembangan *Model-View-Controller*. MVC adalah pendekatan perangkat lunak yang memisahkan logika aplikasi dari presentasi. Dalam praktiknya, memungkinkan halaman *web* pengguna mengandung *scripting minimal* karena presentasi terpisah dari *scripting* PHP.

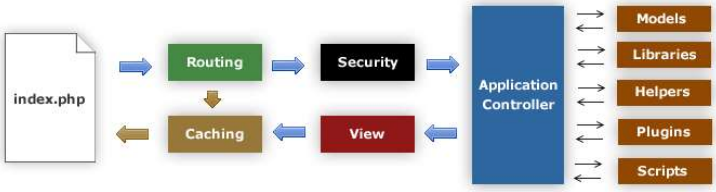
a. *Model*, mewakili struktur data. Biasanya kelas model akan berisi fungsi-fungsi yang membantu untuk mengambil, memasukkan, dan memperbarui informasi dalam *database*.

b. *View*, adalah informasi yang sedang disajikan kepada pengguna. *View* biasanya adalah halaman *web*, tetapi di *CodeIgniter*, tampilan juga bisa berupa fragmen halaman seperti *header* atau *footer*. Hal ini juga dapat berupa halaman RSS, atau jenis "halaman" lainnya.

c. *Controller*, berfungsi sebagai perantara antara *Model*, *View*, dan sumber daya lain yang diperlukan untuk memproses permintaan HTTP dan menghasilkan halaman web.

CodeIgniter memiliki pendekatan yang cukup longgar untuk MVC karena *Model* tidak diperlukan. Jika pengguna tidak memerlukan pemisahan tambahan, atau menemukan bahwa mempertahankan *model* memerlukan lebih banyak

kerumitan daripada yang diinginkan, pengguna dapat mengabaikannya dan membangun aplikasi secara *minimal* menggunakan Pengontrol dan Tampilan.



Gambar II.7 Alur Kerja *CodeIgniter*

(Sumber: *CodeIgniter*, 2015)

Indeks.php berfungsi sebagai pengontrol depan, menginisialisasi sumber daya dasar yang dibutuhkan untuk menjalankan *CodeIgniter*.

1. *Router* memeriksa permintaan HTTP untuk menentukan apa yang harus dilakukan.
2. Jika *file* cache ada, akan dikirim langsung ke *browser*, melewati eksekusi sistem normal.
3. Keamanan. Sebelum pengontrol aplikasi dimuat, permintaan HTTP dan data yang dikirim oleh pengguna disaring untuk keamanan.
4. *Controller* memuat *model*, pustaka inti, pembantu, dan sumber daya lain yang diperlukan untuk memproses permintaan khusus.
5. Tampilan akhir ditampilkan kemudian dikirim ke *browser web* untuk dilihat. Jika cache diaktifkan, tampilan akan di-cache terlebih dahulu sehingga pada permintaan berikutnya dapat dilayani.

2.18 *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*).

Berdasarkan Utami dan Asnawati (2015), kategori *error* yang akan diketahui melalui *black box testing* adalah:

a. Fungsi yang hilang atau tak benar.
b. *Error* dari antar-muka.
c. *Error* dari struktur data atau akses *database*.
d. *Error* dari kinerja atau tingkah laku.

2.18.1 Metode *Black Box Testing*

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

2.18.2 Perbedaan *White Box* dan *Black Box*

White Box (*Struktural*)

1. Dilakukan oleh penguji yang mengetahui tentang QA.
2. Melakukan *testing* pada *software*/program aplikasi menyangkut *security* dan *performance* program tersebut (meliputi tes *code*, desain implementasi, *security*, *data flow*, *software failure*).
3. Dilakukan seiring dengan tahapan pengembangan *software* atau pada tahap *testing*.

Black Box (Fungsional)

1. Dilakukan oleh penguji *independent*.
2. 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.*

2.19 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.

BAB III
METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metodologi berasal dari bahasa yunani yaitu "*methodos*" dan "*logos*". *Methodos* (metode) terdiri dari 2 kata yaitu "*metha*" yang artinya melewati, menempuh atau melalui dan kata "*hodos*" yang artinya cara atau jalan. Maka dari itu, pengertian dari "metode" ialah cara atau jalan yang harus ditempuh untuk mencapai sebuah tujuan dan sedangkah *logos* berarti ilmu atau bersifat ilmiah. Berikut ini adalah pengertian dari metodologi penelitian menurut para ahli:

a. Menurut KBBI (Kamus Besar Bahasa Indonesia), metode merupakan cara kerja yang mempunyai sistem dalam memudahkan pelaksanaan dari suatu kegiatan untuk mencapai sebuah tujuan tertentu.

b. Menurut Sugiyono, metode penelitian adalah cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu.

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

3.2 Jenis dan Sumber Data

Data yang diperlukan dalam penelitian ini ada dua jenis, yaitu data primer dan data sekunder. Sumber dari data-data ini berasal dari tempat yang diamati untuk penelitian adalah di PT Teknikatama Karya Mandiri:

1. Data Primer

   Data Primer adalah data yang diperoleh secara langsung dari PT Teknikatama Karya Mandiri, dimana pengumpulan data atau informasi dilakukan langsung dari objek yang diteliti. Data-data tersebut adalah data yang digunakan dalam proses penyimpanan barang jadi diantaranya,

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

2. Data Sekunder

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

3.3 Metode Pengumpulan Data

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

a. Studi Lapangan

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

1. Pengamatan, tahap ini dilakukan secara langsung pada Departemen *Warehouse* di PT Teknikatama Karya Mandiri dengan mengamati proses penyimpanan barang jadi dari bagian *Quality Control* memeriksa standar kualitas barang yang telah diproduksi oleh bagian Produksi, lalu barang jadi dikemas, barang jadi dikirim ke gudang, Operator *Warehouse* mencatat barang jadi yang masuk, dan barang jadi yang keluar untuk dikirim jika waktu pengiriman telah tiba.
2. Wawancara, yaitu mencari data yang dibutuhkan secara langsung melalui memberikan pertanyaan yang ingin ditanyakan terhadap segala hal yang diperlukan pada penyusunan Tugas Akhir ini. Wawancara ini dilakukan kepada Operator *Warehouse* di PT Teknikatama Karya Mandiri.

b. Studi Kepustakaan

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

3.4 Metode Pengembangan Sistem

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

1. Tahap Perencanaan (*Planning*)

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

2. Tahap Analisis (*Analysis*)

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

3. Tahap Desain (*Design*)

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

4. Tahap Implementasi (*Implementation*)

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

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

3.5 Kerangka Penelitian

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

1. Studi Pendahuluan

   Studi Pendahuluan merupakan langkah awal yang dilakukan untuk memulai penelitian. Studi pendahuluan dilakukan dengan turun langsung ke lokasi untuk mengetahui gambaran yang jelas mengenai sistem yang sedang berjalan pada Departemen *Warehouse*. Studi dapat dilakukan dengan melakukan observasi, wawancara, dan studi pustaka.

2. Identifikasi Masalah

   Pada tahap ini, peneliti mempersempit ruang penelitian dengan menganalisis sistem penyimpanan barang jadi yang sedang berjalan lalu mengidentifikasi permasalahan yang ada pada sistem yang sedang berjalan dengan melakukan wawancara dan observasi. Ketika wawancara dilakukan, Operator *Warehouse* menyampaikan masukan jika sistem yang tengah berjalan belum terkomputerisasi sehingga Operator *Warehouse* memasukkan data dalam kartu stok maupun membuat laporan secara manual. Adapun permasalahan lain yang teridentifikasi melalui observasi adalah daftar stok yang tidak dicatat dengan rutin sehingga aktivitas keluar/masuknya barang jadi kurang terkontrol dengan baik. Serta pengambilan barang jadi dilakukan secara acak.

3. Identifikasi Solusi

   a. Maksud dan tujuan dari penelitian ini yakni:

      Membuat rancang bangun sistem informasi sehingga pencatatan stok barang dapat terkontrol dengan baik. Membuat suatu sistem informasi

dan *database* untuk aktivitas keluar/masuknya barang jadi yang mana data tersebut dapat membantu Operator Gudang untuk membuat sebuah laporan dengan cepat. Menerapkan metode FIFO (*First In First Out*) dalam penyimpanan barang jadi. Barang jadi yang pertama masuk ke dalam gudang setelah waktu produksi, adalah barang jadi yang keluar dari gudang untuk dikirim ke pelanggan. Hal ini guna meminimalisir adanya barang jadi yang kotor, ataupun tergores karena bertumpukan di gudang setelah waktu produksi.

4. Batasan Masalah

Pada tahap ini bertujuan untuk membatasi masalah agar penelitian lebih terarah sehingga dapat menunjukkan gambaran yang lebih spesifik mengenai arah pemecahannya. Berikut batasan dalam penulisan tugas akhir ini:

a. Penelitian dilakukan selama satu bulan bertempat di PT Teknikatama Karya Mandiri J*alan Raya Bahkilong* No.88 Cikarang Selatan – Bekasi 17330.
b. Pengamatan dilakukan pada bagian Departemen *Warehouse.* J*angka waktu untuk penelitian ini, mulai dari 31 Juli 2017 sampai dengan 28 Agustus 2017.*

5. Penerapan Metode *Waterfall*

a. Perencanaan (*Planning*)

Pada tahap ini akan dilakukan perencanaan untuk membangun suatu sistem informasi.

b. Melakukan Analisis (*Analysis)*

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

c. Membuat Desain (*Design)*

Membuat pemodelan sistem dengan *Unified Modeling Language* (UML) yaitu menggunakan *usecase* diagram, *activity* diagram, *sequence* diagram dan *class* diagram dan membuat rancangan

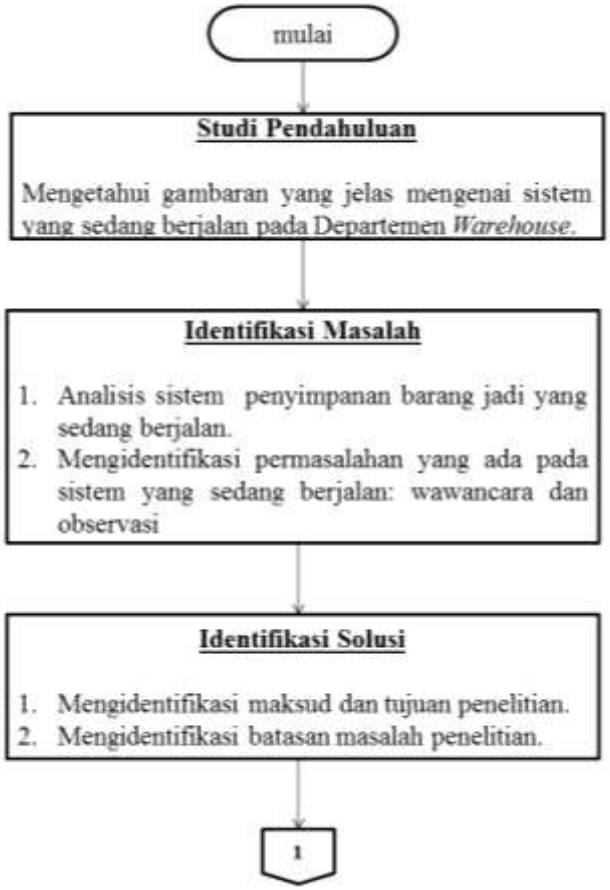
antarmuka. Data-data yang didapat dari tahap analisis diterapkan dalam tahap desain.

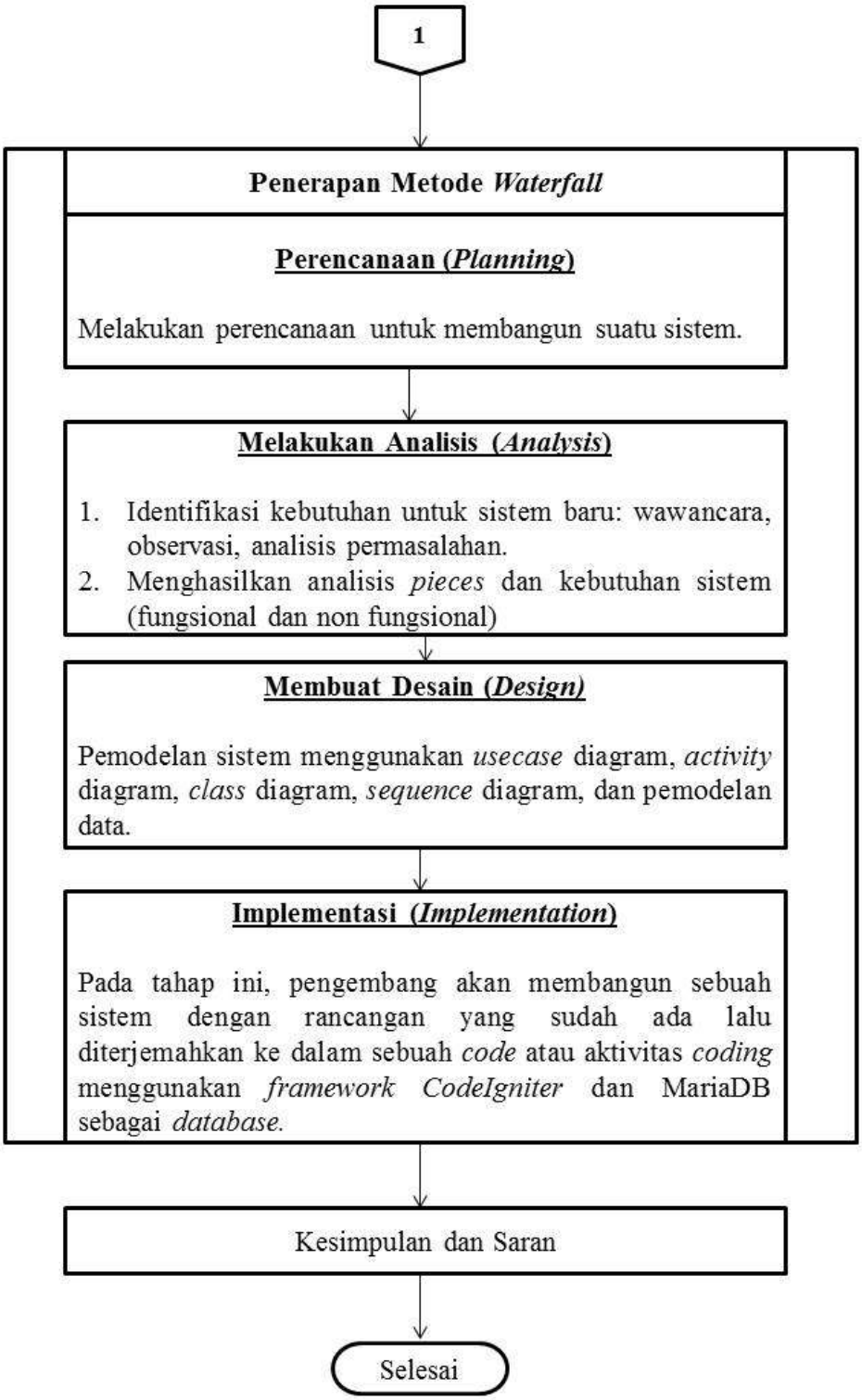
d. Implementasi (*Implementation*)

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

6. Kesimpulan dan Saran

Kesimpulan digunakan untuk membandingkan hasil penelitian atau pengembangan sistem dengan sistem sebelumnya. Sedangkan saran berisi masukan untuk pengembangan sistem selanjutnya. Berikut merupakan diagram alir kerangka penelitian yang digunakan dalam laporan tugas akhir ini dapat dilihat pada Gambar III.1.





Gambar III.1 Kerangka Penelitian

(Sumber: Pengolahan Data, 2018)

BAB IV
PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Sekilas Perusahaan

PT Teknikatama Karya Mandiri merupakan salah satu perusahaan kecil menengah yang bergerak dalam bidang industri otomotif yang memproduksi komponen untuk kendaraan roda empat seperti *pin shaft, gusset, end housing,* dan *reinforcement.*

PT Teknikatama Karya Mandiri didirikan oleh seorang pemilik perusahaan bernama Bapak Anton Saputra. Pertama kali didirikan di kawasan industri Jababeka, tepatnya di Jl. Jababeka III Blok C, No.45, kawasan industri Jababeka I, Cikarang Selatan - Bekasi. PT Teknikatama Karya Mandiri awalnya didirikan dengan bantuan tenaga kerja hanya 2 orang saja yaitu di bagian produksi.

Untuk awal pendirian PT Teknikatama Karya Mandiri memiliki 8 pelanggan, dan 11 rekanan *subcontractor*. Pemilik memilih kawasan industri Jababeka sebagai tempat pendirian usaha dengan alasan untuk mendekati pelanggannya dan untuk meminimalisir biaya distribusi pengiriman barang kepada pelanggan. Untuk pendistribusian produk ke pelanggan, perusahaan memiliki kendaraan berupa mobil *pick up* ukuran kecil untuk proses pengiriman.

Seiring berjalannya waktu, pada maret 2016, PT Teknikatama Karya Mandiri akhirnya membeli gedung untuk pabrik dan kantor yang berada di Cikarang Selatan. Dan PT Teknikatama Karya Mandiri pun berpindah lokasi menjadi di Jalan Raya Bahkilong No.88, Cikarang Selatan-Bekasi sampai dengan saat ini. Hingga sampai saat ini PT Teknikatama Karya Mandiri sudah memiliki 22 karyawan untuk bagian produksi dan *staff* kantor. Sudah sejak lama pula PT Teknikatama Karya Mandiri menerapkan Sistem Manajemen Mutu ISO 9000:2008 yang menjamin segala bentuk kegiatan industri yang berlaku sesuai dengan standar sistem manajemen mutu ISO 9001:2008, dan tidak sedikit pula penghargaan sertifikat yang telah didapat oleh PT Teknikatama Karya Mandiri.

Mesin-mesin yang digunakan untuk produksi saat ini 99% sudah berteknologi CNC. Sehingga produk yang dihasilkan sangat mampu untuk bersaing dengan perusahaan pesaing otomotif lainnya. Ditambah dengan kemampuan *skill* karyawan yang terus didukung oleh perusahaan untuk diikutsertakan dalam pelatihan dan pengembangan.

4.2 Profil Umum Perusahaan

Adapun profil usaha lebih rinci sebagai berikut:

Nama perusahaan	: PT Teknikatama Karya Mandiri
Nama pemilik	: Anton Saputra
Jabatan	: Direktur
Tahun berdiri	: 2002
Jenis Usaha	: Manufaktur suku cadang kendaraan roda empat
Daerah kerja	: Cikarang Selatan - Bekasi
Alamat usaha	: Jalan Raya Bahkilong No. 88 Cikarang Selatan - Bekasi 17330
Telepon	: 62-21-22104845
Fax	: 62-21-22104846
Nomor SIUP	: 10261 – 03 / PK / 1.824.271
Nomor TDP	: 09.02.1.46.46396
Nomor NPWP	: 02.187.829.3-036.000
Jumlah pekerja	: 22karyawan
Email	: marketing@teknikatama.co.id

4.3 Visi dan Misi Perusahaan

PT Teknikatama Karya Mandiri sendiri mempunyai visi, misi, dan motto untuk perusahaannya. Adapun visi, misi dan motto tersebut adalah sebagai berikut:

Visi :

"Menjadi perusahaan *manufacturing* yang berkelas dunia dan pemimpin pasar di Indonesia".

Misi :

Menjadi industri pendukung penghasil part OEM, untuk industri otomotif yang dapat dipercaya dengan cara :

1. Menghasilkan produk yang berkualitas tinggi melalui sistem manajemen mutu.
2. Pengembangan sumber daya manusia yang profesional dan inovatif.
3. Mensejahterakan karyawan dan memberikan nilai tambah kepada setiap *stakeholder*.

Motto :

***TEAM WORK, KUALITAS* DAN MANFAAT.**

TEAM WORK

Kami menyadari bahwa tanpa adanya *team work*, kami tidak akan menghasilkan produk yang baik.

KUALITAS

Kami sepakat menghasilkan produk dengan kualitas terbaik

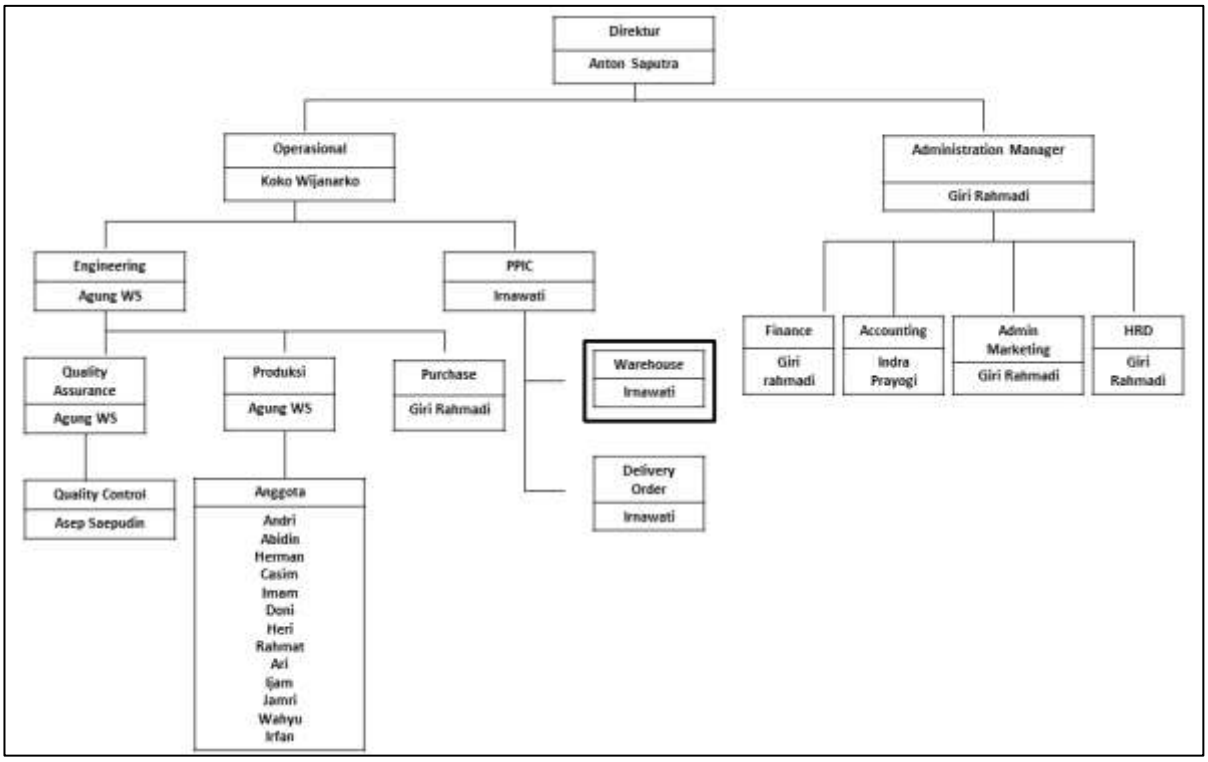
MANFAAT

Kami selalu berusaha agar dimanapun kami berada, kami dapat memberikan mafaat bagi orang lain.

4.4 Struktur Organisasi Perusahaan

Setiap organisasi atau perusahaan pasti mempunyai struktur organisasi yang ada didalamnya untuk saling berkoordinasi mengurus jalannya suatu perusahaan dalam mencapai tujuan. Adapun struktur dari PT Teknikatama Karya Mandiri dapat dilihat pada Gambar IV.1. Keterangan :

= Departemen untuk penelitian



Gambar IV.1 Struktur Organisasi Perusahaan
(Sumber: PT Teknikatama Karya Mandiri, 2017)

4.4.1 Tugas dan Wewenang Jabatan di PT Teknikatama Karya Mandiri

Setiap perusahaan pada umumnya pasti mempunyai sebuah struktur organisasi. Struktur organisasi tersebut sangatlah berguna untuk melaksanakan fungsi perencanaan, pengorganisasian, pengarahan, dan pengawasan. Dengan kata lain, struktur organisasi menjelaskan tentang wewenang, tugas, dan tanggung jawab seseorang yang berada pada setiap departemen. Dan berikut ini adalah gambaran mengenai tugas dan tanggung jawab masing-masing bagian atau departemen yang berada di PT Teknikatama Karya Mandiri:

1. Direktur Perusahaan

a. Memimpin dan mengkoordinir organisasi agar tetap berjalan lancar.

b. Menentukan kebijakan perusahaan.

c. Mempunyai kewenangan dalam membuat perencanaan.

d. Mengendalikan segala sesuatu yang terjadi dalam perusahaan.

e. Mengatur dan mengawasi semua yang terlibat dalam perusahaan.

f. Membangun relasi dengan mitra kerja.

g. Memberikan pelatihan dan pengembangan karyawan.

h. Memiliki kewenangan mengangkat dan memutuskan hubungan kerja dengan karyawan.

2. Departemen Produksi:

a. Bertugas untuk mempersiapkan, merencanakan dan melakukan produksi.

b. Membangun kerjasama yang baik antar bagian produksi.

c. Bertanggungjawab terhadap proses produksi.

d. Mengatur jalannya operasional sehingga berjalan dengan lancar.

e. Mengawasi proses produksi.

f. Mencatat seluruh jadwal produksi pengiriman barang.

3. Departemen *Warehouse* :

a. Mencatat persediaan barang yang ada di gudang.

b. Barang yang ada di gudang berupa barang *Not Good, Finish Good, Repair,* dan *Row Materials.*

c. Memproses barang yang diminta oleh produksi untuk diolah.

d. Membuat surat jalan untuk mengirim barang jadi ke pelanggan.

4. Departemen *Delivery Order* :

a. Bertugas untuk mengirim produk barang jadi ke pelanggan menurut pesanan.

5. Departemen *Engineering* :

a. Bertugas untuk menjamin kondisi mesin dapat beroperasi dengan baik dan membuat proses produksi berjalan dengan baik dan efisien.

6. Departemen *Purchasing* :

a. Bertugas sebagai penyedia barang dan jasa agar proses produksi berjalan dengan baik.

7. Departemen *Human Resource Development* :

a. Bertugas penanganan sumber daya manusia seperti dalam perekrutan karyawan baru, pengajuan cuti dan peningkatan *skill* karyawan.

8. Departemen *Quality Assurance* :

a. Bertugas untuk menjamin produk yang dihasilkan memenuhi standar kualitas yang diinginkan pelanggan.

9. Departemen *Marketing* :

a. Bertugas untuk melakukan analisa pelanggan dan mencari *order* untuk produksi.
b. Mempromosikan barang yang akan dipasarkan.
c. Mencatat barang yang akan dipasarkan.
d. Mencatat seluruh jadwal yang berhubungan dengan pemasaran seperti jadwal pameran.
e. Bertanggung jawab atas penjualan produk.
f. Membangun relasi dengan mitra kerja.

10. Departemen *Finance* :

a. Bertugas menangani masalah keuangan perusahaan.

11. Departemen *Accounting* :

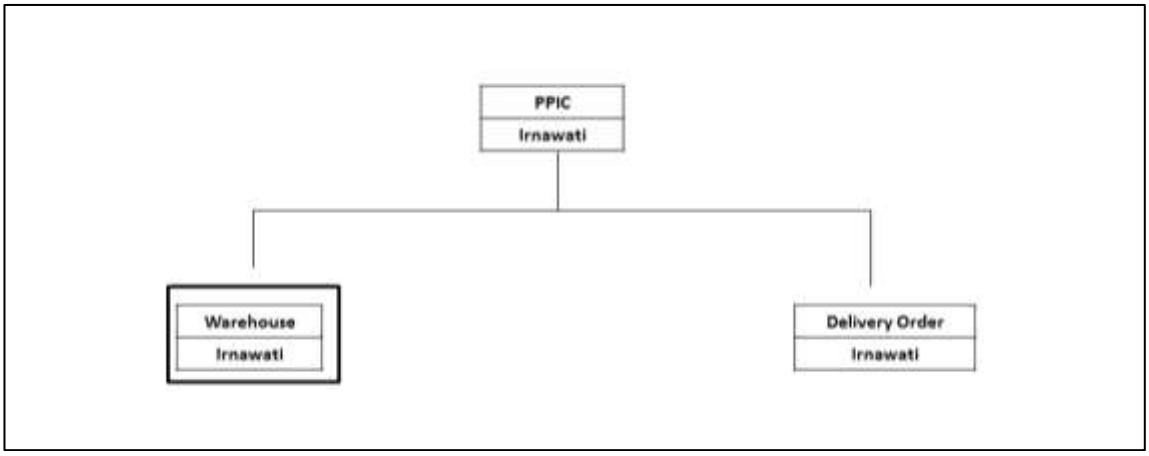
a. Bertugas menangani semua laporan keuangan perusahaan.
b. Membuat seluruh laporan keuangan perusahaan.
c. Mengatur keluar masuknya keuangan perusahaan.
d. Memeriksa keuangan perusahaan.

4.4.2 Struktur Organisasi pada Departemen *Warehouse*

Warehouse atau pergudangan berfungsi untuk menyimpan barang dalam sementara waktu sebelum barang tersebut dikeluarkan dari gudang untuk dikirim ke pelanggan atau diproduksi.

PT Teknikatama Karya Mandiri hanya terdapat satu gudang untuk menyimpan barang jadi atau bahan baku. Tetapi jenis barang yang ada di gudang akan dipisah menurut areanya. Untuk bahan baku, akan diletakkan pada area bahan baku. Untuk barang jadi akan diletakkan pada area barang jadi.

Berikut merupakan struktur organisasi pada Departemen *Warehouse* yang dapat dilihat pada Gambar IV.2



Gambar IV.2 Struktur Organisasi Departemen *Warehouse*
(Sumber: PT Teknikatama Karya Mandiri, 2017)

4.5 Barang Jadi pada PT Teknikatama Karya Mandiri

PT Teknikatama Karya Mandiri mendapatkan bahan bakunya dari pelanggan. Maka dari itu, PT Teknikatama Karya Mandiri hanya mengolah produk bahan baku dari pelanggan dengan melakukan penghalusan *body* material, pemotongan bentuk atau melakukan pembolongan pada bahan baku. Adapun macam-macam barang jadi adalah sebagai berikut:

1. *Gusset*

Bahan baku yang digunakan adalah lempengan besi berbentuk persegi yang sudah dipotong sesuai standar ukuran produk. Untuk membuat 1 produk

Gusset diperlukan 4 kali proses produksi yaitu melalui operator 1, 2, dan 3. Berikut ini adalah urutan dari proses produksi *Gusset :*

1. Proses 1 digunakan untuk proses *facing* yaitu, membentuk dan menghaluskan bagian sisi samping kiri dan kanan, bagian atas dan bawah. Proses ini dikerjakan oleh satu orang operator produksi dengan menggunakan mesin POLO. Kapasitas kerja mesin tersebut adalah 8 jam kerja dengan 1 jam istirahat. Hasil proses produksi 1 dapat dilihat pada Gambar IV.3.

Gambar IV.3 Hasil Proses Produksi 1 *Gusset*
(Sumber: PT Teknikatama Karya Mandiri, 2017)

2. Proses 2 digunakan untuk proses *driling* yaitu, proses memberi lubang berdiameter kecil pada produk. Proses ini dikerjakan oleh 1 operator produksi. Mesin yang digunakan adalah mesin dengan teknologi CNC dengan merk *Brather*, mesin ini memilki 8 jam kerja dengan 1 jam istirahat. Kapasitas produksi 120 pcs dalam 1 *shift*. Hasil proses produksi 2 dapat dilihat pada Gambar IV.4.

Gambar IV.4 Hasil Proses Produksi 2 *Gusset*
(Sumber: PT Teknikatama Karya Mandiri, 2017)

3. Proses 3 adalah proses *borring* yaitu, proses memberi lubang berdiameter besar dibagian tengah produk. Proses produksi dilakukan oleh 1 orang operator produksi dengan menggunakan mesin berteknologi CNC dengan merek mesin *Headman 3 Speedio S500X*. Kapasitas jam kerja mesin adalah 8 jam dengan 1 jam istirahat. Kapasitas produksi 120 pcs dalam 1 *shift*. Hasil proses produksi 3 dapat dilihat pada Gambar IV.5.

Gambar IV.5 Hasil Proses Produksi 3 *Gusset*
(Sumber: PT Teknikatama Karya Mandiri, 2017)

4. Proses 4 adalah pengecekan kualitas barang oleh Departemen *Quality Control*. Adapun hasil barang jadi produk *Gusset* dapat dilihat pada Gambar IV.6.

Gambar IV.6 Barang Jadi *Gusset*
(Sumber: PT Teknikatama Karya Mandiri, 2017)

2. *Reinforcement*

Untuk membuat produk *Reinforcement* hanya dibutuhkan 1 kali proses produksi yaitu proses pemberian lekukan pada bahan baku yang ada dan

menghaluskan seluruh sisi produk. Proses pengerjaan dikerjakan oleh 1 operator produksi dengan mesin manual bermerek *Kiwa*. Hasil produksi barang jadi *Reinforcement* dapat dilihat pada Gambar IV.7.

Gambar IV.7 Barang Jadi *Reinforcement*
(Sumber: PT Teknikatama Karya Mandiri, 2017)

3. *Pin Shaft*

Untuk membuat produk *Pin Shaft* diperlukan 3 kali proses produksi. Berikut ini adalah urutan proses produksi *Pin Shaft :*

1. Proses pertama dan kedua adalah proses *facing* and *machining* yaitu, untuk membentuk ukuran bahan baku produk sesuai dengan standar ukuran produk yang sudah ditentukan, lalu menghaluskan bagian atas, tengah dan bawah sisi produk. Mesin yang digunakan masih manual dengan merek *Mazak* yang memiliki kapasitas produksi 140 pcs dalam 1 *shift* dan kapasitas jam kerja mesin 8 jam kerja dengan 1 jam istirahat. Proses pertama dikerjakan oleh 1 operator produksi.
2. Proses kedua yaitu proses *milling* yang merupakan proses pembuatan lekukan bagian tengah produk, proses ini dikerjakan oleh 1 operator produksi dengan menggunakan mesin berteknologi CNC bermerek *Kiwa*, memiliki kapasitas jam kerja mesin 8 jam kerja dengan 1 jam istirahat. Dan memiliki kapasitas produksi 140 pcs dalam 1 *shift*.

Produk *Pin Shaft* dibedakan dalam 2 jenis antara lain, produk *Pin Shaft* model lama dengan 1 undakan dan, *Pin Shaft* model baru dengan 2 undakan. Hasil produksi barang jadi *Pin Shaft* dapat dilihat pada Gambar IV.8.

Pin Shaft model lama	*Pin Shaft* model baru

Gambar IV.8 Barang Jadi *Pin Shaft*
(Sumber: PT Teknikatama Karya Mandiri, 2017)

4. *End Housing*

Untuk produk *End Housing*, terdapat 2 proses produksi. Berikut adalah urutan dari proses produksi *End Housing :*

1. Proses pertama dikerjakan oleh 1 operator produksi adalah proses *facing*, yaitu membentuk ukuran produk sesuai dengan standar ukuran produk dan proses *machining* yaitu, proses menghaluskan diameter bagian dalam produk. Proses produksi menggunakan mesin berteknologi CNC dengan merek mesin *Headman* 1 yang berkapasitas 8 jam kerja dengan 1 jam istirahat dan kapasitas produksi 140 pcs dalam 1 *shift*. Hasil proses produksi 1 *End Housing* dapat dilihat pada Gambar IV.9.

Gambar IV.9 Hasil Proses Produksi 1 *End Housing BZ 080*

(Sumber: PT Teknikatama Karya Mandiri, 2017)

2. Proses kedua dilakukan oleh 1 operator produksi adalah untuk proses *facing* dan *machining* yaitu proses pembentukan ukuran produk sesuai ukuran standar produk dan menghaluskan diameter bagian luar produk. Proses produksi menggunakan mesin berteknologi CNC ber merek *Headman* 2 dengan kapasitas jam kerja mesin 8 jam kerja dengan 1 jam istirahat. Kapasitas produksi 140 pcs dalam 1 *shift*. Hasil proses produksi 2 *End Housing* dapat dilihat pada Gambar IV.10

Gambar IV.10 Hasil Proses Produksi 2 *End Housing BZ 080*

(Sumber: PT Teknikatama Karya Mandiri, 2017)

Ada 2 jenis *End Housing* yang di produksi oleh PT Teknikatama Karya Mandiri. Kedua produk tersebut adalah *End Housing BZ 080 dan BZ 270*. Perbedaannya adalah, *end housing BZ 080*, memiliki badan produk yang lebih besar serta permukaan pada sisi-sisi dari lingkaran atas produk lebih halus dari pada produk *end housing BZ 270* dan tidak mempunyai undakan.

Sementara untuk *End Housing BZ 270*, memiliki badan produk yang lebih kecil serta permukaan pada sisi-sisi dari lingkaran atas produk lebih kasar dari pada permukaan *End Housing BZ 080* dan memiliki 1 undakan. Hasil dari barang jadi *End Housing* dapat dilihat pada Gambar IV.11.

End Housing BZ 080	*End Housing BZ 270*

Gambar IV.11 Barang Jadi *End Housing*
(Sumber: PT Teknikatama Karya Mandiri, 2017)

4.6 Penyimpanan

Gudang berpengaruh besar terhadap perusahaan, tanpa adanya gudang maka barang jadi tidak dapat disimpan sampai tiba di tangan pelanggan. Pengendalian dan pengaturan barang ini harus dilakukan dengan baik agar barang yang ada di gudang dapat terkontrol dengan baik. Kegiatan penyimpanan adalah salah satu tugas Departemen *Warehouse* pada PT Teknikatama Karya Mandiri.

Proses penyimpanan dilakukan untuk menjaga barang jadi hasil produksi sebelum dikirimkan kepada pelanggan. Selain itu penyimpanan dengan beberapa teknik penyimpanan salah satunya *First In First Out* (FIFO) diharapkan dapat membantu mengendalikan penyimpanan barang dengan baik.

4.6.1 Siklus Penyimpanan Barang Jadi

Setiap perusahaan memiliki gudang dan ada satu Departemen yang bertugas untuk mengatur atau mengelola gudang tersebut. Proses penyimpanan harus dilakukan agar barang yang telah diproduksi tidak mengalami kerusakan dan dapat dikeluarkan sewaktu-waktu. Adapun prosedur penyimpanan barang jadi adalah sebagai berikut:

1. Barang yang telah diproduksi akan diserahkan kepada Departemen *Quality Control* untuk diperiksa sesuai standar kualitasnya.
2. Barang jadi yang telah diperiksa oleh pihak *Quality Control* lalu dikirim ke gudang dengan menggunakan *box-box* untuk disimpan.
3. Barang jadi akan diletakkan menurut nama barangnya. Barang jadi dengan nama *Gusset* akan diletakkan di area *Gusset*, barang jadi dengan nama *Pin Shaft* akan diletakkan di area *Pin Shaft*, barang jadi dengan nama *End Housing* akan diletakkan di area *End Housing*, dan barang jadi dengan nama *Reinforcement* akan diletakkan di area *Reinforcement*.
4. Jika waktu pengiriman tiba, Operator *Warehouse* akan mengeluarkan barang dari gudang.
5. Operator *Warehouse* dibantu oleh *Delivery Order* mengeluarkan barang jadi secara acak dari gudang dan dipindahkan ke dalam mobil *pick up*.

4.6.2 Sistem Penyimpanan Barang Jadi yang Berjalan

Barang jadi yang masuk ke dalam gudang setelah di cek kualitasnya oleh Departemen *Quality Control*, akan dikirim ke dalam gudang untuk disimpan menunggu waktu pengiriman tiba. Contohnya untuk produk *Gusset*, akan dimasukkan ke dalam kotak khusus *Gusset*. Lalu, kotak tersebut terdapat nama dari pelanggan yang memesan barang. Setelah dimasukkan ke dalam kotak, barang jadi akan diletakkan menurut nama barangnya. Jika kapasitas gudang telah terisi penuh, maka barang akan ditumpuk. Operator *Warehouse* akan mencatat barang yang masuk ke dalam gudang. Jika waktu pengiriman telah tiba, barang jadi akan dikeluarkan dari gudang lalu Operator mencatat barang yang keluar dari gudang ke dalam daftar stok dan membuat surat jalan untuk pihak *Delivery Order*.

Barang jadi akan diletakkan ke dalam mobil *pick up* untuk dikirim ke pelanggan. Untuk tata letak barang di gudang, dapat dilihat pada Gambar IV. 12.

Gambar IV.12 Tata Letak Barang di Gudang PT Teknikatama Karya Mandiri
(Sumber: PT Teknikatama Karya Mandiri, 2017)

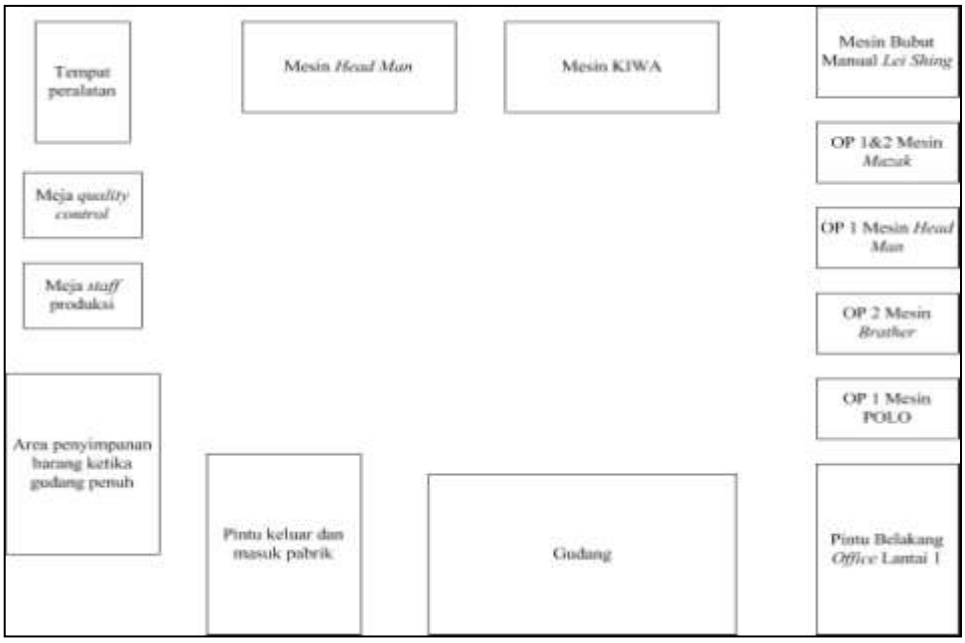
Adapun nama-nama pelanggan yang memesan barang pada PT Teknikatama Karya Mandiri adalah dapat dilihat pada Tabel IV.1.

Tabel IV.1. Nama Pelanggan Barang Jadi PT Teknikatama Karya Mandiri

No	Nama Pelanggan	Barang yang Dipesan
1.	PT Gemala Kempa Daya	*Gusset* dan *Reinforcement*
2.	PT Inti Ganda Perdana	*End Housing*
3.	PT Yazuho	*Pin Shaft*

Sumber: PT Teknikatama Karya Mandiri 2017

Untuk lokasi gudang PT Teknikatama Karya Mandiri, terletak dekat dengan pintu keluar. Pada samping kiri gudang, terdapat ruangan *office* lantai 1. Depan gudang, terdapat mesin-mesin yang berjejer untuk mengolah bahan baku yang tengah diproduksi. Penggambaran lokasi gudang pada pabrik di PT Teknikatama Karya Mandiri, dapat dilihat pada Gambar IV.13.



Gambar IV.13 Lokasi Gudang PT Teknikatama Karya Mandiri
(Sumber: PT Teknikatama Karya Mandiri, 2017)

4.6.3 Dokumen pada Penyimpanan Barang Jadi yang Berjalan

Dalam proses penyimpanan barang jadi, tentunya terdapat dokumen-dokumen yang berperan untuk membantu Operator *Warehouse*. Dokumen yang terlibat pada Departemen *Warehouse* ini tidak hanya satu dokumen. Tetapi ada beberapa macam dokumen. Berikut ini adalah dokumen-dokumen yang terlibat dalam sistem penyimpanan barang jadi:

1. Daftar Stok

Daftar stok ini digunakan oleh Operator *Warehouse* untuk mencatat arus barang yang keluar maupun masuk ke dalam gudang. Seperti untuk mencatat berapa jumlah barang yang keluar, berapa jumlah barang yang masuk, berapa stok barang terakhir di gudang. Berikut adalah keterangan yang terdapat dalam daftar stok yang digunakan oleh Operator *Warehouse* untuk mencatat stok dalam gudang:

Nama Barang : untuk mencatat nama barang yang disimpan dalam gudang

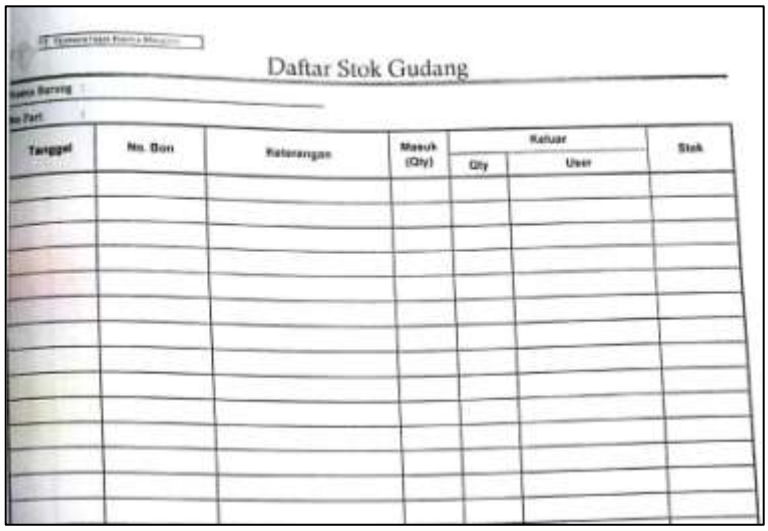
No. *Part* : untuk mencatat nomor barang

Tanggal : kolom tanggal untuk mencatat kapan tanggal aktivitas keluar-masuk barang tersebut terjadi

No. Bon : diambil dari nomor surat jalan

Qty : kolom *qty* masuk, dimaksudkan untuk mencatat berapa banyak barang yang masuk. Kolom *qty* keluar dan *user*, dimaksudkan untuk mencatat berapa banyak barang yang keluar dan ditujukan ke siapa barang tersebut keluar.

Stok : stok digunakan untuk mencatat berapa stok barang yang ada dalam gudang. Untuk lebih jelasnya, gambar dari daftar stok gudang dapat dilihat pada Gambar IV.14.


Daftar Stok Gudang

Tanggal	No. Bon	Keterangan	Masuk (Qty)	Keluar		Stok
				Qty	User	

Gambar IV.14 Daftar Stok
(Sumber: PT Teknikatama Karya Mandiri, 2017)

2. Kartu Tanda Kotak

Kartu tanda yang terdapat diluar kotak yang berisi barang jadi digunakan sebagai informasi jumlah barang jadi yang terdapat dalam kotak tersebut, nama pelanggan untuk barang jadi yang akan dikirim dan nama barang yang ada dalam kotak tersebut. Berikut adalah keterangan yang terdapat dalam kartu tanda kotak:

Customer : nama pelanggan

Product No : nomor barang

Product Name : nama barang

Qty Packing : jumlah barang dalam kotak

PolyBox Type : nama kotak yang dipakai

Berikut ini adalah gambar dari kartu tanda kotak yang dapat dilihat pada Gambar IV.15, IV.16, IV.17, dan IV.18.

* Delivery Tag *	
Customer	Gemala Kempa Daya, PT
Product No	42113-OW110
Product Name	REINFORCEMENT RR
Qty Packing	30 Pcs
PolyBox Type	R.6022

Gambar IV.15 Kartu Tanda Kotak *Reinforcement*
(Sumber: PT Teknikatama Karya Mandiri, 2017)

* Delivery Tag *	
Customer	Inti Ganda Perdana, PT
Product No	42121-BZ270
Product Name	END HOUSING
Qty Packing	20 Pcs
PolyBox Type	R.6022

Gambar IV.16 Kartu Tanda Kotak *End Housing*
(Sumber: PT Teknikatama Karya Mandiri, 2017)

* Delivery Tag *	
Customer	Yazuho, PT
Product No	42114-PS320
Product Name	PIN SHAFT
Qty Packing	20 Pcs
PolyBox Type	R.6022

Gambar IV.17 Kartu Tanda Kotak *Pin Shaft*
(Sumber: PT Teknikatama Karya Mandiri, 2017)

* Delivery Tag *	
Customer	Gemala Kempa Daya, PT
Product No	42120-GS120
Product Name	GUSSET
Qty Packing	50 Pcs
PolyBox Type	R.6022

Gambar IV.18 Kartu Tanda Kotak *Gusset*
(Sumber: PT Teknikatama Karya Mandiri, 2017)

3. Surat Jalan

Jika barang jadi sudah tersimpan di gudang dan waktu pengiriman telah tiba, maka Operator *Warehouse* akan membuat surat jalan untuk pihak *delivery order* agar barang jadi dapat segera dikirim ke pelanggan. Berikut adalah keterangan yang terdapat pada lembar surat jalan:

Kolom kosong	: untuk mencatat tempat dan tanggal dibuatnya surat jalan
Kepada Yth	: untuk mencatat nama *customer*
No	: untuk menuliskan nomor urut
Nama barang	: untuk mencatat nama barang yang akan dikirim
Jumlah	: untuk mencatat jumlah barang yang akan dikirim
Keterangan	: untuk menambahkan pesan lain dalam pengiriman
Jenis kendaraan	: dikirim menggunakan mobil *pick up*
No. Polisi	: mencatat nomor dari kendaraan yang mengirim barang
Penerima	: ditandatangani oleh pelanggan
Pengantar	: ditandatangani oleh *driver*
Pengirim	: ditandatangani oleh pihak PT Teknikatama Karya Mandiri

Surat akan diberikan kepada pihak tagihan Operator *Warehouse* sebanyak satu lembar berwarna putih, kepada pihak administrasi berwarna kuning, kepada pelanggan merah dan biru. Untuk gambar dari surat jalan, dapat dilihat pada Gambar IV.19.

TKM PT. TEKNIKATAMA KARYA MANDIRI
Jigs, Dies, And Machine Parts Specialist

Cikarang, 21 Maret 2018

Kepada Yth :
PT. Inti Ganda Perdana
JAKARTA

SURAT JALAN

NO. 105 / SJ / TKM / III / 2018

Kami kirimkan barang-barang terperinci dibawah ini :

NO.	NAMA BARANG	JUMLAH	KETERANGAN
1	END RR AXLE HOUSING # 42121-82270	600 Pcs	J1487/PO-IGF/0218
			No PO IIAAN : IJB001093
			(Polybag 30 box)
	WO : 014		

Tanda Terima,

Jenis Kendaraan : Mobil
No. Polisi : DB9711AN

Hormat Kami,

Keterangan
Putih : Tagihan
Kuning : Administrasi
Merah : Pelanggan
Biru : Pelanggan

(.....) PENERIMA

(.....) PENGANTAR

(.....) PENGIRIM

FR-MPP-05-Rev 00

Gambar IV.19 Surat Jalan
(Sumber: PT Teknikatama Karya Mandiri, 2017)

4. PO *Customer*

PO *customer* ini digunakan untuk acuan Operator *Warehouse* dalam membuat surat jalan guna mengirim barang jadi ke *customer* PT Teknikatama Karya Mandiri. Berikut adalah keterangan dari *purchase order customer* yang terlampir:

To: ditujukan untuk PT Teknikatama Karya Mandiri

Deliv Date : tanggal kirim

No : urutan

No : nomor PO

Kode : kode barang

Nama barang : nama barang yang dipesan

Jumlah : banyak barang yang dipesan

Keterangan : diisi jika ada keterangan yang ingin ditambahkan

Price: menerangkan harga barang per pcs

Amount: menerangkan jumlah harga

Untuk gambar dari PO *Customer* dapat dilihat pada Gambar IV.20.

PT INTI GANDA PERDANA
JlPegangsaanDua, Blok A1, Pulogadung
P.O Box 1038/JAT Jakarta 13010
Phone: 4602755 (Hunting System) Fax :4602755

PURCHASE ORDER
No. RF1949227

To :Subcon Teknikatamakaryamandiri
Jalan Raya Bekasi Km 66 Sukadanil Cikarang-Selatan
Telp
02189833219

Deliv Date : 21-MAR-2018

No	KODE	NAMA BARANG	JUMLAH	KETERANGAN	PRICE	AMOUNT
1	ACE8BVTY-6227B-00F	END HOOKING 270	6000 PCS		2500	15.000.000

TOTAL 15.000.000

DELIVERY TO PT INTI GANDA PERDANA
PAYMENT 1 MONTH
REMARKS APR 18

SUPPLIER	CHECKED BY	APPROVED BY

Gambar IV.20 PO *Customer*
(Sumber: PT Teknikatama Karya Mandiri, 2017)

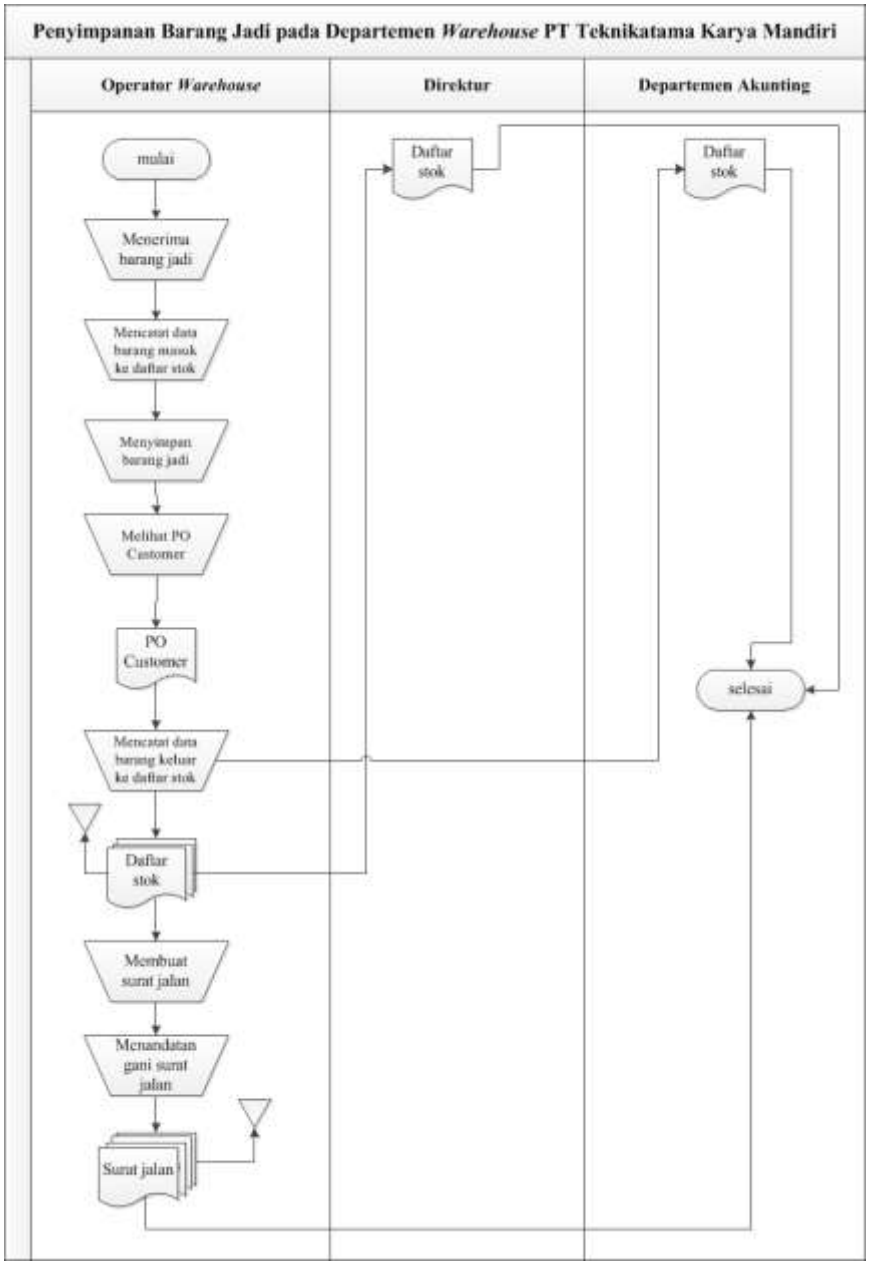
4.6.4 Arus Dokumen yang Berjalan pada Penyimpanan Barang Jadi

Dalam setiap tahap penyimpanan barang jadi ke dalam gudang hingga barang dikeluarkan dari gudang, terdapat dokumen-dokumen yang berjalan dalam proses tersebut. Berikut ini adalah arus dokumen yang berjalan dalam penyimpanan barang jadi:

1. Barang jadi dikirim ke dalam gudang setelah pengecekan oleh *Quality Control* untuk menunggu waktu pengiriman. Ketika barang jadi memasuki gudang, Operator *Warehouse* akan mencatatnya dalam daftar stok.
2. Ketika barang disimpan, terdapat kartu tanda kotak yang terdapat diluar kotak sebagai informasi untuk kotak tersebut seperti nama barang yang ada dalam kotak.
3. Jika waktu pengiriman telah tiba, Operator *Warehouse* membuat surat jalan.
4. Operator *Warehouse* menandatangani surat jalan untuk diserahkan kepada pihak *Delivery Order* untuk dikirim..
5. Barang jadi akan dikeluarkan dari gudang secara acak. Ketika Operator *Warehouse* mengeluarkan barang jadi dari gudang, akan dicatat dalam

daftar stok. Lalu barang diangkut ke dalam mobil *pick up* dan dikirim ke pelanggan.

6. Direktur dan Departemen Akunting melihat daftar stok dari Operator *Warehouse*. Berikut merupakan *flowmap* penyimpanan barang jadi yang berjalan pada Departemen *Warehouse* dapat dilihat pada Gambar IV.20.



Gambar IV.20 *Flowmap* Penyimpanan Barang Jadi yang Berjalan
(Sumber: PT Teknikatama Karya Mandiri, 2017)

4.7 Penggambaran Penyimpanan Barang Jadi dengan UML (*Unified Modeling Language)*

Use case diagram penyimpanan barang jadi yang sedang berjalan pada Departemen *Warehouse* di PT Teknikatama Karya Mandiri adalah sebagai berikut:

1. Definisi Aktor

   Pendefinisian aktor pada *use case diagram* penyimpanan barang jadi yang sedang berjalan pada Departemen *Warehouse* PT Teknikatama Karya Mandiri dapat dilihat pada Tabel IV.2 berikut:

Tabel IV.2. Definisi Aktor Penyimpanan Barang Jadi

No	Aktor	Definisi
1.	Operator *Warehouse*	Mencatat adanya barang jadi yang masuk ke gudang dalam daftar stok, menyimpan barang jadi, jika waktu pengiriman tiba akan membuat surat jalan dan menandatanganinya, lalu mengeluarkan barang jadi dari gudang dan mencatatnya ke dalam daftar stok, dan mengangkut barang jadi ke dalam mobil *pick up* dibantu oleh pihak *Delivery Order*.
2.	Direktur	Melihat daftar stok barang untuk dapat mengetahui aktivitas barang yang keluar atau masuk ke dalam gudang.
3.	Departemen Akunting	Melihat daftar stok barang untuk dapat mengetahui aktivitas barang yang keluar atau masuk ke dalam gudang.

Sumber: Hasil Analisis (2018)

2. Definisi *Usecase*

Pendefinisian *use case* pada *use case diagram* penyimpanan barang jadi pada Departemen *Warehouse* di PT Teknikatama Karya Mandiri dapat dilihat pada Tabel IV.3 berikut:

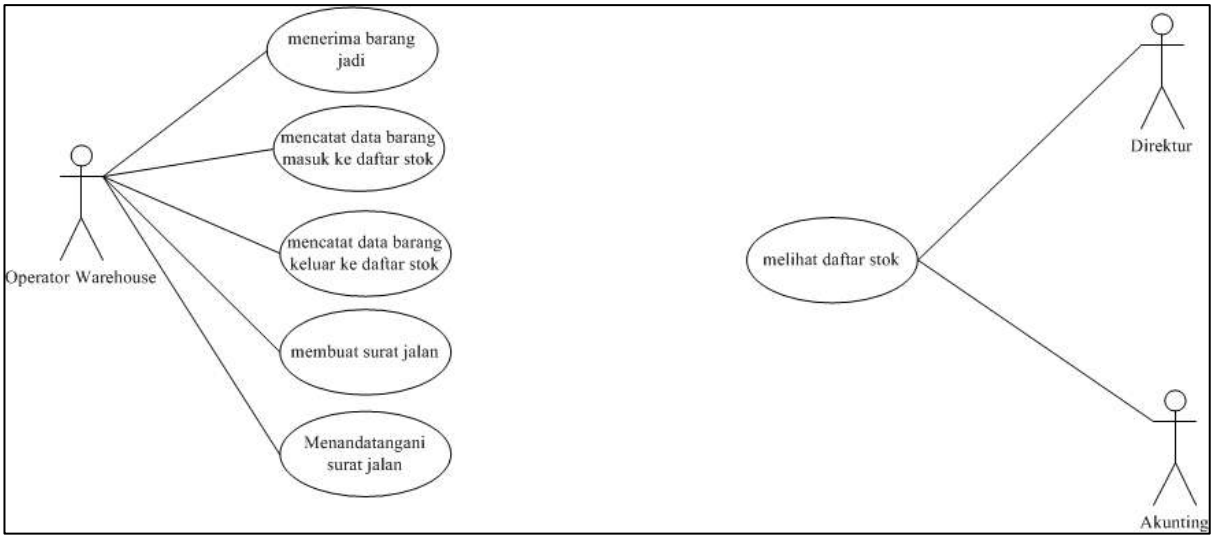
Tabel IV.3 Definisi *Use Case* Penyimpanan Barang Jadi

No	***Use case***	**Deskripsi**
1.	Menerima barang jadi	Operator *Warehouse* menerima barang jadi yang masuk ke dalam gudang dari departemen *Quality Control*.
2.	Mencatat data barang masuk ke daftar stok	Ketika barang masuk, Operator *Warehouse* mencatat data barang yang masuk ke dalam gudang dalam daftar stok.
3.	Mencatat data barang keluar ke daftar stok	Ketika barang keluar, Operator *Warehouse* mencatat data barang yang keluar dari dalam gudang dalam daftar stok.
4.	Membuat surat jalan	Jika sudah mencatat data barang keluar ke dalam daftar stok, Operator *Warehouse* membuat surat jalan untuk diberikan oleh pihak *Delivery Order*.
5.	Melihat daftar stok	Direktur dan Departemen Akunting melihat daftar stok. Direktur dan Departemen Akunting untuk mengetahui aktivitas barang dalam gudang.
6.	Menandatangani surat jalan	Sesudah membuat surat jalan, Operator *Warehouse* menandatangani surat jalan tersebut.

Sumber: Hasil Analisis (2018)

3. *Use Case Diagram*

Use case diagram penyimpanan barang jadi yang sedang berjalan pada Departemen *Warehouse* terdiri dari 3 aktor sebagaimana digambarkan pada Gambar IV.21 di bawah ini.



Gambar IV.21 *Use Case Diagram* Penyimpanan Barang Jadi yang Berjalan
(Sumber: Pengolahan Data, 2018)

4.8 Analisis *Pieces*

Analisis *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*, *Eficiency* 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 variable evaluasi yaitu :

1. *Performance* (Kinerja)
2. *Information* (Informasi)
3. *Economic* (Ekonomi)

4. *Control* (Pengendalian)
5. *Efficiency* (Efisiensi)
6. *Service* (Layanan)

Berikut ini adalah Analisis *Pieces* untuk penyimpanan barang jadi yang ada pada Departemen *Warehouse* PT Teknikatama Karya Mandiri dapat dilihat pada Tabel IV. 4.

Tabel IV.4 Analisis *Pieces* Penyimpanan Barang Jadi

Analisis	**Uraian Masalah**	**Akibat**	**Sebab**	**Solusi**
Perfomance (kinerja)	Daftar stok tidak dicatat dengan rutin oleh Operator *Warehouse* sehingga menyebabkan minimnya informasi detail barang yang keluar/masuk dalam gudang.	Arus barang jadi yang keluar/masuk dari gudang kurang terorganisir dengan baik yang mengakibatkan penumpukan barang di gudang.	Operator *Warehouse* seringkali lupa mencatat data barang yang masuk/keluar dari gudang. Dan adanya dokumen daftar stok yang hilang.	Pencatatan daftar stok dilakukan secara terkomputerisasi dan disimpan ke dalam basis data/ *database*.
Information (informasi)	Belum adanya pembuatan laporan bulanan.	Kurangnya informasi bagi Departemen Akunting dan Direktur untuk melihat perkembangan barang yang masuk dan keluar dari gudang.	Dikarenakan daftar stok yang tidak dicatat dengan baik sehingga minimnya data untuk membuat laporan.	Pembuatan laporan bulanan dibuat secara terkomputerisasi. Data yang ada diambil berdasarkan data stok, data transaksi

Analisis	Uraian Masalah	Akibat	Sebab	Solusi
				barang

Tabel IV.4 Analisis *Pieces* Penyimpanan Barang Jadi (lanjutan)

Analisis	Uraian Masalah	Akibat	Sebab	Solusi
				keluar dan masuk.
Economics (Biaya)	-	-	-	-
Control (Kendali)	Barang dikeluarkan dari gudang secara acak.	Barang yang diproduksi pada jam pertama mengalami penumpukan terlalu lama di gudang sehingga dikhawatirkan dapat merusak barang.	Tidak adanya teknik pengambilan barang, sehingga barang jadi yang diambil dipilih secara acak.	Menerapkan teknik pengambilan barang jadi dengan *First In First Out* Sehingga Operator *Warehouse* dapat mengetahui barang jadi mana saja yang harus dikeluarkan dari dalam gudang.
Efficiency (Efisiensi)	-	-	-	-
Service (Layanan)	-	-	-	-

Sumber: Hasil Penelitian (2018)

Pada masalah dibidang ekonomi tidak cukup data untuk meneliti permasalahannya. Juga tidak ditemukan data yang akurat untuk meneliti masalah

pada bidang efisiensi. Untuk bidang *service* tidak ditemukan permasalahan dari sebab, akibat, uraian masalah dan solusinya.

Kesimpulan dengan adanya analisis *pieces* ini, permasalahan yang didapat yaitu dalam hal kinerja atau *performance* daftar stok tidak dicatat dengan rutin oleh Operator *Warehouse* sehingga menyebabkan minimnya informasi detail barang yang keluar/masuk dalam gudang, dalam hal informasi atau *information* adalah belum adanya pembuatan laporan bulanan. Serta dalam hal kendali atau *control* yaitu barang dikeluarkan dari gudang secara acak. Permasalahan-permasalahan ini akan digunakan untuk dasar acuan dalam membuat sistem informasi penyimpanan barang jadi guna membantu Operator *Warehouse* untuk pencatatan aktivitas keluar/masuknya barang, pembuatan laporan bulanan, dan dapat mengetahui barang mana saja yang harus dikeluarkan.

BAB V
ANALISIS DAN PEMBAHASAN

5.1 Analisis Sistem Informasi Usulan

Analisis sistem informasi usulan untuk penyimpanan barang jadi pada Departemen *Warehouse* menggunakan pemodelan sistem UML, (*Unified Modeling Language*) yang meliputi pembuatan *use case diagram, activity diagram, sequence diagram, class diagram, entity relationship diagram, windows navigation diagram* dan pembuatan kamus data. Tahapan analisis ini akan memberikan gambaran mengenai aliran informasi dan data pada sistem informasi yang akan dibangun.

5.1.1 Analisis Kebutuhan Sistem

Berdasarkan hasil analisis sistem yang berjalan, maka adanya analisis kebutuhan sistem untuk sistem usulan yang khusus untuk menangani penyimpanan barang jadi pada Departemen *Warehouse*. Tabel V.1 menjelaskan daftar kebutuhan sistem informasi penyimpanan barang jadi pada PT Teknikatama Karya Mandiri.

Tabel V.1 Analisis Kebutuhan Sistem

Identifikasi Masalah	**Kebutuhan Pengguna**	**Kebutuhan Sistem**	**Uraian**	
			Fungsional	**Non Fungsional**
daftar stok masih menggunakan *form* kertas dan tidak dicatat dengan rutin oleh Operator	Informasi daftar stok dapat dilakukan secara terkomputerisasi dan disimpan ke dalam *database.*	Menggunakan sistem informasi berbasis *web* dan adanya menu untuk menyimpan	Sistem menyediakan menu stok barang yang mana data tersebut diambil dari	Transaksi keluar dan masuk wajib diisi.

Tabel V.1 Analisis Kebutuhan Sistem (lanjutan)

Identifikasi	**Kebutuhan**	**Kebutuhan**	**Uraian**

Masalah	Pengguna	Sistem	Fungsional	Non Fungsional
Warehouse.		stok barang.	transaksi keluar dan masuk.	
Belum adanya pembuatan laporan bulanan.	Mencetak laporan bulanan melalui sistem berbasis *web*.	Menggunakan sistem informasi berbasis *web* dan adanya menu untuk data laporan yang berisikan transaksi barang keluar/masuk, dan stok.	Adanya menu data laporan dan tombol cetak untuk melihat data lalu mencetak data sebagai PDF.	
Belum adanya pencatatan detail transaksi barang keluar/masuk dalam daftar stok.	Adanya detail transaksi barang keluar/masuk secara terkomputerisasi dan disimpan ke dalam *database*.	Menggunakan sistem informasi berbasis *web* dan adanya menu untuk mencatat transaksi barang.	Sistem menyediakan *form* transaksi barang keluar/masuk untuk diisi dilengkapi dengan aksi simpan, ubah, hapus, cari.	Transaksi barang harus diisi untuk memudahkan informasi untuk daftar stok.

Sumber: Hasil Analisis (2018)

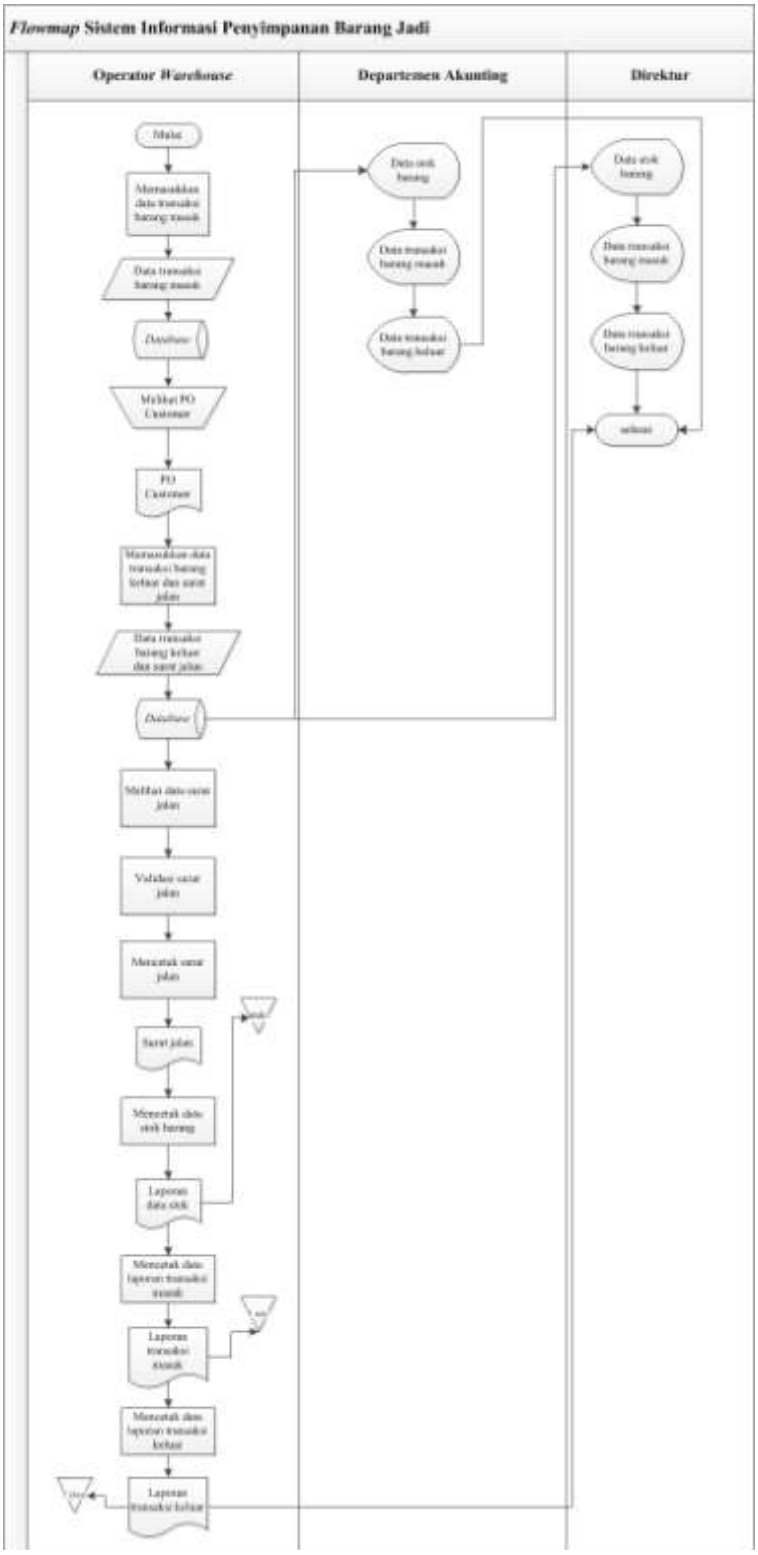
5.1.2 Perancangan *Flowmap* Sistem Usulan

Urutan prosedur dan *flowmap* sistem informasi penyimpanan barang jadi yang diusulkan sebagai berikut:

7. Dimulai dengan Operator *Warehouse* memasukkan data transaksi barang masuk ketika adanya barang jadi yang masuk ke dalam gudang.

8. Data *input* transaksi barang jadi siap disimpan ke dalam *database.*
9. Data transaksi barang jadi disimpan dalam *database*.
10. Sistem memperbaharui daftar stok.
11. Operator *Warehouse* memasukkan data transaksi barang keluar dan surat jalan ketika adanya barang yang keluar dari dalam gudang.
12. Data *input* transaksi barang keluar dan surat jalan siap disimpan ke dalam *database.*
13. Data-data yang telah di*input*, akan ditampilkan kepada Direktur dan Departemen Akunting.
14. Data transaksi barang keluar dan surat jalan tersimpan dalam *database*.
15. Sistem memperbaharui daftar stok.
16. Operator *Warehouse* melihat data surat jalan.
17. Operator *Warehouse* melakukan validasi surat jalan.
18. Operator *Warehouse* mencetak surat jalan lalu menghasilkan dokumen surat jalan.
19. Operator *Warehouse* mencetak data stok barang, dan transaksi barang keluar/masuk lalu menghasilkan dokumen laporan. Laporan akan diserahkan ke bagian Akunting dan Direktur bila diminta.
20. Direktur dan Departemen Akunting dapat melihat data stok, transaksi masuk/keluar untuk mengetahui arus barang jadi yang keluar dari gudang.
21. Selesai.

Flowmap sistem informasi penyimpanan barang jadi yang diusulkan dapat dilihat pada Gambar V.1.



Gambar V.1 *Flowmap* Sistem Informasi Penyimpanan Barang Jadi Usulan
Sumber: Hasil Analisis (2018)

5.1.3 *Use Case Diagram* Usulan

Berdasarkan Dennis *et al* (2015), *use case diagram* adalah untuk menggambarkan kebutuhan bisnis untuk sistem dan menggambarkan interaksi antara sistem dan lingkungannya. Dalam *use case diagram* ini dilakukan oleh tiga *actor*. Berdasarkan analisis yang telah dilakukan, maka *use case diagram* sistem informasi penyimpanan barang jadi usulan dapat dilihat pada Gambar V.2 sebagai berikut ini:

Gambar V.2 *Use Case Diagram* Sistem Informasi Penyimpanan Barang Jadi Usulan

Sumber: Hasil Analisis (2018)

Penjelasan *use case diagram* sistem informasi penyimpanan barang jadi pada Departemen *Warehouse* usulan dijelaskan sebagai berikut ini:

1. Definisi Aktor

   Pendefinisian aktor pada *use case diagram* sistem informasi penyimpanan barang jadi usulan dijelaskan pada Tabel V.2.

Tabel V.2 Definisi Aktor Pada *Use Case* Diagram

No	Aktor	Deskripsi
1.	Operator *Warehouse*	Operator *Warehouse* merupakan aktor yang menerima barang jadi, memasukkan atau *input* transaksi barang jadi yang masuk ke dalam gudang. Menyimpan barang jadi pada kotak atau *box* yang sudah ada, memasukkan atau *input* transaksi barang keluar dan surat jalan, dan mencetak laporan daftar stok barang dan data transaksi. Untuk dapat mengakses fasilitas-fasilitas tersebut, Operator *Warehouse* diharuskan melakukan proses *login* terlebih dahulu.
2.	Direktur	Direktur merupakan aktor yang dapat melihat laporan data stok barang dan data transaksi. Untuk dapat melihat fasilitas-fasilitas tersebut, Direktur diharuskan melakukan proses *login* terlebih dahulu.
3.	Departemen Akunting	Departemen Akunting merupakan aktor yang melihat laporan data stok barang dan data transaksi. Untuk dapat melihat fasilitas-fasilitas tersebut, Departemen Akunting diharuskan melakukan proses *login* terlebih dahulu.

Sumber: Hasil Analisis (2018)

2. Definisi *Use Case*

Pendefinisian *use case* pada *use case diagram* sistem informasi penyimpanan barang jadi pada Departemen *Warehouse* usulan dijelaskan pada tabel-tabel dibawah ini yang berisikan deskripsi *use case* diagram.

Tabel V.3 Definisi *Use Case* Mengelola Data *Master*

Nama *Use Case*	**Mengelola Data *Master***
Nama Aktor	Operator *Warehouse*

Deskripsi *Use Case*	Proses mengelola data *master* terdiri dari: 1. Mengelola data *master* pengguna, Operator *Warehouse* dapat membuat akun pengguna baru serta mencari,

Tabel V.3 Definisi *Use Case* Mengelola Data *Master* (lanjutan)

Nama *Use Case*	**Mengelola Data *Master***
	dapat menambah, mencari, mengubah dan menghapus data *master* pengguna. 2. Mengelola data *master* pelanggan, Operator Warehouse dapat menambah, mencari, mengubah dan menghapus data *master* pelanggan. 3. Mengelola data *master* barang jadi, Operator Warehouse dapat menambah, mencari, mengubah dan menghapus data *master* barang jadi.
Hubungan	*Generalization*: Mengelola data *master* pengguna, data *master* pelanggan dan data *master* barang jadi.
Aliran *Use Case*	1. Operator *Warehouse* telah sukses melakukan *login*. 2. Sistem menampilkan menu utama. 3. Operator *Warehouse* memilih menu Data *Master*. 4. Operator *Warehouse* memilih Submenu Data *Master*. 5. Sistem menampilkan halaman data *master* yang dipilih. 6. Operator *Warehouse* dapat memilih aksi cari, simpan, ubah dan hapus. 7. Jika memilih masukan kata kunci kemudian sistem akan mencari data. 8. Sistem menampilkan data sesuai kata kunci. 9. Jika memilih Tambah Data kemudian klik simpan maka sistem akan menyimpan data ke basis data (*database*). 10. Jika memilih Ubah, sistem akan menampilkan *form* ubah data lalu aktor mengubah data yang diinginkan lalu klik Simpan. 11. Jika memilih Hapus, sistem akan menghapus data.
Persyaratan	Operator *Warehouse* telah melakukan *Login* terlebih dahulu.

Sumber: Hasil Analisis (2018)

Tabel V.4 Definisi *Use Case* Memasukkan Data Transaksi Barang Masuk

Nama *Use Case*	**Memasukkan Data Transaksi Barang Masuk**
Nama Aktor	Operator *Warehouse*

Tabel V.4 Definisi *Use Case* Memasukkan Data Transaksi Barang Masuk (lanjutan)

Deskripsi *Use Case*	Proses Memasukkan Data Transaksi Barang Masuk
Hubungan	*Association*: Operator *Warehouse*.
Aliran *Use Case*	1. Operator *Warehouse* telah sukses melakukan *login*. 2. Sistem menampilkan Menu Utama. 3. Operator *Warehouse* memilih menu Transaksi Barang. 4. Operator *Warehouse* memilih Submenu Transaksi Barang Masuk. 5. Sistem menampilkan halaman Submenu Data Transaksi Barang Masuk. 6. Operator *Warehouse* memasukkan Data Barang Jadi Masuk Detail, klik Simpan. 7. Sistem menampilkan data Transaksi Barang Masuk Detail. 8. Jika Operator *Warehouse* memilih aksi Hapus, Sistem menghapus transaksi barang masuk detail. 9. Jika Operator *Warehouse* memilih aksi keluarkan barang, Sistem akan menampilkan *form* transaksi detail barang keluar. 10. Jika Operator *Warehouse* memilih aksi Detail, maka Sistem akan menampilkan data detail Transaksi Barang Masuk. 11. Sistem memperbaharui basis data Transaksi Barang Masuk. 12. Barang Jadi yang masuk ke gudang pada jam kerja awal, adalah barang yang pertama kali keluar untuk dikirim.
Persyaratan	Operator *Warehouse* telah melakukan *Login* terlebih dahulu.

Sumber: Hasil Analisis (2018)

Tabel V.5 Definisi *Use Case* Melakukan *Login*

Nama *Use Case*	**Melakukan *Login***

Nama Aktor	Operator *Warehouse*, Direktur, Departemen Akunting.
Deskripsi *Use Case*	Proses *login* yang dilakukan untuk menjaga akses sistem

Tabel V.5 Definisi *Use Case* Melakukan *Login* (lanjutan)

Nama *Use Case*	**Melakukan *Login***
Deskripsi *Use Case*	dari pihak yang tidak diizinkan untuk menggunakan sistem informasi ini. Proses *Login* merupakan proses *input username* dan *password* oleh pengguna yang ingin masuk ke dalam sistem. Sistem nantinya akan menvalidasi data yang dimasukkan. Apabila pengguna terdaftar maka dapat menggunakan sistem sesuai dengan hak aksesnya.
Hubungan	*Association*: Operator *Warehouse*, Direktur, Departemen Akunting.
Aliran *Use Case*	1. Pengguna membuka aplikasi 2. Sistem menampilkan menu *login*. 3. Pengguna mengisi *username* dan *password* di menu *login*. 4. Sistem menvalidasi data yang masuk. Apabila data tidak *valid*,maka muncul pesan gagal *login* dan *user* tersebut tidak dapat menggunakan sistem. 5. Apabila data *valid*, maka sistem akan menampilkan menu utama sesuai hak akses.

Sumber: Hasil Analisis (2018)

Tabel V.6 Definisi *Use Case* Memasukkan Data Transaksi Barang Keluar dan Surat Jalan

Nama *Use Case*	**Memasukkan Data Transaksi Barang Keluar dan Surat Jalan**
Nama Aktor	Operator *Warehouse*
Deskripsi *Use Case*	Proses Memasukkan Data Transaksi Barang Keluar dan Surat Jalan.
Hubungan	*Association:* Operator *Warehouse*
Aliran *Use Case*	1. Operator *Warehouse* telah sukses melakukan *login*. 2. Sistem menampilkan Menu Utama.

	3. Operator *Warehouse* memilih menu Transaksi Barang. 4. Operator *Warehouse* memilih Submenu Transaksi Barang

Tabel V.6 Definisi *Use Case* Memasukkan Data Transaksi Barang Keluar dan Surat Jalan (lanjutan)

Nama *Use Case*	**Memasukkan Data Transaksi Barang Keluar dan Surat Jalan**
Aliran *Use Case*	Masuk. 5. Sistem menampilkan halaman Submenu Transaksi Barang Masuk. 6. Operator *Warehouse* klik aksi keluarkan barang. 7. Sistem menampilkan *form* transaksi detail barang keluar. 8. Operator *Warehouse* memasukkan Data Barang Jadi Keluar Detail dan Surat Jalan, klik Simpan. 9. Jika Operator *Warehouse* memilih aksi Hapus, Sistem menghapus data. 10. Sistem memperbaharui basis data Transaksi Barang Keluar. 11. Jika klik Detail, akan menampilan informasi data barang keluar detail. 12. Barang yang pertama kali keluar adalah barang yang pertama kali di*input* pada Transaksi Barang Masuk.
Persyaratan	Operator *Warehouse* telah melakukan *Login* terlebih dahulu.

Sumber: Hasil Analisis (2018)

Tabel V.7 Definisi *Use Case* Mencetak Laporan Data Stok

Nama *Use Case*	**Mencetak Laporan Data Stok**
Nama Aktor	Operator *Warehouse*
Deskripsi *Use Case*	Proses mencetak laporan data stok
Hubungan	*Association*: Operator *Warehouse*.
Aliran *Use Case*	1. Operator *Warehouse* sukses melakukan *login*. 2. Operator *Warehouse* memilih menu Stok Barang Jadi.

	3. Sistem menampilkan halaman Stok Barang Jadi. 4. Jika Operator *Warehouse* memilih aksi Cetak, Laporan Stok Barang Jadi akan tercetak dalam bentuk PDF.

Tabel V.7 Definisi *Use Case* Mencetak Laporan Data Stok (lanjutan)

Nama *Use Case*	**Mencetak Laporan Data Stok**
Persyaratan	Operator *Warehouse* telah melakukan *login* terlebih dahulu.

Sumber: Hasil Analisis (2018)

Tabel V.8 Definisi *Use Case* Mencetak Laporan Transaksi Masuk

Nama *Use Case*	**Mencetak Laporan Transaksi Masuk**
Nama Aktor	Operator *Warehouse*.
Deskripsi *Use Case*	Proses mencetak laporan transaksi masuk
Hubungan	*Association*: Operator *Warehouse*.
Aliran *Use Case*	1. Operator *Warehouse*, sukses melakukan *login*. 2. Sistem menampilkan menu utama. 3. Operator *Warehouse*, memilih menu Data Laporan. 4. Operator *Warehouse*, memilih sub menu Transaksi Masuk. 5. Operator *Warehouse*, mencari data berdasarkan bulan. 6. Klik cetak untuk mencetaknya dalam format PDF.
Persyaratan	Operator *Warehouse* telah melakukan *login* terlebih dahulu.

Tabel V.9 Definisi *Use Case* Mencetak Laporan Transaksi Keluar

Nama *Use Case*	**Mencetak Laporan Transaksi Keluar**
Nama Aktor	Operator *Warehouse*.
Deskripsi *Use Case*	Proses mencetak data transaksi keluar
Hubungan	*Association*: Operator *Warehouse*.
Aliran *Use Case*	1. Operator *Warehouse* sukses melakukan *login*. 2. Sistem menampilkan menu utama. 3. Operator *Warehouse* memilih menu Data Laporan. 4. Operator *Warehouse* memilih sub menu Transaksi

	Keluar. 5. Operator *Warehouse* mencari data berdasarkan bulan. 6. Klik cetak untuk mencetaknya dalam format PDF.
Persyaratan	Operator *Warehouse* melakukan *login*

Sumber: Hasil Analisis (2018)

Tabel V.10 Definisi *Use Case* Validasi Surat Jalan

Nama *Use Case*	**Validasi Surat Jalan**
Nama Aktor	Operator *Warehouse*
Deskripsi *Use Case*	Proses validasi surat jalan
Hubungan	*Association*: Operator *Warehouse*.
Aliran *Use Case*	1. Operator *Warehouse* sukses melakukan *login*. 2. Sistem menampilkan menu utama. 3. Operator *Warehouse* membuka halaman transaksi barang keluar. 4. Operator *Warehouse* klik tombol surat jalan. 5. Operator *Warehouse* klik tombol kirim barang.
Persyaratan	Operator *Warehouse* melakukan *login*

Sumber: Hasil Analisis (2018)

Tabel V.11 Definisi *Use Case* Melihat Data Transaksi Masuk

Nama *Use Case*	**Melihat Data Transaksi Masuk**
Nama Aktor	Direktur, Departemen Akunting.
Deskripsi *Use Case*	Proses melihat data transaksi masuk
Hubungan	*Association*: Direktur, Departemen Akunting.
Aliran *Use Case*	1. Direktur dan Departemen Akunting sukses melakukan *login*. 2. Sistem menampilkan menu utama. 3. Direktur dan Departemen Akunting memilih menu Data Laporan. 4. Direktur dan Departemen Akunting memilih sub menu Transaksi Masuk. 5. Direktur dan Departemen mencari data berdasarkan bulan.
Persyaratan	Direktur dan Departemen telah melakukan *login* terlebih

	dahulu.

Tabel V.12 Definisi *Use Case* Melihat Data Transaksi Keluar

Nama *Use Case*	**Melihat Data Transaksi Keluar**
Nama Aktor	Direktur, Departemen Akunting.

Tabel V.12 Definisi *Use Case* Melihat Data Transaksi Keluar (lanjutan)

Nama *Use Case*	**Melihat Data Transaksi Keluar**
Deskripsi *Use Case*	Proses melihat data transaksi keluar
Hubungan	*Association*: Direktur, Departemen Akunting.
Aliran *Use Case*	1. Direktur dan Departemen Akunting sukses melakukan *login*. 2. Sistem menampilkan menu utama. 3. Direktur dan Departemen Akunting memilih menu Data Laporan. 4. Direktur dan Departemen Akunting memilih sub menu Transaksi Keluar. 5. Operator *Warehouse*, Direktur dan Departemen mencari data berdasarkan bulan.
Persyaratan	Direktur dan Departemen telah melakukan *login* terlebih dahulu.

Sumber: Hasil Analisis (2018)

Tabel V.13 Definisi *Use Case* Melihat Data Stok Barang

Nama *Use Case*	**Melihat Data Stok Barang**
Nama Aktor	Direktur, Departemen Akunting.
Deskripsi *Use Case*	Proses melihat data stok barang
Hubungan	*Association*: Direktur, Departemen Akunting.
Aliran *Use Case*	1. Direktur dan Departemen Akunting sukses melakukan *login*. 2. Sistem menampilkan menu utama. 3. Direktur dan Departemen Akunting memilih menu Stok Barang Jadi.
Persyaratan	Direktur dan Departemen telah melakukan *login* terlebih

	dahulu.

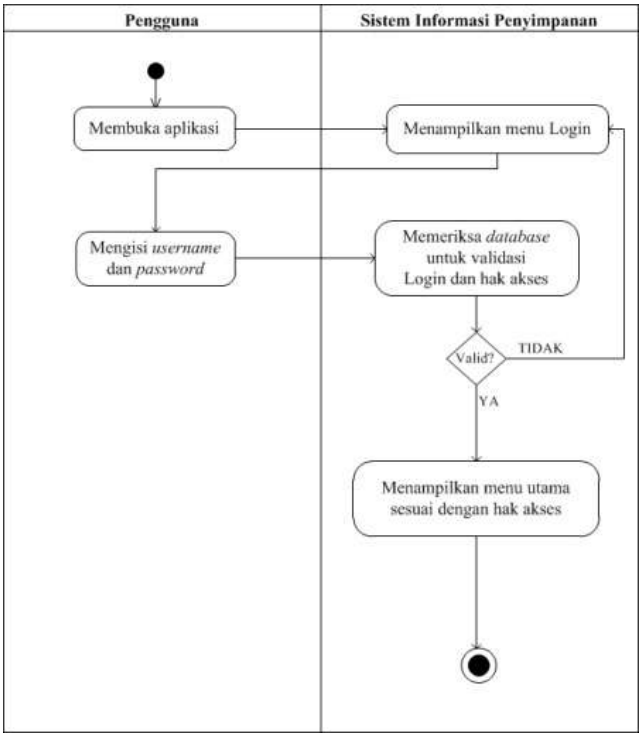
Sumber: Hasil Analisis (2018)

5.1.4 *Activity Diagram*

Activity diagram digunakan untuk menggambarkan aliran aktivitas dari setiap *use case diagram* pada sistem informasi penyimpanan barang jadi. *Activity diagram* masing-masing *use case* digambarkan pada gambar-gambar berikut ini:

1. *Activity Diagram* Melakukan *Login*

   Activity diagram berikut ini menggambarkan aktivitas saat melakukan *login.* Pengguna mengisi *username* dan *password* untuk dapat masuk ke sistem. *Activity diagram* melakukan *login* digambarkan pada Gambar V.3.



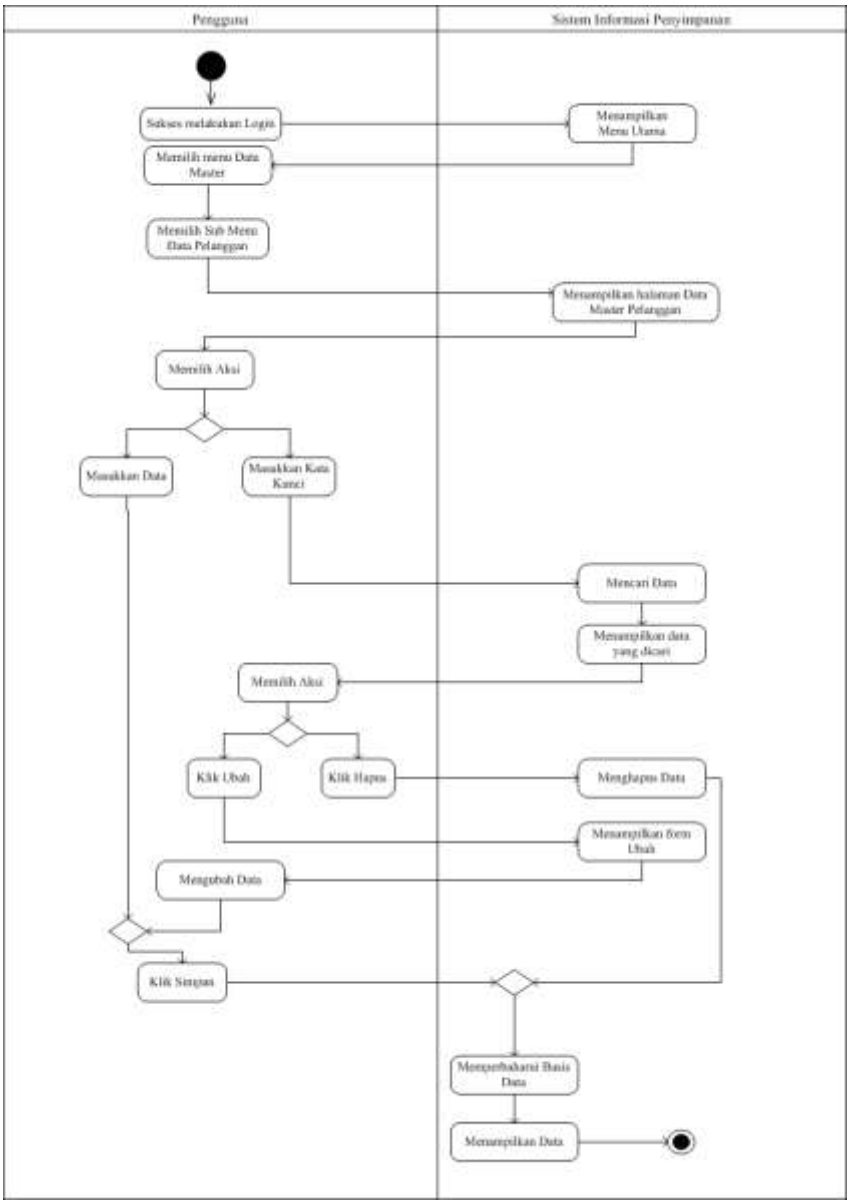
Gambar V.3 *Activity Diagram* Melakukan *Login*
Sumber: Hasil Analisis (2018)

2. *Activity Diagram* Proses Mengelola Data *Master*

   Activity diagram berikut ini menggambarkan aktivitas yang dilakukan saat akan melakukan mengelola data *master*, dimana Operator *Warehouse* dapat menambah, mencari, mengubah dan menghapus data *master.*

Adapun data *master* yang ada adalah data pengguna, data pelanggan, dan data barang.

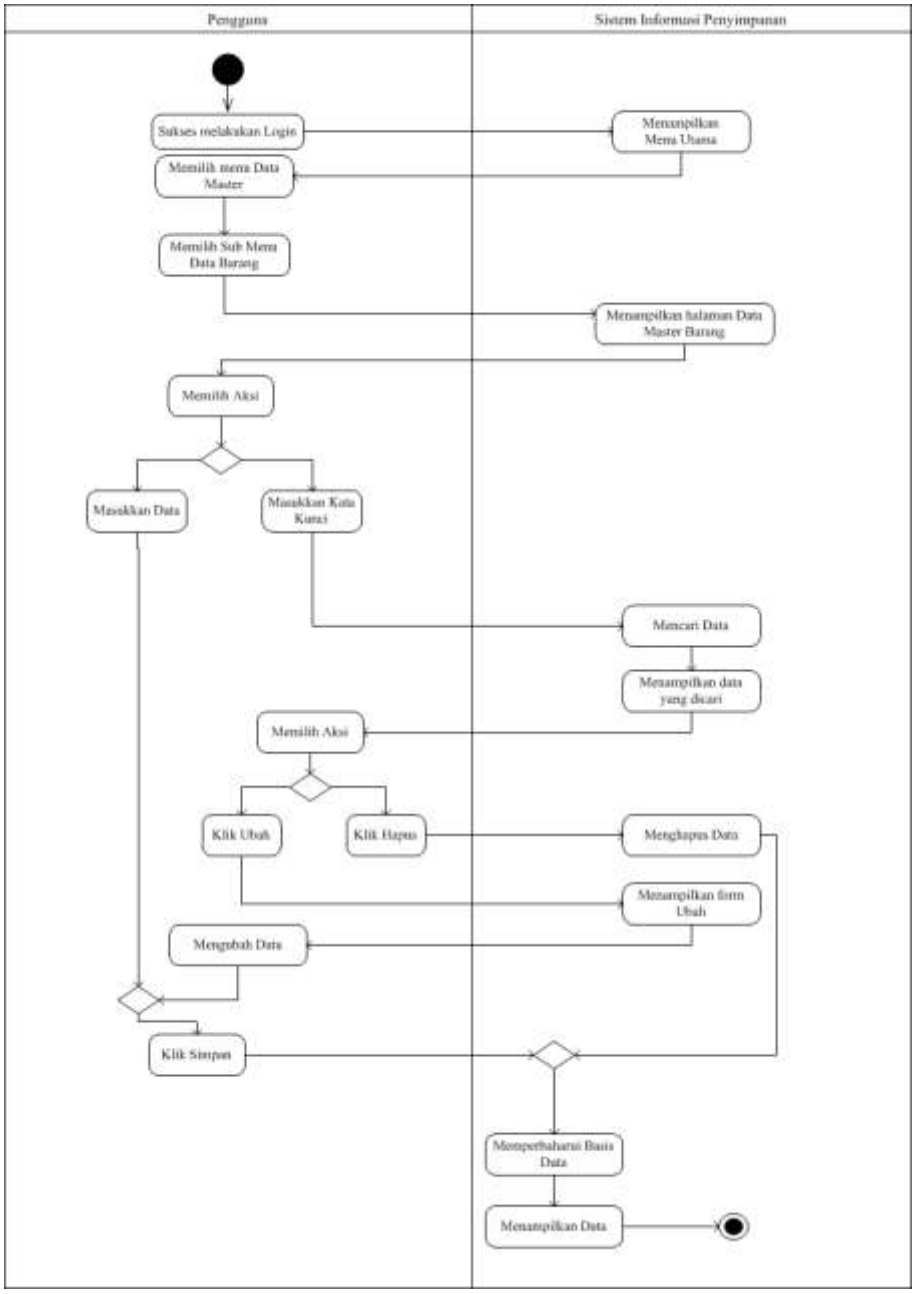
a. *Activity diagram* proses mengelola data *master* pelanggan digambarkan pada Gambar V.4.



Gambar V.4 *Activity Diagram* Mengelola Data *Master* Pelanggan

Sumber: Hasil Analisis (2018)

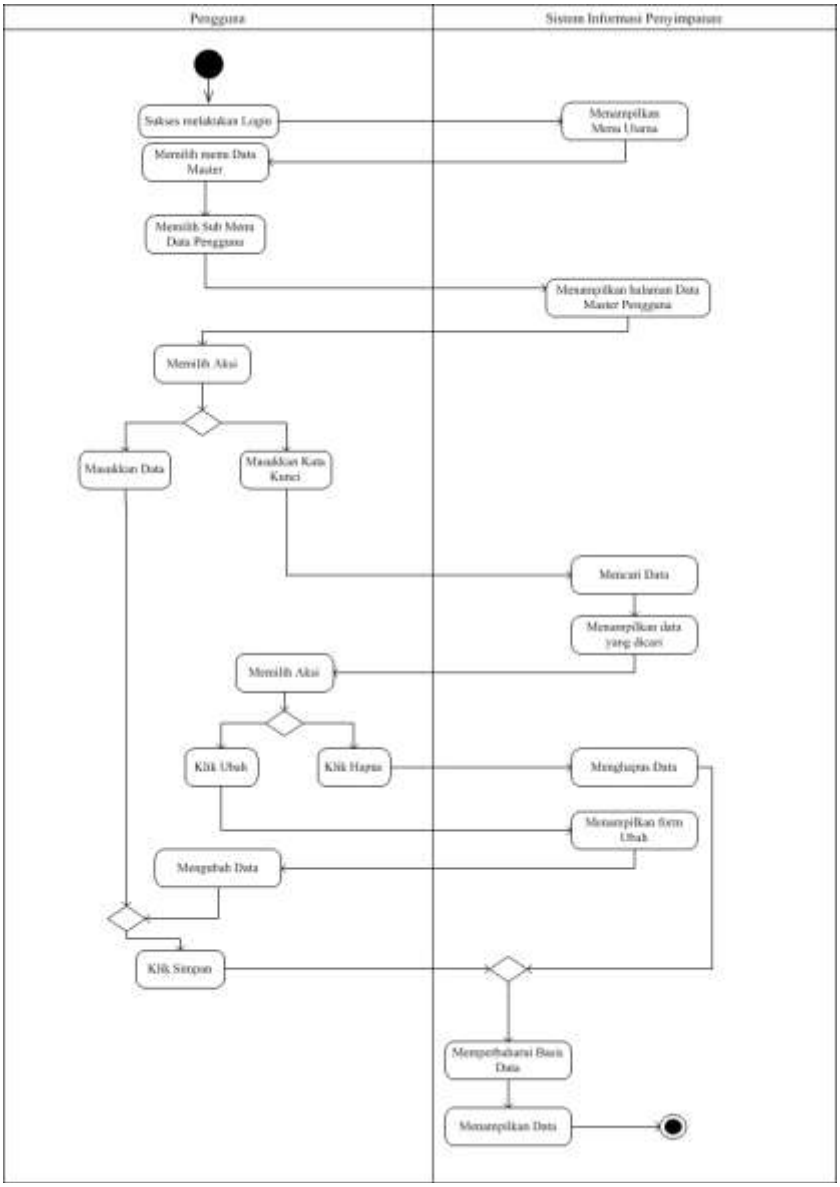
b. *Activity diagram* proses mengelola data *master* barang digambarkan pada Gambar V.5.



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

Sumber: Hasil Analisis (2018)

c. *Activity diagram* proses mengelola data *master* pengguna digambarkan pada Gambar V.6.

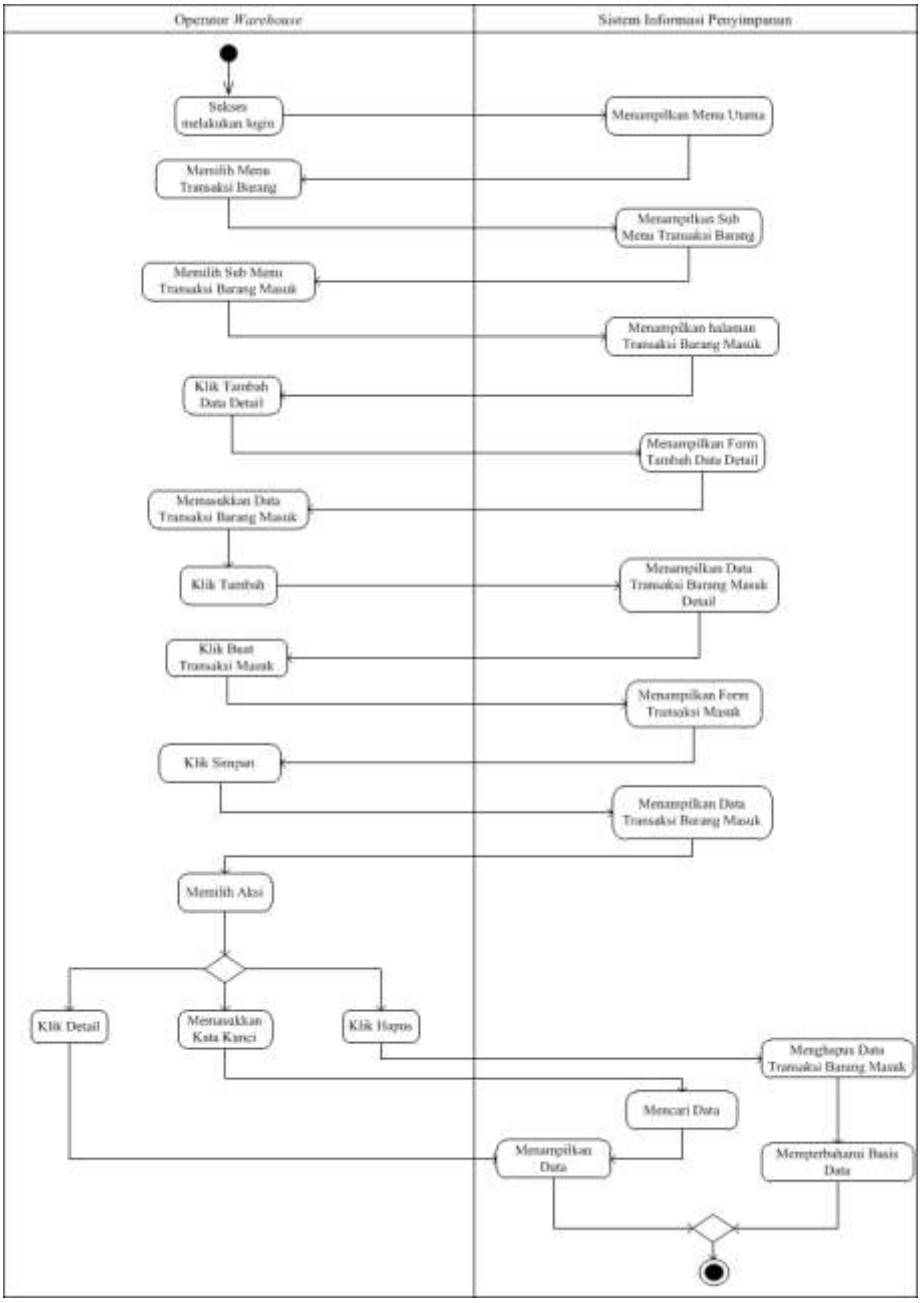


Gambar V.6 *Activity Diagram* Mengelola Data *Master* Pengguna

Sumber: Hasil Analisis (2018)

3. *Activity Diagram* Memasukkan Data Transaksi Barang Masuk

Activity diagram berikut ini menggambarkan aktivitas yang dilakukan saat memasukkan data Transaksi Barang Masuk. Operator *Warehouse* memasukkan data Transaksi Barang Masuk ketika adanya barang yang masuk ke dalam gudang. *Activity diagram* proses memasukkan data Transaksi Barang Masuk digambarkan pada Gambar V.7.



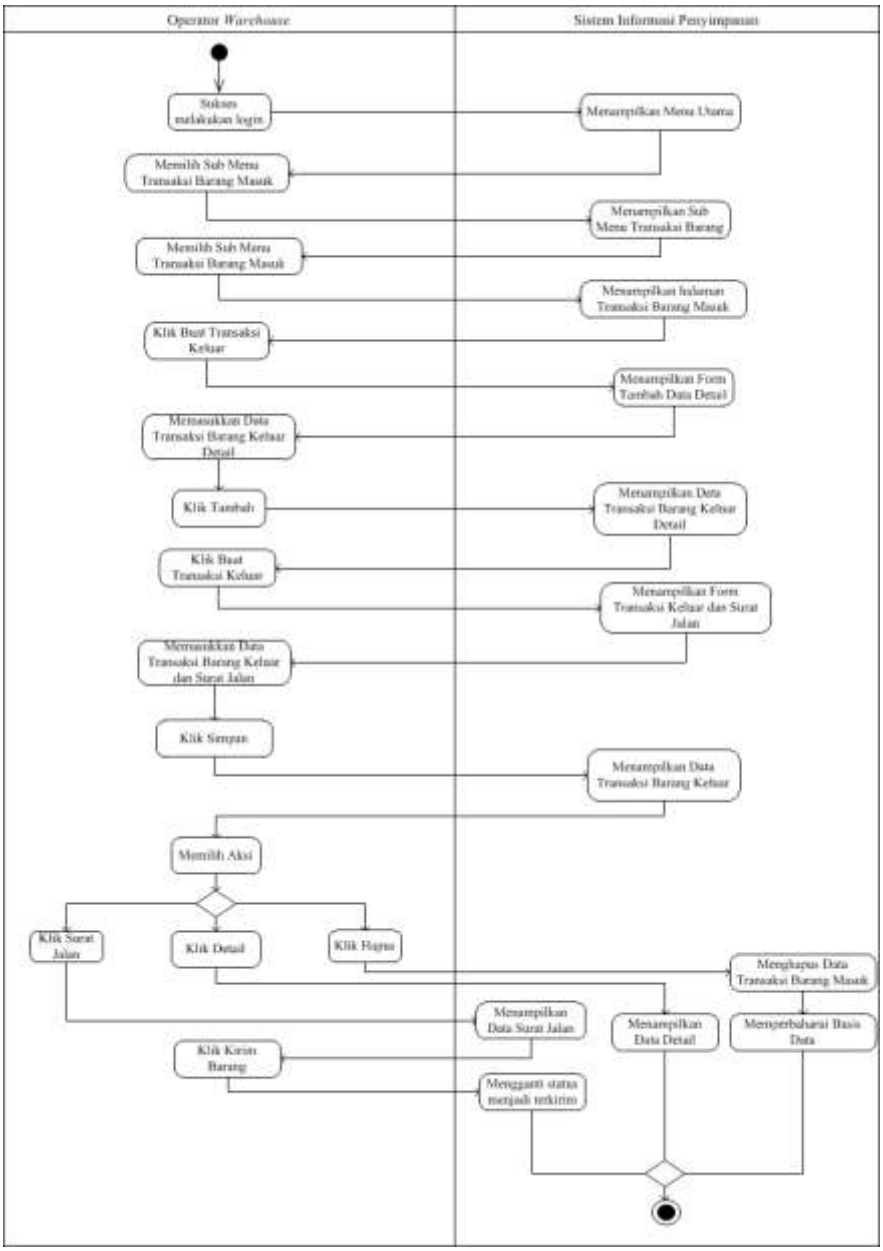
Gambar V.7 *Activity Diagram* Memasukkan Data Transaksi Barang Masuk

Sumber: Hasil Analisis (2018)

4. *Activity Diagram* Memasukkan Data Transaksi Barang Keluar dan Surat Jalan

Activity diagram berikut ini menggambarkan aktivitas yang dilakukan saat memasukkan data Transaksi Barang Keluar sekaligus membuat data Surat Jalan. Operator *Warehouse* akan memasukkan data apabila adanya barang

yang keluar dari gudang. *Activity diagram* proses memasukkan data Transaksi Barang Keluar dan Surat Jalan digambarkan pada Gambar V.8.

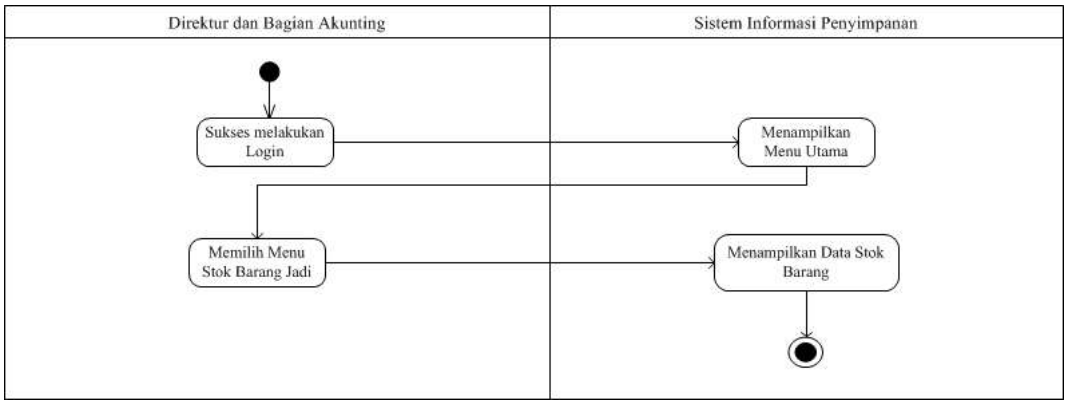


Gambar V.8 *Activity Diagram* Memasukkan Data Transaksi Barang Keluar dan Surat Jalan

Sumber: Hasil Analisis (2018)

5. *Activity Diagram* Melihat Data Stok Barang

Activity diagram berikut ini menggambarkan aktivitas yang dilakukan saat Direktur dan Bagian Akunting akan melihat data Stok Barang. *Activity diagram* melihat stok barang digambarkan pada Gambar V.9.

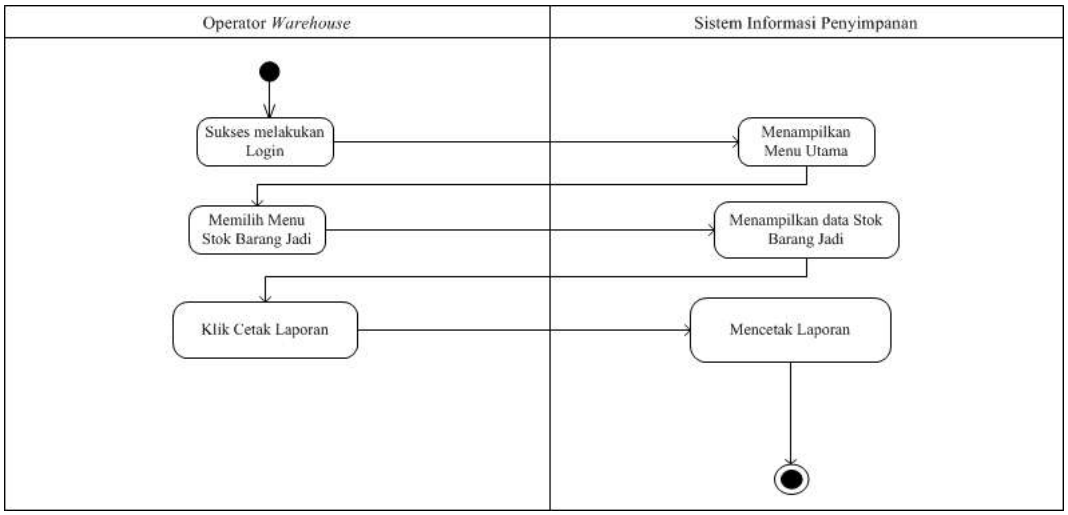


Gambar V.9 *Activity Diagram* Melihat Data Stok Barang

Sumber: Hasil Analisis (2018)

6. *Activity Diagram* Mencetak Laporan Data Stok

Activity diagram berikut ini menggambarkan aktivitas yang dilakukan saat Operator *Warehouse* ingin mencetak data Stok Barang. *Activity diagram* proses mencetak laporan data stok digambarkan pada Gambar V.10.

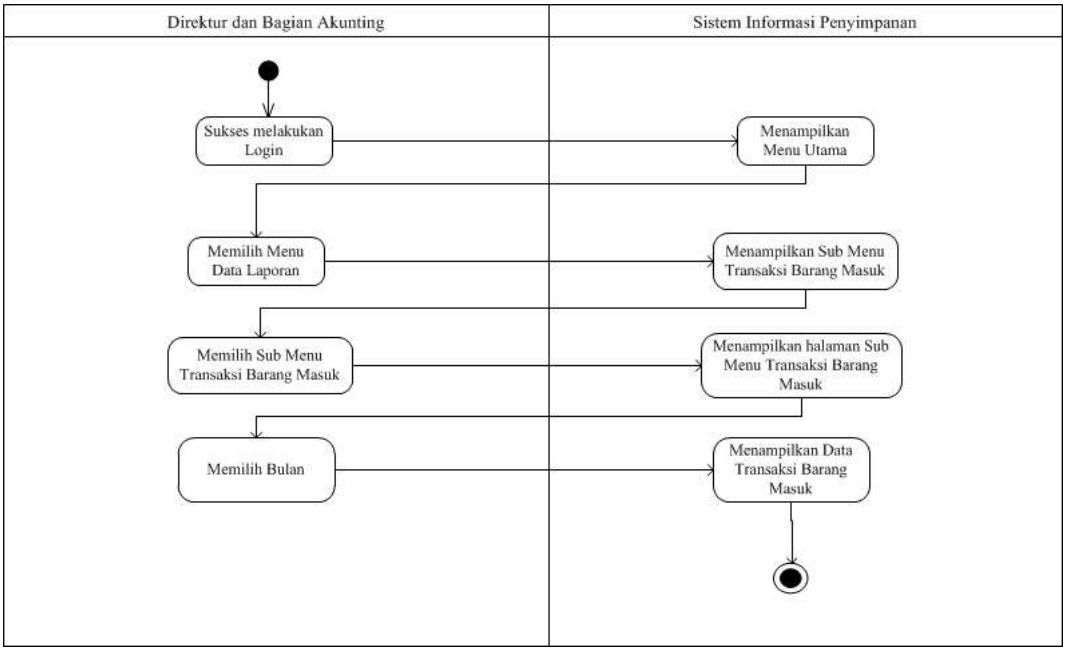


Gambar V.10 *Activity Diagram* Mencetak Laporan Data Stok

Sumber: Hasil Analisis (2018)

7. *Activity Diagram* Melihat Data Transaksi Masuk

Activity diagram berikut ini menggambarkan aktivitas yang dilakukan oleh Direktur dan Bagian Akunting yang dapat melihat laporan transaksi masuk berdasarkan bulan. *Activity diagram* melihat laporan transaksi masuk dapat dilihat pada Gambar V.11.

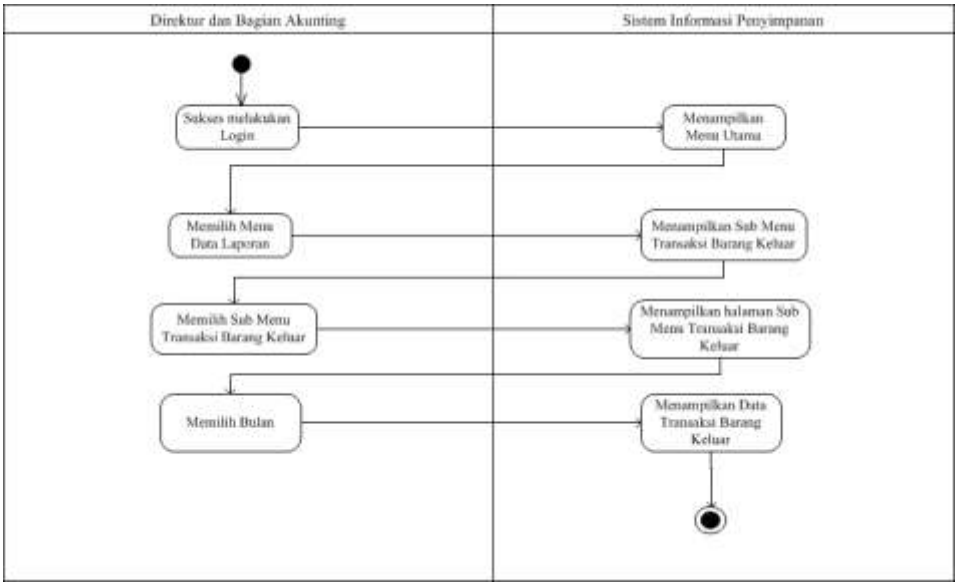


Gambar V.11 *Activity Diagram* Melihat Data Transaksi Masuk

Sumber: Hasil Analisis (2018)

8. *Activity Diagram* Melihat Data Transaksi Keluar

Activity diagram berikut ini menggambarkan aktivitas yang dilakukan oleh Direktur dan Bagian Akunting yang dapat melihat laporan transaksi keluar berdasarkan bulan. *Activity diagram* melihat laporan transaksi keluar dapat dilihat pada Gambar V.12.

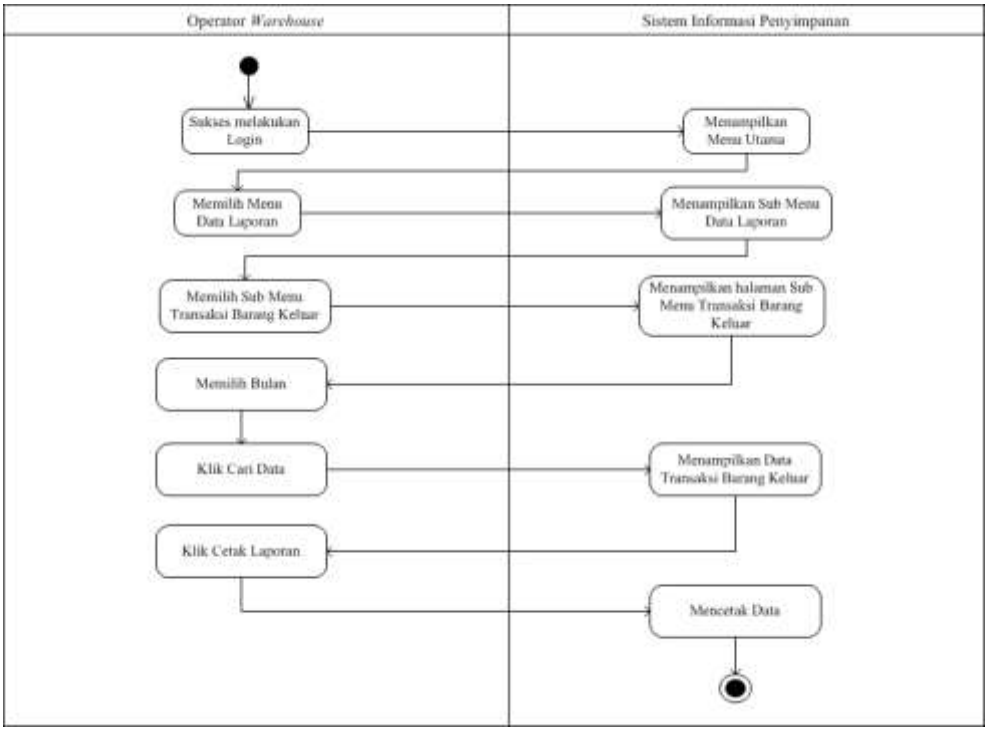


Gambar V.12 *Activity Diagram* Melihat Data Transaksi Keluar

Sumber: Hasil Analisis (2018)

9. *Activity Diagram* Mencetak Laporan Transaksi Keluar

Activity diagram berikut ini menggambarkan aktivitas yang dilakukan oleh Operator *Warehouse* yang dapat mencetak laporan transaksi keluar berdasarkan bulan. *Activity diagram* mencetak laporan transaksi keluar dapat dilihat pada Gambar V.13.

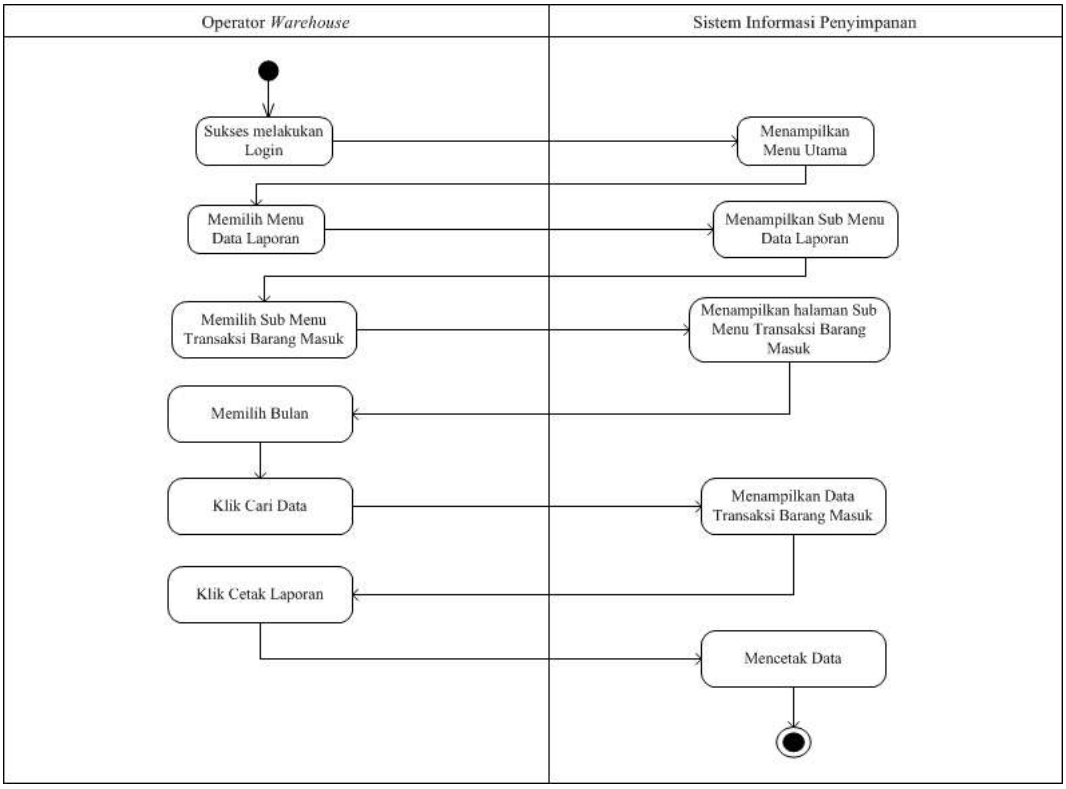


Gambar V.13 *Activity Diagram* Mencetak Laporan Transaksi Keluar

Sumber: Hasil Analisis (2018)

10. *Activity Diagram* Mencetak Laporan Transaksi Masuk

Activity diagram berikut ini menggambarkan aktivitas yang dilakukan oleh Operator *Warehouse* yang dapat mencetak laporan transaksi masuk berdasarkan bulan. *Activity diagram* mencetak laporan transaksi masuk dapat dilihat pada Gambar V.14.

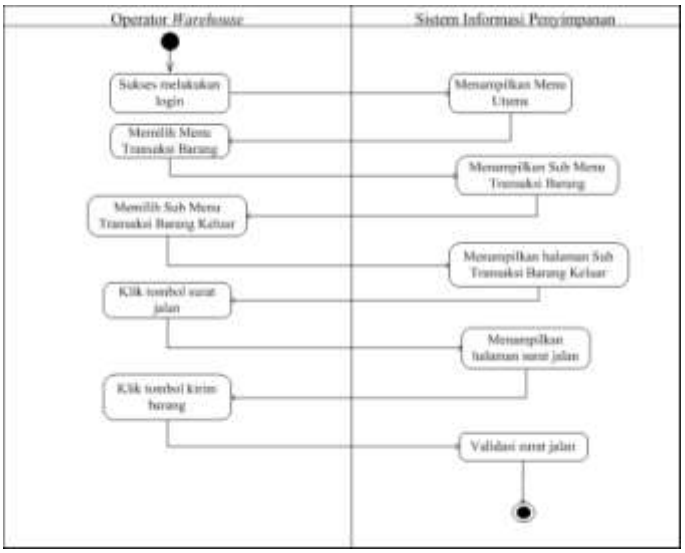


Gambar V.14 *Activity Diagram* Mencetak Laporan Transaksi Masuk

Sumber: Hasil Analisis (2018)

13. *Activity Diagram* Validasi Surat Jalan

Activity diagram berikut ini menggambarkan aktivitas validasi surat jalan yang dilakukan oleh Operator *Warehouse* Gambar V.15.



Gambar V.15 *Activity Diagram* Validasi Surat Jalan

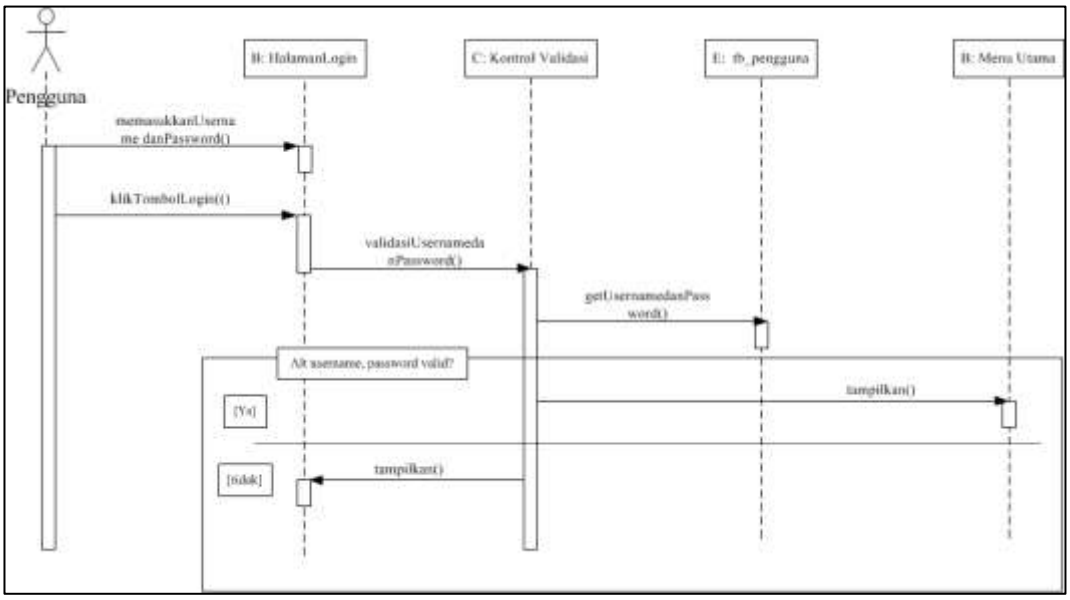
Sumber: Hasil Analisis (2018)

5.1.5 *Sequence Diagram*

Sequence diagram merupakan diagram yang menggambarkan atau memodelkan interaksi antara aktor dengan suatu sistem yang terjadi pada *use case diagram* ketika melakukan suatu proses tertentu. Urutan proses ketika melakukan suatu proses tertentu dapat diketahui dengan melihat gambaran pada *sequence* diagram. Berikut ini adalah *sequence diagram* usulan sistem informasi penyimpanan barang jadi:

1. *Sequence Diagram* Melakukan *Login*

   Sequence diagram melakukan *login* merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses *login*. Proses *login* dalam sistem informasi penyimpanan barang jadi ini dilakukan oleh pengguna agar dapat mengakses sistem, berikut merupakan *sequence diagram* dari *use case* melakukan *login* dapat dilihat pada Gambar V.16.



Gambar V.16 *Sequence Diagram* Melakukan *Login*
Sumber: Hasil Analisis (2018)

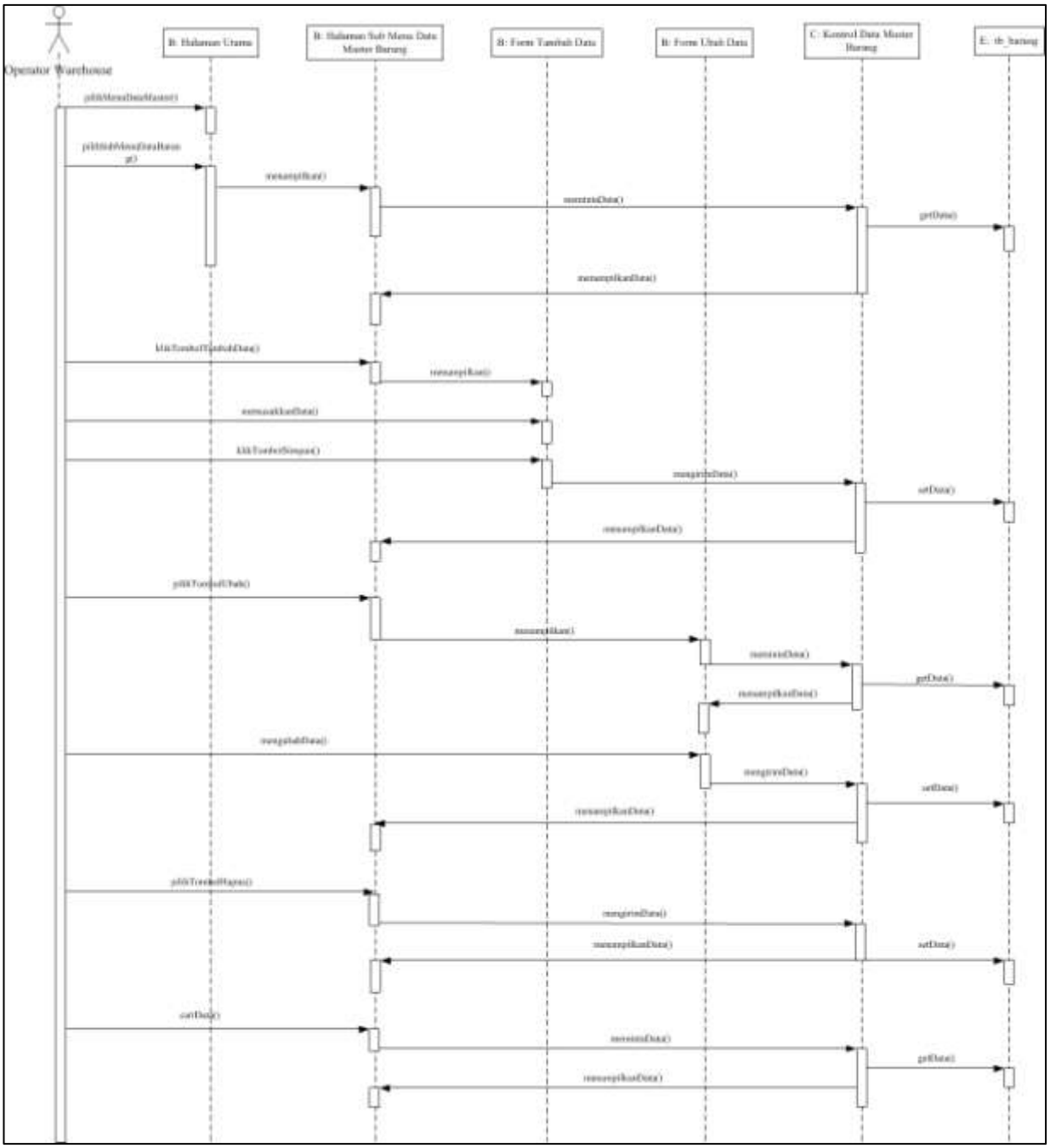
2. *Sequence Diagram* Mengelola *Data* Master

   Sequence diagram mengelola data *master* merupakan sebuah *diagram sequence* yang menjelaskan interaksi objek-objek dalam proses pengelolaan data *master* yang terdapat di dalam sistem informasi

penyimpanan barang jadi, terdapat tiga data master yaitu data barang, data pengguna, dan data pelanggan.

2.1 *Sequence Diagram* Mengelola Data *Master* Barang

Sequence diagram mengelola data *master* barang merupakan sebuah *diagram* yang menjelaskan interaksi antara objek-objek dalam proses pengelolaan data master barang. Berikut *sequence diagram* mengelola data *master* barang dapat dilihat pada Gambar V.17.

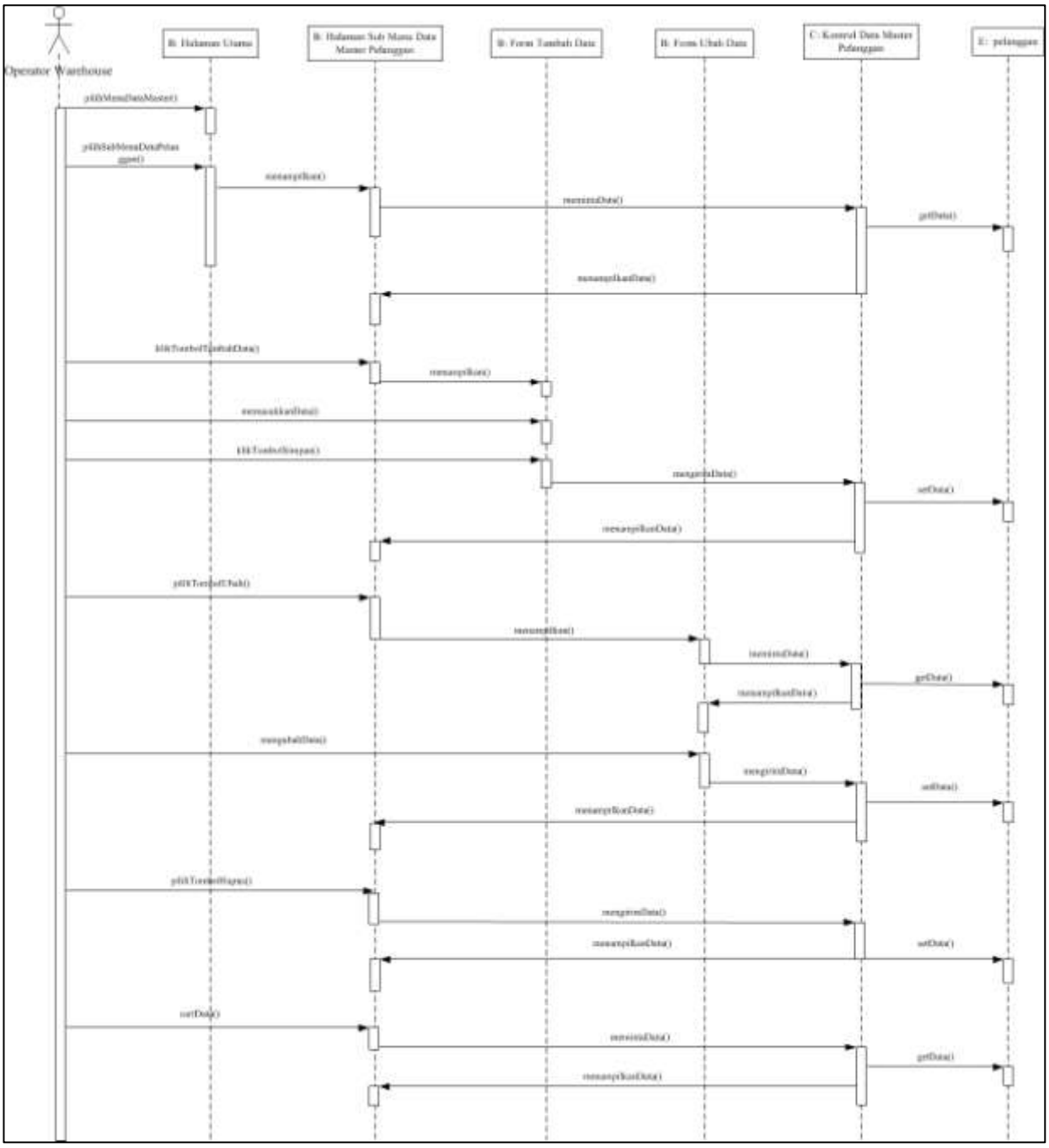


Gambar V.17 *Sequence Diagram* Mengelola Data *Master* Barang
Sumber: Hasil Analisis (2018)

2.2 *Sequence Diagram* Mengelola Data Master Pelanggan

Sequence diagram mengelola data master pelanggan merupakan sebuah *diagram* yang menjelaskan interaksi antara objek-objek dalam proses pengelolaan data master pelanggan. Berikut *sequence*

diagram mengelola data master pelanggan dapat dilihat pada Gambar V.18.

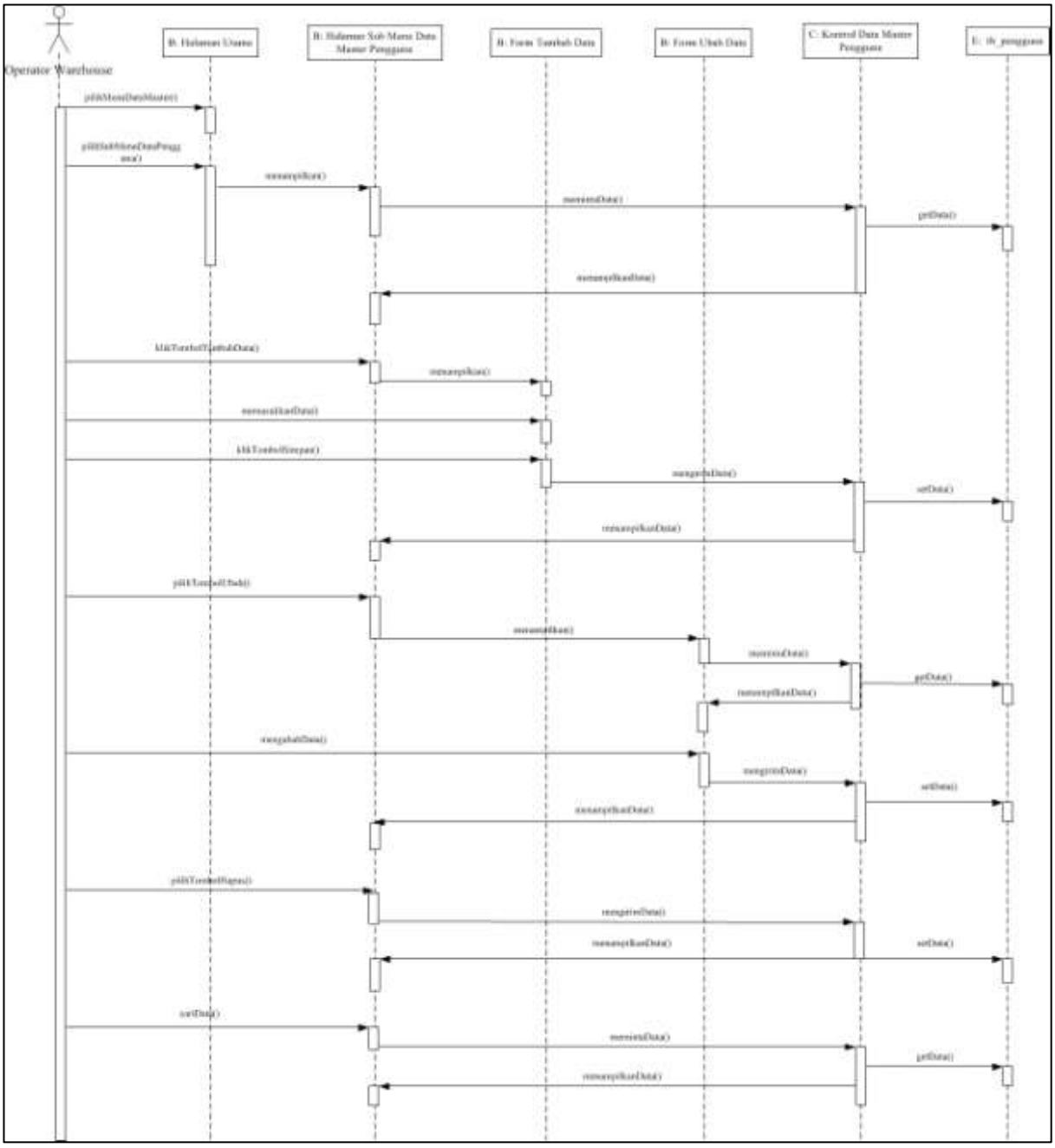


Gambar V.18 *Sequence Diagram* Mengelola Data Master Pelanggan
Sumber: Hasil Analisis (2018)

2.3 *Sequence Diagram* Mengelola Data Master Pengguna

Sequence diagram mengelola data master pengguna merupakan sebuah *diagram* yang menjelaskan interaksi antara objek-objek

dalam proses pengelolaan data master pengguna. Berikut *sequence diagram* mengelola data master pengguna dapat dilihat pada Gambar V.19.

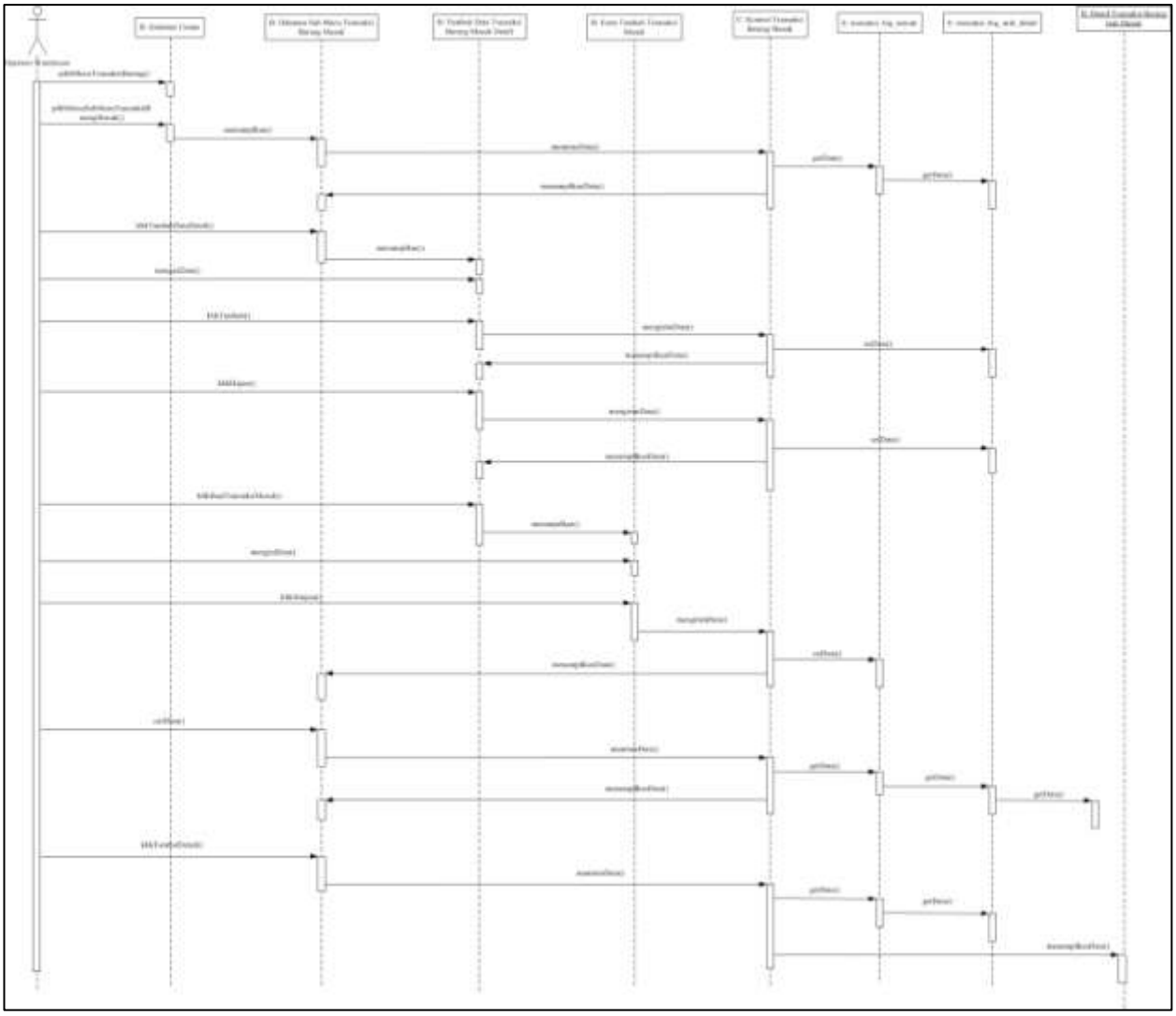


Gambar V.19 *Sequence Diagram* Mengelola Data Master Pengguna

Sumber: Hasil Analisis (2018)

3. *Sequence Diagram* Memasukkan Data Transaksi Barang Masuk

Sequence diagram memasukkan data transaksi barang masuk merupakan sebuah *diagram sequence* yang menjelaskan interaksi objek-objek dalam proses transaksi barang masuk yang terdapat di dalam sistem informasi penyimpanan barang jadi. Berikut *sequence diagram* memasukkan data transaksi barang masuk dapat dilihat pada Gambar V.20.

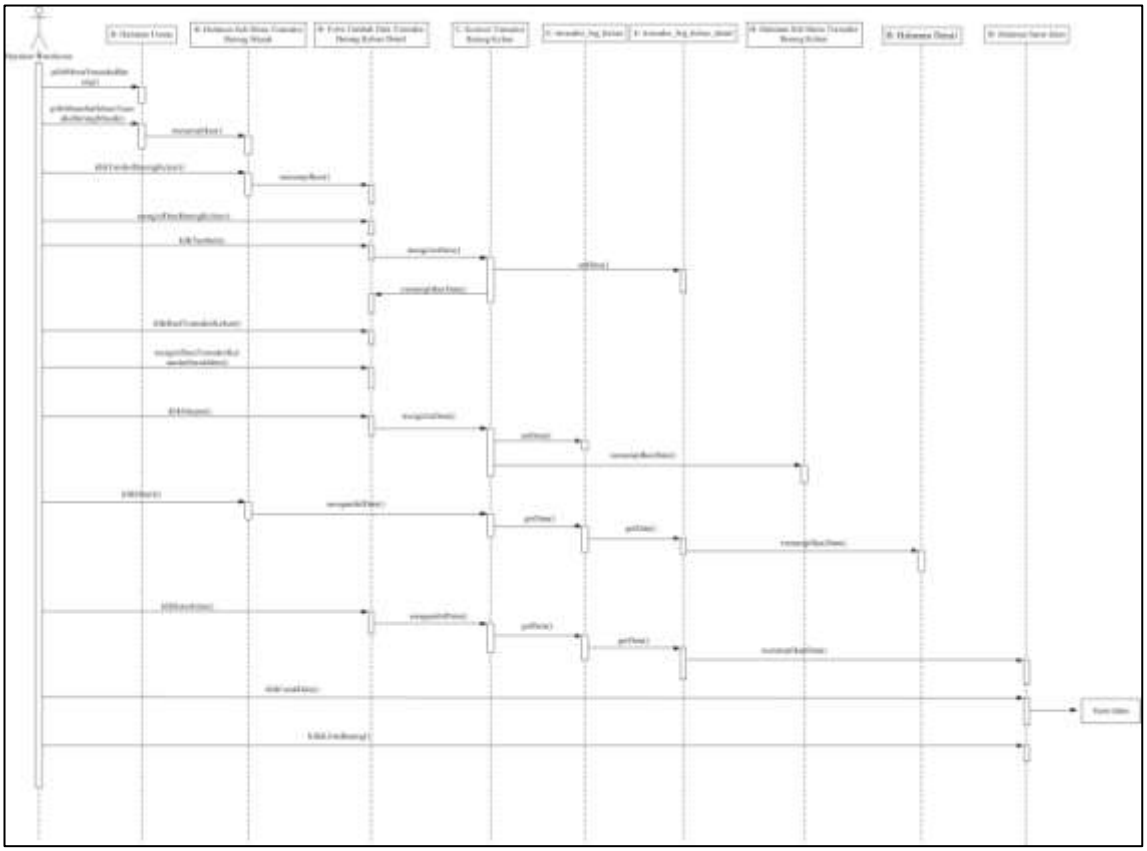


Gambar V.20 *Sequence Diagram* Memasukkan Data Transaksi Barang Masuk

Sumber: Hasil Analisis (2018)

4. *Sequence Diagram* Memasukkan Data Transaksi Barang Keluar dan Surat Jalan

Sequence diagram memasukkan data transaksi barang keluar dan surat jalan merupakan sebuah *diagram sequence* yang menjelaskan interaksi objek-objek dalam proses memasukkan data transaksi barang keluar dan surat jalan yang terdapat di dalam sistem informasi penyimpanan barang jadi. Berikut *sequence diagram* memasukkan data transaksi barang keluar dan surat jalan dapat dilihat pada Gambar V.21.



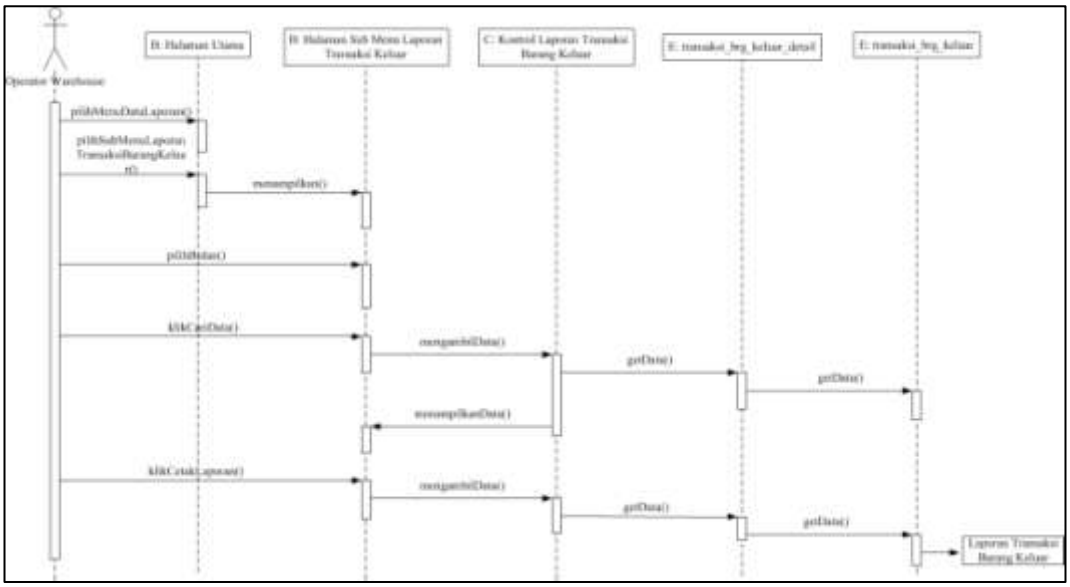
Gambar V.21 *Sequence Diagram* Memasukkan Data Transaksi Barang Keluar dan Surat Jalan

Sumber: Hasil Analisis (2018)

5. *Sequence Diagram* Mencetak Laporan Transaksi Keluar

Sequence diagram mencetak laporan transaksi keluar merupakan sebuah *diagram sequence* yang menjelaskan interaksi objek-objek dalam proses mencetak laporan transaksi keluar yang terdapat di dalam sistem informasi

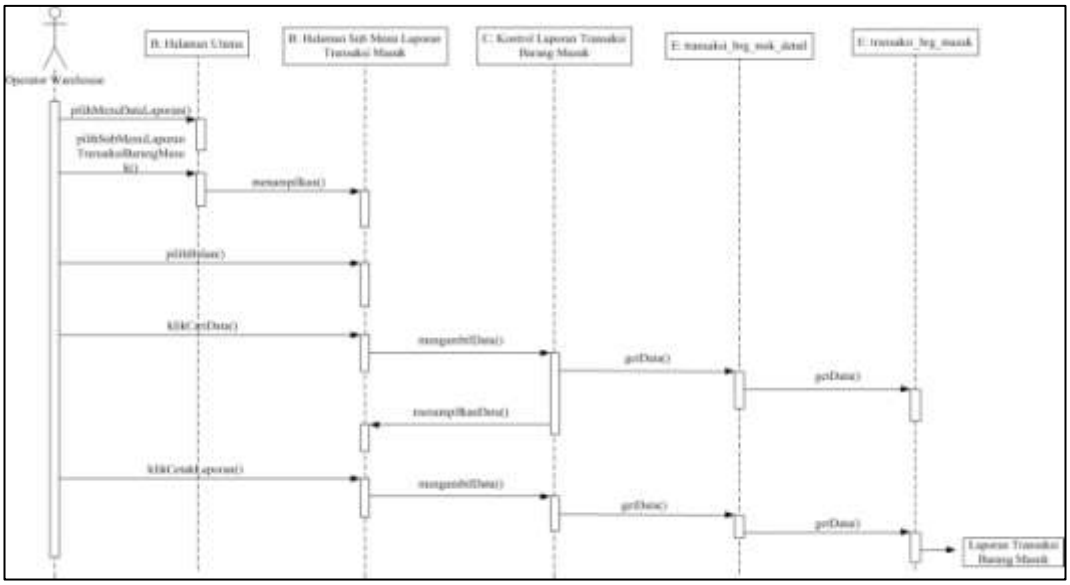
penyimpanan barang jadi. Berikut *sequence diagram* mencetak laporan transaksi keluar dapat dilihat pada Gambar V.22.



Gambar V.22 *Sequence Diagram* Mencetak Transaksi Keluar
Sumber: Hasil Analisis (2018)

6. *Sequence Diagram* Mencetak Laporan Transaksi Masuk

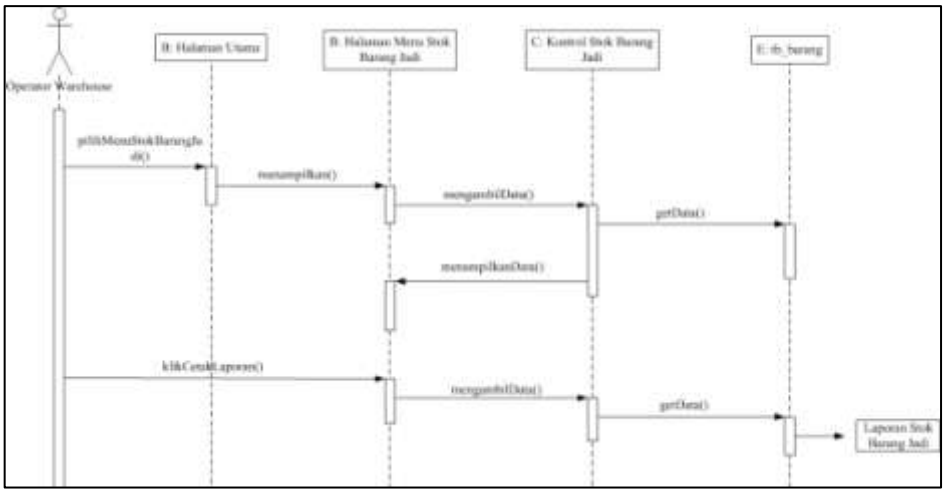
Sequence diagram mencetak laporan transaksi masuk merupakan sebuah *diagram sequence* yang menjelaskan interaksi objek-objek dalam proses mencetak laporan transaksi masuk yang terdapat di dalam sistem informasi penyimpanan barang jadi. Berikut *sequence diagram* mencetak laporan transaksi masuk dapat dilihat pada Gambar V.23.



Gambar V.23 *Sequence Diagram* Mencetak Transaksi Masuk
Sumber: Hasil Analisis (2018)

7. *Sequence Diagram* Mencetak Laporan Data Stok

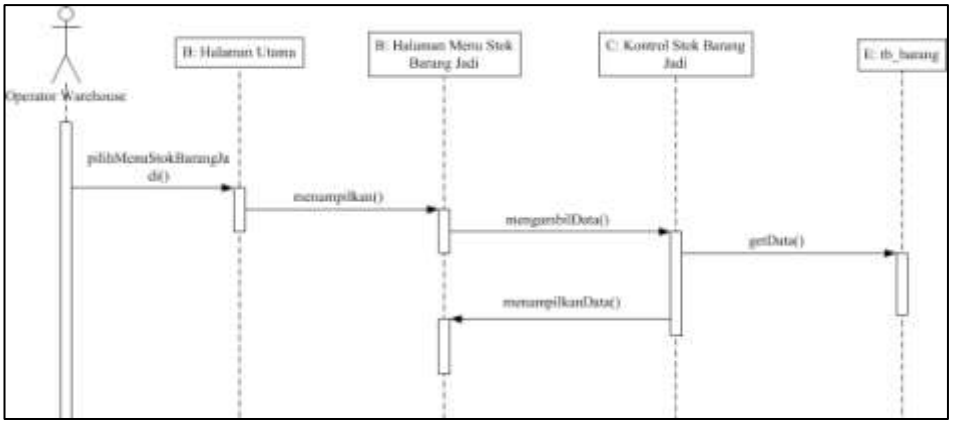
Sequence diagram mencetak laporan data stok merupakan sebuah *diagram sequence* yang menjelaskan interaksi objek-objek dalam proses mencetak laporan data stok yang terdapat di dalam sistem informasi penyimpanan barang jadi. Berikut *sequence diagram* mencetak laporan data stok dapat dilihat pada Gambar V.24.



Gambar V.24 *Sequence Diagram* Mencetak Laporan Data Stok
Sumber: Hasil Analisis (2018)

8. *Sequence Diagram* Melihat Data Stok Barang

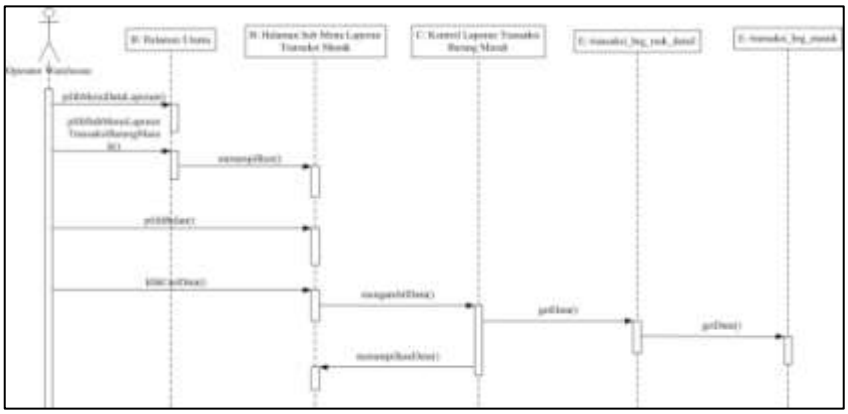
Sequence diagram melihat data stok barang merupakan sebuah *diagram sequence* yang menjelaskan interaksi objek-objek dalam proses melihat data stok barang yang terdapat di dalam sistem informasi penyimpanan barang jadi. Berikut *sequence diagram* melihat data stok barang dapat dilihat pada Gambar V.25.



Gambar V.25 *Sequence Diagram* Melihat Data Stok Barang
Sumber: Hasil Analisis (2018)

9. *Sequence Diagram* Melihat Data Transaksi Masuk

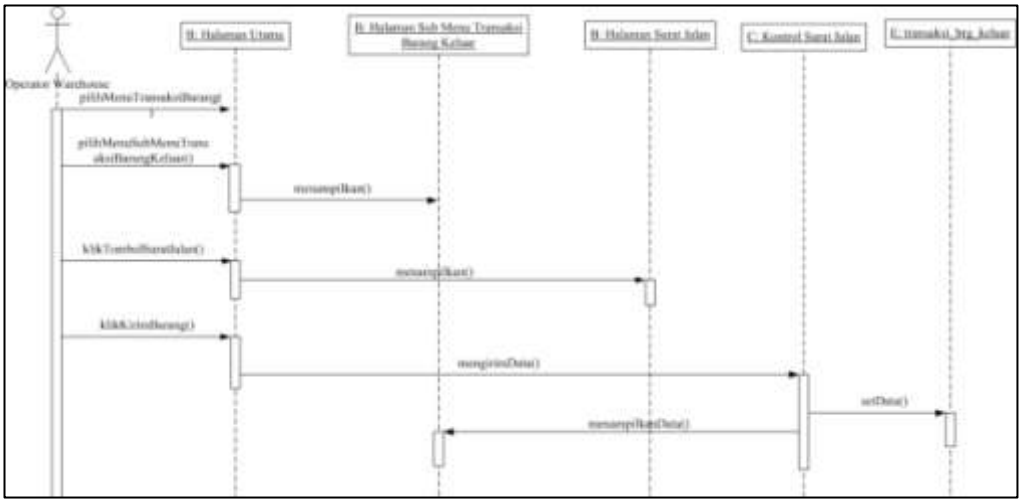
Sequence diagram melihat transaksi masuk merupakan sebuah *diagram sequence* yang menjelaskan interaksi objek-objek dalam proses melihat data transaksi masuk yang terdapat di dalam sistem informasi penyimpanan barang jadi. Berikut *sequence diagram* melihat data transaksi masuk dapat dilihat pada Gambar V.26.



Gambar V.26 *Sequence Diagram* Melihat Data Transaksi Masuk
Sumber: Hasil Analisis (2018)

10. *Sequence Diagram* Validasi Surat Jalan

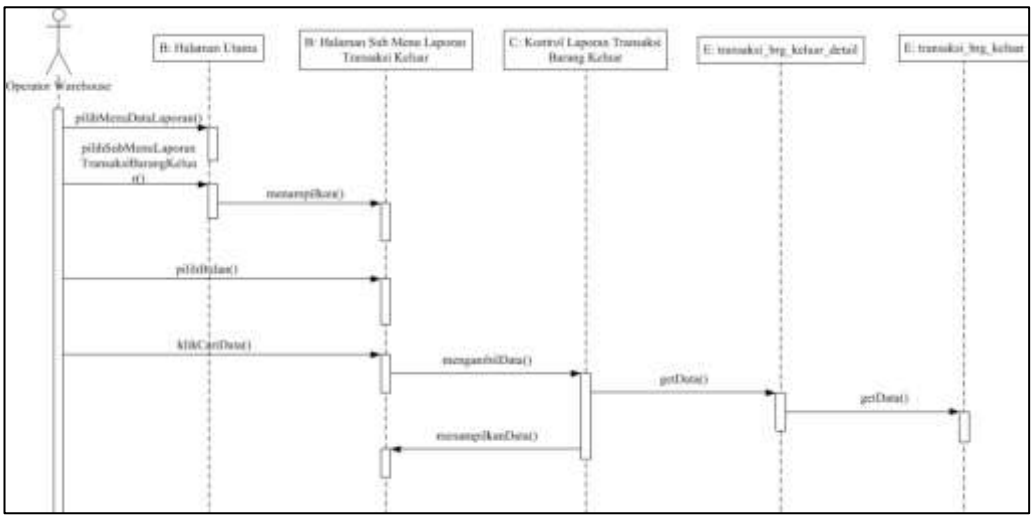
Sequence diagram validasi surat jalan dilakukan oleh Operator *Warehouse* untuk memvalidasi surat jalan. Dapat dilihat pada Gambar V.27.



Gambar V.27 *Sequence Diagram* Validasi Surat Jalan
Sumber: Hasil Analisis (2018)

11. *Sequence Diagram* Melihat Transaksi Barang Keluar

Sequence diagram melihat data transaksi barang keluar merupakan sebuah *diagram sequence* yang menjelaskan interaksi objek-objek dalam proses melihat data transaksi barang keluar yang terdapat di dalam sistem informasi penyimpanan barang jadi. Berikut *sequence diagram* melihat data transaksi barang keluar dapat dilihat pada Gambar V.28.



Gambar V.28 *Sequence Diagram* Melihat Transaksi Barang Keluar
Sumber: Hasil Analisis (2018)

5.1.6 *Class Diagram*

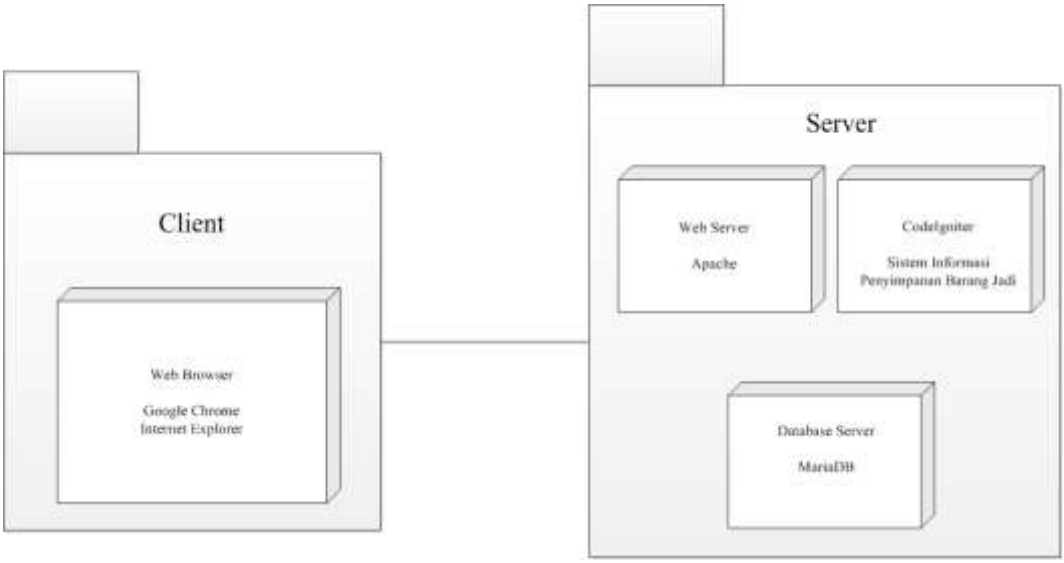
Class diagram merupakan *diagram* yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat, sistem *class diagram* memperlihatkan hubungan antar kelas dan penjelasan *detail* tiap-tiap kelas di dalam model desain dari suatu sistem. *Class diagram* sistem informasi penyimpanan barang jadi yang diusulkan dapat dilihat pada Gambar V.29.

Gambar V.29 *Class Diagram* Usulan

Sumber: Hasil Analisis (2018)

5.1.7 *Deployment Diagram*

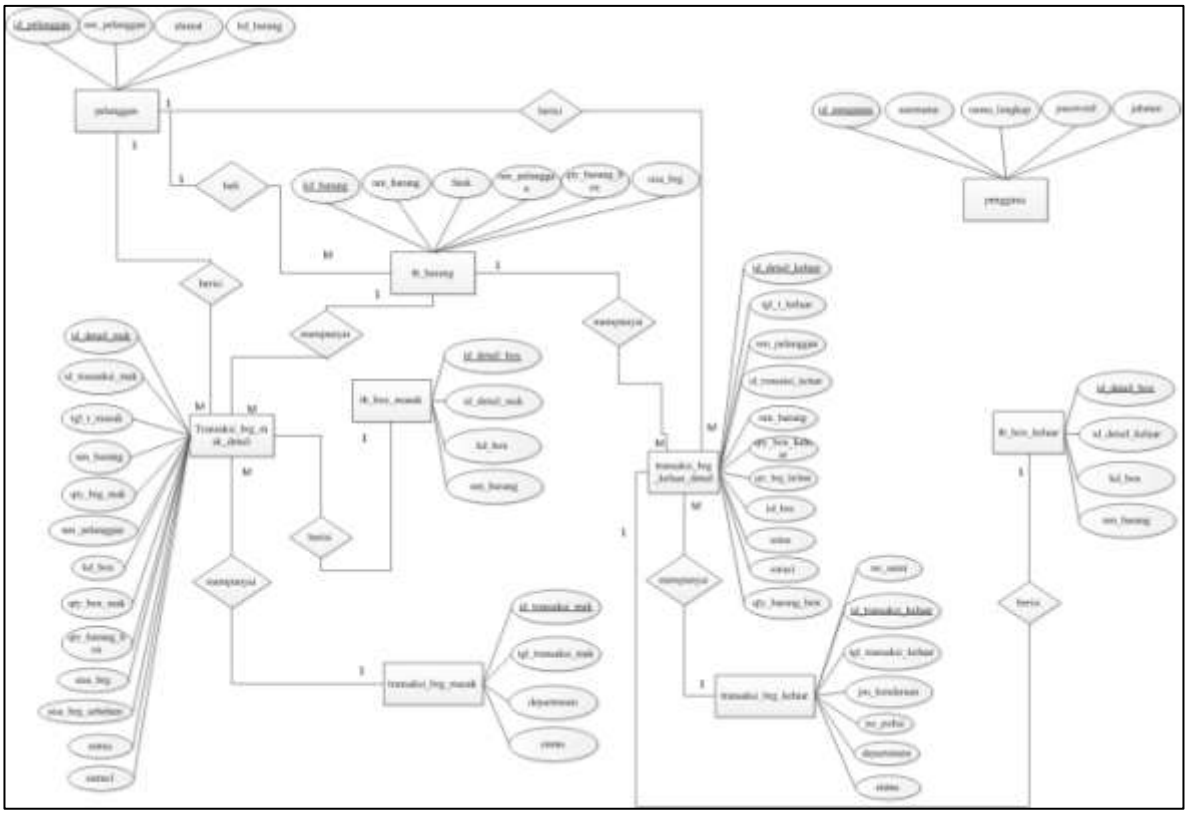
Deployment diagram pada usulan sistem informasi penyimpanan barang jadi digunakan untuk mewakili *software* dan cara *software* ditempatkan pada arsitektur fisik atau infrastruktur sistem informasi, berikut Gambar V.30 merupakan *deployment diagram* sistem informasi penyimpanan barang jadi.



Gambar V.30 *Deployment Diagram*
Sumber: Hasil Analisis (2018)

5.1.8 *Entity Relationship Diagram* (ERD)

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. Untuk menggambarkannya digunakan beberapa notasi dan simbol. ERD sistem informasi penyimpanan barang jadi yang diusulkan dapat dilihat pada Gambar V.31.



Gambar V.31 *Entity Relationship Diagram* Usulan

Sumber: Hasil Analisis (2018)

5.1.9 Kamus Data

Kamus data merupakan daftar data yang terdapat di dalam sebuah sistem dengan maksud untuk mendefinisikan aliran data di dalam sebuah sistem dengan lengkap dan sesuai dengan sistem, sehingga pengguna mempunyai pengertian yang sama tentang *input*, *output* dan komponen data *store*. Pada tahap perancangan sistem, kamus data digunakan untuk merancang basis data. Berikut adalah kamus data sistem informasi penyimpanan barang jadi.

1. Spesifikasi Tabel Data Pelanggan

   Nama Tabel : Data Pelanggan

   Akronim : pelanggan

   Fungsi :Untuk menyimpan data pelanggan agar memudahkan pengguna mengetahui pelanggan dari setiap barang jadi

   Tipe : Data Master

Tabel V.14 Tabel Data Pelanggan

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	Id Pelanggan	id_pelanggan	*char*	10	*Primary Key*
2.	Nama Pelanggan	nm_pelanggan	*varchar*	30	*Not Null*
3.	Alamat Pelanggan	alamat	*text*	-	*Not Null*
4.	No Telp	no_telp	*varchar*	11	*Not Null*

Sumber: Hasil Analisis (2018)

2. Spesifikasi Tabel Data Barang

Nama Tabel : Data Barang

Akronim : tb_barang

Fungsi :Untuk menyimpan data barang jadi yang dihasilkan dari proses produksi

Tipe : Data Master

Tabel V.15 Tabel Data Barang

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	Kode Barang	kd_barang	*char*	11	*Primary Key*
2.	Nama Barang	nm_barang	*varchar*	20	*Not Null*
3.	Nama Pelanggan	nm_pelanggan	*varchar*	30	*Not Null*
4.	Jumlah Barang Per-*box*	qty_barang_box	*tinyint*	2	*Not Null*
5.	Stok	stok	*int*	5	*Not Null*
6.	Sisa Barang *Box*	sisa_brg	*tinyint*	3	*Not Null*

Sumber: Hasil Analisis (2018)

3. Spesifikasi Tabel Data Pengguna

Nama Tabel : Data Pengguna

Akronim : tb_pengguna

Fungsi : Untuk menyimpan data pengguna dan digunakan untuk hak akses ketika pengguna melakukan *login*

Tipe : Data Master

Tabel V.16 Tabel Data Pengguna

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	Id Pengguna	id_pengguna	*char*	5	*Primary Key*
2.	*Username*	username	*varchar*	10	*Not Null*
3.	Nama Lengkap	nama_lengkap	*varchar*	30	*Not Null*
4.	*Password*	password	*varchar*	20	*Not Null*
5.	Jabatan	jabatan	*varchar*	20	*Not Null*

Sumber: Hasil Analisis (2018)

4. Spesifikasi Tabel Data Transaksi Barang Masuk

Nama Tabel : Data Transaksi Barang Masuk

Akronim : transaksi_brg_masuk

Fungsi : Untuk menyimpan data transaksi barang jadi yang masuk ke dalam gudang

Tipe : Data Transaksi

Tabel V.17 Tabel Data Transaksi Barang Masuk

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	Id Transaksi Masuk	id_transaksi_msk	*char*	11	*Primary Key*
2.	Tanggal Transaksi Masuk	tgl_transaksi_msk	*date*	-	*Not Null*
3.	Departemen	departemen	*char*	7	*Not Null*
4.	Status	status	*tinyint*	1	*Not Null*

Sumber: Hasil Analisis (2018)

5. Spesifikasi Tabel Data Transaksi Barang Masuk Detail

Nama Tabel : Data Transaksi Barang Masuk Detail

Akronim : transaksi_brg_msk_detail

Fungsi : Untuk menyimpan data transaksi barang jadi yang masuk ke dalam gudang secara detail

Tipe : Data Transaksi

Tabel V.18 Tabel Data Transaksi Barang Masuk Detail

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	Id Transaksi Masuk Detail	id_detail_msk	*char*	15	*Primary Key*
2.	Id Transaksi Masuk	id_transaksi_msk	*char*	11	*Foreign Key*
3.	Tanggal Transaksi Keluar	tgl_t_masuk	*datetime*	-	*Not Null*
4.	Nama Barang	nm_barang	*varchar*	20	*Not Null*
5.	Nama Pelanggan	nm_pelanggan	*varchar*	30	*Not Null*
6.	Sisa Barang Sebelum	sisa_brg_sebelum	*tinyint*	3	*Not Null*
7.	Jumlah Barang Per-*box*	qty_barang_box	*tinyint*	2	*Not Null*
8.	Jumlah Barang Masuk	qty_brg_msk	*tinyint*	3	*Not Null*
9.	Jumlah *Box* Masuk	qty_box_msk	*tinyint*	2	*Not Null*
10.	Sisa Barang	sisa_brg	*tinyint*	3	*Not Null*
11.	Kode Box	kd_box	*text*	-	*Not Null*
12.	Status	status	*tinyint*	1	*Not Null*

Tabel V.18 Tabel Data Transaksi Barang Masuk Detail (lanjutan)

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
13.	Status1	status1	*tinyint*	1	*Not Null*

Sumber: Hasil Analisis (2018)

6. Spesifikasi Tabel Data Transaksi Barang Keluar

Nama Tabel : Data Transaksi Barang Keluar

Akronim : transaksi_brg_keluar

Fungsi : Untuk menyimpan data transaksi barang jadi yang keluar dari dalam gudang

Tipe : Data Transaksi

Tabel V.19 Tabel Data Transaksi Barang Keluar

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	Nomor Surat	no_surat	*char*	10	*Primary Key*
2.	Id Transaksi Keluar	id_transaksi_keluar	*char*	11	*Unique*
3.	Tanggal Transaksi Keluar	tgl_transaksi_keluar	*date*	-	*Not Null*
4.	Departemen	departemen	*char*	15	*Not Null*
5.	Jenis Kendaraan	jns_kendaraan	*varchar*	15	*Not Null*
6.	Nomor Polisi	no_polisi	*varchar*	10	*Not Null*
7.	Status	Status	*tinyint*	1	*Not Null*

Sumber: Hasil Analisis (2018)

7. Spesifikasi Tabel Data Transaksi Barang Keluar Detail

Nama Tabel : Data Transaksi Barang Keluar Detail

Akronim : transaksi_brg_keluar_detail

Fungsi : Untuk menyimpan data transaksi barang jadi yang keluar dari dalam gudang secara detail

Tipe : Data Transaksi

Tabel V.20 Tabel Data Transaksi Barang Keluar Detail

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	Id Detail Keluar	id_detail_keluar	*char*	15	*Primary Key*
2.	Id Transaksi Keluar	id_transaksi_keluar	*char*	11	*Foreign Key*
3.	Tanggal Transaksi Keluar	tgl_t_keluar	*datetime*	-	*Not Null*
4.	Nama Barang	nm_barang	*varchar*	30	*Not Null*
5.	Nama Pelanggan	nm_pelanggan	*varchar*	30	*Not Null*
6.	Jumlah Barang Keluar	qty_brg_keluar	*tinyint*	2	*Not Null*
7.	Jumlah Box Keluar	qty_box_keluar	*tinyint*	2	*Not Null*
8.	Jumlah Barang Per-*Box*	qty_barang_box	*tinyint*	3	*Not Null*
9.	Kode Box	kd_box	*text*	-	*Not Null*
10.	Status	Status	*tinyint*	1	*Not Null*
11.	Status1	Status1	*tinyint*	1	*Not Null*

Sumber: Hasil Analisis (2018)

8. Spesifikasi Tabel Data *Box* Detail Keluar

Nama Tabel : Data *Box* Detail Keluar

Akronim : tb_box_keluar

Fungsi : Untuk menyimpan data *box* yang keluar dari gudang secara detail

Tipe : Data Transaksi

Tabel V.21 Tabel Data *Box* Keluar Detail

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	Id Detail Box	id_detail_box	*int*	10	*Primary Key*
2.	Id Detail Keluar	id_detail_keluar	*char*	15	*Foreign Key*
3.	Kode Box	kd_box	*varchar*	10	*Not Null*
4.	Nama Barang	nm_barang	*varchar*	30	*Not Null*

Sumber: Hasil Analisis (2018)

9. Spesifikasi Tabel Data Box Detail Masuk

Nama Tabel : Data Box Detail Masuk

Akronim : tb_box_masuk

Fungsi : Untuk menyimpan data box yang masuk ke gudang secara detail

Tipe : Data Transaksi

Tabel V.22 Tabel Data Box Masuk Detail

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	Id Detail Box	id_detail_box	*int*	10	*Primary Key*
2.	Id Detail Masuk	id_detail_masuk	*char*	15	*Foreign Key*
3.	Kode Box	kd_box	*varchar*	10	*Not Null*
4.	Nama Barang	nm_barang	*varchar*	30	*Not Null*

Sumber: Hasil Analisis (2018)

5.1.10 *Windows Navigation Diagram* (WND)

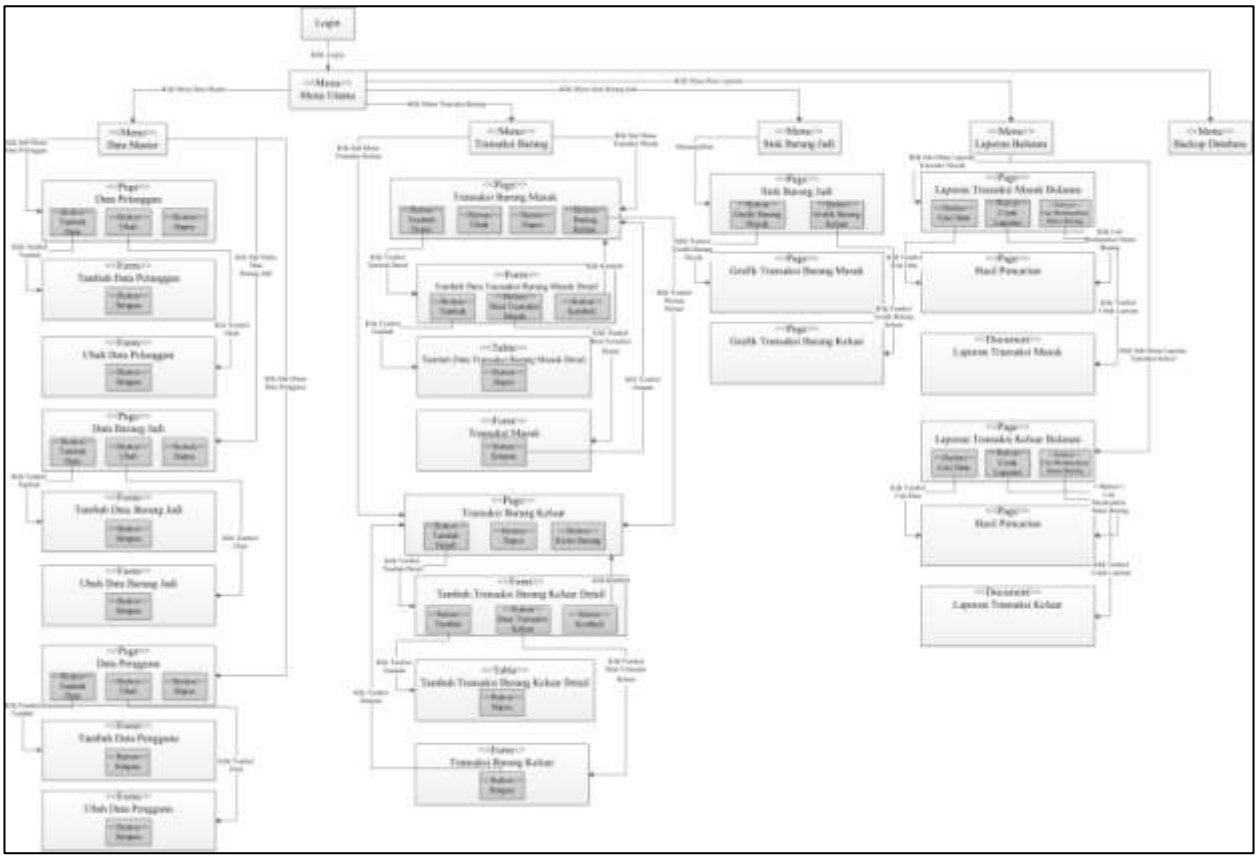
Windows Navigation Diagram merupakan *statechart* diagram khusus yang berfokus pada *user interface.* Yang mana menunjukkan bagaimana semua layar, formulir, dan laporan terkait, menunjukkan bagaimana pengguna bergerak dari satu ke yang lain dan seperti diagram keadaan untuk *user interface*:

d. Kotak mewakili komponen.

e. Panah mewakili transisi.

f. Stereotipe menunjukkan tipe antarmuka.

Windows Navigation Diagram usulan dapat dilihat pada Gambar V.32.



Gambar V.32 *Windows Navigation Diagram* Usulan

Sumber: Hasil Analisis (2018)

5.1.11 Perancangan Sistem Antarmuka

Sistem antarmuka aplikasi pada sistem informasi penyimpanan barang jadi berupa rancangan tampilan yang akan dibuat. Perancangan antarmuka aplikasi usulan adalah sebagai berikut:

1. Halaman *Login*

   Halaman *login* adalah halaman yang digunakan untuk masuk ke dalam halaman utama. Untuk masuk ke dalam halaman utama, pengguna harus memasukkan nama pengguna dan kata sandi yang benar. Rancangan halaman *login* dapat dilihat pada Gambar V.33.

Gambar V.33 Halaman *Login*
Sumber: Hasil Analisis (2018)

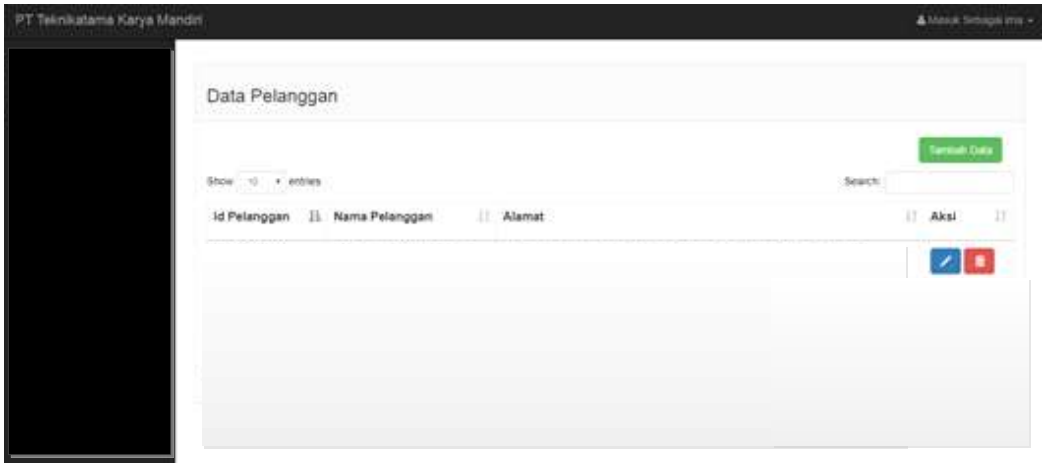
2. Tampilan Menu Utama

Halaman menu utama adalah tampilan awal pada aplikasi yang berisi lima menu. Antara lain adalah halaman utama, data master, transaksi barang, stok barang jadi, dan data laporan. Rancangan menu utama dapat dilihat pada Gambar V.34.

Gambar V.34 Halaman Menu Utama
Sumber: Hasil Analisis (2018)

3. Tampilan Sub Menu Data *Master* Pelanggan

Halaman sub menu dari data *master* pelanggan digunakan untuk mengelola data *master* pelanggan seperti menambah, hapus, ubah data. Rancangan dari halaman sub menu data *master* pelanggan dapat dilihat pada Gambar V.35.



Gambar V.35 Halaman Sub Menu Data *Master* Pelanggan

Sumber: Hasil Analisis (2018)

4. Tampilan *Form* Tambah Pelanggan

Form tambah data pelanggan digunakan untuk menambah adanya data pelanggan baru. Gambar dari *form* tambah pelanggan dapat dilihat pada Gambar V.36.

Gambar V.36 *Form* Tambah Pelanggan

Sumber: Hasil Analisis (2018)

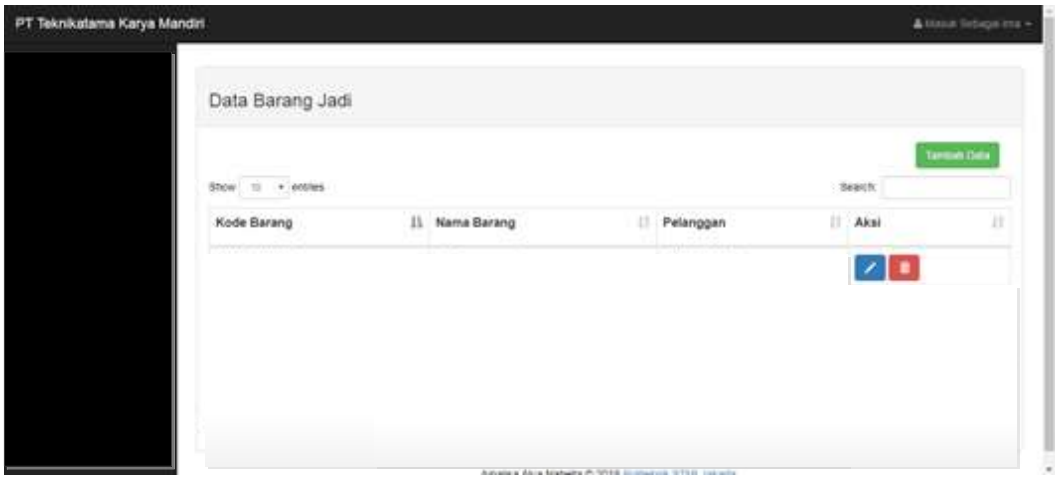
5. Tampilan *Form* Ubah Pelanggan

Form ubah pelanggan digunakan untuk mengubah data pelanggan. Adapun tampilan dari *form* ubah pelanggan dapat dilihat pada Gambar V.37.

Gambar V.37 *Form* Ubah Data Pelanggan
Sumber: Hasil Analisis (2018)

6. Tampilan Halaman Data *Master* Data Barang Jadi

Halaman data *master* barang jadi digunakan untuk mengelola data dari barang jadi. Seperti menambah, hapus, dan ubah data. Gambar dari halaman data *master* pelanggan dapat dilihat pada Gambar V.38.

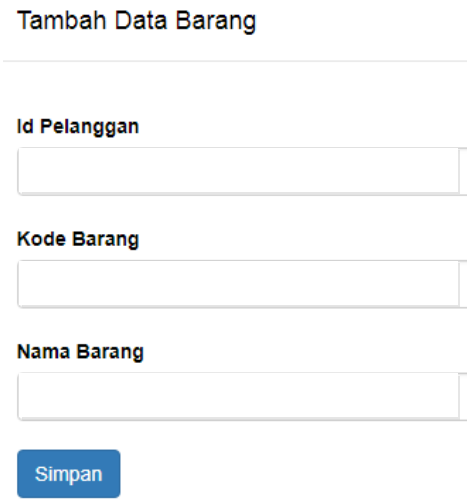


Gambar V.38 Halaman Data Master Barang Jadi

Sumber: Hasil Analisis (2018)

7. *Form* Tambah Data Barang Jadi

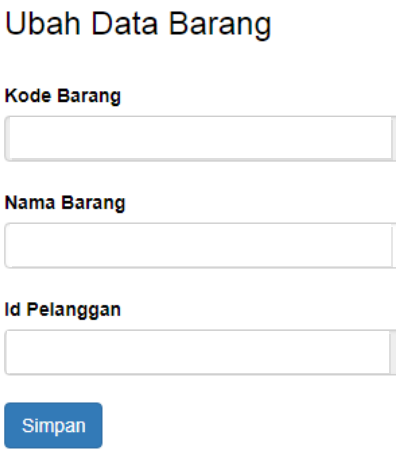
Form tambah data barang jadi digunakan untuk menambah adanya barang jadi yang baru. Gambar dari *form* tambah data barang jadi dapat dilihat pada Gambar V.39.



Gambar V.39 *Form* Tambah Data Barang Jadi
Sumber: Hasil Analisis (2018)

8. Tampilan *Form* Ubah Barang

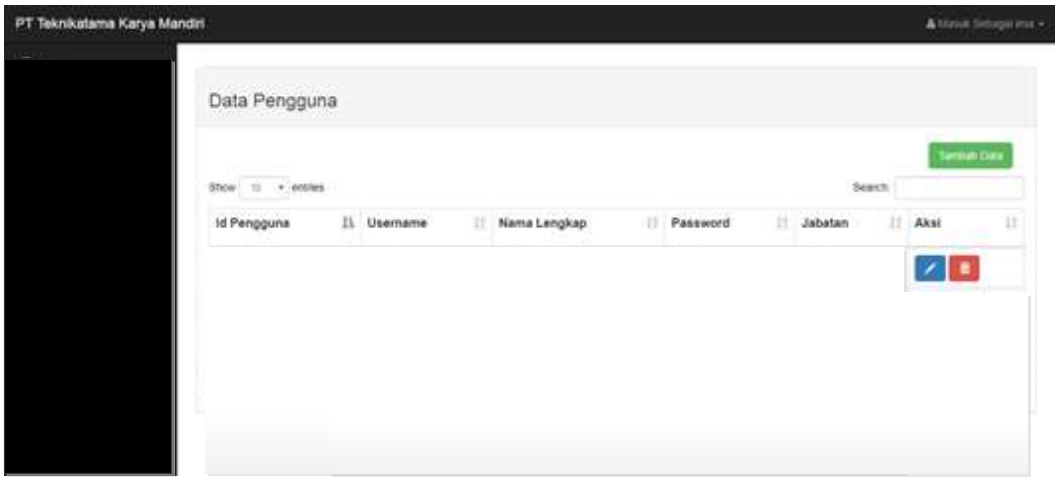
Form ubah barang digunakan untuk mengubah data barang jadi. Adapun tampilan dari *form* ubah barang dapat dilihat pada Gambar V.40.



Gambar V.40 *Form* Ubah Data Barang Jadi
Sumber: Hasil Analisis (2018)

9. Tampilan Halaman Data Master Pengguna

Halaman data master pengguna digunakan untuk menyimpan data-data dari pengguna aplikasi. Hal ini juga digunakan untuk memperoleh hak akses dalam membuka aplikasi *web*. Adapun gambar dari tampilan data master pengguna dapat dilihat pada Gambar V.41.



Gambar V.41 Halaman Data Master Pelanggan
Sumber: Hasil Analisis (2018)

10. *Form* Tambah Data Pengguna

Form tambah data pengguna jadi digunakan untuk menambah adanya pengguna yang baru. Gambar dari *form* tambah data pengguna dapat dilihat pada Gambar V.42.

Gambar V.42 *Form* Tambah Data Master Pengguna
Sumber: Hasil Analisis (2018)

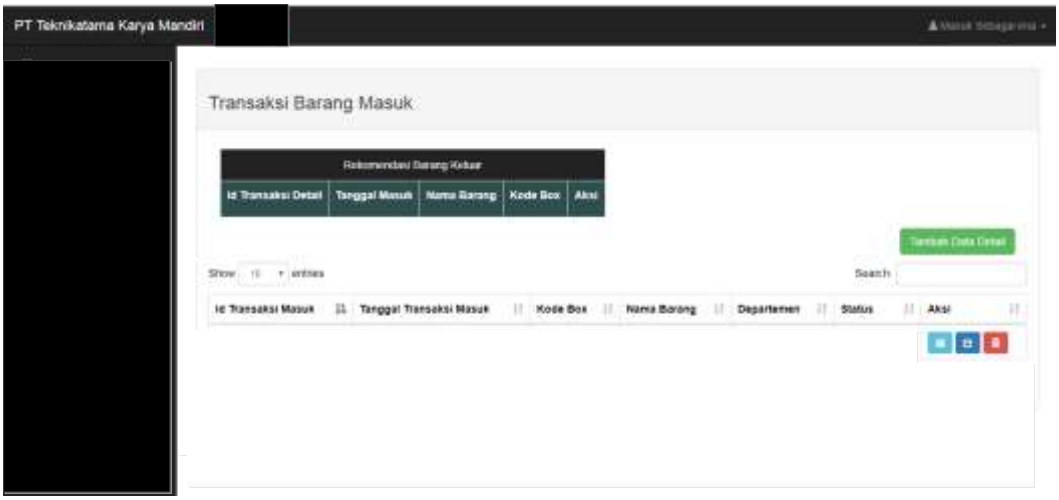
11. *Form* Ubah Data Pengguna

Form ubah pengguna digunakan untuk mengubah data pengguna. Adapun tampilan dari *form* ubah data pengguna dapat dilihat pada Gambar V.43.

Gambar V.43 *Form* Ubah Data Master Pengguna
Sumber: Hasil Analisis (2018)

12. Halaman Data Transaksi Masuk

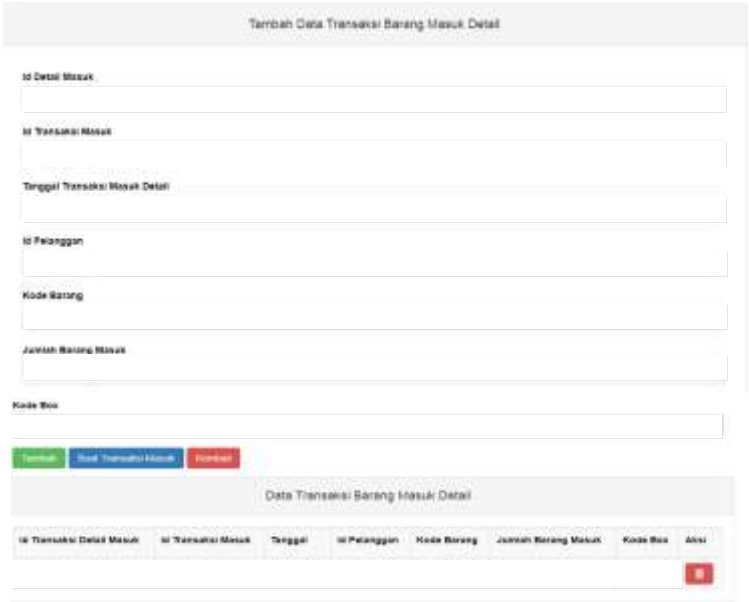
Halaman data transaksi masuk ini digunakan untuk menyimpan data transaksi barang yang masuk ke dalam gudang. Adapun gambar dari halaman data transaksi masuk dapat dilihat pada Gambar V.44.



Gambar V.44 Halaman Data Transaksi Masuk
Sumber: Hasil Analisis (2018)

13. *Form* Tambah Data Detail Barang Masuk

Form tambah data detail barang digunakan untuk menambah adanya data barang yang masuk ke dalam gudang. Gambar dari *form* tambah detail barang dapat dilihat pada Gambar V.45.



Gambar V.45 *Form* Tambah Detail Data Barang
Sumber: Hasil Analisis (2018)

14. *Form* Buat Transaksi Masuk

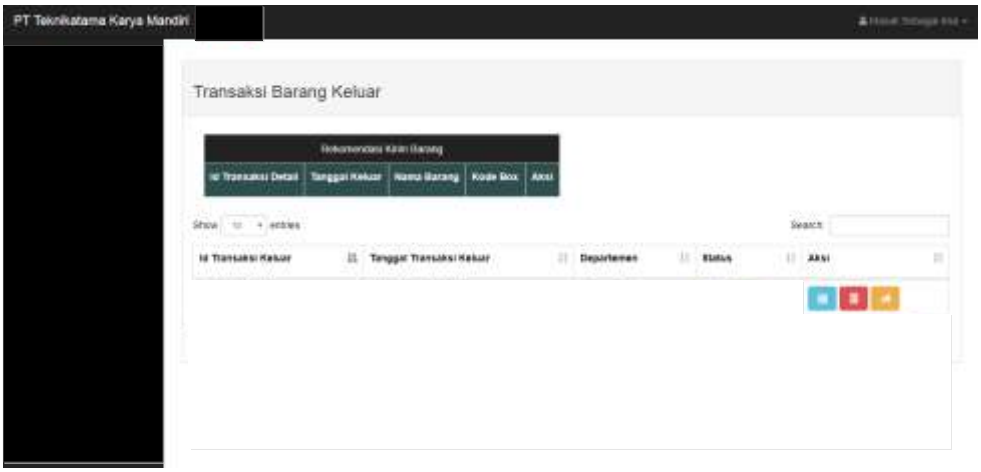
Form buat transaksi masuk ini adalah untuk membuat data transaksi masuk yang di dalamnya berisikan data-data barang yang masuk ke dalam gudang. Adapun gambar dari transaksi masuk dapat diliat pada Gambar V.46.

Gambar V.46 *Form* Buat Transaksi Masuk

Sumber: Hasil Analisis (2018)

15. Halaman Data Transaksi Keluar

Halaman data transaksi keluar ini digunakan untuk menyimpan data transaksi barang yang keluar dari dalam gudang. Adapun gambar dari halaman data transaksi keluar dapat dilihat pada Gambar V.47.

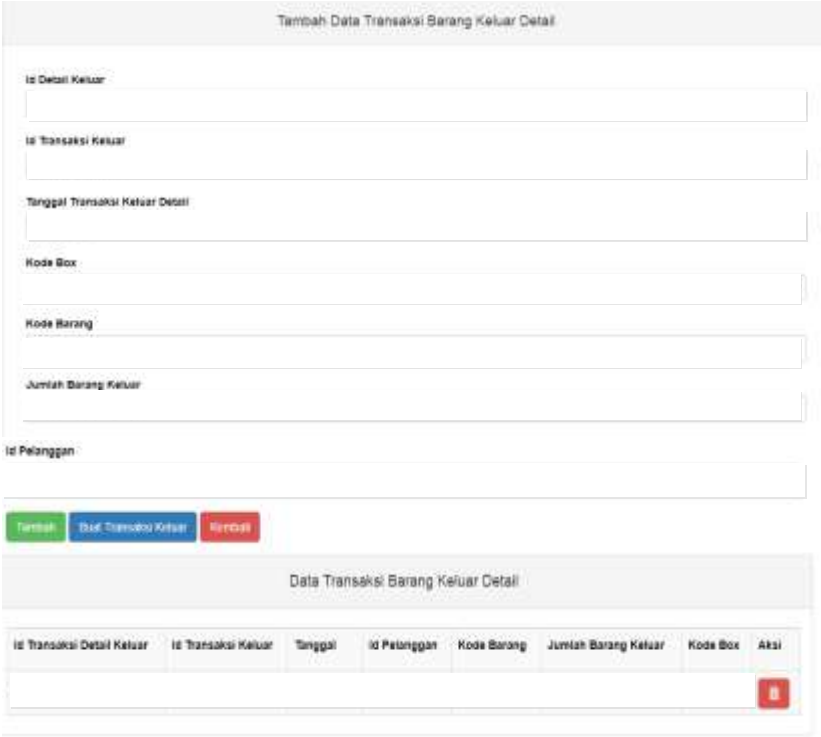


Gambar V.47 Halaman Data Transaksi Keluar

Sumber: Hasil Analisis (2018)

16. *Form* Tambah Data Detail Transaksi Keluar

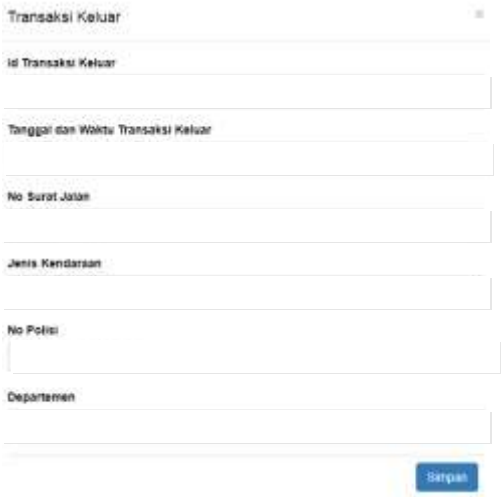
Form tambah data detail transaksi keluar digunakan untuk menambah adanya data barang yang keluar dari dalam gudang. Gambar dari *form* tambah detail barang keluar dapat dilihat pada Gambar V.48.



Gambar V.48 *Form* Tambah Detail Barang Keluar
Sumber: Hasil Analisis (2018)

17. *Form* Buat Transaksi Keluar

Form buat transaksi keluar ini adalah untuk membuat data transaksi keluar yang di dalamnya berisikan data-data barang yang keluar dari dalam gudang. Adapun gambar dari transaksi keluar dapat diliat pada Gambar V.49.



Gambar V.49 *Form* Buat Transaksi Barang Keluar

Sumber: Hasil Analisis (2018)

18. Halaman Detail Transaksi Barang Masuk

Halaman detail transaksi barang masuk digunakan untuk menampilkan data detail barang apa saja yang masuk ke dalam gudang dalam satu kali transaksi. Adapun gambar dari detail transaksi barang masuk dapat dilihat pada Gambar V.50.

Gambar V.50 Halaman Detail Transaksi Barang Masuk

Sumber: Hasil Analisis (2018)

19. Halaman Detail Transaksi Barang Keluar

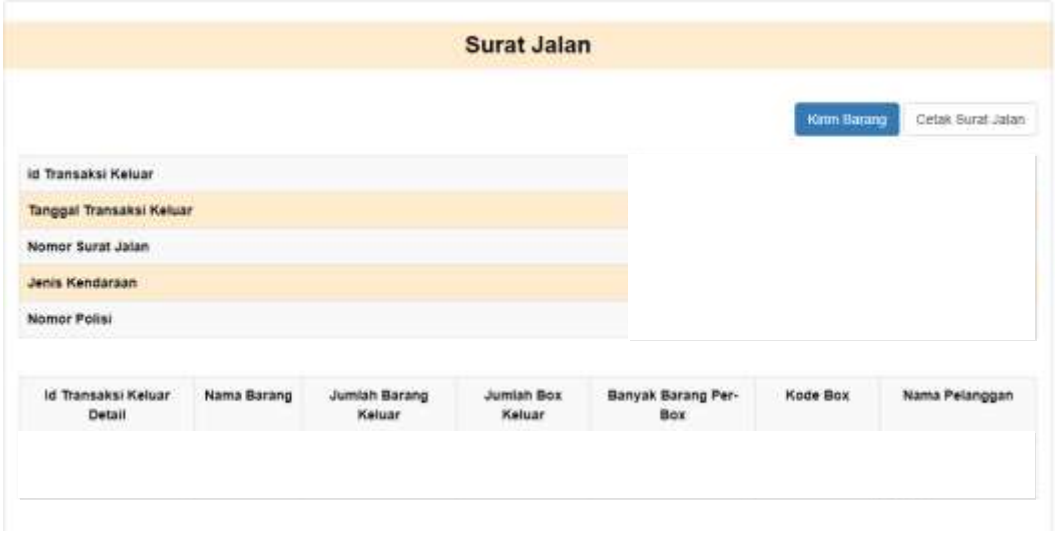
Halaman detail transaksi barang keluar digunakan untuk menampilkan data detail barang apa saja yang keluar dari dalam gudang dalam satu kali transaksi. Adapun gambar dari detail transaksi barang masuk dapat dilihat pada Gambar V.51.

Gambar V.51 Halaman Detail Transaksi Barang Keluar

Sumber: Hasil Analisis (2018)

20. Halaman Data Surat Jalan

Halaman data surat jalan digunakan untuk melihat barang apa saja yang keluar dari dalam gudang dan akan dikirim oleh pihak *delivery order*. Adapun gambar dari data surat jalan dapat dilihat pada Gambar V.52.

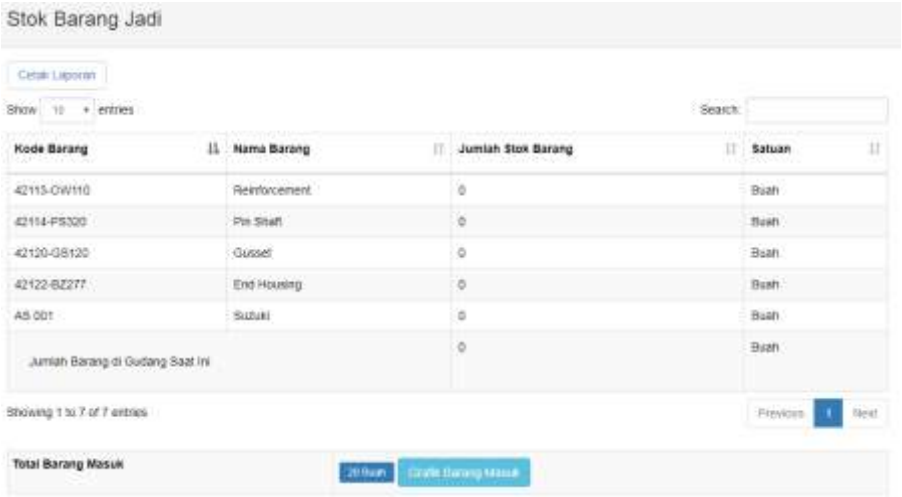


Gambar V.52 Halaman Data Surat Jalan

Sumber: Hasil Analisis (2018)

21. Halaman Stok Barang Jadi

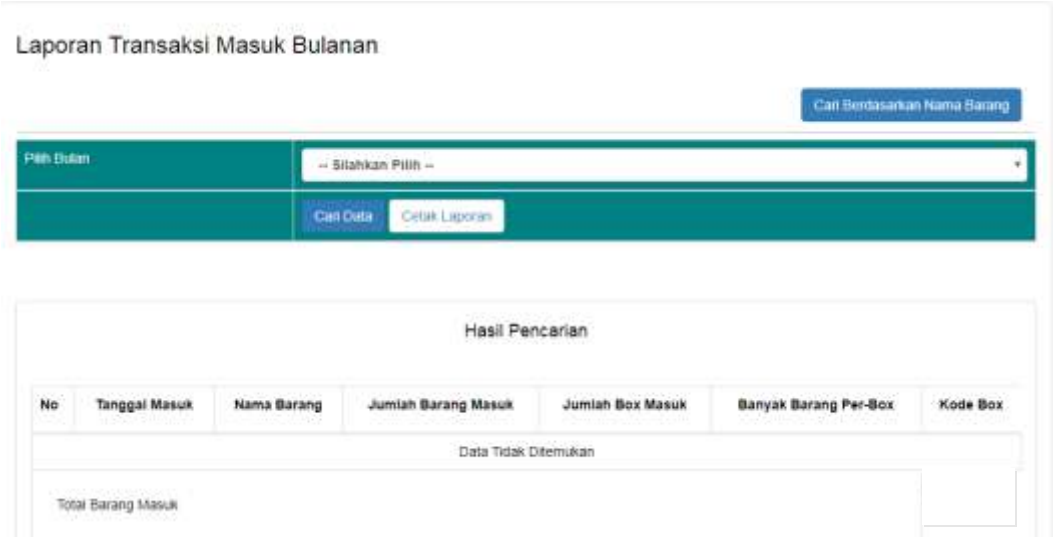
Halaman stok barang jadi digunakan untuk melihat jumlah barang yang terdapat dalam gudang. Adapun gambar dari halaman stok barang jadi dapat dilihat pada Gambar V.53.



Gambar V.53 Halaman Stok Barang Jadi
Sumber: Hasil Analisis (2018)

22. Halaman Data Laporan Transaksi Masuk

Halaman data laporan transaksi masuk digunakan untuk mencetak laporan detail transaksi barang yang masuk berdasarkan bulanan. Pengguna mencari terlebih dahulu data yang ada, lalu menekan tombol cetak untuk dicetak. Adapun gambar dari halaman data laporan transaksi masuk dapat dilihat pada Gambar V.54.



Gambar V.54 Halaman Data Laporan Transaksi Masuk
Sumber: Hasil Analisis (2018)

23. Halaman Data Laporan Transaksi Keluar

Halaman data laporan transaksi keluar digunakan untuk mencetak laporan detail transaksi barang yang keluar berdasarkan bulanan. Pengguna mencari terlebih dahulu data yang ada, lalu menekan tombol cetak untuk dicetak. Adapun gambar dari halaman data laporan transaksi keluar dapat dilihat pada Gambar V.55.

Gambar V.55 Halaman Data Laporan Transaksi Keluar
Sumber: Hasil Analisis (2018)

5.1.12 Pengujian *Black Box Testing*

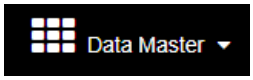
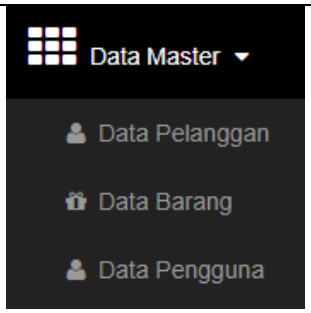
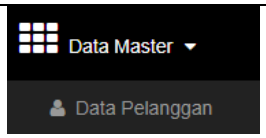
Black box testing digunakan untuk melakukan pengujian terhadap tombol tombol yang ada dalam suatu *interface* (antarmuka). Berikut ini adalah detail dari *black box testing* dari sistem informasi penyimpanan barang jadi dapat dilihat pada Tabel V.23.

Tabel V.23 *Black Box Testing*

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Membuka halaman *login*		Halaman *login* akan terbuka dan menampilkan *text box* untuk mengisi nama pengguna dan kata sandi		*Valid*

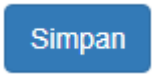
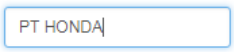
2.	Mengisi nama pengguna dan kata sandi pada halaman *login* lalu klik masuk		*Text box* pada kata sandi dan nama pengguna dapat di *input* dan ketika pengguna melakukan aksi klik Masuk, akan tampil halaman utama		*Valid*

Tabel V.23 *Black Box Testing* (lanjutan)

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
3.	Menampilkan sub menu *drop down* pada menu data master		Sub menu *drop down* dari menu data master akan tampil		*Valid*
4.	Menampilkan halaman data pelanggan		Halaman dari *sub menu* data pelanggan akan tampil		*Valid*

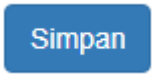
5.	Menambah data pelanggan		*Form* tambah data akan tampil ketika pengguna melakukan aksi klik tombol Tambah Data		*Valid*
6.	Mengisi *form* tambah data pelanggan		*Text box* pada *form* tambah data pelanggan dapat diisi		*Valid*

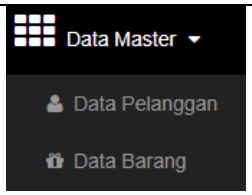
Tabel V.23 *Black Box Testing* (lanjutan)

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
7.	Menyimpan data pelanggan		Ketika pengguna melakukan aksi klik Simpan, maka data akan tersimpan dalam *database* dan tampil		*Valid*
8.	Mencari data		*Text box search* (cari)		*Valid*

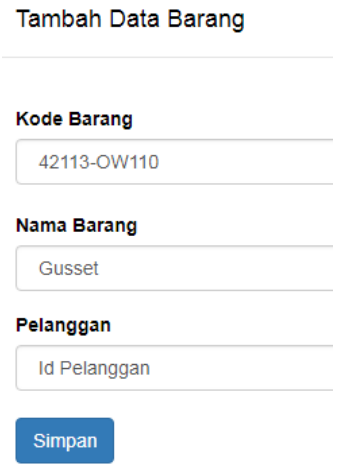
	pelanggan		dapat diisi dan menampilkan data yang di cari		
9.	Mengubah data pelanggan		Ketika pengguna melakukan aksi klik tombol *icon* ubah, maka *form* ubah akan tampil		*Valid*

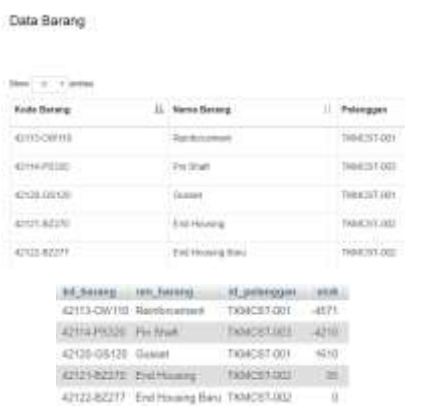
Tabel V.23 *Black Box Testing* (lanjutan)

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
10.	Menyimpan data yang telah diubah		Data yang diubah akan berubah ketika pengguna melakukan aksi klik tombol Simpan		*Valid*
11.	Menghapus data pelanggan		Data yang akan dihapus akan terhapus ketika pengguna klik tombol		*Valid*

			icon hapus		
12.	Menampilkan halaman dari *sub menu* data barang		Halaman dari *sub menu* data barang akan tampil		*Valid*

Tabel V.23 *Black Box Testing* (lanjutan)

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
13.	Menambah data barang		*Form* tambah data akan tampil ketika pengguna melakukan aksi klik tombol Tambah Data		*Valid*
14.	Mengisi data barang		*Text box* pada *form* tambah data barang dapat diisi		*Valid*

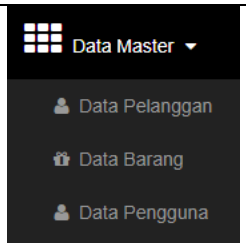
15.	Menyimpan data barang	Simpan	Ketika pengguna melakukan aksi klik		*Valid*

Tabel V.23 *Black Box Testing* (lanjutan)

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
16.	Mengubah data barang		Ketika pengguna melakukan aksi klik tombol *icon* ubah, maka *form* ubah akan tampil		*Valid*
17.	Menyimpan data yang telah diubah	Simpan	Data yang diubah akan berubah ketika pengguna melakukan aksi klik tombol Simpan		*Valid*

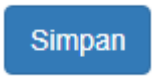
18.	Menghapus data barang		Data yang akan dihapus akan terhapus ketika pengguna klik tombol *icon* hapus		*Valid*

Tabel V.23 *Black Box Testing* (lanjutan)

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
19.	Mencari data barang		*Text box search* (cari) dapat diisi dan menampilkan data yang di cari		*Valid*
20.	Menampilkan halaman dari *sub menu* data pengguna		Halaman dari *sub menu* data pengguna akan tampil		*Valid*
21.	Menambah data pengguna		*Form* tambah data akan tampil ketika pengguna melakukan aksi klik tombol Tambah		*Valid*

			Data		

Tabel V.23 *Black Box Testing* (lanjutan)

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
22.	Mengisi data pengguna		*Text box* pada *form* tambah data pengguna dapat diisi		*Valid*
23.	Menyimpan data pengguna		Ketika pengguna melakukan aksi klik Simpan, maka data akan tersimpan dalam *database* dan tampil		*Valid*

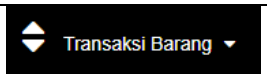
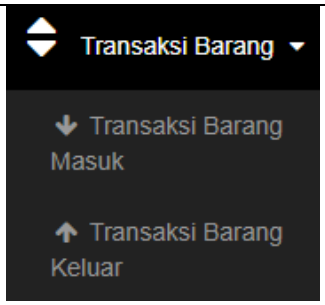
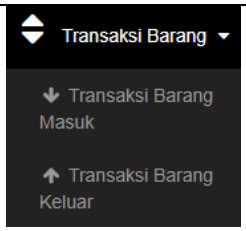
24.	Mengubah data pengguna		Ketika pengguna melakukan aksi klik tombol *icon* ubah, maka *form* ubah akan tampil	Ubah Data Pengguna Id Pengguna Username Nama Lengkap Password Jabatan Simpan	*Valid*

Tabel V.23 *Black Box Testing* (lanjutan)

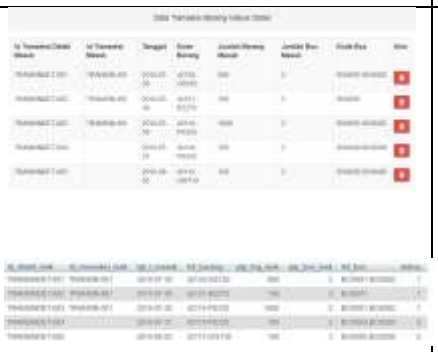
No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
25.	Menyimpan data yang telah diubah	Simpan	Data yang diubah akan berubah ketika pengguna melakukan aksi klik tombol Simpan	Data Pengguna	*Valid*
26.	Menghapus data pengguna		Data yang akan dihapus akan terhapus ketika pengguna klik tombol *icon* hapus	Data Pengguna	*Valid*
27.	Mencari data pengguna	Search:	*Text box search* (cari) dapat diisi dan	Search: indra	*Valid*

			menampilkan data yang di cari		

Tabel V.23 *Black Box Testing* (lanjutan)

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
28.	Menampilkan sub menu *drop down* pada menu Transaksi Barang		Sub menu *drop down* dari menu Transaksi Barang akan tampil		*Valid*
29.	Menampilkan halaman dari *sub menu* Transaksi Barang Masuk		Halaman dari *sub menu* data transaksi barang masuk akan tampil		*Valid*
30.	Menambah data barang detail		*Form* tambah data detail akan tampil ketika pengguna melakukan aksi klik tombol Tambah Data Detail		*Valid*

Tabel V.23 *Black Box Testing* (lanjutan)

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesim pulan
31.	Mengisi data barang masuk detail		*Text box* dalam *form* tambah data barang detail dapat diisi		*Valid*
32.	Menambah data barang detail masuk		Data barang masuk detail tersimpan dalam *database* lalu adanya tampilan data tabel		*Valid*
33.	Membuat data Transaksi Masuk		Ketika pengguna melakukan aksi klik tombol Buat Transaksi Masuk, *form* Transaksi Masuk akan tampil.		*Valid*

Tabel V.23 *Black Box Testing* (lanjutan)

No	Skenario	*Test Case*	Hasil Yang	Hasil Pengujian	Kesim

	Pengujian		Diharapkan		pulan
34.	Menyimpan data transaksi barang masuk	Simpan	Data akan tersimpan dalam *database* dan tampil dalam data tabel		*Valid*
35.	Melihat detail data transaksi barang masuk		Halaman detail berisikan data barang detail masuk akan tampil		*Valid*
36.	Tombol Kembali	Kembali	*Form* transaksi detail masuk pindah ke halaman *sub menu* transaksi masuk		*Valid*

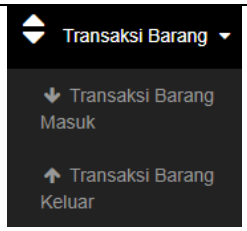
Tabel V.23 *Black Box Testing* (lanjutan)

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesim pulan

37.	Mengubah data transaksi masuk		Ketika pengguna melakukan aksi klik tombol *icon* ubah, maka *form* ubah akan tampil	Ubah Data Transaksi Masuk Id Transaksi Masuk TRANSIN-001 Tanggal Transaksi Masuk 2018-07-30 22:13:06 Departemen Produksi Simpan	*Valid*
38.	Menyimpan data yang telah diubah	Simpan	Data yang diubah akan berubah ketika pengguna melakukan aksi klik tombol Simpan	Transaksi Barang Masuk	*Valid*
39.	Mencari data transaksi masuk	Search:	*Text box search* (cari) dapat diisi dan menampilkan data yang di cari	Search: TRANSIN-001	*Valid*

Tabel V.23 *Black Box Testing* (lanjutan)

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan

40.	Menghapus data transaksi barang masuk		Data yang akan dihapus akan terhapus ketika pengguna klik tombol *icon* hapus		*Valid*
41.	Menampilkan halaman dari *sub menu* Transaksi Barang Keluar		Halaman dari *sub menu* data transaksi barang keluar akan tampil		*Valid*
42.	Mengeluarkan barang yang sudah disimpan		Jika tombol keluarkan barang di klik, maka akan menampilkan *form* transaksi detail keluar		*Valid*

Tabel V.23 *Black Box Testing* (lanjutan)

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan

43.	Mengirim barang yang sudah keluar dari dalam gudang		Jika tombol mengirim barang di klik, maka akan menampilkan data surat jalan. Dan cetak surat jalan		*Valid*
44.	Mengisi data barang keluar detail		*Text box* dalam *form* tambah data barang detail dapat diisi		*Valid*
45.	Menambah data barang detail keluar		Data barang keluar detail tersimpan dalam *database* lalu adanya tampilan data tabel		*Valid*

Tabel V.23 *Black Box Testing* (lanjutan)

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan

46.	Membuat data Transaksi Keluar	Buat Transaksi Keluar	Ketika pengguna melakukan aksi klik tombol Buat Transaksi Keluar, *form* Transaksi Keluar akan tampil.		*Valid*
47.	Menyimpan data transaksi barang keluar	Simpan	Data akan tersimpan dalam *database* dan tampil dalam data tabel		*Valid*
48.	Mengubah data transaksi keluar		Ketika pengguna melakukan aksi klik tombol *icon* ubah, maka *form* ubah akan tampil		*Valid*

Tabel V.23 *Black Box Testing* (lanjutan)

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
49.	Menyimpan data yang telah diubah	Simpan	Data yang diubah akan berubah		*Valid*

			ketika pengguna melakukan aksi klik tombol Simpan		

Sumber: Hasil Analisis (2018)

5.1.13 Implementasi Sistem

Tahap ini adalah tahap pengkodean program menggunakan bahasa pemrograman PHP dengan *framework CodeIgniter* dan dibantu dengan *Sublime Text 3* sebagai aplikasi *editor*. Setiap *interface* berisikan kode program agar program dapat dijalankan sesuai dengan fungsinya. Untuk mendukung kebutuhan implementasi sistem diperlukan suatu spesifikasi perangkat lunak (*software*) maupun perangkat keras (*hardware*). Adapun spesifikasiknya sebagai berikut:

1. Analisis Kebutuhan *Software*
   a. Sistem Operasi : *Microsoft Windows* 10 *Home*
   b. *Database Server* : MariaDB 10.1.31
   c. *Web Server* : PHP 5.6.35
   d. Bahasa Pemrograman : PHP
   e. *Framework* : *CodeIgniter* 3.1.4
2. Analisis Kebutuhan *Hardware*

Tabel V.24 Spesifikasi *Hardware*

Nama	Spesifikasi
Processor	*Processor Intel® Core* i7-8750H®
RAM	8GB DDR4, 2666MHz

Tabel V.24 Spesifikasi *Hardware* (lanjutan)

Hard Drive	1TB SSHD
System Type	32-bit *Operating System*
Tipe Grafis	NVIDIA® GeForce®
Resolusi Layar	15.6-inch *Full* HD (1920x1080) TN Panel, 120Hz, 3ms, 94% NTSC

Koneksi	Wifi, *Bluetooth*
Webcam	720P HD *Web Camera*
Baterai	4 *Cells 64* Whrs *Battery*

BAB VI
KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil pengumpulan, pengolahan dan analisis data yang telah dilakukan dalam penelitian Tugas Akhir ini dapat disimpulkan bahwa:

1. Dengan adanya sistem informasi penyimpanan barang jadi ini, dapat membantu Operator *Warehouse* melakukan pencatatan aktivitas keluar/masuknya barang jadi dari dalam gudang sehingga barang dapat dikontrol dengan baik.
2. Dengan adanya sistem informasi penyimpanan barang jadi, dapat membantu Operator *Warehouse* untuk membuat laporan bulanan menjadi lebih cepat dengan mengambil data dari aktivitas keluar/masuknya barang dari dalam gudang.
3. Dengan adanya sistem informasi penyimpanan barang jadi, dapat membantu Operator *Warehouse* mengetahui barang jadi mana saja yang harus dikeluarkan dari dalam gudang sehingga untuk meminimalisir adanya barang jadi yang terlalu lama disimpan dalam gudang.

6.2 Saran

Saran yang diberikan untuk pengembangan sistem informasi penyimpanan barang jadi ini selanjutnya adalah sebagai berikut:

1. Sebaiknya dilakukan sosialisasi terlebih dahulu kepada Departemen terkait mengenai sistem informasi penyimpanan barang jadi ini agar para karyawan dalam Departemen tersebut dapat menggunakannya dengan baik.
2. Diharapkan untuk kedepannya sistem informasi penyimpanan barang jadi ini dapat dikembangkan menjadi sistem yang lebih baik dari sebelumnya.

DAFTAR PUSTAKA

Adelheid, A., 2013, *1 Hari Menjadi Hacker*, Mediakita, Jakarta.

Adelheid, A. & Nst, Khairil., 2012, *Buku Pintar Menguasai PHP MySQL*, Mediakita, Jakarta.

Ahmar, S. A., 2013, *Modifikasi Template CMS Lokomedia*, Garudhawaca, Yogyakarta.

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

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

Binarto, S., 2012, *Tip & Trik Membuat Program Penjualan Menggunakan Visual Basic 6.0*, Mediakita, Jakarta.

CodeIgniter, *CodeIgniter*, dilihat 8 Agustus 2018, dari www.codeigniter.com/user_guide/index.html.

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

Dennis, A., Wixom, Barbara Haley, & Tegarden, D., 2015, *System Analysis & Design. An Object-Oriented Approach with UML (Fifth Edition)*, Wiley, USA.

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

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

Enterprise, J., 2017, *PHP Komplet*, PT Elex Media Komputindo, Jakarta.

Enterprise,J., 2014, *MySql Untuk Pemula*, PT Elex Media Komputindo, Jakarta.

EMS, Tim., 2016, *PHP 5 dari Nol*, PT Elex Media Komputindo, Jakarta.

Fauzi, R. A., 2017, *Sistem Informasi Akuntansi (Berbasis Akuntansi)*, Deepublish, Yogyakarta.

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

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

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

Jogiyanto, 2005, *Sistem Teknologi Informasi*, CV Andi Offset, Yogyakarta.

Jurnal Id, *Mengenal Metode Fifo*, dilihat 3 Mei 2018, dari Jurnal.id/blog/2018/mengenal-metode-fifo-pada-jurnal.

KBBI, *Kamus Besar Bahasa Indonesia*, dilihat 11 Februari 2018, dari Kbbi.web.id.

Komputer, Wahana., 2014, *Membuat Aplikasi SMS Gateway dengan Codeigniter*, PT Elex Media Komputindo, Jakarta.

Komputer, Wahana, 2010, *Paduan Belajar MySQL Database Server*, Mediakita, Jakarta.

Leod, Mc & P. Schell, G., 2012, *Sistem Informasi Manajemen Edisi Sepuluh*, Salemba Empat, Jakarta.

Manohar & Appaiah, 2017, 'Stabilization of FIFO system and Inventory Management'.

MariaDB, *Definisi Mariadb*, dilihat 17 Juli 2018, dari https://mariadb.com/products/why-mariadb.

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

Master Pendidikan, *19 Pengertian Metodologi Penelitian Menurut Para Ahli*, dilihat 14 Mei 2018, dari masterpendidikan.com/2016/03/19-pengertian-metodologi-penelitian-menurut-para-ahli.html

Muharto & Ambarita A., 2016, *Metode Penelitian Sistem Informasi: Mengatasi Kesulitan Mahasiswa dalam Menyusun Proposal Penelitian*, Deepublish, Yogyakarta.

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

Muslihudin, M. & Oktafianto., 2016, *Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML*, CV Andi Offset, Yogyakarta.

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

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

Pamungkas, Canggih A., 2017, *Pengantar dan Impelementasi Basis Data*, Deepublish, Yogyakarta.

Poros Ilmu, *Penyimpanan Barang di Gudang dengan Sistem FIFO,* dilihat 3 Mei 2018, dari Porosilmu.com/2016/11/penyimpanan-barang-di-gudang-dengan.

Prasetyo & Ambarwati, 2017, 'Perancangan Sistem Informasi Material Produk Kemasan Menggunakan Metode FIFO Pada PT. CRS'.

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

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

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

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

Samryn, L. M., 2015, *Akuntansi Manajemen: Informasi Biaya untuk Mengendalikan Aktivitas Operasi & Investasi (Edisi Revisi)*, Prenada Media Group, Jakarta.

Sattar., 2017, *Buku Ajar Pengantar Bisnis*, Deepublish, Yogyakarta.

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

Sobir & Firmansyah., 2014, *Berkebun Melon Unggul Edisi Revisi*, Penebar Swadaya, Jakarta.

Sriyanto, A. K., 2008, *Cara Mudah Menghadapi Ujian Nasional 2008*, Grasindo, Jakarta.

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

Sutabri, T., 2012, *Analisis Sistem Informasi*, CV Andi Offset, Yogyakarta.

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

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

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

Wikipedia Indonesia, Analisis, dilihat 11 Februari 2018, Id.m.wikipedia.org/wiki/Analisis.

Wikipedia Indonesia, Gudang, dilihat 14 November 2017, dari Id.m.wikipedia.org/wiki/Gudang.

Wikipedia Indonesia, *CodeIgniter*, dilihat 8 Mei 2018, dari Id.wikipedia.org/wiki/CodeIgniter.

Wikipedia Indonesia, FIFO, dilihat 8 Mei 2018, dari Id.wikipedia.org/wiki/Akuntansi_FIFO_dan_LIFO.

Winarno, E. & Zaki, A., 2013, *Buku Sakti Pemrograman PHP*, PT Elex Media Komputindo, Jakarta.

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

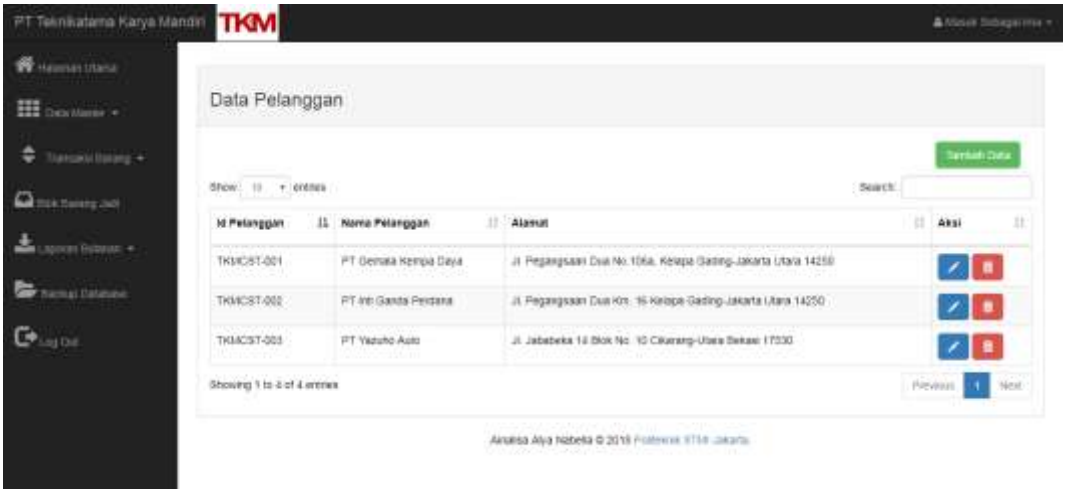
LAMPIRAN A

TAMPILAN PROGRAM

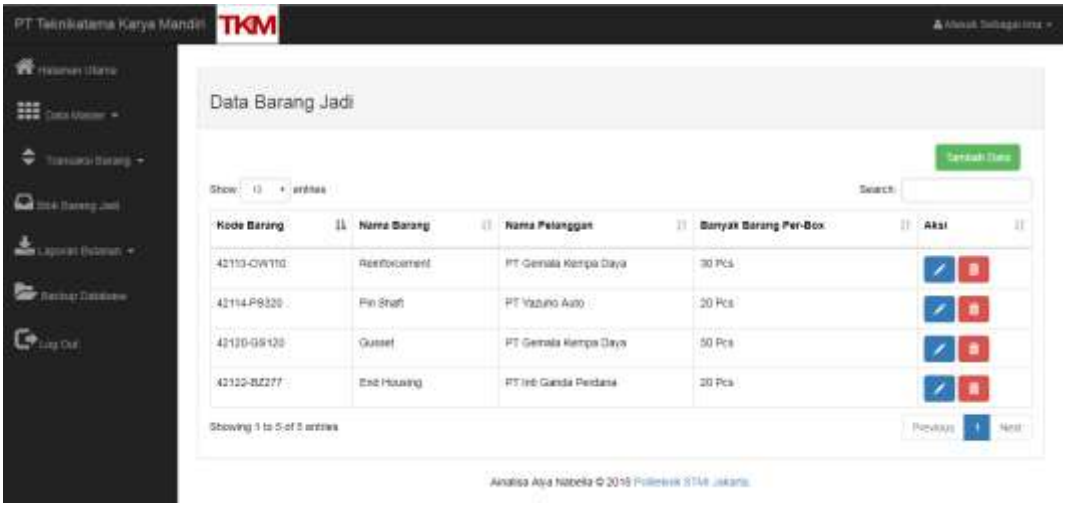
1. **Halaman *Login***

2. **Halaman Utama**

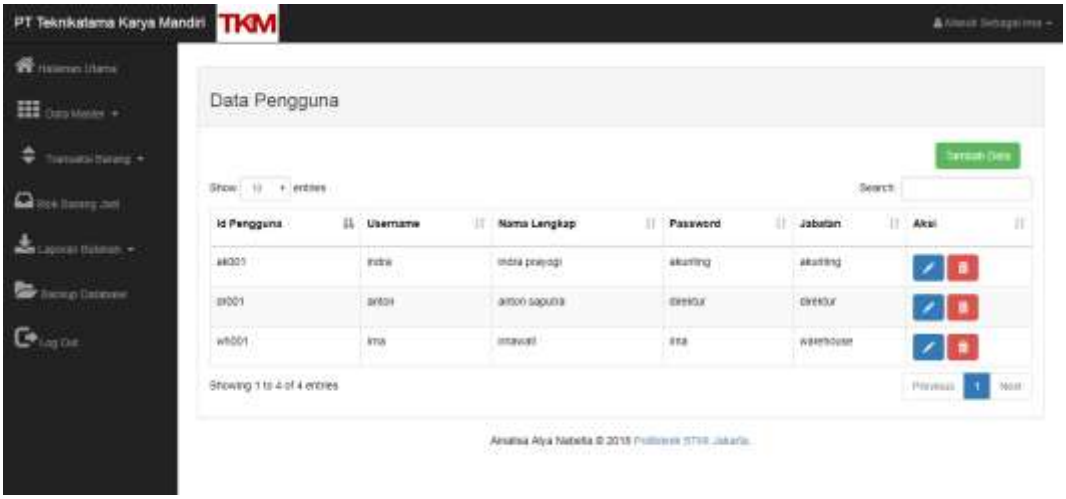
3. **Data Master Pelangan**



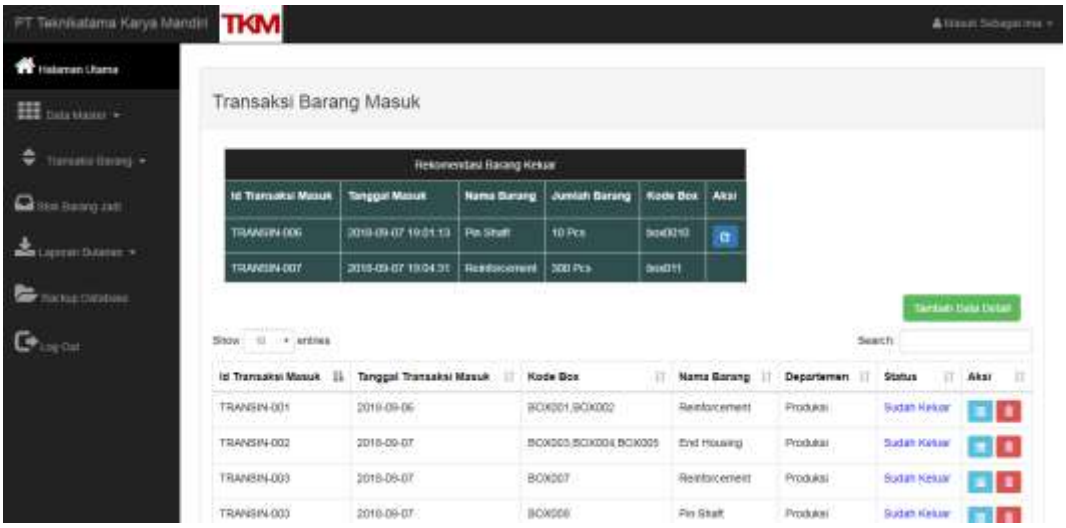
4. Data Master Barang



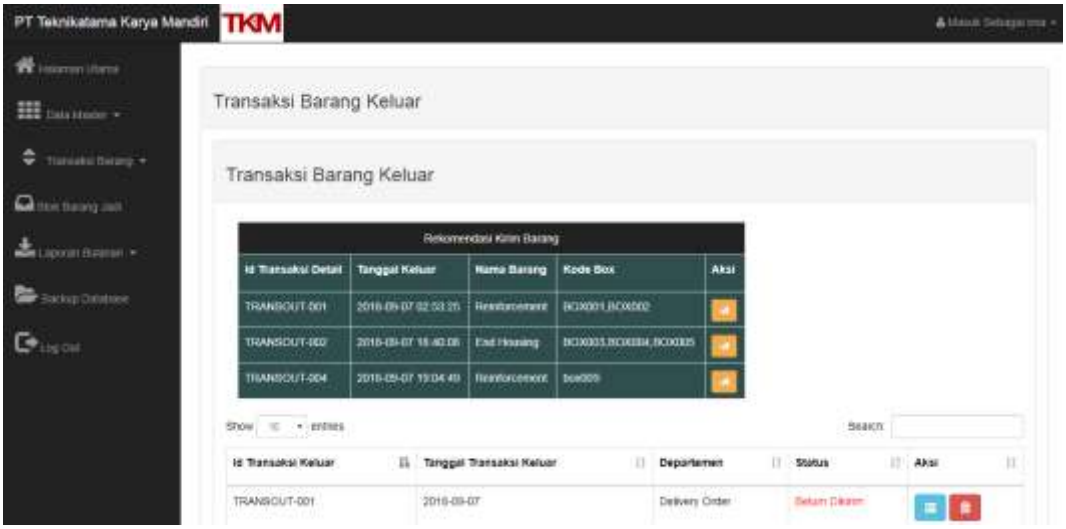
5. Data Master Pengguna



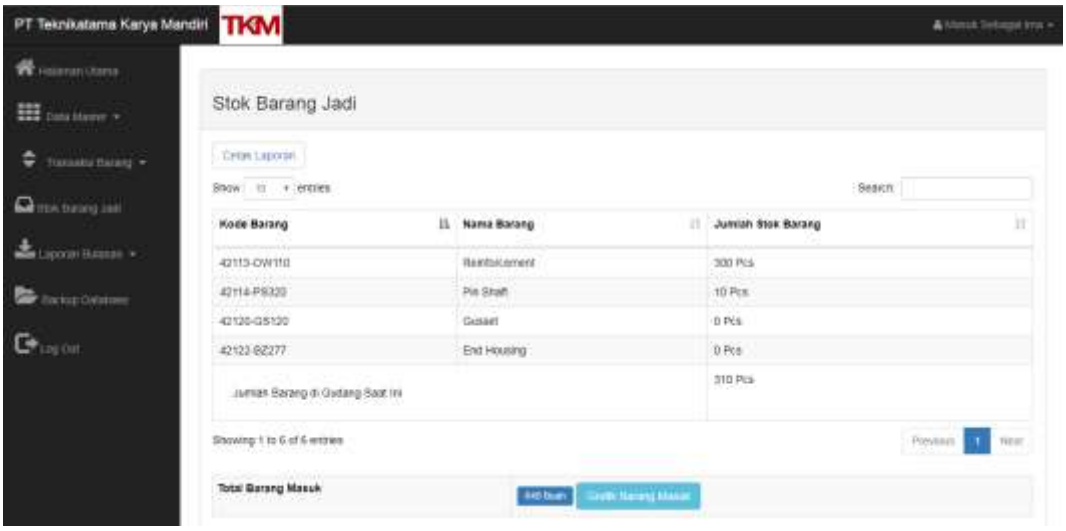
6. Transaksi Barang Masuk



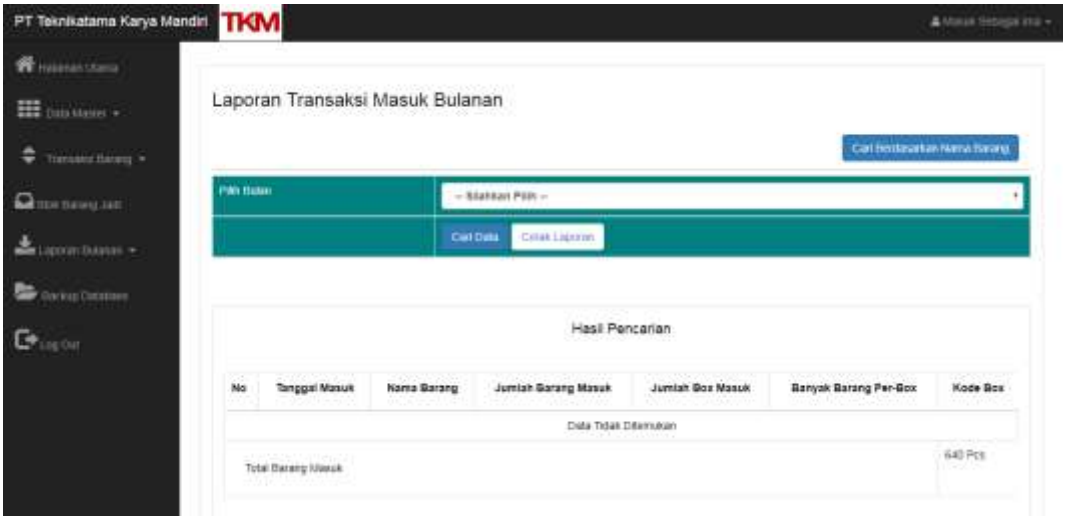
7. Transaksi Barang Keluar



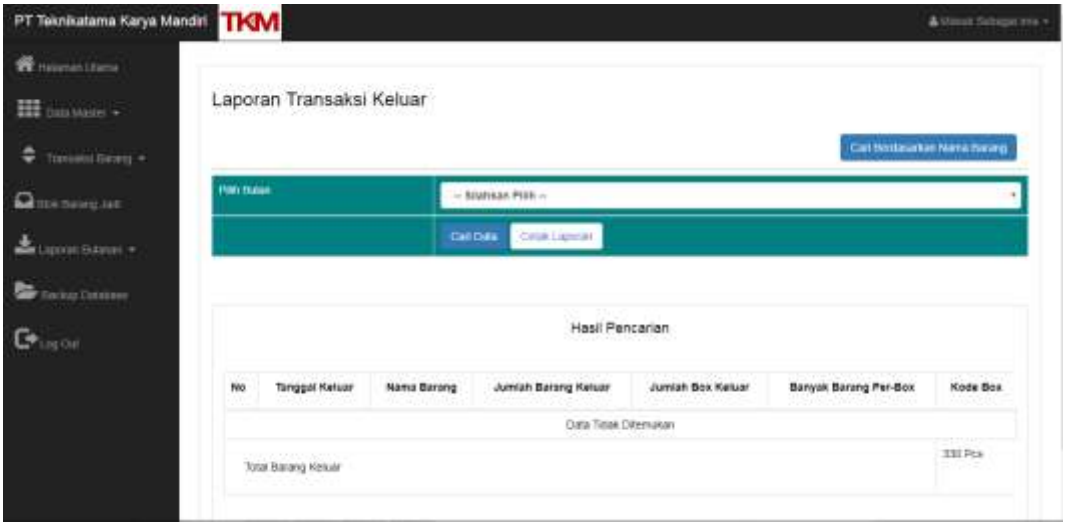
8. Stok Barang Jadi



9. Laporan Transaksi Masuk



10. Laporan Transaksi Keluar



11. Laporan Transaksi Masuk

9/12/2018 localhost/ali/index.php/laporan/cetak_transaksimasuk/9

PT Teknikatama Karya Mandiri

Jalan Raya Bahkilong No. 88 Sukadami Cikarang-Selatan

Laporan Transaksi Masuk Bulan September

No	Tanggal Masuk	Nama Barang	Jumlah Barang Masuk	Jumlah Box Masuk	Banyak Barang Per-Box	Kode Box
1	2018-09-06 07:23:11	Reinforcement	100 Pcs	4 Box	30 Pcs	BOX001,BOX002
2	2018-09-07 02:12:27	End Housing	70 Pcs	4 Box	20 Pcs	BOX003,BOX004,BOX005
3	2018-09-07 18:44:24	Reinforcement	0 Pcs	1 Box	30 Pcs	BOX007
4	2018-09-07 18:44:33	Pin Shaft	70 Pcs	4 Box	20 Pcs	BOX008
5	2018-09-07 18:59:46	Reinforcement	90 Pcs	4 Box	30 Pcs	box009
6	2018-09-07 19:01:13	Pin Shaft	10 Pcs	1 Box	20 Pcs	box0010
7	2018-09-07 19:04:31	Reinforcement	300 Pcs	11 Box	30 Pcs	box011
Total Barang Masuk						640 Pcs

12. Laporan Transaksi Keluar

9/12/2018 localhost/ali/index.php/laporan/cetak_transaksikeluar/9

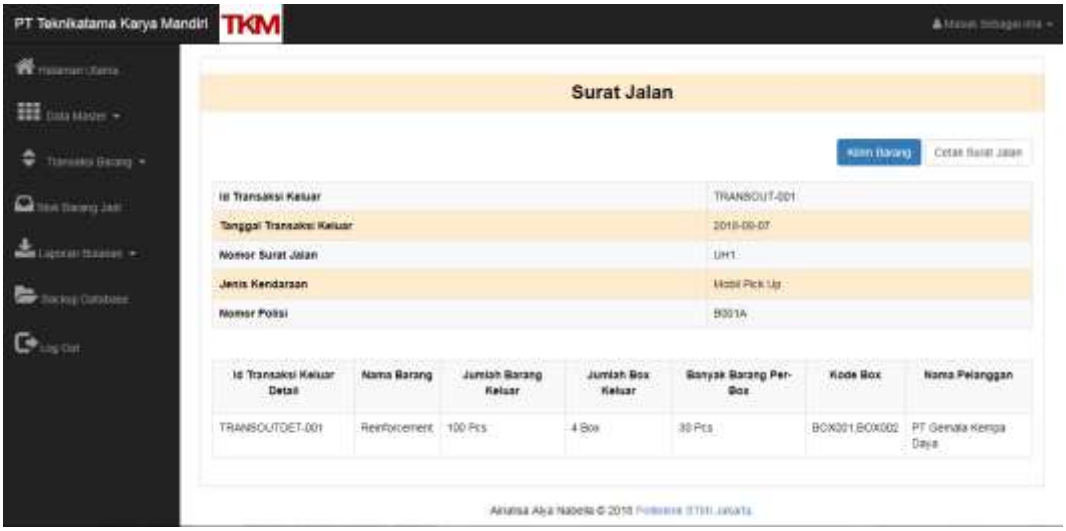
PT Teknikatama Karya Mandiri

Jalan Raya Bahkilong No. 88 Sukadami Cikarang-Selatan

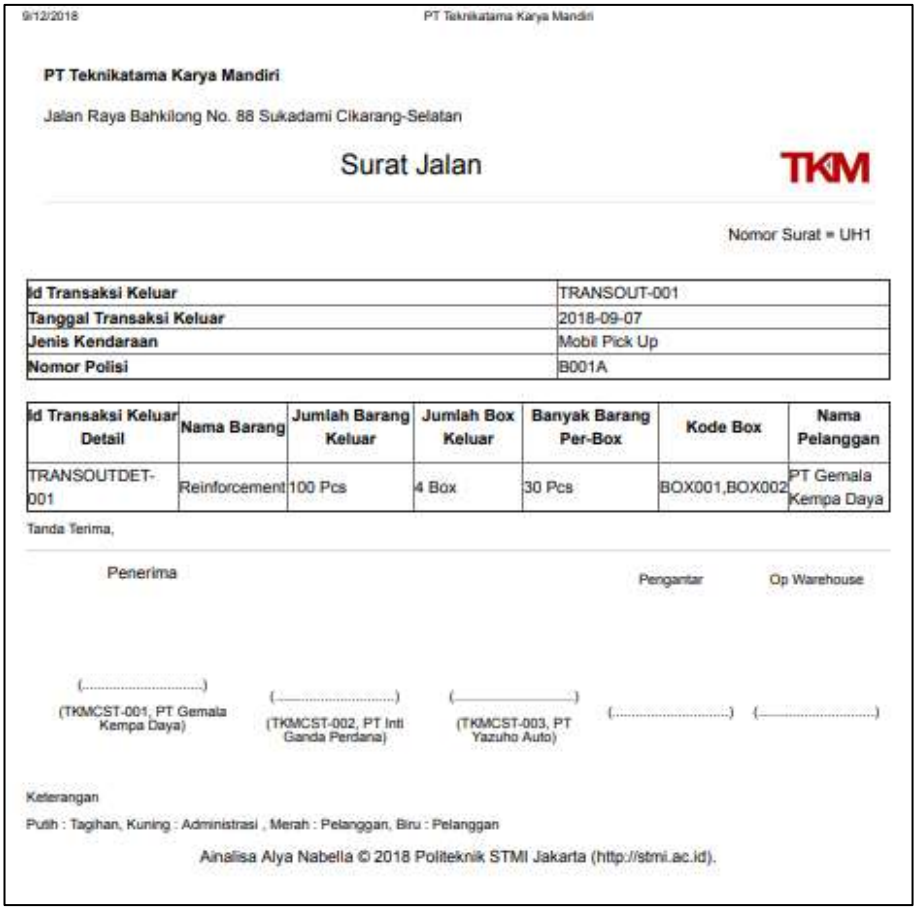
Laporan Transaksi Keluar Bulan September

No	Tanggal Keluar	Nama Barang	Jumlah Barang Keluar	Jumlah Box Keluar	Banyak Barang Per-Box	Kode Box	Nama Pelanggan
1	2018-09-07 02:53:25	Reinforcement	100 Pcs	4 Box	30 Pcs	BOX001,BOX002	PT Gemala Kempa Daya
2	2018-09-07 18:40:08	End Housing	70 Pcs	4 Box	20 Pcs	BOX003,BOX004,BOX005	PT Inti Ganda Perdana
3	2018-09-07 18:45:20	Reinforcement	0 Pcs	1 Box	30 Pcs	BOX007	PT Gemala Kempa Daya
4	2018-09-07 18:45:26	Pin Shaft	70 Pcs	4 Box	20 Pcs	BOX008	PT Yazuho Auto
5	2018-09-07 19:04:49	Reinforcement	90 Pcs	4 Box	30 Pcs	box009	PT Gemala Kempa Daya
Total Barang Keluar							330 Pcs

13. Surat Jalan



14. Cetak Surat Jalan



9/12/2018 PT Teknikatama Karya Mandiri

PT Teknikatama Karya Mandiri

Jalan Raya Bahkilong No. 88 Sukadami Cikarang-Selatan

Surat Jalan

TKM

Nomor Surat = UH1

Id Transaksi Keluar	TRANSOUT-001
Tanggal Transaksi Keluar	2018-09-07
Jenis Kendaraan	Mobil Pick Up
Nomor Polisi	B001A

Id Transaksi Keluar Detail	Nama Barang	Jumlah Barang Keluar	Jumlah Box Keluar	Banyak Barang Per-Box	Kode Box	Nama Pelanggan
TRANSOUTDET-001	Reinforcement	100 Pcs	4 Box	30 Pcs	BOX001,BOX002	PT Gemala Kempa Daya

Tanda Terima,

Penerima Pengantar Op Warehouse

(.....) (TKMCST-001, PT Gemala Kempa Daya)

(.....) (TKMCST-002, PT Inti Ganda Perdana)

(.....) (TKMCST-003, PT Yazuho Auto)

(.....) (.....)

Keterangan

Putih : Tagihan, Kuning : Administrasi , Merah : Pelanggan, Biru : Pelanggan

Ainalisa Alya Nabella © 2018 Politeknik STMI Jakarta (http://stmi.ac.id).

15. Detail Transaksi Barang Jadi Masuk

16. Detail Transaksi Barang Jadi Keluar

LAMPIRAN B

KODE PROGRAM

1. Data *Master* Barang

a. *Controller* Barang

```
<?php
/**
*
*/
class barang extends CI_Controller
{
	function __construct(){
		parent::__construct();
		$this->load-
>model(array('model_barang','model_datapelanggan'));
	}

	function index()
	{
		$data['record']=$this->model_barang->tampil_data();
		$data['idpel'] = $this->model_datapelanggan-
>tampil_data()->result();
		$data['pelanggan'] = $this->model_datapelanggan-
>tampil_pelanggan()->result();
		$this->template->content->view('barang/lihat_data',$data);
		$this->template->publish();
	}



	function post(){
```

```
            if (isset($_POST['submit'])) {

                $kd_barang          =       $this->input-
>post('kd_barang');
                $nm_barang          =       $this->input-
>post('nm_barang');
                $id_pelanggan=      $this->input-
>post('id_pelanggan');
                $data               =
    array('kd_barang'=>$kd_barang,

    'nm_barang'=>$nm_barang,

    'id_pelanggan'=>$id_pelanggan,
                                                                    );

                $this->model_barang->post($data);
                redirect('barang');
            }
            else {
                $this->template->content-
>view('barang/form_input');
                $this->template->publish();
            }
    }

    function delete (){
            $kd_barang=   $this->uri->segment(3);
            $this->model_barang->delete($kd_barang);
            redirect('barang');
```

```
    }

    function edit (){
            if (isset($_POST['submit'])) {

                    $kd_barang          =       $this->input-
>post('kd_barang');
                    $nm_barang          =       $this->input-
>post('nm_barang');
                    $id_pelanggan       =       $this->input-
>post('id_pelanggan');
                    $data               =
    array('kd_barang'=>$kd_barang,

    'nm_barang'=>$nm_barang,

    'id_pelanggan'=>$id_pelanggan,

    );

                    $this->model_barang->edit($data,$kd_barang);
                    redirect('barang');
            }
            else {
                    $id=    $this->uri->segment(3);
                    $this->load->model('model_barang');
                    $data['barang']=        $this->model_barang-
>tampil_data()->result();
                    $data['record']=        $this->model_barang-
>get_one($id)->row_array();
```

```
                        $this->template->content-
>view('barang/form_edit',$data);
                        $this->template->publish();
                }
        }
}
```

b. Model Barang

```
<?php
/**
*
*/
class Model_barang extends CI_Model
{



        function tampil_data()
        {
                return $this->db->get('tb_barang');
        }

        function post ($data){

                $this->db->insert('tb_barang',$data);
        }

        function delete($kd_barang){
                $this->db->where('kd_barang',$kd_barang);
                $this->db->delete('tb_barang');
```

```
        }

        function edit($data, $kd_barang){

                $this->db->where('kd_barang',$kd_barang);
                $this->db->update('tb_barang',$data);
        }

        function get_one($id){

                $param= array('kd_barang'=>$id);

                return $this->db->get_where('tb_barang',$param);
        }
}
```

c. *View* Barang

```
<div class="panel panel-default" id="panel">
<div class="panel-heading"><b><h3> Data Barang Jadi
</h3></b></div>
 <div class="panel-body">

<table  class="table table-bordered" cellspacing="0" width="100%"
id="example1">

        <thead>
        <tr>
                <th> <p> Kode Barang</th> </p>
                <th> <p> Nama Barang</th> </p>
                <th> <p> Pelanggan</th> </p>
                <th> <p> Aksi</th> </p>
```

```
        </tr>
        </thead>
        <tr>
        <?php
        foreach ($record->result() as $d)
        {
                echo "<tr>
                <td>$d->kd_barang</td>
                <td>$d->nm_barang</td>
                <td>$d->id_pelanggan</td>

                <td>".anchor('barang/edit/'.$d->kd_barang,' ',array('class'=>
'btn btn-primary glyphicon glyphicon-pencil'))."  "
                .anchor('barang/delete/'.$d->kd_barang,' ',array('class'=>
'btn btn-danger glyphicon glyphicon-trash','onClick'=>"return
confirm('Apakah Anda Yakin Menghapus Data Ini?')")))." </td>
                </tr>";

        }
        ?>
        <div class="col-md-12" style="text-align:right; margin-bottom:
10px; margin-top: -5px; ">



<button class="btn btn-success" type="button" data-
target="#ModalAdd" data-toggle="modal"> Tambah Data</button>

         </div>

</table>
```

```
</div>

<!-- ini modal -->
<div id="ModalAdd" class="modal fade" tabindex="-1"
role="dialog">

                        <div class="modal-dialog">
                                <div class="modal-content">
                                        <div class="modal-header">
                                                <button type="button"
class="close" data-dismiss="modal" aria-label="Close"><span aria-
hidden="true">&times;</span></button>
                                                <h4 class="modal-
title">Tambah Data Barang</h4>
                                        </div>

                                        <div class="modal-body">

<?php
echo form_open('barang/post', "name='modal_popup'");
?>
<table class="table table-bordered">
        <tr>

        <br>
        <div class="form">
                        <label>Id Pelanggan</label>
                    <select name="id_pelanggan" class="form-control"
id="id_pelanggan"  onChange="changeValue(this.value)">
                        <!-- <option value="0" disabled="disabled"
selected/> Id Pelanggan</option> -->
```

```
                <?php
                        $jsArray = "var dtBB = new Array();\n";
                        echo "<option value='0' selected/> --
Pelanggan -- </option>";
                        foreach ($pelanggan as $pel) {
                                echo "<option value='$pel-
>id_pelanggan'> $pel->id_pelanggan,$pel->nm_pelanggan
</option>";

                                $jsArray .= "dtBB['" . $pel-
>id_pelanggan . "'] = {kd_barang:'" . addslashes($pel->kd_barang)
."'};\n";
                        }
                                ?>
            </select>
    </br>

    <div class="form">
    <label>Kode Barang</label>
    <input type="text" class="form-control" name="kd_barang"
id="kd_barang" placeholder="Kode Barang" required="">
    </div>
    </br>

    <div class="form">
    <label>Nama Barang</label>
    <input type="text" class="form-control" name="nm_barang"
placeholder="Nama Barang" required="">
    </div>

    <br>
```

```
        <div class="box-footer">
                        <button type="submit" class ="btn btn-primary"
name="submit">Simpan</button></td>
        </div>


        <script type="text/javascript">
 <?php echo $jsArray; ?>
 function changeValue(id_pelanggan){
   // document.getElementById('kd_barang').value =
dtBB[kd_box].kd_barang;
   document.getElementById('kd_barang').value =
dtBB[id_pelanggan].kd_barang;
  // document.getElementById('qty_brg_keluar').value =
dtBB[kd_box].qty_brg_msk;
   // document.getElementById('banyak').value =
dtBB[kd_material].jumlah;
 };
</script>
</table>
</form>
</div>
</div>
                                </div>
                                </div>
```

LAMPIRAN C

WAWANCARA DENGAN OPERATOR WAREHOUSE

Peneliti : "selamat pagi bu.. saya ingin melakukan wawancara terkait dengan penyimpanan barang jadi dalam gudang. Apakah ibu bersedia?."

Op W : "Baik. Akan saya jelaskan mengenai prosedur penyimpanan barang yang dalam gudang yang saya tangani. Jadi di dalam gudang ada 3 jenis barang, NG, RM, sama FG. NG itu kumpulan barang yang sudah jadi dengan kondisi barang yang bagus. FG itu kumpulan barang yang sudah jadi tapi kondisinya reject atau jelek. RM itu kumpulan bahan baku."

Peneliti : "Oke bu.."

Op W : "Untuk prosedur barang yang ada di gudang sendiri, seperti ini. PT TKM mendapatkan bahan baku dari suppliernya sebelum diolah. Contoh nih, gusset sekali minta 1300pcs. Sisanya nyusul. Hal tersebut dikarenakan mobil truknya masih kecil dan kelebihan beban sekali kirim. Setelah barang dari supplier sampai, dikasih ke gudang RM. Terus nanti operator produksinya mengajukan PKB (permintaan kebutuhan barang) ke operator warehouse. Jika sudah di ACC, lalu barang di produksi. Nanti kalau waktu pengirimannya tiba, saya buatkan surat jalan untuk delivery ordernya. Tapi ngga semua barang dikirim. Jumlah max barang diangkut di mobil pick up 1200pcs. Sementara sehari kita produksi bisa kena 2000pcs. Jadi sisanya disimpan dulu di gudang buat nunggu waktu pengiriman. Pengiriman biasanya dari hari senin sampai jumat."

Peneliti : "Untuk barang sekali keluar dari gudang bisa berapa pcs ya Bu?"

Op W : "300pcs. Jenis barang kita kan ada 4. 300pcs x 4 = 1200pcs. Jumlah max yang bisa diangkut ke mobil pick up."

Peneliti : "Oh ya Bu.. untuk teknis penyimpanannya bagaimana ya Bu? Saya lihat banyak sekali tumpukan box dalam gudang."

Op W : "Kita simpan barang ke gudang dalam bentuk box-box. Jadi masing-masing box ada kartu tagnya dari PT mana dan ada nama barangnya.

Tapi kita letakkan di area menurut jenisnya. Misal, barang gusset, ditempatkan di area khusus gusset."

Peneliti : "Dalam satu box ada berapa pcs barang ya Bu?"

Op W : "Untuk gusset 50pcs, pin shaft 20 pcs, end housing 20 pcs, reinforcement 30 pcs. 300/5 = 6. 300/20 = 15. 300/20=15. 300/30 = 10. Totalkan saja ada 46 box dalam sekali pengiriman."

Peneliti : "Untuk daftar stoknya memang kosong ya Bu?"

Op W : "Oh ya. Karna saya biasa nulis di note. Laporan bulanan juga belum ada."

Peneliti : "Oke bu.. Terimakasih banyak untuk informasi yang diberikan. Selamat bekerja kembali"

LAMPIRAN D
OBSERVASI

Tanggal	Pengamat	Departemen Observasi	Lokasi Observasi	Kegiatan Observasi	Dokumen
07-08-2017 s.d. 11-08-2017	Ainalisa Alya N	Ibu Irna (Operator *Warehouse*)	Ruang Kantor PT Teknikatama Karya Mandiri	Proses penyimpanan barang jadi	Lembar daftar stok (terlampir dalam BAB IV)
14-08-2017 s.d. 18-10-2017	Ainalisa Alya N	Departemen *Warehouse*	Ruang Kantor PT Teknikatama Karya Mandiri	Melihat barang jadi yang berada di dalam gudang	Kartu Tanda *Box* (terlampir dalam BAB IV)
21-08-2017 s.d. 25-08-2017	Ainalisa Alya N	Ibu Irna (Operator *Warehouse*)	Ruang Kantor PT Teknikatama Karya Mandiri	Proses penyimpanan hingga pengeluaran barang jadi	Lembar surat jalan (terlampir dalam BAB IV)

Penulis melakukan observasi selama satu bulan dimulai dari tanggal 31 Juli 2017 sampai dengan 28 Agustus 2017. Observasi dilakukan di PT Teknikatama Karya Mandiri. Terletak di Jalan Raya Bahkilong No.88 Desa Sukadami Cikarang-Selatan. PT Teknikatama Karya Mandiri menghasilkan produk *sparepart* untuk kendaraan roda empat. Adapun nama produk barang jadi tersebut adalah *Gusset, Pin Shaft, Reinforcement,* dan *End Housing*. PT Teknikatama Karya Mandiri mempunyai gudang untuk menyimpan barang jadi

yang telah diproduksi untuk menunggu waktu pengiriman. Adapun alur penyimpanan barang jadi pada PT Teknikatama Karya Mandiri adalah, hal yang pertama barang jadi telah diproduksi diperiksa kualitasnya oleh pihak *Quality Control*. Setelah itu barang jadi dikirim oleh pihak *Quality Control* masuk ke dalam gudang. Barang yang masuk ke dalam gudang tak tentu jumlahnya. Namun, total barang jadi yang diproduksi dalam sehari dapat mencapai 2000pcs. Ketika barang masuk ke dalam gudang, barang jadi ditempatkan dalam suatu *box-box* sesuai dengan nama barangnya. Dalam *box-box* tersebut sudah tercantum kartu tanda *box* untuk membedakan antar nama barang. Barang jadi yang masuk ke dalam gudang, dicatat oleh Operator *Warehouse* dalam lembar daftar stok. Selanjutnya, barang jadi disimpan didalam gudang. Sistem penyimpanannya, diletakkan menurut nama barangnya. Dikarenakan ruangan gudang kecil, jika *box* sudah banyak, maka akan ditumpuk. Waktu pengiriman pada sore harinya. Ketika waktu pengiriman tiba, Operator *Warehouse* mencatat data barang jadi yang keluar dalam daftar stok dan membuat surat jalan untuk *Delivery Order*. Berikut adalah gambar-gambar dari hasil observasi yang telah dilakukan.

Reinforcement	*Gusset*	*Pin Shaft*	*End Housing*	Ruangan Gudang