

No. Doc: 6171  
Copy : 1

D 658.785 02855  
cli  
R

**RANCANG BANGUN SISTEM INFORMASI  
PENDISTRIBUSIAN UNIT PADA DEPARTEMEN  
LOGISTIC 2W DENGAN ALGORITMA DIJKSTRA  
MENGUNAKAN FUNGSI EUCLIDEAN  
DISTANCE DI PT SUZUKI INDOMOBIL MOTOR  
PLANTI TAMBUN**

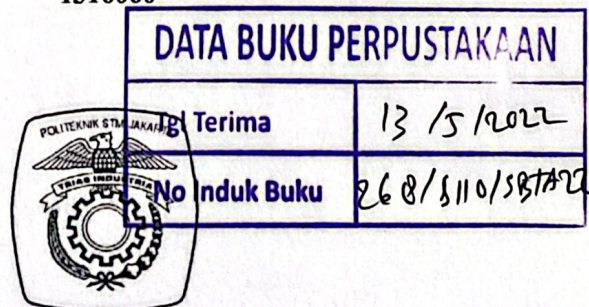
**TUGAS AKHIR**

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

**OLEH**

**YOSEP PETRHAL TETRA CLINTON**

1316060



**POLITEKNIK STMI JAKARTA  
KEMENTERIAN PERINDUSTRIAN RI  
JAKARTA  
2020**

**SUMBANGAN ALUMNI**

**POLITEKNIK STMI JAKARTA**  
**KEMENTERIAN PERINDUSTRIAN RI**

**TANDA PERSETUJUAN DOSEN PEMBIMBING**

JUDUL TUGAS AKHIR:

**RANCANG BANGUN SISTEM INFORMASI PENDISTRIBUSIAN UNIT  
PADA DEPARTEMEN *LOGISTIC 2W* DENGAN ALGORITMA *DIJKSTRA*  
MENGUNAKAN FUNGSI *EUCLIDEAN DISTANCE* DI PT SUZUKI  
INDOMOBIL MOTOR *PLANTI TAMBUN***

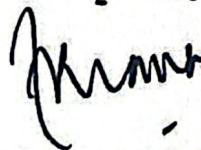
**Disusun Oleh:**

Nama : Yosep Petrhal Tetra Clinton  
NIM : 1316060  
Program Studi : Sistem Informasi Industri Otomotif  
Tanggal Seminar : 01 Oktober 2020  
Tanggal Sidang : 27 Oktober 2020  
Tanggal Lulus : 27 Oktober 2020

Telah diperiksa dan disetujui untuk diajukan dan dipertahankan dalam Ujian  
Tugas Akhir Politeknik STMI Jakarta

Jakarta, 18 September 2020

Dosen Pembimbing



Triana Fatmawati, S.T., M.T.  
NIP. 198005142005022001

**POLITEKNIK STMI JAKARTA**  
**KEMENTERIAN PERINDUSTRIAN RI**

**LEMBAR PENGESAHAN**

JUDUL TUGAS AKHIR:

**RANCANG BANGUN SISTEM INFORMASI PENDISTRIBUSIAN UNIT  
PADA DEPARTEMEN *LOGISTIC 2W* DENGAN ALGORITMA *DIJKSTRA*  
MENGUNAKAN FUNGSI *EUCLIDEAN DISTANCE* DI PT SUZUKI  
INDOMOBIL MOTOR *PLANT I* TAMBUN**

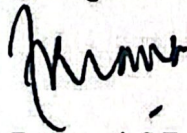
**Disusun Oleh:**

Nama : Yosep Petrhal Tetra Clinton  
NIM : 1316060  
Program Studi : Sistem Informasi Industri Otomotif

Telah diuji oleh Tim Penguji Sidang Tugas Akhir Program Studi Sistem Informasi Industri Otomotif Politeknik STMI Jakarta Kementerian Perindustrian R.I. pada hari Selasa, 27 Oktober 2020.

Jakarta, 27 Oktober 2020

Dosen Pembimbing



Triana Fatmawati, S.T, M.T.  
NIP. 198005142005022001

Ketua Penguji



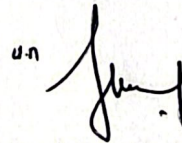
Dr. Ridzky Kramanandita, S.Kom., M.T.  
NIP. 197403022002121001

Dosen Penguji



Ahlan Ismono, S.Kom., M.M.S.I.  
NIP. 197901072006041002

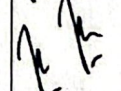
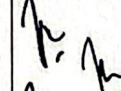
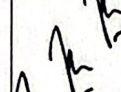
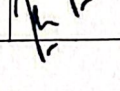
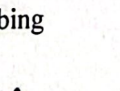
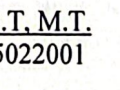
Dosen Penguji



Drs. Jacob Saragih, M.M.  
NIP. 195404281986031002

**LEMBAR BIMBINGAN PENYUSUNAN TUGAS AKHIR**

Nama : Yosep Petthal Tetra Clinton  
 NIM : 1316060  
 Judul Tugas Akhir : Rancang Bangun Sistem Informasi Pendistribusian  
 Unit Pada Departemen *Logistic 2W* Dengan  
*Algoritma Dijkstra* Menggunakan Fungsi *Euclidean*  
*Distance* di PT Suzuki Indomobil Motor *Plant I*  
 Tambun  
 Pembimbing : Triana Fatmawati, S.T., M.T.

Tanggal	Bab	Keterangan	Paraf
21-01-2020	I	Bimbingan Bab I	
29-01-2020	I	Revisi Bab I	
06-02-2020	II	Bimbingan Bab II	
12-02-2020	II	Revisi Bab II	
19-02-2020	III	Bimbingan Bab III	
28-02-2020	III & IV	Revisi Bab III & Bimbingan Bab IV	
10-03-2020	IV	Revisi Bab IV	
15-06-2020	V	Bimbingan Bab V	
20-06-2020	V	Revisi Bab V	
09-09-2020		Demo Program I	
12-09-2020		Demo Program II	
14-09-2020	I-VI	Cek Kelengkapan	

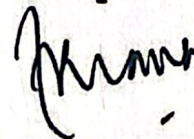
Mengetahui,

Ketua Program Studi Sistem Informasi  
 Industri Otomotif



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

Dosen Pembimbing



Triana Fatmawati, S.T., M.T.  
 NIP. 198005142005022001

## LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : Yosep Petrhal Tetra Clinton

NIM : 1316060

Berstatus mahasiswa Program Studi Sistem Informasi Industri Otomotif di Politeknik STMI Jakarta Kementerian Perindustrian Republik Indonesia. Dengan ini menyatakan bahwa hasil karya Tugas Akhir yang saya buat dengan judul: **“Rancang Bangun Sistem Informasi Pendistribusian Unit Pada Departemen Logistic 2W Dengan Algoritma Dijkstra Menggunakan Fungsi Euclidean Distance di PT Suzuki Indomobil Motor Plant I Tambun”**

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

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

Jakarta, September 2020

Yosep Petrhal Tetra Clinton

## ABSTRAK

PT Suzuki Indomobil Motor *Plant I Tambun* adalah salah satu perusahaan manufaktur yang bergerak di bidang industri otomotif dengan memproduksi kendaraan roda dua. Departemen *Logistic 2W* merupakan bagian penting dari perusahaan dalam proses pendistribusian unit. Permasalahan yang terjadi pada Departemen *Logistic 2W* yaitu adanya keterlambatan dalam proses pendistribusian produk hingga sampai pada distributor maupun proses kembalinya truk pengangkut ke dalam pabrik, sehingga menyebabkan pengiriman selanjutnya yang tertunda. Permasalahan keterlambatan tersebut terjadi karena belum menggunakan suatu metode pendistribusian tertentu sehingga belum adanya penentuan jalur terpendek atau rute tercepat mana yang harus dilalui. Selain itu, laporan gudang yang memuat ketersediaan unit diolah dengan proses manual dan disimpan dalam *Microsoft Excel*, yang membuat pekerjaan membutuhkan waktu yang lebih lama. Kegiatan tukar informasi dalam satu departemen seperti: memberikan informasi penerbitan surat jalan dan ketersediaan unit masih menggunakan *note/kertas* maupun disampaikan dengan media telepon, membuat waktu pengerjaan semakin lama karena perlu diantarkan langsung yang dimana lokasi kerja antar karyawan berbeda. Untuk mengatasi masalah tersebut diperlukan pengembangan suatu sistem terkomputerisasi yang saling terintegrasi dalam satu departemen dan mampu menyampaikan informasi yang dapat diterima secara langsung serta menggunakan suatu metode pendistribusian unit yaitu dengan Algoritma *Dijkstra* dilengkapi dengan fungsi *Euclidean Distance*. Sistem informasi ini dibangun dengan berbasis *website* dengan PHP sebagai bahasa pemrograman dengan *framework CodeIgniter-3.1.11* dan database *MariaDB-10.4.11*. Dengan adanya sistem informasi pendistribusian unit yang dibangun diharapkan dapat membantu perusahaan dalam mengelola manajemen data yang ada dan dapat mengintegrasikan data yang dibutuhkan ke bagian terkait. Hasil yang diperoleh dapat menunjukkan bahwa Algoritma *Dijkstra* membantu dalam menentukan rute terpendek untuk mencapai distributor. Disarankan untuk pengembangan sistem pendistribusian unit yang akan datang dapat ditambahkan fitur otomatis dalam mencari titik rute awal dan titik rute tujuan.

**Kata kunci:** Sistem Informasi, Pendistribusian Unit, Rute Terpendek, Algoritma *Dijkstra*, *Euclidean Distance*.

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Tuhan Yang Maha Esa atas limpahan kasih, rahmat serta karunia yang senantiasa diberikan sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul **“Rancang Bangun Sistem Informasi Pendistribusian Unit Pada Departemen *Logistic 2W* Dengan Algoritma *Dijkstra* Menggunakan Fungsi *Euclidean Distance* di PT Suzuki Indomobil Motor *Plant I Tambun*”**.

Penyusunan Tugas Akhir ini bertujuan untuk memenuhi sebagian syarat penyelesaian Program Sarjana Terapan pada Program Studi Sistem Informasi Industri Otomotif Politeknik STMI Jakarta. Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan baik dilihat dari segi isi maupun penyajiannya, walaupun penulis telah berupaya melakukannya sebaik mungkin. Untuk itu, Penulis mengharapkan kritik dan saran yang bersifat membangun kepada semua pihak guna perbaikan dimasa yang akan datang.

Untuk itu penulis mengucapkan banyak terima kasih kepada pihak-pihak yang telah membantu dan mendukung selama penyusunan Tugas Akhir ini. Penulis mengucapkan terima kasih kepada:

1. Tuhan Yang Maha Esa atas segala berkat, kasih serta kemudahan yang diberikan
2. Orang Tua dan kakak terutama Vina Margareth dan Maria Magdalena serta Bapak Darren Alkaviano dan Bang Jemris yang selalu mendoakan dan memberikan dukungan semangat, materi, kasih sayang karena berkat mereka saya dapat menjalankan pendidikan di Politeknik STMI Jakarta.
3. Bapak Dr. Mustofa, ST, MT, selaku Direktur Politeknik STMI Jakarta.
4. Bapak Lucky Heriyanto, S.T., M.T.I selaku Ketua Program Studi Sistem Informasi Industri Otomotif.

5. Ibu Triana Fatmawati, S.T., M.T. selaku dosen pembimbing yang telah memberikan arahan, saran, serta meluangkan waktu untuk berdiskusi dalam penyusunan tugas akhir. Beliau juga membimbing dengan sangat luar biasa dapat mendorong semangat saya untuk kembali melawan rasa sakit dan minder terhadap kondisi saya hingga mampu menyelesaikan tugas akhir ini.
6. Seluruh dosen di Politeknik STMI Jakarta yang telah memberikan ilmunya guna menambah pengetahuan dan pengalaman yang sangat bermanfaat.
7. Bapak Agustinus Soebajikto selaku *Assistant Manager* pada Departemen *Logistic 2W* Suzuki Indomobil Motor *Plant I* Tambun.
8. Bapak Imam Santoso, Ibu Puji Lestari, Bapak Arinowo Mukti, Bapak Ibnu, Bapak Heri dan seluruh pegawai di PT Suzuki Indomobil Motor *Plant I* Tambun yang telah membantu memberikan arahan serta informasi dalam pengumpulan data.
9. Felicia Dwi Ananda yang kerap saya panggil peli atau peloy adalah teman terbaik saya yang selalu ada membantu, memberikan waktu dan menghibur untuk tetap semangat menyelesaikan tugas akhir ini.
10. Imelda Setiawati sebagai teman dari awal kuliah hingga dapat melewati banyak hal bersama-sama, memberikan dukungan, dan menghibur saya untuk tetap semangat dalam menyelesaikan tanggung jawab.
11. Kiray Nabila sebagai teman bercanda saya yang selalu memberikan kesan positif dan semangat untuk menuntaskan apa yang wajib diselesaikan.
12. Kamila Nur Zafirah sebagai teman kuliah akrab yang selalu memberikan dorongan, menghibur bahkan mau memberikan ilmu dan pengalamannya untuk berbagi.
13. Tiara Finindhia teman seperbimbingan saya yang selalu sabar dan mau membantu serta meberikan saran-saran yang sangat berguna dalam penyusunan tugas akhir ini.
14. Sri Widianti teman yang selalu gerak cepat dalam merespon dan selalu memberi semangat untuk menyelesaikan tugas akhir.

15. Teman-teman *backsquad* Arsyi, Alvin, Tambun, Dia, Oji dan Mica yang selalu meluangkan waktu berkumpul untuk berbagi pengalaman baik dan memberikan semangat untuk menyelesaikan tugas akhir ini.
16. Teman-teman SA02 2016 yang mengawali masa kuliah bersama dan selalu membawa cerita baru serta memberikan pengalaman luar biasa yang tak bisa dilupakan.
17. Seluruh teman-teman mahasiswa/i Politeknik STMI Jakarta Program Studi Sistem Informasi Industri Otomotif angkatan 2016.
18. Semua orang-orang baik yang hadir disekitar saya yang selalu memberikan pelajaran, semangat bahkan nilai-nilai positif lainnya.

Demikianlah yang dapat Penulis sampaikan, semoga Tuhan Yang Maha Esa membalas kebaikan dan ketulusan semua pihak yang telah membantu dalam menyelesaikan Tugas Akhir ini dengan melimpahkan berkat dan kasih-Nya. Penulis berharap agar Tugas Akhir ini dapat memberikan manfaat serta pengetahuan bagi pembacanya. Terima kasih.

Jakarta, September 2020

Penulis

# DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
LEMBAR PERSETUJUAN DOSEN PEMBIMBING.....	ii
LEMBAR PENGESAHAN.....	iii
LEMBAR BIMBINGAN PENYUSUNAN TUGAS AKHIR.....	iv
LEMBAR PERNYATAAN KEASLIAN.....	v
ABSTRAK.....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xv
DAFTAR GAMBAR .....	xviii
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1. Latar Belakang Masalah .....	1
1.2. Pokok Permasalahan.....	3
1.3. Tujuan Praktik Kerja Lapangan.....	3
1.3. Batasan Masalah .....	4
1.4. Manfaat Tugas Akhir.....	4
1.5. Sistematika Penulisan .....	5
<b>BAB II LANDASAN TEORI.....</b>	<b>7</b>
2.1. Kajian Penelitian .....	7
2.2. Pengertian sistem.....	9
2.2.1. Karakteristik Sistem .....	9
2.2.2. Klasifikasi Sistem.....	11
2.2.3. Daur Hidup Sistem .....	12
2.3. Pengertian informasi .....	14
2.3.1. Klasifikasi Informasi .....	15
2.3. Klasifikasi Informasi .....	15
2.4. Pengertian Sistem Informasi .....	17

2.5. Konsep Pendistribusian .....	17
2.5.1. Pengertian Distribusi .....	18
2.5.2. Manajemen Distribusi .....	18
2.5.3. Saluran Distribusi .....	19
2.5.3.1. Fungsi Saluran Distribusi .....	19
2.5.3.2 Jenis Saluran Distribusi .....	22
2.6. Gudang .....	23
2.6.1. Fungsi Gudang .....	24
2.6.2. Jenis Gudang .....	24
2.6.3. Aktifitas Gudang .....	25
2.7. Penyimpanan .....	27
2.8. <i>Euclidean Distance</i> .....	27
2.8.1. Lintang/Latitude .....	28
2.8.2. Bujur/Longitude .....	28
2.9. Algoritma <i>Dijkstra</i> .....	29
2.9.1. Graf.....	31
2.9.1.1 Graf Berarah dan Berbobot .....	32
2.10. <i>System Development Life Cycle</i> .....	32
2.11. Metode <i>Waterfall</i> .....	33
2.12. Analisis dan Perancangan Sistem Berbasis Objek .....	35
2.13. Fishbone Diagram .....	35
2.14. <i>Business Process Modelling Notation</i> .....	37
2.15. <i>Unified Modelling Language</i> .....	41
2.15.1. Diagram UML .....	42
2.15.2. <i>Use Case Diagram</i> .....	43
2.15.3. Diagram Aktivitas ( <i>Activity Diagram</i> ).....	45
2.15.4. <i>Sequence Diagram</i> .....	46
2.15.5. <i>Class Diagram</i> .....	47
2.15.6. <i>Deployment Diagram</i> .....	48
2.16. <i>Entity Relationship Diagram</i> .....	49
2.17. <i>Conceptual Data Model</i> .....	51

2.18. Kamus Data .....	53
2.19. <i>Windows Navigation Diagram</i> .....	54
2.20. <i>Framework</i> .....	54
2.21. PHP ( <i>Hypertext Preprocessor</i> ).....	55
2.22. <i>CodeIgniter</i> (CI) .....	56
2.23. MariaDB .....	57
2.24. Pengujian Perangkat Lunak .....	60
<b>BAB III METODOLOGI PENELITIAN</b> .....	61
3.1. Metodologi Penelitian .....	61
3.2. Jenis dan Sumber Data .....	61
3.3. Metode Pengumpulan Data .....	62
3.4. Metode Pengembangan Sistem .....	63
3.5. Fungsi <i>Euclidean Distance</i> .....	64
3.5.1. Perhitungan Fungsi <i>Euclidean Distance</i> .....	64
3.6. Algoritma <i>Dijkstra</i> .....	65
3.7. Kerangka Penelitian .....	66
<b>BAB IV PENGUMPULAN DAN PENGOLAHAN DATA</b> .....	70
4.1. Sejarah Umum Perusahaan.....	70
4.2. Logo Perusahaan .....	72
4.3. Profil Perusahaan.....	72
4.4. Visi dan Misi Perusahaan.....	73
4.5. Jam Kerja .....	74
4.6. Produk yang Dihasilkan .....	76
4.7. Jaringan Pemasaran dan Distribusi .....	78
4.8. Departemen <i>Logistic 2W</i> .....	79
4.8.1 Struktur Organisasi Departemen <i>Logistic 2W</i> .....	79
4.8.2 <i>Job Description</i> Departemen <i>Logistic 2W</i> .....	80
4.9. Proses Pendistribusian Unit.....	81
4.10. Analisis Dokumen Pendistribusian Unit .....	81
4.11. Analisis Sistem Pendistribusian Unit .....	93
4.12. BPNM Proses Pendistribusian Unit .....	94

4.13. Analisis Masalah .....	96
4.14. Perhitungan Fungsi <i>Euclidean Distance</i> .....	98
4.14.1. Perhitungan Data Sampel Distributor.....	98
4.15. Perhitungan Algoritma <i>Dijkstra</i> .....	101
<b>BAB V ANALISIS DAN PEMBAHASAN .....</b>	<b>107</b>
5.1. <i>System Request</i> .....	107
5.2. Analisis Kebutuhan Sistem .....	107
5.2.1. Kebutuhan Fungsional Sistem.....	108
5.2.2. Kebutuhan Sistem ( <i>Non Fuctional Requirment</i> ).....	108
5.3. Proses Bisnis Pendistribusian Unit Usulan .....	109
5.4. <i>Use Case Diagram</i> Sistem Informasi Pendistribusian Unit.....	111
5.4.1. <i>Use Case Description</i> .....	112
5.5. <i>Activity Diagram</i> Sistem Informasi Pendistribusian Unit.....	122
5.6. <i>Sequence Diagram</i> Sistem Informasi Pendistribusian Unit .....	132
5.7. <i>Class Diagram</i> Sistem Informasi Pendistribusian Unit .....	142
5.8. <i>Deployment Diagram</i> Sistem Informasi Pendistribusian Unit.....	143
5.9. <i>Entity Relationship Diagram</i> Sistem Informasi Pendistribusian Unit .....	143
5.10. <i>Conceptual Data Model (CDM)</i> .....	144
5.11. Kamus Data .....	145
5.12. <i>Windows Navigation Diagram</i> .....	150
5.13. Perancangan Sistem Antarmuka.....	152
5.14. Spesifikasi Kebutuhan <i>Hardware</i> dan <i>Software</i> .....	166
5.15. <i>Blackbox Testing</i> .....	168
5.16. Simulasi Pencarian Fungsi <i>Euclidean Distance</i> Dengan <i>Microsoft Excel</i> .....	183
<b>BAB VI PENUTUP .....</b>	<b>186</b>
6.1. Kesimpulan.....	186
6.2. Saran.....	186

DAFTAR PUSTAKA .....	184
DAFTAR LAMPIRAN.....	188

## DAFTAR TABEL

	Halaman
Tabel II.1 Simbol-simbol <i>Flow Objects</i> .....	38
Tabel II.2 Simbol-simbol <i>Connecting Objectcs</i> .....	39
Tabel II.3 Simbol-simbol <i>Data Object</i> . ....	41
Tabel II.4 Simbol-simbol <i>Use-Case Diagram</i> .....	44
Tabel II.5 Simbol-simbol <i>Activity Diagram</i> .....	45
Tabel II.6 Simbol- simbol <i>Sequence Diagram</i> .....	47
Tabel II.7 Simbol-simbol <i>Class Diagram</i> .....	48
Tabel II.8 Simbol- simbol <i>Deployment Diagram</i> .....	49
Tabel II.9 Simbol-simbol <i>Entity Relationdhip Diagram</i> .....	51
Tabel II.10 Simbol-simbol CDM .....	52
Tabel II.11 Simbol Aturan CDM .....	52
Tabel II.12 Jenis Tipe Data pada MariaDB .....	58
Tabel IV.1 Jam Kerja <i>Shift</i> 1 Karyawan Operasional .....	74
Tabel IV.2 Jam Kerja <i>Shift</i> 1 Karyawan <i>Office</i> 1 & 2 .....	75
Tabel IV.3 Jam Kerja <i>Shift</i> 3 Karyawan PT Suzuki Indomobil Motor .....	76
Tabel IV.4 Produk PT Suzuki Indomobil Motor <i>Plant</i> Tambun I.....	76
Tabel IV.5 Daftar Jaringan Pemasaran Domestik.....	78
Tabel IV.6 Daftar Jaringan Pemasaran Mancanegara.....	78
Tabel IV.7 Deskripsi <i>Field-field Form</i> D.O ( <i>Delivery Order</i> ) .....	83
Tabel IV.8 Deskripsi <i>Field-field</i> Surat Jalan.....	84
Tabel IV.9 Deskripsi <i>Field-field</i> Laporan Harian Stok <i>Warehouse</i> Tambun .....	87
Tabel IV.10 Deskripsi <i>Field-field</i> Laporan Harian Stok <i>Warehouse</i> Ex-Berikat .	91
Tabel IV.11 Data Sampel Distributor .....	98
Tabel IV.12 Nilai Jarak Distributor dengan Perusahaan.....	100
Tabel IV.13 Perhitungan <i>Dijkstra</i> .....	104
Tabel V.1 <i>System Request</i> .....	107

Tabel V.2 Analisis Kebutuhan Sistem Informasi Pendistribusian Unit.....	108
Tabel V.3 Deskripsi Aktor .....	113
Tabel V.4 <i>Use Case Description Login</i> .....	114
Tabel V.5 Mengelola Data <i>User</i> .....	114
Tabel V.6 <i>Use Case</i> Mengelola Data Distributor .....	115
Tabel V.7 <i>Use Case</i> Mengelola Data Produk .....	116
Tabel V.8 <i>Use Case</i> Mengisi Stok Ketersediaan .....	117
Tabel V.9 <i>Use Case</i> Melihat Notifikasi Ketersediaan .....	118
Tabel V.10 <i>Use Case</i> Melihat Data DO.....	119
Tabel V.11 <i>Use Case</i> Membuat Surat Jalan .....	119
Tabel V.12 <i>Use Case</i> Membuat Jalur Distribusi.....	120
Tabel V.13 <i>Use Case</i> Melihat Notifikasi SJ .....	121
Tabel V.14 <i>Use Case</i> Melihat Laporan Gudang .....	121
Tabel V.15 Kamus Data <i>Users</i> .....	146
Tabel V.16 Kamus Data Distributor .....	146
Tabel V.17 Kamus Data Areal.....	147
Tabel V.18 Kamus Data Produk Item .....	147
Tabel V.19 Kamus Data <i>Category</i> .....	148
Tabel V.20 Kamus Data Warna .....	148
Tabel V.21 Kamus Data Gudangl .....	148
Tabel V.22 Kamus Data DO .....	149
Tabel V.23 Kamus Data Surat Jalan .....	149
Tabel V.24 Kamus Data Rute .....	150
Tabel V.25 <i>Test Case Login</i> .....	168
Tabel V.26 <i>Test Case</i> Mengelola Data <i>Users</i> .....	169
Tabel V.27 <i>Test Case</i> Mengelola Data Produk.....	171
Tabel V.28 <i>Test Case</i> Mengelola Data Distributor .....	173
Tabel V.29 <i>Test Case</i> Mengisi Stok Ketersediaan.....	175
Tabel V.30 <i>Test Case</i> Melihat Notifikasi Ketersediaan.....	176
Tabel V.31 <i>Test Case</i> Melihat Data DO .....	177
Tabel V.32 <i>Test Case</i> Membuat Surat Jalan .....	177

Tabel V.33 <i>Test Case</i> Melihat Notifikasi SJ.....	179
Tabel V.34 <i>Test Case</i> Membuat Jalur Distribusi .....	180
Tabel V.35 <i>Test Case</i> Melihat Laporan Gudang.....	182

## DAFTAR GAMBAR

	Halaman
Gambar II.1 Karakteristik Sistem .....	11
Gambar II.2 Daur Hidup Sistem .....	14
Gambar II.3 Siklus Informasi.....	15
Gambar II.4 Alur Distribusi .....	19
Gambar II.5 Fungsi Distribusi.....	21
Gambar II.6 Garis <i>Latitude</i> dan <i>Longitude</i> .....	29
Gambar II.7 Graf.....	31
Gambar II.8 Graf Berarah dan Berbobot.....	32
Gambar II.9 Model <i>Waterfall</i> .....	35
Gambar II.10 Diagram Tulang Ikan Tipe Rangkuman Sebab .....	37
Gambar II.11 Diagram Tulang Ikan Tipe Klasifikasi Proses Produksi .....	37
Gambar II.12 <i>Swimlanes</i> .....	40
Gambar II.13 Simbol dari <i>Group</i> dan <i>Annotation</i> .....	41
Gambar II.14 Contoh <i>Windows Navigation Diagram</i> .....	54
Gambar III.1 Kerangka Penelitian .....	69
Gambar IV.1 Logo PT Suzuki Indomobil Motor.....	72
Gambar IV.2 Struktur Organisasi PT Suzuki Indomobil Motor .....	80
Gambar IV.3 Dokumen Register <i>Outstanding D.O</i> .....	82
Gambar IV.4 Dokumen Surat Jalan Unit.....	86
Gambar IV.5 Laporan Harian Stok <i>Warehouse Tambun</i> .....	87
Gambar IV.6 Laporan Harian Stok <i>Warehouse Ex-Berikat</i> .....	90
Gambar IV.7 Proses Bisnis Sistem Pendistribusian Unit yang Berjalan .....	95
Gambar IV.8 <i>Fishbone Diagram</i> Sistem Pendistribusian Unit pada Departemen <i>Logistic 2W</i> .....	96
Gambar IV.9 Graf Berarah Studi Kasus Perhitungan Jarak .....	102
Gambar V.1 Proses Bisnis Sistem Pendistribusian Unit Usulan .....	110
Gambar V.2 Proses <i>Manual Task</i> Pada BPNM .....	110

Gambar V.3 Proses <i>Recieve Task</i> Pada BPNM .....	111
Gambar V.4 Proses <i>Service Task</i> Pada BPNM .....	111
Gambar V.5 Proses <i>User Task</i> Pada BPNM .....	111
Gambar V.6 <i>Use Case Diagram</i> Sistem Pendistribusian Unit Usulan .....	112
Gambar V.7 <i>Activity Diagram Login</i> .....	123
Gambar V.8 <i>Activity Diagram</i> Mengelola Data <i>User</i> .....	124
Gambar V.9 <i>Activity Diagram</i> Mengelola Data Distributor .....	125
Gambar V.10 <i>Activity Diagram</i> Mengelola Data Produk .....	126
Gambar V.11 <i>Activity Diagram</i> Mengisi Stok Ketersediaan .....	127
Gambar V.12 <i>Activity Diagram</i> Melihat Notifikasi Ketersediaan .....	128
Gambar V.13 <i>Activity Diagram</i> Melihat Data DO.....	128
Gambar V.14 <i>Activity Diagram</i> Membuat Surat Jalan .....	129
Gambar V.15 <i>Activity Diagram</i> Membuat Jalur Distribusi .....	130
Gambar V.16 <i>Activity Diagram</i> Melihat Notifikasi SJ .....	131
Gambar V.17 <i>Activity Diagram</i> Melihat Laporan Gudang.....	131
Gambar V.18 <i>Sequence Diagram</i> Melakukan <i>Login</i> .....	132
Gambar V.19 <i>Sequence Diagram</i> Mengelola Data <i>User</i> .....	133
Gambar V.20 <i>Sequence Diagram</i> Mengelola Data Distributor .....	134
Gambar V.21 <i>Sequence Diagram</i> Mengelola Data Produk .....	135
Gambar V.22 <i>Sequence Diagram</i> Mengisi Stok Ketersediaan .....	136
Gambar V.23 <i>Sequence Diagram</i> Melihat Notifikasi Ketersediaan .....	137
Gambar V.24 <i>Sequence Diagram</i> Melihat Data DO.....	138
Gambar V.25 <i>Sequence Diagram</i> Membuat Surat Jalan .....	139
Gambar V.26 <i>Sequence Diagram</i> Membuat Jalur Distribusi .....	140
Gambar V.27 <i>Sequence Diagram</i> Melihat Notifikasi SJ .....	141
Gambar V.28 <i>Sequence Diagram</i> Melihat Laporan Gudang.....	141
Gambar V.29 <i>Class Diagram</i> Sistem Informasi Pendistribusian Unit .....	142
Gambar V.30 <i>Deployment Diagram</i> Sistem Informasi Pendistribusian Unit .....	143
Gambar V.31 ERD Sistem Informasi Pendistribusian Unit.....	144
Gambar V.32 CDM Sistem Informasi Pendistribusian Unit.....	145
Gambar V.33 <i>Windows Navigation Diagram</i> Usulan.....	152

Gambar V.34 <i>Form Login</i> .....	152
Gambar V.35 Halaman Utama Kepala Gudang.....	153
Gambar V.36 Halaman Utama Admin Gudang.....	153
Gambar V.37 Halaman Utama Admin Surat Jalan.....	154
Gambar V.38 Halaman Utama Staf <i>Delivery</i> .....	154
Gambar V.39 Halaman Menu Data User.....	155
Gambar V.40 Halaman <i>Create User</i> .....	155
Gambar V.41 Halaman <i>Update User</i> .....	156
Gambar V.42 Halaman Menu Data Distributor.....	156
Gambar V.43 Halaman <i>Add Distributor</i> .....	157
Gambar V.44 Halaman <i>Edit Distributor</i> .....	157
Gambar V.45 Halaman Menu Data Produk.....	158
Gambar V.46 Halaman <i>Add Items</i> .....	158
Gambar V.47 Halaman <i>Edit Items</i> .....	158
Gambar V.48 Halaman Menu Data Laporan.....	159
Gambar V.49 Halaman Menu Galeri.....	160
Gambar V.50 Halaman Menu Data Gudang.....	160
Gambar V.51 Halaman Input Data Gudang.....	161
Gambar V.52 Halaman Detail Data Gudang.....	161
Gambar V.53 Halaman Notifikasi Ketersediaan.....	162
Gambar V.54 Halaman Data <i>Delivery Order</i> .....	162
Gambar V.55 Halaman Detail <i>ViewDO</i> .....	163
Gambar V.56 Halaman Data Detail <i>Delivery Order</i> .....	163
Gambar V.57 Halaman Data Surat Jalan.....	164
Gambar V.58 Halaman Notifikasi SJ.....	164
Gambar V.59 Halaman Rute Terpendek SJ.....	165
Gambar V.60 Halaman <i>Create Rute</i> .....	165
Gambar V.61 Halaman <i>View Tampil Full Rute</i> .....	166
Gambar V.62 Halaman <i>View Tampil Rute Tercepat</i> .....	166

Gambar V.63 Perhitungan Fungsi *Euclidean Distance* Pada *Microsoft Excel* ... 184  
Gambar V.64 Perhitungan Fungsi *Euclidean Distance* Pada Program..... 185

# BAB I

## PENDAHULUAN

MILIK PERPUSTAKAAN STMI  
Membaca : Ibadah, Mengambil : Dosa

### 1.1 Latar Belakang Masalah

PT Suzuki Indomobil Motor merupakan salah satu perusahaan manufaktur yang bergerak di bidang industri otomotif yang melakukan kegiatan produksi serta mengaplikasikan mesin, peralatan dan tenaga kerja serta suatu medium proses untuk mengubah bahan mentah menjadi barang jadi yang memiliki nilai jual. Pada proses produksinya perusahaan memproduksi unit kendaraan, dimana *Plant I* memproduksi roda dua (motor), sedangkan *Plant II* memproduksi mobil.

PT Suzuki Indomobil Motor *Plant I* Tambun menganut sistem produksi MTO (*make to order*) memproduksi kendaraan berdasarkan pesanan dari konsumen. Volume produksi yang diterapkan pada perusahaan yaitu *batch*. Hal tersebut dibuktikan dengan menerapkan minimal lot yaitu kelipatan 60 unit motor dalam sekali produksi tiap jenis motor.

Unit kendaraan yang telah selesai diproduksi akan disimpan pada gudang penyimpanan barang jadi (*finished good*), dimana nantinya akan dikirimkan sesuai *Delivery Order* (DO) yang diterima oleh Departemen *Logistic 2W* atau yang sering dikenal CBU (*Completely Build Up*). *Delivery Order* yang diterima dibuat berdasarkan *Purchase Order* (PO) yang diterima oleh Bagian Marketing.

Pentingnya sistem distribusi sangat mempengaruhi nilai dan nama baik perusahaan. Seperti halnya bagi PT Suzuki Indomobil Motor *Plant I* Tambun yang menurut data diperoleh dari Perusahaan (2019), memiliki penjualan motor sebanyak 6.383.108 unit pada tahun 2018, sehingga menuntut perusahaan untuk mengutamakan proses distribusi yang baik selain kualitas. Proses pengiriman unit yang dilakukan pada perusahaan ialah setelah Departemen *Logistic 2W* menerbitkan Surat Jalan (SJ) unit.

Pengiriman unit dilakukan oleh pihak ketiga yaitu ekspedisi yang bekerja sama dengan perusahaan. Pada prosesnya pihak ekspedisi membawa dokumen surat jalan beserta unit. Pengantaran unit yang dilakukan mencakup seluruh

wilayah Indonesia. Unit yang diantarkan tidak langsung sampai pada konsumen akhir melainkan menggunakan distributor sebagai tangan distribusi.

Unit yang telah selesai diproduksi namun belum mendapat giliran pengantaran, akan masuk ke dalam gudang penyimpanan menunggu tanggal pengirimannya. PT Suzuki Indomobil Motor *Plant I Tambun* memiliki dua lokasi gudang yaitu: Gudang Tambun dan Gudang Ex-Berikat. Unit yang disimpan dalam gudang kemudian akan dicatat oleh admin CBU dan dibuatkan laporan.

Laporan ketersediaan unit pada gudang berfungsi sebagai informasi kapasitas gudang, tipe unit dan jumlah unit yang disimpan. Namun pembuatan laporan gudang tersebut masih belum efektif dan masih menggunakan proses konvensional serta tidak saling terintegrasi dengan data yang dimiliki. Selama ini perusahaan dalam melakukan pencatatan unitnya masih menggunakan *Microsoft Excel* sebagai media untuk informasi pada gudang.

Selain itu, penyampaian informasi antar bagian berupa *note* yang dibuat dengan menggunakan kertas. Kemudian dalam penyampaiannya ialah secara langsung diantarkan kepada karyawan lain. Hal ini membutuhkan waktu yang lama dalam penyebaran informasi dengan bagian terkait karena lokasi kerja antar karyawan yang berbeda.

Perusahaan saat ini juga belum menerapkan jalur pengiriman dengan rute terpendek atau masih dilakukan secara acak. Kelemahan dari pengiriman yang dilakukan secara acak diantaranya jadwal kembalinya truk ke gudang yang seringkali melewati dari jadwal yang sudah ditentukan, sehingga menghambat pengiriman selanjutnya dan terjadinya penumpukan di gudang.

Oleh karena itu perlunya diterapkan pengaturan rute terpendek agar pengantaran lebih sistematis dan dapat mengurangi biaya serta menghemat waktu. Salah satu metode yang dapat digunakan untuk menentukan rute terpendek ialah Algoritma *Dijkstra*. Contoh penerapan algoritma ini adalah lintasan terpendek yang menghubungkan antara dua kota berlainan tertentu (Kusuma, Jefri & Agung, 2019). Algoritma ini akan mencari jalur dengan *cost* yang paling minimum antara titik yang satu dengan titik yang lainnya. Selain itu, algoritma *dijkstra* juga bisa

digunakan untuk menghitung total biaya atau *cost* dari lintasan terpendek yang sudah terbentuk (Yusuf, Zahra & Apriyanti, 2017)

Berdasarkan uraian di atas maka dalam penelitian Tugas Akhir ini akan dilakukan analisis, perancangan dan pengembangan sistem informasi. Penelitian ini dilakukan agar dapat mendukung sistem pendistribusian unit serta laporan penyimpanan unit pada gudang menjadi terkomputerisasi, informatif dan saling terintegrasi. Oleh karena itu, penelitian pada Tugas Akhir ini diberi judul “Sistem Informasi Pendistribusian Unit pada Departemen *Logistic 2W* Dengan Algoritma *Dijkstra* Menggunakan Fungsi *Euclidean Distance* di PT Suzuki Indomobil Motor Plant I Tambun”.

## 1.2 Pokok Permasalahan

Berdasarkan permasalahan yang terjadi pada Departemen *Logistic 2W* dalam pendistribusian unit maka ditetapkan pokok permasalahan pada penelitian ini adalah sebagai berikut:

1. Bagaimana agar penyebaran informasi berupa *note* antar bagian terkait dapat diterima secara langsung?
2. Bagaimana mengimplementasi algoritma pembuatan jalur terpendek pada sistem informasi agar menghasilkan rute pengiriman yang efisien?
3. Bagaimana agar pembuatan laporan penyimpanan pada gudang menjadi lebih informatif dan tersedia setiap saat serta dapat mengintegrasikan antar bagian?

## 1.3 Tujuan Penelitian

Adapun tujuan penelitian Tugas Akhir ini adalah menganalisis, merancang dan membangun sistem informasi yang mampu:

1. Membuat perancangan sistem informasi yang terkomputerisasi, sehingga dapat menyampaikan *note* yang berisi informasi antar bagian terkait untuk dapat diterima secara langsung.
2. Membuat sistem informasi usulan pada rute pengiriman unit dengan mengimplementasikan Algoritma *Dijkstra* menggunakan fungsi *euclidean*

*distance* pada aplikasi pendistribusian unit agar rute yang akan dilalui lebih terarah dan efisien.

3. Menyediakan laporan penyimpanan unit pada gudang untuk informasi ketersediaan stok menjadi informatif dan tersedia setiap saat serta saling terintegrasi antar bagian.

#### 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 dua bulan bertempat PT Suzuki Indomobil Motor *Plant I Tambun* tepatnya di Jalan Raya Diponegoro KM. 38,2, Jatimulya, Kec. Tambun Sel., Bekasi, Jawa Barat 17510. Departemen yang diteliti adalah pada Departemen *Logistic 2W* atau sering dikenal dengan CBU (*Completely Build Up*).
2. Jangka waktu penelitian ini ialah selama dua bulan dari 5 Agustus 2019 s.d 5 Oktober 2019.
3. Ruang lingkup yang dianalisis ialah proses pendistribusian unit pada distributor dengan batasan wilayah pengiriman hanya domestik dan urutan langkah kerja yang terjadi pada departemen tersebut.
4. Pengembangan sistem menggunakan bahasa pemrograman PHP dengan *framework Codeigniter* dan MariaDB sebagai *database* yang digunakan.

#### 1.5 Manfaat Tugas Akhir

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

1. Bagi Penulis  
Sebagai alat untuk mempraktikkan teori-teori yang telah didapat selama perkuliahan, sehingga penulis dapat menambah pengetahuan secara praktis tentang masalah-masalah yang dihadapi di dunia kerja dan yang terjadi pada perusahaan, khususnya perusahaan industri otomotif.
2. Bagi perusahaan

Hasil penelitian ini diharapkan dapat diimplementasikan di perusahaan untuk membantu kinerja karyawan pada Departemen *Logistic 2W* dalam mengelola informasi dan melakukan pendistribuan unit pada distributor.

### 3. Bagi Pihak Lain

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

## 1.6 Sistematika Penulisan

Tugas Akhir ini disusun secara sistematis dari beberapa tahapan berdasarkan hasil pengamatan yang dilakukan, sehingga mendapatkan gambaran yang sesuai dengan isi laporan. Adapun tahapan-tahapan dalam penulisan laporan ini adalah sebagai berikut:

### **BAB I            PENDAHULUAN**

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

### **BAB II           LANDASAN TEORI**

Pada bab ini membahas tentang berbagai teori yang diperoleh dari buku-buku literatur dan penelitian yang sejenis ataupun berbagai macam referensi lainnya yang berkaitan dengan tema yang diambil. Teori-teori yang dipaparkan pada laporan ini adalah seputar Algoritma *Dijkstra* dan teori-teori lainnya yang berhubungan dengan pokok bahasan serta perancangan sistem informasi.

### **BAB III          METODOLOGI PENELITIAN**

Pada bab ini membahas tentang langkah-langkah yang akan dilakukan dalam perumusan dan pemecahan masalah, jenis dan

sumber data, metode pengumpulan data serta metodologi pengembangan sistem yang digunakan.

#### **BAB IV            PENGUMPULAN DAN PENGOLAHAN DATA**

Pada bab ini membahas mengenai data yang diperoleh berdasarkan hasil pengamatan selama di PT Suzuki Indomobil Motor *Plant I Tambun* yang meliputi sejarah perusahaan, profil perusahaan, struktur organisasi, proses bisnis yang saat ini berjalan pada Departemen *Logistic 2W* atau yang sering dikenal CBU dan dokumen yang terkait pada proses tersebut.

#### **BAB V             ANALISIS DAN PEMBAHASAN**

Pada bab ini berisi tentang analisis yang lebih rinci dari pengolahan data, yakni mulai dari analisis kebutuhan sistem, pemodelan sistem dengan *Unified Modelling Language (UML)*, pemodelan basis data dengan *Entity Relationship Diagram (ERD)*, *Conceptual Data Model (CDM)*, kamus data, perancangan antar muka, mendokumentasikan bagaimana navigasi dari komponen yang ada pada sebuah aplikasi dengan *Windows Navigation Diagram (WND)* dan pembuatan spesifikasi sistem yang diperlukan, serta pengujian pada sistem.

#### **BAB VI            PENUTUP**

Pada bab ini diuraikan kesimpulan-kesimpulan dari hasil penelitian dan saran-saran untuk pihak perusahaan dalam berbagai hal yang berhubungan dengan proses pendistribusian unit serta sebagai saran untuk penelitian selanjutnya.

## BAB II

## LANDASAN TEORI

## 2.1 Kajian Penelitian

Kajian penelitian merupakan hasil-hasil penelitian yang sudah dimuat dalam bentuk jurnal maupun karya tulis ilmiah lainnya. Terdapat beberapa jurnal yang telah dikaji satu persatu sebagai referensi yang digunakan sebagai acuan dalam pengerjaan tugas akhir ini antara lain:

1. **Aplikasi Perhitungan dan Visualisasi Jarak Terpendek Berdasarkan Data *Coordinate* Dengan Algoritma *Dijkstra* Dalam Kasus Pengantaran Barang di Kawasan Jabodetabek.**

Penelitian yang dilakukan oleh Elsa Kusuma, Jefri dan Halim Agung (2019), yaitu pembuatan suatu aplikasi yang dapat melakukan perhitungan dan visualisasi jarak terpendek berdasarkan data *coordinate* dengan Algoritma *Dijkstra* dalam kasus pengantaran barang di kawasan Jabodetabek. Tujuan dalam penelitian ini adalah menentukan rute terpendek dalam pengantaran barang dalam wilayah Jabodetabek, sehingga dapat mengoptimalkan kinerja dalam proses pengantaran yang seringkali terhambat karena faktor lapangan yang tidak dapat diprediksi.

Metodologi pengembangan sistem yang dilakukan pada penelitian ini dengan metodologi *iterative-enhancement*. *Iterative-enhancement* adalah model linier yang diterapkan secara berulang-ulang. Pada model ini, proses dipecah menjadi beberapa modul yang dikembangkan secara bertahap. Kebutuhan utama yang telah diketahui dapat diproses terlebih dahulu, kebutuhan termasuk perubahan lainnya dapat diproses setelahnya.

Kesimpulan dari hasil penelitian ini adalah implementasi sistem pengantaran barang yang menggunakan Algoritma *Dijkstra*. Penelitian terbukti membantu untuk menemukan rute terpendek yang optimal. Sistem juga menawarkan kemudahan dalam memvisualisasikan peta bagi setiap pengguna sehingga mudah untuk dimanfaatkan.

## 2. Pencarian SPBU Terdekat dan Penentuan Jarak Terpendek Menggunakan Algoritma *Dijkstra* (Studi Kasus di Kabupaten Jember).

Penelitian ini dilakukan oleh Windi Eka, Dwiretno dan Abdul Roqib (2015), yaitu pembuatan suatu aplikasi yang dapat melakukan pencarian SPBU wilayah Kabupaten Jember dengan jarak terpendek dari *user*. Tujuan dari penelitian ini adalah agar pengguna dalam melakukan pencarian SPBU lebih efektif dan efisien. Dimana sistem akan melakukan perhitungan dengan menjadikan lokasi *user* sebagai input yang akhirnya merekomendasikan posisi terdekat SPBU dari *user*.

Penelitian ini dimulai dari pengumpulan data dan pemetaan lokasi SPBU di Kabupaten Jember. Data lain yang dibutuhkan adalah data jalan, persimpangan, dan info jalur satu arah. Data tersebut akan digunakan untuk penggambaran peta lokasi SPBU. Pencarian data yang dibutuhkan sebagian besar adalah data *spatial*, sehingga dibutuhkan GPS, metode *tracking*, dan wawancara kepada pemilik/karyawan SPBU.

Kesimpulan dari hasil penelitian ini adalah implementasi algoritma *dijkstra* pada aplikasi pencarian SPBU terdekat dari *user*. Hasil pengambilan data pemetaan SPBU di Kabupaten Jember didapatkan 33 SPBU dimana semua SPBU menjual bahan bakar premium dan bio solar. Hasil perhitungan dalam pencarian, dipengaruhi oleh nilai kriteria, *cost*, dan *reverse cost*. Sedangkan nilai waktu tempuh didapat dari perhitungan jarak dibagi dengan kecepatan.

## 3. Penentuan Rute Terpendek Pendistribusian Naskah Ujian Nasional Menggunakan Algoritma *Dijkstra* (Dinas Pendidikan dan Pengajaran di Kota Binjai).

Penelitian ini dilakukan oleh Siswan Syahputra (2017) yaitu pembuatan *prototype* dalam penyelesaian kasus pendistribusian naskah ujian nasional menggunakan algoritma *dijkstra* di Kota Binjai. Tujuan dari penelitian ini adalah memahami bagaimana algoritma *dijkstra*

menghasilkan model rute terpendek dalam pendistribusian naskah ujian nasional dan membuat *prototype* perangkat lunak rute terpendek menuju sekolah-sekolah SMA/SMK di Kota Binjai.

Kerangka kerja penelitian merupakan langkah-langkah yang akan dilakukan dalam penyelesaian masalah yang akan dibahas. Adapaun kerangka kerja penelitian ini antara lain persiapan penelitian, studi pustaka, pengumpulan data, analisis sistem, perancangan sistem dan implementasi sistem. Dalam perancangan sistem, UML digunakan untuk memodelkan interaksi yang akan terjadi pada sistem.

Kesimpulan dari hasil penelitian ini adalah penerapan algoritma *dijkstra* pada aplikasi *prototype* dapat menyimpan dan menampilkan hasil pencarian rute terpendek dari lokasi awal hingga tujuan akhir. Masalah-masalah yang mungkin muncul bisa diminimalisasi dengan pencarian rute terpendek menggunakan algoritma *dijkstra*.

## 2.2 Pengertian Sistem

Sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu. Suatu sistem dibuat untuk menangani sesuatu yang berulang kali atau yang secara rutin terjadi. Keberhasilan komponen-komponen yang dipertimbangkan secara bersama sebagai suatu sistem jauh lebih besar daripada jumlah keberhasilan setiap komponen yang dipertimbangkan secara terpisah (Sutabri, 2012).

### 2.2.1 Karakteristik Sistem

Menurut Sutabri (2012), model umum sebuah sistem terdiri dari *input*, proses, dan *output*. Hal ini merupakan konsep sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat beberapa masukan dan keluaran sekaligus. Selain itu sebuah sistem juga memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun karakteristik yang dimaksud adalah sebagai berikut:

1. **Komponen Sistem (*Components*)**

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat-sifat sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.
2. **Batasan Sistem (*Boundary*)**

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.
3. **Lingkungan Luar Sistem (*Environment*)**

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem. Lingkungan luar sistem ini dapat menguntungkan dan dapat juga merugikan sistem tersebut. Lingkungan luar sistem yang menguntungkan merupakan energi bagi sistem tersebut, yang dengan demikian lingkungan luar tersebut harus selalu dijaga dan dipelihara.
4. **Penghubung Sistem (*Interface*)**

Media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lain.
5. **Masukan Sistem (*Input*)**

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

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain.

Contoh sistem informasi, keluaran yang dihasilkan adalah informasi yang dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang merupakan input bagi subsistem lainnya.

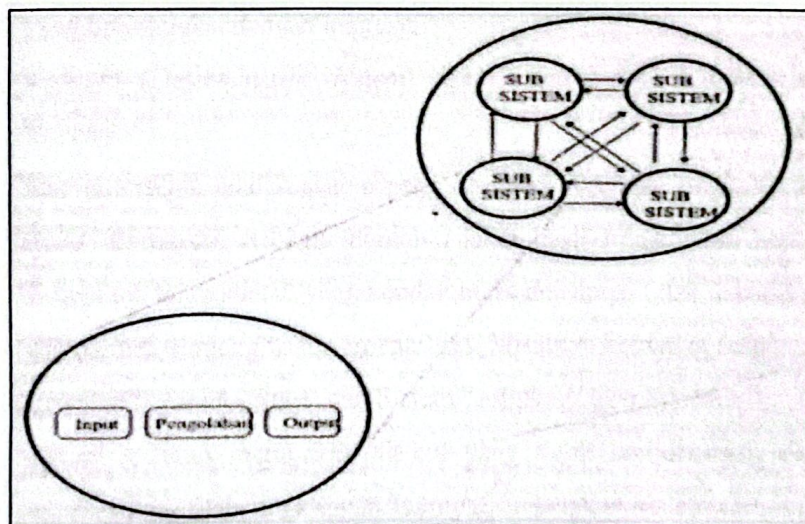
#### 7. Pengolah Sistem (Proses)

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

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

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

Berikut Gambar II.1 merupakan Karakteristik Sistem:



Gambar II.1 Karakteristik Sistem  
Sumber: (Sutabri, 2012)

### 2.2.2 Klasifikasi Sistem

Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi dalam sistem tersebut. Oleh karena itu, sistem dapat diklasifikasikan dari beberapa sudut pandang di antaranya (Sutabri, 2012):

1. Sistem Abstrak (*Abstract System*) dan Sistem Fisik (*Physical System*)  
Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem teologia yang berupa pemikiran hubungan antara manusia dengan Tuhan, sedangkan sistem fisik merupakan sistem yang ada secara fisik, misalnya sistem komputer, sistem produksi, sistem penjualan, dan lain sebagainya.

2. Alamiah (*Natural System*) dan Sistem Buatan Manusia (*Human Made System*)

Sistem alamiah adalah sistem yang terjadi melalui proses alam; tidak dibuat oleh manusia, misalnya sistem perputaran bumi, terjadinya siang malam, pergantian musim. Sedangkan sistem buatan manusia merupakan sistem yang melibatkan interaksi manusia dengan mesin yang disebut *human machine* sistem. Sistem informasi berbasis komputer merupakan contoh *human machine* sistem karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

3. Tertentu (*Deterministic System*) dan Sistem Tak Tentu (*Probabilistic System*)

Sistem tertentu adalah sistem yang beroperasi dengan tingkah laku yang dapat diprediksi. Sebagai contoh adalah hasil pertandingan sepak bola. Sistem tak tentu adalah yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas. Misalnya kematian seseorang.

4. Tertutup (*Closed System*) dan Sistem Terbuka (*Open System*)

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sedangkan sistem terbuka adalah sistem yang berhubungan oleh lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainnya.

### 2.2.3 Daur Hidup Sistem

Menurut Sutabri (2012), siklus hidup sistem (*system life cycle*) adalah proses evolusioner yang diikuti dalam penerapan sistem atau subsistem informasi berbasis komputer. Siklus hidup sistem terdiri dari serangkaian tugas yang

mengikuti langkah-langkah pendekatan sistem, karena tugas-tugas tersebut mengikuti pola yang teratur dan dilakukan secara *top down*. Ada beberapa tahapan daur hidup suatu sistem adalah sebagai berikut:

1. Mengenali adanya kebutuhan

Sebelum segala sesuatu terjadi, terlebih dahulu timbul suatu kebutuhan atau problema yang harus dapat dikenali. Kebutuhan dapat terjadi sebagai hasil perkembangan organisasi. Volume kebutuhan itu meningkat melebihi kapasitas dari sistem yang ada. Semua kebutuhan itu harus didefinisikan dengan jelas. Tanpa adanya kejelasan mengenai kebutuhan yang ada, pembangunan sistem akan kehilangan arah dan efektivitasnya.

2. Pembangunan sistem

Suatu proses atau seperangkat prosedur yang harus diikuti guna menganalisis kebutuhan yang timbul dan membangun suatu sistem untuk memenuhi kebutuhan tersebut.

3. Pemasangan sistem

Pemasangan sistem merupakan tahap yang penting dalam daur hidup sistem, dimana peralihan dari tahap pembangunan menuju tahap operasional adalah pemasangan sistem, yang merupakan langkah akhir dari suatu pembangunan sistem.

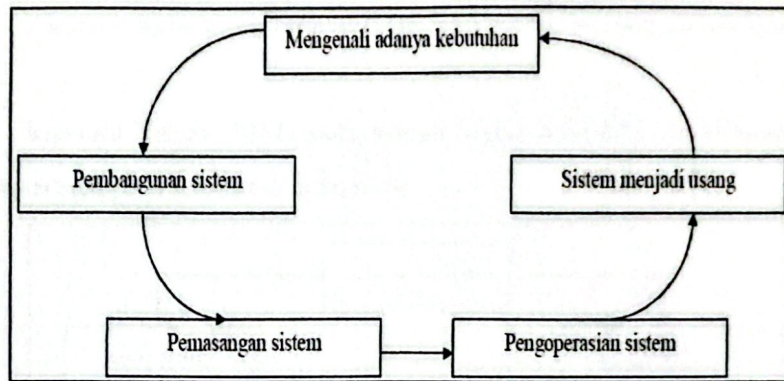
4. Pengoperasian sistem

Program-program komputer dan prosedur-prosedur pengoperasian yang membentuk suatu sistem informasi semuanya bersifat statis, sedangkan organisasi yang ditunjang oleh sistem informasi selalu mengalami perubahan karena pertumbuhan kegiatan, perubahan peraturan dan kebijaksanaan, ataupun kemajuan teknologi. Untuk mengatasi perubahan-perubahan tersebut, sistem harus diperbaharui atau diperbaiki.

5. Sistem menjadi usang

Kadang-kadang perubahan yang terjadi begitu drastis sehingga tidak dapat diatasi hanya dengan melakukan perbaikan sistem yang sedang berjalan. Pada saat itu, sistem tersebut sudah tidak layak lagi untuk dioperasikan dan sistem yang baru perlu dibangun untuk menggantikannya.

Sistem informasi kemudian akan melanjutkan daur hidupnya. Sistem dibangun untuk memenuhi kebutuhan. Sistem beradaptasi terhadap aneka perubahan lingkungan yang dinamis hingga kemudian sampai pada kondisi di mana sistem tidak dapat lagi beradaptasi. Sistem baru kemudian dibangun untuk menggantikannya. Daer sistem ini dapat dilihat pada Gambar II.2 di bawah ini:



Gambar II.2 Daur Hidup Sistem  
Sumber: Sutabri (2012).

### 2.3 Pengertian Informasi

Informasi adalah data yang sudah diolah menjadi sebuah bentuk yang berarti bagi pengguna, yang bermanfaat dalam pengambilan keputusan saat ini atau mendukung sumber informasi. Data belum memiliki nilai sedangkan informasi sudah memiliki nilai. Informasi dikatakan bernilai bila manfaatnya lebih besar dibanding biaya untuk mendapatnya (Kusrini, 2007).

Informasi merupakan kata benda dari *informare* yang berarti aktivitas dalam “pengetahuan yang dikomunikasikan”. Nilai informasi ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan dengan biaya mendapatkannya (Jogiyanto, 2013).

Menurut (Jogiyanto, 2013), kualitas dari informasi (*quality of information*) sangat tergantung dari tiga hal yaitu sebagai berikut :

1. Relevan (*relevance*)

Informasi tersebut mempunyai manfaat untuk pemakainya dan relevansi informasi untuk tiap-tiap orang akan berbeda-beda.

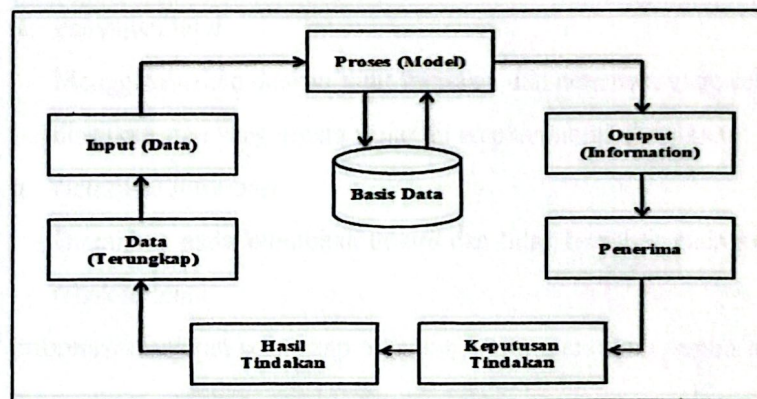
2. Tepat pada waktunya

Informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai guna lagi, karena informasi merupakan landasan didalam pengambilan keputusan.

### 3. Akurat (*accuracy*)

Informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya.

Menurut Yakub (2012) suatu bentuk Siklus Informasi (*information cycle*) seperti terlihat pada gambar di bawah ini:



Gambar II.3 Siklus Informasi  
Sumber: Yakub (2012)

Data diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, melakukan keputusan berdasarkan informasi tersebut, yang berarti menghasilkan suatu tindakan lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap kembali sebagai input, diproses kembali melalui suatu model dan seterusnya sebagai suatu siklus.

#### 2.3.1 Klasifikasi Informasi

Informasi merupakan hal yang sangat penting dalam pengambilan keputusan sehari-hari. Menurut Mulyanto (2009) informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Romey & Steinbart (2015) mengemukakan kriteria informasi yang baik sebagai berikut:

1. Informasi yang relevan adalah yang memiliki antara lain:

- a. Memiliki umpan balik  
Informasi memungkinkan pengguna untuk menegaskan atau mengoreksi ekspektasi mereka di masa lalu.
  - b. Memiliki manfaat prediktif  
Informasi dapat membantu pengguna untuk memprediksi masa yang akan datang berdasarkan hasil masa lalu dan kejadian masa kini.
2. Andal dimana informasi harus bebas dari pengertian yang menyesatkan dan kesalahan material, menyajikan setiap fakta secara jujur. Informasi yang akurat memenuhi karakteristik:
    - a. Penyajian jujur  
Menggambarkan dengan jujur transaksi dan peristiwa yang seharusnya disajikan atau yang secara wajar diharapkan untuk disajikan.
    - b. Netralitas Informasi  
Diarahkan pada kebutuhan umum dan tidak berpihak pada kebutuhan pihak tertentu.
  3. Informasi disajikan selengkap mungkin, yaitu mencakup semua informasi yang dapat mempengaruhi pengambilan keputusan. Informasi yang lengkap memenuhi karakteristik:
    - a. Disajikan dengan lengkap
    - b. Sesuai dengan ketentuan dan kebutuhan
  4. Tepat waktu dimana informasi disajikan tepat waktu sehingga dapat berpengaruh dan berguna dalam pengambilan keputusan. Informasi yang tepat waktu memenuhi karakteristik:
    - a. Tersedia pada saat dibutuhkan.
    - b. Informasi yang disajikan terbaru.
  5. Dapat dipahami dimana informasi yang disajikan dinyatakan dalam bentuk serta istilah yang disesuaikan dengan batas pemahaman para pengguna.
  6. Dapat diverifikasi dimana informasi yang disajikan dapat diuji, dan apabila pengujian dilakukan lebih dari sekali oleh pihak yang berbeda, hasilnya tetap menunjukkan simpulan yang tidak berbeda jauh.

## 2.4 Pengertian Sistem Informasi

Menurut O'Brian dikutip oleh Yakub (2012), sistem informasi (*information system*) merupakan kombinasi teratur dari orang-orang, perangkat keras, perangkat lunak, jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah, menyebarkan informasi dalam sebuah organisasi.

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

## 2.5 Konsep Pendistribusian

Dalam kegiatan manufaktur, distribusi memegang peranan penting dalam menjamin produk yang dipasarkan supaya produk tersebut tersedia dan sampai ke konsumen dengan baik. Ada beberapa faktor yang membatasi penyaluran secara langsung dari produsen ke konsumen, yakni sebagai berikut (Suryanto, 2016):

1. *Geographical Gap*, perbedaan jarak geografis yang disebabkan oleh perbedaan tempat produksi dengan lokasi konsumen yang tersebar luas.
2. *Time Gap*, perbedaan jarak waktu yang disebabkan waktu produksi dengan kebutuhan konsumen dalam jumlah besar dan menimbulkan nilai guna waktu (*time utility*).
3. *Quantity Gap*, perbedaan dalam jumlah produksi yang lebih besar akan lebih efisien biaya per-unitnya dibandingkan produksi kecil.
4. *Variety Gap*, pihak produsen memproduksi suatu variasi produk tertentu dalam jumlah besar tapi kenyataannya kebutuhan konsumen lebih kecil.
5. *Communication and Information Gap*, perbedaan informasi dan komunikasi saat produsen tidak mengetahui produk yang dibutuhkan dan siapa konsumen potensialnya.

Menurut Walter dikutip oleh Suryanto (2016) dalam mengatasi keterbatasan tersebut, maka produsen wajib menyusun saluran distribusinya.

Produsen membutuhkan suatu mitra untuk membantu memindahkan suatu produk yang dihasilkan kepada konsumen sebagai pemakainya. Dengan adanya mitra, konsumen akan dengan mudah memperoleh produk yang dibutuhkan.

### 2.5.1 Pengertian Distribusi

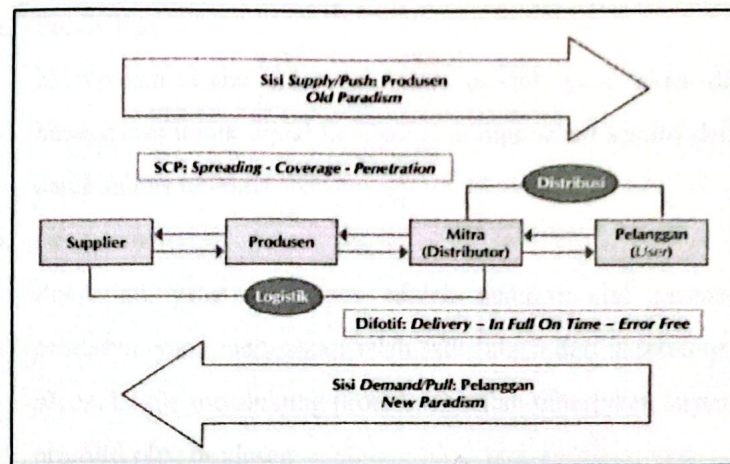
Distribusi adalah salah satu aspek dari pemasaran yang berusaha memperlancar dan mempermudah penyampaian barang dan jasa dari produsen kepada konsumen, sehingga penggunaannya sesuai dengan yang diperlukan (jenis, jumlah, harga, tempat, dan saat dibutuhkan).

Menurut Winardi dikutip oleh Arif (2018), distribusi merupakan sekumpulan perantara yang terhubung erat antara satu dengan yang lainnya dalam kegiatan penyaluran produk-produk kepada konsumen. Sedangkan menurut Kotler dikutip oleh Arif (2018), distribusi merupakan sekumpulan organisasi yang membuat sebuah proses kegiatan penyaluran suatu barang atau jasa untuk dipakai atau dikonsumsi oleh para konsumen. Oleh karena itu pentingnya kegiatan distribusi pada suatu perusahaan atau organisasi. Adapun fungsi distribusi adalah:

1. Menyalurkan barang-barang dari produsen ke konsumen.
2. Membantu memperlancar pemasaran, sehingga barang-barang yang dihasilkan produsen dapat segera terjual kepada konsumen.

### 2.5.2 Manajemen Distribusi

Manajemen distribusi adalah suatu strategi dalam mengembangkan saluran distribusi dari perencanaan (*planning*), mengorganisasi (*organization*), mengoperasikan (*operation*), dan pengawasan (*controlling*) guna mencapai tujuan perusahaan. Dalam hal distribusi, ada dua sisi yang berperan, yakni produsen dan konsumen. Produsen sebagai sisi prinsipal berperan supaya produk bisa tersebar merata. Sementara pada sisi konsumen adalah bagaimana mereka sebagai pemakai produk dapat memperoleh produk itu dengan mudah. Di bawah ini Gambar II.4 adalah Alur Distribusi dengan manajemen distribusi:



Gambar II.4 Alur Distribusi  
Sumber: Suryanto (2016).

### 2.5.3 Saluran Distribusi

Waktu memegang peranan yang penting. Kegunaan barang akan maksimal jika barang yang dibutuhkan itu dapat diperoleh pada saat diperlukan. Sebaliknya distribusi yang tidak tepat waktunya akan menimbulkan kerugian bagi produsen atau konsumen, yaitu produsen kehilangan keuntungan dan konsumen kepuasannya berkurang. Menurut Kotler dikutip oleh Arif (2018), mengemukakan bahwa saluran distribusi adalah serangkaian organisasi yang saling tergantung dan terlibat dalam proses untuk menjadikan suatu barang atau jasa siap untuk digunakan atau dikonsumsi. Saluran distribusi pada dasarnya merupakan perantara yang menjembatani antara produsen dan konsumen.

#### 2.5.3.1 Fungsi Saluran Distribusi

Dalam pelaksanaan operasional pendistribusian, produsen memiliki mitra dalam hal pendistribusian yang dikenal sebagai distributor. Ada tiga fungsi yang dilakukan oleh mitra/distributor (Suryanto, 2016):

##### 1. Fungsi Pertukaran (*Transaction Function*)

Fungsi pertukaran ini memerlukan adanya kegiatan transaksi antara dua pihak atau lebih, baik kegiatan transaksi yang dilakukan pihak mitra/distributor dengan pihak pelanggan maupun produsen. Fungsi dalam pertukaran tersebut adalah sebagai berikut:

a. Pembelian

Merupakan usaha dalam memilih produk yang akan dibeli, baik barang/jasa untuk dijual kembali atau digunakan sendiri dengan harga dan kualitas tertentu.

b. Penjualan

Penjualan yang dilakukan adalah sebagai alat pemasaran bagi produsen, yang merupakan salah satu fungsi dari *marketing mix*, yaitu *place*. Untuk mendukung proses penjualan diperlukan target pasar dan promosi oleh produsen

c. Pengambilan Resiko

Resiko yang berkaitan dengan saluran distribusi produk dari perusahaan sampai pada konsumen akhir meliputi pemrosesan pesanan, tempat penyimpanan, persediaan, dan pengiriman pesanan kepada pelanggan, serta kredit tidak terbayar.

2. Fungsi Penyedia Fisik (*Logistical Function*)

Fungsi penyedia fisik terjadi setelah adanya proses distribusi, yakni perpindahan produk atau jasa dari produsen melalui mitra/distributor ke konsumen akhir. Ada empat macam yang dilakukan dalam penyedia fisik produk/jasa antara lain:

a. Pengumpulan

Sebagai saluran distribusi, pihak mitra/distributor melakukan fungsi perantara dalam mengumpulkan produk atau jasa dari beberapa sumber penyedia (produsen).

b. Penyimpanan

Fungsi penyimpanan menciptakan nilai guna waktu karena adanya penyesuaian penawaran dengan permintaan.

c. Pemilihan

Para mitra/distributor akan melakukan fungsi pemilihan produk atau jasa yang akan dipasarkan yang meliputi penggolongan, pemeriksaan, menentukan jenis barang yang dikonsumsi secara retail (eceran).

d. Pengangkutan

Penggunaan sarana pengangkutan mempengaruhi harga produk atau jasa antara lain jenis pengangkutan, jarak perjalanan, lamanya perjalanan dan kondisi barang.

3. Fungsi Penunjang (*Supporting Function*)

Fungsi penunjang memiliki peranan membantu pelaksanaan fungsi lainnya, diantaranya adalah sebagai berikut:

a. Pelayanan Purnajual

Jaminan purnajual yang dilakukan ini bertujuan meningkatkan loyalitas konsumen dalam menggunakan produk atau jasa.

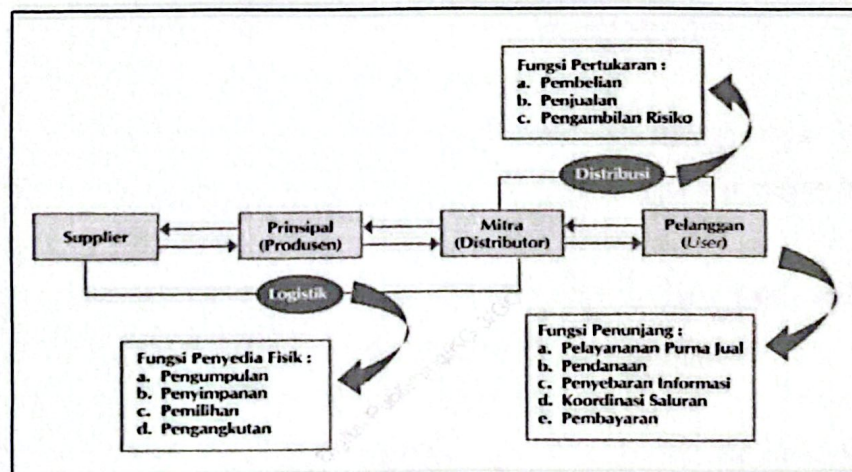
b. Pendanaan

Distribusi memerlukan dana yang dikeluarkan akibat permintaan dan penyebaran untuk menutup biaya dari saluran distribusi.

c. Penyebaran Informasi

Penyebaran informasi tersebut bisa meliputi perubahan harga, promosi, perubahan kemasan, *lauching* produk baru dan kondisi puitang pelanggan.

Untuk menjelaskan lebih lanjut mengenai posisi masing-masing antara mitra, *supplier*, produsen dan konsumen, di bawah ini ialah Gambar II.5 Fungsi Distribusi dengan arus dari masing-masing peran:



Gambar II.5 Fungsi Distribusi

Sumber: Suryanto (2016).

### 2.5.3.2 Jenis Saluran Distribusi

Menurut Vernon dan Jackson dikutip oleh Arif (2018) jenis saluran distribusi berdasarkan intensitasnya dibagi menjadi tiga, yaitu:

1. Bentuk intensif, yaitu jenis saluran yang memanfaatkan banyak pedagang besar dan kecil.
2. Bentuk selektif, yaitu jenis distribusi yang hanya memanfaatkan beberapa grosir dan sejumlah kecil pengecer.
3. Bentuk eksklusif, yaitu saluran distribusi yang hanya melibatkan satu perantara dalam lingkungan masyarakat tertentu untuk menangani produk. Saluran ini dibagi menjadi dua, yaitu saluran langsung dan saluran tidak langsung. Saluran langsung ialah dari produsen langsung kepada konsumen. Biasanya hanya sedikit barang yang dipasarkan secara langsung. Saluran tidak langsung, yang dibagi menjadi dua yaitu dari produsen ke pengecer ke konsumen dan dari produsen ke grosir ke pengecer ke konsumen.

Terdapat perbedaan dari beberapa jenis saluran distribusi yang ada di masyarakat saat ini. Berikut adalah pembagiannya masing-masing (Arif, 2018):

1. Tipe Saluran Distribusi Untuk Barang Konsumsi:
  - a. Produsen – Konsumen  
Produsen dapat menjual barang yang dihasilkannya langsung kepada konsumen akhir. Saluran ini disebut distribusi langsung.
  - b. Produsen – Pengecer – Konsumen  
Produsen hanya melayani penjualan dalam jumlah besar kepada pengecer, tidak menjual dalam jumlah sedikit kepada konsumen biasa.
  - c. Produsen – Pedagang Besar – Pengecer – Konsumen  
Banyak digunakan oleh produsen dan disini produsen hanya melayani penjualan dalam jumlah besar kepada pedagang besar saja
  - d. Produsen – Agen – Pengecer – Konsumen  
Disini produsen memilih agen sebagai penyalurnya. Agen menjalankan kegiatan perdagangan besar dalam saluran distribusi yang ada.

- e. Produsen – Agen – Pedagang Besar – Pengecer – Konsumen  
Produsen sering menggunakan agen sebagai perantara untuk menyalurkan barangnya kepada pedagang besar
2. Tipe Saluran Distribusi Untuk Barang Industri:
- a. Produsen – Pemakai Industri  
Saluran distribusi barang industri ini ialah saluran yang paling pendek dan disebut saluran distribusi langsung.
  - b. Produsen – Distributor Industri – Pemakai Industri  
Produsen menggunakan distributor industri untuk mencapai pasarnya. Tujuan lainnya agar barang-barang dengan pasti dan mudah diperoleh oleh pemakai. Biasanya distributor industri dapat melakukan penjualan lebih sempurna dari produsen karena mereka lebih dekat hubungannya dengan pemakai dalam suatu daerah tertentu.
  - c. Produsen – Agen Distributor – Pemakai Industri  
Dipakai oleh produsen yang tidak memiliki departemen pemasaran. Biasanya juga dipakai oleh perusahaan yang ingin memasuki daerah pemasaran baru dan lebih suka menggunakan agen.

## 2.6 Gudang

Menurut Warman (2010) gudang adalah bangunan yang digunakan untuk menyimpan barang. Barang-barang yang disimpan di dalam gudang dapat berupa bahan baku, barang setengah jadi, suku cadang, atau barang dalam proses yang disiapkan untuk diserap oleh proses produksi. Sedangkan menurut (Yunarto & Santika, 2005) Gudang adalah suatu tempat yang digunakan untuk menyimpan barang baik yang berupa *raw material*, barang *work in process* atau *finished good*.

Menurut Yunarto & Santika (2005) kegiatan tersebut dapat meliputi kegiatan *movement* (perpindahan), *storage* (penyimpanan) dan *information transfer* (transfer informasi). Dari pandangan di atas, dapat ditarik kesimpulan bahwa gudang adalah bangunan atau tempat yang digunakan untuk menyimpan barang baik yang berupa *raw material*, barang *work in process* atau *finished good*.

### 2.6.1 Fungsi Gudang

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

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

### 2.6.2 Jenis Gudang

Menurut Pandiangan (2017) gudang harus menjadi titik *transshipment* semua barang yang diterima maupun yang dikirim dengan cepat, efektif, dan se-efisien mungkin. Gudang terus memainkan peran utama dalam rantai pasok dan akan terus melakukannya di masa mendatang, meskipun gudang akan muncul dalam bentuk yang berbeda. Dari perkembangannya, maka gudang dapat dioperasikan oleh pemiliknya atau disubkontrakkan kepada penyedia *logistic* (pihak ketiga). Berikut merupakan jenis gudang antara lain:

1. Gudang Bahan Baku (*Raw Material Warehouse*)

Gudang bahan baku atau gudang bahan mentah adalah tempat penyimpanan bahan baku sebelum dipergunakan untuk proses produksi oleh perusahaan yang bersangkutan.

2. Gudang Barang Setengah Jadi (*Semi Finished Goods Warehouse*)

Terdapat produksi yang sudah diproses tetapi belum selesai atau memerlukan proses lanjutan (*work in process*) disebut barang setengah jadi. Barang setengah jadi ini membutuhkan waktu tunggu dalam antrian

proses produksi, sehingga diperlukan tempat penyimpanan di gudang tersendiri disebut persediaan *on line* (*inventory on line*).

3. Gudang Barang Jadi (*Finished Goods Warehouse*)

Gudang untuk barang jadi merupakan gudang yang disiapkan oleh perusahaan untuk menyimpan barang jadi atau produk akhir proses produksi atau dapat juga berupa barang atau produk yang siap didistribusikan atau dijual.

4. Gudang Terminal Pusat (*Central Warehouse*)

Gudang yang digunakan untuk mengumpulkan beberapa jenis barang dari masing-masing sumber atau pemasok. Selanjutnya menggabungkannya untuk dikirimkan ke tempat tujuan tertentu atau pelanggan.

5. Pusat Distribusi Gudang

Digunakan untuk mengumpulkan beberapa jenis barang dari sumber tunggal (hasil satu perusahaan manufaktur) untuk dikirimkan ke tempat tujuan (pelanggan). Perusahaan induk menyewa atau membuat satu anak perusahaannya berbentuk pengolahan pergudangan yang berfungsi untuk mendistribusikan seluruh hasil produksinya kepada pelanggan.

6. *Break-bulk Operation* Gudang

Digunakan untuk menerima barang atau produk dalam jumlah atau volume besar, kemudian dipecah-pecah atau dibagi-bagi dalam jumlah atau volume yang lebih kecil dan selanjutnya dikirimkan ke beberapa tempat tujuan

7. *Cross-Docking* Gudang

*Cross docking* disebut juga gudang *in-transit mixing*. Digunakan untuk menerima atau mengumpulkan beberapa jenis barang dari beberapa pemasok dan kemudian dibagi-bagi, digabungkan atau dikombinasikan sesuai dengan jumlah, ragam barang dari permintaan pelanggan.

### 2.6.3 Aktivitas Gudang

Menurut Pandiangan (2017) aktivitas secara teknis di dalam gudang dilaksanakan mengikuti prosedur yang baku dan dibuat sesederhana mungkin.

Menurut Purnomo (2004) terdapat tiga fungsi utama dalam aktivitas pergudangan, yaitu:

1. Perpindahan (*Movement*)

Salah satu kegiatannya adalah memperbaiki perputaran persediaan dan mempercepat proses pesanan dari produksi hingga ke pengiriman utama.

Fungsi *movement* dibagi menjadi aktivitas-aktivitas meliputi:

a. Penerimaan (*Receiving*)

Aktivitas penerimaan barang seperti pembongkaran muatan, penghitungan kuantitas yang diterima dan inspeksi kualitas dan kerusakan.

b. *Put Away*

Merupakan proses pemindahan barang dari penerimaan ke gudang penyimpanan.

c. *Customer Order Picking*

Merupakan aktivitas pemindahan barang dari gudang penyimpanan atau dari lokasi picking untuk kemudian disiapkan untuk proses pengiriman.

d. *Packing*

Proses *packing* merupakan proses pengepakan barang yang akan dikirim ke konsumen.

e. *Cross Docking*

Proses ini merupakan pemindahan barang dari area *receiving* langsung ke lokasi *shipping* tanpa melalui aktivitas penyimpanan di gudang.

f. *Shipping*

Aktivitas ini merupakan pengiriman produk dan meliputi proses pembuatan pemeriksaan kesempurnaan pesanan

2. Penyimpanan (*Storage*)

Merupakan aktivitas penyimpanan barang berupa bahan baku (*raw material*) dan barang jadi (*finished goods*).

3. Pertukaran informasi (*Transfer Information*)

Merupakan aktivitas pertukaran informasi seperti informasi mengenai stok barang yang ada di gudang atau informasi lain yang berguna.

## 2.7 Penyimpanan

Penyimpanan adalah menempatkan barang untuk disimpan pada tempat yang ditetapkan atau menempatkan barang dalam kondisi tunggu untuk dipersiapkan pada proses atau distribusinya.

Menurut Hadiguna (2008), pengaturan dan tata letak suatu gudang dapat dilihat dalam beberapa bentuk kebijakan penyimpanan yang ditentukan perusahaan, dimana metode terbaik yang akan diambil tergantung pada karakteristik item. Kebijakan-kebijakan tersebut antara lain:

1. Kebijakan Penyimpanan Acak (*Random Storage Policy*)

Penyimpanan item yang datang di setiap lokasi yang tersedia, dimana setiap item mempunyai probabilitas sama pada setiap lokasi.

2. Kebijakan Penyimpanan Tetap (*Dedicated Storage Policy*)

Item disimpan pada lokasi tertentu tergantung tipe itemnya. Kebijakan demikian didesain dengan luas penyimpanan setiap item sama dengan level maksimal persediaan.

3. *Cube Per-Order Index Policy*

Rasio kebutuhan *space* penyimpanan item dengan jumlah transaksi *shipping* dan *receiving* untuk itemnya. Item *shipping* dan *receiving* terbesar sedikit dekat dengan titik *Input/Output (I/O)*.

## 2.8 Euclidean Distance

Dalam matematik, *euclidean distance* atau jarak *euclidean* adalah jarak garis lurus “biasa” antara dua titik dalam ruang *euclidean*. Literatur yang lebih tua menyebut metrik sebagai metrik *pythagoras*. Sistem pengukuran dengan jarak *euclidean* sering digunakan karena lebih mudah dimengerti, serta hasil dari pengukurannya memiliki tingkat akurat yang tinggi. Fungsi *euclidean* juga sering digunakan dalam menentukan jarak geografis suatu wilayah. Biasanya fungsi ini digunakan untuk mengetahui jarak dari lokasi awal ke tujuan akhir dengan

pertimbangan tertentu. Adapun rumus yang digunakan untuk menghitung jarak dengan *Euclidean Distance* dapat dilihat di bawah ini (Adiwilaga, 2014):

1. Rumus *Euclidean*

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Keterangan:

$x_1$  = koordinat x pada pusat penelitian/titik awal

$y_1$  = koordinat y pada pusat penelitian/titik awal

$x_2$  = koordinat x pada tujuan penelitian/titik tujuan

$y_2$  = koordinat y pada tujuan penelitian/titik tujuan

$d$  = jarak antara pusat penelitian x dan y

2. Rumus Jarak *Euclidean*

$$\text{Jarak} = \sqrt{(\text{Latitude1} - \text{Latitude2})^2 + (\text{Longitude1} - \text{Longitude2})^2}$$

Keterangan:

*Latitude1* = koordinat *Latitude1* pada titik awal

*Latitude2* = koordinat *Latitude2* pada titik tujuan

*Longitude1* = koordinat *Longitude1* pada titik awal

*Longitude2* = koordinat *Longitude2* pada titik tujuan

### 2.8.1 Lintang/*Latitude*

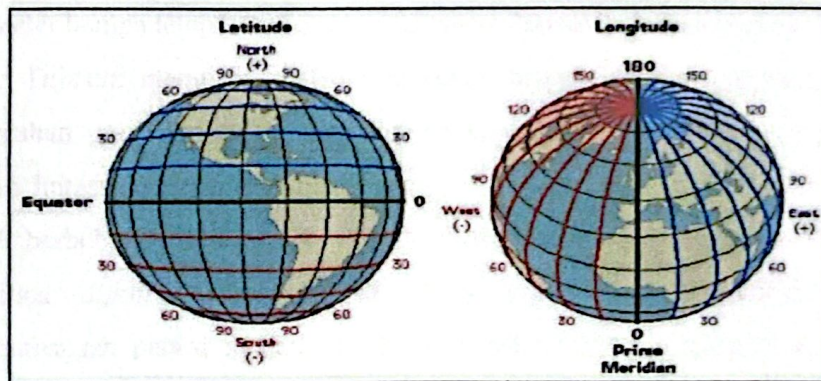
*Latitude* adalah garis mendatar. Titik 0 adalah sudut ekuator tanda + menunjukkan arah ke atas menuju kutub utara, sementara tanda minus di koordinat menuju ke kutub selatan (Awangga, 2019). *Latitude* adalah koordinat geografis yang menentukan posisi utara - selatan dari suatu titik di permukaan bumi. *Latitude* juga merupakan sudut yang berkisar dari 0 ° di Ekuator hingga 90 ° (Utara atau Selatan) di kutub.

### 2.8.2 Bujur/*Longitude*

*Longitude* adalah garis bujur, dimana garis bujur ini diawali dari titik 0 sampai 180 ke arah sebaliknya (Awangga, 2019). *Longitude* atau garis bujur juga merupakan koordinat geografis yang menentukan posisi Timur - Barat dari suatu

titik pada permukaan bumi, atau permukaan benda angkasa. Garis bujur adalah pengukuran sudut, biasanya dinyatakan dalam derajat.

Adapun perbedaan garis Lintang/*Latitude* dengan Bujur/*Longitude* dapat dilihat pada Gambar II.6 di bawah ini:



Gambar II.6 Garis *Latitude* dan *Longitude*  
Sumber: Awangga (2019).

## 2.9 Algoritma *Dijkstra*

Pada bidang seperti transportasi, jaringan komunikasi dan lain-lain, terdapat suatu masalah yang berhubungan dengan penentuan lintasan terpendek. Masalah lintasan terpendek berkaitan penentuan lintasan dengan biaya minimal dari titik awal dalam graf berbobot ke titik tujuan dalam graf tersebut. Pencarian rute terpendek termasuk ke dalam materi teori graf. Algoritma yang sangat terkenal untuk menyelesaikan persoalan ini adalah algoritma *dijkstra*. Algoritma *dijkstra* dianggap cocok karena mudah digunakan oleh *user* dalam penggunaannya hanya dengan menentukan titik awal dan titik tujuan.

Menurut Marsudi (2016), algoritma *dijkstra* adalah sebuah algoritma yang digunakan untuk menentukan lintasan terpendek dalam graf tak berarah maupun graf berarah berbobot tanpa mengenumerasi secara eksplisit semua lintasan yang mungkin. Algoritma *dijkstra* digunakan untuk menentukan lintasan terpendek diantara pasangan-pasangan titik dalam suatu graf.

*Dijkstra* adalah algoritma yang digunakan untuk mencari lintasan terpendek pada sebuah graf berarah. Contoh penerapan algoritma ini adalah

lintasan terpendek yang menghubungkan antara dua kota berlainan tertentu. Kasus ini sering disebut *single-source single destination shortest path problems*. Cara kerja algoritma *dijkstra* memakai strategi *greedy*, dimana pada setiap langkah dipilih sisi dengan bobot terkecil yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih. Algoritma *Dijkstra* membutuhkan parameter berupa tempat asal dan tempat tujuan (Kusuma, Jefri, & Agung, 2019).

*Dijkstra* merupakan salah satu varian bentuk algoritma populer dalam pemecahan persoalan terkait masalah optimasi pencarian lintasan terpendek sebuah lintasan yang mempunyai panjang minimum dari verteks *a* ke *z* dalam *graph* berbobot (Simamora, 2018). Di dalam melakukan perhitungan dengan algoritma *dijkstra*, ada beberapa skema umum yang digunakan untuk pengembangan pencarian dari satu titik ke titik selanjutnya tahap demi tahap. Berikut urutan logika dari Algoritma *Dijkstra* (Geovani, 2016):

1. Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu *set* nilai 0 pada *node* awal dan nilai tak hingga terhadap *node* lain (belum terisi)
2. *Set* semua *node* “belum terpilih” dan *set node* awal sebagai “*node* keberangkatan”
3. Dari *node* keberangkatan, pertimbangkan *node* tetangga yang belum terpilih dan hitung jaraknya dari titik keberangkatan. Sebagai contoh, jika titik keberangkatan A ke B memiliki bobot jarak 6 dan dari B ke *node* C berjarak 2, maka jarak ke C melewati B menjadi  $6+2=8$ . Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru.
4. Saat kita selesai mempertimbangkan setiap jarak terhadap *node* tetangga, tandai *node* yang telah terpilih sebagai “*Node* terpilih”. *Node* terpilih tidak akan pernah dicek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.
5. *Set* “*Node* belum terpilih” dengan jarak terkecil (dari *node* keberangkatan) sebagai “*Node* Keberangkatan” selanjutnya dan lanjutkan dengan kembali ke *step* 3.

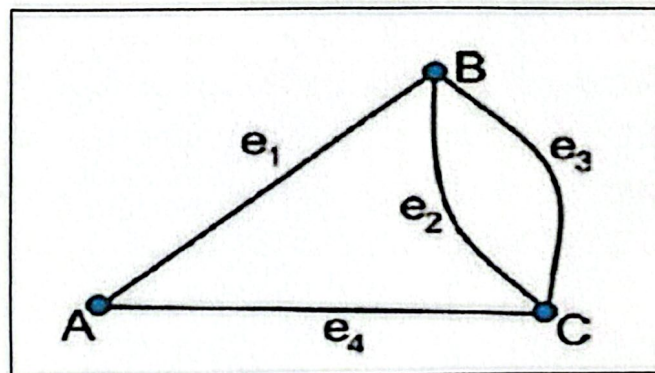
### 2.9.1 Graf

Graf adalah suatu diagram yang memuat informasi tertentu jika diinterpretasikan secara tepat. Dalam kehidupan sehari-hari, graf digunakan untuk menggambarkan berbagai macam struktur yang ada. Tujuannya adalah sebagai visualisasi obyek-obyek agar lebih mudah dimengerti. Beberapa contoh graf yang sering dijumpai dalam kehidupan sehari-hari antara lain: struktur organisasi, bagan alir pengambilan mata kuliah, peta, rangkaian listrik, dan lain-lain (Simamora, 2018).

Menurut Geovani (2016), *Graf* adalah kumpulan simpul (*node*) yang dihubungkan satu sama lain melalui sisi/busur. Suatu *graf* terdiri dari dua himpunan yaitu himpunan  $V$  dan himpunan  $E$ .

1. *Vertex* (simpul) –  $V$  = Himpunan simpul yang terbatas dan tidak kosong.
2. *Edge* (sisi/busur) –  $E$  = Himpunan busur yang menghubungkan sepasang simpul

Dapat dikatakan graf adalah kumpulan dari simpul-simpul yang dihubungkan oleh sisi-sisi. Di bawah ini Gambar II.7 adalah Graf:

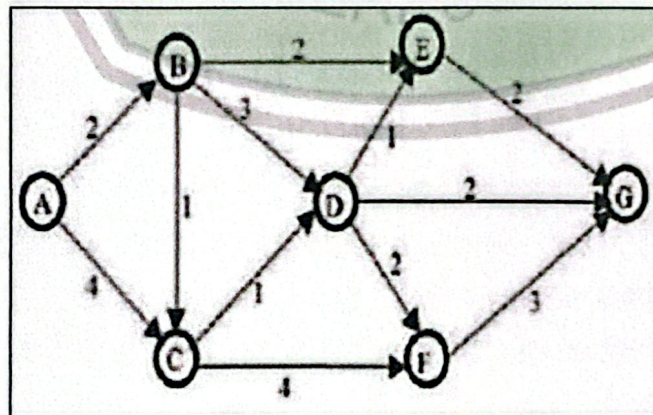


Gambar II.7 Graf  
Sumber: Geovani (2016).

Dari gambar diatas yang disebut *vertex*,  $V = \{A, B, C\}$  dan yang disebut *edge*,  $E = \{e_1, e_2, e_3, e_4\}$ .

### 2.9.1.1 Graf Berarah dan Berbobot

Graf berarah dan berbobot ialah graf yang tiap busur mempunyai anak panah dan bobot. Pada Gambar II.8 menunjukkan graf berarahan berbobot yang terdiri dari tujuh titik yaitu titik A, B, C, D, E, F, G. Titik menunjukkan arah ke titik B dan C, titik B menunjukkan arah ke titik D dan C dan seterusnya. Bobot antar titik A dan titik B pun telah diketahui.



Gambar II.8 Graf Berarah dan Berbobot  
Sumber: Geovani (2016).

### 2.10 System Development Life Cycle (SDLC)

Menurut Dennis, Wixom, & Tegarden (2015), *System Development Life Cycle* (SDLC) memiliki 4 perangkat fase dasar yaitu *planning*, *analysis*, *design*, dan *implementation*. Berbeda proyek mungkin menekankan perbedaan bagian-bagian dari SDLC atau pendekatan fase SDLC dalam cara yang berbeda, tetapi semua proyek memiliki elemen dari 4 fase tersebut. Berikut di bawah ini merupakan penjabaran dari setiap fase dalam SDLC:

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

Tahap *planning* adalah proses dasar untuk memahami mengapa sistem informasi harus dibangun dan menentukan bagaimana tim proyek akan membangunnya.

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

Tahap ini menjawab pertanyaan siapa yang akan menggunakan sistem, apa yang akan dilakukan sistem dan di mana serta kapan akan digunakan. Tim

proyek menyelidiki sistem yang ada saat ini, mengidentifikasi peluang untuk perbaikan dan mengembangkan sistem baru.

### 3. *Design* (Perancangan)

Tahap *design* memutuskan bagaimana sistem akan beroperasi, dalam hal perangkat keras, perangkat lunak, infrastruktur jaringan, antarmuka pengguna, formulir, laporan, program spesifik, *database* dan *file* yang akan dibutuhkan., langkah-langkah dalam tahap perancangan menentukan secara tepat bagaimana sistem akan beroperasi.

### 4. *Implementation* (Implementasi)

Tahap akhir dalam SDLC adalah tahap implementasi dimana sistem benar-benar dibangun atau dibeli. Contohnya paket *design software*, implementasi adalah fase yang paling mendapat perhatian, karena merupakan bagian terlama dan paling mahal dari proses pembangunan.

## 2.11 Metode *Waterfall*

Menurut Pressman (2015), metode pengembangan sistem dengan *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Disebut dengan *waterfall* karena model pengembangannya yang seperti “air mancur” secara berurutan dari satu fase ke fase berikutnya. Tahap demi tahap yang dilalui pada metode ini harus menunggu selesainya tahap sebelumnya dan berjalan berurutan hingga akhir. Menurut Dennis, Wixom, & Tegarden (2015) terdapat 5 langkah dalam metode *waterfall* sebagai berikut:

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

Dalam tahapan ini, menjelaskan untuk melanjutkan proyek yang telah dipilih, rencana kerja yang matang juga disusun untuk menjalankan tahapan lainnya. Ditentukan secara detail rencana kerja yang harus dikerjakan, durasi yang diperlukan masing-masing tahap, sumber daya manusia, perangkat lunak, dokumentasi, perangkat keras, maupun *financial*. Pembuatan perencanaan bukan langkah mudah karena untuk mengestimasi beban kerja dan durasi dari masing-masing tahap dibutuhkan pengalaman yang banyak. Kesalahan pada tahap ini akan

mengakibatkan keuntungan yang diperoleh tidak maksimal, bahkan bisa rugi.

## 2. Analisis (*Analysis*)

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

## 3. Perancangan (*Design*)

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

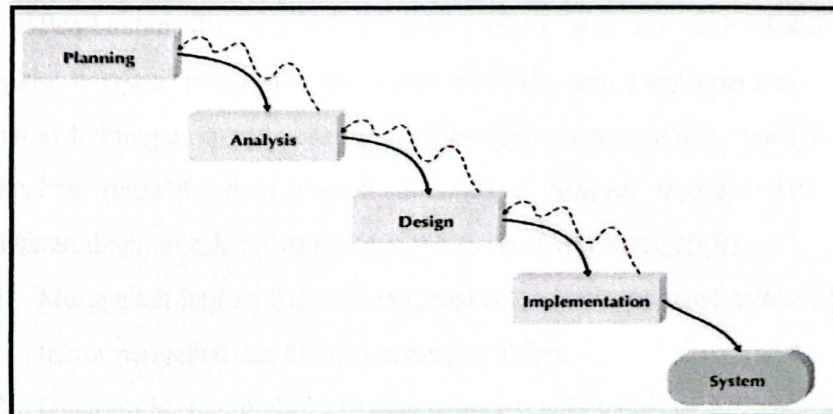
## 4. Implementasi (*Implementation*)

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

## 5. Sistem

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

Berikut di bawah ini adalah Gambar II.9 yang merupakan lima langkah dari *Model Waterfall*:



Gambar II.9 Model *Waterfall*  
Sumber: Dennis, Wixom, & Tegarden (2015)

### 2.12 Analisis dan Perancangan Sistem Berorientasi Objek

Analisis berorientasi objek (OOA) adalah proses menganalisis tugas untuk mengembangkan sebuah model konseptual yang kemudian dapat digunakan untuk menyelesaikan tugas. Sebuah model OOA yang mempunyai ciri khas khusus akan menjelaskan perangkat lunak komputer yang dapat digunakan untuk memenuhi keperluan pengguna berdasarkan kriteria yang telah ditentukan. Menurut Satzinger et al. dalam Surya dkk (2014) OOA adalah suatu cara untuk menentukan semua tipe object yang bekerja di dalam sebuah sistem dan menggambarkan interaksi yang dibutuhkan user untuk menyelesaikan tugas-tugasnya. Model konseptual yang dihasilkan dari OOA biasanya akan terdiri dari serangkaian kasus penggunaan, satu atau lebih UML diagram kelas, dan sejumlah diagram interaksi. Ini juga termasuk beberapa jenis antarmuka pengguna *mock-up*.

### 2.13 Fishbone Diagram

Profesor Kouru Ishikawa mengembangkan diagram Ishikawa, yang biasa dikenal sebagai diagram “tulang ikan” karena bentuknya yang unik. Diagram tulang ikan (*fishbone diagram*) adalah suatu cara untuk secara visual mengorganisasi dan menguji semua faktor yang bisa memengaruhi situasi tertentu

dengan mengidentifikasi semua penyebab yang mungkin, yang menghasilkan suatu efek (Michalko, 2010).

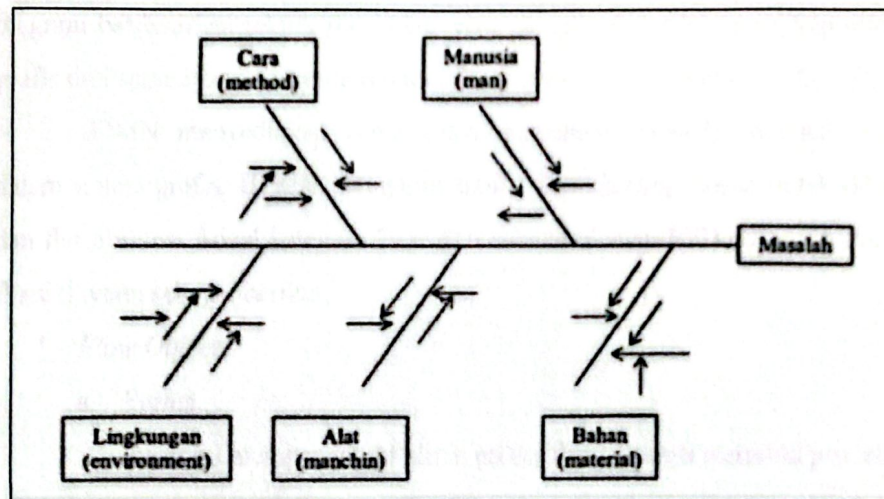
Pembuatan diagram ini bertujuan untuk mencari faktor-faktor yang mungkin menjadi penyebab dari suatu masalah atau penyimpangan. Dengan diketahui hubungan antara sebab dan akibat dari suatu masalah, maka tindakan pemecahan masalah akan mudah ditentukan. Adapun manfaat dari proses pembuatan diagram tulang ikan adalah antara lain (Michalko, 2010):

1. Merupakan latihan dalam menggunakan logika bagaimana mencari faktor-faktor penyebab dan hubungan dengan akibat.
2. Diagram ini merupakan alat atau pemandu dalam diskusi.
3. Dapat diperoleh kemungkinan penyebab yang sebanyak mungkin, yang menimbulkan suatu akibat.

Dalam pembuatan diagram tulang ikan, akibat atau permasalahan digambarkan dalam bagian kepala ikan, sedangkan faktor-faktor penyebab diletakkan sebagai tulang ikan. Pertama, permasalahan biasanya digolongkan menjadi beberapa golongan besar, kemudian penjabaran selanjutnya yang lebih terperinci dapat dibuat dengan mengajukan pertanyaan “mengapa” secara terus-menerus. Penggolongan dalam garis besar faktor-faktor penyebab dimaksud biasa dibagi atas:

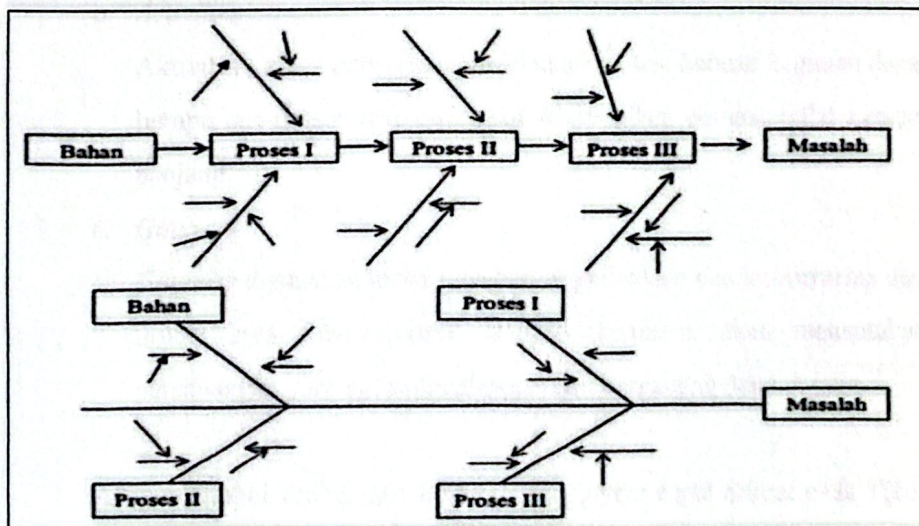
1. Bahan (*material*).
2. Alat (*machine*).
3. Manusia (*man*)
4. Cara (*method*).
5. Lingkungan (*environment*).

Pembuatan diagram tulang ikan berdasarkan pengelompokkan sebab-sebab tersebut di atas sering disebut dengan diagram tulang ikan tipe rangkuman sebab. Berikut Gambar II.10 Diagram Tulang Ikan Tipe Rangkuman Sebab:



Gambar II.10 Diagram Tulang Ikan Tipe Rangkuman Sebab  
Sumber: Kuswadi & Mutiara (2004).

Diagram tulang ikan dapat juga digambarkan berdasarkan proses produksi (tipe klasifikasi proses produksi), dapat dilihat pada Gambar II.11 di bawah ini:



Gambar II.11 Diagram Tulang Ikan Tipe Klasifikasi Proses Produksi  
Sumber: Kuswadi & Mutiara (2004).

#### 2.14 Business Process Modelling Notation (BPMN)

BPMN merupakan teknik yang memungkinkan semua pihak yang terlibat dalam proses berkomunikasi secara jelas, benar dan efisien. Dengan cara ini, mendefinisikan notasi dan *semantic* Diagram Proses Bisnis (BPD). BPD adalah

diagram berdasarkan teknik *flowchart*, yang dirancang untuk menyajikan urutan grafis dari semua kegiatan yang terjadi selama proses (Maniah & Hamidin, 2017).

BPMN menyediakan kemampuan memahami prosedur internal bisnis dalam notasi grafis. BPMN mengikuti tradisi *flowcharting* notasi untuk dibaca dan fleksibilitas. Ada 4 kategori dasar dari elemen-elemen BPD (Diagram Proses Bisnis), yaitu sebagai berikut:

1. *Flow Objects*

a. *Events*

*Event* ini mempengaruhi aliran proses dan biasanya memiliki penyebab (pemicu) atau dampak (hasil). *Event* disimbolkan dengan lingkaran kecil dengan garis tipis (*start event*), lingkaran kecil dengan garis ganda (*intermediate event*), lingkaran kecil dengan garis tebal (*end event*).

b. *Activities*


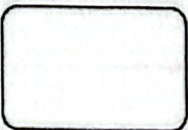
Aktivitas artinya perusahaan melakukan proses. Sebuah kegiatan dapat berupa sub-proses dan *task* yang disimbolkan dengan bulat persegi panjang.

c. *Gateway*

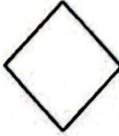
*Gateway* digunakan untuk mengontrol perbedaan dan konvergensi dari urutan arus dalam proses. Dengan demikian, akan menentukan percabangan, *forking*, penggabungan dan bergabung dengan jalur.

Adapun simbol-simbol dari elemen *Flow Object* dapat dilihat pada Tabel di bawah ini:

Tabel II.1 Simbol-Simbol *Flow Objects*

Nama Simbol	Gambar Simbol
<i>Events</i>	
<i>Activities</i>	

Tabel II.1 Simbol-Simbol *Flow Objects* (lanjutan)

Nama Simbol	Gambar Simbol
<i>Gateways</i>	

(Sumber: Maniah dan Hamidin 2017)

## 2. *Connecting Objects*

*Flow object* yang dihubungkan satu sama lain dalam suatu diagram untuk menciptakan struktur dasar proses bisnis. Ada 3 *connecting object* yang ada di fungsi ini, yaitu:

### a. *Sequence Flow*

Sebuah arus urutan digunakan untuk menunjukkan urutan kegiatan yang akan dilakukan dalam proses, disimbolkan dengan garis dengan panah solid.

### b. *Message Flow*


Direpresentasikan oleh garis terputus-putus dengan kepala panah terbuka dan digunakan untuk menunjukkan aliran pesan antara dua partisipan proses (entitas bisnis atau peran bisnis).

### c. *Association*



Asosiasi direpresentasikan oleh garis titik-titik dengan kepala panah garis dan digunakan untuk asosiasi data, teks dan artefak lainnya dengan *flow objects*. Asosiasi digunakan untuk memperlihatkan *input* dan *output* aktivitas.

Adapun simbol-simbol dari elemen *Connecting Objects* dapat dilihat pada Tabel di bawah ini:

Tabel II.2 Simbol-Simbol *Connecting Objects*

Nama Simbol	Gambar Simbol
<i>Sequence Flow</i>	

Tabel II.2 Simbol-Simbol *Connecting Objects* (lanjutan)

Nama Simbol	Gambar Simbol
<i>Message Flow</i>	
<i>Association</i>	

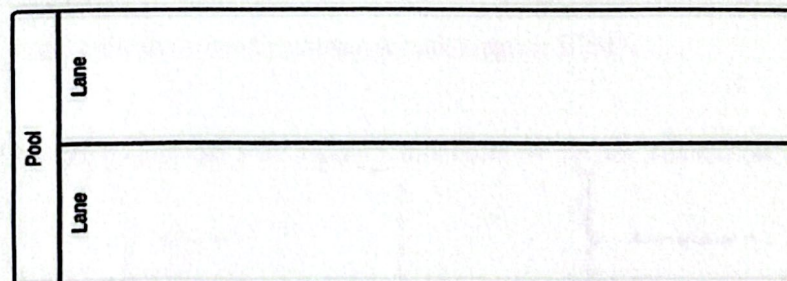
(Sumber: Maniah dan Hamidin 2017)

### 3. *Swimlanes*

BPMN membagi *swimlanes* menjadi 2 *construct* utama, yaitu:

- Pool*: merepresentasikan partisipan dalam sebuah proses. *Pool* juga bertindak sebagai garis *container* untuk memisahkan sekumpulan aktivitas dari *pool-pool* yang lainnya.
- Lane*: sebuah *lane* merupakan sub-partisi di dalam *pool* dan akan memperpanjang seluruh panjang *pool*, baik secara vertikal atau horizontal.

Di bawah ini Gambar II.12 adalah Simbol *Swimlanes*:



Gambar II.12 Simbol *Swimlanes*  
(Sumber: Maniah dan Hamidin, 2017)





### 4. *Artifacts*

Sejumlah artifak dapat ditambahkan ke dalam sebuah diagram yang sesuai untuk konteks proses bisnis yang dimodelkan. Artifak tersebut adalah sebagai berikut:

- Data object*: merupakan mekanisme untuk menunjukkan bagaimana data dibutuhkan atau diproduksi oleh kegiatan. Mereka dikoneksikan ke

aktivitas melalui *association*. Adapun simbol-simbol dari elemen *Data Object* dapat dilihat pada Tabel di bawah ini:

Tabel II.3 Simbol-Simbol *Data Objects*

Nama Simbol	Gambar Simbol
<i>Data Input Symbol</i>	
<i>Data Output Symbol</i>	
<i>Data Collection Symbol</i>	
<i>Data Storage Symbol</i>	

(Sumber: Maniah dan Hamidin 2017)

- b. *Group*: sebuah *group* direpresentasikan melalui empat persegi panjang yang sudutnya bulat dengan garis terputus-putus. *Group* dapat digunakan untuk dokumentasi atau analisis fungsi, tetapi tidak berdampak pada *sequence flow*.
- c. *Annotation*: mekanisme untuk *modeler* untuk memberikan informasi teks tambahan untuk pembaca sebuah diagram BPMN.

Di bawah ini Gambar II.13 adalah Simbol dari *Group* dan *Annotation*:



Gambar II.13 Simbol Group & Annotation  
(Sumber: Maniah dan Hamidin, 2017)

## 2.15 Unified Modelling Language (UML)

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

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

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

Blok pembangunan utama UML adalah diagram. Beberapa diagram ada yang rinci dan lainnya ada yang bersifat umum misalnya diagram kelas. Para pengembang sistem berorientasi objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang dirancang. UML memungkinkan para anggota tim untuk bekerja sama dengan bahasa model yang sama dalam mengaplikasikan beragam sistem.

### 2.15.1 Diagram UML

Menurut Widodo (2011), beberapa literatur menyebutkan bahwa UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain (Widodo, 2011):

1. Diagram Kelas (*Class Diagram*)

Bersifat statis, diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi serta relasi-relasi. Diagram ini umumnya dijumpai pada pemodelan sistem berorientasi objek. Meski bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif.

2. Diagram Paket (*Package Diagram*)

Bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.


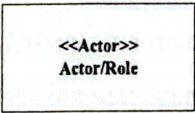
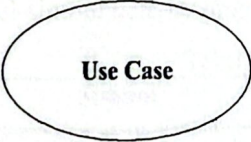
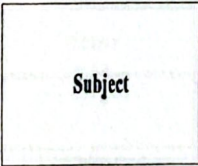
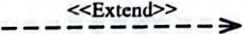
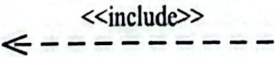
3. Diagram *Use Case* (*Use Case Diagram*)  
Bersifat statis. Diagram ini memperlihatkan himpunan *use case* dan aktor-aktor. Diagram ini sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.
4. Diagram *Sequence* (*Sequence Diagram*)  
Bersifat dinamis. Diagram *sequence* adalah interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.
5. Diagram Komunikasi (*Communication Diagram*)  
Bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi UML yang menekankan organisasi struktural dari objek-objek yang menerima serta menerima pesan.
6. Diagram *Statechart* (*Statechart Diagram*)  
Bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat sistem (*state*), transisi, kejadian serta aktivitas.
7. Diagram Aktivitas (*Activity Diagram*)  
Bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan member tekanan pada aliran kendali antar objek.
8. Diagram Komponen (*Component Diagram*)  
Bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem pada komponen yang telah ada sebelumnya.
9. Diagram *Deployment* (*Deployment Diagram*)  
Bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan. Memuat simpul-simpul beserta komponen yang di dalamnya.

### 2.15.2 *Use Case Diagram*



*Use case diagram* merupakan suatu diagram yang menangkap kebutuhan bisnis untuk sistem dan untuk menggambarkan interaksi antara sistem dan lingkungannya (Dennis, Wixom, & Tegarden 2015). *Use case diagram* secara grafis menggambarkan interaksi secara sistem eksternal dan pengguna. Dengan

kata lain *use case diagram* secara grafis mendeskripsikan siapa yang akan menggunakan sistem dan dalam cara apa pengguna (*user*) mengharapkan interaksi dengan sistem itu. *Use case diagram* secara naratif digunakan untuk secara tekstual menggambarkan sekuensi langkah-langkah dari tiap interaksi. Adapun simbol-simbol *use case diagram* dapat dilihat pada Tabel II.4 di bawah ini:

Tabel II. 4 Simbol-simbol *Use-Case Diagram*

Simbol	Deskripsi
 <p>Actor/Role</p> 	<ol style="list-style-type: none"> <li>1. Menyatakan seseorang atau sistem yang mendapatkan keuntungan dari sistem.</li> <li>2. Digambarkan sebagai gambar <i>stick</i>/gambar orang (<i>default</i>) atau jika bukan seorang aktor manusia, digambarkan dengan suatu kotak dengan tanda <code>&lt;&lt;actor&gt;&gt;</code> di dalamnya (alternatif).</li> <li>3. Dilabelkan dengan peran/<i>role</i> dari aktor.</li> </ol>
 <p>Use Case</p>	<ol style="list-style-type: none"> <li>1. Merepresentasikan bagian utama.</li> <li>2. Dapat berupa perluasan <i>use case</i> lain.</li> <li>3. Dapat termasuk di dalam <i>use case</i> lain.</li> <li>4. Diletakkan di dalam batas sistem.</li> <li>5. Dinamakan dengan frasa kata kerja.</li> </ol>
 <p>Subject</p>	<ol style="list-style-type: none"> <li>1. Menyertakan nama subjek di dalam maupun di atas.</li> <li>2. Merepresentasikan ruang lingkup dari subjek sistem atau proses bisnis.</li> </ol>
	<p>Merepresentasikan perluasan (<i>extend</i>) dari <i>use case</i> lain untuk menyertakan perilaku opsional/<i>optional</i> (tidak wajib). Disimbolkan dengan anak panah yang digambarkan dari perluasan <i>use case</i> ke <i>use case</i> dasar.</p>
	<ol style="list-style-type: none"> <li>1. Merepresentasikan fungsionalitas suatu <i>use case</i> dengan <i>use case</i> lainnya.</li> <li>2. Disimbolkan dengan anak panah dari sebuah <i>use case</i> dasar ke <i>use case</i> yang digunakan.</li> </ol>

Tabel II. 4 Simbol-simbol *Use-Case Diagram* (lanjutan).

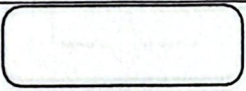
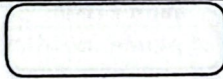
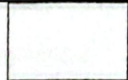
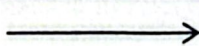
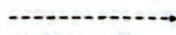
Simbol	Deskripsi
 Generalisasi	<ol style="list-style-type: none"> <li>1. Merepresentasikan <i>use case</i> khusus ke satu <i>use case</i> yang lebih umum.</li> <li>2. Disimbolkan dengan anak panah yang digambarkan dari <i>use case</i> khusus ke <i>use case</i> umum.</li> </ol>
	Menghubungkan suatu aktor dengan <i>use case</i> dengan interaksi antara keduanya.

Sumber: Dennis, Wixom, & Tegarden (2015)






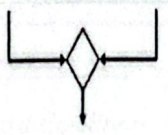
### 2.15.3 Diagram Aktivitas (*Activity Diagram*)

*Activity diagram* dapat juga digunakan untuk memodelkan *action* yang akan dilakukan saat operasi dieksekusi dan memodelkan hasil dari *action* tersebut. *Activity diagram* ialah yang menggambarkan alur kerja bisnis independen dari aliran kegiatan dalam *use case* (Dennis, Wixom, & Tegarden 2015). Adapun simbol-simbol *activity diagram* dapat dilihat pada Tabel II.5:

Tabel II.5 Simbol-simbol *Activity Diagram*

Simbol	Deskripsi
 <i>Actor</i>	Digunakan untuk melakukan tindakan
 <i>Activity</i>	Digunakan untuk menggambarkan dari kejadian.
 <u><i>Class Name</i></u>	Digunakan untuk mewakili suatu objek yang mengalir, diberi label dengan nama kelasnya.
 <i>Control Flow</i>	Menunjukkan urutan eksekusi.
 <i>Object Flow</i>	Menunjukkan aliran dari sebuah objek dari satu kegiatan (atau tindakan)

Tabel II.5 Simbol-simbol *Activity Diagram* (Lanjutan)

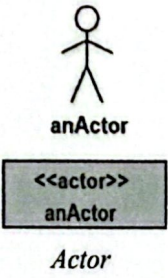
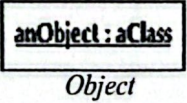
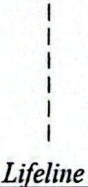
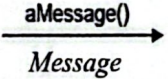

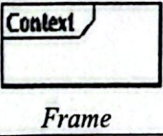

Simbol	Deskripsi
 <i>Initial Node</i>	Merupakan tanda awal dari sebuah aktivitas.
 <i>Final Activity Node</i>	Merupakan tanda berakhirnya sebuah aktifitas.
 <i>Final Flow Node</i>	Digunakan untuk menghentikan aliran kontrol.
 <i>Fork Node</i>	Digunakan untuk membagi perilaku menjadi satu set arus paralel atau bersamaan kegiatan.
 <i>Join Node</i>	Digunakan untuk menunjukkan kegiatan yang digabungkan.
 <i>Merge Node</i>	Digunakan untuk membawa kembali jalur keputusan bersama yang berbeda yang diciptakan menggunakan simpul keputusan.

Sumber: Dennis, Wixom, & Tegarden (2015)

#### 2.15.4 *Sequence Diagram*

*Sequence Diagram* merupakan urutan model dinamis yang menggambarkan contoh *class* yang berpartisipasi dalam *use case* dan pesan yang lewat diantara mereka dari waktu ke waktu (Dennis, Wixom, & Tegarden 2015). Cara membacanya ialah dari atas ke bawah. Setiap diagram sekuensial mempresentasikan satu *flow* dari beberapa *flow* di dalam *use case*. Adapun simbol-simbol *Sequence Diagram* dapat dilihat pada tabel di bawah ini:

Tabel II.6 Simbol- Simbol *Sequence Diagram*

Simbol	Deskripsi
	<ol style="list-style-type: none"> <li>1. Seseorang atau sistem yang mendapat manfaat dan bersifat eksternal dari sistem.</li> <li>2. Berpartisipasi dalam urutan dengan mengirim dan / atau menerima pesan.</li> <li>3. Digambarkan baik sebagai stik (<i>default</i>) atau, jika aktor bukan manusia terlibat, sebagai persegi panjang dengan &lt;&lt;actor&gt;&gt; di dalamnya (alternatif)</li> </ol>
	<ol style="list-style-type: none"> <li>1. Berpartisipasi secara berurutan oleh mengirim dan/atau menerima pesan.</li> <li>2. Ditempatkan di bagian atas diagram.</li> </ol>
	<ol style="list-style-type: none"> <li>1. Menunjukkan kehidupan sebuah objek selama <i>Sequence</i>.</li> <li>2. Berisi sebuah X pada titik dimana kelas tidak lagi berinteraksi</li> </ol>
	Menyampaikan informasi dari satu objek ke objek lainnya.
	Merupakan tes yang harus dipenuhi untuk pesan yang akan dikirim.
	Menunjukkan konteks dalam <i>sequence diagram</i>
	Merupakan sebuah X ditempatkan pada akhir suatu garis hidup untuk menunjukkan bahwa itu akan keluar dari eksistensi.


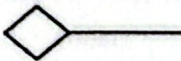
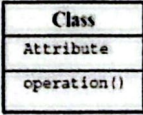
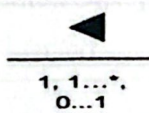

Sumber: Dennis, Wixom, & Tegarden (2015)

### 2.15.5 Class Diagram

Menurut Dennis, Wixom dan Tegarden (2015), *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. *Class diagram* dibangun berdasarkan *use case diagram*, *sequence diagram*, yang telah dibuat sebelumnya. Adapun simbol-simbol *class diagram* dapat dilihat pada Tabel II.7.

Tabel II.7 Simbol-simbol *Class Diagram*

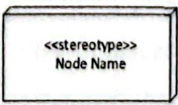
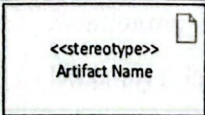
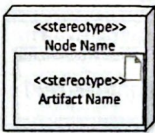
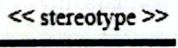
Simbol	Deskripsi
 <i>Generalization</i>	Merupakan sebuah <i>taxonomic relationship</i> antara <i>class</i> yang lebih umum dengan <i>class</i> yang lebih khusus.
 <i>Aggregation</i>	Mewakili hubungan <i>logic</i> antara <i>class</i> dan beberapa <i>class</i> .
	Kelas pada struktur sistem.
 1, 1...* 0...1	Asosiasi yang menghubungkan <i>class</i> dengan beberapa <i>class</i> . Bisa ada di antara satu atau lebih kelas.
 <i>Composition</i>	Mewakili hubungan fisik antara <i>class</i> dan beberapa <i>class</i>

Sumber: Dennis, Wixom, & Tegarden (2015)

### 2.15.6 Deployment Diagram

*Deployment diagram* digunakan untuk mewakili hubungan antara komponen *hardware* yang digunakan dalam infrastruktur fisik sistem Informasi. *Deployment diagram* dapat digunakan untuk menunjukkan hubungan komunikasi antara *node* yang berbeda dalam jaringan. *Deployment diagram* juga dapat digunakan untuk mewakili komponen-komponen *software* dan cara *software* ditempatkan pada arsitektur fisik atau infrastruktur sistem informasi (Dennis, Wixom, & Tegarden 2015). Adapun simbol-simbol *deployment diagram* dapat dilihat pada Tabel II.8 di bawah ini:

Tabel II.8 Simbol- Simbol *Deployment Diagram*

Simbol	Deskripsi
 <i>Node</i>	Menggambarkan sumber daya komputasi dalam sebuah sistem misalnya, <i>computer klien, server, jaringan</i> yang terpisah, atau individu perangkat jaringan.
 <i>Artifact</i>	Menggambarkan spesifikasi dari <i>software</i> atau <i>database</i> , misalnya <i>file sumber, tabel database, executable file</i> .
 <i>Node with a Deployed Artifact</i>	Menggambarkan <i>artifact</i> yang ditempatkan pada <i>node</i> fisik. Mendukung pemodelan distribusi perangkat lunak melalui jaringan.
 <i>Communication Path</i>	Menggambarkan hubungan antara dua node untuk bertukar pesan.

Sumber: Dennis, Wixom, & Tegarden (2015)

## 2.16 Entity Relationship Diagram

*Entity Relationship Diagram* (ERD) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi. ERD sebagai teknik atau alat peraga memberikan dasar untuk desain *database* relasional yang mendasari sistem informasi yang akan dikembangkan. ERD dengan detail merupakan model pendukung dari data yang akan digunakan sebagai spesifikasi untuk menjelaskan *database* (Brandy & Loonam, 2010).

ERD adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak (Ladjamudin, 2013). Penjelasan mengenai komponen-komponen ERD adalah sebagai berikut:

1. *Entity*

*Entity* digambarkan dengan sebuah bentuk persegi panjang. *Entity* adalah sesuatu yang ada di dalam sistem, nyata maupun abstrak dimana data tersimpan atau terdapat data.

2. *Relationship*

*Relationship* digambarkan dengan sebuah bentuk belah ketupat. *Relationship* menggambarkan hubungan yang terjadi antara entitas. Umumnya *Relationship* diberi nama dengan kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya.

3. *Relationship Degree*

*Relationship Degree* atau derajat *relationship* adalah jumlah entitas yang berpartisipasi dalam satu *relationship*.

4. Atribut

Secara umum atribut adalah sifat atau karakteristik dari tiap entitas maupun *relationship*. Maksudnya, atribut adalah elemen dari setiap entitas dan *relationship*.

5. Kardinalitas (*Cardinality*)

Kardinalitas relasi menunjukkan jumlah maksimum tupelo yang dapat berelasi dengan entitas pada entitas yang lain. Terdapat 3 macam kardinalitas relasi, yaitu

- a. *One to One*

Tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.

- b. *One to Many* atau *Many to One*

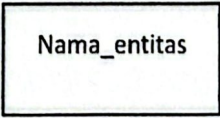
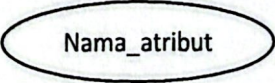

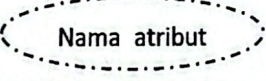

Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu.

- c. *Many to Many*

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya.

Berikut Notasi-notasi yang digunakan dalam ERD dengan notasi Chen, dapat dilihat pada Tabel II.9 di bawah ini:

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

Simbol	Deskripsi
 <p>Nama_entitas</p> <p>Entitas</p>	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama <i>table</i>
 <p>Nama_atribut</p> <p>Atribut</p>	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
 <p>Nama Kunci Primer</p> <p>Atribut Kunci Primer</p>	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id. Kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).
 <p>Nama atribut</p> <p>Atribut Derivatif</p>	<i>Field</i> atau kolom data yang dihasilkan dari hasil perhitungan entitas lain.
 <p>Asosiasi/association</p>	Penghubung antara relasi dan entitas.

Sumber: Sukamto dan Shalahuddin (2013)

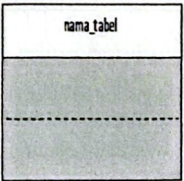
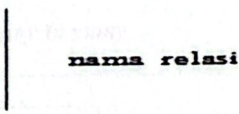
### 2.17 *Conceptual Data Model*

Menurut Lutfifarid (2018), CDM adalah model yang dibuat berdasarkan anggapan bahwa dunia nyata terdiri dari koleksi obyek-obyek dasar yang dinamakan entitas (*entity*) serta hubungan (*relationship*) antara entitas-entitas itu. Biasanya CDM direpresentasikan dalam bentuk *Entity Relationship*. Manfaat penggunaan CDM dalam perancangan *database* adalah sebagai berikut:

1. Memberikan gambaran yang lengkap dari struktur basis data yaitu arti, hubungan, dan batasan-batasan.
2. Alat komunikasi antar pemakai basis data, designer, dan analis.

CDM adalah konsep yang berkaitan dengan pandangan pemakai terhadap data yang disimpan dalam *database*. CDM dibuat dalam bentuk tabel-tabel tanpa tipe data yang menggambarkan relasi antar tabel untuk keperluan implementasi ke basis data. CDM merupakan hasil penjabaran lebih lanjut dari ERD (Rosa & Shalahuddin, 2015). Di bawah ini Tabel II.10 adalah Simbol-simbol CDM:

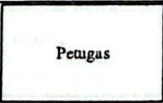
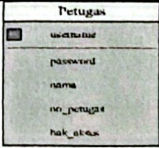
Tabel II.10 Simbol-simbol CDM

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


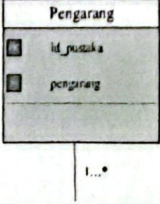
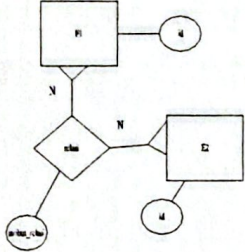
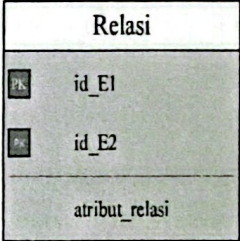
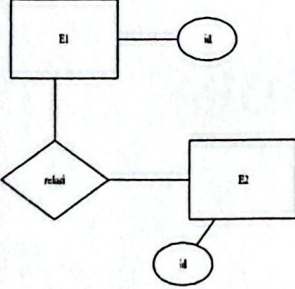
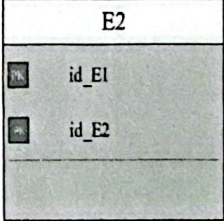
Sumber: (Rosa dan Shalahuddin, 2018)

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

Tabel II.11 Simbol Aturan CDM

ERD	CDM
 <p>Entitas</p>	 <p>Menjadi sebuah tabel tersendiri</p>

Tabel II.11 Simbol Aturan CDM (lanjutan)

ERD	CDM
 <p data-bbox="403 387 603 421">Atribut <i>Multivalue</i></p>	 <p data-bbox="675 506 1286 636">Menjadi sebuah tabel tersendiri dengan kunci primer (<i>primary key</i>) adalah kunci primer pada entitas dan memiliki atribut dengan nama seperti pada atribut entitas</p>
 <p data-bbox="352 983 651 1066">Relasi dengan kardinalitas <i>many to many</i></p>	 <p data-bbox="675 947 1286 1077">Menjadi sebuah tabel tersendiri dengan kunci primer adalah atribut yang menjadi kunci primer di kedua entitas yang direlasikannya</p>
 <p data-bbox="352 1460 651 1543">Relasi dengan kardinalitas <i>one to one</i></p>	 <p data-bbox="675 1402 1286 1576">Kunci primer salah satu entitas akan dijadikan kunci asing (<i>foreign key</i>) pada tabel yang lain dan kunci asing itu dijadikan kunci primer juga, dengan kata lain, relasi tidak menjadi tabel sendiri</p>

Sumber: (Rosa dan Shalahuddin, 2018)

## 2.18 Kamus Data

Menurut Rosa dan Shalahuddin (2018), mengemukakan bahwa kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak

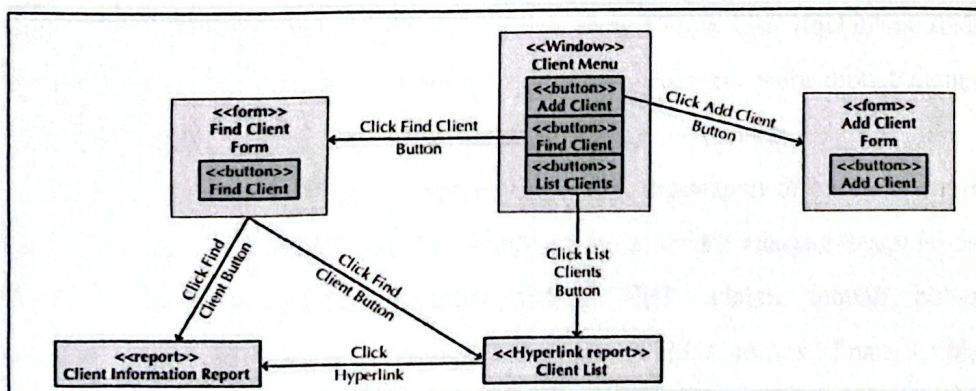
sehingga masukan (*Input*) data keluaran (*output*) dapat dipahami secara umum.

Kamus data biasanya berisi:

1. Nama-nama dari data
2. Digunakan pada-merupakan proses-proses yang terkait data
3. Deskripsi – merupakan deskripsi data
4. Informasi tambahan – seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data

### 2.19 Windows Navigation Diagram (WND)

*Windows Navigation Diagram* merupakan struktur navigasi yang menentukan cara kerja masing-masing *interface* untuk menyediakan fungsionalitas pengguna. WND digunakan untuk menunjukkan bagaimana semua *interface*, *form*, dan *report* yang digunakan oleh sistem terkait dan bagaimana pengguna berpindah dari satu *interface* ke *interface* yang lain (Dennis, Wixom, & Tegarden, 2015). Berikut ini merupakan contoh dari *windows navigation diagram* dapat dilihat pada Gambar II.14 di bawah ini:



Gambar II.14 Contoh Windows Navigation Diagram  
Sumber: Dennis, Wixom, & Tegarden (2015)

### 2.20 Framework

*Framework* adalah kumpulan perintah atau fungsi dasar yang membentuk aturan-aturan tertentu dan saling berinteraksi satu sama lain sehingga dalam pembuatan aplikasi *website*, harus mengikuti aturan dari *framework* tersebut (Wardana, 2010). Dengan *framework*, tidak perlu memikirkan kode perintah/fungsi dasar dari aplikasi *website* yang akan dibuat. Hal-hal penunjang

lainnya yang telah disediakan oleh *framework* seperti koneksi *database*, validasi *form*, GUI dan keamanan telah disediakan oleh *framework*. Adapun keuntungan yang didapat dari penggunaan *framework* adalah sebagai berikut:

1. Waktu pembuatan aplikasi *website* akan jauh lebih singkat
2. Kode aplikasi *website* menjadi lebih mudah dibaca, karena sedikit dan sifatnya pokok.
3. *Website* menjadi lebih mudah diperbaiki, karena tidak perlu fokus pada semua komponen kode *website*, terutama kode *system framework*.
4. Tidak perlu lagi membuat kode penunjang aplikasi seperti koneksi *database*, validasi *form*, GUI, dan keamanan.
5. Pikiran menjadi lebih fokus pada kode alur permasalahan *website*, seperti apa saja yang ditampilkan dan layanan apa saja pada *website* tersebut.

### 2.21 PHP (*Hypertext Preprocessor*)

Menurut Nugroho (2012) PHP merupakan salah satu bahasa pemrograman yang digunakan untuk membuat aplikasi berbasis web (*website*, blog, atau aplikasi web). PHP termasuk bahasa pemrograman yang hanya bisa dijalankan disisi *server*, atau sering disebut *Side Server Language*. Program yang dibuat dengan kode PHP tidak bisa berjalan kecuali dijalankan pada *server web*.

PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin seperti *Linux*, *Unix*, *Macintos*, dan *Windows* serta secara *runtime console* dan dapat menjalankan perintah-perintah sistem. PHP adalah sebuah bahasa pemrograman yang dapat menghasilkan kode HTML secara dinamis, bisa berubah-ubah sesuai dengan keinginan programmernya, dan merupakan sebuah *web* berbasis server (*server-side*). Artinya kode pemrograman dijalankan di server, kalau tidak ada server maka kode PHP tidak bisa dijalankan. Untuk web, PHP adalah bahasa *scripting* yang bisa dipakai untuk tujuan apapun (Winarno, Zaki, & Community, 2013).

## 2.22 CodeIgniter (CI)

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

### 1. View

Merupakan bagian yang menangani *presentation logic*. *View* berfungsi untuk menerima dan mempresentasikan data kepada *user*. Bagian ini tidak memiliki akses langsung terhadap bagian model.

### 2. Model

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

### 3. Controller

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

*Codeigniter* menawarkan kemudahan serta standarisasi dalam proses pengembangan aplikasi berbasis *website*. Dengan *codeigniter*, proses pengembangan menjadi lebih cepat dan terstandar. Standar ini sangat berguna dalam pengembangan yang dikerjakan oleh tim. *Codeigniter* juga menyediakan *library*, dan *helper* yang berguna dan mempermudah proses. Terdapat beberapa kelebihan penggunaan *codeigniter* dalam pengembangan suatu proyek, yaitu:

### 1. Menghemat Waktu

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

### 2. Gratis

*CodeIgniter* berlisensi di bawah Apache/BSD *opensource*.

### 3. Dukungan Komunitas

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

#### 4. Berukuran Kecil

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

### 2.23 MariaDB

Menurut Efendi (2019), MariaDB merupakan sebuah pengembangan mandiri dari MySQL yang mana aplikasi ini hadir semenjak MySQL telah berhasil diakuisi oleh Oracle pada sekitar bulan September 2010. MariaDB adalah sebuah implementasi atau penerapan dari RDMS atau sistem manajemen basis data rasional yang bisa anda gunakan secara gratis. Adapun fungsi dari MariaDB sebagai berikut (Zakaria, 2019):

1. MariaDB merupakan aplikasi yang dapat digunakan untuk memproses data SQL dalam waktu yang bersamaan.
2. MariaDB juga bisa digunakan sebagai aplikasi *database* untuk ranah yang lebih besar dimana jumlah rekamannya bisa mencapai kurang lebih 50 juta. Selain itu, aplikasi ini juga mampu menampung kurang lebih 60 ribu tabel dan juga 5 miliar baris. Pada setiap tabelnya mampu menampung hingga mencapai 32 indeks.
3. Menghubungkan *client* dengan menggunakan protokol TCP/IP, Named Pipes atau NT, dan juga UNIX Socket.
4. Mendeteksi pesan *error* yang diterima oleh klien dengan setidaknya menggunakan 20 bahasa yang berbeda-beda.
5. Mempermudah pengguna ketika ingin menggunakan aplikasi *database* yang bukan hanya mampu menangani skala besar, melainkan juga menyediakan banyak pilihan data.
6. MariaDB memiliki fungsi *multi-user*. Artinya, aplikasi ini bisa digunakan oleh banyak *user* dalam waktu yang bersamaan. Inilah mengapa *software* yang satu ini lebih banyak dipilih dibandingkan dengan yang lainnya.

Setelah mendapatkan informasi tentang pengertian beserta fungsi dari MariaDB. Zakaria (2019) juga menyebutkan kelebihan dari MariaDB sebagai pertimbangan lain untuk menggunakan MariaDB, antara lain:

1. Portabilitas

mampu dipasang dan berjalan pada semua sistem operasi yang ada, seperti yang menggunakan sistem operasi seperti *Linux, Windows, Mac Os X, FreeBSD*, dan lain sebagainya.

2. *Open Source*

Aplikasi yang bersifat *open source*. *User* dapat menggunakan MariaDB secara gratis, termasuk juga menggunakan seluruh fitur yang ada

3. *Multi-user*

Digunakan oleh banyak *user* atau pengguna dalam waktu yang bersamaan. Ketika digunakan secara bersama-sama, aplikasi ini tidak akan melambat, melainkan *software* ini tetap stabil.

4. Aplikasi *Database* yang Cepat

Berkaitan dengan *performance tuning*. Aplikasi ini mampu berjalan dalam waktu yang lumayan cepat ketika ingin menangani *query* yang sederhana.

5. Memiliki Banyak Pilihan Data

MariaDB juga merupakan aplikasi yang telah menyediakan banyak pilihan data yang bisa memberikan banyak alternatif untuk pengguna. Penjelasan singkat kegunaan masing-masing jenis data akan sangat bermanfaat dalam memilih suatu jenis data yang dipakai dalam merancang table. Beberapa jenis data yang tersedia pada MySQL dapat dilihat pada Tabel II. 12 di bawah ini:

Tabel II. 12 Jenis Tipe Data pada MariaDB

Jenis Data	Keterangan
<i>CHAR</i>	Sebuah <i>string</i> dengan panjang tetap. Sisa jumlah karakter yang belum terisi akan diisi dengan spasi, akan tetapi spasi ini dibuang jika data dipanggil. Jangkauan nilai M adalah 1-255 karakter.

Tabel II. 12 Jenis Tipe Data pada MariaDB (lanjutan)

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

(Sumber: Sutaji, 2012)

## 6. Keamanan

Dilengkapi dengan beberapa *firewall* misalnya level *subnetmask*, izin akses, nama *host*, dan berbagai perizinan lain yang bersifat detail yang harus dilalui semua itu untuk bisa masuk ke dalam sistem.

## 2.24 Pengujian Perangkat Lunak

Menurut Simarmata (2010), pengujian perangkat lunak merupakan aktivitas dan langkah-langkah yang dilakukan diantaranya adalah inisialisasi dan penentuan lingkup kerja pengujian, termasuk di dalamnya batasan-batasan serta lingkungan pengujian, perencanaan pengujian, eksekusi dan pengendalian, evaluasi serta peninjauan ulang hasil pengujian, dan penutup yaitu pelaporan dari hasil pengujian yang diikuti dengan perbaikan atau pembenahan bila diperlukan.

Dalam pengujian perangkat lunak diperlukan suatu metode pengujian, salah satu metode yang dapat digunakan ialah *Black Box Testing*. *Black box testing* merupakan salah satu jenis pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak, *tester* dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. *Black box testing* cenderung untuk menemukan hal-hal berikut (Mustaqbal, Firdaus, & Rahmadi, 2015):

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*).
5. Kesalahan inisialisasi dan terminasi

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Metodologi Penelitian

Menurut Woody dikutip Anshori & Iswati (2017), penelitian adalah sebuah metode untuk menemukan kebenaran yang juga merupakan sebuah pemikiran yang kritis. Penelitian meliputi pemberian definisi, dan redefinisi terhadap masalah, memformulasikan hipotesa atau hipotesis, membuat simpulan dan sekurang-kurangnya mengadakan pengujian yang hati-hati atas suatu simpulan untuk menentukan apakah akan cocok dengan hipotesis.

Metodologi penelitian merupakan sebuah cara untuk mengetahui hasil dari sebuah permasalahan yang spesifik, dimana permasalahan tersebut disebut juga dengan permasalahan penelitian. Dalam metodologi, peneliti menggunakan berbagai kriteria yang berbeda untuk memecahkan masalah penelitian yang ada. Sumber yang berbeda menyebutkan bahwa penggunaan berbagai jenis metode adalah untuk memecahkan masalah. Dengan menggunakan metodologi penelitian, peneliti akan dapat mengambil kesimpulan-kesimpulan sehingga dapat menemukan solusi dari permasalahan. Kesimpulan-kesimpulan tersebut dapat dipercaya, sebab menggunakan pengukuran-pengukuran secara *scientific* (Hidayat, 2016).

#### 3.2 Jenis dan Sumber Data

Data yang dibutuhkan dalam penelitian ini ada dua jenis, yaitu data primer dan data sekunder. Sumber dari data-data ini berasal dari tempat yang diamati pada praktik kerja lapangan di PT Suzuki Indomobil Motor *Plant* I Tambun:

##### 1. Data Primer

Data primer adalah data yang didapat secara langsung dari PT Suzuki Indomobil Motor *Plant* I Tambun, dimana pengumpulan data atau informasi dilakukan langsung dari objek yang diteliti. Data-data tersebut adalah data yang digunakan dalam proses pendistribusian unit diantaranya,

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

## 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 diantaranya 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:

#### 1. Studi lapangan

Studi lapangan adalah usaha melakukan pengamatan lapangan untuk mendapatkan data yang sesuai, yaitu dengan cara:

##### a. Pengamatan

Melakukan pengamatan secara langsung di PT Suzuki Indomobil Motor *Plant I* Tambun pada Departemen *Logistic 2W* atau yang sering dikenal dengan CBU untuk mengetahui jalannya sistem dan proses pendistribusian unit kendaraan yang berlangsung di perusahaan. Melalui teknik ini, data yang dibutuhkan diamati, dikumpulkan lalu diolah sebagai bahan di dalam penelitian.

##### b. Wawancara

Wawancara dilakukan secara langsung maupun tidak langsung kepada pembimbing atau karyawan PT Suzuki Indomobil Motor *Plant I* Tambun dengan memberikan pertanyaan mengenai proses dari pendistribusian unit yang berjalan dan untuk memperoleh data dan informasi mengenai topik lainnya yang ingin diketahui.

c. Analisis Dokumen

Analisis dokumen merupakan cara untuk menganalisis dan memahami data yang telah dikumpulkan yang berkaitan dengan proses pendistribusian unit pada PT Suzuki Indomobil Motor *Plant I Tambun*.

2. Studi Pustaka

Studi pustaka dilakukan dengan membaca buku dan literatur dalam lingkup perkuliahan maupun di luar lingkup perkuliahan yang berhubungan dengan permasalahan, sehingga dapat menunjang penulisan tugas akhir ini. Studi pustaka yang dilakukan adalah dengan menggunakan buku dan mencari data yang diperlukan melalui Internet.

### 3.4 Metode Pengembangan Sistem

Metode pengembangan sistem informasi pendistribusian unit 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*, *class diagram* dan *deployment diagram*. Pemodelan data menggunakan *Entity Relationship Diagram (ERD)*, *Conceptual Data Model (CDM)*, dan kamus data. Perancangan sistem menggunakan *Windows Navigation*

*Diagram* (WND), dan perancangan *interface*. 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. Selain itu, pengembang akan melakukan uji *testing* terhadap sistem yang sudah dibuat apakah sistem tersebut berjalan dengan baik dan sesuai dengan harapan atau tidak.

### 3.5 Fungsi *Euclidean Distance*

Dalam Jurnal yang ditulis oleh Setiawan et al. (2018) yang berjudul “Menghitung Rute Terpendek Menggunakan Algoritma A\* Dengan Fungsi *Euclidean Distance*” mengenai fungsi *euclidean distance* yang digunakan dalam menghitung jarak suatu wilayah yang ingin diketahui. *Euclidean distance* atau jarak *euclidean* adalah jarak garis lurus “biasa” antara dua titik dalam ruang *euclidean*. Literatur yang terdahulu menyebut metrik sebagai metrik *pythagoras*. Sistem pengukuran dengan jarak *euclidean* sering digunakan karena lebih mudah dimengerti, serta hasil dari pengukurannya memiliki tingkat akurat yang tinggi. Fungsi *euclidean* juga sering digunakan dalam menentukan jarak geografis suatu wilayah.

#### 3.5.1 Perhitungan Fungsi *Euclidean Distance*

*Euclidean distance* ialah fungsi yang digunakan dalam melakukan perhitungan jarak. *Euclidean* ini berkaitan dengan *Teorema Phytagoras* dan biasanya diterapkan pada 1, 2 dan 3 dimensi. Tapi juga sederhana jika diterapkan pada dimensi yang lebih tinggi. Adapun rumus dasar perhitungan pada fungsi *euclidean distance* ialah sebagai berikut:

1. Rumus Dasar Perhitungan *Euclidean Distance*.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (3.1)$$

dimana:

$x_1$  = koordinat x pada pusat penelitian/titik awal

$y_1$  = koordinat y pada pusat penelitian/titik awal

$x_2$  = koordinat x pada tujuan penelitian/titik tujuan

$y_2$  = koordinat y pada tujuan penelitian/titik tujuan

$d$  = jarak antara pusat penelitian x dan y

## 2. Rumus Perhitungan Jarak *Euclidean Distance*.

$$\text{Jarak} = \sqrt{(\text{Latitude1} - \text{Latitude2})^2 + (\text{Longitude1} - \text{Longitude2})^2}$$

dimana:

*Latitude1* = koordinat *Latitude1* pada titik awal

*Latitude2* = koordinat *Latitude2* pada titik tujuan

*Longitude1* = koordinat *Longitude1* pada titik awal

*Longitude2* = koordinat *Longitude1* pada titik tujuan

Dalam melakukan perhitungan jarak melakukan fungsi *euclidean distance* ada beberapa hal yang harus diperhatikan, antara lain:

1. Harus menetapkan titik awal dan titik tujuan.
2. Titik awal ialah yang menjadi ajuan jarak pada masing-masing titik tujuan dengan nilai jarak yang merupakan garis lurus.
3. Nilai pada titik awal dan akhir dalam perhitungan jarak diperoleh dari bobot *latitude* dan *longitude*.
4. Hasil perhitungan (jarak) masih dalam satuan *decimal degree* (sesuai dengan format longlat yang dipakai) sehingga untuk menyesuikannya perlu dikalikan dengan 111.319 km (1 derajat bumi = 111.319 km).

### 3.6 Algoritma *Dijkstra*

Dalam Jurnal yang ditulis oleh Kusuma et all. (2019) berjudul “Aplikasi Perhitungan Dan Visualisasi Jarak Terpendek Berdasarkan Data *Coordinate* Dengan Algoritma *Dijkstra* Dalam Kasus Pengantaran Barang Di Kawasan Jabodetabek” mengenai pemilihan jalur efisien dengan rute terpendek. Algoritma yang sangat terkenal untuk menyelesaikan persoalan ini adalah algoritma *Dijkstra*.

Algoritma *dijkstra* dianggap cocok karena mudah digunakan oleh *user* dalam penggunaannya hanya dengan menentukan titik awal dan titik tujuan.

*Dijkstra* adalah algoritma yang digunakan untuk mencari lintasan terpendek pada sebuah graf berarah. Contoh penerapan algoritma ini adalah lintasan terpendek yang menghubungkan antara dua kota berlainan tertentu. Kasus ini sering disebut *single-source single destination shortest path problems*. Cara kerja algoritma *dijkstra* memakai strategi *greedy*, dimana pada setiap langkah dipilih sisi dengan bobot terkecil yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih. Algoritma *dijkstra* membutuhkan parameter berupa tempat asal dan tempat tujuan.

### 3.7 Kerangka Penelitian

Dalam penelitian yang dibahas dalam Tugas Akhir ini, dilakukan langkah-langkah atau tahapan dalam penelitian seperti dijelaskan 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 *Logistic 2W* atau yang sering dikenal dengan sebutan CBU. Studi dapat dilakukan dengan melakukan pengamatan, *browsing Internet*, wawancara, dan studi pustaka.

#### 2. Identifikasi Masalah

Pada tahap ini dilakukan cara dalam mengidentifikasi suatu masalah yang terjadi pada Departemen *Logistic 2W* yang berfokus pada pendistribusian unit. Dalam mengidentifikasi suatu masalah dilakukan beberapa cara seperti:

- a. Mengamati Sistem Pendistribusian unit yang sedang berjalan.

- b. Mengidentifikasi terhadap hasil dalam suatu masalah dilakukan dengan menggunakan *Fishbone Diagram* yang bertujuan untuk dapat mencari jalan keluar dari masalah yang ada.
  - c. Analisis dokumen masuk dan keluar yang terkait pada proses pendistribusian unit.
3. Identifikasi Solusi
- Setelah mengidentifikasi suatu masalah, maka penulis melakukan pemecahan suatu masalah berdasarkan data-data yang telah didapat pada tahap sebelumnya. Terdapat beberapa pemecahan masalah yang penulis lakukan seperti:
- a. Mengidentifikasi batasan masalah yang ada pada Departemen *Logistic 2W*.
  - b. Menerapkan fungsi *Euclidean Distance* untuk memperoleh nilai (jarak) serta menerapkan Algoritma *Dijkstra* untuk menentukan rute terpendek pengantaran unit.
  - c. Merancang, membangun dan mengimplementasikan sistem informasi pendistribusian unit dengan menggunakan metode *waterfall*.
4. Penerapan Metode *Waterfall*
- a. Perencanaan (*Planning*)  
Penulis melakukan perencanaan untuk aplikasi yang akan dibuat dengan beberapa tahap yaitu melakukan studi pendahuluan untuk mengetahui gambaran sistem yang sedang berjalan pada perusahaan, identifikasi masalah perusahaan dan mengidentifikasi solusi untuk perbaikan dari sistem yang sudah ada.
  - b. Analisis (*Analysis*)  
Menganalisis kebutuhan sistem yang didapatkan dengan wawancara terhadap *user*. Hasil dari analisis kebutuhan sistem adalah daftar kebutuhan fungsional dan kebutuhan nonfungsional. Selain itu pada tahap ini dilakukan penyusunan proses bisnis usulan yang merupakan perbaikan dari sistem sebelumnya dengan menggunakan *Business Process Model and Notation (BPMN)*.

c. Desain (*Design*)

Membuat perancangan sistem yang diawali dengan tahap pemodelan sistem menggunakan *Unified Modeling Language* (UML) diantaranya *usecase diagram*, *activity diagram*, *sequence diagram*, *class diagram*, *deployment diagram*. Pemodelan data menggunakan *Entity Relationship Diagram* (ERD), *Conceptual Data Model* (CDM), dan kamus data. Perancangan sistem menggunakan *Windows Navigation Diagram* (WND), dan perancangan *interface*. Data-data yang didapat dari tahap analisis diterapkan dalam tahap desain.

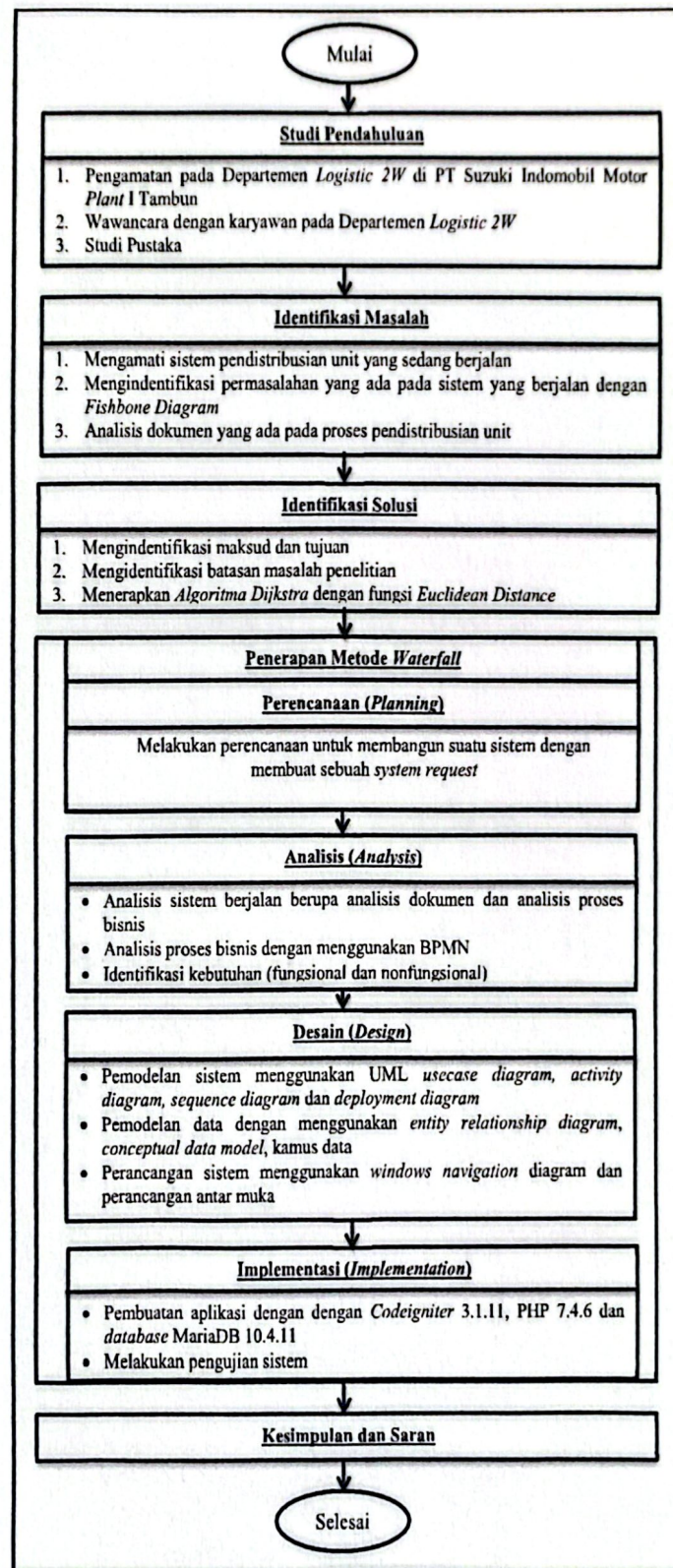
d. Implementasi (*Implementation*)

Pada tahap ini, pengembang akan membangun sebuah sistem dengan rancangan yang sudah ada diterjemahkan ke dalam sebuah kode program atau aktivitas *coding*. Bahasa pemrograman yang digunakan adalah PHP dengan menggunakan *framework CodeIgniter* dan *MariaDB* sebagai perangkat lunak sistem basis data (*Database Management System*, DBMS) yang digunakan. Setelah itu program akan dilakukan pengujian menggunakan *black box tesing*.

5. Kesimpulan dan Saran

Kesimpulan ialah hasil dari penelitian yang dilakukan. 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 (2020)

## BAB IV

### PENGUMPULAN DAN PENGOLAHAN DATA

#### 4.1 Sejarah Umum Perusahaan

Suzuki Indomobil Motor merupakan perusahaan yang bergerak dalam bidang industri otomotif, yang menghasilkan kendaraan antara lain sepeda motor dan mobil. PT Suzuki Indomobil Motor berdiri pada tahun 1970, dengan nama awal perusahaan PT Indohero Steel & Engineering Co. Tahun 1976 di bawah kepemimpinan Soebronto Laras, didapati manajemen baru yang merupakan awal pengembangan sepeda motor melalui PT Indohero Steel & Engineering Co, dan mobil melalui PT Suzuki Indomobil Utama.

Indomobil Suzuki International (ISI) dirubah menjadi PT Suzuki Indomobil Motor (SIM) yang merupakan sebuah perusahaan penanaman modal asing (PMA) yang berdiri dengan kekuatan 5 (lima) buah perusahaan. Perusahaan tersebut adalah PT Indohero Steel & Engineering Co, PT Indomobil Utama, PT Suzuki Indonesia Manufacturing, PT Suzuki Engine Industry, PT First Chemical Industry.

Lima perusahaan tersebut bergabung (*merger*) dengan persetujuan dari Presiden Republik Indonesia tertanggal 1 Januari 1990, dan diperingati sebagai berdirinya PT Suzuki Indomobil Motor, yang bergerak dalam bidang usaha industri komponen dan perakitan kendaraan bermotor merek Suzuki roda dua (sepeda motor) dan roda empat (mobil). Lokasi pabrik dan pusat perakitan kendaraan merek Suzuki tersebar di beberapa tempat antara lain:

1. *Plant Cakung (Perakitan Engine)*

*Plant Cakung* Lokasinya berada di Jalan Raya Penggilingan, Cakung, Jakarta Timur. Berdiri di areal tanah seluas 82.000 m<sup>2</sup> dan didukung oleh ± 634 karyawan. Di sini diproduksi berbagai macam komponen dan *part* sepeda motor dan mobil. Di sini pula dirakit berbagai macam peralatan transmisi dan kemudi baik sepeda motor maupun mobil.

2. *Plant Pulogadung (Service & Sales)*

*Plant* Pulogadung berada di Jalan Raya Bekasi Km. 19, Jakarta Timur, berdiri di areal tanah seluas 39.555 m<sup>2</sup>. Saat ini *Plant* Pulogadung hanya ada beberapa bagian saja, karena *assembling* untuk kendaraan roda empat sebagian besar telah pindah ke *Plant* Tambun II. Saat ini *Plant* Pulogadung dipergunakan sebagai kantor pusat. Pada kantor pusat ini merupakan tempat manajemen serta sistem pemasaran PT Suzuki Indomobil Motor.

3. *Plant* Tambun I (Perakitan Motor)

*Plant* Tambun I sebelumnya dikenal dengan nama PT Indohero Steel & Engineering Co. *Plant* Tambun I mampu menyerap tenaga kerja sebanyak ± 1128 orang. Berada di Jalan Raya Diponegoro Km. 38,2 Bekasi. Pada *Plant* Tambun I diproses, diproduksi, dan dirakit berbagai komponen kendaraan roda dua (sepeda motor) merek Suzuki, dan dari sini lahir berbagai sepeda motor Suzuki *type* mutakhir.

4. *Plant* Tambun II (Perakitan Mobil)

*Plant* Tambun II merupakan proyek khusus untuk kendaraan roda empat Suzuki. *Plant* Tambun II merupakan yang terbesar di Asia Tenggara untuk saat ini. *Plant* Tambun II berdiri di area tanah seluas 420.000 m<sup>2</sup>, dengan luas bangunan seluas 89.100 m<sup>2</sup>, dan mampu menyerap tenaga kerja sebanyak (±) 3.000 karyawan.

5. *Plant Spare Part*

Guna memberikan pelayanan purna jual bagi pemilik kendaraan merek Suzuki baik roda dua maupun roda empat, PT Indomobil Suzuki International memindahkan tempat penyediaan suku cadang dari *Plant* Sunter ke *spare part* yang berlokasi di Jl. P. Diponegoro Km. 38,2 Tambun, Bekasi. Di sana tersedia berbagai suku cadang asli untuk kendaraan bermotor merek Suzuki, serta menjual berbagai souvenir Suzuki.

2. GIIC - Cikarang (Perakitan Mobil)

Pabrik baru yang diresmikan ditahun 2013 digunakan untuk memproduksi kendaraan roda empat, dimana produksi yang meningkat dan semakin

berkembangnya produksi otomatis, sehingga Suzuki mengembangkan sayapnya untuk meningkatkan produktifitas.

#### 4.2 Logo Perusahaan

Logo merupakan suatu gambar atau sketsa dengan arti tertentu, dan mewakili suatu arti dari perusahaan, daerah, organisasi, produk, negara, lembaga, dan hal lainnya membutuhkan sesuatu yang singkat dan mudah diingat sebagai pengganti dari nama sebenarnya. Logo lebih lazim dikenal oleh penglihatan atau visual, seperti ciri khas berupa warna dan bentuk logo tersebut. Berikut Gambar IV.1 merupakan logo PT Suzuki Indomobil Motor:



Gambar IV.1. Logo PT Suzuki Indomobil Motor  
(Sumber: PT Suzuki Indomobil Motor, 2019)

#### 4.3 Profil Perusahaan

Profil perusahaan merupakan salah satu aspek penting dalam publikasi sebuah perusahaan. Fungsi sebuah profil perusahaan adalah sebagai media komunikasi dan penyampaian informasi tentang perusahaan terhadap pihak-pihak terkait, baik internal maupun eksternal perusahaan. Berikut profil perusahaan PT Suzuki Indomobil Motor:

Nama Perusahaan	: PT Suzuki Indomobil Motor
Produksi	: Motor dan mobil
Status	: Perseroan Terbatas
Alamat Kantor	: Jl. Raya Diponegoro Km. 38,2 Tambun, Jawa Barat, Indonesia
No Telefon	: +62 21 8801 251, +62 21 8801 235

Fax : +62 21 8807 401  
Website : www.suzuki.co.id  
Luas Pabrik : 142.000 m<sup>2</sup> (*Plant Tambun I*)  
Jumlah Tenaga Kerja : 6.434 orang (2019)

#### 4.4 Visi dan Misi Perusahaan

Visi berarti pandangan atau wawasan ke depan. PT Suzuki Indomobil Motor memiliki visi, yaitu “Menjadi perusahaan terkemuka di dalam Suzuki Global Operation yang dihargai dan dikagumi di Indonesia”.

1. Terkemuka menjadi yang terdepan diantara semua pabrik Suzuki (mobil dan motor) di seluruh dunia, berdasarkan kriteria tertentu yang telah ditetapkan manajemen.
2. Suzuki Global Operation  
Semua pabrik Suzuki (mobil dan motor) di seluruh dunia yang tergabung dalam jaringan Suzuki Corporation.
3. Dihargai  
Respek dari semua unsur *stakeholder* karena prestasi yang dicapai (*operational excellence*).
4. Dikagumi  
Komitmen perusahaan terhadap kualitas pelayanan pelanggan dan perhatian terhadap kualitas lingkungan hidup.

Misi merupakan suatu pernyataan tentang apa yang harus dikerjakan oleh perusahaan atau lembaga dalam usaha mewujudkan visi tersebut. Misi dari PT Suzuki Indomobil Motor adalah “Kami menginginkan pertumbuhan dan perkembangan perusahaan yang berimbang berdasarkan asas kerja keras, integritas dan kebersamaan untuk selalu mencapai hasil lebih baik dalam mendahului harapan pelanggan”.

1. Kami
  - a. Pabrikan, distributor, *branch*, sub distributor.
  - b. *Supplier*/pemasok/industri penunjang.

- c. *Stakeholder*/prinsipal pemilik teknologi/pemegang merek.
  - d. Karyawan.
2. Pertumbuhan: meningkat secara vertikal.
  3. Perkembangan: bertambah secara horizontal.
  4. Berimbang: menekankan pada efektifitas, efisiensi, dan profitabilitas yang terkontrol agar dapat memberikan nilai tambah bagi perusahaan maupun pelanggan.
  5. Kerja keras: kesediaan untuk berbuat lebih untuk kemajuan perusahaan, kemajuan kuat untuk mencapai hasil kerja yang lebih baik.
  6. Integritas: jujur, mendahulukan kepentingan perusahaan, memiliki komitmen terhadap janji, *policy* dan prosedur yang sudah diputuskan, disiplin.
  7. Kebersamaan: kemauan menjalin kerjasama yang konstruktif dalam upaya mencapai sasaran bersama.
  8. Mencapai hasil lebih baik: semangat untuk selalu belajar, dorongan untuk meningkatkan yang lebih baik.
  9. Mendahului harapan pelanggan: mengantisipasi kebutuhan pelanggan dan upaya-upaya untuk memenuhi harapan pelanggan.

#### 4.5 Jam Kerja

Waktu kerja di PT Suzuki Indomobil Motor yaitu 5 hari kerja dalam seminggu (Senin-Jumat) sedangkan jam kerja karyawan yang diberlakukan pada PT Suzuki Indomobil Motor dibagi dalam 2 *shift* yaitu *shift* 1 dan *shift* 3. Adapun pembagian waktu kerja pada *shift* 1 dan *shift* 3 dapat dilihat pada tabel di bawah ini:

Tabel IV.1 Jam Kerja *Shift* 1 Karyawan Operasional

<i>Shift</i> 1 (Waktu Normal)		
Senin – Kamis	Jumat	Keterangan
07.30 – 07.35	07.30 – 07.35	<i>Meeting</i> awal
07.35 – 10.00	07.35 – 10.00	Kerja
10.00 – 10.05	10.00 – 10.05	<i>Break</i>

Tabel IV.1 Jam Kerja *Shift* 1 Karyawan Operasional (lanjutan)

<i>Shift</i> 1 (Waktu Normal)		
10.05 – 12.00	10.05 – 11.25	Kerja
12.00 – 12.40	11.25 – 12.55	<i>Break</i>
12.40 – 15.00	12.55 – 15.00	Kerja
15.00 – 15.05	15.00 – 15.05	<i>Break</i>
15.05 – 16.25	15.05 – 16.25	Kerja
16.25 – 16.30	16.25 – 16.30	<i>Cleaning</i>

(Sumber : PT Suzuki Indomobil Motor, 2019)

Jam kerja karyawan *shift* I ialah dari pagi hingga sore dengan sedikit perbedaan pada waktu istirahat pada karyawan operasional dan karyawan *office*. Berikut ialah tabel detail jam kerja *shift* I untuk karyawan operasional:

Tabel IV.2 Jam Kerja *Shift* 1 Karyawan *Office* 1 & 2

<i>Shift</i> 1 (Waktu Normal)		
Senin – Kamis	Jumat	Keterangan
07.30 – 07.35	07.30 – 07.35	<i>Meeting</i> awal
07.35 – 10.00	07.35 – 10.00	Kerja
10.00 – 10.05	10.00 – 10.05	<i>Break</i>
10.05 – 12.20	10.05 – 11.25	Kerja
12.20 – 13.00	11.25 – 12.55	<i>Break</i>
13.00 – 15.00	12.55 – 15.00	Kerja
15.00 – 15.05	15.00 – 15.05	<i>Break</i>
15.05 – 16.25	15.05 – 16.25	Kerja
16.25 – 16.30	16.25 – 16.30	<i>Cleaning</i>

(Sumber : PT Suzuki Indomobil Motor, 2019)

Jam kerja *shift* 3 ialah dari malam hari hingga pagi hari dengan detail waktu dapat dilihat pada tabel di bawah ini:

Tabel IV.3 Jam Kerja *Shift* 3 Karyawan PT Suzuki Indomobil Motor



<b>Shift 3 (Waktu Normal)</b>	
<b>Senin-Jumat</b>	<b>Keterangan</b>
23.50-01.50	Kerja
01.50-02.20	<i>Break</i>
02.20-04.20	Kerja
04.20-04.25	<i>Break</i>
04.25-06.25	Kerja
06.25-06.30	<i>Break</i>
06.30-07.20	Kerja
07.20-07.30	<i>Cleaning</i>

(Sumber : PT Suzuki Indomobil Motor, 2019)






#### 4.6 Produk yang Dihasilkan

PT Suzuki Indomobil Motor menghasilkan produk berupa unit motor dan mobil tetapi pada *Plant* Tambun I hanya memproduksi unit motor, sedangkan untuk mobil pada *Plant* Tambun II. Unit yang dihasilkan di PT Suzuki Indomobil Motor *Plant* Tambun I ialah sebagai berikut:

Tabel IV.4 Produk PT Suzuki Indomobil Motor *Plant* Tambun I

<b>No</b>	<b>Nama Produk</b>	<b>Jenis Produk</b>	<b>Gambar Produk</b>
1.	Address FI	<i>Matic</i>	
2.	Address Playful	<i>Matic</i>	

Tabel IV.4 Produk PT Suzuki Indomobil Motor Plant Tambun I (lanjutan).

No	Nama Produk	Jenis Produk	Gambar Produk
3.	Nex II	<i>Matic</i>	
4.	GSX150 Bandit	<i>Touring</i>	
5.	GSX-R150	<i>Sport</i>	
6.	GSX-S150	<i>Touring</i>	
7.	SATRIA F150	<i>Underbone</i>	

(Sumber : PT Suzuki Indomobil Motor, 2019)

#### 4.7 Jaringan Pemasaran dan Distribusi

Jaringan pemasaran merupakan bentuk komunikasi atau kerjasama antara pihak-pihak terkait untuk mencapai target pasar. Untuk sistem pemasaran pada PT Suzuki Indomobil Motor yaitu perusahaan memasarkan produknya hampir seluruh wilayah Indonesia, untuk *customer* dalam negeri dengan kode wilayah domestik P12. Namun produk tidak langsung sampai ke tangan *customer*, melainkan melalui distributor sebagai tangan distribusi. Berikut Tabel IV.5 ialah Daftar Jaringan Pemasaran Domestik PT Suzuki Indomobil Motor:

Tabel IV.5 Daftar Jaringan Pemasaran Domestik

No	Domestik	Nama Distributor
1.	Sumatera Utara	Sunindo Karya Motor Gemilang
2.	Jawa Tengah	Indosun Motor Gemilang
3.	Jawa Barat	Sanggar Mas Jawa
4.	Jawa Timur	Sejahtera Motor Gemilang
5.	Jabodetabek	Sejahtera Motor Gemilang Jakarta
6.	Bali	Cahaya Surya Bali Indah
7.	Maluku	Citra Perdana Motor
8.	Sulawesi	Sinar Galesang Mandiri
9.	Kalimantan Timur	Samekarindo Indah
10.	Papua	Papua Model Modern

(Sumber : PT Suzuki Indomobil Motor, 2019)

PT Suzuki Indomobil Motor memiliki *customer* tidak hanya di dalam negeri saja tetapi sampai mancanegara. Berikut Tabel IV.6 ialah Daftar Jaringan Pemasaran Mancanegara PT Suzuki Indomobil Motor:

Tabel IV.6 Daftar Jaringan Pemasaran Mancanegara

No	Mancanegara	Kode Negara
1.	Jepang	N00
2.	Inggris	P02
3.	Australia	P24
4.	Europe	P19
5.	Filipina	P31

Tabel IV.6 Daftar Jaringan Pemasaran Mancanegara (lanjutan)

No	Mancanegara	Kode Negara
6.	Thailand	P14
7.	Vietnam	P73
8.	Mexico	P09
9.	Great Britain	P21
10.	Jerman	P21
11.	Swedia	P21
12.	Saudi Arabia	P43

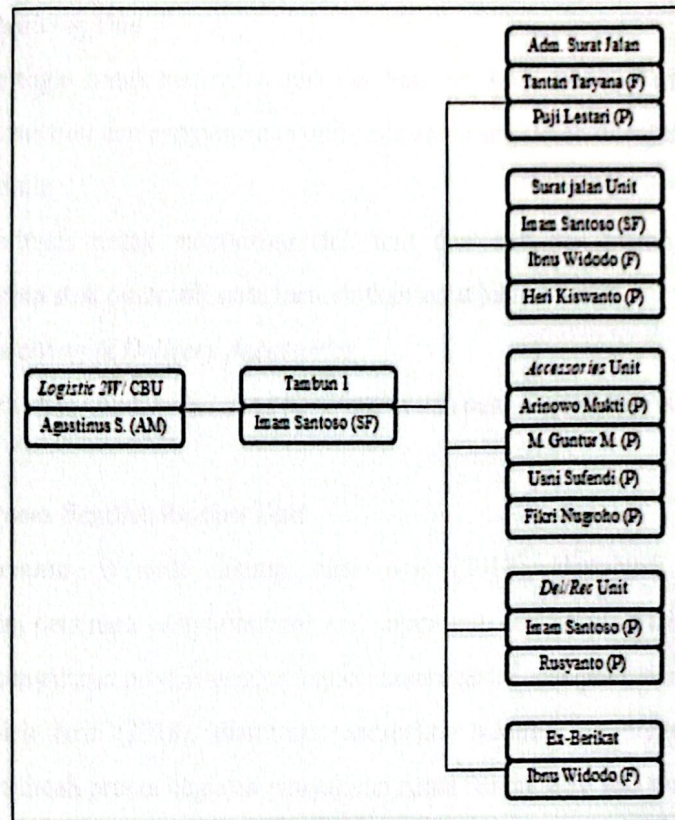
(Sumber : PT Suzuki Indomobil Motor, 2019)

#### 4.8 Departemen *Logistic 2W*

Departemen *Logistic 2W* atau yang sering dikenal CBU (*Completely Build Up*) pada PT Suzuki Indomobil Motor *Plant I* Tambun adalah salah satu bagian utama dari perusahaan yang bertugas untuk mengelola seluruh kegiatan pendistribusian pada distributor, penyimpanan unit pada gudang, kelengkapan unit seperti *accessories* dan lain-lain. Pada PT Suzuki Indomobil Motor *Plant I* Tambun, CBU juga mengatur informasi dari *Plant Cakung* terkait DO (*Delivery Order*) yang masuk dan kemudian diteruskan menjadi SJ (Surat Jalan).

##### 4.8.1 Struktur Organisasi Departemen *Logistic 2W*

Pada Departemen *Logistic 2W* juga terdapat struktur organisasi yang menjelaskan kembali posisi dan tanggung jawab masing-masing karyawan. Jumlah karyawan pada Departemen *Logistic 2W* yaitu 25 (dua puluh lima) orang terdiri dari 1 (satu) orang AM (*Assistant Manager*), 1 (satu) orang senior *foreman*, 2 (dua) orang *foreman* dan 21 (dua puluh satu) pelaksana. Berikut merupakan struktur organisasi pada Departemen *Logistic 2W*, untuk lebih jelasnya bisa dilihat pada Gambar IV.2:



Gambar IV.2 Struktur Organisasi *Logistic 2W*  
(Sumber: PT Suzuki Indomobil Motor, 2019)

#### 4.8.2 Job Description Departemen *Logistic 2W*

*Job description* membuat pembagian kerja dan tanggung jawab masing-masing bagian semakin jelas. Deskripsi mengenai tugas dan tanggung jawab setiap bagian pada CBU sebagai berikut:

##### 1. *Ploting Unit*

Bertugas untuk merealisasikan pengiriman unit sepeda motor di CBU agar sesuai dengan DO maupun alokasi bulanan dan melakukan pengontrolan stok unit di gudang utama maupun di Gudang Ex-Berikat.

##### 2. *Delivery Unit*

Bertugas untuk berkoordinasi dengan ekspedisi dan kuli angkut terkait pengiriman unit pada distributor.

### 3. *Receiving Unit*

Bertugas untuk menerima unit OK hasil produksi dari Departemen *Final Inspection* dan penyimpanan unit pada area yang sudah ditentukan.

### 4. Admin

Bertugas untuk *monitoring* stok unit domestik dan membuat laporan harian stok domestik serta menerbitkan surat jalan.

### 5. *Receiving & Delivery Accessories*

Bertugas untuk *monitoring* penerimaan dan pengeluaran *accessories* unit.

## 4.9 Proses Pendistribusian Unit

Menurut Winardi dikutip oleh Arif (2018), distribusi merupakan sekumpulan perantara yang terhubung erat antara satu dengan yang lainnya dalam kegiatan penyaluran produk-produk kepada konsumen. Sedangkan menurut Kotler dikutip oleh Arif (2018), distribusi merupakan sekumpulan organisasi yang membuat sebuah proses kegiatan penyaluran suatu barang atau jasa untuk dipakai atau dikonsumsi oleh para konsumen.

Masalah pendistribusian unit yang terjadi pada Departemen *Logistic 2W* muncul saat pengiriman unit pada distributor, dimana pada proses pengiriman sering terjadi keterlambatan dalam pengiriman maupun waktu kembali truk ke perusahaan. Keterlambatan tentu akan berakibat pada penumpukan unit di gudang dan dapat menimbulkan kerugian bagi perusahaan, sedangkan dari sisi pelanggan akan timbul rasa ketidakpuasan. Salah satu usaha yang perlu dilakukan untuk mengatasi kendala tersebut adalah dengan menerapkan metode penentuan rute terpendek untuk setiap pengiriman, sehingga pekerjaan dilakukan seoptimal mungkin. Selain itu data antar bagian yang belum terintegrasi dan pembuatan laporan dengan proses konvensional membuat pekerjaan menjadi lambat dan belum efektif. Pendistribusian yang baik tentunya akan menghasilkan dampak yang positif, yaitu rendahnya biaya, waktu pengantaran dan meminimalisasi dampak lainnya.

#### 4.10 Analisis Dokumen Pendistribusian Unit

Analisis dokumen ialah kegiatan menganalisis atau menguraikan dokumen yang digunakan pada proses yang berjalan serta mendapatkan penjelasan mengenai dokumen-dokumen yang digunakan. Dalam pendistribusian unit pada distributor pasti membutuhkan dokumen sebagai informasi mengenai unit, oleh karena itu dokumen sangat penting agar proses pendistribusian unit dapat berjalan dengan baik.

Dokumen yang terlibat akan digunakan sebagai pendukung dalam perbaikan sistem pendistribusian unit pada PT Suzuki Indomobil Motor *Plant I Tambun*, sehingga perusahaan dapat menjalankan proses dengan baik dan lancar. Dokumen yang terkait dalam proses pendistribusian unit adalah sebagai berikut:

##### 1. Dokumen Register *Outstanding DO (Delivery Order)*

Dokumen DO (*delivery order*) adalah dokumen yang menyatakan kepemilikan atas barang. DO dibuat oleh Bagian *Marketing* kemudian dikirimkan pada CBU dan nantinya akan digunakan sebagai syarat pembuatan surat jalan unit. Berikut Dokumen DO dapat dilihat pada Gambar IV.3.

PT. SUZUKI INDOMOBIL MOTOR		REGISTRIER OUTSTANDING DO		TGL : 25/09/2019				
Divisi Roda 2		Periode : 01/09/2019 s/d 25/09/2019		JAM : 08:39:31				
Customer Name ( Customer Code )				HAL : 1				
DO No	DO Date	Model & Accs	Color	Qty DO	Hold	Cancel Qty	SPQDate DO	Remarks
ANAYA SURYA BALI INDAH . PT (6101201)								
D1905341	03-SEP-19	UK110NE2	NI - YVU	5	0	0	5	0
				Subttl	5	0	0	5
D1905483	04-SEP-19	FU150MF	NI - YVU	5	0	0	5	0
				MH - GTA	6	0	0	6
				Subttl	11	0	0	11
D1905484	04-SEP-19	FU150MFX	BI - YSF	5	0	0	5	0
				Subttl	5	0	0	5
D1905485	04-SEP-19	FU150MFA	NI - 4GX	12	0	0	12	0
				Subttl	12	0	0	12
D1905486	04-SEP-19	FU150MPE	NI - 291	2	0	0	2	0
				Subttl	2	0	0	2
D1905487	04-SEP-19	UK110NE2P	NI - YUA	1	0	0	0	1
				NI - 291	1	0	0	1
				Subttl	2	0	0	2
D1905488	04-SEP-19	UK110NE2	NI - YVU	1	0	0	1	0
				Subttl	1	0	0	1
D1905493	04-SEP-19	UK110NE2P	NI - 291	1	0	0	1	0
				Subttl	1	0	0	1
D1905710	10-SEP-19	FU150MFX	BI - YSF	1	0	0	1	0
				Subttl	1	0	0	1
D1905787	16-SEP-19	FU150MF	NI - YVU	1	0	0	1	0
				Subttl	1	0	0	1
<b>Total:</b>				<b>41</b>	<b>0</b>	<b>0</b>	<b>40</b>	<b>1</b>
<b>Grand Total:</b>				<b>41</b>	<b>0</b>	<b>0</b>	<b>40</b>	<b>1</b>

Gambar IV.3 Dokumen Register *Outstanding DO*  
(Sumber: PT Suzuki Indomobil Motor, 2019)

Berikut adalah Tabel IV.7 keterangan pada *field-field* pada form DO (*delivery order*):

Tabel IV.7 Deskripsi *Field-field Form DO (Delivery Order)*

No.	Field	Deskripsi
1.	Tanggal	Keterangan tanggal pencetakan DO ( <i>delivery order</i> ) oleh CBU.
2.	Jam	Keterangan waktu pencetakan DO ( <i>delivery order</i> ) oleh CBU.
3.	Hal	Keterangan jumlah lembar halaman DO ( <i>delivery order</i> ).
4.	Periode	Keterangan periode DO ( <i>delivery order</i> ).
5.	Customer Name ( <i>Customer Code</i> )	Berisikan nama dari distributor yang memesan unit.
6.	DO No	Berisikan nomor dari dokumen DO ( <i>delivery order</i> ).
7.	DO Date	Keterangan tanggal pembuatan DO ( <i>delivery order</i> ) yang dibuatkan oleh Bagian <i>Marketing</i> .
8.	Model & Accs Color	Keterangan unit dan warna pada accessories unit yang dipesan.
9.	Qty DO	Berisikan banyaknya pesanan unit yang dipesan.
10.	Hold	Berisikan banyaknya pesanan unit yang tertahan.
11.	Cancel Qty	Berisikan banyaknya pesanan unit yang dibatalkan.
12.	SJOuts	Berisikan banyaknya pesanan unit yang sudah diterbitkan menjadi surat jalan unit.

Tabel IV.7 Deskripsi *Field-field Form DO (Delivery Order)* (lanjutan).

No.	Field	Deskripsi
13.	DO	Berisikan banyaknya unit yang dipesan namun belum diproses.
14.	Remarks	Berisikan komentar atau keterangan terkait pesanan yang dibuat.
15.	SubTtl	Berisikan jumlah dari masing-masing <i>Qty DO, Hold, Cancel Qty, SJOuts, DO</i> per nomor DO.
16.	Total	Berisikan total <i>Qty DO, Hold, Cancel Qty, SJOuts, DO</i> dari akumulasi semua nomor DO
17.	Grand Total	Berisikan total akumulasi akhir dari setiap <i>field</i> pada dokumen.

(Sumber : PT Suzuki Indomobil Motor, 2019)

## 2. Dokumen Surat Jalan

Dokumen surat jalan adalah surat perintah muat unit pada distributor sesuai dengan DO yang diterima dari Bagian *Marketing*. Dokumen surat jalan dibuat dan dicetak oleh CBU, nantinya dokumen ini akan dikirimkan ke distributor beserta dengan unit. Berikut adalah Tabel IV.8 keterangan *field-field* pada dokumen surat jalan:

Tabel IV.8 Deskripsi *Field-field Surat Jalan*

No.	Field	Deskripsi
1.	Hal	Keterangan jumlah lembar halaman surat jalan.
2.	Nomor	Berisikan nomor dari surat jalan yang akan dikirimkan pada distributor.
3.	Tanggal	Keterangan tanggal pencetakan surat jalan.
4.	Gudang	Keterangan dari lokasi gudang perusahaan yang mengirimkan unit.

Tabel IV.8 Deskripsi *Field-field* Surat Jalan (lanjutan).

No.	<i>Field</i>	Deskripsi
5.	No. DO	Berisikan nomor dari dokumen DO ( <i>delivery order</i> ).
6.	Tgl. DO	Keterangan tanggal pembuatan DO ( <i>delivery order</i> ).
7.	Kepada Yth:	Berisikan alamat distributor yang merupakan tujuan dari pengiriman unit.
8.	No.	Berisikan urutan nomor surat jalan per masing-masing unit.
9.	<i>Type</i>	Keterangan dari <i>type</i> unit yang akan dikirimkan.
10.	Warna	Keterangan dari warna unit yang akan dikirimkan.
11.	Nomor Rangka	Berisikan nomor dari rangka unit yang masing-masing unit memiliki nomor sendiri sebagai informasi produksi terkait unit.
12.	Nomor Mesin	Berisikan nomor dari mesin unit yang masing-masing unit memiliki nomor sendiri sebagai informasi produksi.
13.	Buku Servis	Berisikan nomor buku servis yang merupakan buku catatan servis kendaraan.
14.	Keterangan	Berisikan keterangan tambahan dari <i>field</i> yang ada dan merupakan informasi yang cukup penting terkait unit.
15.	Diterima oleh	Berisikan tanda tangan dari distributor yang akan menerima unit dan kelengkapan lainnya.
16.	Diperiksa oleh	Berisikan tanda tangan dari pemeriksa unit yang akan dikirimkan yaitu pihak keamanan.

Tabel IV.8 Deskripsi *Field-field* Surat Jalan (lanjutan).

No.	Field	Deskripsi
17.	Disetujui oleh	Berisikan tanda tangan dari Kepala Divisi <i>Logistic 2W</i> atau yang sering disebut kepala gudang.
18.	Disiapkan oleh	Berisikan tanda tangan dari petugas gudang yang menyiapkan unit secara keseluruhan.
19.	Dikirimkan dengan	Keterangan tambahan dari perusahaan yang menyatakan pengiriman menggunakan angkutan.

(Sumber: PT Suzuki Indomobil Motor, 2019)

Berikut adalah Gambar IV.4 Dokumen Surat Jalan:

**SUZUKI**  
PT. SUZUKI INDOMOBIL SALES  
JL. RAYA BEKASI KM. 19 RT 009 RW 01,  
RAWA TERATE, CAKUNG, JAKARTA TIMUR 13820  
TELP. 021-29554800 (KUNYING)

**BURAT JALAN**

Kepada Yth. :  
RIALJAYA CEMERLANG PT  
JL. HANUKA TUANKU TAMBUSAI NO 18 J L L  
PEKANBARU

No. DO :  
Tgl. DO :

Bersama ini kami serahkan sejumlah  
unit kendaraan dengan spesifikasi sebagai berikut:

No	BARANG		Nomor Rangka	Nomor Mesin	Buku Service	KETERANGAN
	Type	Warna				
1	PULSOMP	H - TAITAN BIA	209098	206419	X1004139	Setiap unit kendaraan dilengkapi dengan:
2	PULSOMP	H - TAITAN BIA	206407	206440	KP008273	1. BUKU PELIHAN BATAS IN-
3	PULSOMP	H - TAITAN BIA	206137	206420	KP008273	2. INELM PRT
4	PULSOMP	H - TAITAN BIA	206498	206442	KP008273	3. INTRON KEY REAR VIEW L
5	PULSOMP	H - TAITAN BIA	206402	206442	KP008273	4. INTRON KEY REAR VIEW R
6	PULSOMP	H - TAITAN BIA	206545	206388	KP008268	5. LTOOL KEY
7	PULSOMP	H - TAITAN BIA	209099	206847	KP008773	6. LTOOL KEY
8	PULSOMP	H - TAITAN BIA	209042	209098	KD004340	7. LTOOL KEY
9	PULSOMP	H - TAITAN BIA	206404	206111	KD004123	8. LTOOL KEY
10	PULSOMP	H - TAITAN BIA	206553	206119	KP008270	9. LTOOL KEY
11	PULSOMP	H - TAITAN BIA	209090	206781	KP008779	10. LTOOL KEY
12	PULSOMP	H - TAITAN BIA	206217	209107	004331	11. LTOOL KEY
13	PULSOMP	H - TAITAN BIA	206411	206112	004311	12. LTOOL KEY
14	PULSOMP	H - TAITAN BIA	206573	206439	KP008277	13. LTOOL KEY
15	PULSOMP	H - TAITAN BIA	204406	206806	KP008799	14. LTOOL KEY
16	PULSOMP	H - TAITAN BIA	206448	204398	KP008272	15. LTOOL KEY
17	PULSOMP	H - TAITAN BIA	209040	204828	KP008794	16. LTOOL KEY
18	PULSOMP	H - TAITAN BIA	209075	209094	KD004343	17. LTOOL KEY
19	PULSOMP	H - TAITAN BIA	209093	209131	KD004316	18. LTOOL KEY
20	PULSOMP	H - TAITAN BIA	209079	209104	KD004334	19. LTOOL KEY
						20. LTOOL KEY

Diterima oleh Dealer/Ekspedisi

Diperiksa oleh Keamanan

Disetujui oleh Kepala Gudang

Disiapkan oleh Petugas Gudang

Dikirim dengan :

DISTRIBUSI  
CDU-R2

Gambar IV.4 Dokumen Surat Jalan Unit  
(Sumber: PT Suzuki Indomobil Motor, 2019)

### 3. Dokumen Laporan Harian Stok (LHS) *Warehouse Tambun*

Dokumen Laporan Harian Stok (LHS) *Warehouse Tambun* merupakan dokumen yang digunakan untuk mencatat setiap unit yang masuk dalam gudang untuk disimpan serta unit keluar dari gudang. Dokumen LHS *Warehouse Tambun* dibuat sebagai informasi ketersediaan ruang pada gudang penyimpanan unit, dokumen LHS *Warehouse Tambun* juga nantinya akan dicetak sebagai arsip dari Departemen *Logistic 2W*. Adapun Dokumen Laporan Harian Stok *Warehouse Tambun* dapat dilihat pada Gambar IV.6.

LAPORAN HARIAN STOK NORMAL DOMESTIC																												
PER 24 Sep 2019																												
WAREHOUSE: TAMBUN																												
TYPE	COLOR	STOCK AWAL						PENGIRIMAN						PENGIRIMAN						ADJUST						STOCK AKHIR		
		CBU	REPAIR	CBU	REPAIR	IGTS IN	RETURN	REPAIR	CHANGE	MOD	CBU	IGTS OUT	REPAIR	CHANGE	MOD	CBU	REPAIR	CHANGE	MOD	CBU	REPAIR	CHANGE	MOD	CBU	REPAIR	IGTS	REPAIR	
		QTY	QTY	QTY	QTY	QTY	QTY	QTY	QTY	QTY	QTY	QTY	QTY	QTY	QTY	QTY	QTY	QTY	QTY	QTY	QTY	QTY	QTY	QTY	QTY	QTY	QTY	
ME	YGE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	SUB TOTAL	85	0	14	99	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
LUTKMER	JK	17	0	0	122	0	0	0	0	0	0	0	4	2	116	0	0	0	0	0	0	0	0	0	113	0	0	0
	SH	28	0	1	125	0	0	0	0	0	0	0	0	0	1	134	0	0	0	0	0	0	0	0	130	0	0	0
	SUB TOTAL	45	0	1	247	0	0	0	0	0	0	0	4	2	250	0	0	0	0	0	0	0	0	0	251	0	0	0
LUTKMER	ACC	JK	0	0	0	0	0	0	0	0	0	0	0	0	113	0	0	0	109	0	0	0	0	0	0	0	0	0
	SH	0	0	0	0	0	0	0	0	0	0	0	0	0	130	0	0	0	130	0	0	0	0	0	0	0	0	0
	SUB TOTAL	0	0	0	0	0	0	0	0	0	0	0	0	0	251	0	0	0	247	0	0	0	0	0	0	0	0	0
LUTKME	JK	291	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0
	SH	59	0	0	60	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	SUB TOTAL	350	0	0	60	0	0	0	0	0	0	0	0	0	1	0	0	0	15	0	0	0	0	0	0	0	0	0
ME	YUL	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	SUB TOTAL	15	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	SUB TOTAL	75	0	2	79	0	0	0	0	0	0	0	0	0	1	0	0	0	75	0	0	0	0	0	0	0	0	0
	TOTAL	3116	0	31	2887	0	0	0	0	0	0	0	4	1450	134	2430	0	0	530	0	0	0	0	0	0	1450	0	0

Gambar IV.5 Laporan Harian Stok *Warehouse Tambun*  
(Sumber: PT Suzuki Indomobil Motor, 2019)

Berikut Tabel IV.9 keterangan *field-field* pada Dokumen Laporan Harian Stok *Warehouse Tambun*:

Tabel IV.9 Dekripsi *Field-field* Laporan Harian Stok *Warehouse Tambun*

No.	Field	Deskripsi
1.	Tanggal	Keterangan tanggal laporan harian stok dibuat.
2.	<i>Warehouse</i>	Keterangan lokasi gudang yang menjadi tempat penyimpanan unit-unit.
3.	<i>Type</i>	Berisikan kode unit yang menyatakan <i>type</i> dari unit.

Tabel IV.9 Dekripsi *Field-field* Laporan Harian Stok *Warehouse Tambun*  
(lanjutan).

No.	<i>Field</i>	Deskripsi
4.	<i>Color</i>	Keterangan dari warna unit-unit yang disimpan pada gudang.
5.	Stok Awal	Berisikan jumlah dari unit yang memang sudah tersimpan dalam gudang.
6.	Penerimaan <i>Qty</i>	Berisikan jumlah dari unit yang diterima CBU untuk disimpan dalam gudang.
7.	Penerimaan <i>Cum</i>	Berisikan jumlah yang diperoleh dari penambahan stok awal dan penerimaan <i>qty</i> .
8.	Penerimaan <i>Return Qty</i>	Berisikan jumlah unit-unit yang mengalami pengembalian dari toko ke gudang.
9.	Penerimaan <i>Return Cum</i>	Berisikan total jumlah unit-unit dari <i>return qty</i> .
10.	Penerimaan <i>Repair Qty</i>	Berisikan jumlah unit yang mengalami perbaikan.
11.	Penerimaan <i>Repair Cum</i>	Berisikan total dari jumlah unit yang mengalami perbaikan.
12.	Penerimaan <i>IGTS IN Qty</i>	Berisikan jumlah unit yang masuk dari transfer gudang lainnya, baik itu dari gudang perusahaan internal maupun dari luar perusahaan.
13.	Penerimaan <i>Change Model Qty</i>	Berisikan jumlah dari unit yang mengalami perubahan yang disebabkan beberapa faktor.
14.	Penerimaan <i>IGTS IN Cum</i>	Berisikan total dari jumlah unit yang masuk dari transfer gudang lainnya, baik itu dari gudang perusahaan internal maupun dari luar perusahaan.

Tabel IV.9 Dekripsi *Field-field* Laporan Harian Stok *Warehouse* Tambun  
(lanjutan).

No.	<i>Field</i>	Deskripsi
15.	Penerimaan <i>Change Model Cum</i>	Berisikan total dari jumlah unit yang mengalami perubahan.
16.	Pengiriman <i>Qty</i>	Berisikan jumlah unit yang dikirimkan ke distributor.
17.	Pengiriman <i>Cum</i>	Berisikan total dari jumlah unit yang dikirimkan ke distributor.
18.	Pengiriman <i>IGTS Out Qty</i>	Berisikan jumlah unit yang keluar dari gudang untuk ditransfer ke gudang lain, baik gudang internal maupun gudang lokasi eksternal.
19.	Pengiriman <i>IGTS Out Cum</i>	Berisikan total jumlah unit yang keluar dari gudang untuk ditransfer ke gudang lain.
20.	Pengiriman <i>Repair Qty</i>	Berisikan jumlah unit yang mengalami perbaikan.
21.	Pengiriman <i>Repair Cum</i>	Berisikan total dari jumlah unit yang mengalami perbaikan.
22.	Pengiriman <i>Change Model Qty</i>	Berisikan jumlah dari unit yang mengalami perubahan yang disebabkan beberapa faktor.
23.	Pengiriman <i>Change Model Cum</i>	Berisikan total dari jumlah unit yang mengalami perubahan.
24.	<i>Adjustment Qty</i>	Berisikan jumlah unit pada data yang mengalami selisih kuantitas dibandingkan dengan stok secara fisik.
25.	<i>Adjustment Repair Qty</i>	Berisikan jumlah unit yang selisih dengan stok fisik dan mengalami perbaikan.

Tabel IV.9 Dekripsi *Field-field* Laporan Harian Stok *Warehouse* Tambun  
(lanjutan).

No.	Field	Deskripsi
26.	<i>Adjustment Repair Cum</i>	Berisikan total jumlah unit yang selisih dengan stok fisik dan mengalami perbaikan.
27.	Stok Akhir <i>Qty</i>	Berisikan jumlah unit yang dihitung secara fisik di gudang.
28.	Stok Akhir <i>Repair</i>	Berisikan jumlah unit yang dihitung secara fisik di gudang dan harus mengalami perubahan.
29.	Total	Berisikan total jumlah unit pada gudang.

(Sumber: PT Suzuki Indomobil Motor, 2019)

#### 4. Dokumen Lembar Harian Stok (LHS) *Warehouse* Ex-Berikat

Dokumen LHS *Warehouse* Ex-Berikat ialah dokumen yang hampir sama dengan LHS Tambun hanya lokasi gudang saja yang membedakan. LHS *Warehouse* Ex-Berikat juga tetap mencatat unit yang keluar masuk dalam gudang tersebut. Adapun Lembar Harian Stok *Warehouse* dapat dilihat pada Gambar IV.6.

LAPORAN HARIAN STOK NORMAL DOMESTIC		PER 24 Sep 2019		WAREHOUSE: EX BERIKAT															
TYPE	COLOR	STOCK AWN		PENERIMAAN				PENGIRIAN				ADJUST				STOCK AKHIR			
		CBU	REPA	CBU	IGTS IN	RETURN	REPAIR	CHANGE M	CBU	IGTS OUT	REPAIR	CHANGE M	CBU	REPAIR	CBU	REPAIR			
		QTY	CUM	QTY	CUM	QTY	CUM	QTY	CUM	QTY	CUM	QTY	CUM	QTY	CUM	QTY	CUM		
FV112L2	SH	YML	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0
	WE	YTB	70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	70	0
	HI	YVU	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
			89	0	0	0	0	0	0	0	0	0	0	0	0	0	0	89	0
			SUB TOTAL																
FV119L2	WE	YTB	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37	0
	HI	YVU	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	27	0
			65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	64	0
			SUB TOTAL			46	0	0	0	0	0	0	0	0	0	0	0	0	64
GSF-150MF	KP	ATZ	272	0	0	0	0	0	0	0	0	0	0	0	0	0	0	272	0
	MM	GTA	175	0	0	0	0	0	0	0	0	0	0	0	0	0	0	174	0
	SI	YUC	395	0	0	0	0	0	0	0	0	0	0	0	0	0	0	395	0
	HI	YVU	270	0	0	0	0	0	0	0	0	0	0	0	0	0	0	270	0
			1112	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1111	0
			SUB TOTAL																
GSX-S150MF	HI	291	47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	47	0
			SUB TOTAL																47
JK112NE1	MM	GTA	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	PU	VUH	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
			2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0
			SUB TOTAL																2
JK112NE2	MM	GTA	123	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0
	PU	VUH	71	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
	HI	YVU	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
			194	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0
			SUB TOTAL																150

Gambar IV.6 Laporan Harian Stok *Warehouse* Ex-Berikat

(Sumber: PT Suzuki Indomobil Motor, 2019)

Berikut Tabel IV.10 keterangan *field-field* pada Dokumen Laporan Harian Stok Warehouse Ex-Berikat:

Tabel IV.10 Dekripsi *Field-field* Laporan Harian Stok Warehouse Ex-Berikat

No.	<i>Field</i>	Deskripsi
1.	Tanggal	Keterangan tanggal laporan harian stok dibuat.
2.	<i>Warehouse</i>	Keterangan lokasi gudang yang menjadi tempat penyimpanan unit-unit.
3.	<i>Type</i>	Berisikan kode unit yang menyatakan <i>type</i> dari unit.
4.	<i>Color</i>	Keterangan dari warna unit-unit yang disimpan pada gudang.
5.	Stok Awal	Berisikan jumlah dari unit yang memang sudah tersimpan dalam gudang.
6.	Penerimaan <i>Qty</i>	Berisikan jumlah dari unit yang diterima CBU untuk disimpan dalam gudang.
7.	Penerimaan <i>Cum</i>	Berisikan jumlah yang diperoleh dari penambahan stok awal dan penerimaan <i>qty</i> .
8.	Penerimaan <i>Return Qty</i>	Berisikan jumlah unit-unit yang mengalami pengembalian dari toko ke gudang.
9.	Penerimaan <i>Return Cum</i>	Berisikan total jumlah unit-unit dari <i>return qty</i> .
10.	Penerimaan <i>Repair Qty</i>	Berisikan jumlah unit yang mengalami perbaikan.
11.	Penerimaan <i>Repair Cum</i>	Berisikan total dari jumlah unit yang mengalami perbaikan.
12.	Penerimaan <i>IGTS IN Qty</i>	Berisikan jumlah unit yang masuk dari transfer gudang lainnya, baik itu dari gudang lokasi internal maupun dari lokasi eksternal gudang.

Tabel IV.10 Dekripsi *Field-field* Laporan Harian Stok *Warehouse Ex-Berikat*  
(lanjutan).

No.	<i>Field</i>	Deskripsi
13.	Penerimaan <i>IGTS IN Cum</i>	Berisikan total dari jumlah unit yang masuk dari transfer gudang lainnya, baik itu dari lokasi internal maupun dari lokasi eksternal gudang.
14.	Penerimaan <i>Change Model Qty</i>	Berisikan jumlah dari unit yang mengalami perubahan yang disebabkan beberapa faktor.
15.	Penerimaan <i>Change Model Cum</i>	Berisikan total dari jumlah unit yang mengalami perubahan.
16.	Pengiriman <i>Qty</i>	Berisikan jumlah unit yang dikirimkan ke distributor.
17.	Pengiriman <i>Cum</i>	Berisikan total dari jumlah unit yang dikirimkan ke distributor.
18.	Pengiriman <i>IGTS Out Qty</i>	Berisikan jumlah unit yang keluar dari gudang untuk ditransfer ke gudang lain, baik gudang internal maupun gudang lokasi eksternal.
19.	Pengiriman <i>IGTS Out Cum</i>	Berisikan total jumlah unit yang keluar dari gudang untuk ditransfer ke gudang lain.
20.	Pengiriman <i>Repair Qty</i>	Berisikan jumlah unit yang mengalami perbaikan.
21.	Pengiriman <i>Repair Cum</i>	Berisikan total dari jumlah unit yang mengalami perbaikan.
22.	Pengiriman <i>Change Model Qty</i>	Berisikan jumlah dari unit yang mengalami perubahan yang disebabkan beberapa faktor.
23.	Pengiriman <i>Change Model Cum</i>	Berisikan total dari jumlah unit yang mengalami perubahan.

Tabel IV.10 Dekripsi *Field-field* Laporan Harian Stok *Warehouse* Ex-Berikat  
(lanjutan).

No.	<i>Field</i>	Deskripsi
24.	<i>Adjustment Qty</i>	Berisikan jumlah unit pada data yang mengalami selisih kuantitas dibandingkan dengan stok secara fisik.
25.	<i>Adjustment Cum</i>	Berisikan total jumlah unit pada data yang mengalami selisih kuantitas dibandingkan dengan stok secara fisik.
26.	<i>Adjustment Repair Qty</i>	Berisikan jumlah unit yang selisih dengan stok fisik dan mengalami perbaikan.
27.	<i>Adjustment Repair Cum</i>	Berisikan total jumlah unit yang selisih dengan stok fisik dan mengalami perbaikan.
28.	Stok Akhir <i>Qty</i>	Berisikan jumlah unit yang dihitung secara fisik di gudang.
29.	Stok Akhir <i>Repair</i>	Berisikan jumlah unit yang dihitung secara fisik di gudang dan harus mengalami perubahan.
30.	Total	Berisikan total jumlah unit pada gudang.

(Sumber: PT Suzuki Indomobil Motor, 2019)

#### 4.11 Analisis Sistem Pendistribusian Unit

BPMN (*Business Process Modelling Notation*) merupakan teknik yang memungkinkan semua pihak yang terlibat dalam proses berkomunikasi secara jelas, benar dan efisien. BPMN menyediakan kemampuan memahami prosedur internal bisnis dalam notasi grafis. BPMN mengikuti tradisi *flowcharting* notasi untuk dapat dengan mudah dibaca dan juga fleksibilitas.

Tujuan analisis sistem yang berjalan digambarkan dengan BPMN (*Business Process Modelling Notation*) ialah untuk memudahkan pemahaman

terhadap sistem tersebut dan menyederhanakan rangkaian proses atau prosedur dengan meng gambarkannya secara ringkas, jelas dan logis.

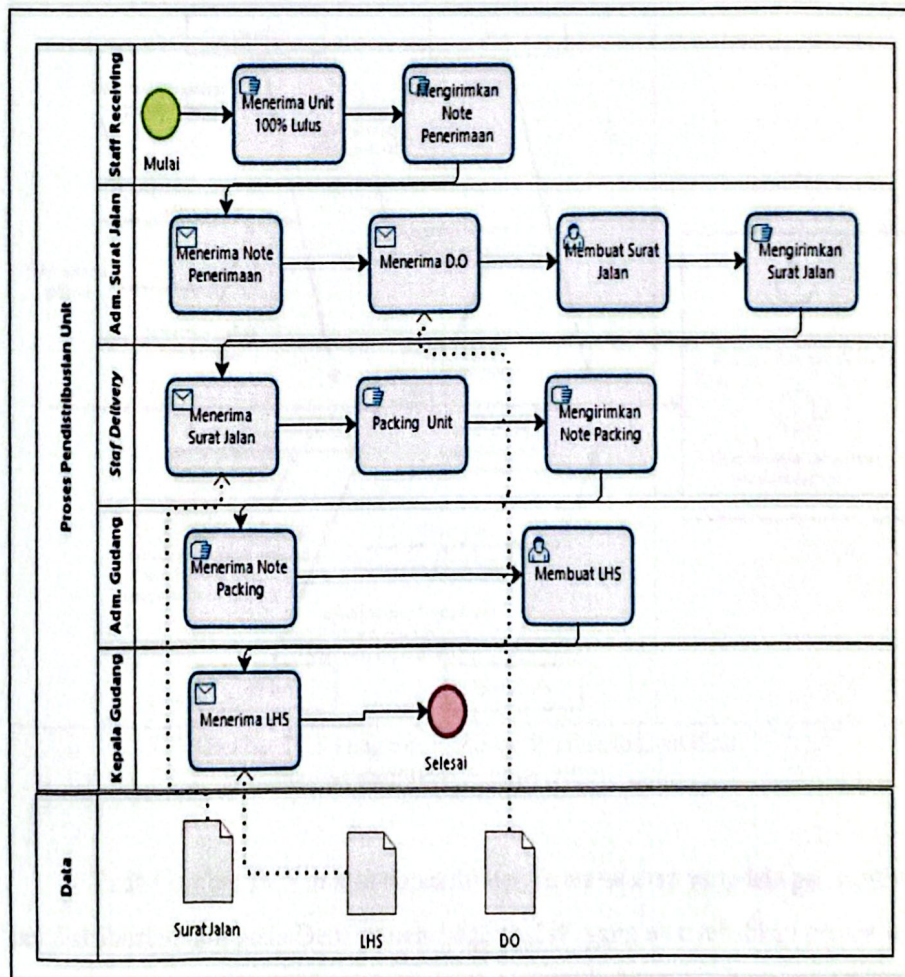
#### 4.12 BPMN Proses Pendistribusian Unit

Berdasarkan analisis sistem yang ada pada PT Suzuki Indomobil Motor *Plant I Tambun*, berikut adalah proses pendistribusian unit pada Departemen *Logistic 2W*:

1. Staf *receiving* menerima unit yang sudah lulus 100% dari proses pengecekan kualitas oleh Departemen *Final Inspection*. Selanjutnya staf *receiving* mengirimkan *note* kertas bahwa unit sudah diterima pada admin surat jalan.
2. Admin surat jalan menerima *note* kertas, lalu admin surat jalan juga menerima DO (*delivery order*) yang masuk dari Bagian Marketing. DO yang diterima kemudian dijadikan input dan nantinya akan diterbitkan surat jalan unit oleh admin surat jalan.
3. Admin mengirimkan dokumen surat jalan pada staf *delivery*. Surat jalan yang diterima oleh staf *delivery* akan dikirimkan bersamaan dengan unit yang dikirim kepada distributor oleh pihak ekspedisi.
4. Staf *delivery* selanjutnya akan melakukan *packing* unit. Sebelum dilakukan proses *packing*, adanya persiapan *packing* seperti: menyiapkan peralatan *packing*, menyiapkan jalur atau lokasi *packing* dan memastikan semua surat yang berkaitan dengan unit sudah siap.
5. Proses *packing* dilakukan dan jika ada kondisi dimana unit yang tidak lengkap, maka akan dilakukan persiapan kembali, seperti dilihat bagian apa yang kurang pada unit. Ketika unit sudah lengkap maka unit dimasukkan dalam truk untuk diangkut dan dikirimkan pada tujuan yaitu distributor.
6. Staf *delivery* selanjutnya akan membuat *note* untuk menjelaskan bahwa sudah dilakukannya aktivitas *packing* dan *note* tersebut dikirimkan kepada admin gudang.

7. Admin gudang menerima *note*, kemudian melakukan pencatatan unit pada gudang PT Suzuki Indomobil Motor *Plant I Tambun*.
8. Laporan yang pertama dibuat oleh admin gudang ialah LHS (laporan stok harian) *warehouse* Gudang Tambun, setelah itu admin membuat LHS untuk Gudang Ex-Berikat.
9. Laporan gudang akhirnya diberikan kepada kepala gudang sebagai arsip untuk kebutuhan perusahaan.

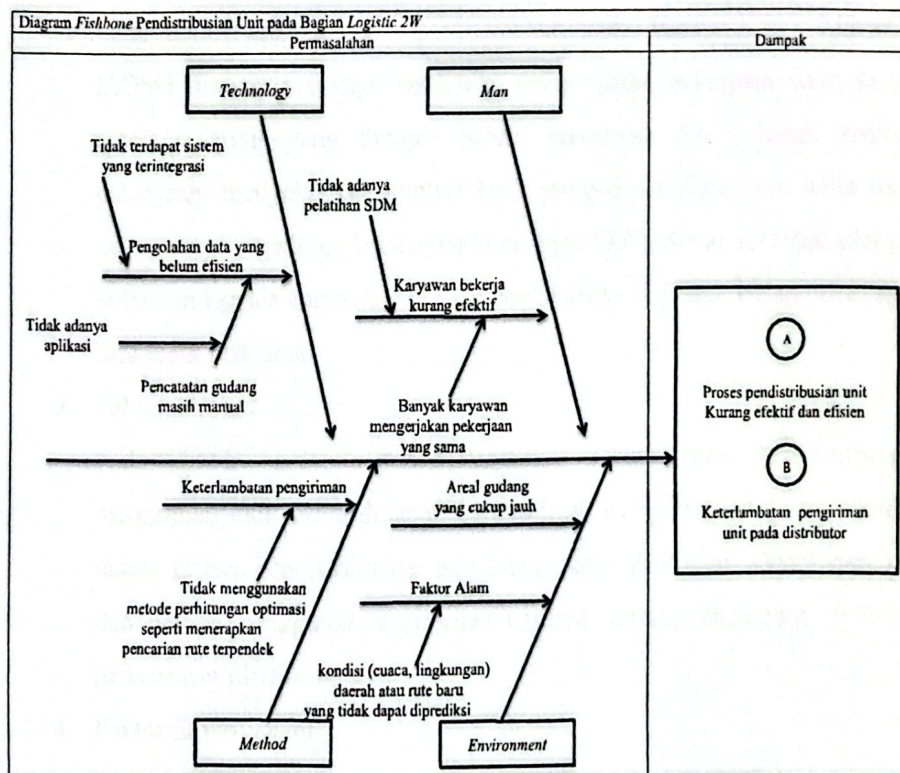
Berikut adalah proses pendistribusian unit yang berjalan pada Departemen *Logistic 2W* PT Suzuki Indomobil Motor *Plant I Tambun*, dapat dilihat pada Gambar IV.7 di bawah ini:



Gambar IV.7 Proses Bisnis Sistem Pendistribusian Unit yang Berjalan  
Sumber: Hasil Analisis (2020)

#### 4.13 Analisis Masalah

Diagram tulang ikan (*fishbone diagram*) atau diagram sebab-akibat merupakan diagram untuk memperlihatkan atau mendeteksi suatu permasalahan yang terjadi dalam suatu proses. Pembuatan diagram ini bertujuan mencari faktor-faktor yang mungkin menjadi penyebab dari suatu masalah atau penyimpangan. Dengan mengumpulkan semua penyebab masalah sehingga menghasilkan suatu gambaran dari permasalahan tersebut kemudian dapat mempermudah pengambilan keputusan dalam menyelesaikan permasalahan tersebut. Berikut adalah masalah yang terjadi pada proses pendistribusian unit pada Departemen *Logistic 2W* di PT Suzuki Indomobil Motor *Plant I Tambun*:



Gambar IV.8 Diagram *Fishbone* Pendistribusian Unit  
Sumber: Hasil Analisis (2020)

Pada Gambar IV.8 di atas dapat dilihat permasalahan yang ada pada sistem pendistribusian unit pada Departemen *Logistic 2W* yang menyebabkan proses ada yang tidak berjalan dengan efektif dan efisien, sehingga menyebabkan beberapa

kendala dalam proses pendistribusian unit. Adapun pokok permasalahan tersebut sebagai berikut:

1. Faktor *Technology*

Pada faktor teknologi, permasalahan yang ada adalah pengolahan data pada perusahaan belum efisien dikarenakan belum adanya aplikasi pendistribusian yang tepat dan proses pencatatan laporan yang dikerjakan juga masih menggunakan proses konvensional. Selain itu, tidak ada aplikasi penyimpanan data sehingga tidak tersimpannya data pada *database* yang tidak terorganisir dan sulit mendapatkan informasi terkait.

2. Faktor *Man*

Pada faktor *Man* terdapat permasalahan pada Sumber Daya Manusia (SDM) dimana beberapa karyawan mengerjakan pekerjaan yang sama sehingga hasil yang didapat belum maksimal dan terkesan sedikit pekerjaan, menyebabkan kontrol kerja yang dapat dikerjakan lebih dari satu orang menjadi lambat karena kurangnya SDM. Selain itu tidak adanya pelatihan kepada karyawan untuk meningkatkan *skill* agar setiap pekerjaan bisa lebih maksimal.

3. Faktor *Method*

Pada faktor *method* permasalahannya ialah kasus keterlambatan pengiriman unit sering didapati karena tidak menggunakan suatu metode dalam proses pendistribusian unit yang masih dilakukan secara *random* dan belum menggunakan prioritas tertentu untuk mendukung efisiensi pada proses distribusi.

4. Faktor *Environment*

Pada faktor *environment* permasalahannya ialah faktor alam dari areal pengiriman unit seperti dari lokasi rute baru, ditambah cuaca dan kondisi alam lainnya yang tidak dapat diprediksi. Selain itu areal gudang penyimpanan unit yang berbeda lokasi membuat waktu pengambilan unit cukup lama.

#### 4.14 Perhitungan Fungsi *Euclidean Distance*

Dalam melakukan perhitungan menggunakan fungsi *Euclidean Distance*, diperlukan langkah-langkah perhitungan sesuai dengan ketentuan perhitungan aturan prioritas yang ada. Dalam kasus ini terdapat data sampel untuk distributor PT Suzuki Indomobil Motor *Plant I Tambun*.

##### 4.14.1 Perhitungan Data Sampel Distributor

Berikut ini merupakan data sampel distributor PT Suzuki Indomobil Motor *Plant I Tambun*, dapat dilihat pada tabel di bawah ini.

Tabel IV.11 Data Sampel Distributor

Distributor	Alamat	<i>Longitudinal</i>	<i>Latitude</i>
Suzuki Takengon	Jl.Lintang, Kebayakan, Takengon Tim., Lut Tawar, Kabupaten Aceh Tengah, Aceh 24519	96.718606	5.203716
PT. Sunindo Varia Motor Gemilang	Jl. H. Adam Malik No.101, Silalas, Kec. Medan Bar., Kota Medan, Sumatera Utara 20234, Indonesia	98.668753	3.602603
Suzuki Elang Perkasa Motor	Jl. Khatib Sulaiman No.87, Ulak Karang Utara, Kec. Padang Utara, Kota Padang, Sumatera Barat 25137, Indonesia	100.352406	-0.904592
Suzuki PT Riau Jaya Cemerlang	Jl. Tuan Tambusai No.18J,K,L, Labuh Baru Tim., Kec. Payung Sekaki, Kota Pekanbaru, Riau 28292, Indonesia	101.423534	0.505329

Tabel IV.11 Data Sampel Distributor (lanjutan)

Distributor	Alamat	Longitudinal	Latitude
Suzuki Jaya Indah Motor Kerinci	Sekungkung, Depati VII, Kabupaten Kerinci, Jambi 37161, Indonesia	101.382779	-1.931757

Sumber: Hasil Analisis (2020)

Tabel IV.11 di atas mempunyai keterangan pada setiap *field* yang ada adalah sebagai berikut:

1. Distributor merupakan keterangan dari nama distributor PT Suzuki Indomobil Motor *Plant I* Tambun.
2. Alamat merupakan keterangan lokasi distributor.
3. *Longitudinal* merupakan keterangan dari nilai garis bujur lokasi distributor.
4. *Latitude* merupakan keterangan dari nilai garis mendatar lokasi distributor.

Dapat dijelaskan untuk melakukan perhitungan dengan fungsi *euclidean distance* sebagai berikut:

1. Rumus dasar perhitungan *Euclidean Distance*

$$d = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$$

dimana:

$x1$  = koordinat  $x$  pada pusat penelitian/titik awal

$y1$  = koordinat  $y$  pada pusat penelitian/titik awal

$x2$  = koordinat  $x$  pada tujuan penelitian/titik tujuan

$y2$  = koordinat  $y$  pada tujuan penelitian/titik tujuan

$d$  = jarak antara pusat penelitian  $x$  dan  $y$

2. Rumus perhitungan jarak *Euclidean Distance*

$$\text{Jarak} = \sqrt{(\text{Latitude1} - \text{Latitude2})^2 + (\text{Longitude1} - \text{Longitude2})^2}$$

dimana:

*Latitude1* = koordinat *Latitude1* pada titik awal

*Latitude2* = koordinat *Latitude2* pada titik tujuan

$Longitude_1$  = koordinat  $Longitude_1$  pada titik awal

$Longitude_2$  = koordinat  $Longitude_1$  pada titik tujuan

Dalam melakukan perhitungan jarak melakukan fungsi *euclidean distance* ada beberapa hal yang harus diperhatikan, antara lain:

1. Harus menetapkan titik awal dan titik tujuan.
2. Titik awal ialah yang menjadi ajuan jarak pada masing-masing titik tujuan dengan nilai jarak yang merupakan garis lurus.
3. Nilai pada titik awal dan akhir dalam perhitungan jarak diperoleh dari bobot *latitude* dan *longitude*.
4. Hasil perhitungan (jarak) masih dalam satuan *decimal degree* (sesuai dengan format longlat yang dipakai) sehingga untuk menyesuaikannya perlu dikalikan dengan 111.319 km (1 derajat bumi = 111.319 km).

Didapatkan contoh perhitungan menggunakan *euclidean distance* untuk mengetahui jarak perusahaan dengan distributor dapat dilihat pada Tabel IV.12 di bawah ini:

Tabel IV.12 Nilai Jarak Perusahaan dengan Distributor

Distributor	Perusahaan	Jarak
Suzuki Takengon	PT Suzuki Indomobil Motor <i>Plant</i> I Tambun	1676.227048 Km
PT. Sunindo Varia Motor Gemilang	PT Suzuki Indomobil Motor <i>Plant</i> I Tambun	1438.905015 Km
Suzuki Elang Perkasa Motor	PT Suzuki Indomobil Motor <i>Plant</i> I Tambun	952.6129801 Km
Suzuki PT Riau Jaya Cemerlang	PT Suzuki Indomobil Motor <i>Plant</i> I Tambun	977.6573139 Km
Suzuki Jaya Indah Motor Kerinci	PT Suzuki Indomobil Motor <i>Plant</i> I Tambun	791.8094641 Km

Sumber: Hasil Analisis (2020)

#### 4.15 Perhitungan Algoritma *Dijkstra*

Dalam melakukan perhitungan menggunakan Algoritma *Dijkstra*, diperlukan langkah-langkah perhitungan sesuai dengan ketentuan perhitungan aturan prioritas yang ada. Berikut urutan logika pada algoritma *dijkstra* adalah sebagai berikut:

1. Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu *set* nilai 0 pada *node* awal dan nilai tak hingga terhadap *node* lain (belum terisi)
2. *Set* semua *node* “belum terpilih” dan *set node* awal sebagai “*node* keberangkatan”
3. Dari *node* keberangkatan, pertimbangkan *node* tetangga yang belum terpilih dan hitung jaraknya dari titik keberangkatan. Sebagai contoh, jika titik keberangkatan A ke B memiliki bobot jarak 6 dan dari B ke *node* C berjarak 2, maka jarak ke C melewati B menjadi  $6+2=8$ . Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru.
4. Saat selesai mempertimbangkan setiap jarak terhadap *node* tetangga, tandai *node* yang telah terpilih sebagai “*Node* terpilih”. *Node* terpilih tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.
5. *Set* “*Node* belum terpilih” dengan jarak terkecil (dari *node* keberangkatan) sebagai “*Node* Keberangkatan” selanjutnya dan lanjutkan dengan kembali ke step 3.

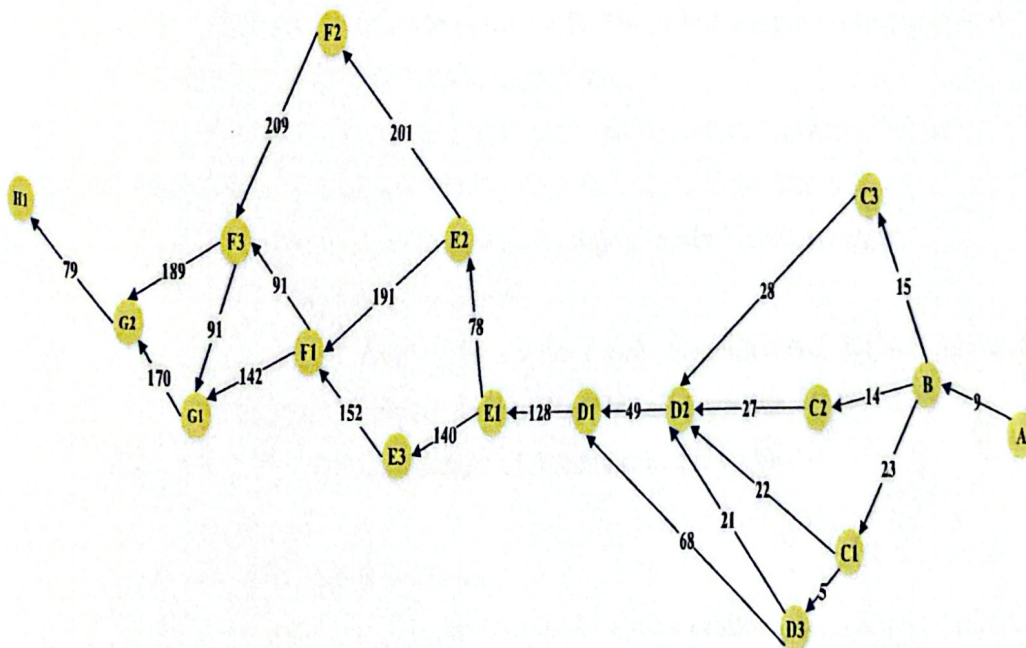
Selain logika pada algoritma *dijkstra*, adapun urutan dari langkah penyelesaian kasus, antara lain:

##### 1. Pengambilan Data

Dibutuhkan data untuk melakukan perhitungan nilai (jarak) minimum. Dalam hal ini, data diambil berdasarkan nilai (jarak) diperoleh melalui posisi koordinat suatu wilayah. Posisi koordinat dapat diambil melalui sumber yang dimiliki seperti *Google Maps* atau layanan serupa yang memiliki pemetaan suatu wilayah.

## 2. Membuat Graf Berarah

Graf berarah merupakan graf yang tiap busur mempunyai anak panah dan bobot. Graf berarah merupakan suatu bentuk rancangan atau kerangka secara garis besar yang memuat gambaran umum dari apa yang akan diperhitungkan. Graf berarah yang dibuat berupa gambar antar titik yang saling terhubung oleh garis sesuai dengan posisi koordinat yang telah didapat sebelumnya. Adapun Gambar IV.9 adalah contoh graf berarah pada sebuah kasus perhitungan jarak PT Suzuki Indomobil Motor *Plant I Tambun* menggunakan algoritma *dijkstra*:



Gambar IV.9 Graf Berarah Studi Kasus Perhitungan Jarak  
Sumber: Hasil Analisis (2020)

Dari Gambar IV.9 studi kasus perhitungan jarak menggunakan algoritma *dijkstra* setiap *node* menggambarkan wilayah yang akan dilalui. Adapun keterangan setiap *node* sebagai berikut:

- a. *Node A* = PT Suzuki Indomobil Motor *Plant I Tambun* atau sebagai titik awal keberangkatan.
- b. *Node B* = Wilayah Bekasi.
- c. *Node C* = Wilayah Jakarta.
  - 1) *Node C1* = Daerah Pondok Indah.

- 2) *Node C2* = Daerah Saharjo, Tebet.
- 3) *Node C3* = Daerah Sunter, Tanjung Priok.
- d. *Node D* = Wilayah Banten
  - 1) *Node D1* = Daerah Kota Serang, Banten.
  - 2) *Node D2* = Daerah Parung Serap, Kota Tangerang, Banten.
  - 3) *Node D3* = Daerah Cipayung, Kota Tangerang Selatan, Banten.
- e. *Node E* = Wilayah Lampung
  - 1) *Node E1* = Daerah Telukbetung Selatan, Kota Bandar Lampung.
  - 2) *Node E2* = Daerah Kotabumi, Kabupaten Lampung Utara.
  - 3) *Node E3* = Daerah Liwa Balik Bukit, Kabupaten Lampung Barat.
- f. *Node F* = Wilayah Sumatera Selatan
  - 1) *Node F1* = Daerah Lahat, Kabupaten Lahat, Sumatera Selatan.
  - 2) *Node F2* = Daerah Sebrang Ulu I, Kota Palembang.
  - 3) *Node F3* = Daerah Lubuk Linggau, Kota Lubuk Linggau.
- g. *Node G* = Wilayah Bengkulu
  - 1) *Node G1* = Daerah Jembatan Kecil, Singaran Pati, Kota Bengkulu
  - 2) *Node G2* = Daerah Sungai Rumbai, Kabupaten Mukomuko
- h. *Node H1* = Wilayah Jambi atau sebagai titik tujuan.

### 3. Membuat Tabel Perhitungan

Langkah berikutnya adalah membuat tabel perhitungan, dimana langkah ini sangat penting karena tabel ini akan menentukan proses perhitungan secara keseluruhan. Contohnya dalam tabel ini, pertama dilakukan adalah memberikan nilai awal titik keberangkatan yaitu 0, dan nilai tak hingga ( $\infty$ ) kepada titik yang lain. Kemudian membandingkan titik dengan nilai (jarak) terkecil dengan titik lain yang terhubung dengan melakukan perhitungan nilainya. Proses atau langkah tersebut terus berlanjut hingga ditemukan nilai (jarak) terkecil atau minimum dari titik awal hingga ke titik tujuan. Pada studi kasus ini titik tujuan ialah H1 dan titik awal keberangkatan ialah A. Berikut merupakan Tabel IV.13 Perhitungan *Dijkstra*.

Tabel IV.13 Perhitungan Dijkstra

V	A	B	C1	C2	C3	D1	D2	D3	E-1	E2	E3	F1	F2	F3	G1	G2	H1
A	0A	9A	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
B		9A	32B	23B	24B	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
C2			32B	23B	24B	∞	50C2	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
C3			32B	23B	24B	∞	50C2-52C3	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
C1			32B	23B	24B	∞	50C2-54C1	37C1	∞	∞	∞	∞	∞	∞	∞	∞	∞
D3						105D3	50C2-58D3	37C1	∞	∞	∞	∞	∞	∞	∞	∞	∞
D2						99D2	50C2-58D3	37C1	∞	∞	∞	∞	∞	∞	∞	∞	∞
D1						99D2	50C2-58D3	37C1	227D1	∞	∞	∞	∞	∞	∞	∞	∞
E1									227D1	305/E1	367/E1	∞	∞	∞	∞	∞	∞
E2										305/E1	367/E1	496/E2	506/E2	∞	∞	∞	∞
E3											367/E1	496/E2 - 519E3	506/E2	∞	∞	∞	∞
F1												496/E2 - 519E3	506/E2	587/F1	638/F1	∞	∞
F2													506/E2	587/F1-715/F2	638/F1	∞	∞
F3														587/F1-715/F2	638/F1-678/F3	776F3	∞
G1															638/F1-678/F3	776/F3 - 848/G1	∞
G2																776/F3 - 848/G1	855G2

Sumber: Hasil Analisis (2020)

#### 4. Kesimpulan

Kesimpulan merupakan langkah akhir, dimana akan diperoleh hasil penentuan rute terpendek dari titik awal keberangkatan A hingga titik akhir atau tujuan H1. Pada langkah terakhir ini dalam contoh kasus di atas, dapat disimpulkan bahwa nilai (jarak) terpendek dari titik awal (A) menuju titik tujuan (H1) adalah 855 Km, dengan lintasan melalui  $A \rightarrow B \rightarrow C2 \rightarrow D2 \rightarrow D1 \rightarrow E1 \rightarrow E2 \rightarrow F1 \rightarrow F3 \rightarrow G2 \rightarrow H1$

Berikut merupakan hasil perbandingan total jarak setiap jalur yang didapatkan dari titik awal keberangkatan (A) hingga titik akhir/tujuan (H1) sebagai berikut:

##### 1. Jalur 1

$A \rightarrow B \rightarrow C1 \rightarrow D2 \rightarrow D1 \rightarrow E1 \rightarrow E3 \rightarrow F1 \rightarrow G1 \rightarrow G2 \rightarrow H1$

Total jarak tempuh = 914 Km

##### 2. Jalur 2

$A \rightarrow B \rightarrow C1 \rightarrow D2 \rightarrow D1 \rightarrow E1 \rightarrow E2 \rightarrow F2 \rightarrow F3 \rightarrow G1 \rightarrow G2 \rightarrow H1$

Total jarak tempuh = 1058 Km

##### 3. Jalur 3

$A \rightarrow B \rightarrow C1 \rightarrow D2 \rightarrow D1 \rightarrow E1 \rightarrow E2 \rightarrow F2 \rightarrow F3 \rightarrow G2 \rightarrow H1$

Total jarak tempuh = 987 Km

##### 4. Jalur 4

$A \rightarrow B \rightarrow C1 \rightarrow D2 \rightarrow D1 \rightarrow E1 \rightarrow E2 \rightarrow F1 \rightarrow F3 \rightarrow G1 \rightarrow G2 \rightarrow H1$

Total jarak tempuh = 931 Km

##### 5. Jalur 5

$A \rightarrow B \rightarrow C1 \rightarrow D2 \rightarrow D1 \rightarrow E1 \rightarrow E2 \rightarrow F1 \rightarrow F3 \rightarrow G2 \rightarrow H1$

Total jarak tempuh = 859 Km

##### 6. Jalur 6

$A \rightarrow B \rightarrow C2 \rightarrow D2 \rightarrow D1 \rightarrow E1 \rightarrow E2 \rightarrow F1 \rightarrow F3 \rightarrow G2 \rightarrow H1$

Total jarak tempuh = 855 Km

##### 7. Jalur 7

$A \rightarrow B \rightarrow C2 \rightarrow D2 \rightarrow D1 \rightarrow E1 \rightarrow E3 \rightarrow F1 \rightarrow F3 \rightarrow G2 \rightarrow H1$

Total jarak tempuh = 878 Km

8. Jalur 8

$A \rightarrow B \rightarrow C2 \rightarrow D2 \rightarrow D1 \rightarrow E1 \rightarrow E3 \rightarrow F1 \rightarrow G1 \rightarrow G2 \rightarrow H1$

Total jarak tempuh = 910 Km

9. Jalur 9

$A \rightarrow B \rightarrow C2 \rightarrow D2 \rightarrow D1 \rightarrow E1 \rightarrow E2 \rightarrow F2 \rightarrow F3 \rightarrow G1 \rightarrow G2 \rightarrow H1$

Total jarak tempuh = 1055 Km

10. Jalur 10

$A \rightarrow B \rightarrow C2 \rightarrow D2 \rightarrow D1 \rightarrow E1 \rightarrow E2 \rightarrow F2 \rightarrow F3 \rightarrow G2 \rightarrow H1$

Total jarak tempuh = 964 Km

11. Jalur 11

$A \rightarrow B \rightarrow C1 \rightarrow D3 \rightarrow D2 \rightarrow D1 \rightarrow E1 \rightarrow E2 \rightarrow F1 \rightarrow F3 \rightarrow G2 \rightarrow H1$

Total jarak tempuh = 863 Km

12. Jalur 12

$A \rightarrow B \rightarrow C1 \rightarrow D3 \rightarrow D1 \rightarrow E1 \rightarrow E2 \rightarrow F1 \rightarrow F3 \rightarrow G2 \rightarrow H1$

Total jarak tempuh = 861 Km

13. Jalur 13

$A \rightarrow B \rightarrow C3 \rightarrow D2 \rightarrow D1 \rightarrow E1 \rightarrow E3 \rightarrow F1 \rightarrow G1 \rightarrow G2 \rightarrow H1$

Total jarak tempuh = 912 Km

14. Jalur 14

$A \rightarrow B \rightarrow C3 \rightarrow D2 \rightarrow D1 \rightarrow E1 \rightarrow E2 \rightarrow F1 \rightarrow F3 \rightarrow G2 \rightarrow H1$

Total jarak tempuh = 857 Km

15. Jalur 15

$A \rightarrow B \rightarrow C3 \rightarrow D2 \rightarrow D1 \rightarrow E1 \rightarrow E2 \rightarrow F1 \rightarrow F3 \rightarrow G1 \rightarrow G2 \rightarrow H1$

Total jarak tempuh = 929 Km

Dari nilai setiap jalur pada studi kasus di atas, didapatkan rute terpendek ialah jalur 6 dengan total jarak tempuh 855 Km. Algoritma *Dijkstra* terbukti dapat membantu untuk menentukan rute terpendek pada kasus pendistribusian unit.

## BAB V

### ANALISIS DAN PEMBAHASAN

#### 5.1 *System Request*

Proses ini menjelaskan kebutuhan perusahaan dan bisnis yang akan didapatkan oleh perusahaan. Berikut *system request* yang akan diusulkan pada Tabel V.I:

Tabel V.1. *System Request*

<i>Project Element</i>	<i>Description</i>
<i>Project Name</i>	Sistem Informasi Pendistribusian Unit pada PT Suzuki Indomobil Motor <i>Plant I Tambun</i>
<i>Business Need</i>	Proyek ini bermaksud untuk membangun sebuah sistem informasi pendistribusian unit untuk mengintegrasikan data yang terkait pada proses pendistribusian
<i>Business Requirements</i>	<ol style="list-style-type: none"> <li>1. Dapat melakukan perhitungan rute terpendek pada proses pendistribusian unit ke distributor</li> <li>2. Dapat mengintegrasikan data antar bagian</li> </ol>
<i>Business Value</i>	<ol style="list-style-type: none"> <li>1. Proses bisnis di perusahaan menjadi lebih mudah</li> <li>2. Waktu dalam melakukan pengantaran lebih cepat</li> <li>3. Pengelolaan data terintegrasi dengan <i>database</i></li> <li>4. Pekerjaan lebih terukur dalam hal waktu dan tenaga</li> </ol>
<i>Constraints</i>	Sistem informasi yang dapat mendukung kebutuhan proses pendistribusian unit menggunakan PHP <i>Codeigniter</i> dengan <i>database</i> MariaDB

Sumber: Hasil Analisis (2020)

#### 5.2 *Analisis Kebutuhan Sistem*

Berdasarkan hasil analisis sistem pendistribusian unit yang berjalan pada PT Suzuki Indomobil Motor *Plant I Tambun*, maka diajukan sistem informasi usulan untuk menangani kekurangan dan memperbaiki sistem yang sudah ada sebelumnya.

### 5.2.1 Kebutuhan Fungsional Sistem

Berikut adalah analisis kebutuhan sistem informasi pendistribusian unit dapat dilihat pada Tabel V.2 di bawah ini:

Tabel V.2 Analisis Kebutuhan Sistem Informasi Pendistribusian Unit

No.	Permasalahan	Kebutuhan <i>User</i>	Solusi	<i>Functional Requirement</i>
1.	Proses penyebaran informasi berupa <i>note</i> ketersediaan unit dan SJ yang belum diterima secara langsung	Sistem terkomputerisasi yang dapat membantu proses penyebaran <i>note</i> antar bagian terkait	Merancang dan membangun sistem yang membantu penyebaran informasi	Sistem mampu memproses data terkait untuk menyampaikan <i>note</i> dan diterima secara langsung
2	Proses pengiriman unit yang belum menerapkan metode pengaturan rute dalam pengantaran	Sistem yang dapat membantu menyelesaikan masalah pengantaran unit dengan rute terpendek	Merancang suatu sistem informasi yang mampu menentukan rute terpendek dalam pengantaran unit	Sistem dapat memproses data yang terkait untuk menghasilkan informasi rute terpendek
3	Pengolahan laporan gudang yang belum terintegrasi antar bagian dan kurang informatif	Sistem pengolahan laporan gudang yang dapat menghemat waktu kerja dan lebih informatif	Membuat proses pengolahan laporan gudang yang informatif dan saling terintegrasi antar bagian terkait	Sistem dapat memproses data terkait untuk menghasilkan laporan gudang yang informatif

Sumber: Hasil Analisis (2020)

### 5.2.2 Kebutuhan Sistem (*Non Functional Requirement*)

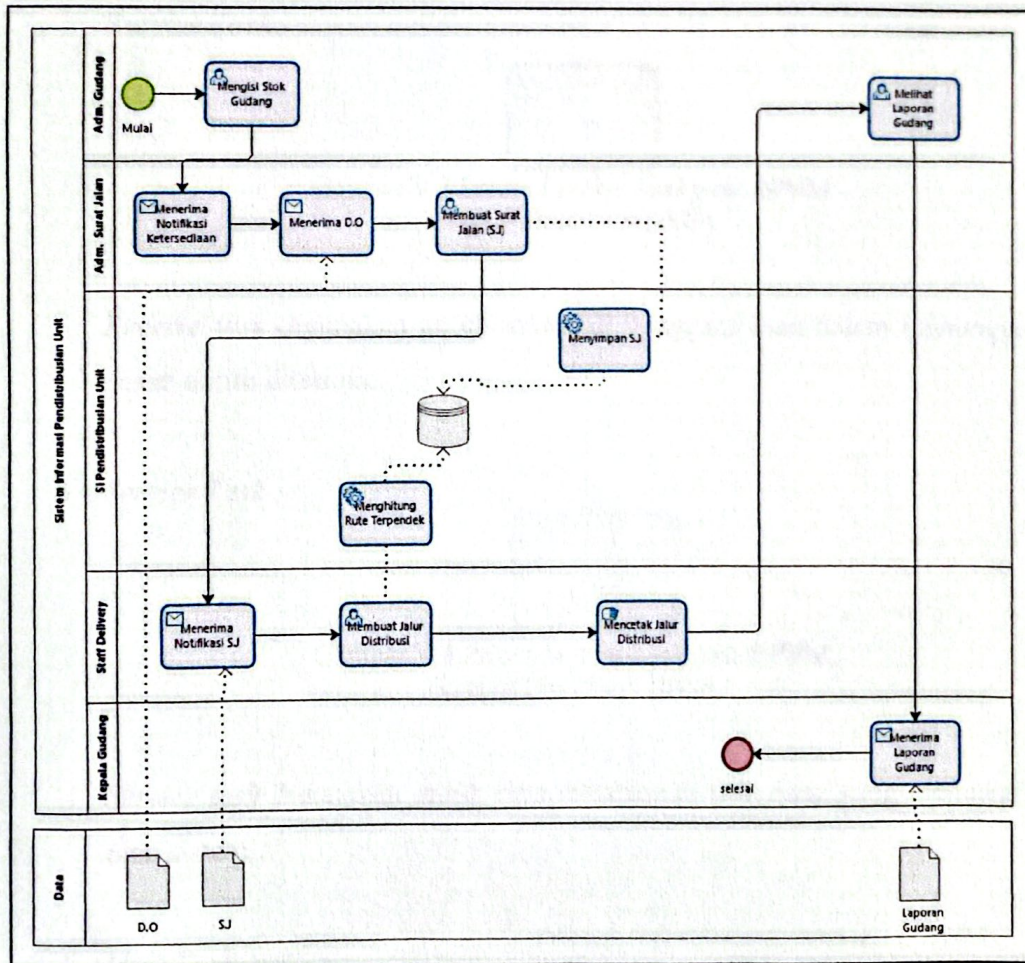
*Non functional requirement* dari sistem informasi pendistribusian unit yang diusulkan antara lain:

1. Aplikasi berbasis web sehingga harus dijalankan dengan koneksi internet.
2. Aplikasi ini menggunakan *username* dan *password* sebagai syarat akses masuk ke dalam sistem.
3. Aplikasi ini dapat diakses oleh Departemen *Logistic 2W* dengan penggunaan hak akses yang berbeda.
4. Sistem dapat dijalankan oleh beberapa *software web browser*.
5. Aplikasi ini menggunakan *printer* untuk mencetak laporan yang terkait pendistribusian unit.
6. Aplikasi dijalankan dengan bantuan perangkat *input* berupa *keyboard* dan *mouse*.

### 5.3 Proses Bisnis Pendistribusian Unit Usulan

Berdasarkan analisis sistem yang sedang berjalan pada PT Suzuki Indomobil Motor *Plant I Tambun*, maka diusulkan suatu proses pendistribusian unit sebagai berikut:

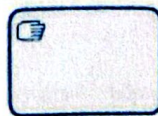
1. Admin gudang melakukan input stok pada gudang sebagai informasi ketersediaan unit pada gudang dan sistem dengan otomatis mengirimkan notifikasi ketersediaan.
2. Admin surat jalan menerima notifikasi ketersediaan yang dikirim oleh admin gudang. Selanjutnya, admin surat jalan menerima *delivery order* yang masuk dari Bagian *Marketing*.
3. Setelah itu, admin surat jalan membuat surat jalan unit yang selanjutnya diserahkan pada staf *delivery*. Sistem dengan otomatis kembali mengirimkan notifikasi SJ (surat jalan).
4. Staf *delivery* menerima notifikasi SJ, kemudian membuat jalur distribusi dengan memperhitungkan jarak dan menghasilkan rute terpendek untuk pengiriman.
5. Selanjutnya, admin gudang kemudian melihat laporan ketersediaan unit gudang yang *update* otomatis.
6. Kepala gudang juga dapat melihat laporan ketersediaan unit gudang yang masuk dan melakukan *save* dengan men-*download* file laporan.



Gambar V.1 Proses Bisnis Sistem Pendistribusian Unit Usulan  
 Sumber: Hasil Analisis (2020)

Keterangan:

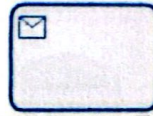
1. *Manual Task*



Gambar V.2 Proses *Manual Task* pada BPNM  
 Sumber: Hasil Teori (2020)

*Manual task* digunakan untuk menggambarkan proses yang membutuhkan keterlibatan manusia dalam menyelesaikannya.

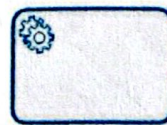
## 2. *Receive Task*



Gambar V.3 Proses *Receive Task* pada BPNM  
Sumber: Hasil Teori (2020)

*Receive task* digunakan untuk mewakili penggambaran dalam menunggu pesan untuk diterima.

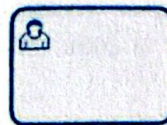
## 3. *Service Task*



Gambar V.4 Proses *Service Task* pada BPNM  
Sumber: Hasil Teori (2020)

*Service task* digunakan untuk menggambarkan pekerjaan yang dilakukan oleh sistem.

## 4. *User Task*

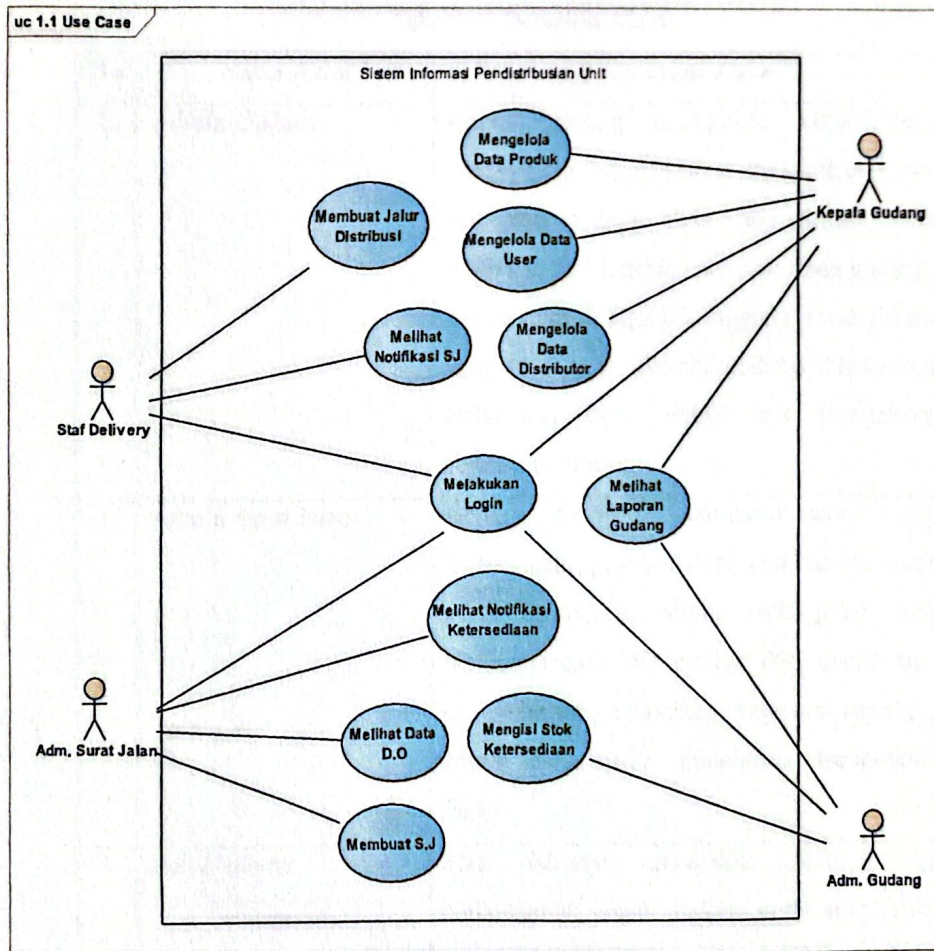


Gambar V.5 Proses *User Task* pada BPNM  
Sumber: Hasil Teori (2020)

*User task* digunakan untuk menggambarkan proses yang dilakukan pengguna dengan antarmuka yang dibuat.

### 5.4 *Use Case Diagram* Sistem Informasi Pendistribusian Unit

*Use Case Diagram* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Berikut adalah *Use Case Diagram* sistem informasi pendistribusian unit pada PT Suzuki Indomobil Motor Plant I Tambun yang diusulkan dapat dilihat pada Gambar V.6 di bawah ini:



Gambar V.6 Use Case Diagram Sistem Pendistribusian Unit Usulan  
Sumber: Hasil Analisis (2020)

#### 5.4.1 Use Case Description

Berikut adalah deskripsi dari *Use Case Diagram* yang telah dibuat:

##### 1. Deskripsi Aktor

Aktor merupakan penggambaran tokoh atau sistem yang memperoleh keuntungan dan berada di luar sistem. Pendefinisian aktor dari proses pendistribusian unit pada PT Suzuki Indomobil Motor *Plant I Tambun* sebagai berikut:

Tabel V.3 Deskripsi Aktor

No	Nama Aktor	Deskripsi Aktor
1.	Admin Gudang	Admin gudang merupakan aktor yang bertanggung jawab dalam melakukan input stok pada gudang. Stok yang diinput ialah informasi dari ketersediaan unit pada gudang. Admin gudang juga bertanggung jawab dalam laporan gudang. Admin gudang diharuskan melakukan <i>login</i> untuk bisa mengakses kegiatan tersebut.
2.	Admin Surat Jalan	Admin surat jalan merupakan aktor yang bertanggung jawab dalam pembuatan surat jalan. Selain itu admin surat jalan juga menerima data DO produk dari distributor. Untuk dapat melakukan kegiatan tersebut, admin surat jalan diharuskan melakukan <i>login</i> .
3.	Staf <i>Delivery</i>	Staf <i>delivery</i> merupakan aktor yang bertanggung jawab dalam pembuatan rute terpendek dalam pengiriman unit, rute terpendek ialah informasi dari jalur yang harus dilewati pada saat pengiriman dilakukan. Staf <i>delivery</i> diharuskan melakukan <i>login</i> untuk dapat mengakses kegiatan tersebut.
4.	Kepala Gudang	Kepala gudang merupakan aktor yang bertanggung jawab dalam mengelola data produk, data distributor dan data <i>user/karyawan</i> pada Departemen <i>Logistic 2W</i> . Selain itu kepala gudang juga menerima laporan gudang setiap harinya sebagai arsip. Untuk dapat melakukan kegiatan tersebut kepala gudang diharuskan <i>login</i> .

Sumber: Hasil Analisis (2020)

## 2. Login

Tabel V.4 Use Case Description Login

Nama Use Case	Login
Aktor	Adm. Surat Jalan, Staf <i>Delivery</i> , Adm. Gudang, dan Kepala Gudang.
Deskripsi	Use case ini menggambarkan setiap aktor dapat masuk ke dalam sistem dengan hak akses yang dimiliki
Normal Flow of Events	<ol style="list-style-type: none"> <li>1. User membuka aplikasi</li> <li>2. Sistem akan menampilkan halaman login</li> <li>3. User memasukkan <i>username</i> dan <i>password</i> pada form login</li> <li>4. Sistem akan melakukan validasi data yang dimasukkan oleh user</li> <li>5. Jika data valid, maka user akan masuk ke halaman utama</li> <li>6. Jika data tidak valid, maka sistem akan menampilkan pesan kesalahan</li> </ol>

Sumber: Hasil Analisis (2020)

## 3. Mengelola Data User

Tabel V.5 Use Case Mengelola Data User

Nama Use Case	Mengelola Data User
Aktor	Kepala Gudang
Deskripsi	Use case ini menggambarkan aktivitas untuk mengelola data user yang meliputi <i>create</i> , <i>update</i> , <i>delete</i> yang dilakukan oleh kepala gudang
Normal Flow of Events	<ol style="list-style-type: none"> <li>1. User membuka menu user</li> <li>2. Sistem akan menampilkan semua informasi data user yang sudah tersimpan pada database</li> </ol>

Tabel V.5 *Use Case* Mengelola Data *User* (lanjutan)

<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> <li>3. <i>User</i> dapat menambah data dengan memilih tombol <i>create</i></li> <li>4. Sistem akan menampilkan <i>form create user</i>, kemudian <i>user</i> mengisi <i>form</i> tersebut dan memilih tombol <i>create</i>. Sistem akan menyimpan data tersebut ke <i>database</i></li> <li>5. <i>User</i> dapat mengubah data dengan memilih tombol <i>update</i></li> <li>6. Sistem akan menampilkan <i>form update</i>, kemudian <i>user</i> mengisi <i>form</i> tersebut dan memilih tombol <i>update</i>. Sistem akan menyimpan data tersebut ke <i>database</i></li> <li>7. <i>User</i> dapat menghapus data dengan memilih tombol <i>delete</i>. Sistem akan menghapus data yang dipilih.</li> <li>8. Sistem akan meng-<i>update database</i></li> </ol>
------------------------------	--

Sumber: Hasil Analisis (2020)

## 4. Mengelola Data Distributor

Tabel V.6 *Use Case* Mengelola Data Distributor

Nama <i>Use Case</i>	Mengelola Data Distributor
Aktor	Kepala Gudang
Deskripsi	<i>Use case</i> ini menggambarkan aktivitas untuk mengelola data distributor yang meliputi <i>create</i> , <i>update</i> , <i>delete</i> yang dilakukan oleh kepala gudang
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> memilih menu data distributor</li> <li>2. Sistem akan menampilkan semua informasi data distributor yang sudah tersimpan pada <i>database</i></li> <li>3. <i>User</i> dapat menambah data dengan memilih <i>create</i></li> </ol>

Tabel V.6 *Use Case* Mengelola Data Distributor (lanjutan).

<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> <li>4. Sistem akan menampilkan <i>form add</i> distributor, kemudian <i>user</i> mengisi <i>form</i> tersebut dan memilih tombol <i>create</i>. Sistem akan menyimpan data tersebut ke <i>database</i></li> <li>5. <i>User</i> dapat mengubah data dengan memilih tombol <i>update</i></li> <li>6. Sistem akan menampilkan <i>form edit</i>, kemudian <i>user</i> mengisi <i>form</i> tersebut dan memilih tombol <i>update</i>. Sistem akan menyimpan data tersebut ke <i>database</i></li> <li>7. <i>User</i> dapat menghapus data dengan memilih tombol <i>delete</i>. Sistem akan menghapus data yang dipilih.</li> <li>8. Sistem akan meng-<i>update database</i></li> </ol>
------------------------------	--

Sumber: Hasil Analisis (2020)

## 5. Mengelola Data Produk

Tabel V.7 *Use Case* Mengelola Data Produk

Nama <i>Use Case</i>	Mengelola Data Produk
Aktor	Kepala Gudang
Deskripsi	<i>Use case</i> ini menggambarkan aktivitas untuk mengelola data produk yang meliputi <i>create</i> , <i>update</i> , <i>delete</i> yang dilakukan oleh kepala gudang
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> membuka menu item</li> <li>2. Sistem akan menampilkan semua informasi data produk yang sudah tersimpan pada <i>database</i></li> <li>3. <i>User</i> dapat menambah data dengan memilih <i>create product items</i></li> </ol>

Tabel V.7 *Use Case* Mengelola Data Produk (lanjutan).

<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> <li>4. Sistem akan menampilkan <i>form add items</i>, kemudian <i>user</i> mengisi <i>form</i> tersebut dan memilih tombol <i>create</i>. Sistem akan menyimpan data tersebut ke <i>database</i></li> <li>5. <i>User</i> dapat mengubah data dengan memilih tombol <i>update</i></li> <li>6. Sistem akan menampilkan <i>form update</i>, kemudian <i>user</i> mengisi <i>form</i> tersebut dan memilih tombol <i>update</i>. Sistem akan menyimpan data tersebut ke <i>database</i></li> <li>7. <i>User</i> dapat menghapus data dengan memilih tombol <i>delete</i>. Sistem akan menghapus data yang dipilih.</li> <li>8. Sistem akan meng-<i>update database</i></li> </ol>
------------------------------	--

Sumber: Hasil Analisis (2020)

## 6. Mengisi Stok Ketersediaan

Tabel V.8 *Use Case* Mengisi Stok Ketersediaan

Nama <i>Use Case</i>	Mengisi Stok Ketersediaan
Aktor	Adm. Gudang
Deskripsi	<i>Use case</i> ini menggambarkan aktivitas mengisi stok ketersediaan unit pada gudang oleh admin gudang
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> akan membuka menu gudang</li> <li>2. Sistem akan menampilkan semua data stok pada gudang yang sudah tersimpan pada <i>database</i></li> <li>3. <i>User</i> dapat menambahkan stok pada gudang dengan memilih tombol <i>create</i></li> <li>4. Sistem akan menampilkan <i>form</i> input stok kemudian <i>user</i> mengisi <i>form</i> tersebut</li> </ol>

Tabel V.8 *Use Case* Mengisi Stok Ketersediaan (lanjutan).

<i>Normal Flow of Events</i>	<p>dan selanjutnya memilih tombol simpan. Sistem akan menyimpan ke dalam <i>database</i></p> <ol style="list-style-type: none"> <li>5. <i>User</i> dapat melihat detail dengan memilih tombol detail</li> <li>6. Sistem akan menampilkan detail data</li> <li>7. <i>User</i> dapat melakukan hapus data dengan memilih tombol <i>delete</i>. Kemudian sistem akan menghapus data yang dipilih</li> <li>8. Sistem akan meng-<i>update database</i></li> </ol>
------------------------------	--

Sumber: Hasil Analisis (2020)

## 7. Melihat Notifikasi Ketersediaan

Tabel V.9 *Use Case* Melihat Notifikasi Ketersediaan

Nama <i>Use Case</i>	Melihat Notifikasi Ketersediaan
Aktor	Adm. Surat Jalan
Deskripsi	<i>Use case</i> ini menggambarkan aktivitas yang dilakukan oleh admin surat jalan yaitu melihat notifikasi ketersediaan unit yang masuk pada sistem dengan otomatis.
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> membuka halaman utama pada sistem</li> <li>2. Sistem akan menampilkan tampilan awal dan notifikasi yang terlihat di-<i>header</i></li> <li>3. <i>User</i> kemudian memilih tombol notifikasi tersebut</li> <li>4. Sistem akan menampilkan notifikasi masuk, berupa informasi ketersediaan unit yang sudah diisi oleh aktor sebelumnya yaitu admin gudang</li> </ol>

Sumber: Hasil Analisis (2020)

## 8. Melihat Data DO

Tabel V.10 *Use Case* Melihat Data DO

Nama <i>Use Case</i>	Melihat Data DO
Aktor	Adm. Surat Jalan
Deskripsi	<i>Use case</i> ini menggambarkan aktivitas yang dilakukan oleh admin surat jalan untuk melihat DO ( <i>delivery order</i> ) yang masuk.
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> akan membuka menu <i>delivery order</i></li> <li>2. Sistem akan menampilkan semua data DO yang masuk</li> <li>3. <i>User</i> dapat melihat detail dari DO dengan memilih tombol detail</li> <li>4. Sistem akan menampilkan detail data DO</li> </ol>

Sumber: Hasil Analisis (2020)

## 9. Membuat Surat Jalan

Tabel V.11 *Use Case* Membuat Surat Jalan

Nama <i>Use Case</i>	Membuat Surat Jalan
Aktor	Adm. Surat Jalan
Deskripsi	<i>Use case</i> ini menggambarkan aktivitas yang dilakukan oleh admin surat jalan untuk membuat surat jalan.
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> akan membuka menu <i>delivery order</i></li> <li>2. Sistem akan menampilkan semua data DO</li> <li>3. <i>User</i> dapat membuat surat jalan dengan langkah awal memilih tombol <i>view DO</i>.</li> <li>4. Sistem akan menampilkan data dari setiap DO yang telah masuk pada masing-masing distributor yang dipilih.</li> </ol>

Tabel V.11 *Use Case* Membuat Surat Jalan (lanjutan).

<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> <li>5. <i>User</i> selanjutnya dapat memilih tombol <i>create S.J</i> untuk membuat surat jalan</li> <li>6. Sistem akan <i>men-download</i> dokumen surat jalan dengan otomatis.</li> </ol>
------------------------------	--

Sumber: Hasil Analisis (2020)

## 10. Membuat Jalur Distribusi

Tabel V.12 *Use Case* Membuat Jalur Distribusi

Nama <i>Use Case</i>	Membuat Jalur Distribusi
Aktor	Staf <i>Delivery</i>
Deskripsi	<i>Use case</i> ini menggambarkan aktivitas dari aktor untuk membuat jalur distribusi dalam pengantaran unit
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> memilih menu rute terpendek</li> <li>2. Sistem akan menampilkan seluruh data rute yang telah tersimpan di dalam <i>database</i></li> <li>3. <i>User</i> dapat membuat rute dengan memilih tombol <i>create</i> rute. Sistem akan menampilkan <i>form create</i> rute kemudian <i>user</i> mengisi <i>form</i> tersebut dan memilih tombol simpan.</li> <li>4. Sistem akan menampilkan seluruh rute pengiriman</li> <li>5. <i>User</i> dapat memilih tombol rute tercepat</li> <li>6. Sistem akan menghitung rute dan akan menampilkan rute terpendk</li> <li>7. <i>User</i> dapat memilih tombol <i>download</i> rute</li> <li>8. Sistem akan menampilkan lokasi penyimpanan untuk <i>men-download</i> dokumen rute terpendek</li> </ol>

Sumber: Hasil Analisis (2020)

## 11. Melihat Notifikasi SJ

Tabel V.13 *Use Case* Melihat Notifikasi SJ

Nama <i>Use Case</i>	Melihat Notifikasi SJ
Aktor	Staf <i>Delivery</i>
Deskripsi	<i>Use case</i> ini menggambarkan aktivitas yang dilakukan oleh staf <i>delivery</i> yaitu, melihat notifikasi SJ yang diterima dengan otomatis
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> membuka halaman utama pada sistem</li> <li>2. Sistem akan menampilkan tampilan awal dan notifikasi yang terlihat di-<i>header</i></li> <li>3. <i>User</i> kemudian memilih tombol notifikasi tersebut</li> <li>4. Sistem akan menampilkan notifikasi masuk, berupa informasi ketersediaan unit yang sudah diisi oleh aktor sebelumnya yaitu admin surat jalan</li> </ol>

Sumber: Hasil Analisis (2020)

## 12. Melihat Laporan Gudang

Tabel V.14 *Use Case* Melihat Laporan Gudang

Nama <i>Use Case</i>	Melihat Laporan Gudang
Aktor	Adm. Gudang dan Kepala Gudang
Deskripsi	<i>Use case</i> ini menggambarkan aktivitas melihat laporan gudang oleh admin gudang
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> akan membuka menu gudang</li> <li>2. Sistem akan menampilkan data laporan ketersediaan unit pada gudang <i>update</i> otomatis dari setiap unit yang keluar</li> <li>3. <i>User</i> dapat memilih tombol detail</li> </ol>

Tabel V.14 *Use Case* Melihat Laporan Gudang (lanjutan)

<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> <li>4. Sistem akan menampilkan detail laporan</li> <li>5. <i>User</i> juga dapat memilih tombol simpan dokumen</li> <li>6. Sistem akan menampilkan lokasi penyimpanan untuk <i>men-download</i> dokumen.</li> </ol>
------------------------------	--

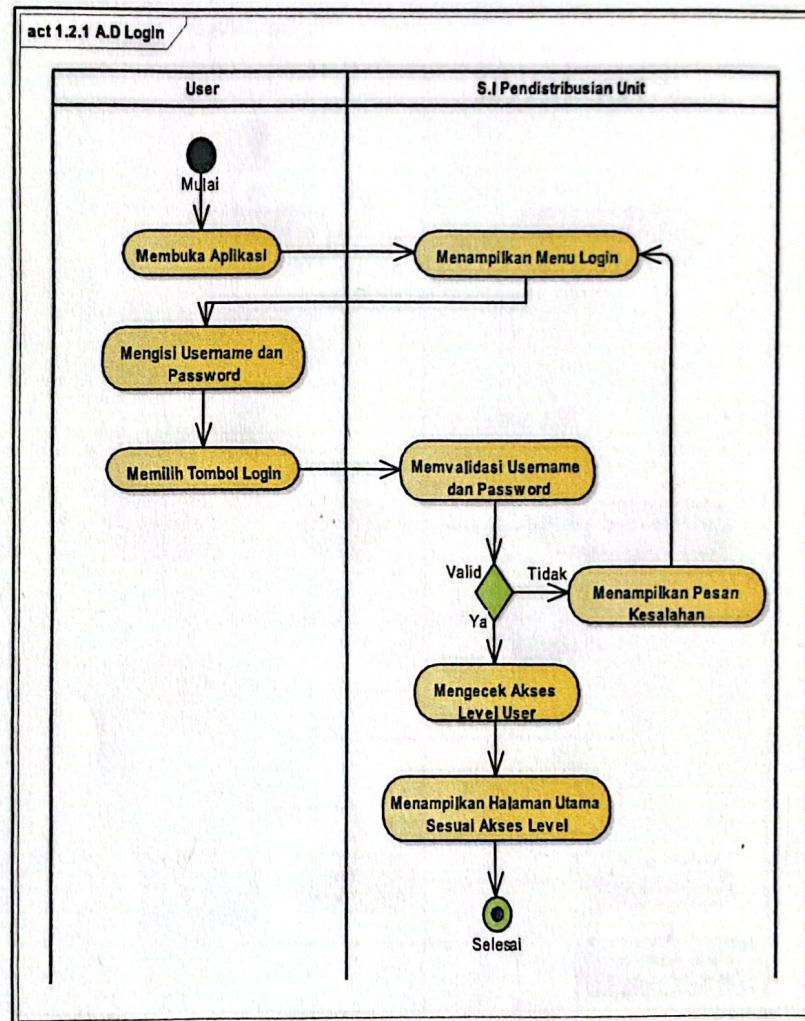
Sumber: Hasil Analisis (2020)

### 5.5 *Activity Diagram* Sistem Informasi Pendistribusian Unit

*Activity diagram* merupakan sebuah diagram yang menggambarkan urutan aktivitas proses bisnis pada suatu sistem. Berikut merupakan beberapa *activity diagram* usulan dari sistem informasi pendistribusian unit:

#### 1. *Activity Diagram Login*

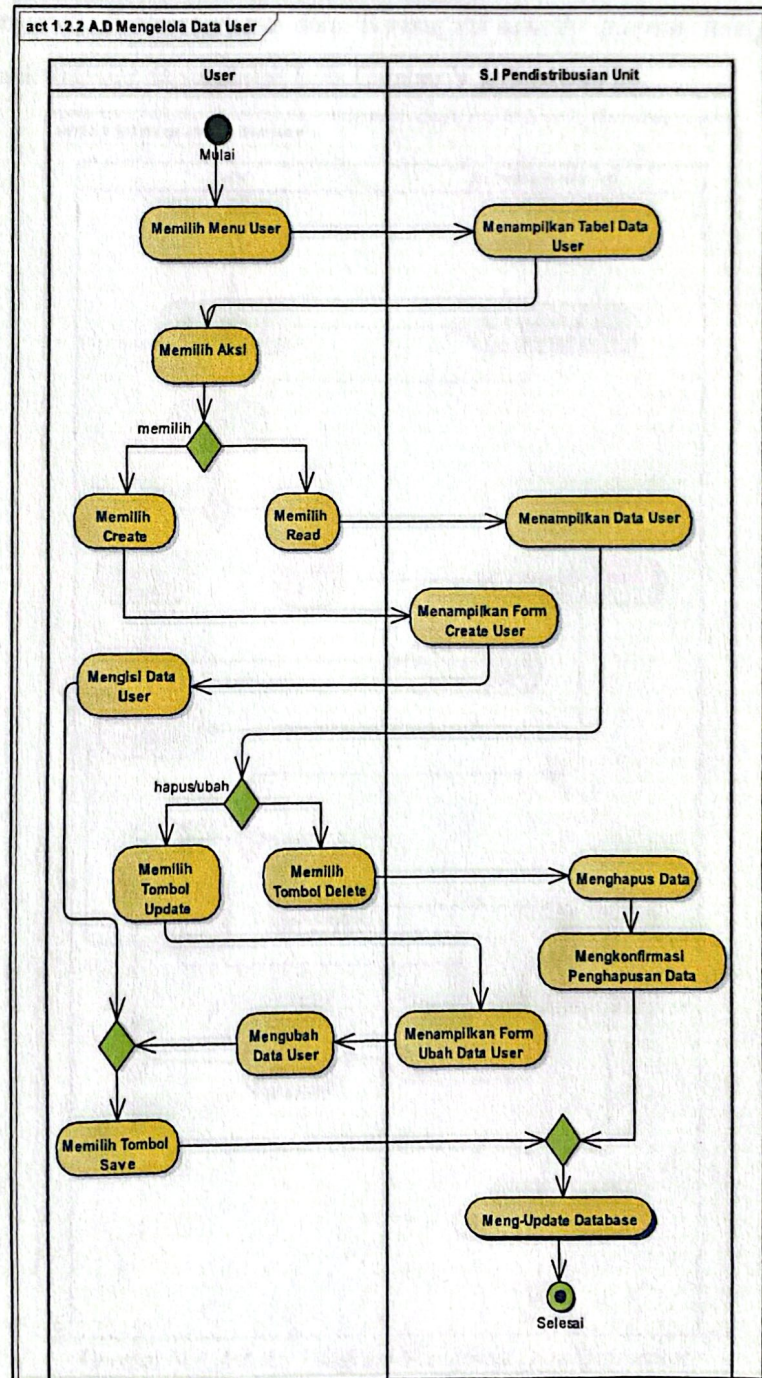
*Activity diagram* ini menjelaskan mengenai alur aktivitas yang dilakukan oleh admin gudang, admin surat jalan, staf *delivery* dan kepala gudang untuk mendapatkan akses masuk ke dalam sistem informasi pendistribusian unit. Berikut merupakan *activity diagram login*:



Gambar V.7 Activity Diagram Login  
Sumber: Hasil Analisis (2020)

## 2. Activity Diagram Mengelola Data User

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh kepala gudang dalam mengelola data user yaitu dapat menambahkan, mengubah, dan menghapus data. Berikut ini activity diagram mengelola data user dapat dilihat pada Gambar V.8 di bawah ini:

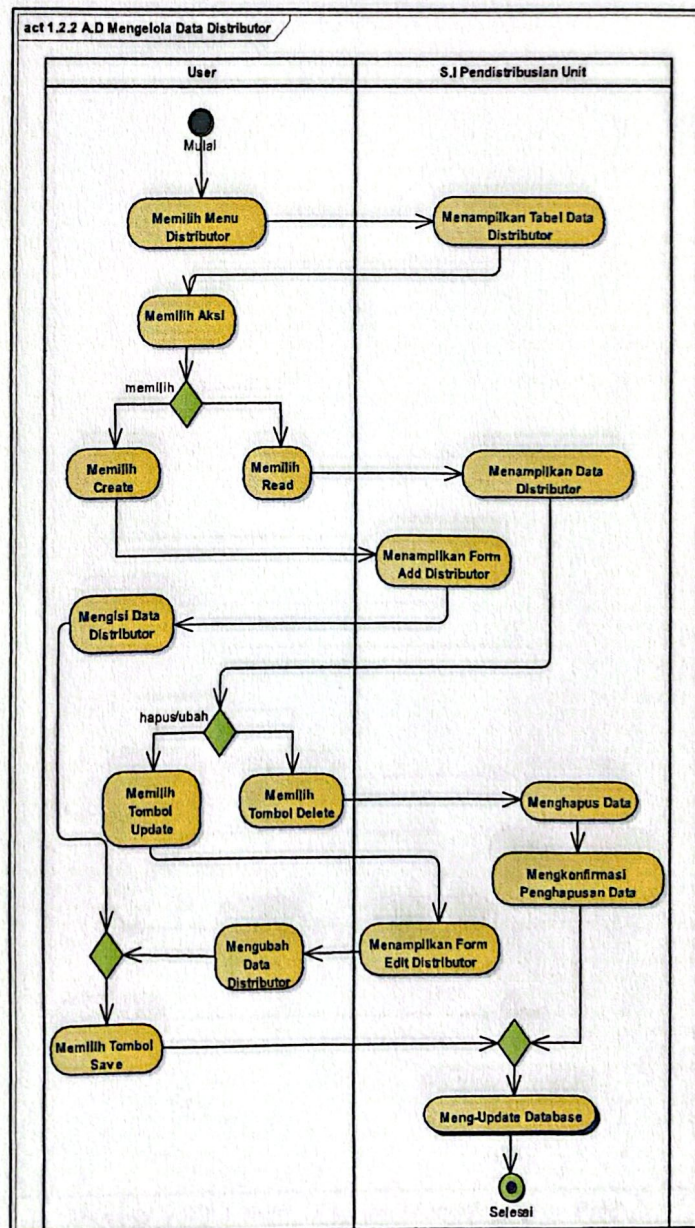


Gambar V.8 Activity Diagram Mengelola Data User  
Sumber: Hasil Analisis (2020)

### 3. Activity Diagram Mengelola Data Distributor

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh kepala gudang dalam mengelola data distributor yaitu dapat menambahkan,

mengubah, dan menghapus data. Berikut ini *activity diagram* mengelola data distributor dapat dilihat pada Gambar V.9 di bawah ini:

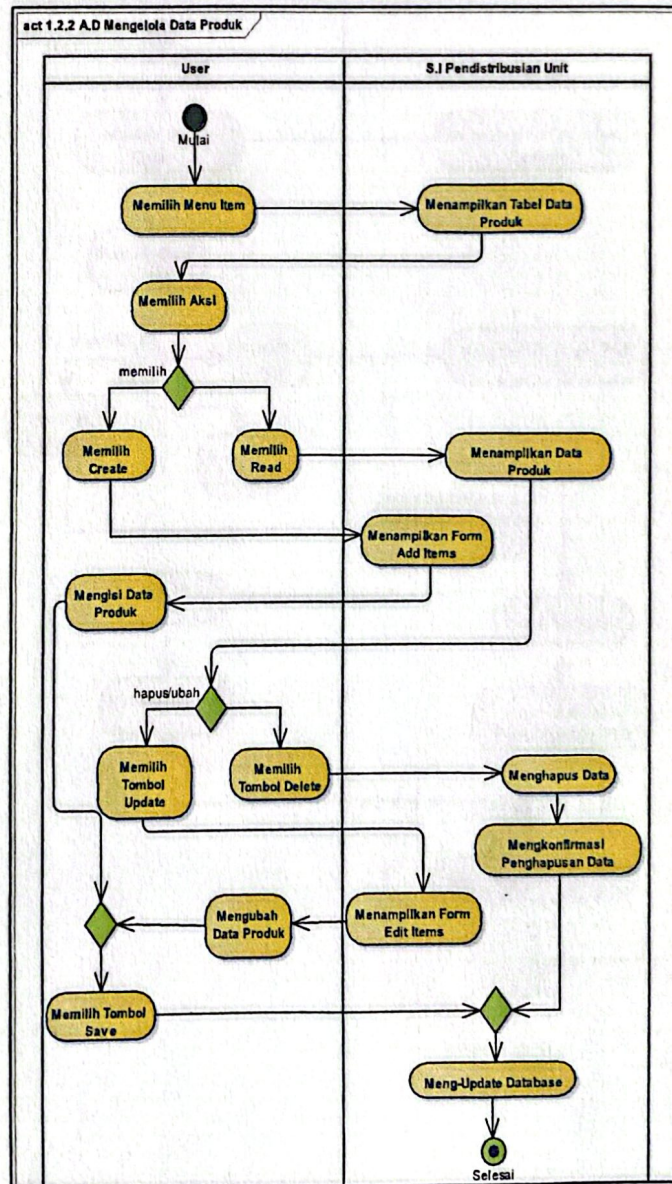


Gambar V.9 *Activity Diagram* Mengelola Data Distributor  
Sumber: Hasil Analisis (2020)

#### 4. *Activity Diagram* Mengelola Data Produk

*Activity diagram* ini menjelaskan alur aktivitas yang dilakukan oleh kepala gudang dalam mengelola data produk yaitu dapat menambahkan,

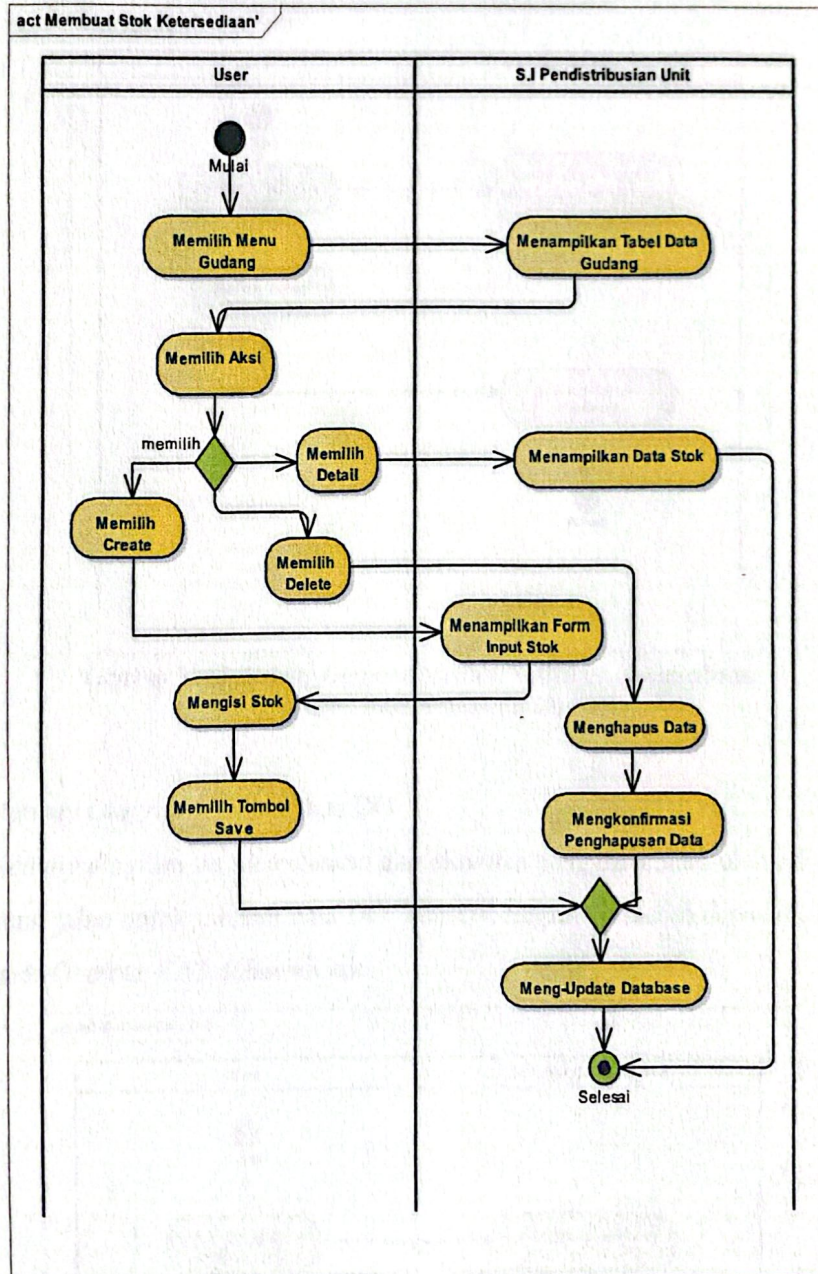
mengubah, dan menghapus data. Berikut ini *activity diagram* mengelola data produk dapat dilihat pada Gambar V.10 di bawah ini:



Gambar V.10 *Activity Diagram* Mengelola Data Produk  
Sumber: Hasil Analisis (2020)

##### 5. *Activity Diagram* Mengisi Stok Ketersediaan

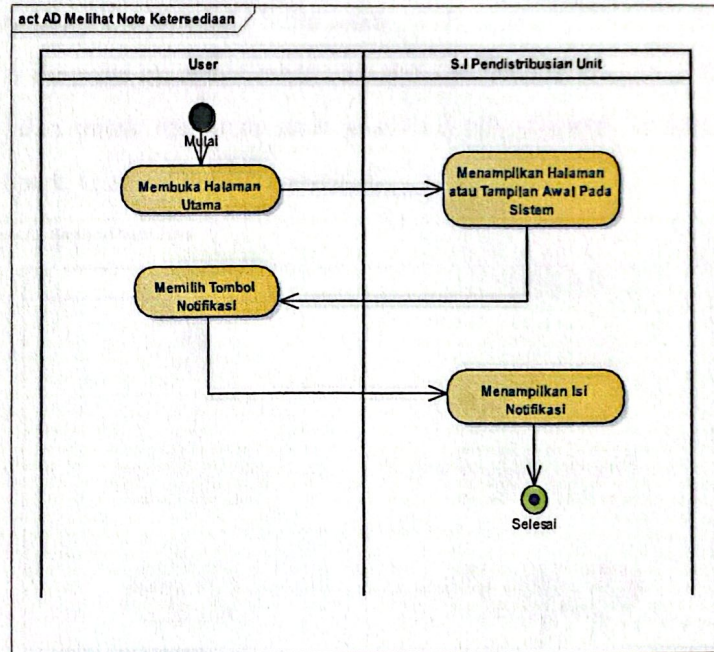
*Activity diagram* ini menjelaskan alur aktivitas yang dilakukan oleh admin gudang untuk mengisi stok ketersediaan. *Activity diagram* tersebut dapat dilihat pada Gambar V.11 di bawah ini:



Gambar V.11 Activity Diagram Mengisi Stok Ketersediaan  
Sumber: Hasil Analisis (2020)

#### 6. Activity Diagram Melihat Notifikasi Ketersediaan

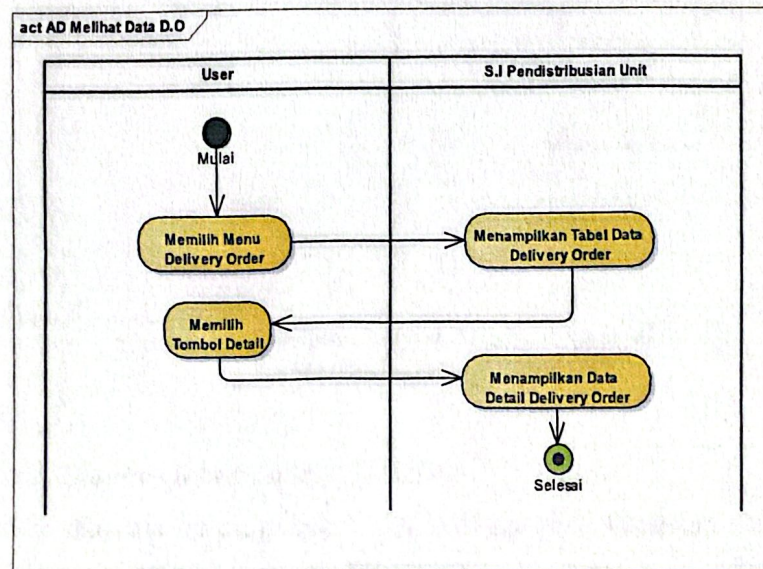
Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh admin surat jalan untuk melihat notifikasi ketersediaan. Activity diagram tersebut dapat dilihat pada Gambar V.12 di bawah ini:



Gambar V.12 Activity Diagram Melihat Notifikasi Ketersediaan  
Sumber: Hasil Analisis (2020)

#### 7. Activity Diagram Melihat Data DO

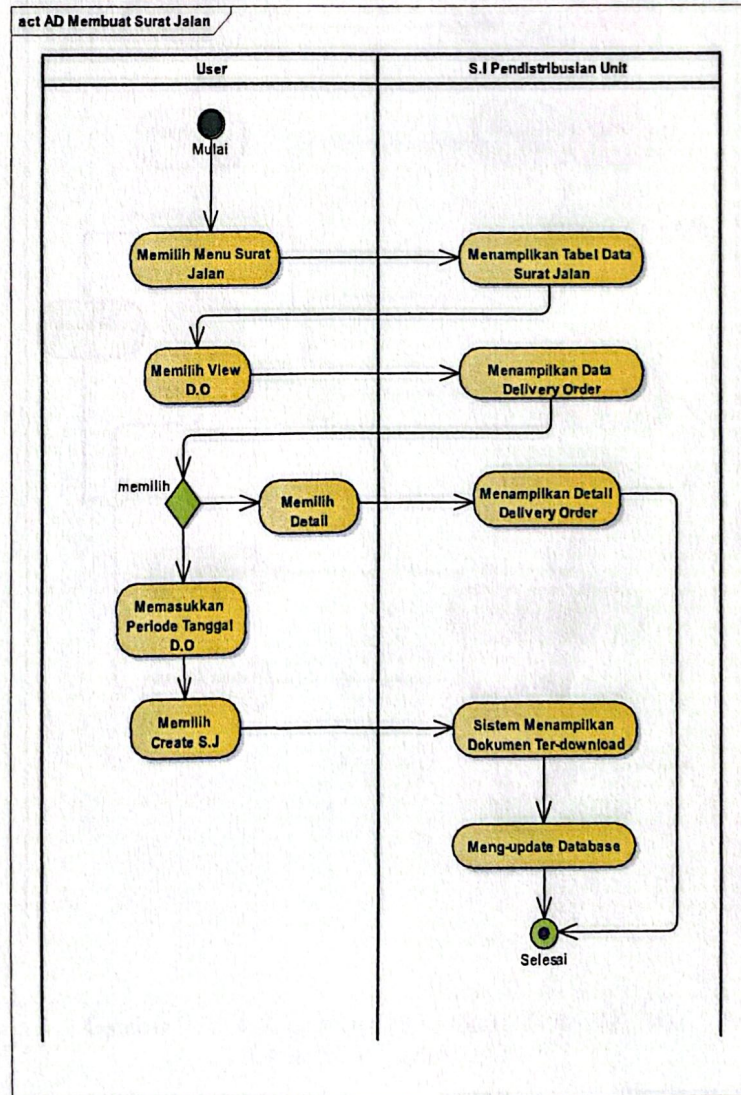
Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh admin surat jalan untuk melihat data DO. Activity diagram tersebut dapat dilihat pada Gambar V.13 di bawah ini:



Gambar V.13 Activity Diagram Melihat Data DO  
Sumber: Hasil Analisis (2020)

### 8. Activity Diagram Membuat Surat Jalan

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh admin surat jalan untuk membuat surat jalan. Activity diagram tersebut dapat dilihat pada Gambar V.14 di bawah ini:

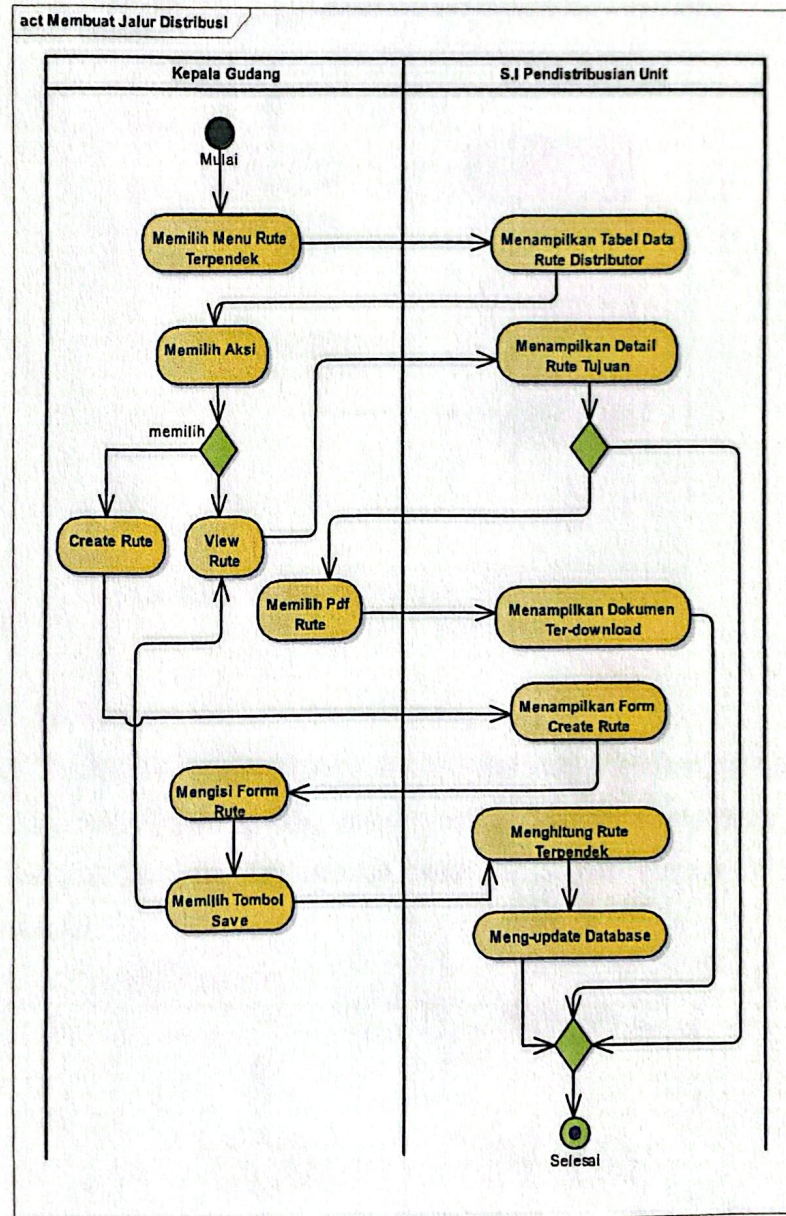


Gambar V.14 Activity Diagram Membuat Surat Jalan

Sumber: Hasil Analisis (2020)

### 9. Activity Diagram Membuat Jalur Distribusi

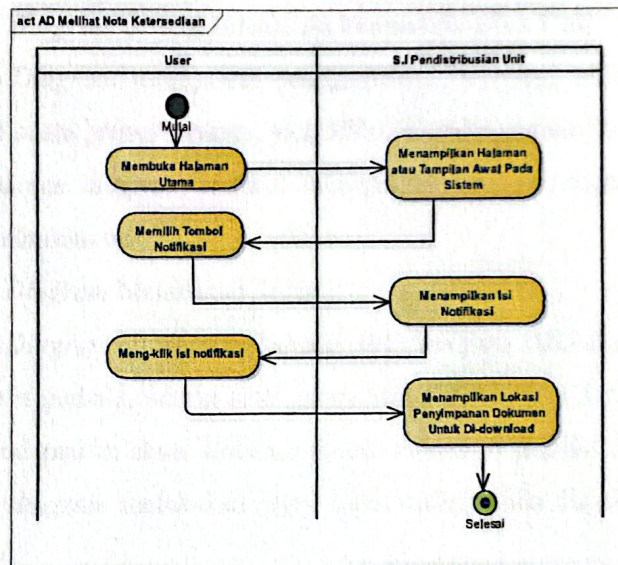
Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh staf delivery untuk membuat jalur distribusi. Activity diagram tersebut dapat dilihat pada Gambar V.15 di bawah ini:



Gambar V.15 Activity Diagram Membuat Jalur Distribusi  
Sumber: Hasil Analisis (2020)

#### 10. Activity Diagram Melihat Notifikasi SJ

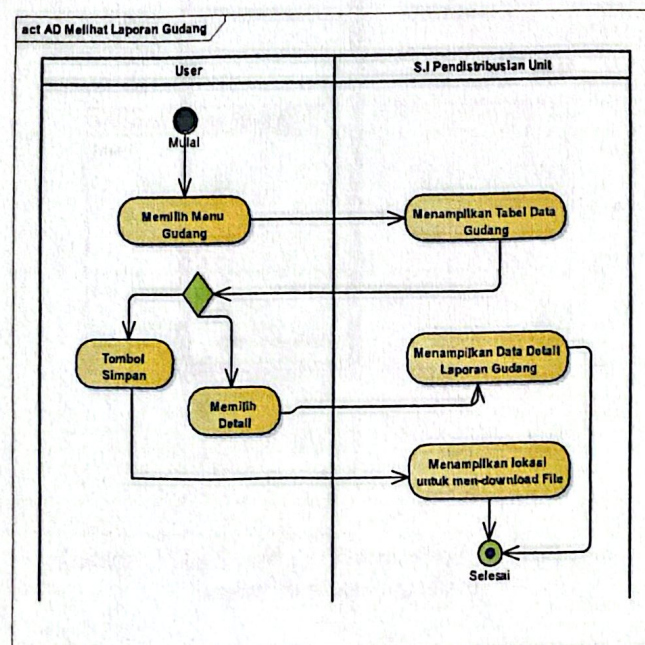
Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh staf *delivery* untuk melihat notifikasi SJ yang masuk dengan otomatis. Activity diagram tersebut dapat dilihat pada Gambar V.16 di bawah ini:



Gambar V.16 *Activity Diagram* Melihat Notifikasi SJ  
Sumber: Hasil Analisis (2020)

### 11. *Activity Diagram* Melihat Laporan Gudang

*Activity diagram* ini menjelaskan alur aktivitas yang dilakukan oleh admin gudang dan kepala gudang untuk melihat laporan ketersediaan unit gudang. *Activity diagram* tersebut dapat dilihat pada Gambar V.17 di bawah ini:



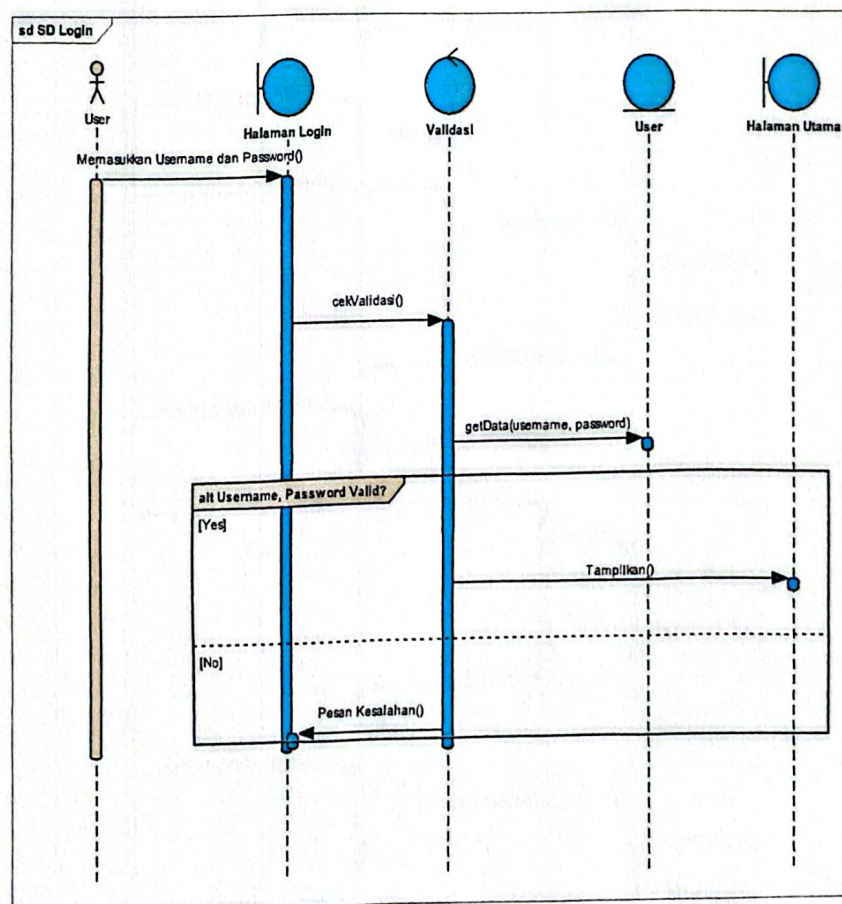
Gambar V.17 *Activity Diagram* Melihat Laporan Gudang  
Sumber: Hasil Analisis (2020)

## 5.6 Sequence Diagram Sistem Informasi Pendistribusian Unit

*Sequence Diagram* merupakan penggambaran interaksi terhadap objek ketika melakukan suatu proses tertentu, yang dimana urutan proses tersebut dapat dilihat pada *sequence diagram*. Berikut merupakan *sequence diagram* sistem informasi pendistribusian unit:

### 1. Sequence Diagram Melakukan Login

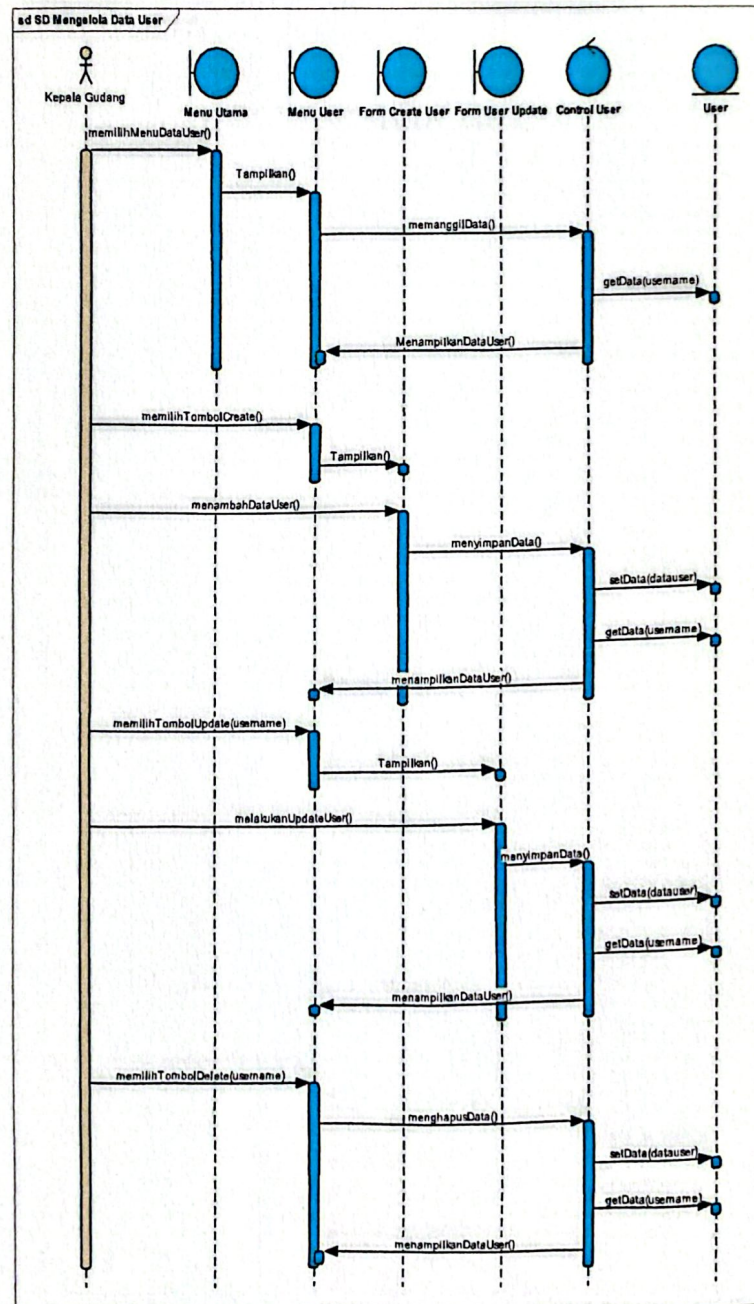
*Sequence Diagram* ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu admin gudang, admin surat jalan, staf *delivery* dan kepala gudang untuk mendapatkan akses kedalam sistem informasi pendistribusian unit. *Sequence diagram* melakukan *login* dapat dilihat pada Gambar V.18 di bawah ini:



Gambar V.18 Sequence Diagram Melakukan Login  
Sumber: Hasil Analisis (2020)

## 2. Sequence Diagram Mengelola Data User

Sequence diagram ini menjelaskan alur aktivitas yang dilakukan oleh kepala gudang dalam mengelola data user dapat mengubah, menghapus maupun menambah data. Berikut Gambar V.19 merupakan sequence diagram mengelola data user:

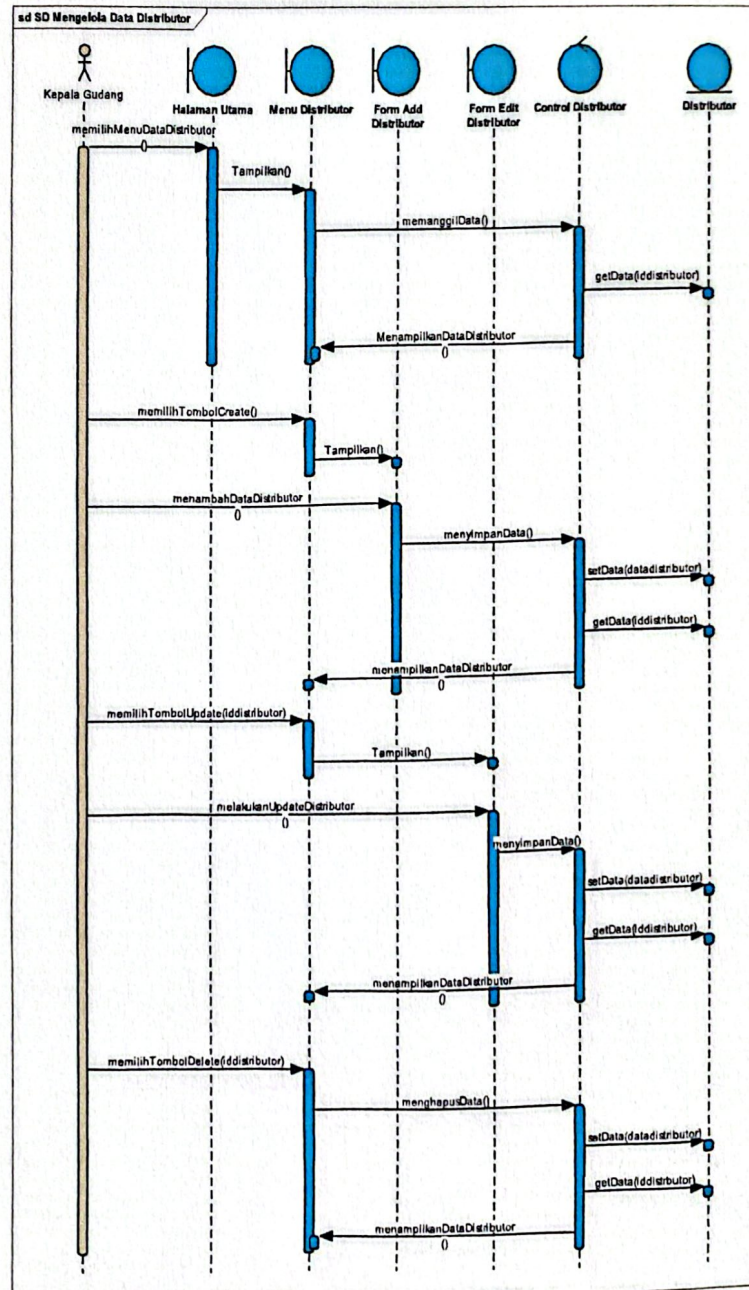


Gambar V.19 Sequence Diagram Mengelola Data User

Sumber: Hasil Analisis (2020)

### 3. Sequence Diagram Mengelola Data Distributor

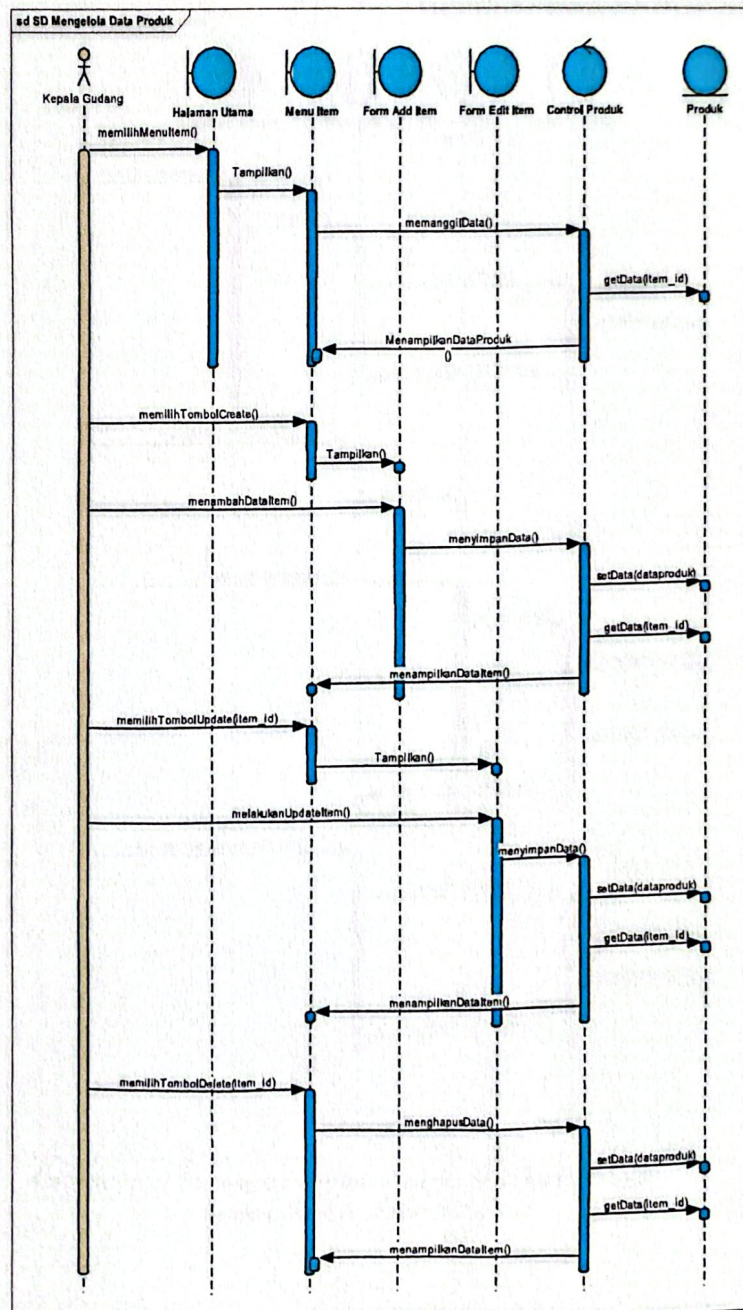
*Sequence diagram* ini menjelaskan alur aktivitas yang dilakukan oleh kepala gudang dalam mengelola data distributor dapat mengubah, menghapus maupun menambah data. Berikut Gambar V.20 merupakan *sequence diagram* mengelola data distributor:



Gambar V.20 *Sequence Diagram* Mengelola Data Distributor  
Sumber: Hasil Analisis (2020)

#### 4. Sequence Diagram Mengelola Data Produk

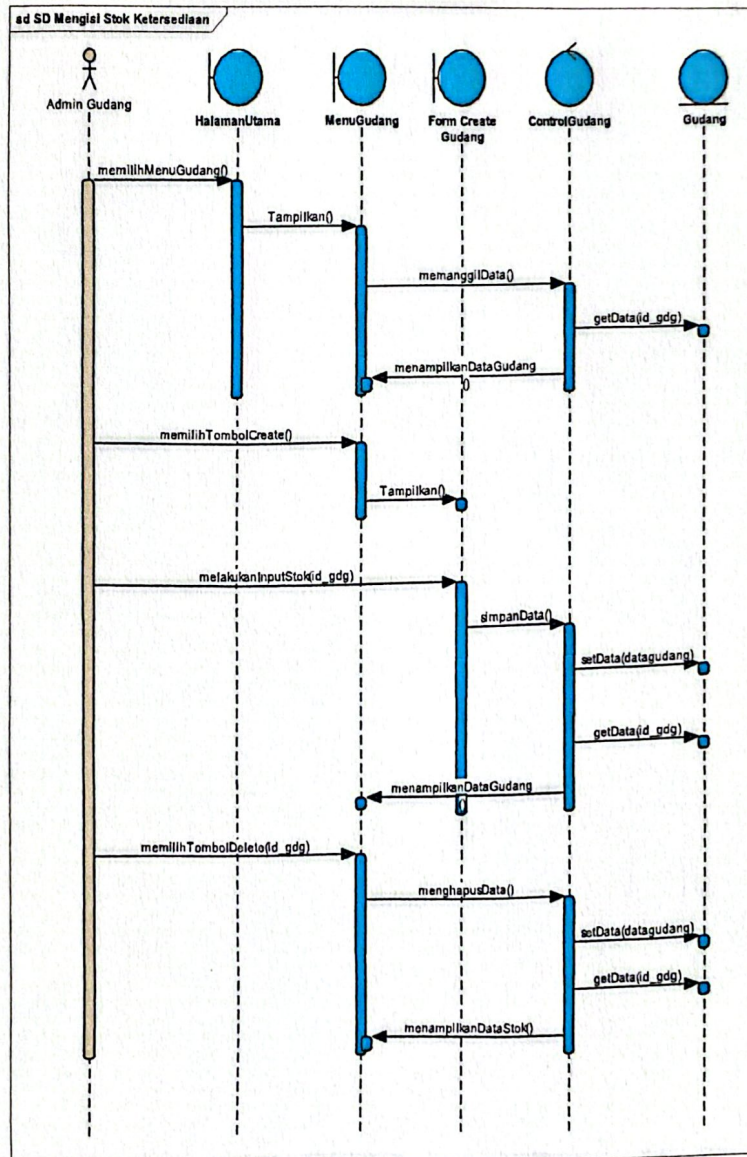
*Sequence diagram* ini menjelaskan alur aktivitas yang dilakukan oleh kepala gudang dalam mengelola data produk dapat mengubah, menghapus maupun menambah data. Berikut Gambar V.21 merupakan *sequence diagram* mengelola data produk:



Gambar V.21 *Sequence Diagram* Mengelola Data Produk  
Sumber: Hasil Analisis (2020)

### 5. Sequence Diagram Mengisi Stok Ketersediaan

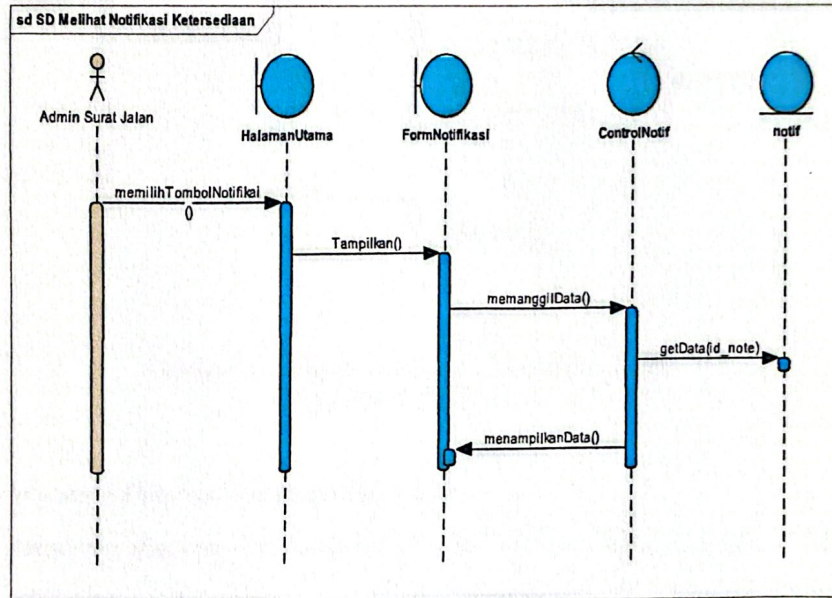
*Sequence diagram* ini menjelaskan alur aktivitas yang dilakukan oleh admin gudang dalam mengisi stok ketersediaan pada gudang. Berikut merupakan *sequence diagram* mengisi stok ketersediaan dapat dilihat pada Gambar V.22 di bawah ini:



Gambar V.22 *Sequence Diagram* Mengisi Stok Ketersediaan  
Sumber: Hasil Analisis (2020)

### 6. Sequence Diagram Melihat Notifikasi Ketersediaan

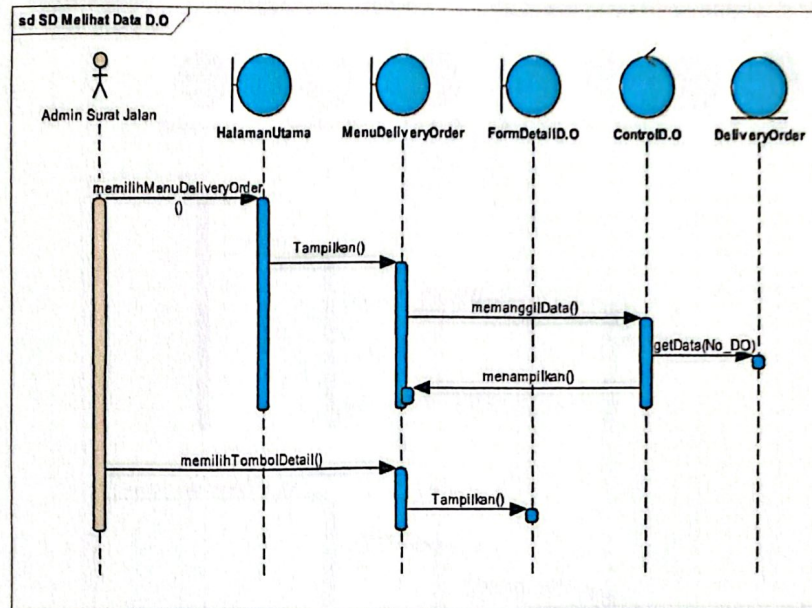
Sequence diagram ini menjelaskan alur aktivitas yang dilakukan oleh admin surat jalan dalam melihat notifikasi ketersediaan. Berikut merupakan sequence diagram melihat notifikasi ketersediaan dapat dilihat pada Gambar V.23 di bawah ini:



Gambar V.23 Sequence Diagram Melihat Notifikasi Ketersediaan  
Sumber: Hasil Analisis (2020)

### 7. Sequence Diagram Melihat Data DO

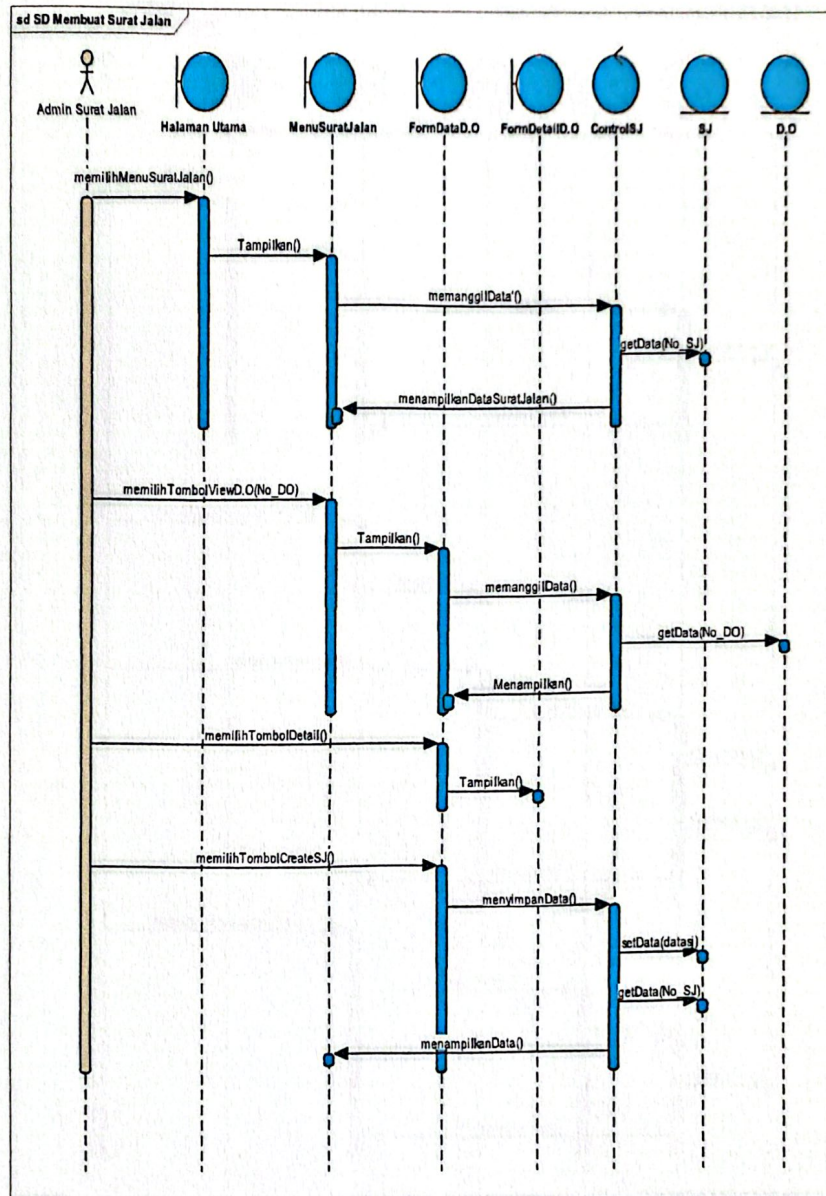
Sequence diagram ini menjelaskan alur aktivitas yang dilakukan oleh admin surat jalan dalam melihat data DO. Berikut merupakan sequence diagram melihat data DO dapat dilihat pada Gambar V.24 di bawah ini:



Gambar V.24 *Sequence Diagram* Melihat Data DO  
 Sumber: Hasil Analisis (2020)

#### 8. *Sequence Diagram* Membuat Surat Jalan

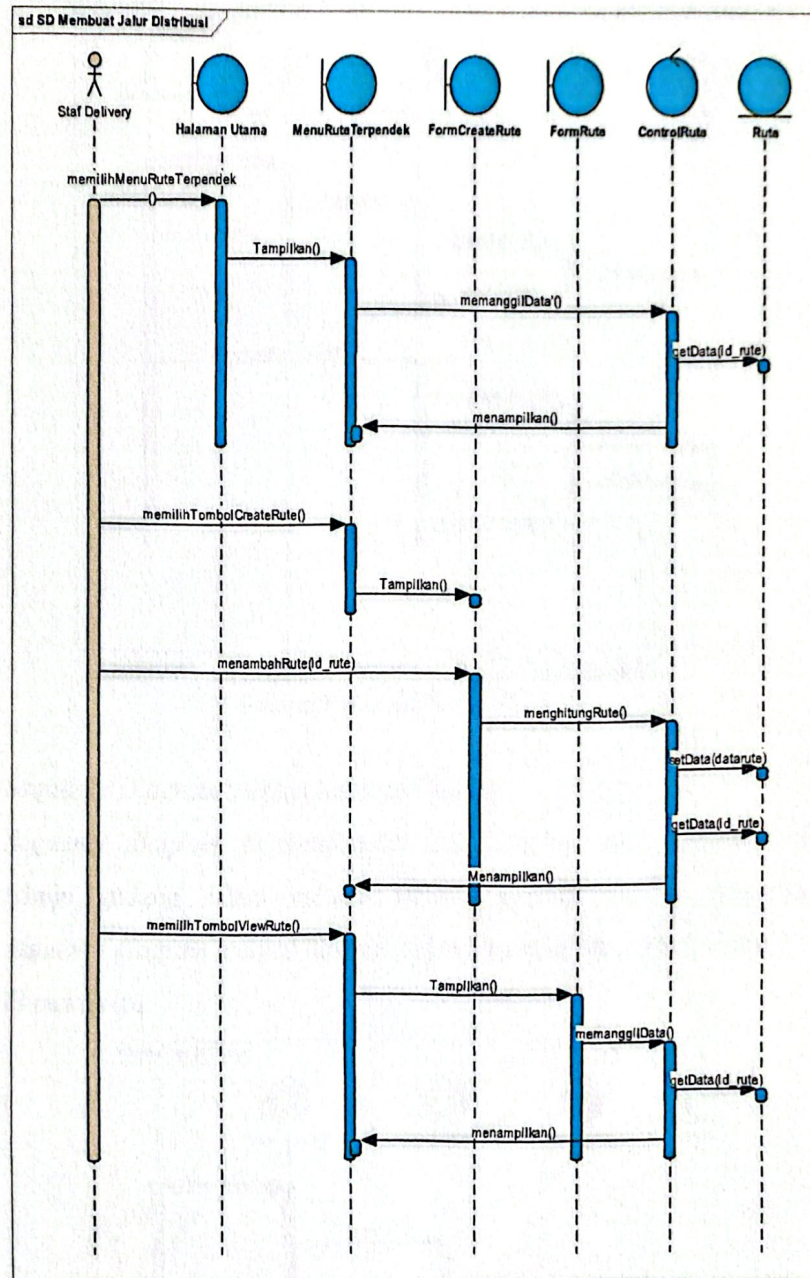
*Sequence diagram* ini menjelaskan alur aktivitas yang dilakukan oleh admin surat jalan dalam membuat surat jalan. Berikut merupakan *sequence diagram* membuat surat jalan dapat dilihat pada Gambar V.25 di bawah ini:



Gambar V.25 Sequence Diagram Membuat Surat Jalan  
Sumber: Hasil Analisis (2020)

### 9. Sequence Diagram Membuat Jalur Distribusi

Sequence diagram ini menjelaskan alur aktivitas yang dilakukan oleh staf *delivery* dalam membuat jalur distribusi dengan menentukan rute terpendek untuk pengiriman. Berikut merupakan sequence diagram membuat jalur distribusi dapat dilihat pada Gambar V.26 di bawah ini:

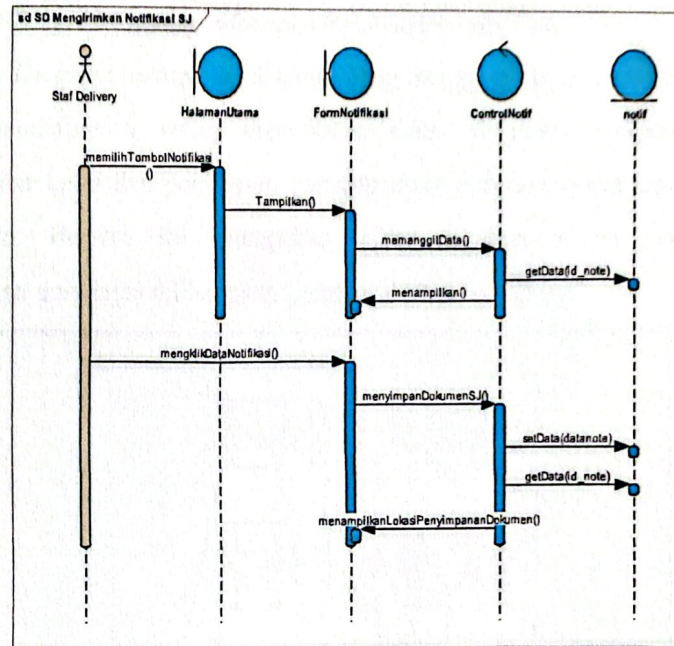


Gambar V.26 Sequence Diagram Membuat Jalur Distribusi

Sumber: Hasil Analisis (2020)

#### 10. Sequence Diagram Melihat Notifikasi SJ

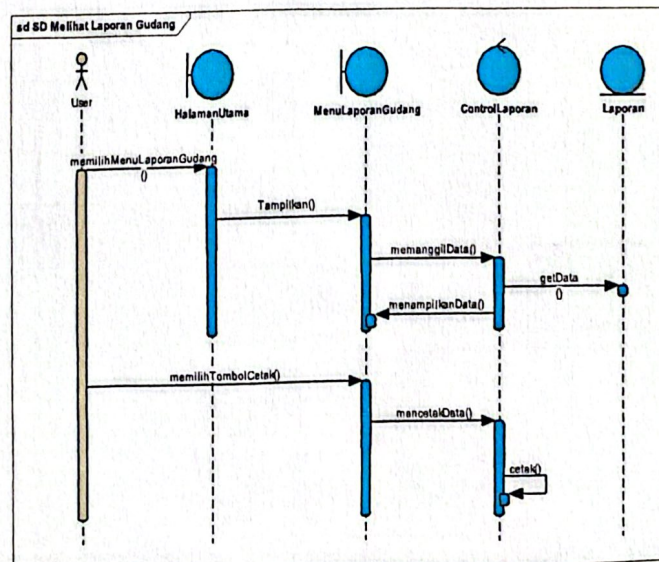
*Sequence diagram* ini menjelaskan alur aktivitas yang dilakukan oleh staf *delivery* dalam melihat notifikasi SJ dan juga dapat menyimpan dokumen. Berikut merupakan *sequence diagram* membuat jalur distribusi dapat dilihat pada Gambar V.27 di bawah ini:



Gambar V.27 Sequence Diagram Melihat Notifikasi SJ  
Sumber: Hasil Analisis (2020)

### 11. Sequence Diagram Melihat Laporan Gudang

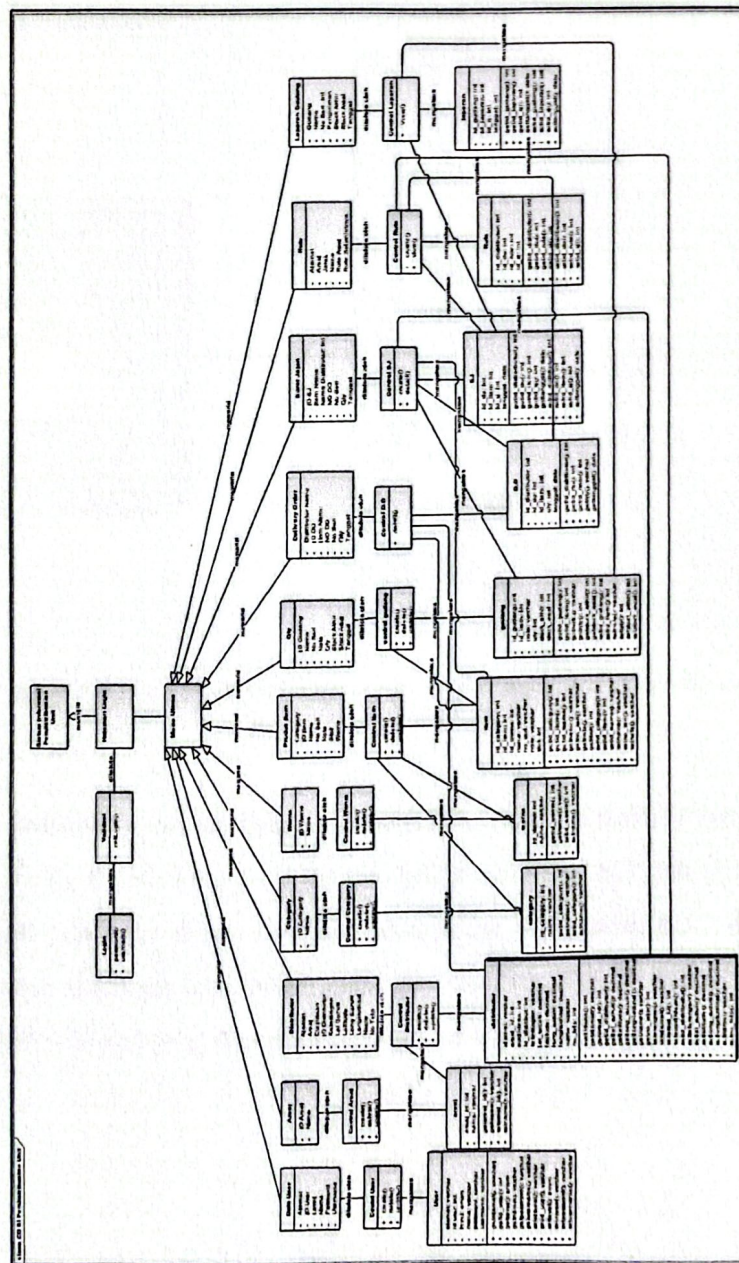
Sequence diagram ini menjelaskan alur aktivitas yang dilakukan oleh admin gudang dalam melihat laporan gudang. Berikut merupakan sequence diagram melihat laporan gudang dapat dilihat pada Gambar V.28 di bawah ini:



Gambar V.28 Sequence Diagram Melihat Laporan Gudang  
Sumber: Hasil Analisis (2020)

### 5.7 Class Diagram Sistem Informasi Pendistribusian Unit

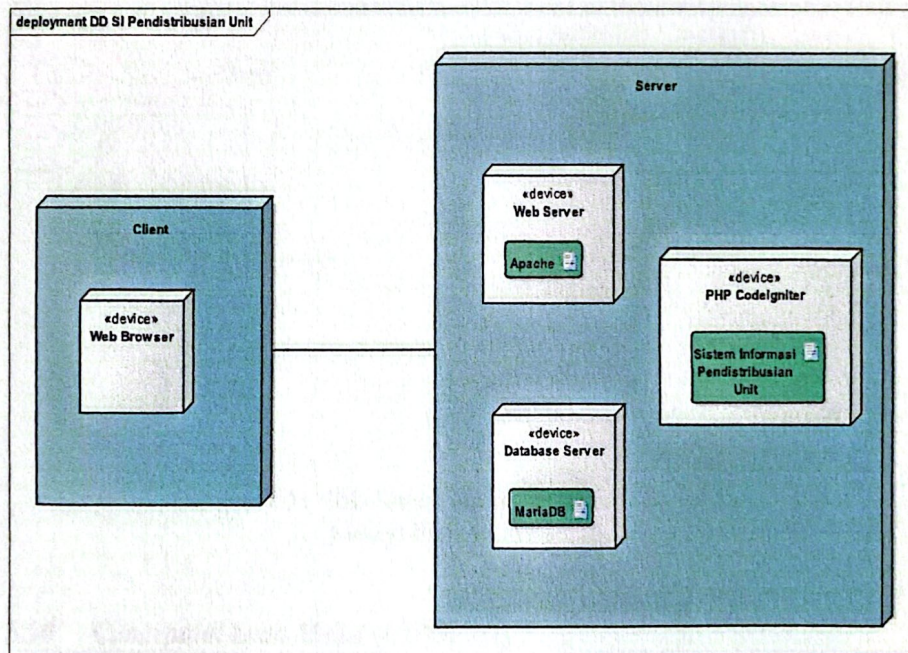
*Class Diagram* merupakan diagram yang menggambarkan struktur sistem dari segi pendefinisian setiap kelas-kelas. *Class diagram* memperlihatkan hubungan antar kelas dan penjelasan tiap-tiap kelas didalam model desain dari suatu sistem. Berikut ini merupakan *class diagram* sistem informasi pendistribusian unit dapat dilihat pada Gambar V.29 di bawah ini:



Gambar V.29 *Class Diagram* Sistem Informasi Pendistribusian Unit  
Sumber: Hasil Analisis (2020)

### 5.8 *Deployment Diagram Sistem Informasi Pendistribusian Unit*

*Deployment diagram* pada usulan sistem informasi pendistribusian unit digunakan untuk mewakili komponen-komponen *software* dan cara *software* ditempatkan pada arsitektur fisik atau infrastruktur sistem informasi. Berikut Gambar V.30 merupakan *deployment diagram* sistem informasi pendistribusian unit:

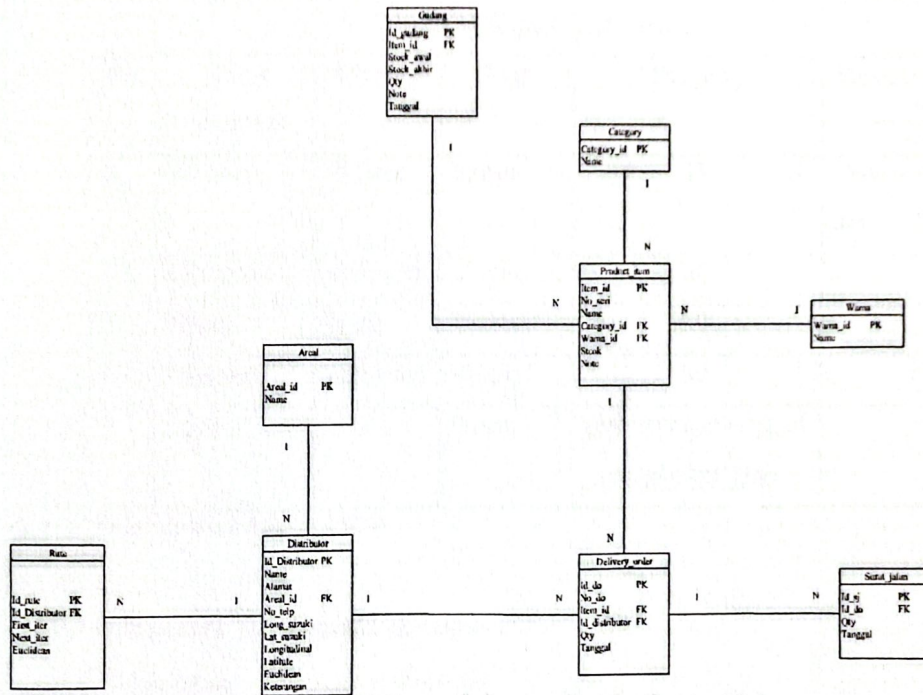


Gambar V.30 *Deployment Diagram* Sistem Informasi Pendistribusian Unit  
Sumber: Hasil Analisis (2020)

### 5.9 *Entity Relationship Diagram Sistem Informasi Pendistribusian Unit*

*Entity Relationship Diagram* merupakan suatu model untuk menjelaskan hubungan antar data dalam *database* berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. *Entity Relationship Diagram* sistem informasi pendistribusian unit yang diusulkan dapat dilihat pada Gambar V.31





Gambar V.32 CDM Sistem Informasi Pendistribusian Unit  
Sumber: Hasil Analisis (2020)

### 5.11 Kamus Data

Kamus menggambarkan sebuah katalog data yang terdapat di dalam sistem dengan maksud untuk mendefinisikan data yang mengalir di dalam sistem dengan lengkap sehingga *user* dan analis sistem mempunyai pengertian yang sama tentang *input*, *output*, dan komponen data *store*. Pada tahap perancangan sistem, kamus data digunakan untuk merancang *input*, laporan dan basis data. Berikut adalah kamus data sistem informasi pendistribusian unit:

#### 1. Spesifikasi Tabel *User*

Nama tabel	: <i>user</i>
Akronim	: <i>Users</i> (Pengguna)
Fungsi	: Untuk menyimpan data login pengguna
Tipe	: Data master

Tabel V.15 Kamus Data Users

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	ID User	id_user	Integer	11	Primary key
2	Username	username	Varchar	30	
3	Nama	nama	Varchar	30	
4	Password	password	Varchar	30	
5	Level	level	Enum	(delivery,admin_sj,adm_gdg,kpl_gdg)	

Sumber: Hasil Analisis (2020)

## 2. Spesifikasi Tabel Master Distributor

Nama tabel : p\_distribut  
 Akronim : Distributor  
 Fungsi : Untuk menyimpan data distributor  
 Tipe : Data master

Tabel V.16 Kamus Data Distributor

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	ID Distributor	id_distribut	Integer	11	Primary key
2	Name	name	Varchar	100	
3	Alamat	alamat	Varchar	250	
4	ID Areal	areal_id	Integer	11	Foreign Key
5	No Telp	no_telp	Integer	18	
6	Longitudinal Suzuki	long_suzuki	Decimal	(10,5)	
7	Latitude Suzuki	lat_suzuki	Decimal	(10,5)	
8	Longitudinal	longitudinal	Decimal	(10,5)	
9	Latitude	latitude	Decimal	(10,5)	
10	Euclidean	euclidean	Integer	11	
11	Keterangan	keterangan	Varchar	200	

Sumber: Hasil Analisis (2020)

## 3. Spesifikasi Tabel Data Areal

Nama tabel : p\_areal  
 Akronim : Areal  
 Fungsi : Untuk menyimpan data areal distributor  
 Tipe : Data Master

Tabel V.17 Kamus Data Areal

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	ID Areal	areal_id	Integer	11	<i>Primary key</i>
2	Name	name	Varchar	30	

Sumber: Hasil Analisis (2020)

## 4. Spesifikasi Tabel Produk Item

Nama tabel : p\_item  
 Akronim : Produk item  
 Fungsi : Untuk menyimpan data produk item  
 Tipe : Data Master

Tabel V.18 Kamus Data Produk Item

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	ID Item	item_id	Integer	11	<i>Primary key</i>
2	NO Seri	no_seri	Varchar	20	
3	Name	Name	Varchar	100	
4	Category	category_id	Integer	11	<i>Foreign Key</i>
5	Warna	warna_id	Integer	11	<i>Foreign Key</i>
6	Stock	Stock	Integer	11	
7	Note	Note	Varchar	100	

Sumber: Hasil Analisis (2020)

5. Spesifikasi Tabel Produk *Category*

Nama tabel : p\_category  
 Akronim : *Category*  
 Fungsi : Untuk menyimpan data produk *category*  
 Tipe : Data Master

Tabel V.19 Kamus Data *Category*

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	ID Category	category_id	Integer	11	<i>Primary key</i>
2	Name	Name	Varchar	30	

Sumber: Hasil Analisis (2020)

6. Spesifikasi Tabel Produk *Warna*

Nama tabel : p\_warna

Akronim : Warna

Fungsi : Untuk menyimpan data produk warna

Tipe : Data Master

Tabel V.20 Kamus Data *Warna*

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	ID Warna	warna_id	Integer	11	<i>Primary key</i>
2	Name	Name	Varchar	30	

Sumber: Hasil Analisis (2020)

## 7. Spesifikasi Tabel Gudang

Nama tabel : tambun1

Akronim : Gudang

Fungsi : Untuk menyimpan data stok ketersediaan

Tipe : Data Transaksi

Tabel V.21 Kamus Data *Gudang*

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	ID Gudang	id_gdg	Integer	11	<i>Primary key</i>
2	ID Item	item_id	Integer	11	<i>Foreign Key</i>
3	Stok Awal	stock_awal	Integer	11	
4	Stok Akhir	stock_akhir	Integer	11	
5	<i>Quantity</i>	Qty	Integer	11	
6	Note	Catatan	Varchar	100	
7	Tanggal	Tanggal	Date	-	

Sumber: Hasil Analisis (2020)

8. Spesifikasi Tabel *Delivery Order*

Nama tabel : p\_delivery  
 Akronim : *Delivery Order* (DO)  
 Fungsi : Untuk menyimpan data DO yang masuk  
 Tipe : Data Transaksi

Tabel V.22 Kamus Data DO

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	ID DO	id_do	Integer	11	<i>Primary key</i>
2	NO DO	no_do	Varchar	20	
3	ID Item	item_id	Integer	11	<i>Foreign Key</i>
4	ID Distributor	id_distribut	Integer	11	<i>Foreign Key</i>
5	<i>Quantity</i>	qty	Integer	11	
6	Tanggal	tanggal	Date	-	

Sumber: Hasil Analisis (2020)

## 9. Spesifikasi Tabel Surat Jalan

Nama tabel : p\_sj  
 Akronim : Surat Jalan  
 Fungsi : Untuk menyimpan data surat jalan yang keluar  
 Tipe : Data Transaksi

Tabel V.24 Kamus Data Surat Jalan

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	ID S.J	id_sj	Integer	11	<i>Primary key</i>
2	ID DO	id_do	Integer	11	<i>Foreign key</i>
3	<i>Quantity</i>	qty	Integer	11	
4	Tanggal	tanggal	Date	-	

Sumber: Hasil Analisis (2020)

## 10. Spesifikasi Tabel Rute

Nama tabel	: p_rute
Akronim	: Rute
Fungsi	: Untuk menyimpan data rute pengiriman
Tipe	: Data Transaksi

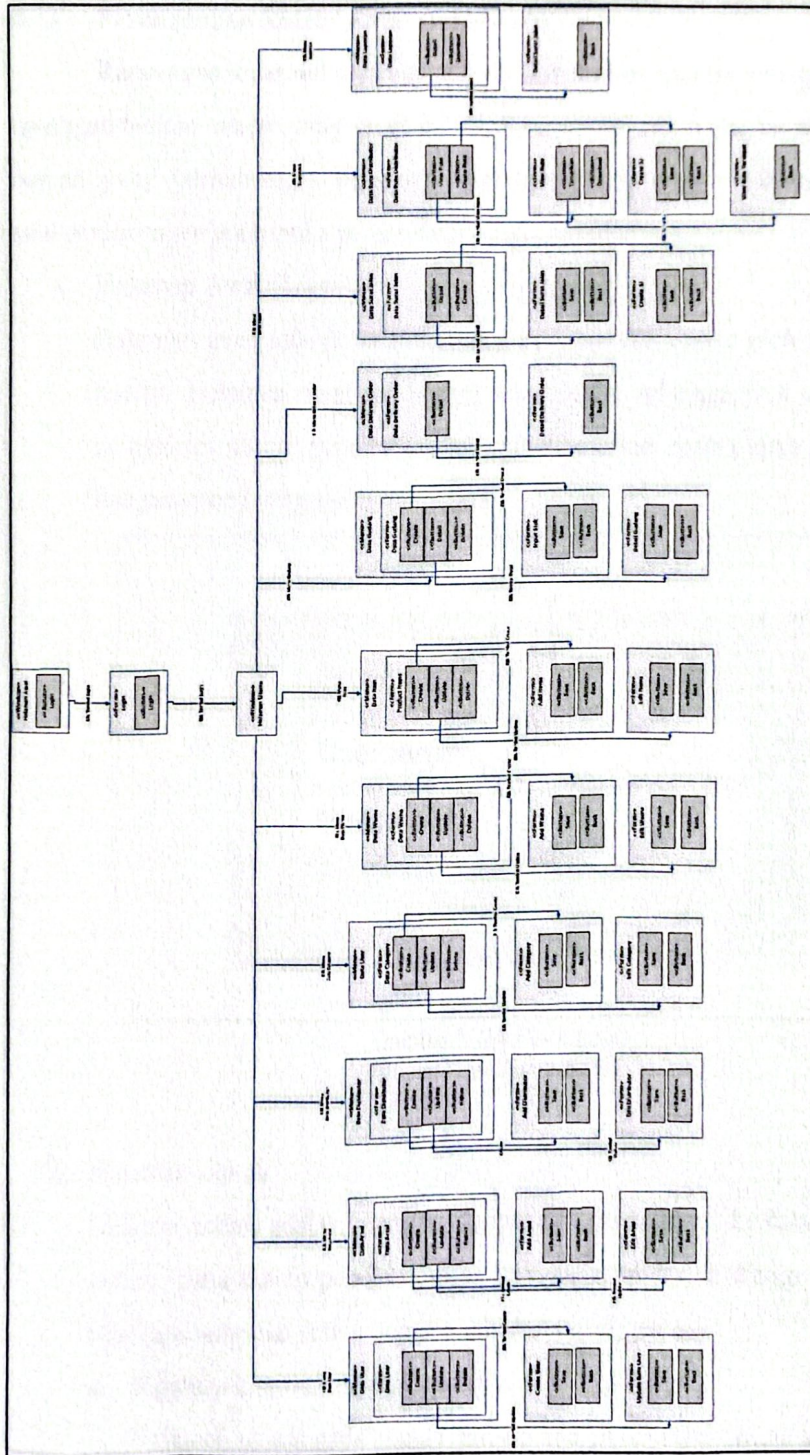
Tabel V.25 Kamus Data Rute

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	ID Rute	id_rute	Integer	11	<i>Primary key</i>
2	ID Distributor	id_distribut	Integer	11	<i>Foreign Key</i>
3	<i>First Node</i>	first_iter	Integer	11	
4	<i>Next Node</i>	next_iter	Integer	11	
5	<i>Euclidean</i>	euclidean	Decimal	(10,6)	

Sumber: Hasil Analisis (2020)

5.12 *Window Navigation Diagram*

*Windows Navigation Diagram* (WND) pada sistem usulan untuk menunjukkan bagaimana navigasi dari halaman-halaman yang terdapat di dalam aplikasi. Berikut ini *Windows Navigation Diagram* (WND) usulan pada sistem pendistribusian unit dapat dilihat pada Gambar V.33 di bawah ini:



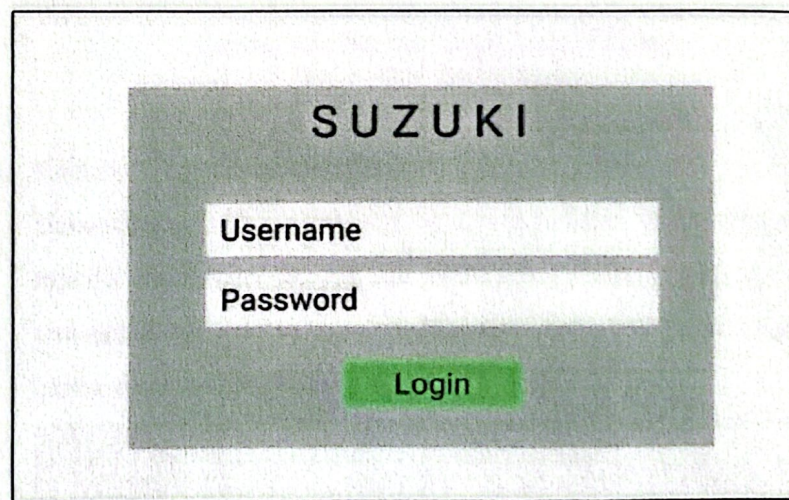
Gambar V.33 Windows Navigation Diagram Usulan  
Sumber: Hasil Analisis (2020)

### 5.13 Perancangan Sistem Antarmuka

Rancangan antarmuka dari sistem pendistribusian unit ini bertujuan untuk menggambarkan sistem yang telah dibuat. Rancangan antarmuka ini merupakan bagian yang berhubungan langsung antara pengguna dengan sistem, berikut adalah rancangan antarmuka pada sistem pendistribusian unit:

#### 1. Halaman Awal (*Login*)

Halaman awal adalah halaman yang pertama kali dilihat oleh *user* pada sistem. Halaman awal ini berupa *form login*, sehingga jika *user* ingin mengakses sistem pendistribusian unit diharuskan memasukan *username* dan *password* yang sudah terdaftar.

The image shows a login form with a dark grey background. At the top, the word "SUZUKI" is written in large, bold, black capital letters. Below this, there are two white rectangular input fields. The first field is labeled "Username" and the second is labeled "Password". Below the input fields, there is a green rectangular button with the word "Login" written in white text.

Gambar V.34 *Form Login*  
Sumber: Hasil Analisis (2020)

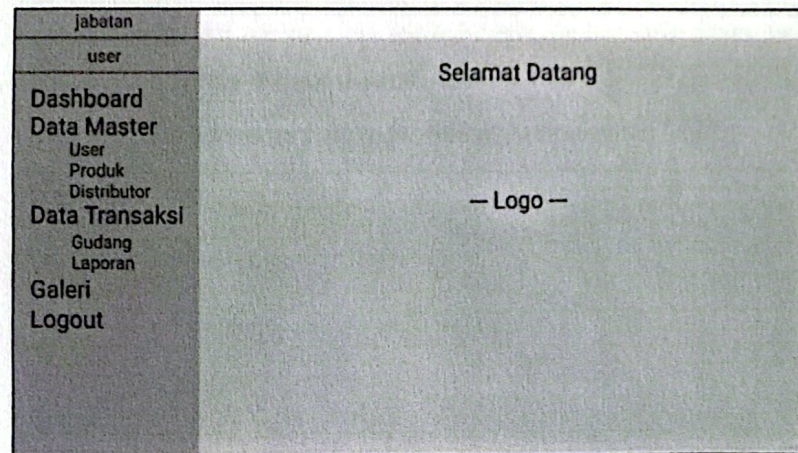
#### 2. Halaman Utama

Halaman utama adalah tampilan utama ketika *user* masuk ke dalam sebuah sistem, pada sistem pendistribusian unit halaman utama dibagi ke dalam beberapa tampilan sesuai dengan hak akses *user*, yaitu:

##### a. Halaman Utama Kepala Gudang

Halaman utama pada kepala gudang adalah halaman utama yang akan muncul ketika *user* dengan hak akses sebagai kepala gudang yaitu terdapat menu data maseter yang meliputi menu *user*, data distributor,

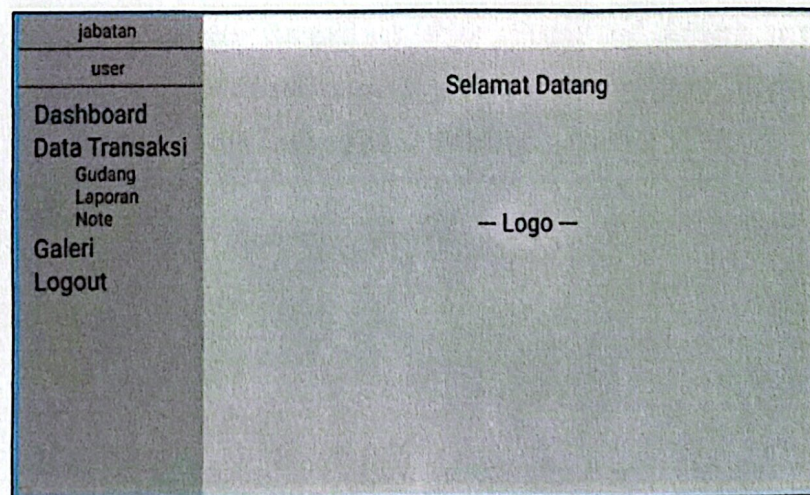
data produk serta menu data transaksi gudang dan laporan. Berikut halaman utama kepala gudang dapat dilihat pada Gambar V.35



Gambar V.35 Halaman Utama Kepala Gudang  
Sumber: Hasil Analisis (2020)

b. Halaman Utama Admin Gudang

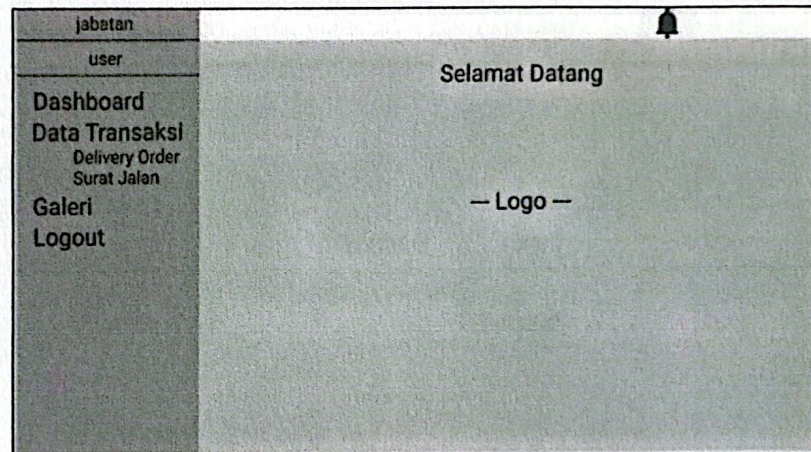
Halaman utama pada admin gudang adalah halaman utama yang akan muncul ketika *user* dengan hak akses sebagai admin gudang yaitu terdapat menu data transaksi gudang dan laporan. Berikut halaman utama admin gudang dapat dilihat pada Gambar V.36:



Gambar V.36 Halaman Utama Admin Gudang  
Sumber: Hasil Analisis (2020)

c. Halaman Utama Admin Surat Jalan

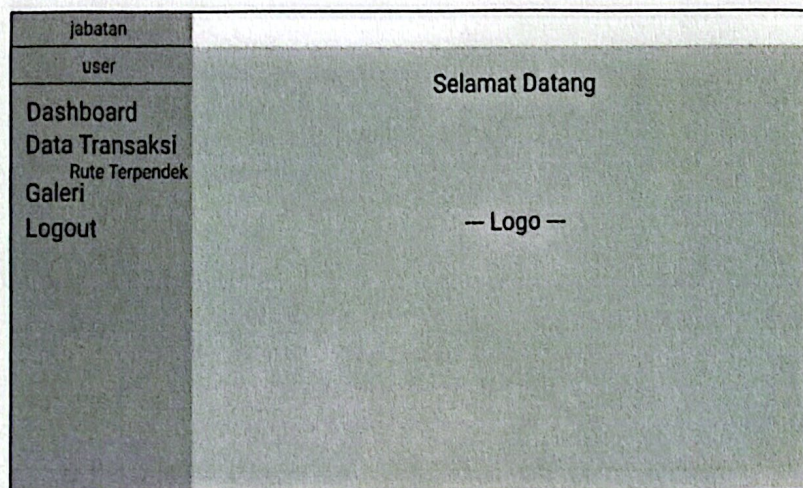
Halaman utama pada admin surat adalah halaman utama yang akan muncul ketika *user* dengan hak akses sebagai admin surat jalan yaitu terdapat menu data transaksi surat jalan dan *delivery order*. Berikut halaman utama admin gudang dapat dilihat pada Gambar V.37:



Gambar V.37 Halaman Utama Admin Surat Jalan  
Sumber: Hasil Analisis (2020)

d. Halaman Utama Staf *Delivery*

Halaman utama pada staf *delivery* adalah halaman utama yang akan muncul ketika *user* dengan hak akses sebagai staf *delivery* yaitu terdapat menu data transaksi surat jalan dan *delivery order*. Berikut halaman utama admin gudang dapat dilihat pada Gambar V.38:

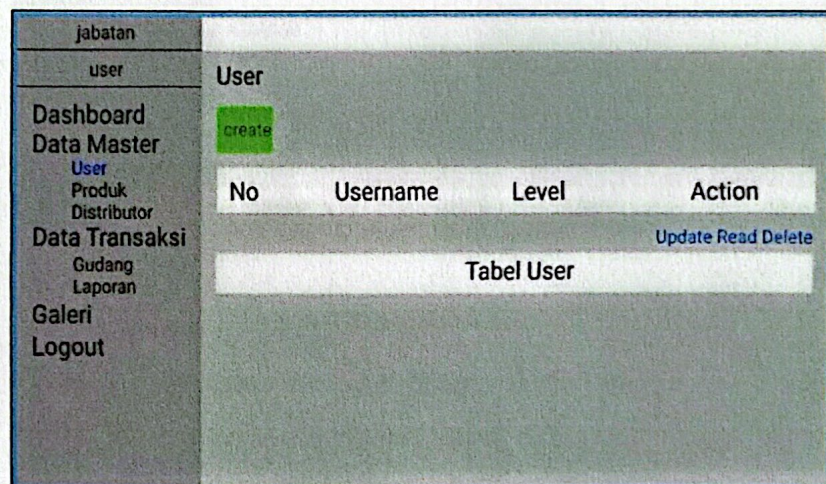


Gambar V.38 Halaman Utama Staf *Delivery*  
Sumber: Hasil Analisis (2020)

### 3. Halaman Menu Kepala Gudang

#### a. Data User

Halaman pada menu data *user* adalah halaman yang akan muncul pada halaman *user* dengan hak akses sebagai kepala gudang. Kepala gudang dapat mengubah, menghapus dan menambah data *user*. Berikut Gambar V.39 merupakan halaman menu data *user*:



Gambar V.39 Halaman Menu Data User  
Sumber: Hasil Analisis (2020)

Gambar V.40 Halaman Create User  
Sumber: Hasil Analisis (2020)

jabatan	
user	User Update
Dashboard	
Data Master	Username
User	
Produk	nama
Distributor	
Data Transaksi	password
Gudang	
Laporan	
Galeri	alamat
Logout	level
	<input type="button" value="update"/> <input type="button" value="cancel"/>

Gambar V.41 Halaman *Update User*  
Sumber: Hasil Analisis (2020)

b. Data Distributor

Halaman pada menu data distributor adalah halaman yang akan muncul pada halaman distributor dengan hak akses sebagai kepala gudang. Kepala gudang dapat mengubah, menghapus dan menambah data distributor. Berikut Gambar V.42 merupakan halaman menu data distributor:

jabatan	
user	Distributor
Dashboard	
Data Master	<input type="button" value="create"/>
User	
Produk	No Nama Alamat Areal No Telp Longitudinal Latitude Euclidean Action
Distributor	<input type="button" value="Update"/> <input type="button" value="Delete"/>
Data Transaksi	
Gudang	
Laporan	
Galeri	
Logout	
	Tabel Distributor

Gambar V.42 Halaman Menu Data Distributor  
Sumber: Hasil Analisis (2020)

jabatan	
user	Add Distributor
Dashboard	Distributor Name
Data Master	Alamat
User	Areal
Produk	No Telp
Distributor	Longitudinal
Data Transaksi	Latitude
Gudang	Keterangan
Laporan	
Galeri	
Logout	

create delete

Gambar V.43 Halaman *Add Distributor*  
Sumber: Hasil Analisis (2020)

jabatan	
user	Edit Distributor
Dashboard	Distributor Name
Data Master	Alamat
User	Areal
Produk	No Telp
Distributor	Longitudinal
Data Transaksi	Latitude
Gudang	Keterangan
Laporan	
Galeri	
Logout	

create delete

Gambar V.44 Halaman *Edit Distributor*  
Sumber: Hasil Analisis (2020)

### c. Data Produk

Halaman pada menu data produk adalah halaman yang akan muncul pada halaman produk dengan hak akses sebagai kepala gudang. Kepala gudang dapat mengubah, menghapus dan menambah data produk. Berikut Gambar V.45 ialah halaman menu produk dapat dilihat di bawah ini:

jabatan	
user	Produk
Dashboard	<a href="#">create</a>
Data Master	
User	
Produk	
Distributor	
Data Transaksi	No No Seri Nama Category Warna Stok Action
Gudang	
Laporan	<a href="#">Update</a> <a href="#">Delete</a>
Galeri	
Logout	Tabel Data Produk

Gambar V.45 Halaman Menu Data Produk  
Sumber: Hasil Analisis (2020)

jabatan	
user	Add Items
Dashboard	
Data Master	No Seri
User	
Produk	
Distributor	Product Name
Data Transaksi	Category
Gudang	
Laporan	Warna
Galeri	Note
Logout	<a href="#">create</a> <a href="#">cancel</a>

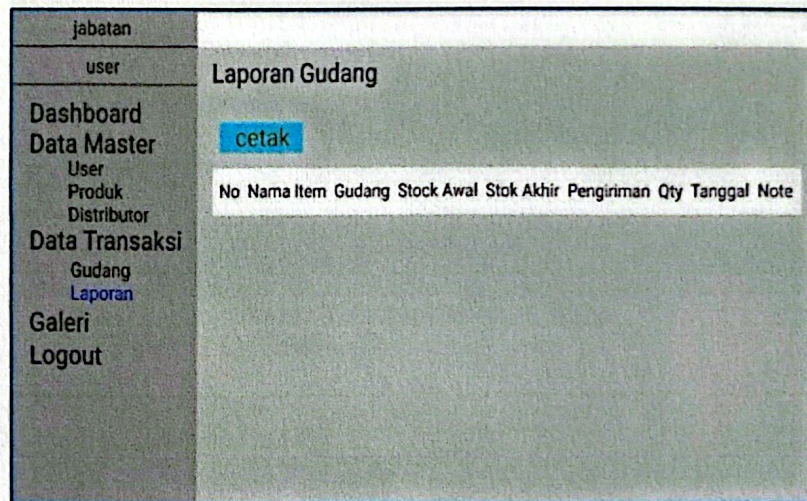
Gambar V.46 Halaman *Add Items*  
Sumber: Hasil Analisis (2020)

jabatan	
user	Edit Items
Dashboard	
Data Master	No Seri
User	
Produk	
Distributor	Product Name
Data Transaksi	Category
Gudang	
Laporan	Warna
Galeri	Note
Logout	<a href="#">create</a> <a href="#">cancel</a>

Gambar V.47 Halaman *Edit Items*  
Sumber: Hasil Analisis (2020)

d. Data Laporan

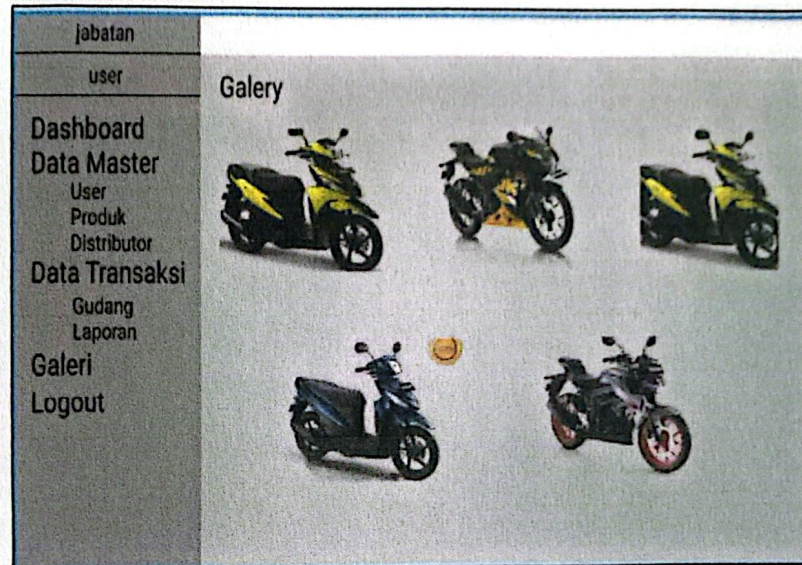
Halaman pada menu data laporan adalah halaman yang akan muncul pada halaman laporan dengan hak akses sebagai kepala gudang. Kepala gudang dapat melihat laporan gudang tersebut. Berikut Gambar V.48 merupakan halaman menu data laporan:



Gambar V.48 Halaman Menu Data Laporan  
Sumber: Hasil Analisis (2020)

e. Data Galeri

Halaman pada menu galeri adalah halaman yang akan muncul pada halaman galeri dengan hak akses ialah semua aktor yang tersimpan dalam sistem. Kepala gudang dapat melihat galeri yang berisikan motor-motor yang diproduksi maupun dijual oleh PT Szuzuki Indomobil Motor *Plant I Tambun*. Berikut Gambar V.49 merupakan halaman menu galeri:

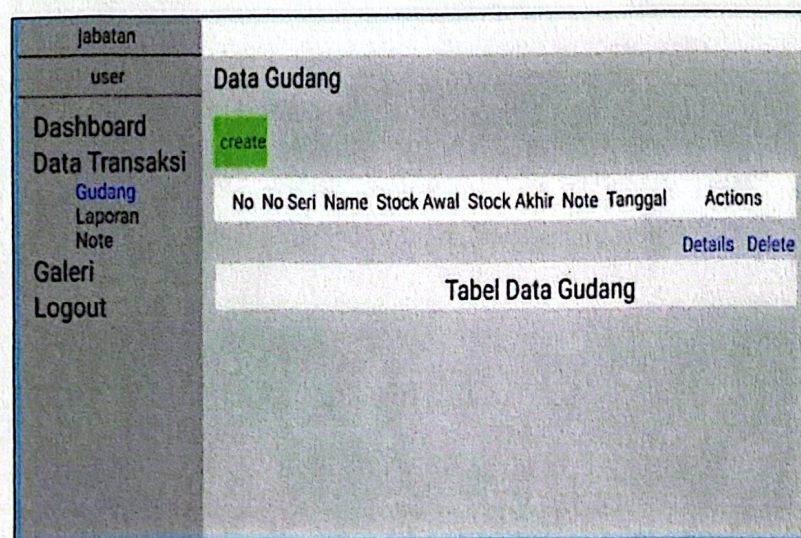


Gambar V.49 Halaman Menu Galeri  
Sumber: Hasil Analisis (2020)

#### 4. Halaman Menu Admin Gudang

##### a. Data Gudang

Halaman pada menu gudang adalah halaman yang akan muncul pada halaman gudang dengan hak akses ialah admin gudang. Admin gudang dapat menambahkan stok ketersediaan unit, melihat detail dan menghapus data. Berikut Gambar V.50 merupakan halaman menu gudang:



Gambar V.50 Halaman Menu Data Gudang  
Sumber: Hasil Analisis (2020)

jabatan	
user	Input Data Gudang
Dashboard	Tanggal
Data Transaksi	No Seri
Gudang	Nama Item
Laporan	Category    Warna
Note	Initial Stock
Galeri	Note
Logout	Qty
	create    reset    back

Gambar V.51 Halaman Input Data Gudang  
Sumber: Hasil Analisis (2020)

jabatan		Detail Data Gudang	
user	Data Gud	No Seri	
Dashboard	create	Name	
Data Transaksi		Stock Awal	
Gudang	No No Se	Stock Akhir	Actions
Laporan		Catatan	Details Delete
Galeri		Tanggal	
Logout			

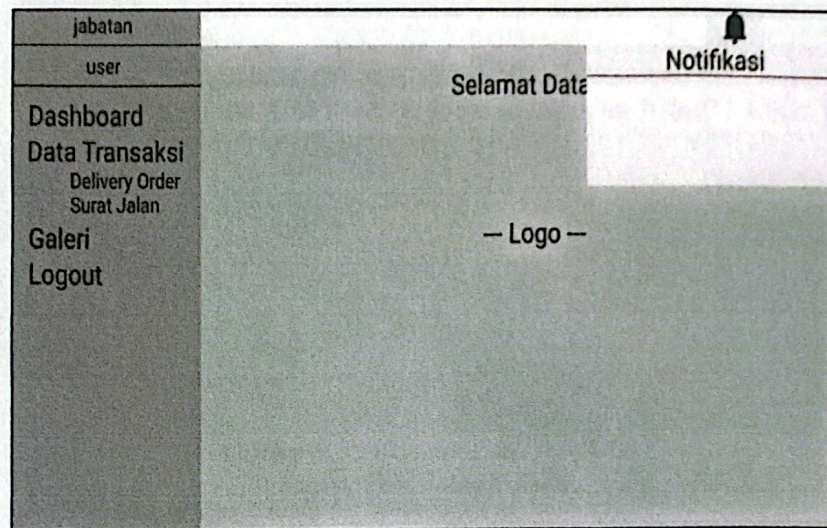
Gambar V.52 Halaman Detail Data Gudang  
Sumber: Hasil Analisis (2020)

## 5. Halaman Menu Admin Surat Jalan

### a. Notifikasi Ketersediaan Unit

Halaman pada menu notifikasi adalah halaman yang akan muncul pada halaman notifikasi dengan hak akses ialah admin surat jalan. Admin surat jalan hanya dapat melihat notifikasi yang masuk ke dalam

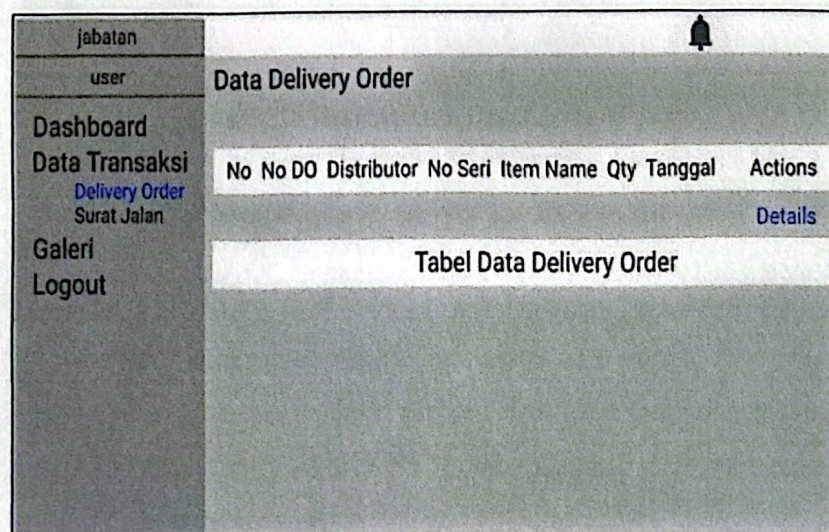
sistemnya. Berikut Gambar V.53 merupakan halaman menu notifikasi ketersediaan unit.



Gambar V.53 Halaman Notifikasi Ketersediaan Unit  
Sumber: Hasil Analisis (2020)

b. *Delivery Order*

Halaman pada menu *delivery order* adalah halaman yang akan muncul pada halaman DO dengan hak akses ialah admin surat jalan. Admin surat jalan hanya dapat melihat data *delivery order* dan melihat detail dari DO tersebut. Berikut Gambar V.54 merupakan halaman menu *delivery order*:



Gambar V.54 Halaman Data *Delivery Order*  
Sumber: Hasil Analisis (2020)

jabatan							
user	Data Delivery Order						
Dashboard	Periode		Periode		create sj		
Data Transaksi							
Delivery Order	No	No DO	Distributor	No Seri	Item Name	Qty	Tanggal
Surat Jalan							Actions
Galeri							Details
Logout							

Gambar V.55 Halaman Data View DO  
Sumber: Hasil Analisis (2020)

jabatan	Details Delivery Order						
user	Data Deliv	No Delivery Order					
Dashboard		Distributor					
Data Transaksi	No	No DC	Alamat Distributor			ggal	Actions
Delivery Order		No Telp					
Surat Jalan		Distributor Longitud.					Details Delete
Galeri		Distributor Latitute					
Logout		Distributor Euclidean					
		No Seri					
		Nama Item					
		Warna					
		Qty					
		Tanggal					

Gambar V.56 Halaman Data Detail *Delivery Order*  
Sumber: Hasil Analisis (2020)

### c. Surat Jalan

Halaman pada menu surat jalan adalah halaman yang akan muncul pada halaman surat jalan dengan hak akses ialah admin surat jalan. Admin surat jalan hanya dapat melihat data surat jalan dan melihat detail dari data surat jalan tersebut. Berikut Gambar V.57 merupakan halaman menu *delivery order*:

jabatan	
user	Data Surat Jalan
Dashboard	
Data Transaksi	No No Surat Jalan Distributor Periode Tanggal Dibuat Action
Delivery Order	
Surat Jalan	View S.J
Galeri	Tabel Data Surat Jalan
Logout	

Gambar V.57 Halaman Data Surat Jalan  
Sumber: Hasil Analisis (2020)

## 6. Halaman Menu Staf *Delivery*

### a. Notifikasi SJ

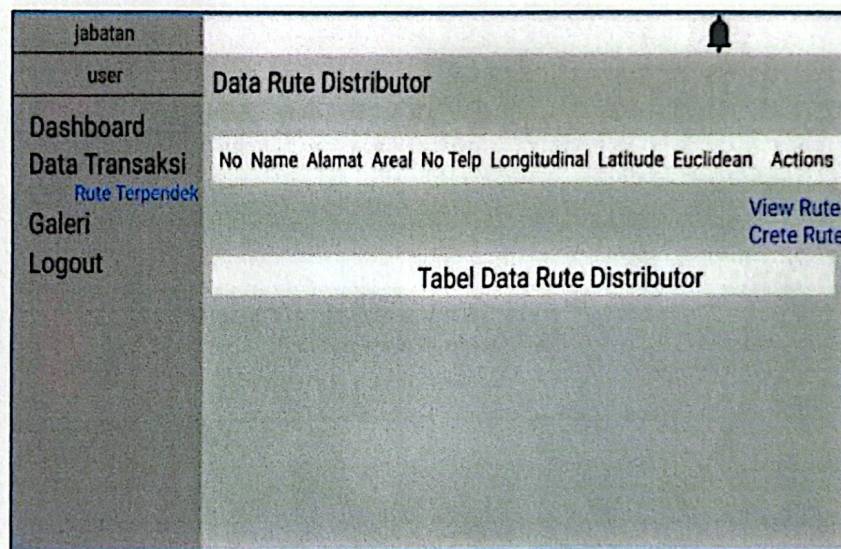
Halaman pada menu notifikasi SJ adalah halaman yang akan muncul pada halaman notifikasi SJ dengan hak akses ialah staf *delivery*. Staf *delivery* dapat melihat notifikasi tersebut atau dapat menyimpannya dengan mengklik notifikasi tersebut. Berikut Gambar V.58 merupakan halaman menu notifikasi SJ:

jabatan	
user	Selamat Data Notifikasi
Dashboard	
Data Transaksi	
Rute Terpendek	
Galeri	-- Logo --
Logout	

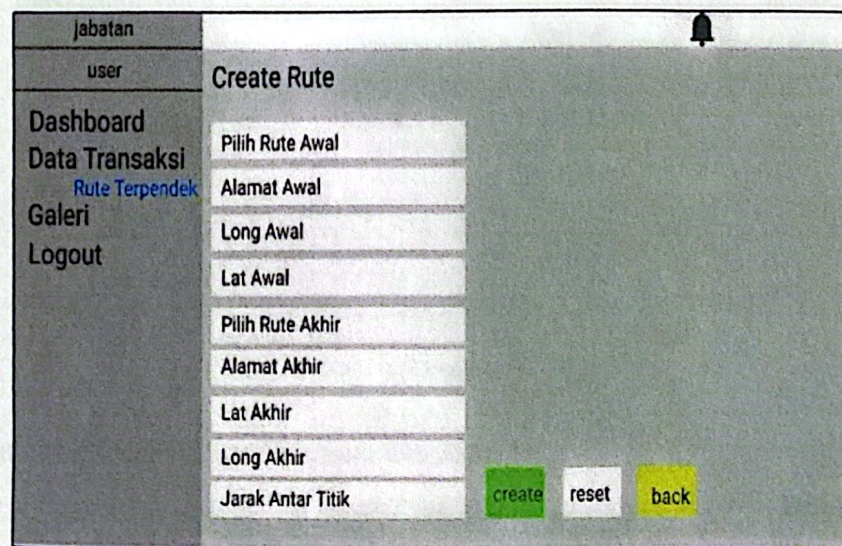
Gambar V.58 Halaman Notifikasi SJ  
Sumber: Hasil Analisis (2020)

b. Rute Terpendek

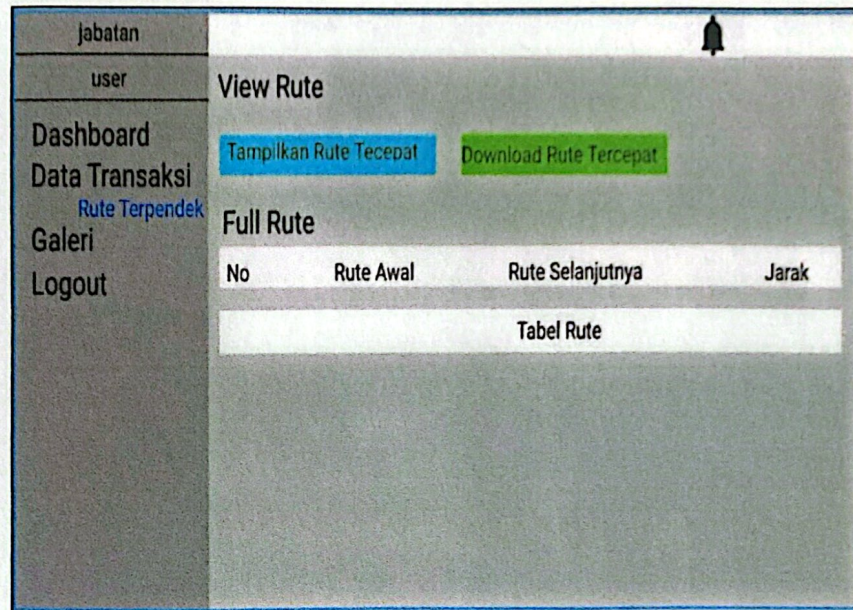
Halaman pada menu rute terpendek adalah halaman yang akan muncul pada halaman rute terpendek dengan hak akses ialah staf *delivery*. Staf *delivery* dapat mengkalkulasi dengan proses yang terkomputerisasi sehingga menghasilkan informasi rute tercepat untuk pendistribusian unit. Berikut Gambar V.59 merupakan halaman menu rute terpendek:



Gambar V.59 Halaman Rute Terpendek  
Sumber: Hasil Analisis (2020)



Gambar V.60 Halaman *Create Rute*  
Sumber: Hasil Analisis (2020)



Gambar V.61 Halaman *View Tampil Full Rute*  
Sumber: Hasil Analisis (2020)



Gambar V.62 Halaman *View Tampil Rute Tercepat*  
Sumber: Hasil Analisis (2020)

#### 5.14 Spesifikasi Kebutuhan *Hardware* dan *Software*

Spesifikasi perangkat lunak maupun perangkat keras untuk mendukung kebutuhan implementasi sistem sangat diperlukan. Implementasi sistem itu sendiri

merupakan tahap pengkodean program menggunakan bahasa pemrograman *PHP* dengan *framework Codeigniter*. Adapun spesifikasinya berikut ini:

1. Kebutuhan sistem *server*

Berikut adalah kebutuhan sistem untuk *server*:

a. Kebutuhan *software*

- 1) Sistem operasi : *Ubuntu*
- 2) *Database server* : *MariaDB 10.4.11*
- 3) Bahasa pemrograman : *PHP*
- 4) *Web server* : *Apache*
- 5) *Framework* : *Codeigniter 3.1.11*
- 6) *Web browser* : *Google Chrome*
- 7) *Software* : *Web Mail Server, DHCP server, ProFTPD*
- 8) Bahasa Pemrograman : *Hypertext Preprocessor (PHP)*
- 9) *Text Editor* : *VSCode, Sublime*

b. Kebutuhan *hardware*

- 1) *Processor* : *Intel Xeon E7*
- 2) *RAM* : *RAM 32 GB*
- 3) *Harddisk* : *2 TB*
- 4) *Media input* : *Mouse, Keyboard, Monitor*
- 5) *Media output* : *Printer*

2. Kebutuhan sistem *client*

Berikut adalah kebutuhan sistem untuk *client*:

a. Kebutuhan *software*

- 1) Sistem Operasi : *Minimal Microsoft Windows 8*
- 2) *Web browser* : *Google Chrome*

b. Kebutuhan *hardware*

- 1) *Processor* : *Minimal Core i7*
- 2) *Harddisk* : *Minimal 500 GB*
- 3) *RAM* : *Minimal 8 RAM*
- 4) *Media Input* : *Mouse, Keyboard, Monitor*
- 5) *Media Output* : *Printer*

### 5.15 Blackbox Testing

*Blackbox testing* dilakukan untuk menguji perangkat lunak yang berfokus dalam spesifikasi fungsional dari perangkat lunak. *Blackbox testing* menguji sistem dari *usecase* yang telah dibuat, berikut pengujian yang dilakukan:

#### 1. Login

*Test Case ID* : Login001.

*Function* : Operasi validasi saat melakukan *login*.

*Data Assumption* : Fungsi operasi validasi saat melakukan *login* berjalan dengan baik. Penggunaan huruf kapital dan huruf kecil tidak mempengaruhi pada saat validasi *login*.

*Deskripsi* : Melakukan *login* ke dalam sistem dengan menguji kesalahan *username* dan *password*.

Tabel V.25 Test Case Login

<i>Test ID</i>	<i>Test Case</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
001	Validasi login	Input dengan <i>username</i> yang salah dan <i>password</i> benar lalu klik tombol login	Tidak dapat masuk, lalu program memberikan pesan bahwa <i>username</i> dan <i>password</i> tidak sesuai	Sesuai
002	Validasi login	Input dengan benar <i>username</i> dan <i>password</i> lalu klik tombol login	Berhasil login	Sesuai
003	Validasi login	<i>Username</i> dan <i>password</i> tidak diisi lalu klik tombol login	Tidak berhasil masuk, lalu program memberikan pesan <i>field</i> harus diisi	Sesuai

Sumber: Hasil Analisis (2020)

## 2. Mengelola Data User

*Test Case ID* : User001.

*Function* : Operasi mengelola data *user*

*Data Assumption* : Fungsi operasi data *user* digunakan untuk mengelola data *user*

Deskripsi : Mengklik *create*, *update*, dan *delete* pada data *user*.

Tabel V.26 *Test Case* Mengelola Data User

<i>Test ID</i>	<i>Test Case</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
001	Mengklik <i>create</i> pada form <i>user</i>	Program akan menampilkan form <i>create user</i>	Menampilkan form <i>create user</i>	Sesuai
002	Mengklik <i>update</i> pada data <i>user</i>	Program akan menampilkan form <i>update user</i>	Menampilkan form <i>update user</i>	Sesuai
003	Mengklik <i>delete</i> pada data <i>user</i>	Program akan menghapus data yang dipilih	Menghapus data	Sesuai
004	Mengklik <i>read</i> pada data <i>user</i>	Program akan menampilkan detail data yang dipilih	Menampilkan detail data yang dipilih	Sesuai
005	Memasukkan <i>keyword</i> pada pencarian	Program akan menampilkan data yang berhubungan dengan <i>keyword</i>	Menampilkan data sesuai <i>keyword</i>	Sesuai
006	Menambah data yang sesuai lalu klik tombol <i>save</i> atau <i>enter</i>	Program akan <i>update</i> data yang telah diinput ke dalam <i>database user</i>	Program menambah data ke <i>database user</i>	Sesuai

Tabel V.26 *Test Case* Mengelola Data *User* (lanjutan).

<i>Test ID</i>	<i>Test Case</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
007	Menambah data dan mengosongkan salah satu data lalu klik tombol <i>save</i> atau <i>enter</i>	Program akan memberikan pesan bahwa data harus diisi	Program memberi pesan bahwa data harus diisi	Sesuai
008	Mengklik tombol kembali	Program akan kembali ke menu <i>form data user</i>	Program kembali ke menu <i>form data user</i>	Sesuai
009	Mengubah data sesuai yang ingin diubah lalu klik <i>save</i> atau <i>enter</i>	Program akan mengubah data di dalam <i>database user</i>	Program mengubah data di dalam <i>database user</i>	Sesuai
010	Mengubah data yang sesuai dan mengosongkan data lalu klik <i>save</i> atau <i>enter</i>	Program akan memberikan pesan bahwa data harus diisi	Program memberi pesan bahwa data harus diisi	Sesuai

Sumber: Hasil Analisis (2020)

### 3. Mengelola Data Produk

*Test Case ID* : Produk001.

*Function* : Operasi mengelola data produk

*Data Assumption* : Fungsi operasi data produk digunakan untuk mengelola data produk

Deskripsi : Mengklik *create*, *update*, dan *delete* pada data produk.

Tabel V.27 *Test Case* Mengelola Data Produk

<i>Test ID</i>	<i>Test Case</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
001	Mengklik <i>create</i> pada form <i>product items</i>	Program akan menampilkan form <i>add items</i>	Menampilkan form <i>add items</i>	Sesuai
002	Mengklik <i>update</i> pada produk	Program akan menampilkan form <i>edit items</i>	Menampilkan form <i>edit item</i>	Sesuai
003	Mengklik <i>delete</i> pada data produk	Program akan menghapus data yang dipilih	Menghapus data	Sesuai
004	Menambah data yang sesuai lalu klik tombol <i>save</i> atau <i>enter</i>	Program akan <i>update</i> data yang telah diinput ke dalam <i>database</i> produk	Program menambah data ke <i>database</i> produk	Sesuai
005	Menambah data dan mengosongkan salah satu data lalu klik tombol <i>save</i> atau <i>enter</i>	Program akan memberikan pesan bahwa data harus diisi	Program memberi pesan bahwa data harus diisi	Sesuai
006	Mengklik tombol kembali	Program akan kembali ke menu form data produk	Program kembali ke menu form data produk	Sesuai

Tabel V.27 *Test Case* Mengelola Data Produk (lanjutan)

<i>Test ID</i>	<i>Test Case</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
007	Mengubah data sesuai yang ingin diubah lalu klik <i>save</i> atau <i>enter</i>	Program akan mengubah data di dalam <i>database</i> produk	Program mengubah data di dalam <i>database</i> produk	Sesuai
008	Mengubah data yang sesuai dan mengosongkan data lalu klik <i>save</i> atau <i>enter</i>	Program akan memberikan pesan bahwa data harus diisi	Program memberi pesan bahwa data harus diisi	Sesuai
009	Mengklik tombol reset pada form <i>add items</i>	Program akan mereset data yang diinput	Program mereset data yang diinput	Sesuai
010	Mengklik tombol reset pada form <i>edit items</i>	Program akan mereset data yang diinput	Program mereset data yang diinput	Sesuai
011	Memasukkan <i>keyword</i> pada pencarian	Program akan menampilkan data yang berhubungan dengan <i>keyword</i>	Menampilkan data sesuai <i>keyword</i>	Sesuai

Sumber: Hasil Analisis (2020)

#### 4. Mengelola Data Distributor

*Test Case ID* : Distributor001.

*Function* : Operasi mengelola data distributor

*Data Assumption* : Fungsi operasi data distributor digunakan untuk mengelola data distributor

Deskripsi : Mengklik *create*, *update*, dan *delete* pada data distributor.

Tabel V.28 *Test Case* Mengelola Data Distributor

<i>Test ID</i>	<i>Test Case</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
001	Mengklik <i>create</i> pada <i>form</i> data distributor	Program akan menampilkan <i>form add</i> distributor	Menampilkan <i>form add</i> distributor	Sesuai
002	klik <i>update</i> pada data distributor	Program akan menampilkan <i>form edit</i> distributor	Menampilkan <i>form edit</i> distributor	Sesuai
003	Mengklik <i>delete</i> pada data distributor	Program akan menghapus data yang dipilih	Menghapus data	Sesuai
004	Menambah data yang sesuai lalu klik tombol <i>save</i> atau <i>enter</i>	Program akan <i>update</i> data yang telah diinput ke dalam <i>database</i> distributor	Program menambah data ke <i>database</i> distributor	Sesuai
005	Menambah data dan mengosongkan salah satu data lalu klik tombol <i>save</i>	Program akan memberikan pesan bahwa data harus diisi	Program memberi pesan bahwa data harus diisi	Sesuai
006	Mengklik tombol kembali	Program akan kembali ke menu <i>form</i> data distributor	Program kembali ke menu <i>form</i> data distributor	Sesuai

Tabel V.28 Test Case Mengelola Data Distributor (lanjutan)

<i>Test ID</i>	<i>Test Case</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
007	Mengubah data sesuai yang ingin diubah lalu klik <i>save</i> atau <i>enter</i>	Program akan mengubah data di dalam <i>database</i> produk	Program mengubah data di dalam <i>database</i> produk	Sesuai
008	Mengubah data yang sesuai dan mengosongkan data lalu klik <i>save</i> atau <i>enter</i>	Program akan memberikan pesan bahwa data harus diisi	Program memberi pesan bahwa data harus diisi	Sesuai
009	Mengklik tombol reset pada form <i>add</i> distributor	Program akan mereset data yang diinput	Program mereset data yang diinput	Sesuai
010	Mengklik tombol reset pada form <i>edit</i> distributor	Program akan mereset data yang diinput	Program mereset data yang diinput	Sesuai
011	Memasukkan <i>keyword</i> pada pencarian	Program akan menampilkan data yang berhubungan dengan <i>keyword</i>	Menampilkan data sesuai <i>keyword</i>	Sesuai

Sumber: Hasil Analisis (2020)

### 5. Mengisi Stok Ketersediaan

*Test Case ID* : gudang001.

*Function* : Operasi mengisi stok ketersediaan

*Data Assumption* : Fungsi operasi data stok ketersediaan digunakan untuk mengelola data transaksi

Deskripsi : Mengklik *create*, detail, dan *delete* pada data distributor.

Tabel V.29 Test Case Mengisi Stok Ketersediaan

<i>Test ID</i>	<i>Test Case</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
001	Mengklik <i>create</i> pada <i>form</i> data gudang	Program akan menampilkan <i>form</i> input stok	Menampilkan <i>form</i> input stok	Sesuai
002	klik detail pada data gudang	Program akan menampilkan <i>form</i> detail data gudang	Menampilkan <i>form</i> detail data gudang	Sesuai
003	Mengklik <i>delete</i> pada data gudang	Program akan menghapus data yang dipilih	Menghapus data	Sesuai
004	Menambah data yang sesuai lalu klik tombol <i>save</i> atau <i>enter</i>	Program akan <i>update</i> data yang telah diinput ke dalam <i>database</i> gudang	Program menambah data ke <i>database</i> gudang	Sesuai
005	Menambah data dan mengosongkan salah satu data lalu klik tombol <i>save</i>	Program akan memberikan pesan bahwa data harus diisi	Program memberi pesan bahwa data harus diisi	Sesuai
006	Mengklik tombol kembali	Program akan kembali ke menu <i>form</i> data gudang	Program kembali ke menu <i>form</i> data gudang	Sesuai
007	Mengklik tombol reset pada <i>form add</i> distributor	Program akan mereset data yang diinput	Program mereset data yang diinput	Sesuai

Tabel V.30 *Test Case* Mengisi Stok Ketersediaan (lanjutan)

<i>Test ID</i>	<i>Test Case</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
008	Memasukkan <i>keyword</i> pada pencarian	Program akan menampilkan data yang berhubungan dengan <i>keyword</i>	Menampilkan data sesuai <i>keyword</i>	Sesuai

Sumber: Hasil Analisis (2020)

## 6. Melihat Notifikasi Ketersediaan

*Test Case ID* : notif001.*Function* : Operasi melihat notifikasi ketersediaan*Data Assumption* : Fungsi operasi melihat notifikasi ketersediaan digunakan sebagai informasi bagi aktor

Deskripsi : Mengklik notifikasi yang masuk dan melihat notifikasi tersebut

Tabel V.30 *Test Case* Melihat Notifikasi Ketersediaan

<i>Test ID</i>	<i>Test Case</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
001	Mengklik notifikasi pada <i>header</i> disistem	Program akan menampilkan <i>alert</i> notifikasi masuk berupa angka	Menampilkan notifikasi masuk	Sesuai
002	Me- <i>refresh</i> halaman pada sistem	Program akan menampilkan <i>alert</i> notifikasi kembali 0	Menampilkan notifikasi masuk kembali 0	Sesuai

Sumber: Hasil Analisis (2020)

## 7. Melihat Melihat Data DO

*Test Case ID* : do001.*Function* : Operasi melihat data DO*Data Assumption* : Fungsi operasi melihat data DO yang diterimaDeskripsi : Mengklik *view* DO pada data DO yang masuk

Tabel V.31 *Test Case* Melihat Data DO

<i>Test ID</i>	<i>Test Case</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
001	Mengklik <i>view</i> DO pada <i>form</i> data <i>delivery order</i>	Program akan menampilkan <i>form</i> data <i>delivery order</i> dengan pilihan periode waktu	<i>form</i> data <i>delivery order</i> dengan pilihan periode waktu	Sesuai
002	Mengklik detail DO pada <i>form</i> data <i>delivery order</i>	Program akan menampilkan <i>form</i> detail data <i>delivery order</i>	menampilkan <i>form</i> detail data <i>delivery order</i>	Sesuai
003	Mengklik tombol kembali	Program akan kembali ke menu <i>form</i> data <i>delivery order</i>	Program kembali ke menu <i>form</i> data gudang	Sesuai
004	Memasukkan <i>keyword</i> pada pencarian	Program akan menampilkan data yang berhubungan dengan <i>keyword</i>	Menampilkan data sesuai <i>keyword</i>	Sesuai

Sumber: Hasil Analisis (2020)

## 8. Membuat Surat Jalan

*Test Case ID* : sj001.

*Function* : Operasi membuat surat jalan

*Data Assumption* : Fungsi operasi membuat surat jalan yang digunakan untuk proses data transaksi selanjutnya

Deskripsi : Mengklik *view*, *create* SJ, detail pada data SJ

Tabel V.32 *Test Case* Membuat Surat Jalan

<i>Test ID</i>	<i>Test Case</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
001	Mengklik <i>view</i> DO pada <i>form</i> data DO	Program akan menampilkan <i>form</i> data <i>delivery order</i>	<i>form</i> data <i>delivery order</i>	Sesuai

Tabel V.32 *Test Case* Membuat Surat Jalan (lanjutan)

<i>Test ID</i>	<i>Test Case</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
002	Mengisi tanggal sesuai periode DO dan memilih tombol <i>create</i> SJ	Program akan <i>download</i> dokumen surat jalan	mendownload dokumen surat jalan	Sesuai
003	Mengklik pdf <i>ter-download</i>	Program akan menampilkan dokumen surat jalan berdasarkan periode waktu yang dipilih	menampilkan dokumen surat jalan berdasarkan periode waktu yang dipilih	Sesuai
004	Mengklik detail DO pada <i>form</i> data <i>delivery order</i>	Program akan menampilkan <i>form</i> detail data <i>delivery order</i>	menampilkan <i>form</i> detail data <i>delivery order</i>	Sesuai
005	Mengisi tanggal periode DO yang tidak sesuai dan memilih tombol <i>create</i> SJ	Program akan menampilkan pesan peringatan untuk memasukkan tanggal tanggal yang sesuai	Menampilkan pesan peringatan tanggal yang harus diinput ada	Sesuai
006	Mengklik tombol <i>create</i> SJ tanpa memasukkan periode tanggal	Program akan menampilkan pesan peringatan bahwa tanggal perlu diisi terlebih dahulu	Menampilkan pesan peringatan untuk mengisi tanggal	Sesuai

Tabel V.32 *Test Case* Membuat Surat Jalan (lanjutan)

<i>Test ID</i>	<i>Test Case</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
007	Mengklik <i>view</i> SJ pada <i>form</i> data surat jalan	Program akan <i>men-download</i> dokumen surat jalan	Men- <i>donwload</i> surat jalan	Sesuai
008	Memasukkan <i>keyword</i> pada pencarian	Program akan menampilkan data yang berhubungan dengan <i>keyword</i>	Menampilkan data sesuai <i>keyword</i>	Sesuai

Sumber: Hasil Analisis (2020)

#### 9. Melihat Notifikasi SJ

*Test Case ID* : notifsj001.

*Function* : Operasi melihat notifikasi notifikasi SJ

*Data Assumption* : Fungsi operasi melihat notifikasi SJ

Deskripsi : Mengklik notifikasi yang masuk, melihat serta menyimpan dokumen terlampir

Tabel V.34 *Test Case* Melihat Notifikasi SJ

<i>Test ID</i>	<i>Test Case</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
001	Mengklik notifikasi pada <i>header</i> disistem	Program akan menampilkan <i>alert</i> notifikasi masuk berupa angka	Menampilkan notifikasi masuk	Sesuai
002	Mengklik notifikasi yang masuk	Program akan <i>men-download</i> secara otomatis notifikasi masuk yang berupa surat jalan yang selesai dibuat oleh admin surat jalan	Men- <i>download</i> surat jalan dari notifikasi yang masuk	Sesuai

Tabel V.34 *Test Case* Melihat Notifikasi SJ (lanjutan)

<i>Test ID</i>	<i>Test Case</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
003	Me-refresh halaman pada sistem	Program akan menampilkan <i>alert</i> notifikasi kembali 0	Menampilkan notifikasi masuk kembali 0	Sesuai

Sumber: Hasil Analisis (2020)

#### 10. Membuat Jalur Distribusi

*Test Case ID* : rute001.

*Function* : Operasi membuat jalur distribusi

*Data Assumption* : Fungsi operasi membuat jalur distribusi yang digunakan sebagai informasi rute terpendek untuk pendistribusian unit

Deskripsi : Mengklik *view* rute, *create* rute, rute tercepat, download *file* rute pada data rute.

Tabel V.34 *Test Case* Membuat Jalur Distribusi

<i>Test ID</i>	<i>Test Case</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
001	Mengklik <i>create</i> rute pada data rute distributor	Program akan menampilkan <i>form create</i> rute	menampilkan <i>form create</i> rute	Sesuai
002	Mengklik <i>view</i> rute pada data rute distributor	Program akan menampilkan <i>form</i> rute yang menampilkan lengkap setiap rute	menampilkan <i>form</i> rute yang menampilkan lengkap setiap rute	Sesuai
003	Menambah data yang sesuai lalu klik tombol <i>save</i> atau <i>enter</i>	Program akan <i>update</i> data yang telah diinput ke dalam <i>database</i> rute	Program menambah data ke <i>database</i> rute	Sesuai

Tabel V.34 *Test Case* Membuat Jalur Distribusi (lanjutan)

<i>Test ID</i>	<i>Test Case</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
004	Menambah data dan mengosongkan salah satu data lalu klik tombol <i>save</i>	Program akan memberikan pesan bahwa data harus diisi	Program memberi pesan bahwa data harus diisi	Sesuai
005	Mengklik tombol kembali	Program akan kembali ke menu <i>form</i> data gudang	Program kembali ke menu <i>form</i> data gudang	Sesuai
006	Mengklik tombol reset pada form <i>create rute</i>	Program akan mereset data yang dimasukkan	Program mereset data yang dimasukkan	Sesuai
007	Mengklik tampilan rute tercepat pada data rute distributor	Program akan menampilkan <i>form</i> rute yang menampilkan lengkap setiap rute dan juga rute tercepat dan juga kalkulasi jarak tempuhnya	menampilkan <i>form</i> rute yang menampilkan lengkap setiap rute dan juga rute tercepat dan juga kalkulasi jarak tempuhnya	Sesuai
008	Mengklik <i>download</i> rute tercepat pada data rute distributor	Program akan menampilkan lokasi untuk menyimpan file rute terpendek yang akan di <i>download</i>	menampilkan lokasi untuk menyimpan file rute terpendek yang akan di <i>download</i>	Sesuai

Sumber: Hasil Analisis (2020)

## 11. Melihat laporan gudang

*Test Case ID* : laporan001.

*Function* : Operasi melihat laporan gudang

*Data Assumption* : Fungsi operasi laporan gudang yang digunakan untuk melihat laoran ketersediaan gudang

Deskripsi : Mengklik cetak laporan data gudang.

Tabel V.35 *Test Case* Melihat Laporan Gudang

<i>Test ID</i>	<i>Test Case</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
001	Memasukkan periode tanggal pada <i>form</i> data gudang dan klik tombol cetak laporan	Program akan <i>download</i> laporan gudang	Men- <i>download</i> laporan gudang	Sesuai
002	Memasukkan <i>keyword</i> pada pencarian	Program akan menampilkan semua data yang berhubungan dengan <i>keyword</i> yang dimasukkan pada kolom pencarian	Menampilkan semua data yang berhubungan dengan <i>keyword</i>	Sesuai
003	Mengklik cetak laporan tanpa memasukkan periode tanggal	Program akan menampilkan pesan peringatan bahwa periode tanggal harus diisi terlebih dahulu	Menampilkan pesan peringatan bahwa periode tanggal perlu diisi	Sesuai

Tabel V.35 *Test Case* Melihat Laporan Gudang (lanjutan)

<i>Test ID</i>	<i>Test Case</i>	<i>Description</i>	<i>Expected Result</i>	<i>Actual Result</i>
004	Memasukkan keyword pada kolom pencarian	Program akan menampilkan semua data yang berhubungan dengan keyword yang dimasukkan pada kolom pencarian	Menampilkan semua data yang berhubungan dengan <i>keyword</i>	Sesuai

Sumber: Hasil Analisis (2020)

### 5.16 Simulasi Pencarian Fungsi *Euclidean* Dengan *Microsoft Excel*

Simulasi pencarian fungsi *euclidean* dengan *microsoft excel* dilakukan agar mengetahui apakah fungsi yang diterapkan pada program sudah sesuai dengan perhitungan.

#### 1. Perhitungan Fungsi *Euclidean Distance*

Fungsi yang digunakan bertujuan untuk mencari nilai jarak antara satu titik dengan titik yang lainnya. Dalam kasus ini fungsi *euclidean* digunakan untuk mencari nilai jarak antara PT Suzuki Indomobil Motor Plant I Tambun dengan distributor. Berikut ini langkah perhitungan perhitungan jarak dengan fungsi *euclidean distance* pada *microsoft excel*:

##### a. Rumus perhitungan jarak *Euclidean Distance*

$$\text{Jarak} = \sqrt{(\text{Latitude1} - \text{Latitude2})^2 + (\text{Longitude1} - \text{Longitude2})^2}$$

dimana:

*Latitude1* = koordinat *Latitude1* pada titik awal

*Latitude2* = koordinat *Latitude2* pada titik tujuan

*Longitude1* = koordinat *Longitude1* pada titik awal

*Longitude2* = koordinat *Longitude1* pada titik tujuan

##### b. Hasil perhitungan (jarak) masih dalam satuan *decimal degree* (sesuai dengan format longlat yang dipakai) sehingga untuk menyesuaikan perlu dikalikan dengan 111.319 km (1 derajat bumi = 111.319 km).















## c. Keterangan pada Tabel

- 1) Lat suzuki: koordinat *latitude* pada titik awal
- 2) Long suzuki: koordinat *longitude* pada titik awal
- 3) Lat distributor: koordinat *latitude* pada titik tujuan
- 4) Long distributor: koordinat *longitude* pada titik tujuan
- 5) Derajat bumi: nilai dari derajat bumi 111.319 digunakan untuk menyesuaikan perkalian
- 6) Nilai *euclidean*: nilai dari perhitungan rumus yang digunakan untuk mendapatkan nilai jarak
- 7) Lokasi tujuan: Nama distributor
- 8) Alamat: keterangan alamat distributor

Berikut ini adalah Perhitungan Fungsi *Euclidean Distance* yang dilakukan dengan *microsoft excel* dapat dilihat pada Gambar V.65 di bawah ini:

Lokasi Tujuan	Lat Suzuki	Long Suzuki	Lat Distributor	Long Distributor	Derajat Bumi	Nilai Euclidean	Alamat
Suzuki Meulaboh Armada	-6.251901	107.033513	4.151338	96.147188	111.319	1676.227048	Gampong, Jl. Meul
Suzuki Takengon	-6.251901	107.033513	5.203716	96.718606	111.319	1716.004924	Jl. Lintang takengon
Suzuki Trans Sumatera Agung SM Raja	-6.251901	107.033513	3.538332	98.709465	111.319	1430.51818	Jalan Sisingamang
PT. Sunindo Varia Motor Gemilang	-6.251901	107.033513	3.602603	98.668753	111.319	1438.905015	Jl. H. Adam Malik N
Suzuki Selamat Motor	-6.251901	107.033513	-0.951674	100.357544	111.319	948.8987772	Jl. Moh. Yamin, Be
Suzuki Elang Perkasa Motor	-6.251901	107.033513	-0.904592	100.352406	111.319	952.6129801	Jl. Khatib Sulaiman
Distributor A	-6.251901	107.033513	-6.25753	107.00936	111.319	2.760740488	Jl. Chairil Anwar, R
Distributor B	-6.251901	107.033513	-6.250333	106.971611	111.319	6.893079068	Jl. Nasional 1, RT.0
Distributor C	-6.251901	107.033513	-6.21091	106.807158	111.319	25.60744688	RT.6/RW.3, Bend. h
Distributor D	-6.251901	107.033513	-6.130937	106.891105	111.319	20.79977799	North Jakarta, RT.1
Distributor E	-6.251901	107.033513	-6.224512	106.622401	111.319	45.86602634	Jl. Mina Raya, RT.0
Distributor F	-6.251901	107.033513	-6.13853	106.253699	111.319	87.72070402	Jl. Nasional 1, Kas

Gambar V.63 Perhitungan Fungsi *Euclidean Distance* Pada *Microsoft Excel*  
Sumber: Hasil Analisis (2020)

No	Name	Alamat	Areal	No Telp	Longitudinal	Latitude	Euclidean	Kelurahan	Artis
1.	Suzuki Takengon	Jl. Lintang, Kabeayakan, Takengon Tim., Lut. Tassar, Kabupaten Aceh Tengah, Aceh 24519	Areal A	64324385	96.851900	4.626920	1656 Km	Aceh	 
2.	Suzuki Meulaboh Armada Banda Jaya	Gampong, Jl. Meulaboh-Tapakuan No.99, Ujong Drien, Meureubo, Kabupaten Aceh Barat, Aceh 24251, Indonesia	Areal A	7007697	96.147186	4.151358	1673 Km	Aceh	 
3.	Suzuki Trans Sumatera Agung SM Raja	Jalan Sisingamangaraja, Harjosari II, Medan Ampas, Harjosari I, Kec. Medan Ampas, Kota Medan, Sumatera Utara 20148, Indonesia	Areal A	7873699	98.709465	3.538332	1428 Km	Sumut	 
4.	PT. Sunindo Veria Motor Gemilang	Jl. H. Adam Malik No.101, Silalasi, Kec. Medan Bar., Kota Medan, Sumatera Utara 20234, Indonesia	Areal A	6618485	98.668753	3.802603	1436 Km	Sumut	 
5.	Suzuki Selamat Motor	Jl. Moh. Yamin, Belakong Tangsi, Kec. Padang Bar., Kota Padang, Sumatera Barat, Indonesia	Areal A	6777889	100.357544	-0.951614	946 Km	Sumbar	 
6.	Suzuki Elang Perkasa Motor	Jl. Khatib Sulaiman No.87, Ulak Karang Utara, Kec. Padang Utara, Kota Padang, Sumatera Barat 25137, Indonesia	Areal A	6789909	100.352406	-0.904592	980 Km	Sumbar	 
7.	Distributor A	Jl. Chairil Anwar, RT.004/RW.009, Margahayu, Kec. Bekasi Tim., Kota Bekasi, Jawa Barat 17113	Areal B	456690	107.009380	-6.257530	3 Km		 

Gambar V.64 Perhitungan Fungsi *Euclidean Distance* pada Program  
Sumber: Hasil Analisis (2020)

## BAB VI

### PENUTUP

#### 6.1 Kesimpulan

Berdasarkan hasil pengumpulan, pengolahan dan analisis data yang telah dilakukan di PT Suzuki Indomobil Motor *Plant I Tambun* pada Departemen *Logistic 2W* yang berfokus pada proses pendistribusian unit dengan membangun sistem informasi pendistribusian unit pada perusahaan tersebut maka dapat diambil kesimpulan sebagai berikut:

1. Sistem informasi pendistribusian unit berdasarkan permasalahan yang ditemukan di PT Suzuki Indomobil Motor *Plant I Tambun* mampu membuat penyebaran informasi berupa *note* ketersediaan unit dan *note* surat jalan yang diterima oleh karyawan secara langsung dan saling terintegrasi antar bagian terkait.
2. Adanya suatu usulan yang dibuat dalam proses pendistribusian produk kepada distributor mampu memberikan saran rute terpendek yang harus dilalui dengan mengimplementasikan metode Algoritma *Dijkstra* dan fungsi *Euclidean Distance* untuk mencegah keterlambatan dalam proses pendistribusian yang terbukti dapat meminimalisir jarak tempuh sampai tujuan sehingga lebih efisien baik dalam hal jarak maupun biaya.
3. Sistem informasi pendistribusian unit yang mudah dalam pengaksesan informasi dan dapat mengintegrasikan data antar bagian sangat membantu mempersingkat waktu proses pengerjaan laporan gudang dan proses lainnya yang terkait dalam pendistribusian unit.

#### 6.2 Saran

Adapun saran yang diberikan untuk pengembangan sistem informasi pendistribusian unit adalah sebagai berikut:

1. Untuk pengembangan sistem selanjutnya dapat ditambahkan fitur mendapatkan titik koordinat perusahaan dengan titik koordinat distributor

sebagai lokasi pengantaran produk dengan cara otomatis menggunakan *Google Maps*.

2. Pengembangan sistem selanjutnya juga dapat ditambahkan fitur lacak proses pendistribusian yang bertujuan mengetahui titik lokasi saat melakukan pengantaran produk.
3. Melakukan pengembangan aplikasi dengan cara bekerja sama dengan Bagian Ekspedisi yang merupakan pihak ketiga dalam perusahaan, sehingga dapat memudahkan proses pendistribusian produk.
4. Melakukan pemeliharaan aplikasi secara berkala sehingga aplikasi dapat berjalan dengan baik.

## DAFTAR PUSTAKA

- Arif. 2018. *Supply Chain Management*. Jakarta: deepublish.
- Asmoko H. 2013. *Teknik Ilustrasi Masalah-Fishbone Diagrams*. Magelang (ID): Pusdiklat SDM Balai Penelitian dan Pendidikan Keuangan.
- Brandy, M., & Loonam, J. (2010). *Exploring the use of entity relationship diagramming as a technique to support grounded theory inquiry*. Bradford: Emerald Group.
- Dennis, et. all. 2015. *System Analysis & Design An Object-Oriented Approach With UML*. USA: Library of Congress Cataloging in Publication Data.
- Elsa, et. all. 2019 *Aplikasi Perhitungan Dan Visualisasi Jarak Terpendek Berdasarkan Data Coordinate Dengan Algoritma Dijkstra Dalam Kasus Pengantaran Barang Di Kawasan Jabodetabek*: Universitas Bunda Mulia
- Hadiguna, A. dan Setiawan, H. 2008. *Tata Letak Pabrik*. Yogyakarta: ANDI
- Hutahean, J. 2014. *Konsep Sistem Informasi*. Yogyakarta: Penerbit Deepublish.
- Jogiyanto, H. 2010. *Analisis dan Desain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi.
- Jogiyanto, H. 2013. *Analisis dan Desain Sistem Informasi*. Yogyakarta: Andi
- Kusrini. 2007. *Strategi Perancangan dan Pengolahan Basis Data*. Yogyakarta: Andi
- Kuswadi dan Erna. 2004. *Delapan Langkah dan Tujuh Alat Statistik Untuk Peningkatan Mutu Berbasis Komputer*. Jakarta: PT Elex Media Komputerindo
- Kotler, P. dan Keller, K.L. 2009. *Manajemen Pemasaran*. Jakarta: Erlangga.
- Marsudi, M. 2016. *Teori Graf*. Malang: UB Press,
- Maniah dan Dini. 2006. *Analisis dan Perancangan Sistem Informasi: Pembahasan Secara Praktis dengan Contoh Kasus*. Jakarta: Deepublish
- Mulyanto, A. (2009). *Sistem Informasi Konsep dan Aplikasi*. Yogyakarta: Pustaka Pelajar.

- Mulyani. 2016. *Metode Analisis dan Perancangan Sistem*. Bandung: Abdi Sistematika.
- Muslihudin dan Oktafiano. 2016. *Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML*. Yogyakarta: Penerbit Andi.
- Mustaqbal, M. S., Firdaus, R. F., & Rahmadi, H. (2015). Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN). *Jurnal Ilmiah Teknologi Informasi Terapan*, 31-36.
- Nugroho. 2018. *Sistem Informasi Manajemen; Tinjauan Praktisi Teknologi Informasi*. Yogyakarta: Teknosain.
- Nurhayati. 2010. *Manajemen Proyek*. Yogyakarta: Graha Ilmu.
- Pandiangan, S. (2017). *Operasional Manajemen Pergudangan*. Jakarta: Mitra Wacana Media.
- Pressman. 2012. *Rekayasa Perangkat Lunak*. Yogyakarta: Andi.
- PT Suzuki Indomobil Motor. *Company Profile*. Jakarta: Suzuki Indomobil Motor.
- Purnomo, H. (2004). *Perencanaan dan Perancangan Fasilitas*. Yogyakarta: Graha Ilmu.
- Rudyanto dan Arifin. 2010. "Seminar Nasional Aplikasi Teknologi Informasi "dalam Penerapan Metode Earliest Due Date Pada Penjadwalan Produksi Paving Pada CV. Eko Joyo. Yogyakarta: Sekolah Tinggi Manajemen Informatika dan Teknik Komputer.
- Rosa dan Shalahuddin. 2015. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika Bandung.
- Satzinger, dkk. 2012. *Systems Analysis and Design in a Changing World*. New York: Cengage Learning.
- Simamora dan Parulian. 2018. *Implementasi Algoritma Dijkstra dalam Pencarian Rute Terpendek Tempat Wisata Kota Medan*. Medan: Universitas Sumatera Utara
- Supono dan Putrarama. 2016. *Pemrograman Web Dengan Menggunakan PHP dan Framework CI*. Yogyakarta: Deepublish.

- Sutanta, Edhy. 2011. *Basis Data Dalam Tinjauan Konseptual*. Yogyakarta: Penerbit Andi.
- Sukamto dan Shalahuddin. 2013. *Rekayasa Perangkat Lunak*. Bandung: Informatika.
- Sukamto dan Shalahuddin. 2013. *Analisa dan Desain Sistem Informasi*. Yogyakarta: Andi.
- Suryanto. 2016. *Sistem Operasional Manajemen Distribusi*. Jakarta: Percetakan PT Gramedia.
- Sutabri. 2012. *Analisis Sistem Informasi*. Yogyakarta: Andi.
- Sutaji. 2012. *Sistem Inventory Mini Market Dengan PHP dan JQuery*. Yogyakarta: Andi.
- Tanuwijaya, H., & Setyawan, H. B. (2012). *Manajemen Produksi dan Operasi*. Surabaya: STIKOM Surabaya.
- Theodor & Thies. 2012. *Secure Software Design*. USA: Library of Congress Cataloging in Publication Data.
- Timotius. 2017. *Pengantar Metodologi Penelitian: Pendekatan Manajemen Pengetahuan untuk Perkembangan Pengetahuan*. Yogyakarta: Andi.
- Warman, J. (2010). *Manajemen Pergudangan*. Jakarta: Pustaka Sinar Harapan
- Winarno, E., Zaki, A., & Community, S. (2013). *Buku Sakti Pemrograman PHP*. Semarang: PT Elex Media Komputindo.
- Yunarto, H. I., & Santika, M. G. (2005). *Business Concept Implementation Series in Inventory Management*. Jakarta: Elex Media.
- Yakub. (2012). *Pengantar Sistem Informasi*. Yogyakarta: Graha Ilmu.
- Worldfram Mathworld. Distance. Dipetik April 20, 2020, dari <https://mathworld.wolfram.com/Distance.html>
- Wikipedia Indonesia. Euclidean Distance. Dipetik Maret 16, 2020, dari [http://en.wikipedia.org/wiki/Euclidean\\_distance](http://en.wikipedia.org/wiki/Euclidean_distance)
- Wikipedia Indonesia. Garis Lintang. Dipetik April 20, 2020, dari <https://translate.google.com/translate?u=https://en.wikipedia.org/wiki/Latitude&hl=id&sl=en&tl=id&client=srp>

Wikipedia Indonesia. Garis Bujur. Dipetik April, 2020, dari <https://translate.google.com/translate?u=https://en.wikipedia.org/wiki/Latitude&hl=id&sl=en&tl=id&client=srp>

## LAMPIRAN A

### KODE PROGRAM

#### a. Controller

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

class Rute extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
        $this->load->model(['distribut_m', 'rute_m']);
        date_default_timezone_set("Asia/Jakarta");
        $this->load->helper('new_helper');

        $this->rute = array();
        $this->fast_rute = array();
    }

    public function index()
    {
        $data['row'] = $this->distribut_m->get()->result();
        $this->template->load('template', 'rute/rute_data', $data);
    }

    public function view($id_distribut)
    {
        $id_distribut = $this->uri->segment(3);
```

```

$this->db->where('id_distribut', $id_distribut);
$this->db->group_by('level');
$ruteSum = $this->db->get('p_rute')->num_rows();

for ($i=0; $i < $ruteSum; $i++) {
    $this->db->where('id_distribut', $id_distribut);
    $this->db->where('level', ($i+1));
    $result = $this->db->get('p_rute')->result_array();

    array_push($this->rute, $result);
}

$getLvl = $this->db->query("SELECT level FROM p_rute WHERE
id_distribut = 15 GROUP BY level ORDER BY SUM(euclidean) ASC
LIMIT 1")->row_array();
$this->db->where('id_distribut', $id_distribut);
$this->db->where('level', $getLvl['level']);
$fastRute = $this->db->get('p_rute')->result_array();
array_push($this->fast_rute, $fastRute);

$data['rute'] = $this->rute;
$data['fast_rute'] = $this->fast_rute;

$this->template->load('template', 'rute/rute_distribut', $data);
}

public function create($id_distribut)
{
    $data['row'] = $this->distribut_m->get($id_distribut)->result();
    $data['rutes'] = $this->rute_m->get()->result_array();
    $this->template->load('template', 'rute/rute_form', $data);
}

```

```
}

public function get_full_rute($id_titik = null)
{
    if ($id_titik == null) {
        $result = $this->rute_m->get()->result_array();

        echo json_encode($result);
    } else {
        $result = $this->rute_m->get_not($id_titik)->result_array();

        echo json_encode($result);
    }
}

public function set_rute()
{
    $id_titik_rute = $_POST['id_rute'];
    $result = $this->rute_m->get($id_titik_rute)->row_array();

    echo json_encode($result);
}

public function process()
{
    $first_rute = $_POST['first_rute_choose'];

    if ($first_rute == "baru") {

        $last_rute = $_POST['last_rute_choose'];
```

```
if ($last_rute == "akhir") {

    $data2 = array(
        'id_distribut' => $_POST['id_distribut'],
        'first_iter' => 0,
        'next_iter' => 1,
        'euclidean' => getDistanceBetween($_POST['lat_awal'],
        $_POST['long_awal'], $_POST['lat_akhir'], $_POST['long_akhir'], "Km"),
        'level' => $_POST['level']
    );

    $this->db->insert('p_rute', $data2);

} else {

    $data2 = array(
        'id_distribut' => $_POST['id_distribut'],
        'first_iter' => 0,
        'next_iter' => $last_rute,
        'euclidean' => getDistanceBetween($_POST['lat_awal'],
        $_POST['long_awal'], $_POST['lat_akhir'], $_POST['long_akhir'], "Km"),
        'level' => $_POST['level']
    );

    $this->db->insert('p_rute', $data2);

}
} else {
    $last_rute = $_POST['last_rute_choose'];
```

```
if ($last_rute == "akhir") {

    $data2 = array(
        'id_distribut' => $_POST['id_distribut'],
        'first_iter' => $first_rute,
        'next_iter' => 1,
        'euclidean' => getDistanceBetween($_POST['lat_awal'],
$_POST['long_awal'], $_POST['lat_akhir'], $_POST['long_akhir'], "Km"),
        'level' => $_POST['level']
    );

    $this->db->insert('p_rute', $data2);

} else {

    $data2 = array(
        'id_distribut' => $_POST['id_distribut'],
        'first_iter' => $first_rute,
        'next_iter' => $last_rute,
        'euclidean' => getDistanceBetween($_POST['lat_awal'],
$_POST['long_awal'], $_POST['lat_akhir'], $_POST['long_akhir'], "Km"),
        'level' => $_POST['level']
    );

    $this->db->insert('p_rute', $data2);

}

}

redirect('rute/create/' . $_POST['id_distribut']);
```

```

    }
}

/* End of file Rute.php */

```

#### b. Model

```

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

class Rute_m extends CI_Model {

    public function get($id=null)
    {
        if ($id != null) {
            $this->db->where('id_titik_rute', $id);
        }
        return $this->db->get('titik_rute');
    }

    public function get_not($id=null)
    {
        if ($id != null) {
            $this->db->where('id_titik_rute !=', $id);
        }
        return $this->db->get('titik_rute');
    }

}

/* End of file Rute_m.php */

```

## c. Views

Rute\_data.php

```

<!-- Content Header (Page header) -->
<section class="content-header">
  <div class="container-fluid">
    <div class="row mb-2">

      <div class="col-sm-6">
        <ol class="breadcrumb float-sm-right">

          </ol>
        </div>
      </div>
    </div><!-- /.container-fluid -->
  </section>

  <!-- Main content -->
  <section class="content">

    <!-- Default box -->
    <div class="card">
      <div class="card-header">
        <h3 class="card-title">Logistic 2W / CBU</h3>

        <div class="card-tools">
          <button type="button" class="btn btn-tool" data-card-
widget="collapse" data-toggle="tooltip" title="Collapse">
            <i class="fas fa-minus"></i></button>
          <button type="button" class="btn btn-tool" data-card-
widget="remove" data-toggle="tooltip" title="Remove">
            <i class="fas fa-times"></i></button>

```

```

</div>
</div>
<div class="card-body">

  <section class="content">
    <div class="box">
      <div class="box-header">
        <h3 class="box-title">Data Rute Distributor</h3>
      </div>
      <div class="box-body table-responsive">
        <table class="table table-bordered table-striped" id="table1">
          <br>
          <thead>
            <tr>
              <th>No</th>
              <th>Name</th>
              <th>Alamat</th>
              <th>Areal</th>
              <th>No Telp</th>
              <th>Longitudinal</th>
              <th>Latitude</th>
              <th>Euclidean</th>
              <th>Actions</th>
            </tr>
          </thead>
          <tbody>
            <?php $no = 1;
            foreach($row as $key => $data) { ?>
              <tr>
                <td style="width: 5%"><?=$no++?>.</td>
                <td><?=$data->name?></td>

```

```

<td><?=$data->alamat?></td>
<td><?=$data->areal_name?></td>
<td><?=$data->no_telp?></td>

<td><?=$data->longitudinal?></td>
<td><?=$data->latitude?></td>
<td><?=$data->euclidean?> Km</td>
<td class="text-center" width="160px">
  <a href="<?=$site_url('rute/view/'.<?=$data->id_distribut)?>"
class="btn btn-primary btn-xs">
  <i class="fa fa-pencil"></i> View Rute
  </a>
  <a href="<?=$site_url('rute/create/'.<?=$data->id_distribut)?>"
class="btn btn-success btn-xs">
  <i class="fa fa-pencil"></i> Create Rute
  </a>
</td>
</tr>
<?php
} ?>
</tbody>
</table>
</section>

<div class="col-md-6 text-right">
  </div>
</div>
<!-- /.card-body -->

</div>

```

```

<!-- /.card -->
</section>
<!-- /.content -->

```

#### Rute\_Distribut.php

```

<!-- Content Header (Page header) -->
<section class="content-header">
  <div class="container-fluid">
    <div class="row mb-2">

      <div class="col-sm-6">
        <ol class="breadcrumb float-sm-right">

          </ol>
        </div>
      </div>
    </div><!-- /.container-fluid -->
  </section>

  <!-- Main content -->
  <section class="content">
    <!-- Default box -->
    <div class="card">
      <div class="card-header">
        <h3 class="card-title">Logistic 2W / CBU</h3>

        <div class="card-tools">
          <button type="button" class="btn btn-tool" data-card-
widget="collapse" data-toggle="tooltip" title="Collapse">
            <i class="fas fa-minus"></i></button>

```

```

        <button type="button" class="btn btn-tool" data-card-
widget="remove" data-toggle="tooltip" title="Remove">

```

```

        <i class="fas fa-times"></i></button>

```

```

    </div>

```

```

</div>

```

```

    <div class="card-body">

```

```

        <div class="pull-right">

```

```

        <a href="<?=site_url('rute')?>" class="btn btn-warning btn-flat">

```

```

        <i class="fa fa-undo"></i>Back

```

```

    </a>

```

```

</div>

```

```

<br>

```

```

        <button id="fast-rute" class="btn btn-sm btn-primary mb-
3">Tampilkan Rute Terpendek</button>

```

```

        <a href="<?php echo base_url('cetak/rute/'.&this->uri-
>segment(3))?>" class="btn btn-sm btn-success mb-3">Download
Rute</a>

```

```

<div id="fast-rute-container" style="display:none">

```

```

    <h4>Fast Rute</h4>

```

```

    <table class="table table-striped table-bordered">

```

```

        <thead class="thead-dark">

```

```

            <tr>

```

```

                <th>No</th>

```

```

                <th>Rute Awal</th>

```

```

                <th>Rute Selanjutnya</th>

```

```

                <th width="150px">Jarak</th>

```

```

            </tr>

```

```

        </thead>

```

```

        <tbody>

```

```

            <?php

```

```

$jarak = 0;
$first_step = "";
$next_step = "";
for ($i=0; $i < count($fast_rute[0]); $i++) {
    $jarak += $fast_rute[0][$i]['euclidean'];

    if ($fast_rute[0][$i]['first_iter'] == 0) {
        $first_step = "PT Suzuki Indomobil Motor Plant I
Tambun";
    } else {
        $first_step =
getTitikHelper($fast_rute[0][$i]['first_iter']);
    }

    if ($fast_rute[0][$i]['next_iter'] == 1) {
        $next_step = "Salsai";
    } else {
        $next_step =
getTitikHelper($fast_rute[0][$i]['next_iter']);
    }
}
?>
<tr>
<td><?php echo ($i+1); ?></td>
<td><?php echo $first_step; ?></td>
<td><?php echo $next_step; ?></td>
<td><?php echo $fast_rute[0][$i]['euclidean']; ?></td>
</tr>
<?php } ?>
<tr>
<td colspan="3"><strong>Total Jarak</strong></td>

```

```

        <td><strong><?php      echo      $jarak;      ?>
Km</strong></td>
    </tr>
</tbody>
</table>
</div>
<h4>Full Rute</h4>
<?php
    for ($i=0; $i < count($rute); $i++) {
?>
<table class="table table-striped table-bordered mb-3">
    <thead class="thead-dark">
        <tr>
            <th>No</th>
            <th>Rute Awal</th>
            <th>Rute Selanjutnya</th>
            <th width="150px">Jarak</th>
        </tr>
    </thead>
    <tbody>
        <?php
            $jarak = 0;
            $first_step = "";
            $next_step = "";
            for ($j=0; $j < count($rute[$i]); $j++) {
                $jarak += $rute[$i][$j]['euclidean'];

                if ($rute[$i][$j]['first_iter'] == 0) {
                    $first_step = "PT Suzuki Indomobil Motor Plant I
Tambun";

                } else {

```

```

        $first_step =
getTitikHelper($rute[$i][$j]['first_iter']);
    }

    if ($rute[$i][$j]['next_iter'] == 1) {
        $next_step = "Salsai";
    } else {
        $next_step =
getTitikHelper($rute[$i][$j]['next_iter']);
    }
?>
<tr>
    <td><?php echo ($j+1); ?></td>
    <td><?php echo $first_step; ?></td>
    <td><?php echo $next_step; ?></td>
    <td><?php echo $rute[$i][$j]['euclidean']; ?></td>
</tr>
<?php } ?>
<tr>
    <td colspan="3"><strong>Total Jarak</strong></td>
    <td><strong><?php echo $jarak; ?> Km</strong></td>
</tr>
</tbody>
</table>
<?php } ?>
</div>
</div>
</section>

```

Rute\_form.php

```

<!-- Content Header (Page header) -->
<section class="content-header">
  <div class="container-fluid">
    <div class="row mb-2">

      <div class="col-sm-6">
        <ol class="breadcrumb float-sm-right">

          </ol>
        </div>
      </div>
    </div><!-- /.container-fluid -->
  </section>
  <!-- Main content -->
  <section class="content">

    <!-- Default box -->
    <div class="card">
      <div class="card-header">
        <h3 class="card-title">Logistic 2W / CBU</h3>

        <div class="card-tools">
          <button type="button" class="btn btn-tool" data-card-
widget="collapse" data-toggle="tooltip" title="Collapse">
            <i class="fas fa-minus"></i></button>
          <button type="button" class="btn btn-tool" data-card-
widget="remove" data-toggle="tooltip" title="Remove">
            <i class="fas fa-times"></i></button>
        </div>
      </div>
      <div class="card-body">

```

```

<section class="content">
  <div class="box">
    <div class="box-header">
      <h3 class="box-title"> Create Rute </h3>
      <div class="pull-right">
        <a href="<?=site_url('rute')?>" class="btn btn-warning
btn-flat">
          <i class="fa fa-undo"></i>Back
        </a>
      </div>
    </div>
    <div class="box-body">
      <br>
      <div class="row">
        <div class="col-md-4 col-md-offset-4">
          <form      action="<?=site_url('rute/process')?>"
method="post">
            <input  type="hidden"  name="id_distribut"
id="id_distribut" value="<?php echo $this->uri->segment(3)?>">
            <div class="form-group" id="sec_rute_choose">
              <label>Rute Ke : </label>
              <?php
                $this->db->where('id_distribut', $this->uri-
>segment(3));
                $this->db->group_by('level');
                $level      =      $this->db->get('p_rute')-
>num_rows();
              ?>
              <select name="level" id="level" class="form-
control" required>

```

```

        <option value="">Pilih</option>
        <?php for ($i=0; $i < ($level+1); $i++) { ?>
        <option value="<?php echo ($i+1);
?>">Rute ke-<?php echo ($i+1); ?></option>
        <?php } ?>
    </select>
</div>
<div class="form-group" id="sec_rate_choose">
    <label>Pilih Rute Awal *</label>
    <select
        name="first_rate_choose"
id="first_rate_choose" class="form-control" required>
        <option value="">Pilih</option>
        <option value="baru">PT Suzuki Indomobil
Motor Plant I Tambun</option>
        <?php foreach ($rutes as $key => $value):
?>

            <option
                value="<?php
                echo
                $value['id_titik_rute']; ?>"><?php echo "(" . $value['id_titik_rute'] . ") - " .
                $value['alamat']; ?></option>

            <?php endforeach; ?>
        </select>
    </div>
<div class="form-group">
    <label>Alamat Awal</label>
    <input
        type="text"
        name="titik_awal"
id="titik_awal" class="form-control" readonly>
    </div>
<div class="form-group">
    <label>Long Awal</label>

```

```

        <input type="text" name="long_awal"
id="long_awal" class="form-control" readonly>
    </div>
    <div class="form-group">
        <label>Lat Awal</label>
        <input type="text" name="lat_awal"
id="lat_awal" class="form-control" readonly>
    </div>

    <div class="form-group"
id="sec_last_rute_choose" style="display:none">
        <label>Pilih Rute Akhir *</label>
        <select name="last_rute_choose"
id="last_rute_choose" class="form-control">
            </select>
    </div>
    <div class="form-group">
        <label>Alamat Akhir</label>
        <input type="text" name="titik_akhir"
id="titik_akhir" class="form-control" readonly>
    </div>
    <div class="form-group">
        <label>Long Akhir</label>
        <input type="text" name="long_akhir"
id="long_akhir" class="form-control" readonly>
    </div>
    <div class="form-group">
        <label>Lat Akhir</label>
        <input type="text" name="lat_akhir"
id="lat_akhir" class="form-control" readonly>
    </div>

```



```

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></s
cript>
<script>
$(document).ready(function(){

    $('#first_rute_choose').on('change', function () {
        var id_rute = $(this).val();

        $('#titik_akhir').val("");
        $('#long_akhir').val("");
        $('#lat_akhir').val("");

        if (id_rute == "baru") {
            $('#titik_awal').val("PT Suzuki Indomobil Motor Plant I
Tambun");
            $('#long_awal').val("<?php echo $row[0]->long_suzuki; ?>");
            $('#lat_awal').val("<?php echo $row[0]->lat_suzuki; ?>");

            $.ajax({
                url: '<?php echo base_url(); ?>rute/get_full_rute',
                type : 'GET',
                async : true,
                dataType : 'json',
                success: function (data) {
                    $('#sec_last_rute_choose').show();

                    var html = '<option value="">Pilih</option><option
value="akhir"><?php echo $row[0]->name; ?></option>';
                    for(i=0; i<data.length; i++) {

```

```

        html += '<option value="'+data[i].id_titik_rute+'>(' +
data[i].id_titik_rute + ') - ' + data[i].alamat + '</option>';
    }

    $('#last_rute_choose').html(html);
}
});
} else {
$.ajax({
    url: '<?php echo base_url(); ?>rute/set_rute',
    type: 'POST',
    data: {id_rute:id_rute},
    success: function (data) {
        var id_titik = $("#first_rute_choose").val();
        var json = $.parseJSON(data);

        $('#titik_awal').val(json.alamat);
        $('#long_awal').val(json.longitudinal);
        $('#lat_awal').val(json.latitude);

$.ajax({
    url: '<?php echo base_url(); ?>rute/get_full_rute/' + id_titik,
    type : 'GET',
    async : true,
    dataType : 'json',
    success: function (data) {
        $('#sec_last_rute_choose').show();

        var html = '<option value="">Pilih</option><option
value="akhir"><?php echo $row[0]->name; ?></option>';
        for(i=0; i<data.length; i++) {

```

```

        html += '<option value="'+data[i].id_titik_rute+'>(' +
data[i].id_titik_rute + ') - ' + data[i].alamat + '</option>';
    }

```

```

        $('#last_rute_choose').html(html);
    }
});
}
});
}
});

```

```

$('#last_rute_choose').on('change', function () {
    var id_rute = $(this).val();

```

```

    if (id_rute == "akhir") {
        $('#titik_akhir').val("<?php echo $row[0]->alamat; ?>");
        $('#long_akhir').val("<?php echo $row[0]->longitudinal; ?>");
        $('#lat_akhir').val("<?php echo $row[0]->latitude; ?>");

```

```

        set_jarak();
    } else {
        $.ajax({
            url: '<?php echo base_url(); ?>route/set_rute',
            type: 'POST',
            data: {id_rute:id_rute},
            success: function (data) {
                var id_titik = $("#first_rute_choose").val();
                var json = $.parseJSON(data);

                $('#titik_akhir').val(json.alamat);

```

```

        $('#long_akhir').val(json.longitudinal);
        $('#lat_akhir').val(json.latitude);

        set_jarak();
    }
});
}
});

```

```

function set_jarak() {
    var first_rute = $('#first_rute_choose').val();
    var last_rute = $('#last_rute_choose').val();

    if (first_rute && last_rute) {
        var origin_lat = $('#lat_awal').val();
        var origin_long = $('#long_awal').val();
        var destination_lat = $('#lat_akhir').val();
        var destination_long = $('#long_akhir').val();

        $.ajax({
            url:
'https://dev.virtualearth.net/REST/v1/Routes/DistanceMatrix?origins='+ori
gin_lat+','+origin_long+'&destinations='+destination_lat+','+destination_l
ong+'&travelMode=driving&key=AiRLzVfcwEBtjMLTjsEcOdA0V-
_NBwEZggalw14pXqZjyZJt3Uh8Vynsvn9A_01z',
            type : 'GET',
            async : true,
            dataType : 'json',
            success: function (data) {
                var                travelDistance                =
data['resourceSets'][0]['resources'][0]['results'][0]['travelDistance']

```

```
        $('#jarak').val(travelDistance)
    }
});
}
}

});
</script>
```

## LAMPIRAN B

### TAMPILAN PROGRAM

#### 1. Tampilan *Login*



#### 2. Tampilan *Dashboard*



### 3. Tampilan Data User

No	Username	Level	Action
1	Delivery001	delivery	Read   Update   Delete
2	Kepala Gudang	kepala_gdg	Read   Update   Delete
3	Receiving001	receiving	Read   Update   Delete
4	Receiving002	receiving	Read   Update   Delete
5	AdmSU001	admin_uj	Read   Update   Delete
6	AdmCdG001	admin_gdg	Read   Update   Delete
7	AdmSU002	admin_uj	Read   Update   Delete

### 4. Tampilan Form Create User

**User Create**

**Username**  
UserCreate

**Nama**  
Nama

**Password**  
Password

**Alamat**  
Alamat

**Level**  
Level

### 5. Tampilan User Read

Username	Nama	Password	Alamat	Level
Delivery001	Hary Cahya	12345	Peringgip	delivery

## 6. Tampilan Data *Categories*

No.	Name	Actions
1.	UK 120 NE	[Edit] [Delete]
2.	UD110NEX	[Edit] [Delete]
3.	GSK-S130	[Edit] [Delete]
4.	GSK-P150	[Edit] [Delete]
5.	FU 150	[Edit] [Delete]

## 7. Tampilan Form *Add Category*

Logistik 2W / CBU

**Add Category**

+ Create

Search:

Category Name \*

[Submit]

## 8. Tampilan Data *Warna*

No.	Name	Actions
1.	Biru	[Edit] [Delete]
2.	Merah	[Edit] [Delete]
3.	Kuning	[Edit] [Delete]
4.	Hijau	[Edit] [Delete]
5.	Abu-abu	[Edit] [Delete]

9. Tampilan *Form Add Warna*

Kepala Gudang  
Hendra Pustakawan

Dashboard  
Data Master  
Data Transaksi  
Gallery  
Logout

Logistic 2W

**Add Warna**

Save

Input Warna \*

10. Tampilan *Product Items*

Kepala Gudang  
Hendra Pustakawan

Dashboard  
Data Master  
Data Transaksi  
Gallery  
Logout

Logistic 2W

**Product Items**

+ Add Product Item

Show 5 entries

Search:

No.	No_Seri	Name	Category	Warna	Stock	Note	Actions
1.	MO03	Suzuki Address 110	UK 110 NE	Biru	15		View Edit
2.	Z002	Suzuki Sport	GSX S150	Biru	14		View Edit
3.	MO01	Suzuki Address 110	UK 110 NE	Merah	30		View Edit
4.	Z001	Suzuki Sport	GSX S150	Merah	0		View Edit
5.	Y001	Suzuki Sport	GSX-R150	Merah	12		View Edit

11. Tampilan *Form Add Items*

Kepala Gudang  
Hendra Pustakawan

Dashboard  
Data Master  
Data Transaksi  
Gallery  
Logout

Logistic 2W / CBU

**Add Items**

Save

No Seri \*

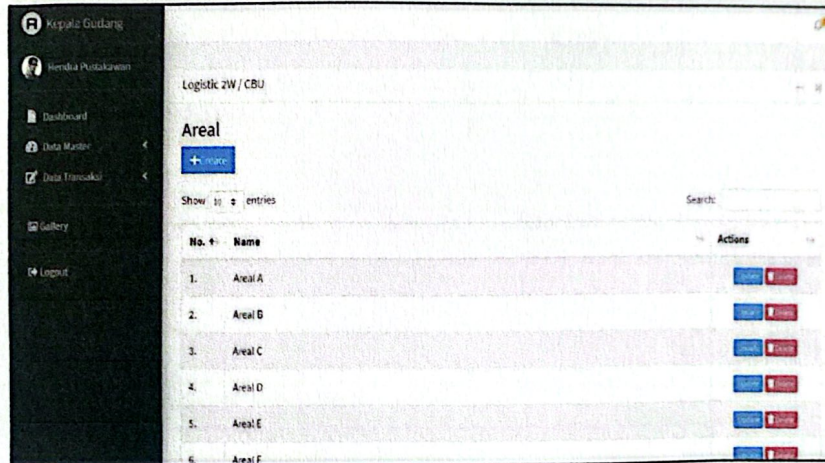
Product Name \*

Category \*

Warna \*

Note

## 12. Tampilan Data Areal



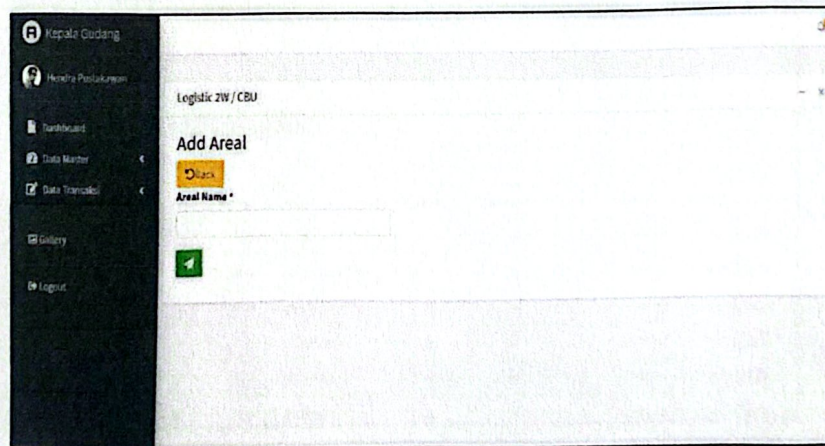
Logistic ZW / CBU

**Areal**

Show 10 entries

No.	Name	Actions
1.	Areal A	<a href="#">Add</a> <a href="#">Edit</a> <a href="#">Delete</a>
2.	Areal B	<a href="#">Add</a> <a href="#">Edit</a> <a href="#">Delete</a>
3.	Areal C	<a href="#">Add</a> <a href="#">Edit</a> <a href="#">Delete</a>
4.	Areal D	<a href="#">Add</a> <a href="#">Edit</a> <a href="#">Delete</a>
5.	Areal E	<a href="#">Add</a> <a href="#">Edit</a> <a href="#">Delete</a>
6.	Areal F	<a href="#">Add</a> <a href="#">Edit</a> <a href="#">Delete</a>

## 13. Tampilan Form Add Areal



Logistic ZW / CBU

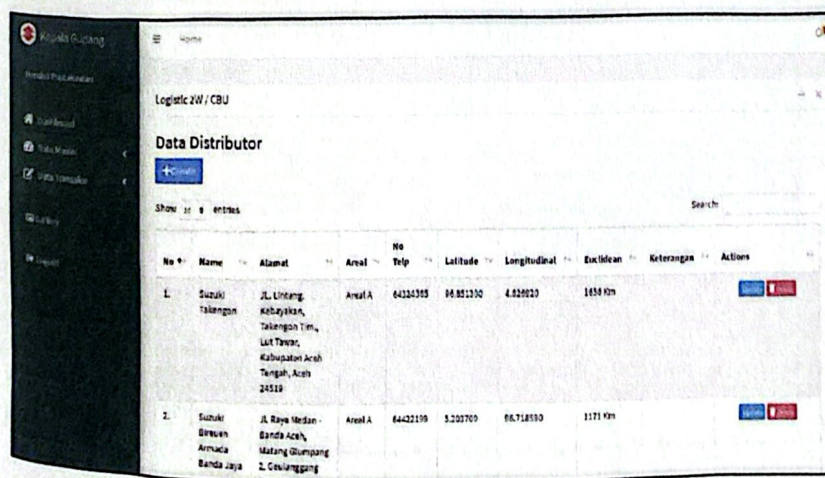
**Add Areal**

[Back](#)

Areal Name \*

[Add](#)

## 14. Tampilan Data Distributor



Logistic ZW / CBU

**Data Distributor**

Show 20 entries

No.	Name	Alamat	Areal	No Telp	Latitude	Longitudinal	Eucclidean	Keterangan	Actions
1.	Suzuki Talangron	Jl. Lintang, Kabayahan, Talangron Tim., Liat Tawar, Kabupaten Aceh Tengah, Aceh 24518	Areal A	64336395	04.851300	4.828923	1658 Km		<a href="#">Add</a> <a href="#">Edit</a> <a href="#">Delete</a>
2.	Suzuki Bireuen Aruacia Banda Jaya	Jl. Raya Medan - Banda Aceh, Matang Gumpang 2, Cewilanggang	Areal A	64422195	5.203700	96.718190	1171 Km		<a href="#">Add</a> <a href="#">Edit</a> <a href="#">Delete</a>

### 15. Tampilan *Add Distributor*

### 16. Tampilan Data Gudang

No.	No Seri	Name	Stock Awal	Stock Akhir	Note	Tanggal	Actions
1.	Y001	Suzuki Sprint	0	20		14/08/2020	Detail Edit
2.	M002	Suzuki Address 110	15	30		14/08/2020	Detail Edit
3.	M001	Suzuki Address 110	25	35		14/08/2020	Detail Edit
4.	A001	Suzuki Nex 110	25	35		14/08/2020	Detail Edit
5.	A001	Suzuki Nex 110	20	30		07/08/2020	Detail Edit
6.	A001	Suzuki Nex 110	17	27		07/08/2020	Detail Edit

### 17. Tampilan *Form Input Stok*

18. Tampilan Data *Delivery Order*

No	Name	Alamat	Areal	No Telp	Longitudinal	Latitude	Eudclidean	Actions
1.	Suzuki Takengon	Jl. Lintang, Kebyakan, Takengon Tim., Lut Tawar, Kabupaten Aceh Tengah, Aceh 24519	Areal A	64324585	98.851300	4.626920	1655.540 Km	<a href="#">Detail</a>
2.	Suzuki Bireuen	Jl. Raya Medan - Banda Aceh, Armada Matang Olumpang 2, Geulumpang Jaya, Kota Juang,	Areal A	64422199	96.718999	5.203700	1712.700 Km	<a href="#">Detail</a>

19. Tampilan Data *View Detail Delivery Order*

No	No DO	Distribut Name	No Seri	Item Name	Qty	Tanggal	Actions
1.	D0000199	Suzuki Takengon	MOG3	Suzuki Address 110	5	2020-09-17	<a href="#">Detail</a>
2.	D0000199	Suzuki Takengon	2002	Suzuki Sport	4	2020-09-17	<a href="#">Detail</a>

## 20. Tampilan Data Surat Jalan

No	No Surat Jalan	Nama Distributor	Periode	Tanggal Dibuat	Actions
1.	S2020091500001	Suzuki Takengon	2020-09-15 s/d 2020-09-15	2020-09-15	<a href="#">Detail</a>
2.	S2020091500003	Suzuki Bireuen Armada Banda Jaya	2020-09-15 s/d 2020-09-15	2020-09-15	<a href="#">Detail</a>
3.	S2020091700004	Suzuki Takengon	2020-09-17 s/d 2020-09-17	2020-09-17	<a href="#">Detail</a>

## 21. Tampilan Data Rute Distributor

Logistic 2W / CBU

Data Rute Distributor

Show 2 entries

No	Name	Alamat	Areal	No Telp	Longitudinal	Latitudo	Euclidean	Actions
1.	Suzuki Takemgon	Jl. Lintang, Kibanyakan, Takemgon Tim., Lut. Towar, Kabupaten Aceh Tengah, Aceh 24519	Areal A	64124585	96.851300	4.620300	1655.549 Km	<a href="#">View</a> <a href="#">Edit</a>
2.	Suzuki Eireuen Amrada Baska	Jl. Raya Medas Banda Aceh, Matang Glumpang 2, Seulanggang Baro,	Areal A	64422199	96.718590	5.203700	1712.700 Km	<a href="#">View</a> <a href="#">Edit</a>

## 22. Tampilan Form Create Rule

Logistic 2W / CBU

Create Rule

[Cancel](#)

Pilih Rute Awal \*

Pilih

Alamat Awal

Long Awal

Lat Awal

## 23. Tampilan Informasi Rute

Logistic 2W / CBU

[Tambahkan Rute Baru](#) [Hapus Rute](#)

Full Rule

No	Rute Awal	Rute Selanjutnya	Jarak
1	P1 Suzuki Indomobi Motor Plant 1 Tambun	RT.0/RW.3, Bend. Hill, Kecamatan Tanah Abang, Kota Jakarta Pusat, Daerah Khusus Ibukota Jakarta 10210	409.000
2	R.0/RW.3, Bend. Hill, Kecamatan Tanah Abang, Kota Jakarta Pusat, Daerah Khusus Ibukota Jakarta 10210	Jl. Mina Raya, RT.002/RW.012, Panunggagan Bar., Kec. Cibodas, Kota Tangerang, Banten 15138	490.930
3	Jl. Mina Raya, RT.002/RW.012, Panunggagan Bar., Kec. Cibodas, Kota Tangerang, Banten 15138	Sekeloa	1280.330
4	P1 Suzuki Indomobi Motor Plant 1 Tambun	Jl. Chari Anwar, RT.004, RW.005, Margahayu, Kec. Bekasi Tim., Kota Bekasi, Jawa Barat 17113	678.150



