

**PEMBANGUNAN SISTEM INFORMASI MANAJEMEN
PENJUALAN CV. MITRA TEKNIK BERBASIS WEB**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Mohamad Reza Mawardi
NIM: 135150201111061



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019

PENGESAHAN

PEMBANGUNAN SISTEM INFORMASI MANAJEMEN PENJUALAN CV. MITRA
TEKNIK BERBASIS WEB

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Mohamad Reza Mawardi
NIM: 135150201111061

Skripsi ini telah diuji dan dinyatakan lulus pada

24 Juli 2019

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Faizatul Amalia, S.Pd., M.Pd
NIK. 201309 860821 2 001

Dosen Pembimbing II

Djoko Pramono, S.T., M.Kom
NIP. 19780108 200501 1 002

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP. 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 24 Juli 2019



Mohamad Reza Mawardi

NIM: 135150201111061

PRAKATA

Puji syukur penulis panjatkan kehadiran Allah SWT. Karena berkat rahmat dan hidayah-Nya, penulis dapat menyelesaikan penyusunan skripsi dengan judul "Pembangunan Sistem Informasi Manajemen Penjualan CV. Mitra Teknik Berbasis Web" dengan cukup baik.

Dalam penulisan skripsi ini, penulis banyak mendapatkan bantuan, bimbingan, serta dukungan baik materil maupun moril dari berbagai pihak. Dengan selesainya dokumentasi penelitian ini, penulis mengucapkan terima kasih kepada:

1. Ibu Faizatul Amalia S.Pd., M.Pd selaku dosen pembimbing satu, dan Bapak Djoko Pramono S.T., M.Kom selaku dosen pembimbing dua yang telah meluangkan waktu dan pikiran dengan memberikan pengarahan dan bimbingan bagi penulis.
2. Saudara Adam Kukuh Kurniawan, memberikan bantuan dan kerjasamanya dalam penggerjaan penelitian ini.
3. Bapak Sunarto S.T., selaku kepala perusahaan CV. Mitra Teknik yang telah bersedia untuk diwawancara untuk keperluan penggerjaan penelitian ini.
4. Fariz Setyawan, Teddy Syahrial, Abdul Aziz, Ari Nurdin, selaku validator yang telah meluangkan waktu untuk menguji sistem dalam pelaksanaan penelitian ini.
5. Kedua orang tua serta keluarga penulis atas segala doa, perhatian, bimbingan, kesabaran dan dukungan dalam penggerjaan penelitian
6. Segenap bapak ibu dosen dan seluruh staf dan karyawan Fakultas Ilmu Komputer Universitas Brawijaya atas dukungan dan kerjasamanya.
7. Sahabat dan teman seperjuangan Fakultas Ilmu Komputer yang tak bisa disebutkan satu persatu, atas segala bantuan dan dukungan pada penulis dalam pelaksanaan penelitian ini.

Malang, 24 Juli 2019

Mohamad Reza Mawardi
rezamawardi18@gmail.com

ABSTRAK

Mohamad Reza Mawardi, Pembangunan Sistem Informasi Manajemen Penjualan CV. Mitra Teknik Berbasis Web

Dosen Pembimbing: Faizatul Amalia S.Pd., M.Pd dan Djoko Pramono, S.T., M.Kom

Persaingan semakin kompetitif dalam lingkungan industri saat ini Perusahaan harus mampu mencari dan mengidentifikasi faktor-faktor yang menjadi pengaruh kunci untuk meraih kesuksesan. Perkembangan teknologi sangat berpengaruh pada kekuatan sebuah perusahaan untuk berkompetisi dalam iklim persaingan antar kompetitor. CV. Mitra Teknik merupakan perusahaan yang bergerak dalam bidang manufaktur yang melayani jasa, penjualan, perbaikan material mesin industri. Dalam menjalankan bisnisnya CV. Mitra Teknik memerlukan sistem yang mampu untuk mengelola data dan informasi dalam perusahaan untuk menjalankan roda industrinya. Namun, saat ini proses bisnis yang berjalan seperti pencatatan setiap transaksi masih tercatat manual dalam buku perusahaan, hal ini dapat mengganggu operasional sehari-hari dalam perusahaan. Solusi untuk menyelesaikan permasalahan yaitu dengan dibuatnya sistem informasi manajemen penjualan. Dengan dibangunnya sistem informasi penjualan diharapkan mampu memberikan kemudahan untuk pengguna dalam proses pengolahan data sehingga dapat menghemat waktu dalam menjalankan bisnis proses perusahaan. Dari pengujian yang dilakukan dengan pengujian *white-box*, pengujian tersebut mendapatkan tingkat keberhasilan 100%, untuk pengujian *black-box* yang menguji validasi sistem berhasil didapatkan nilai 100%. Dan hasil dari pengujian *User Acceptance Testing* (UAT) dalam perhitungan yang telah dilakukan oleh 4 validator didapatkan nilai diatas 75% yang menurut tabel skala penilaian sistem dinyatakan mampu dipahami oleh pengguna.

Kata kunci: Manajemen, Sistem Informasi, Sistem Informasi Manajemen Penjualan, *Waterfall*, *Web Service*.

ABSTRACT

Mohamad Reza Mawardi, Web Based Sales Management Information System Development of CV. Mitra Teknik

Supervisors: Faizatul Amalia S.Pd., M.Pd and Djoko Pramono, S.T., M.Kom

Competition is increasingly competitive in today's industrial environment. Companies must be able to find and identify the factors that are the key influences for success. The development of technology is very influential on the strength of a company to compete in a climate of competition between competitors. CV. Mitra Teknik is a company engaged in manufacturing that serves a services, sales, repair of machinery materials industry. In running this business, CV. Mitra Teknik need a system that can manage data and information within a company to run its industrial wheels. However, current business processes that are running such as recording each transaction are still recorded manually in the company book, this can disrupt the daily operations of the company. The solution to solve the problem is by making a sales management information system. With the construction of the sales information system, it is expected to be able to provide convenience for users in data processing so that it can save time in running the company's business processes. From test conducted using white-box testing, the test gained 100% success rate, for black-box testing that tests the system validation successfully obtained a value of 100%. And the results of the User Acceptance Testing (UAT) in the calculation that has been done by 4 validators obtained values above 75%, which according to the system rating scale table declared capable of being understood by the user.

Keywords: *Management, Information Systems, Sales Management Information System, Waterfall, Web service.*

DAFTAR ISI

PEMBANGUNAN SISTEM INFORMASI MANAJEMEN PENJUALAN CV. MITRA TEKNIK BERBASIS WEB.....	I
PENGESAHAN	II
PERNYATAAN ORISINALITAS	III
PRAKATA.....	IV
ABSTRAK	V
ABSTRACT.....	VI
DAFTAR ISI.....	VII
DAFTAR TABEL.....	XII
DAFTAR GAMBAR	XV
DAFTAR LAMPIRAN	XVII
BAB 1 PENDAHULUAN	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan penelitian	2
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Tinjauan Pustaka.....	5
2.2 Landasan Teori.....	5
2.2.1 Sistem Informasi.....	5
2.2.2 Komponen Sistem Informasi.....	6
2.3 Rekayasa Perangkat Lunak	7
2.3.1 <i>Waterfall Model</i>	8
2.4 <i>Web Service</i>	10
2.4.1 Definisi <i>Web Service</i>	10
2.4.2 Arsitektur <i>Web Service</i>	10
2.4.3 <i>REST (Representational State Transfer)</i>	11

2.5 <i>Bussines Procces Modelling Notation (BPMN)</i>	12
2.5.1 <i>Flow Object</i>	12
2.5.2 <i>Connecting Object</i>	13
2.5.3 <i>Swimlanes</i>	13
2.5.4 <i>Artifacts</i>	13
2.6 Proses bisnis CV. Mitra Teknik	15
2.6.1 Proses bisnis sebelum dibuatnya sistem informasi penjualan ...	15
2.6.2 Proses bisnis sesudah dibangun sistem informasi penjualan	17
2.7 Teknologi Perangkat Lunak.....	19
2.7.1 <i>Cascading Style Sheet (CSS)</i>	19
2.7.2 <i>Hypertext Preprocessor (PHP)</i>	19
2.7.3 <i>Hyper Text Markup Languange (HTML)</i>	19
2.7.4 <i>Framework Model View Control (MVC)</i>	19
2.7.5 <i>Database Management System (DBMS) MySQL</i>	20
2.8 Sistem informasi penjualan	20
2.8.1 Konsep Penjualan.....	20
2.8.2 Teknologi Informasi (TI) Pada Perusahaan	22
2.8.3 Manfaat Teknologi Informasi (TI) Pada Perusahaan.....	23
2.9 Pengujian	23
2.9.1 Pengujian <i>White-Box</i>	24
2.9.2 Pengujian <i>Black-box</i>	24
2.9.3 Pengujian fungsional.....	24
2.9.4 <i>User Acceptence Testing</i>	25
2.9.5 Pengujian <i>Security</i>	26
2.9.6 Pengujian <i>Performance</i>	26
BAB 3 METODOLOGI PENELITIAN	27
3.1 Studi Literatur	28
3.2 Pengumpulan Data	28
3.3 Analisis Kebutuhan Sistem.....	28
3.4 Perancangan Sistem.....	28
3.4.1 Tahap perancangan sistem	29
3.5 Implementasi	29

3.6 Pengujian sistem	29
3.7 Kesimpulan dan Saran	30
BAB 4 ANALISIS KEBUTUHAN	31
4.1 Elisitasi Kebutuhan.....	31
4.1.1 Teknik Elisitasi	31
4.2 Identifikasi <i>Stakeholder</i>	31
4.3 Spesifikasi Kebutuhan	32
4.3.1 Kebutuhan Fungsional Sistem.....	32
4.3.2 Kebutuhan Non-Fungsional Sistem.....	36
4.4 Diagram <i>Use Case</i>	37
4.5 <i>Use Case Scenario</i>	38
4.5.1 <i>Use case scenario</i> Melihat Penawaran	38
4.5.2 <i>Use case scenario</i> Menambahkan Penawaran	38
4.5.3 <i>Use case scenario</i> Mengubah Penawaran	39
4.5.4 <i>Use case scenario</i> Mencetak Penawaran.....	39
4.5.5 <i>Use case scenario</i> Menghapus Penawaran	40
4.5.6 <i>Use case scenario</i> Melihat <i>Order</i> masuk.....	40
4.5.7 <i>Use case scenario</i> Menambahkan <i>order</i> masuk	41
4.5.8 <i>Use case scenario</i> Mengubah <i>order</i> masuk	41
4.5.9 <i>Use case scenario</i> Menghapus <i>order</i> masuk.....	42
4.5.10 <i>Use case scenario</i> Menambahkan pengiriman	42
4.5.11 <i>Use case scenario</i> Mengubah pengiriman	43
4.5.12 <i>Use case scenario</i> Menghapus pengiriman.....	43
4.5.13 <i>Use case scenario</i> Menambahkan pembelian barang	44
4.5.14 <i>Use case scenario</i> Melihat Stok Barang	44
4.5.15 <i>Use case scenario</i> Menambahkan <i>supplier</i>	45
4.5.16 <i>Use case scenario</i> Mengubah <i>supplier</i>	45
4.5.17 <i>Use case scenario</i> Menghapus <i>supplier</i>	46
4.5.18 <i>Use case scenario</i> Menambahkan pelanggan.....	46
4.5.19 <i>Use case scenario</i> Mengubah pelanggan.....	47
4.5.20 <i>Use case scenario</i> Menghapus pelanggan	47
4.5.21 <i>Use case scenario</i> Logout	48

4.5.22 Use case scenario Login	48
4.5.23 Use case scenario Menambahkan user.....	49
4.5.24 Use case scenario Melihat Pengiriman	49
4.5.25 Use case scenario Melihat Supplier.....	50
4.5.26 Use case scenario Melihat Pelanggan.....	50
4.5.27 Use case scenario Melihat Laporan Pengiriman	50
4.5.28 Use case scenario Melihat Laporan Pembelian	51
4.5.29 Use case scenario Melihat Laporan Pengiriman	51
4.5.30 Use case scenario Menambahkan Stok Barang	51
BAB 5 PERANCANGAN DAN IMPLEMENTASI.....	53
5.1 Perancangan	53
5.1.1 Arsitektur <i>Web Service</i>	53
5.1.2 Pemodelan <i>Sequence Diagram</i>	53
5.1.3 Pemodelan <i>Class Diagram</i>	57
5.1.4 <i>Entity Relation Diagram</i>	59
5.1.5 Perancangan Pseudocode	60
5.1.6 Perancangan Antarmuka.....	65
5.2 Implementasi	75
5.2.1 Spesifikasi Sistem	76
5.2.2 Implementasi <i>Class</i>	76
5.2.3 Implementasi Arsitektur	77
5.2.4 Implementasi Basis Data	78
5.2.5 Implementasi Pseudocode.....	78
5.2.6 Implementasi Antarmuka	84
BAB 6 PENGUJIAN DAN ANALISIS	92
6.1 Pengujian <i>White-box</i>	92
6.1.1 Pengujian Fungsi <i>Tambah_order_masuk</i>	92
6.1.2 Pengujian Fungsi <i>Edit_order_masuk</i>	94
6.1.3 Pengujian Fungsi <i>Hapus_order_masuk</i>	96
6.2 Pengujian <i>Black-box</i>	97
6.2.1 Pengujian Fungsional	97
6.2.2 Pengujian Kebutuhan Non-Fungsional.....	102

6.3 Pengujian <i>User Acceptance Testing</i> (UAT).....	104
6.4 Analisis Pengujian	111
BAB 7 PENUTUP.....	112
7.1 Kesimpulan.....	112
7.2 Saran	112
DAFTAR REFERENSI.....	113
LAMPIRAN KUISIONER <i>USER ACCEPTANCE TESTING</i> (UAT).....	115
LAMPIRAN.....	116
LAMPIRAN.....	117
LAMPIRAN.....	118

DAFTAR TABEL

Tabel 2.1 Predikat kelayakan aplikasi	25
Tabel 2.2 Skala Penilaian (Qomaruddin, 2014).....	26
Tabel 4.1 Temuan kebutuhan	31
Tabel 4.2 Identifikasi <i>Stakeholder</i>	32
Tabel 4.3 Spesifikasi kebutuhan Fungsional Sistem Informasi Penjualan	32
Tabel 4.4 Kebutuhan fungsional aktor tamu	35
Tabel 4.5 Kebutuhan non-fungsional.....	36
Tabel 4.6 Tabel <i>use case scenario</i> Melihat Penawaran	38
Tabel 4.7 Tabel <i>use case scenario</i> Menambahkan Penawaran	38
Tabel 4.8 Tabel <i>use case scenario</i> Mengubah Penawaran	39
Tabel 4.9 Tabel <i>use case scenario</i> Mencetak Penawaran.....	39
Tabel 4.10 Tabel <i>use case scenario</i> Menghapus Penawaran.....	40
Tabel 4.11 Tabel <i>use case scenario</i> Melihat <i>Order</i> masuk.....	40
Tabel 4.12 Tabel <i>use case scenario</i> Menambahkan <i>order</i> masuk	41
Tabel 4.13 Tabel <i>use case scenario</i> Mengubah <i>order</i> masuk	41
Tabel 4.14 Tabel <i>use case scenario</i> Menghapus <i>order</i> masuk.....	42
Tabel 4.15 Tabel <i>use case scenario</i> Menambahkan pengiriman	42
Tabel 4.16 Tabel <i>use case scenario</i> Mengubah pengiriman	43
Tabel 4.17 Tabel <i>use case scenario</i> Menghapus pengiriman.....	43
Tabel 4.18 Tabel <i>use case scenario</i> Menambahkan pembelian barang	44
Tabel 4.19 Tabel <i>use case scenario</i> Melihat stok barang.....	44
Tabel 4.20 Tabel <i>use case scenario</i> Menambahkan <i>supplier</i>	45
Tabel 4.21 Tabel <i>use case scenario</i> Mengubah <i>supplier</i>	45
Tabel 4.22 Tabel <i>use case scenario</i> Menghapus <i>supplier</i>	46
Tabel 4.23 Tabel <i>use case scenario</i> Menambahkan pelanggan.....	46
Tabel 4.24 Tabel <i>use case scenario</i> Mengubah pelanggan.....	47
Tabel 4.25 Tabel <i>use case scenario</i> Menghapus pelanggan	47
Tabel 4.26 Tabel <i>use case scenario</i> <i>Logout</i>	48
Tabel 4.27 Tabel <i>use case scenario</i> <i>Login</i>	48
Tabel 4.28 Tabel <i>use case scenario</i> Tambah <i>user</i>	49

Tabel 4.29 Tabel <i>use case scenario</i> Melihat data Pengiriman	49
Tabel 4.30 Tabel <i>use case scenario</i> Mengelola data <i>Supplier</i>	50
Tabel 4.31 Tabel <i>use case scenario</i> Melihat Pelanggan	50
Tabel 4.32 Tabel <i>use case scenario</i> laporan pengiriman	50
Tabel 4.33 Tabel <i>use case scenario</i> laporan pembelian.....	51
Tabel 4.34 Tabel <i>use case scenario</i> laporan <i>order</i> masuk.....	51
Tabel 4.35 Tabel <i>use case scenario</i> menambahkan stok barang.....	51
Tabel 5.1 <i>Class</i> Gudang (<i>Server</i>).....	57
Tabel 5.2 <i>Class</i> Gudang (<i>Client</i>).....	58
Tabel 5.3 Pseudocode <i>aksi_login()</i>	61
Tabel 5.4 Pseudocode <i>Tambah_order_masuk Ci_Controller</i>	62
Tabel 5.5 Pseudocode <i>index_post() class Order_masuk RESTController</i>	62
Tabel 5.6 Pseudocode <i>index_put() class Order_masuk RESTController</i>	63
Tabel 5.7 Pseudocode <i>edit_pembelian() class Pembelian Ci_Controller</i>	64
Tabel 5.8 Tabel Spesifikasi Perangkat Keras	76
Tabel 5.9 Tabel Spesifikasi Perangkat Lunak.....	76
Tabel 5.10 Implementasi <i>Class</i>	77
Tabel 5.11 Implementasi fungsi <i>aksi_login()</i>	79
Tabel 5.12 Implementasi fungsi <i>Tambah_Order_Masuk</i>	80
Tabel 5.13 Implementasi fungsi <i>index_post()</i>	81
Tabel 5.14 Implementasi fungsi <i>index_put()</i>	81
Tabel 5.15 Implementasi fungsi <i>Edit_Pembelian()</i>	83
Tabel 6.1 Algoritme fungsi <i>tambah_order_masuk</i>	92
Tabel 6.2 Pengujian unit fungsi <i>Tambah_order_masuk</i>	93
Tabel 6.3 Algoritme fungsi <i>Edit_pembelian</i>	94
Tabel 6.4 Pengujian unit fungsi <i>edit_pembelian</i>	95
Tabel 6.5 Algoritme fungsi <i>hapus_order_masuk</i>	96
Tabel 6.6 Algoritme fungsi <i>hapus_order_masuk</i>	97
Tabel 6.7 Pengujian validasi kebutuhan fungsional.....	98
Tabel 6.8 Pengujian kebutuhan non-fungsional	103
Tabel 6.9 Data tabel penilaian untuk pengguna	105
Tabel 6.10 Data tabel penilaian fitur penawaran	105

Tabel 6.11 Data tabel penilaian fitur <i>order</i> masuk	106
Tabel 6.12 Data tabel penilaian fitur pengiriman	106
Tabel 6.13 Data tabel penilaian fitur pembelian	107
Tabel 6.14 Data tabel penilaian fitur pelanggan.....	107
Tabel 6.15 Data tabel penilaian fitur <i>supplier</i>	108
Tabel 6.16 Data tabel penilaian fitur <i>login</i>	108
Tabel 6.17 Penilaian pengguna	108
Tabel 6.18 Penilaian fitur penawaran	109
Tabel 6.19 Penilaian fitur order masuk.....	109
Tabel 6.20 Penilaian fitur pengiriman	109
Tabel 6.21 Penilaian fitur pembelian	110
Tabel 6.22 Penilaian fitur pelanggan	110
Tabel 6.23 Penilaian fitur <i>supplier</i>	110
Tabel 6.24 Penilaian fitur <i>login</i>	111

DAFTAR GAMBAR

Gambar 2.1 <i>Waterfall Model</i>	8
Gambar 2.2 Arsitektur <i>Web Service</i>	10
Gambar 2.3 Interaksi <i>REST Web Service</i>	11
Gambar 2.4 Elemen <i>Start, intermadate</i> dan <i>event</i>	12
Gambar 2.5 Elemen-elemen <i>Acitvity</i>	13
Gambar 2.6 Elemen <i>sequence flow, message flow, dan Association</i>	13
Gambar 2.7 <i>Pool</i> dan <i>Lane</i>	13
Gambar 2.8 Elemen data <i>object, group, dan annotation</i>	14
Gambar 2.9 Proses bisnis CV. Mitra Teknik	15
Gambar 2.10 Proses bisnis setelah dibangun sistem informasi manajemen penjualan.....	17
Gambar 2.11 Notasi aliran graph (Pressman, 2010).....	24
Gambar 4.1 <i>Use case diagram</i> sistem informasi penjualan.....	37
Gambar 5.1 Arsitektur Sistem Informasi Penjualan.....	53
Gambar 5.2 <i>Sequence diagram Login</i>	54
Gambar 5.3 <i>Sequence diagram</i> Menambahkan <i>Order Masuk</i>	55
Gambar 5.4 <i>Sequence diagram</i> Mengubah <i>Order Masuk</i>	56
Gambar 5.5 <i>Sequence diagram</i> Menghapus <i>Order Masuk</i>	56
Gambar 5.6 <i>Sequence diagram</i> Menambahakan Pembelian Barang	57
Gambar 5.7 Perancangan <i>class diagram</i> sistem informasi penjualan.....	59
Gambar 5.8 <i>Entity Relationship Diagram (ERD)</i>	60
Gambar 5.9 Antarmuka <i>login</i>	65
Gambar 5.10 Antarmuka menambahkan <i>user</i>	66
Gambar 5.11 Antarmuka menambahkan penawaran	67
Gambar 5.12 Antarmuka list penawaran.....	67
Gambar 5.13 Antarmuka menambahkan <i>order masuk</i>	68
Gambar 5.14 Antarmuka list <i>order masuk</i>	69
Gambar 5.15 Antarmuka menambahkan pembelian barang	69
Gambar 5.16 Antarmuka list stok barang	71
Gambar 5.17 Antarmuka menambahkan pengiriman	71

Gambar 5.18 Antarmuka List Pengiriman	72
Gambar 5.19 Antarmuka menambahkan <i>supplier</i>	72
Gambar 5.20 Antarmuka list <i>supplier</i>	73
Gambar 5.21 Antarmuka menambahkan pelanggan.....	74
Gambar 5.22 Antarmuka list pelanggan	75
Gambar 5.23 Antarmuka menampilkan laporan	75
Gambar 5.24 Implementasi Arsitektur	78
Gambar 5.25 Implementasi Basis Data.....	78
Gambar 5.26 Antarmuka <i>login</i>	84
Gambar 5.27 Antarmuka menambahkan <i>user</i>	84
Gambar 5.28 Antarmuka menambahkan penawaran	85
Gambar 5.29 Antarmuka list penawaran.....	85
Gambar 5.30 Antarmuka list <i>order</i> masuk.....	86
Gambar 5.31 Antarmuka menambahkan <i>order</i> masuk	86
Gambar 5.32 Antarmuka menambahkan pembelian barang	87
Gambar 5.33 Antarmuka list stok barang	87
Gambar 5.34 Antarmuka menambahkan pengiriman	88
Gambar 5.35 Antarmuka list pengiriman.....	88
Gambar 5.36 Antarmuka menambahkan <i>supplier</i>	89
Gambar 5.37 Antarmuka list <i>supplier</i>	89
Gambar 5.38 Antarmuka menambahkan pelanggan.....	90
Gambar 5.39 Antarmuka list pelanggan	90
Gambar 5.40 Antarmuka menampilkan laporan	91
Gambar 6.1 <i>Flowgraph</i> tambah_ <i>order</i> _masuk.....	93
Gambar 6.2 <i>Flowgraph</i> edit_pembelian	94
Gambar 6.3 <i>Flowgraph</i> hapus_ <i>order</i> _masuk.....	96
Gambar 6.4 Hasil Pengujian <i>Security</i>	104
Gambar 6.5 Hasil Pengujian <i>Performance</i>	104

DAFTAR LAMPIRAN

LAMPIRAN KUISIONER <i>USER ACCEPTANCE TESTING (UAT)</i>	115
LAMPIRAN	116
LAMPIRAN	117
LAMPIRAN	118

BAB 1 PENDAHULUAN

1.1 Latar belakang

Pada era global seperti sekarang ini lingkungan industri berkembang secara dinamis hal ini menjadi pengaruh banyak perusahaan untuk mengembangkan potensi yang dimiliki. Dalam persaingan semakin kompetitif dalam lingkungan industri, perusahaan harus mampu mencari dan mengidentifikasi faktor-faktor yang menjadi pengaruh kunci untuk meraih kesuksesan. Perkembangan teknologi sangat berpengaruh pada kekuatan sebuah perusahaan untuk berkompetisi dalam iklim persaingan antar kompetitor. Usaha-usaha yang dilakukan untuk bertahan dalam persaingan dengan cara memberikan pelayanan terbaik kepada *client* untuk mendapatkan kepercayaan.

Untuk mencapai tujuan tersebut, sebuah perusahaan membutuhkan sesuatu yang ekonomis dan efektif guna untuk memperoleh keuntungan dalam menjalankan roda industri. Kelancaran proses transaksi antara *supplier* maupun dengan *client* juga sangat diperlukan, agar terjaga kerjasama antara dua belah pihak dan membangun relasi yang saling menguntungkan. Tanpa *supplier* atau *client* yang memadai, roda bisnis dalam industri tidak akan berjalan baik seiring berkembangnya pasar seperti saat ini. Pembangunan sistem informasi diperlukan oleh sebuah perusahaan untuk pengawasan dan perencanaan yang bertujuan untuk meminimalkan investasi yang dikeluarkan untuk keperluan seperti persediaan bahan mentah, biaya yang dikeluarkan saat produksi, dan dapat meningkatkan efisiensi yang berkaitan penggunaan biaya dan sumber daya yang harus dikeluarkan oleh perusahaan. Termasuk untuk manajerial barang dalam persediaan barang, penetapan terhadap besarnya persediaan barang harus dijalankan untuk menjamin kegiatan produksi maupun distribusi (Ristono, 2009).

CV. Mitra Teknik merupakan perusahaan yang menyediakan layanan jasa pembuatan, perbaikan material-material mesin industri. Dalam roda bisnisnya CV. Mitra Teknik memerlukan sistem informasi yang mampu untuk mengolah data dan informasi dalam perusahaan. Tanpa pengolahan data dan informasi akurat, perusahaan tidak dapat menetukan keputusan, kebijakan, atau peraturan yang menunjang kinerja dalam perusahaan. Untuk saat ini proses bisnis dalam perusahaan masih bersifat konvensional atau masih manual seperti pencatatan transaksi pembelian, pengiriman dan lain-lain. Kondisi ini membuat operasional dalam perusahaan menjadi terganggu dan berdampak pada penyesuaian data antara unit satu dengan unit yang lainnya. Permasalahan lain yang terjadi dalam perusahaan CV. Mitra Teknik saat ini yaitu belum adanya sistem yang terkomputerisasi yang mampu untuk mengolah data yang mengakibatkan adanya data-data hilang yang dapat mengganggu proses operasional dalam perusahaan. Sistem informasi yang dibangun ini nantinya memberikan kemudahan pengguna untuk mengatur proses transaksi, rekap data, pengolahan data dan lain-lain sehingga dapat menghemat waktu dalam menjalankan bisnis proses yang berlangsung. Sistem penjualan yang nantinya dibuat, selain mampu membantu proses transaksi, pengolahan data, sistem penjualan ini nantinya juga dapat

mengurangi biaya pengeluaran yang dikeluarkan oleh perusahaan. Sistem dapat membantu pembukuan atau pengolahan data dalam perusahaan untuk admin menjadi lebih terbantu karena adanya sistem penjualan.

Aplikasi berbasis *web* merupakan salah satu teknologi yang banyak digunakan. Dalam menjalankan aplikasi berbasis *web* dibutuhkan *web server* dan *browser*. Untuk *database* ataupun *file* lainnya dapat diinstal dalam sebuah *server*, sehingga sistem dapat diakses meskipun dari jarak yang jauh. Saat ini telah tersedia metode untuk mengembangkan aplikasi *web* antara lain *framework* dengan dukungan berbagai bahasa pemrograman seperti ASP, PHP, Java serta bahasa pemrograman lainnya. Penelitian ini menyajikan salah satu alternatif dalam membangun sistem informasi penjualan berbasis web menggunakan MySQL untuk menyimpan basis data, *framework* digunakan untuk tampilan menggunakan *bootstrap* berbasis HTML dan CSS, sedangkan untuk *backend* menggunakan bahasa pemrograman PHP dan *framework* *Codeigniter* untuk membangun arsitektur sistem. Untuk proses pembangunan sistem informasi penjualan ini menggunakan pendekatan metode *Software Development Life Cycle* (SDLC). Metode *waterfall* ialah metode yang mempunyai alur yang harus dilakukan secara berurutan pada setiap fasenya. Berikut fase tahapan dalam *waterfall* yakni analisis kebutuhan, perancangan, implementasi, pengujian, dan perbaikan. Metode *waterfall* digunakan karena semua kebutuhan pembangunan diperoleh pada fase awal, dan teknologi *web service* digunakan untuk menunjang proses pembangunan sistem informasi penjualan CV. Mitra Teknik.

Dari pemaparan latar belakang diatas, dibuatlah sistem informasi penjualan berbasis web. Dengan dibuatnya sistem diharapkan dapat membantu proses transaksi, distribusi, dan pencatatan barang sehingga dapat meminimalisir kesalahan *input* data barang, data transaksi dan pencarian data sehingga kinerja perusahaan menjadi lebih cepat dan efisien.

1.2 Rumusan masalah

1. Kebutuhan apa saja yang dibutuhkan untuk membangun Sistem Informasi Manajemen Informasi Penjualan Berbasis Web pada CV. Mitra Teknik?
2. Bagaimana membangun Sistem Informasi Manajemen Penjualan Berbasis Web pada CV. Mitra Teknik?
3. Bagaimana hasil pengujian pada sistem informasi penjualan CV. Mitra Teknik?

1.3 Tujuan penelitian

1. Mendapatkan analisis dari kebutuhan yang diperlukan untuk membangun Sistem Informasi Manajemen Penjualan Berbasis Web pada CV. Mitra Teknik.
2. Dari proses pembangunan sistem didapatkan hasil yaitu analisis sistem, perancangan sistem, dan implementasi dari sistem informasi penjualan CV. Mitra Teknik.
3. Mengetahui hasil dari pengujian pada sistem penjualan CV. Mitra Teknik dengan menggunakan pengujian *White-box*, *Black-box*, dan *User Acceptance Test* (UAT).

1.4 Manfaat

1. Untuk perusahaan

Manfaat yang nantinya didapat dari pembangunan sistem ini adalah sistem dapat memberi kemudahan untuk mengatur dan memanajemen data barang, data transaksi dan lain-lain yang nantinya sistem dikelola oleh admin. Sistem yang dibuat bertujuan untuk mengurangi kesalahan *input* data dan dapat meminimalisir kesalahan admin yang mengakibatkan pengeluaran dalam perusahaan yang berlebihan, dan dapat mengganggu proses bisnis antara *supplier* maupun perusahaan mitra kerja, dengan dibuatnya sistem informasi penjualan ini diharapkan operasional perusahaan dapat berjalan dengan lancar dan efisien.

2. Untuk Mahasiswa

Manfaat yang lainnya yaitu untuk mahasiswa, mahasiswa yang bersangkutan dapat menerapkan ilmu yang didapat selama masa perkuliahan dan mampu mengimplementasikan dengan baik dalam proses penggerjaan sistem informasi penjualan.

1.5 Batasan masalah

1. Pengembangan sistem menggunakan model *Software Development Lifecycle (SDLC) waterfall* tidak sampai tahap terakhir yaitu *Operation and Maintenance*.
2. Teknologi berbasis *web* digunakan dalam pembangunan sistem.
3. Implementasi sistem hanya sampai menghasilkan *prototype*.
4. Data yang diolah berasal dari aktivitas penjualan perusahaan CV. Mitra Teknik.

1.6 Sistematika pembahasan

BAB I PENDAHULUAN

Dalam bab pendahuluan menjelaskan mengenai latar belakang dibuatnya sistem informasi manajemen penjualan, merumuskan masalah yang ada, tujuan yang didapatkan setelah pembangunan sistem, batasan yang harus diperhatikan dan rangkaian penelitian yang dibahas secara sistematika.

BAB II LANDASAN KEPUSTAKAAN

Pada bab landasan kepustakaan berisi tentang pembahasan mengenai tentang penelitian sebelumnya pada topik sistem informasi penjualan dan kajian pustaka yang dijadikan sebagai teori pendukung penelitian.

BAB III METODOLOGI

Pada bab ini berisi tentang metode atau tahap-tahap yang dilakukan dalam penelitian. Berikut tahap-tahap yang harus dilakukan yaitu studi literatur, menganalisis kebutuhan sistem, merancang kebutuhan sistem, implementasi sistem, tahap pengujian sistem, pengambilan saran dan kesimpulan.

BAB IV ANALISIS KEBUTUHAN

Pada bab analisis kebutuhan berisi tentang penggalian kebutuhan sistem yang terdiri dari elisitasi kebutuhan dan identifikasi pemangku kepentingan atau *stakeholder*, untuk spesifikasi kebutuhan sistem kebutuhan sesuai dengan solusi permasalahan yang telah diidentifikasi, dan dirancang dalam *use case diagram* dan *use case scenario*.

BAB V PERANCANGAN DAN IMPLEMENTASI

Pada bab ini menjelaskan perancangan apa saja yang dibutuhkan dalam membangun sistem informasi manajemen penjualan CV. Mitra Teknik berbasis *web* dan implementasi sistem sesuai dengan rancangan pada sistem yang telah dibuat.

BAB VI PENGUJIAN DAN ANALISIS

Pada bab ini menjelaskan tentang tahap pengujian yang dilakukan pada sistem. Dari hasil pengujian sistem akan ditarik kesimpulan apakah sistem telah sesuai berjalan sesuai dengan rancangan dan telah memenuhi kebutuhan.

BAB VII PENUTUP

Pada bab ini bersisi kesimpulan dan saran tentang pembangunan sistem informasi penjualan CV. Mitra Teknik berbasis *web* yang telah dilakukan. Dan berisi saran untuk penelitian selanjutnya.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Tinjauan Pustaka

Pada landasan pustaka menjelaskan bagaimana peneliti memilih judul yang digunakan sebagai penelitian. Landasan pustaka menjelaskan tentang topik penelitian yang sebelumnya dilakukan dan dijadikan sebagai perbandingan dengan topik yang saat ini peneliti lakukan. Penelitian sejenis sebelumnya juga pernah dilakukan oleh Risnandar (2010) yang berjudul “Integrasi Teknologi Informasi dan *Supply Chain Management* (Studi kasus: PT. X, West java)” yakni membuat sistem yang akan digunakan untuk mengolah pendistribusian produk-produk kepada konsumen dikarenakan distribusi produk kepada konsumen belum optimal karena disebabkan oleh ketidak seimbangan informasi antar tiap aktivitas proses, mulai dari manufaktur, *supplier*, pergudangan dan lain-lain.

Penelitian yang selanjutnya dilakukan oleh Sri Harjunawati (2016) berjudul “Sistem Informasi Akuntansi Penjualan Berbasis *waterfall model* Untuk Perusahaan Dagang”, dari prasurvei yang sudah dilakukan di beberapa perusahaan dapat ditarik sebuah kesimpulan dimana perusahaan dagang yang masih menggunakan sistem manual menghadapi beberapa masalah terkait dengan data atau informasi yang dihasilkan, sehingga secara tidak langsung akan berpengaruh pada pemasukan atau pengeluaran pada suatu operasional perusahaan, dengan itu penulis membuat sistem yang dapat membantu proses pengendalian intern perusahaan dengan menerapkan *waterfall model* untuk pengembangan sistem.

Dari ketiga penelitian di atas, penulis coba menggabungkan masalah yang diangkat pada penelitian tersebut. Untuk pengembangan sistem mengikuti model SDLC *waterfall*. Sistem informasi penjualan dibangun berbasis web dengan menggunakan bahasa pemrograman PHP, HTML, CSS, dan MySql sebagai basis data.

2.2 Landasan Teori

2.2.1 Sistem Informasi

Sistem Informasi ialah sistem yang digunakan pada suatu kelompok organisasi untuk dapat membantu memanajemen kebutuhan transaksi harian, bersifat manajerial membantu organisasi strategis yang terbantu dan menyediakan laporan-laporan bagi orang luar (Jogiyanto, 2005). Dari pernyataan tersebut dapat disimpulkan bahwa sisstem informasi ialah berisi alat yang didalamnya digunakan kerangka kerja untuk mengolah data yang keluarannya sebuah informasi bertujuan untuk membantu sebuah organisasi untuk mencapai sebuah tujuan dengan memerlukan bantuan berupa, komputer, manusia, dan prosedur kerja.

2.2.2 Komponen Sistem Informasi

Komponen dari sistem informasi terdiri dari:

1. Komponen *input*

Komponen *input* ialah informasi yang masuk kedalam sistem. Komponen *Input* termasuk media dan cara memperoleh suatu data yang akan diolah menjadi dokumen pokok.

2. Komponen model

Komponen model ialah kombinasi logika, model, prosedur, dan matematik berfungsi sebagai data *input* dimanipulasi dan data yang tersimpan kemudian data tersimpan dalam basis data untuk menghasilkan keluaran diinginkan dengan cara tertentu.

3. Komponen *output*

Komponen *output* adalah hasil dari oleh data yang merupakan dokumentasi yang berguna untuk pengguna sistem.

4. Komponen teknologi

Teknologi merupakan alat yang digunakan dalam sistem informasi, teknologi dimanfaatkan untuk proses *input* data, menjalankan model, mengakses dan menyimpan data kemudian mengirimkan *output* dan membantu penggunaan sistem secara keseluruhan.

5. Komponen *hardware*

Komponen ini berperan sebagai media untuk penyimpanan yang sangat penting untuk sistem informasi yaitu sebagai tempat menampung *database* sehingga lebih memudahkan dan memperlancar kerja sistem informasi

6. Komponen *software*

Komponen *Software* bekerja sebagai tempat menghitung, mengolah, dan memanipulasi data yang menghasilkan informasi dan data diambil dari *hardware*.

7. Komponen basis data

Komponen ini berisi kumpulan data yang tersimpan dalam sistem yang saling berkaitan dengan yang lain. Data pada sistem informasi perlu disimpan dalam *database* agar pengguna sistem informasi lebih mudah untuk mengolah data yang tersimpan didalam *database*. Komponen ini sangat bermanfaat untuk efisiensi pengelolaan data.

8. Komponen kontrol

Komponen ini mencegah gangguan sistem informasi seperti air, temperatur, debu, bencana alam, kegagalan, kekurangan, ataupun yang lainnya. Berberapa alat dibuat dan dirancang untuk mengendalikan agar tidak merusak sistem dan dapat mencegah kesalahan yang nantinya bisa diatasi.

2.3 Rekayasa Perangkat Lunak

Penggunaan konsep rekayasa lunak yang bertujuan untuk membuat membuat sebuah aplikasi yang mempunyai *value* pada sebuah perangkat lunak dan mampu bekerja secara efisien merupakan konsep dari rekayasa perangkat lunak. *Software* banyak dibutuhkan hingga akhirnya tidak terpakai karena faktor kebutuhan *client* atau non-teknis menjadi masalah. Oleh karena itu, perangkat lunak dibuat agar terpakai dan lebih bermanfaat (Salahuddin, 2011). Rekayasa perangkat lunak lebih fokus pada pembangunan *software* yang bermanfaat kepada pelanggan (*costumer*). Pada teori ilmu komputer lebih berfokus pada teori dasar perangkat komputer. Rekayasa perangkat lunak berfokus bagaimana cara membangun sebuah *software* yang mampu memenuhi permintaan konsumen:

1. Mampu dikembangkan setelah *software* selesai dibuat mengikuti berkembangnya lingkup teknologi (*Maintainability*).
2. Dapat dihadalkan dengan dijalankannya proses bisnis dan pergantian yang akan terjadi (*Dependability* dan *robust*).
3. Penggunaan *resource* yang lebih efisien.
4. Kemampuan *software* dalam memenuhi kebutuhan (*Usability*) menjadi perangkat lunak yang berguna bagi pelanggan.

Pengembangan perangkat lunak yang memenuhi kriteria diinginkan pelanggan merupakan pengembangan perangkat lunak yang benar, pengembangan perangkat lunak bukan berkonsentrasi pada pengembang atau pembuat perangkat lunak.

Rekayasa perangkat lunak mempunyai beberapa tahapan yang harus dilalui. Jika tahapan-tahapan tersebut tidak dilalui, untuk mendapatkan perangkat lunak yang berkualitas baik. Berikut tahap-tahap pembangunan perangkat lunak.

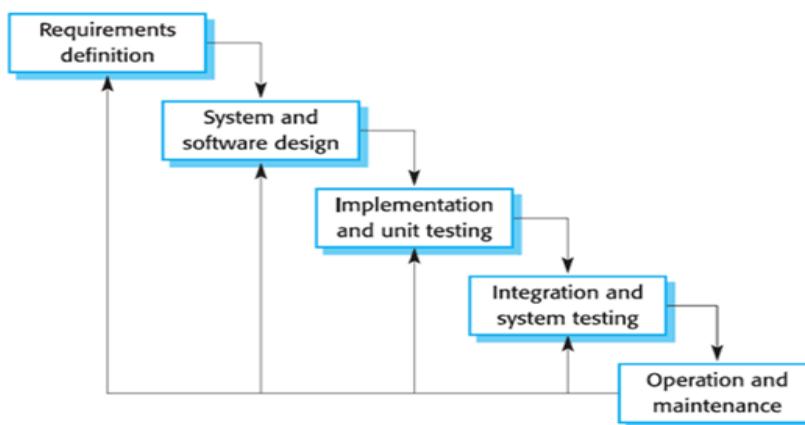
1. Pengumpulan Spesifikasi (*Specification*)
Mengetahui batasan dalam pengembangan perangkat lunak dan dapat mengetahui apa saja yang harus dikerjakan oleh perangkat lunak.
2. Pengembangan (*Development*)
Proses ini dilakukan apabila perangkat lunak perlu dikembangkan.
3. Validasi (*Validation*)
Menguji apakah perangkat lunak yang dibangun telah memenuhi kebutuhan pelanggan (*customer*).
4. Evolusi (*Evolution*)
Mengembangkan atau merubah perangkat lunak yang diakibatkan oleh lingkungan sekitar yang menharuskan perangkat lunak harus dikembangkan.

Dalam proses pengembangan perangkat lunak terdapat tahapan-tahapan yang harus diikuti. Yang bertujuan untuk memberikan arahan dan panduan dalam melakukan pengembangan sebuah perangkat lunak. Terdapat beberapa model

atau pendekatan yang dilakukan pada pengembangan perangkat lunak. Penggunaan metode *waterfall* kali ini diterapkan dalam pembangunan sistem informasi manajemen penjualan berbasis *web*.

2.3.1 Waterfall Model

Prosedur pengembangan yang diterapkan pada sistem informasi manajemen penjualan ini menggunakan metode pengembangan *waterfall model*. Metode ini sistematis untuk pendekatannya mempunyai tahap-tahap berikut tahapan yang dilakukan tahap analisis, desain, *coding*, *testing* atau *verification*, dan *maintenance*. Tahap demi tahap dilakukan karena harus menunggu tahap yang selanjutnya maka tahap-demi tahap dilalui secara berurutan maka dari itu disebut dengan *waterfall*. Tahapan-tahapan yang dilalui dalam model *waterfall* dapat dilihat pada gambar berikut:



Gambar 2.1 Waterfall Model

Pada gambar 2.1 (Sommerville, 2011) di atas adalah tahapan umum dari model proses ini. Akan tetapi Pressman memecah model ini menjadi enam tahapan meskipun secara garis besar sama dengan tahapan-tahapan model *waterfall* pada umumnya. Berikut adalah penjelasan dari tahap-tahap yang dilakukan:

1. *Requirements definition*. Proses pencarian kebutuhan diintensifkan dan difokuskan pada *software*. Untuk mengetahui sifat dari program yang akan dibuat. Dari dua aktivitas tersebut (pencarian kebutuhan sistem dan *software*) harus didokumentasikan dan ditunjukkan kepada pelanggan, berikut proses dalam analisis kebutuhan:
 - a. Elisitasi kebutuhan. Proses ini dilakukan untuk mengamati kebutuhan yang ada pada sistem yang akan dibuat.
 - b. Spesifikasi kebutuhan. Proses ini menjelaskan secara detail tentang spesifikasi kebutuhan yang dijelaskan oleh pengguna.
 - c. Validasi kebutuhan. Proses ini dilakukan untuk memeriksa apakah pencarian kebutuhan apakah sudah benar dan sesuai yang diharapkan.

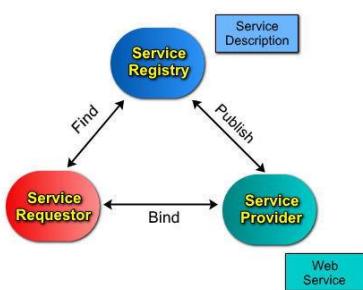
2. *System and Software Design.* Proses ini digunakan untuk mengubah kebutuhan-kebutuhan di atas menjadi representasi ke dalam bentuk “*blueprint*” *software* sebelum *coding* dimulai. Perancangan harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Maka proses ini juga harus didokumentasikan sebagai konfigurasi pada *software*. Berikut proses bagian-bagian perancangan sistem:
- a. Perancangan arsitektur. Proses ini mengidentifikasi komponen utama dalam sistem, hubungan antar sistem, dan bagaimana sistem didistribusikan.
 - b. Perancangan antarmuka. Proses ini yaitu mendesain *interface* sistem sesuai dengan kebutuhan oleh pengguna yang akan diimplementasikan dalam sistem.
 - c. Perancangan basis data. Proses ini merancang struktur data dalam *database* yang baru untuk proses penyimpanan data dalam sebuah sistem.
3. *Implementation and Unit Testing.* Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses *coding*. Tahap ini merupakan implementasi dari tahap desain yang secara teknis nantinya dikerjakan oleh *programmer*.
4. *Integration and System Testing.* Sesuatu yang dibuat haruslah diujicobakan. Demikian juga dengan *software*. Semua fungsi-fungsi *software* harus diujicobakan, agar *software* bebas dari *error*, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya. Dalam software testing ini terdapat dua teknik pengujian yaitu *white-box testing* dan *black-box testing*, berikut penjelasannya:
- a. *White-box testing* yaitu pengujian yang dilakukan terhadap setiap kode program atau logika pada setiap fungsi pada implementasi program, *Basis path testing* merupakan salah satu jenis pengujian dalam *white-box testing* yang digunakan (Pressman & Maxim, 2015).
 - b. *Black-box testing* yaitu pengujian yang dilakukan dengan cara memeriksa hasil keluaran atau masukan sistem sudah sesuai dengan analisis kebutuhan yang telah ditentukan.
 - e. *Operation and Maintenance.* Pemeliharaan suatu *software* diperlukan, termasuk di dalamnya adalah pengembangan, karena *software* yang dibuat tidak selamanya hanya seperti itu. Ketika dijalankan mungkin saja masih ada *error* kecil yang tidak ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada *software* tersebut. Pengembangan diperlukan ketika adanya perubahan dari eksternal perusahaan seperti ketika ada pergantian sistem operasi, atau perangkat lainnya.

2.4 Web Service

2.4.1 Definisi Web Service

Web service dapat dijalankan diaplikasi yang berbeda dan bisa dijalankan diberbagai platform maupun *framework*. *Web service* menyediakan standar komunikasi dengan berbagai aplikasi yang bermacam-macam. *Web service* dapat diartikan sebuah antarmuka yang memaparkan serangkaian operasi yang dapat diakses melalui jaringan, seperti internet dalam bentuk pesan XML (Kreger, 2001). *Web service* disebut juga bagian-bagian dari informasi proses yang setiap orang dapat mengakses, kapan saja dengan menggunakan perangkat apa saja. Yang tidak terikat sistem operasi atau bahasa pemrograman yang digunakan (Manes, 2001). *Web service* menggunakan *platform* dasar XML ditambah *HTTP*. Protokol yang dapat diimplementasikan di atas protokol yang lain atau di atas jaringan lain yang disebut dengan *HTTP*. Sedangkan XML dapat menjadi aplikasi dengan bahasa apapun untuk berinteraksi dengan platform apapun dan aplikasi tersebut menerima respons XML dari aplikasi tersebut. (Siswoutomo, 2004).

2.4.2 Arsitektur Web Service



Gambar 2.2 Arsitektur Web Service

Pada *web service* terdapat tiga komponen berbeda dan berperan yang saling berinteraksi (Kreger, 2001). Interaksi meliputi, menemukan, mempublikasikan dan menautkan operasi. Pada saat yang sama dan operasi *web service* ini mempengaruhi artefak dari perangkat lunak *web service* dan penjelasannya.

Pada gambar 2.2 (Brittenham, 2002) terdapat tiga macam komponen dalam arsitektur *web service* tiga diantaranya adalah:

- *Service provider*, Dari sudut pandang arsitektur, *service* ialah *platform* yang menjadi tempat untuk menyediakan layanan untuk kumpulan fungsi dalam *web service*.
- *Service requestor (client)*, ialah bagian *client* dari *web service* yakni bagian yang membutuhkan dan memanfaatkan layanan pada sudut pandang bisnis. Sementara dari perspektif arsitektur, *service requestor* bertugas sebagai pencarian dan inisiasi pada layanan. Peran *service* dapat dijalankan dalam *browser* oleh seseorang dengan program tanpa *interface*.

- *Service registry*, Pada arsitektur *web service* merupakan tempat *service provider* menginformasikan layanannya. Fungsi dari *Service Registry* bersifat opsional.

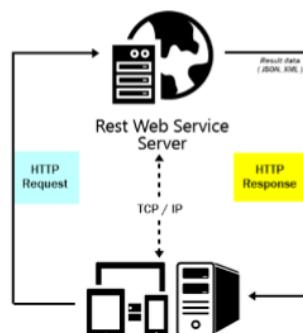
Dari implementasi teknologi *web service* yang digunakan, nantinya sistem yang dibangun dapat dikembangkan dan dihubungkan dalam bermacam-macam aplikasi dengan sistem operasi yang berbeda-beda.

2.4.3 REST (*Representational State Transfer*)

Representational State Transfer (REST) adalah gaya arsitektur perangkat lunak untuk sistem hypermedia terdistribusi seperti *world wide web*. *REST* sering digunakan dalam pengertian lepas untuk menguraikan penghubung sederhana yang mengirimkan data spesifik domain di atas *HTTP* lapisan *messaging* tambahan seperti *SOAP* atau *session tracking*. *SOAP* dan *REST* menyediakan fungsionalitas yang sama tetapi *request* dan *response* dari *web service* berbasis *SOAP* harus ditulis dalam format *SOAP* kemudian dikemas dalam pesan *HTTP*, sedangkan *REST web service* tidak menggunakan format *SOAP* dan hanya menggunakan *HTTP* sebagai *protokol layer* (Susanto, 2017). Metode *REST* dilandasi oleh empat prinsip dasar utama, yaitu:

1. *URI (Resource Identifier through Uniform Resource Identifier)*,
2. *Uniform Interface*, sumber daya yang dimanipulasi adalah *CRUD (Create, Read, Update, Delete)* dengan menggunakan operasi standar *HTTP* yaitu *PUT, GET, POST, DELETE*.
3. *Self-descriptive message*, sumber daya informasi yang tidak terkait, sehingga dapat mengakses berbagai format konten (HTML, PDF, JPEG, XML, Plain Text dan lainnya).
4. *Stateful interaction though hyperlink*, setiap interaksi dengan suatu sumber daya bersifat *stateless*, yaitu *request message* tergantung jenis kontennya.

Interaksi *REST web service* dapat dilihat pada gambar berikut (Susanto, 2017).



Gambar 2.3 Interaksi *REST Web Service*

Gambar 2.3 di atas menjelaskan bagaimana *REST web service* berkomunikasi dengan *client* mengirim sebuah *request* kemudian *request* tersebut dikirim melalui *HTTP request* dan sampai pada *REST server*. Kemudian *server*

membalasnya dengan mengirimkan hasil dari *request client* melalui *HTTP response* berupa *JSON*.

2.5 Bussines Proces Modelling Notation (BPMN)

Bussines Proces Modelling Notation (BPMN) merupakan gambaran diagram suatu proses bisnis yang didasarkan pada teknik diagram alur, dirancang untuk membuat model-model grafis dari operasi bisnis dimana terdapat aktivitas atau kontrol yang mendefinisikan urutan suatu pekerjaan.

BPMN dikembangkan oleh konsorium industri (BPMN.org) yaitu konstituen yang mewakili berbagai vendor alat BPM tetapi bukan sebagai pembuka akhir, mengemukakan bahwa “ *The Business Process Modeling Notation is Emerging as a standard language for capturing business processes, especially at the level of domain analysis and high level systems design* ” (BPMN.org, 2008).

Terdapat beberapa elemen dalam BPMN yaitu:

2.5.1 Flow Object

a. Event

Event direpresentasikan dalam bentuk lingkaran dan menjelaskan apa yang terjadi saat itu. Ada dua jenis *event*, yaitu *start*, *intermediate*, dan *end*. Event-event ini mempengaruhi alur proses dan biasanya menyebabkan terjadinya kejadian (*trigger*) atau sebuah dampak (*result*).

Masing-masing mewakili kejadian dimulainya proses bisnis, interupsi proses bisnis, dan akhir dari proses bisnis. Untuk setiap jenis *event* tersebut sendiri terbagi atas beberapa jenis, misalnya *message start*, yang dilambangkan seperti *start event* namun mendapatkan tambahan lambang amplop di dalamnya, yang berarti ada pesan *event* tersebut dimulai dengan masuknya pesan.

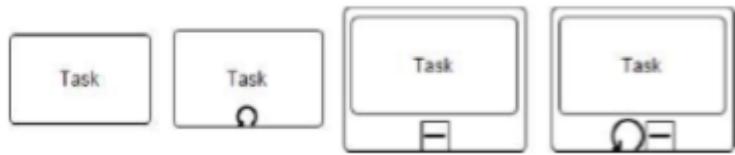


Gambar 2.4 Elemen *Start*, *intermadate* dan *event*

Sumber : BPMN (2008)

b. Activity

Activity merepresentasikan pekerjaan (*task*) yang harus diselesaikan. Ada empat macam *activity* yaitu *task*, *looping task*, *sub process*, dan *looping subprocess*.



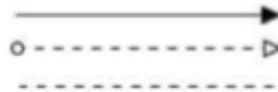
Gambar 2.5 Elemen-elemen Activity

Sumber : BPMN (2008)

2.5.2 Connecting Object

Connecting object merupakan aliran pesan antar proses dimana satu kejadian dengan kejadian yang lain saling berhubungan dan merepresentasikan dari hubungan tersebut. Adapun simbol-simbol atau gambar dalam penulisan *connecting object* ada 3 jenis yaitu:

- a. *Sequence flow*, merepresentasikan pilihan *default* untuk menjalankan proses
- b. *Message flow*, merepresentasikan aliran pesan antar proses
- c. *Association*, digunakan untuk menghubungkan elemen dengan *artifact*

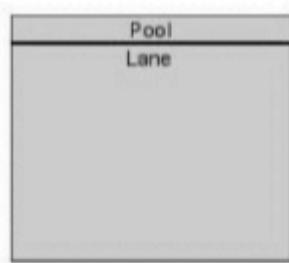


Gambar 2.6 Elemen sequence flow, message flow, dan Association

Sumber : BPMN (2008)

2.5.3 Swimlanes

Elemen ini digunakan untuk mengkategorikan secara visual seluruh elemen dalam diagram. Ada dua jenis *swimlanes*, yaitu *pool* dan *lane*. Perbedaannya adalah *lane* terletak di bagian dalam *pool* untuk mengkategorikan elemen-elemen di dalam *pool* menjadi lebih spesifik.



Gambar 2.7 Pool dan Lane

Sumber : BPMN (2008)

2.5.4 Artifacts

Elemen ini digunakan untuk memberi penjelasan di diagram. Elemen ini terdiri atas tiga jenis, yaitu:

1. *Data object*, digunakan untuk menjelaskan data apa yang dibutuhkan dalam proses.
2. *Group*, untuk mengelompokkan sejumlah aktivitas di dalam proses tanpa mempengaruhi proses yang sedang berjalan.
3. *Annotation*, digunakan untuk memberi catatan agar diagram menjadi lebih mudah dimengerti.



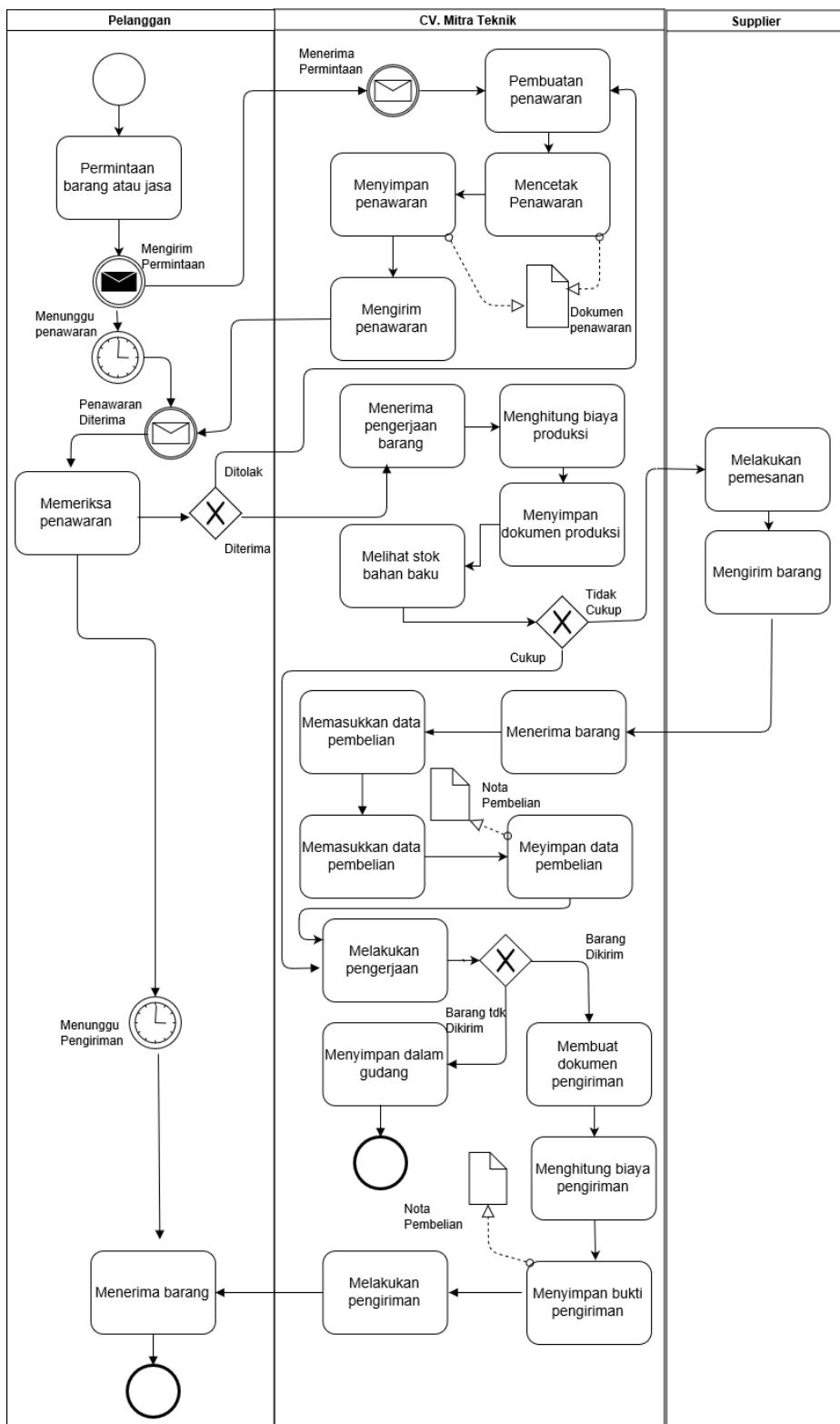
Gambar 2.8 Elemen data *object*, *group*, dan *annotation*

Sumber : (BPMN, 2008)

Notasi dalam BPMN menggambarkan pesan yang disampaikan antara pelaku bisnis dan bagian dari pelaku bisnis. Kelebihan dalam penggunaan BPN yakni mampu memodelkan aliran pesan karena dapat menggambarkan proses aliran bisnis dalam suatu organisasi maupun departemen.

2.6 Proses bisnis CV. Mitra Teknik

2.6.1 Proses bisnis sebelum dibuatnya sistem informasi penjualan



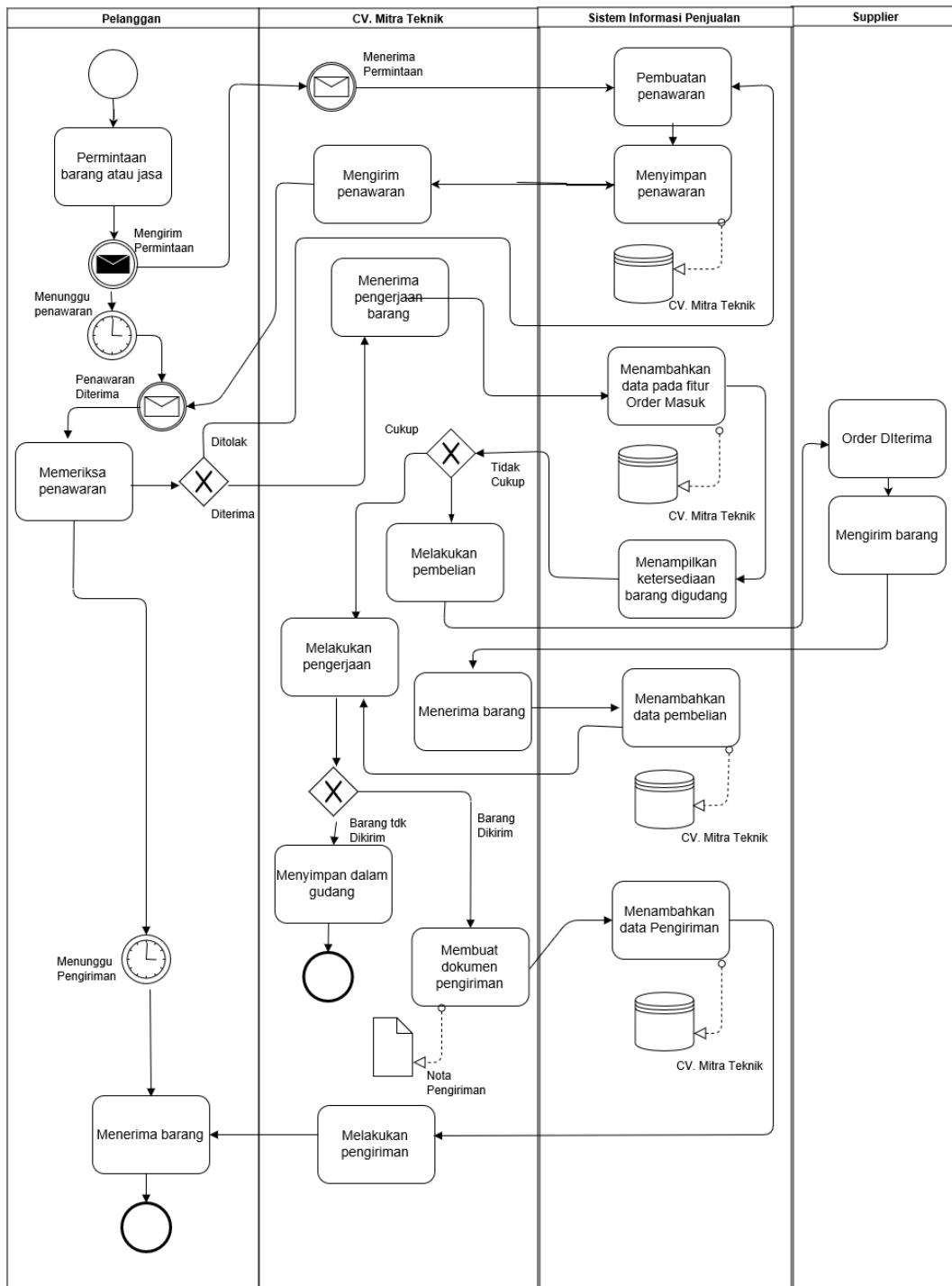
Gambar 2.9 Proses bisnis CV. Mitra Teknik
Sumber: Sunarto (2018)

Terdapat 3 aktifitas tanpa menggunakan sistem pada proses bisnis CV. Mitra Teknik yaitu, proses penjualan dan pengiriman ke pelanggan, proses pembuatan penawaran, proses pembelian ke *supplier*. Berikut penjelasan dari gambar 2.10 sebagai berikut:

1. Dimulai dari pelanggan yang membutuhkan permintaan barang atau jasa.
2. Setelah itu pihak pelanggan membuat permintaan barang atau jasa dan dikirim melalui email atau telepon.
3. Setelah permintaan diterima, lalu CV. Mitra Teknik membuat penawaran.
4. Dokumen penawaran dibuat, disimpan, dan dicetak lalu dikirim ke pelanggan.
5. Apakah penawaran yang telah dikirim “ditolak”, maka mengulang ke proses membuat penawaran.
6. Setelah penawaran diterima, lalu *order* diterima oleh CV. Mitra Teknik dan lanjut ke tahap perhitungan biaya produksi, kemudian menyimpan dokumen perhitungan produksi, setelah itu melihat ketersediaan bahan baku atau barang untuk kegiatan penggerjaan barang.
7. Sebelum ke tahap penggerjaan, dicek apakah bahan baku atau alat yang digunakan untuk proses penggerjaan sudah “cukup”, maka mengecek dalam gudang apakah bahan tersedia?
8. Sebelum ke tahap penggerjaan, dicek apakah bahan baku atau alat yang digunakan untuk proses penggerjaan sudah “tidak cukup”, maka mengecek dalam gudang apakah bahan tersedia?
9. Jika bahan baku atau alat untuk penggerjaan belum tersedia maka, melakukan pembelian ke *supplier*.
10. Barang dikirim dari *supplier* ke CV. Mitra Teknik.
11. Barang diterima oleh CV. Mitra Teknik, kemudian data pengeluaran dan nota pembayaran dimasukkan dalam buku perusahaan.
12. Kemudian proses penggerjaan barang.
14. Setelah proses penggerjaan barang yang akan dikirim dicek apakah sudah siap atau belum.
15. Jika barang sudah siap kirim “barang dikirim”, maka barang bisa langsung dikirim ke pelanggan.
16. Apabila barang sudah siap kirim “barang tidak dikirim”, maka barang masuk ke dalam gudang.
17. Jika barang siap dikirim, maka dokumen pengiriman dibuat, lalu menghitung biaya barang yang dikirim, dan menyimpan dokumen pengiriman disimpan dalam buku perusahaan.
18. Barang dikirim ke pelanggan.

19. Barang diterima pelanggan.

2.6.2 Proses bisnis sesudah dibangun sistem informasi penjualan



Gambar 2.10 Proses bisnis setelah dibangun sistem informasi manajemen penjualan

Aktivitas yang telah dianalisis dalam proses bisnis tidak banyak mengalami perubahan. Sebelumnya terdapat 24 task dan menjadi 19 task. Karena task pada bisnis proses sebelumnya seperti proses memasukkan data ke dalam buku

perusahaan, menyimpan bukti transaksi, atau perhitungan suatu barang sudah diubah dengan menggunakan sistem dan data yang dimasukkan disimpan ke dalam *database*. Berikut penjelasan dari gambar 2.10 sebagai berikut:

1. Dimulai dari pelanggan yang membutuhkan permintaan barang atau jasa.
2. Setelah itu pihak pelanggan membuat permintaan barang atau jasa dan dikirim melalui email atau telepon.
3. Setelah permintaan diterima, lalu CV. Mitra Teknik membuat penawaran.
4. Dokumen penawaran disimpan dalam sistem, dan dikirim melalui email.
5. Apakah penawaran yang telah dikirim “ditolak”, maka mengulang ke proses membuat penawaran.
6. Setelah penawaran diterima, lalu *order* diterima oleh CV. Mitra Teknik dan Memasukkan data biaya produksi kedalam fitur order masuk.
7. Sebelum ke tahap penggerjaan, dicek apakah bahan baku atau alat yang digunakan untuk proses penggerjaan sudah “cukup”, maka mengecek lewat sistem apakah bahan tersedia?
8. Sebelum ke tahap penggerjaan, dicek apakah bahan baku atau alat yang digunakan untuk proses penggerjaan sudah “tidak cukup”, maka mengecek lewat sistem apakah bahan tersedia?
9. Jika bahan baku atau alat untuk penggerjaan belum tersedia maka, melakukan pembelian BARANG ke *supplier*.
10. Barang dikirim dari *supplier* ke CV. Mitra Teknik.
11. Barang diterima oleh CV. Mitra Teknik, kemudian data pengeluaran disimpan dalam sistem.
12. Kemudian proses penggerjaan barang.
14. Setelah proses penggerjaan barang yang akan dikirim dicek apakah sudah siap atau belum.
15. Jika barang sudah siap kirim “barang dikirim”, maka barang bisa langsung dikirim ke pelanggan.
16. Apabila barang sudah siap kirim “barang tidak dikirim”, maka barang masuk ke dalam gudang.
17. Jika barang siap dikirim, maka dokumen pengiriman dibuat, lalu data pengiriman disimpan dalam sistem.
18. Barang dikirim ke pelanggan.
19. Barang diterima pelanggan.

2.7 Teknologi Perangkat Lunak

2.7.1 Cascading Style Sheet (CSS)

Cascading Style Sheet (CSS) dirancang untuk mengontrol tampilan dan nuansa halaman web. CSS pada awalnya dikembangkan di *SGML* pada tahun 1970, dan terus berkembang hingga hari ini. CSS didukung berbagai macam bahasa pemrograman *Markup* seperti: *HTML*, *XHTML*, *SVG (Scalable Vector Graphics)* dan *mozilla XUL (XML User interface Language)*. Dilihat dari pengertian bahasanya, *Cascading Style Sheet* berarti gaya tata letak halaman *multi-level*, jadi setiap elemen yang telah diformat memiliki turunan yang telah di format maka, turunan dari elemen tersebut mengikuti format induknya. (Alexander, 2013).

2.7.2 Hypertext Preprocessor (PHP)

PHP yakni bahasa pemrograman *script* yang dibuat untuk membangun sistem atau aplikasi web (Budi Raharjo, 2012). PHP sejauh ini merupakan bahasa pemrograman *script* yang sering banyak digunakan. PHP banyak diimplementasikan untuk membuat web yang dinamis, meskipun tidak tertutup dapat diimplementasikan untuk tujuan lain. Untuk saat ini penggunaan bahasa PHP paling tertinggi didunia menurut wikipedia. Didapatkan nilai 82% dari total penggunaan *web server* diseluruh dunia yang menggunakan PHP.

2.7.3 Hyper Text Markup Languange (HTML)

HTML yakni singkatan dari *Hypertext Markup Languange* dalam bentuk file teks yang ditulis menggunakan aturan atau kode tertentu kemudian ditampilkan kepada pengguna melalui *browser* web (Budi Raharjo, 2012). File HTML (atau halaman web) yang terdiri dari bagian-bagian *file* yang berfungsi untuk Sistem berhasil menyimpan ke *database* dan sistem menunjukkan tampilan website sesuai keinginan.

Dalam pembuatan web menggunakan bahasa HTML, terdapat beberapa tools untuk membantu pengguna mengerjakan sebuah website seperti contoh, *text editor* dan *web designer* kemudian file yang telah dibuat disimpan dalam ekstensi (*.html*).

2.7.4 Framework Model View Control (MVC)

Framework ialah kerangka kerja yang terstruktur digunakan *programmer* web untuk menulis kode program. MVC (*Model View Controller*) ialah konsep pemrograman yang memisahkan pemrograman logika aplikasi dari presentasinya.

Menurut (Burbeck, 1992) ada tiga komponen untuk membangun sebuah aplikasi yaitu:

1. **Model** merupakan struktur data. Proses berkaitan dengan *database* seperti memasukkan, mengambil, memodifikasi, dan menghapus basis data. Tersedia pada model *class* tertentu berisi fungsi kode-kode.
2. **View** merupakan informasi yang ditampilkan kepada pengguna. Yang berisi keseluruhan umum dari implementasi antarmuka pengguna dengan menarik komponen grafis yang menyampaikan pandangan tentang proses internal aplikasi dan mengarahkan aliran interaksi pengguna dengan aplikasi.
3. **Controller** merupakan wadah antara *Model* dan *View* dan semua *resource* yang diperlukan untuk memproses permintaan *HTTP*. Menjadi fungsi utama karena penampungan *event* yang dibuat oleh *user* dari *model* dan melakukan perbaruan terhadap komponen *view* menggunakan data dari *user*.

2.7.5 Database Management System (DBMS) MySQL

DBMS ialah perangkat lunak untuk membuat, memanajemen, dan menentukan pengaksesan basis data. Tujuannya adalah untuk menyediakan langkah-langkah yang benar, efisien dan nyaman untuk mengekstraksi dan penyimpanan data dari basis data (Bambang, 2004).

MySQL ialah *Relational Database Management System* (RDBMS) yang dibagikan secara *free* dibawah naungan GPL (*General Public License*). MySQL merupakan bagian dari DBMS *open source* yang gratis, dikembangkan, dan didistribusikan. SQL (*Structured Query Language*) ialah bahasa yang digunakan untuk membuat struktur untuk menyimpan data dalam database. SQL sering juga digunakan sebagai *query* dalam proses melakukan sebuah *query* pada sebuah database. SQL juga berfungsi untuk mendefinisikan data yang terstruktur, mementukan batas keamanan (*security*), memodifikasi data pada basis data, untuk mempertahankan kinerja *database* dan mengatur proses pada CPU paralel.

2.8 Sistem informasi penjualan

Penjualan ialah proses yang melibatkan penjual melakukan dan menentukan serta memenuhi kebutuhan atau keinginan pelanggan atau pembeli untuk saling menguntungkan antara penjual dan pembeli (Basu, 2001).

2.8.1 Konsep Penjualan

Konsep penjualan ialah menjual barang yang diperdagangkan yang menjadi berajalan dalam perusahaan dan menjadi bisnis inti perusahaan (Chairul, 2002). Penjualan juga merupakan salah satu langkah perusahaan untuk pemasaran, sehingga perusahaan memperoleh keuntungan dari hasil operasional kegiatan perusahaan kegiatan. Berikut penjelasan penjualan menurut beberapa ahli (Basu Swasta, 2001) dikutip dari bukunya yang membahas tentang manajemen penjualan edisi ketiga, ialah "Menjual adalah ilmu dan seni eksposur manusia yang dilakukan oleh penjual untuk mengundang orang lain agar mau membeli barang dan jasa yang ditawarkan" (Basu Swastha, 2001). Menurut

(Winardi, 2005), "Penjualan ialah bertemuanya seorang penjual dan pembeli yang bertujuan untuk tukar menukar barang dan jasa berdasarkan pertimbangan antara kedua belah pihak misalnya pertimbangan uang". Menurut (Thamrin Abdullah dan Francis Tantri, 2016) "Penjualan ialah bagian dari promosi dan promosi adalah salah satu bagian dari keseluruhan sistem pemasaran". Dalam kutipannya tiga tujuan penting pada penjualan:

1. Sampai batas tertentu penjualan
2. Mendapat keuntungan
3. Mendukung perkembangan perusahaan (Basu Swastha, 2001). Penjualan bila diidentifikasi berdasarkan perusahaannya maka dapat dikategorikan menjadi beberapa jenis, yaitu:

a. Penjualan langsung

Konsep penjualan ini yaitu dengan cara barang diambil dari pemasok dan dikirim ke pelanggan langsung.

b. Penjualan stok gudang

Konsep penjualan ini yaitu dengan cara persediaan barang diambil langsung dari gudang.

c. Penjualan Kombinasi

Ialah penjualan dengan cara mengambil sebagian barang dari *supplier* serta sebagian dari stok yang ada di gudang (Martin S, Colleran G, 2006). Dari pengertian tersebut keuntungan harus didapatkan bagi suatu usaha pada proses penjualan. Proses penjualan efektif dan efisien saat ini saat diperlukan maka diperlukan adanya sistem informasi. Penjualan dapat dibedakan menjadi beberapa bagian sebagai berikut :

1. **Trade selling** proses yang terjadi antara pedagang dan produsen apabila meningkatkan distribusi produk mereka melalui pengecer.
2. **Missionary Selling** proses yang dilakukan untuk mendorong pembeli agar membeli barang-barang yang mengakibatkan meningkatnya keuntungan penjualan.
3. **Technical Selling** proses untuk memberi nasihat kepada pembeli dan penjual yang mempunyai tujuan agar meningkatkan peningkatan penjualan.
4. **New Business Selling** proses untuk mencari transaksi atau membuka transaksi penjualan baru.
5. **Responsive Selling** proses terhadap permintaan barang dari setiap tenaga penjual untuk pembeli melalui route *driving* dan *retailing*. (Basu Swastha, 2001).

d. Penjualan Tunai.

Penjualan tunai merupakan cara untuk melakukan transaksi antara penjual dan pembeli dengan cara, penjual mengirim barang sesuai dengan bukti pembelian yang diterima oleh pembeli untuk proses penagihan dengan syarat pembeli tidak boleh membayar sesuai dengan tagihan pembayaran tidak boleh dilakukan dengan cara mengangsur.

e. Penjualan Kredit.

Penjualan kredit untuk saat ini paling sering dilakukan oleh perusahaan besar untuk melakukan transaksi jual beli. Metode pembayaran dengan cara diangsur dengan tempo dan jangka waktu yang sudah ditentukan. Metode inilah yang saat ini banyak digunakan di masyarakat luas.

f. Penjualan secara Tender.

Metode penjualan tender ini dilakukan dengan cara penjual membuka penawaran dan menunggu penawaran yang cocok dari pembeli. Apakah penawaran pembeli cocok atau tidak dengan penawaran yang dibuka.

g. Penjualan Ekspor.

Metode penjualan ini biasanya digunakan kepada pembeli atau penjual yang terletak di negara berbeda dalam proses transaksinya, Untuk transaksi ekspor menggunakan fasilitas *letter of credit* untuk proses pembayaran.

2.8.2 Teknologi Informasi (TI) Pada Perusahaan

Teknologi informasi menjadi alat komunikasi utama antara kondisi lingkungan dengan struktur atribut organisasi. Menurut (Richardus, 2014) Teknologi informasi berperan penting dan pengaruhnya sangat penting untuk hubungan antara struktur kompleksitas dengan lingkungan yang dinamis, kompleksitas lingkungan yang berhubungan dengan integrasi, hubungan antara kompleksitas lingkungan dengan kompleksitas struktural dan. Berikut 5 peran teknologi informasi pada perusahaan :

1. Fungsi operasional dalam organisasi menjadi lebih teratur dan jauh dari sifat birokratis karena adanya teknologi informasi. Karena teknologi informasi yang mempunyai sifat menyebar keseluruhan organisasi. Maka manajemen teknologi informasi berjalan sesuai dengan mestinya “*supporting agency*” saat ini teknologi informasi bisa dianggap sebagai “*firm infrastructure*”.
2. Fungsi pengawasan atau kontrol untuk keberadaan teknologi informasi merupakan bagian penting dikarenakan aktivitas di level menejerial dalam setiap fungsi manajer, Sehingga struktur pada organisasi terkait harus memiliki teknologi informasi “*span of control*” atau “*peer relationship*” yang membuat interaksi positif manajer antar perusahaan.
3. Fungsi pengambilan keputusan dan perencanaan untuk sebuah organisasi sangatlah penting. Teknologi informasi merupakan tatanan penting pada suatu perusahaan dimana rencana bisnis dan perencanaan sangat diperlukan untuk pimpinan perusahaan “*knowledge generator*”. Untuk saat ini kebanyakan

perusahaan memilih untuk membuat teknologi informasi untuk melangsungkan roda bisnis dan merupakan fungsi yang sangat strategis pada suatu perusahaan.

4. Fungsi Komunikasi termasuk ke dalam “*firm infrastructure*” dalam perkembangan industri saat ini dimana teknologi informasi sangat diperlukan untuk perusahaan dalam hal saling berkomunikasi, berkooperasi, dan saling berinteraksi. Manfaat adanya teknologi informasi mampu menunjang kinerja perusahaan.
5. Fungsi Interorganisasi ialah sebuah kondisi yang unik dikarenakan oleh perubahan pada perusahaan lingkup global yang memaksa antar perusahaan menjalin kerjasama atau bermitra berbasis teknologi informasi seperti implementasi pada *Enterprise resource planning* dan *supply chain management* yang dimana setiap perusahaan melakukan terobosan baru untuk membuat struktur organisasinya menjadi lebih baik. Untuk sekarang mungkin banyak perusahaan menerapkan kegiatan *outsourcing* terkait dengan pengalihdayaan untuk kelancaran bisnis perusahaan.

2.8.3 Manfaat Teknologi Informasi (TI) Pada Perusahaan

Faktor-faktor yang memengaruhi pemanfaatan teknologi Informasi (Richardus, 2014).

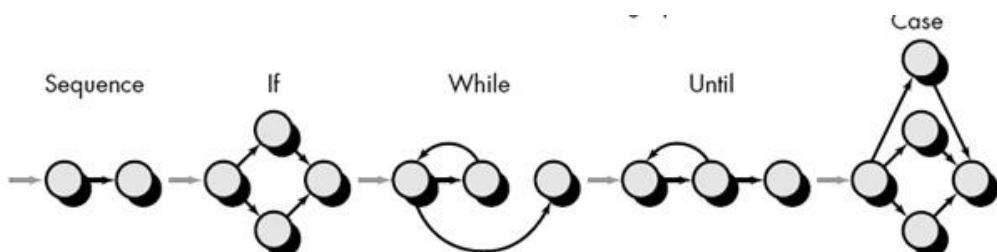
1. Faktor sosial (*social factor*), merupakan faktor internal sebuah kultur subjektif kelompok dalam kondisi persetujuan tertentu yang dibuat oleh individual dalam kondisi tertentu. Kultur subjektif berisi dan nilai-nilai (*value*) dan norma (*norm*).
2. *Affect*, merupakan tingkat perasaan antar individu dengan suatu pekerjaan, rasa suka atau tidak suka, menyenangkan atau tidak menyenangkan antar individu dalam penggunaan teknologi informasi.
3. Kompleksitas (*complexity*), sebagai tingkat inovasi yang dinilai relatif sulit untuk digunakan dan dimengerti. Dengan adanya teknologi informasi apakah mampu meningkatkan kinerja individu. Hubungan yang positif antara teknologi informasi dan individu dapat dibuktikan dengan hasil penelitian
4. Konsekuensi jangka panjang (*long-term consequences*). Kosekuensi yang dihasilkan apakah mampu hanya untuk memenuhi kebutuhan saat ini dan apakah mempunyai keuntungan dimasa yang akan datang.
5. Fasilitas sesuai kondisi (*facilitating condition*), dalam konteks peningkatan teknologi informasi, kondisi merupakan faktor utama dalam proses meningkatnya pemanfaatan teknologi informasi.

2.9 Pengujian

Untuk menilai apakah sistem yang kita bangun sudah layak atau tidak digunakan yaitu dengan cara pengujian. Yang menjadi parameter kondisi *input* dan *output* dari keluaran program yang diinginkan. Berikut pengujian yang dilakukan:

2.9.1 Pengujian White-Box

Pengujian *white-box* yakni metode pengujian dengan teknik memeriksa kode program atau jalur logika kode dalam modul setidaknya dilewati sekali untuk memastikan tidak terdapat masalah (Pressman dan Maxim, 2015). Dalam pengujian *white-box* terdapat *path coverage* atau *basis path testing* yang merupakan jenis pengujian yang sering digunakan untuk mendapatkan jalur independent kode program supaya tiap jalur logika dapat diperiksa dan untuk mengetahui tidak terdapat masalah dalam jalur.



Gambar 2.11 Notasi aliran graph (Pressman, 2010)

Basis Path testing yakni jenis white-box testing yang merupakan *flowgraph* dari pseudocode dari masing-masing fungsi. Selanjutnya menentukan ukuran kompleksitas (*Cyclomatic Complexity*). Pada sistem penjualan CV. Mitra Teknik digunakan untuk melihat apakah fungsi sistem benar-benar memanggil fungsi pada sistem layanan dan tidak memiliki masalah.

2.9.2 Pengujian Black-box

Pengujian *black-box* yakni teknik pengujian untuk melihat fungsi berjalan dengan baik dengan melihat kondisi *input* atau kondisi *output* yang diterima dan hasil yang dilakukan oleh sistem (Pressman dan Maxim, 2015). Dalam pengujian *black-box* terdapat *validation testing* yaitu yang bertujuan untuk menunjukkan apakah sistem berjalan sesuai dengan kebutuhan fungsional. Jika terdapat kesalahan pada saat pengujian, maka dapat dipastikan dengan pengujian *white-box* untuk melihat kesalahan kode program.

Pada sistem informasi manajemen penjualan pada CV. Mitra Teknik ini pengujian validasi pada pengujian *black-box* mengandalkan kasus uji dengan melihat kondisi *input-output* sesuai dengan skenario *use-case*.

2.9.3 Pengujian fungsional

Pada pengujian fungsional ini menggunakan teknik analisis deskriptif, yaitu dengan cara menganalisis persentase hasil dari pengujian setiap fungsi dari aplikasi oleh ahli pemrograman (*programmer* atau *developer*). Analisis data pada faktor *functional suitability* dilakukan dengan cara menghitung hasil pengujian menggunakan persentase keberhasilan, kemudian dilakukan konversi ke dalam bentuk predikat (Guritno *et al*, 2011):

$$\text{Persentase Keberhasilan} = \frac{l}{r} \times 100\%$$

Keterangan:

ι = Jumlah kebutuhan fungsional yang diimplementasikan

r = Jumlah total pengujian kebutuhan fungsional yang dilakukan

Hasil perhitungan kemudian dikonversikan ke dalam tabel predikat. Tabel predikat tersebut digunakan untuk mengukur kelayakan aplikasi yang dibuat (Putra, 2016). Terdapat lima kategori yang dimasukkan ke dalam tabel predikat kelayakan sebagai berikut.

Tabel 2.1 Predikat kelayakan aplikasi

Percentase	Predikat
0%-20%	Sangat Buruk
21%-40%	Buruk
41%-60%	Cukup
61%-80%	Baik
81%-100%	Sangat Baik

Pada tabel 2.1 merupakan persentase dari kelayakan aplikasi berdasarkan hasil total pengujian fungsional yang telah dikonversikan ke dalam persentase keberhasilan. Pada penelitian ini *functional suitability* digunakan untuk mengukur hasil dari pengujian fungsional pada rata-rata persentase keberhasilan pada pembangunan sistem informasi manajemen penjualan yang menggunakan *web service*.

2.9.4 User Acceptance Testing

User Acceptance Test (UAT) atau uji penerimaan pengguna adalah suatu proses pengujian oleh pengguna yang dimaksudkan untuk menghasilkan dokumen yang dijadikan bukti bahwa software yang telah dikembangkan dapat diterima oleh pengguna, apabila hasil pengujian (*testing*) sudah bisa dianggap memenuhi kebutuhan dari pengguna (Lereng & Fugelseth, 2004). Pengujian ini merupakan proses verifikasi bahwa sistem yang telah dibuat telah menyelesaikan masalah yang dihadapi oleh pengguna. Proses ini berbeda dengan pengujian yang dilakukan oleh programmer dalam sistem (menguji dan memastikan program tidak rusak dan sesuai dengan dokumen permintaan pengguna), melainkan memastikan bahwa sistem tersebut memahami penggunaan sistem yang telah dibuat

- a. Penentuan ukuran penelitian beserta bobot nilainya. Adapun penentuannya adalah sebagai berikut:

Tabel 2.2 Skala Penilaian (Qomaruddin, 2014)

Kategori	Bobot	Persentase
<i>Strongly Disagree</i>	1	0%-20%
<i>Disagree</i>	2	21%-40%
<i>Neutral</i>	3	41%-60%
<i>Agree</i>	4	61%-80%
<i>Strongly Agree</i>	5	81%-100%

2.9.5 Pengujian Security

Pengujian *security* menggunakan metode *MD5 (Message-Digest Algorithm 5)* adalah fungsi *hash* yaitu prosedur terdefinisi atau fungsi matematika yang mengubah variabel dari suatu data yang berukuran besar menjadi lebih sederhana) kriptografik yang digunakan secara luas dengan *hash value* 128-bit. *MD5* dimanfaatkan dalam berbagai aplikasi keamanan, dan umumnya digunakan untuk meguti integritas sebuah *file*. Enkripsi menggunakan *MD5* masih mendominasi sebagian besar aplikasi PHP. Enkripsi *MD5* dianggap strong karena enkripsi yang dihasilkannya bersifat '*one way hash*' atau enkripsi satu arah. Berapapun *string* yang di enkripsi hasilnya tetap sepanjang 32 karakter (Munir, 2006).

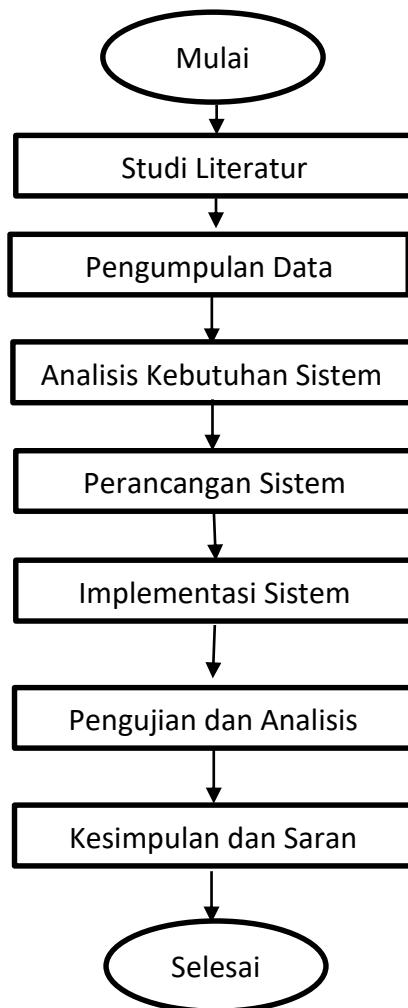
2.9.6 Pengujian Performance

Performance Testing yakni teknik pengujian non-fungsional yang dilakukan untuk mengetahui parameter sistem dalam hal respon dan stabilitas beban aplikasi atau *web*. Pengujian kinerja mengukur atribut kualitas sistem, seperti skalabilitas, kendala dan penggunaan sumber daya. Tidak hanya fitur dan fungsi saja yang harus di uji, *performance testing* juga sangat penting untuk memastikan aplikasi dapat diakses secara mudah oleh *user*.

Tujuan utama untuk *performance testing* bukanlah untuk mencari *bug* atau kesalahan sistem, tetapi untuk menghilangkan atau menghindari sistem tidak dapat diakses atau *load* yang terlalu lama yang mengakibatkan *feedback* buruk dari pengguna (Nielsen, 2010).

BAB 3 METODOLOGI PENELITIAN

Pada bab ini dibahas tahap-tahap yang diperlukan berdasarkan permasalahan yang telah diuraikan sebelumnya. Langkah pertama adalah melakukan studi literatur kemudian menganalisis kebutuhan dalam perancangan aplikasi. Selanjutnya dilakukan perancangan sistem aplikasi sampai implementasi *software*. Terakhir melakukan pengujian dan analisis untuk mendapatkan kesimpulan dari proses pembangunan sistem informasi penjualan pada perusahaan CV. Mitra teknik tersebut.



Gambar 3.1 Diagram Alir Metodologi Penelitian

Pada gambar 3.1 tahapan dalam diagram yang digunakan untuk menyelesaikan masalah.

Penelitian yang dilakukan ini akan menghasilkan produk/artefak baru yang sebelumnya belum pernah diterapkan pada objek penelitian yang lainnya sehingga permasalahan yang ada dapat terselesaikan dalam objek penelitian.

3.1 Studi Literatur

Pada studi literatur ini membahas apa saja yang dibutuhkan dalam membuat aplikasi. Literatur tersebut diambil dari berbagai sumber, jurnal, buku dokumentasi, internet, dan pustaka. Teori penunjang dan pendukung skripsi ini meliputi:

- Sistem Informasi
- Rekayasa Perangkat Lunak
- *Web Service*
- *Bussines Procces Modelling Notation (BPMN)*
- Proses bisnis CV. Mitra Teknik
- Teknologi Perangkat Lunak
- Sistem Informasi Penjualan
- Pengujian

3.2 Pengumpulan Data

Pada tahap ini perolehan data menggunakan tipe pengumpulan data kualitatif. Data kualitatif digunakan dalam aspek pemahaman secara mendalam mengenai suatu masalah dalam penelitian lingkup generalisasi. Data yang dikumpulkan meliputi *form* transaksi penjualan maupun pembelian pada perusahaan CV. Mitra Teknik. Data dikumpulkan guna untuk mendukung penelitian, data yang didapat ada dua meliputi data primer dan sekunder. Data primer diperoleh dari wawancara secara langsung yang berkaitan dengan topik penelitian yaitu pemilik perusahaan CV. Mitra Teknik. Sedangkan data sekunder yang diperoleh berupa berkas atau bukti transaksi penjualan atau pengiriman dalam perusahaan CV. Mitra Teknik.

3.3 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem berfungsi untuk menggali kebutuhan yang dilakukan oleh seseorang guna untuk mengetahui seperti apa spesifikasi kebutuhan sistem yang akan dibangun. Seperti kebutuhan fungsional dan kebutuhan non-fungsional sistem, aktor yang menjadi pengguna yang berkepentingan dalam sistem, dan menentukan fitur apa saja yang akan digunakan dalam sistem. Analisis yang sudah didapatkan dimodelkan kedalam *use case diagram*, Dari penggambaran *use case diagram* dapat diketahui fungsi-fungsi apa saja yang digunakan dan aktor siapa saja yang berkaitan. Setelah dibuat *use case diagram* untuk berikutnya dibuat *use case scenario* yang berfungsi untuk mendeskripsikan cara kerja sistem secara detail.

3.4 Perancangan Sistem

Melakukan desain terhadap aplikasi yang dibuat agar aplikasi berjalan sesuai kebutuhan pengguna, melalui perancangan, analisis, observasi, dan merancang sistem untuk memodelkan bentuk keseluruhan dari aplikasi. Pada bagian perancangan sistem ini membahas gambaran-gambaran umum sistem

dibuat. Dalam rancangan ini mempunyai komponen sistem informasi yang diidentifikasi.

3.4.1 Tahap perancangan sistem

Berikut perancangan yang akan digunakan sebagai referensi untuk memenuhi kebutuhan sistem informasi penjualan.

1. *Sequence Diagram*, pada penelitian ini *Sequence Diagram* akan menggambarkan *scenario* untuk menunjukkan interaksi antara aktor dan sistem.
2. *Class Diagram*, pada penelitian ini *Class Diagram* dibutuhkan untuk mendeskripsikan kelas-kelas yang ada pada sistem. Diagram ini juga menjadi acuan dalam pengimplementasian kode program.
3. *Physical Data Model*, pada penelitian ini *Physical Data Model* digunakan untuk merancang struktur *database* yang digunakan dalam sistem, dari *database* tersebut terdapat tabel-tabel yang saling berelasi. Dengan *database* sistem akan dimudahkan dalam penyimpanan dan integrasi data pada sistem.
4. Rancangan Desain Antarmuka, pada penelitian ini rancangan desain antarmuka dibutuhkan untuk pembuatan sebuah sistem informasi.

3.5 Implementasi

Pada tahap implementasi, sistem akan dibuat mengacu pada tahap perancangan. Implementasi sistem dibuat dengan menggunakan bahasa pemrograman web yang terdiri dari: HTML, CSS, PHP, dan JavaScript kemudian menggunakan teknologi *web service*.

Model yang digunakan untuk pembangunan sistem ini adalah pendekatan *waterfall* model dan menggunakan Codeigniter sebagai *framework* untuk struktur dalam pembuatan sistem informasi penjualan pada CV. Mitra Teknik tersebut.

3.6 Pengujian sistem

Pengujian sistem merupakan tahap terakhir dalam pembangunan sistem. Tahap pengujian ini bertujuan untuk menguji apakah sistem yang dibangun telah memenuhi spesifikasi yang telah diminta oleh pengguna. Dalam penelitian ini pengujian yang dilakukan meliputi pengujian *white-box*, pengujian *black-box*, pengujian *user acceptance testing*.

Pengujian *white-box* berpusat pada pengujian unit pseudocode dalam sistem memastikan apakah komponen yang diuji sudah berjalan dengan baik. Pengujian validasi digunakan pada pengujian *black-box*. Pengujian ini dilakukan memeriksa apakah sistem sudah berjalan dengan baik, bisa dilihat pada proses *input-output* dalam sistem dan bisa dipastikan apakah sistem yang diuji itu *valid* atau *invalid*. Sedangkan pengujian *user acceptance testing* menguji apakah sistem sudah dapat diterima oleh pengguna.

3.7 Kesimpulan dan Saran

Dengan selesainya kegiatan penelitian, perancangan sistem hingga implementasi pada sistem informasi ini, penulis menarik kesimpulan sebagai berikut. Sistem informasi ini memberikan layanan informasi dan kemudahan bagi perusahaan dalam hal mengelola data transaksi, barang, ataupun yang lain-lainnya dan menjawab masalah yang dirumuskan pada latar belakang.

BAB 4 ANALISIS KEBUTUHAN

Bab ini membahas mengenai daftar kebutuhan dan perancangan dari Sistem Informasi Penjualan. Pada tahap ini analisis kebutuhan dimulai dengan tahap elisitasi, Identifikasi kepentingan, Spesifikasi kebutuhan yang berisi kebutuhan fungsional dan kebutuhan non-fungsional. Kemudian tahap perancangan sistem.

4.1 Elitisasi Kebutuhan

Bab elitisasi kebutuhan pada sistem informasi penjualan dilakukan untuk mendeskripsikan masalah sesuai dengan hasil observasi dan proses pada wawancara dalam perusahaan. Selanjutnya dengan mengidentifikasi kebutuhan sistem yang akan digunakan untuk menjawab permasalahan yang ada.

4.1.1 Teknik Elitisasi

Pada tahap ini teknik metode wawancara dan observasi yang digunakan untuk mencari informasi. Wawancara dilakukan kepada pengguna yang berkepentingan yaitu dengan direktur perusahaan yang bersangkutan dengan cara menggali informasi menangani alur pencatatan transaksi penjualan atau pembelian dalam perusahaan. Untuk teknik observasi yang telah dilakukan yaitu untuk melihat alur transaksi penjualan dan pembelian yang telah dikerjakan, berikut kebutuhan yang didapatkan dijelaskan pada tabel 4.1:

Tabel 4.1 Temuan kebutuhan

No.	Temuan Kebutuhan
1	Belum adanya sistem yang dapat mengolah dan menyimpan data penawaran
2	Belum adanya sistem yang dapat mengolah, menyimpan riwayat penjualan dan pembelian yang dilakukan oleh perusahaan
3	Belum adanya sistem yang mengolah data <i>supplier</i> dan pelanggan

4.2 Identifikasi Stakeholder

Pada tahap ini hal yang dilakukan adalah mengidentifikasi dan menganalisis pengguna bersangkutan atau *stakeholder* yang berhubungan dengan pembangunan sistem dengan menjelaskan tipe pemangku kepentingan pada sistem yang akan dibuat. Identifikasi *stakeholder* diantaranya sebagai berikut.

Tabel 4.2 Identifikasi Stakeholder

No	Tipe <i>stakeholder</i>	Deskripsi <i>stakeholder</i>	<i>Stakeholder</i> yang relevan
1	Pengguna	Orang yang berkepentingan untuk menjalankan sistem informasi penjualan	Admin perusahaan
2	Pengembang	Orang yang berkepentingan dan yang bertanggung jawab penuh atas pembangunan sistem informasi manajemen penjualan pada CV. Mitra Teknik	Mahasiswa

Pada tabel 4.2 didapatkan dua pihak yang berkepentingan atau menjadi *stakeholder* dalam sistem, dari identifikasi yang didapatkan yaitu pengguna dan pengembang. Pengguna yakni orang yang menjalankan sistem informasi manajemen penjualan. Sedangkan pengembang atau yang relevan yaitu mahasiswa bertanggung jawab penuh untuk pembangunan sistem informasi penjualan CV. Mitra Teknik.

4.3 Spesifikasi Kebutuhan

Sistem informasi penjualan mempunyai spesifikasi kebutuhan yang dapat dijelaskan sesuai dengan kebutuhan dan solusi dari permasalahan yang telah diidentifikasi.

4.3.1 Kebutuhan Fungsional Sistem

Fungsional sistem yakni fungsi utama yang dibutuhkan untuk berjalannya sistem informasi penjualan ini. Daftar kebutuhan fungsional untuk sistem informasi penjualan dapat dilihat pada tabel 4.3 berikut.

Tabel 4.3 Spesifikasi kebutuhan Fungsional Sistem Informasi Penjualan

ID SIP	<i>Functional Requirement</i>	<i>Use case</i>	User
SIP_1_01	Sistem membutuhkan fitur tombol <i>order</i> masuk untuk mengakses fitur penawaran	Melihat data penawaran	Admin
SIP_1_02	Sistem membutuhkan fitur menambahkan data penawaran	Menambahkan penawaran	Admin
SIP_1_03	Sistem membutuhkan fitur mengubah data penawaran	Mengubah Penawaran	Admin

Tabel 4.3 Spesifikasi kebutuhan Sistem Informasi Penjualan (Lanjutan)

ID SIP	<i>Functional Requirement</i>	<i>Use case</i>	<i>User</i>
SIP_1_04	Sistem membutuhkan fitur fungsi cetak untuk men-generate form ke format PDF	Mencetak penawaran	Admin
SIP_1_05	Sistem membutuhkan fitur menghapus data penawaran	Menghapus penawaran	Admin
SIP_1_06	Sistem membutuhkan fitur tombol penjualan untuk mengakses fitur penjualan	Melihat Penjualan	Admin
SIP_1_07	Sistem membutuhkan fitur tombol <i>order</i> masuk untuk mengakses fitur <i>order</i> masuk	Melihat <i>Order</i> masuk	Admin
SIP_1_08	Sistem membutuhkan fitur untuk menambah data <i>order</i> masuk	Menambahkan <i>order</i> masuk	Admin
SIP_1_09	Sistem membutuhkan fitur untuk merubah data <i>order</i> masuk	Mengubah <i>order</i> masuk	Admin
SIP_1_10	Sistem membutuhkan fitur untuk menghapus data <i>order</i> masuk	Menghapus <i>order</i> masuk	Admin
SIP_1_11	Sistem membutuhkan fitur tombol pengiriman untuk mengakses fitur pengiriman	Melihat Pengiriman	Admin

Tabel 4.3 Spesifikasi kebutuhan Sistem Informasi Penjualan (Lanjutan)

ID SIP	Functional Requirement	Use case	User
SIP_1_12	Sistem membutuhkan fitur untuk menambah data pengiriman	Menambahkan pengiriman	Admin
SIP_1_13	Sistem membutuhkan fitur untuk mengubah data pengiriman	Mengubah pengiriman	Admin
SIP_1_14	Sistem membutuhkan fitur untuk menghapus data pengiriman	Menghapus pengiriman	Admin
SIP_1_15	Sistem membutuhkan fitur tombol list stok barang untuk melihat stok data barang	Melihat stok barang	Admin
SIP_1_16	Sistem membutuhkan fitur untuk menambah data stok barang	Menambahkan stok barang	Admin
SIP_1_17	Sistem membutuhkan fitur untuk menambah pembelian barang	Menambahkan pembelian	Admin
SIP_1_18	Sistem membutuhkan fitur menyediakan tombol <i>supplier</i> untuk mengakses fitur <i>supplier</i>	Melihat <i>Supplier</i>	Admin
SIP_1_19	Sistem membutuhkan fitur tombol tambah untuk menambah data <i>supplier</i>	Menambahkan <i>supplier</i>	Admin
SIP_1_20	Sistem membutuhkan fitur tombol <i>edit</i> untuk mengubah data <i>supplier</i>	Mengubah <i>supplier</i>	Admin
SIP_1_21	Sistem membutuhkan fitur tombol hapus untuk menghapus data <i>supplier</i>	Menghapus <i>supplier</i>	Admin

Tabel 4.3 Spesifikasi kebutuhan Sistem Informasi Penjualan (Lanjutan)

ID SIP	Functional Requirement	Use case	User
SIP_1_22	Sistem membutuhkan fitur tombol pelanggan untuk mengakses fitur pelanggan	Melihat pelanggan	Admin
SIP_1_23	Sistem membutuhkan fitur untuk menambah data pelanggan	Menambahkan pelanggan	Admin
SIP_1_24	Sistem membutuhkan fitur untuk mengubah data pelanggan	Mengubah pelanggan	Admin
SIP_1_25	Sistem membutuhkan fitur untuk menghapus data pelanggan	Menghapus pelanggan	Admin
SIP_1_26	Sistem membutuhkan fitur tombol <i>Logout</i> untuk aksi aktor keluar dari sistem	<i>Logout</i>	Admin
SIP_1_27	Sistem membutuhkan fitur tambah <i>user</i> untuk menambah <i>user</i> baru	Menambahkan <i>user</i>	Admin
SIP_1_28	Sistem membutuhkan fitur tampil laporan untuk menampilkan laporan pembelian	Melihat laporan pembelian	Admin
SIP_1_29	Sistem membutuhkan fitur tampil laporan untuk menampilkan laporan pengiriman	Melihat laporan pengiriman	Admin

Tabel 4.4 Kebutuhan fungsional aktor tamu

ID SIP	Kebutuhan	Use case	User
SIP_2_01	Sistem harus menyediakan autentikasi agar pengguna dapat masuk ke dalam sistem	<i>Login</i>	Tamu

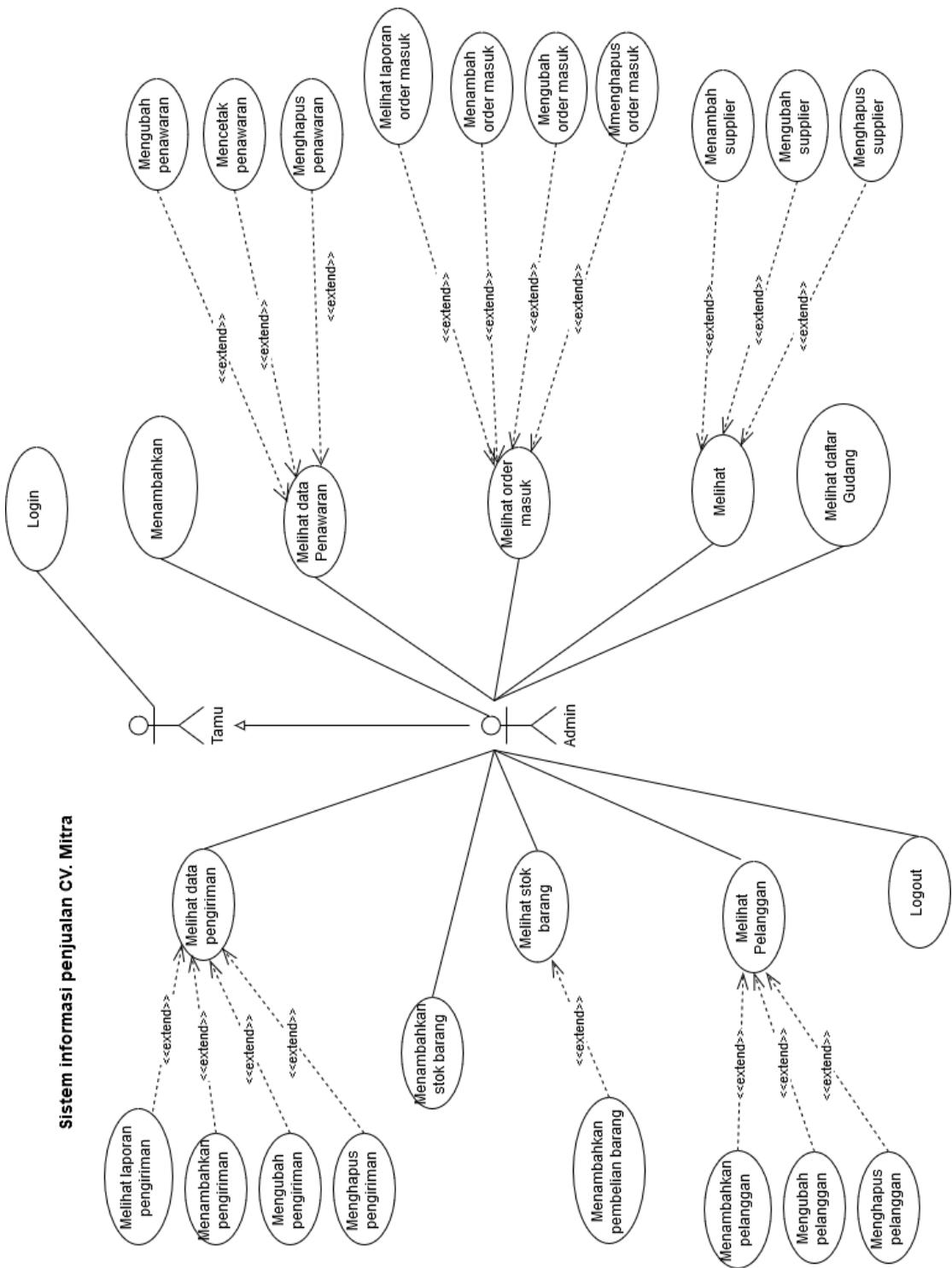
4.3.2 Kebutuhan Non-Fungsional Sistem

Kebutuhan non-fungsional merupakan kebutuhan penunjang yang berhubungan dengan pengujian dengan petunjuk tertentu seperti performa atau keamanan suatu sistem. Berikut adalah daftar kebutuhan non-fungsional dari sistem informasi manajemen penjualan.

Tabel 4.5 Kebutuhan non-fungsional

Parameter	Deskripsi Kebutuhan
<i>Security</i>	Sistem menggunakan metode enkripsi <i>md5</i> untuk data <i>password</i> yang tersimpan dalam <i>database</i>
<i>Performance</i>	Eksekusi dalam sistem dengan waktu maksimal 1000ms (<i>millisecond</i>)

4.4 Diagram Use Case



Gambar 4.1 Use case diagram sistem informasi penjualan

Pada gambar 4.1 di atas memperlihatkan diagram *use case* sistem informasi manajemen penjualan pada CV. Mitra Teknik.

4.5 Use Case Scenario

Use case scenario merupakan alur jalannya dari proses *use case* dari sisi aktor dan sistem. *Use case scenario* dari sistem informasi penjualan akan dideskripsikan pada bagian berikut.

4.5.1 Use case scenario Melihat Penawaran

Tabel 4.6 Tabel *use case scenario* Melihat Penawaran

Actor	Admin
Objective	Fungsi untuk melihat daftar list penawaran
Pre-Condition	Aktor telah masuk kedalam sistem
Main Flow	<ol style="list-style-type: none">1. Sistem menunjukkan tampilan halaman utama2. Aktor memilih fitur penawaran3. Tampilan <i>dropdown</i> menu pada fitur penawaran4. Aktor memilih fitur list penawaran
Alternative Flow	-
Post-Condition	Sistem menunjukkan tampilan data pada fitur list penawaran

4.5.2 Use case scenario Menambahkan Penawaran

Tabel 4.7 Tabel *use case scenario* Menambahkan Penawaran

Actor	Admin
Objective	Fungsi untuk menambah data penawaran
Pre-Condition	Aktor telah masuk kedalam sistem
Main Flow	<ol style="list-style-type: none">1. Aktor memilih fitur tambah penawaran2. Aktor mengisi <i>form</i> yang meliputi tanggal, nama pelanggan, nama barang, material, harga satuan, <i>quantity</i>, total harga, dan keterangan.3. Aktor menekan tombol simpan untuk menyimpan data penawaran4. Apabila berhasil menyimpan data penawaran, maka sistem akan menampilkan fitur list penawaran
Alternative Flow	3.1 Akan muncul notifikasi " <i>This is a required field</i> " yang menandakan form harus diisi, apabila form tidak diisi lengkap maka data gagal disimpan
Post-Condition	Sistem menunjukkan tampilan data terbaru pada fitur list penawaran

4.5.3 Use case scenario Mengubah Penawaran

Tabel 4.8 Tabel use case scenario Mengubah Penawaran

Actor	Admin
Objective	Fungsi untuk mengubah data penawaran
Pre-Condition	Aktor telah masuk ke dalam fitur list <i>order</i> masuk
Main Flow	<ol style="list-style-type: none"> 1. Sistem menampilkan fitur list penawaran 2. Aktor menekan tombol <i>edit</i> 3. Tampilan <i>pop-up</i> yang berisi <i>form</i> meliputi tanggal, nama pelanggan, nama barang, material, harga satuan, <i>quantity</i>, total harga, dan keterangan, status. 4. Aktor menekan tombol simpan, untuk menyimpan data yang telah diedit
Alternative Flow	4.1. Jika aktor menekan tombol “ <i>Cancel</i> ” maka sistem tidak akan melakukan eksekusi edit data
Post-Condition	Sistem menunjukkan tampilan data terbaru pada fitur list penawaran

4.5.4 Use case scenario Mencetak Penawaran

Tabel 4.9 Tabel use case scenario Mencetak Penawaran

Actor	Admin
Objective	Fungsi untuk mencetak Form Penawaran
Pre-Condition	Aktor telah masuk ke dalam fitur list penawaran
Main Flow	<ol style="list-style-type: none"> 1. Sistem menampilkan fitur list penawaran 2. Aktor memilih barang yang akan dicetak 3. Aktor menekan tombol cetak untuk menampilkan detail barang yang dipilih ke dalam PDF
Alternative Flow	-
Post-Condition	Sistem menunjukkan tampilan form PDF

4.5.5 Use case scenario Menghapus Penawaran

Tabel 4.10 Tabel use case scenario Menghapus Penawaran

Actor	Admin
Objective	Fungsi untuk menghapus data penawaran
Pre-Condition	Aktor telah masuk ke dalam fitur list penawaran
Main Flow	<ol style="list-style-type: none">1. Aktor telah masuk fitur dalam list penawaran2. Aktor menekan tombol hapus dalam tabel aksi kemudian muncul notif kemudian tekan tombol "OK"3. Sistem menghapus barang yang dipilih
Alternative Flow	3.1. Jika aktor menekan tombol "Cancel" maka sistem tidak akan melakukan eksekusi hapus data
Post-Condition	Sistem menunjukkan tampilan data terbaru pada fitur list penawaran

4.5.6 Use case scenario Melihat Order masuk

Tabel 4.11 Tabel use case scenario Melihat Order masuk

Actor	Admin
Objective	Fungsi untuk mengakses fitur List Order masuk
Pre-Condition	Aktor telah masuk ke dalam sistem informasi penjualan
Main Flow	<ol style="list-style-type: none">1. Aktor memilih fitur list Order masuk2. Tampilan dropdown menu pada fitur penawaran
Alternative Flow	-
Post-Condition	menunjukkan tampilan data pada fitur halaman list order masuk

4.5.7 Use case scenario Menambahkan *order* masuk

Tabel 4.12 Tabel use case scenario Menambahkan *order* masuk

Actor	Admin
Objective	Fungsi untuk menambah data <i>order</i> masuk
Pre-Condition	Aktor telah masuk ke dalam fitur Tambah <i>order</i> masuk
Main Flow	<ol style="list-style-type: none"> 1. Sistem menampilkan fitur tambah <i>order</i> masuk 2. Aktor mengisi <i>form</i> yang meliputi no po, nama pelanggan, nama barang, harga satuan, <i>quantity</i>, harga total, keterangan, tanggal masuk, status. 3. Aktor menekan tombol simpan untuk menyimpan data <i>order</i> masuk 4. Apabila berhasil menyimpan, maka sistem akan menampilkan fitur list <i>order</i> masuk
Alternative Flow	<ol style="list-style-type: none"> 2.1. Akan muncul notifikasi "<i>This is a required field</i>" yang menandakan form harus diisi, apabila form tidak diisi lengkap maka data gagal tersimpan
Post-Condition	Sistem menunjukkan tampilan data terbaru pada fitur list <i>order</i> masuk

4.5.8 Use case scenario Mengubah *order* masuk

Tabel 4.13 Tabel use case scenario Mengubah *order* masuk

Actor	Admin
Objective	Fungsi untuk mengubah data <i>order</i> masuk
Pre-Condition	Aktor telah masuk ke dalam fitur list <i>order</i> masuk
Main Flow	<ol style="list-style-type: none"> 1. Sistem menampilkan fitur list <i>order</i> masuk 2. Aktor menekan tombol <i>edit</i> 3. Tampilan <i>pop-up</i> yang berisi <i>form</i> meliputi no po, nama pelanggan, nama barang, harga satuan, <i>quantity</i>, harga total, keterangan, tanggal masuk, status. 4. Aktor menekan tombol simpan, untuk menyimpan data yang telah diedit
Alternative Flow	<ol style="list-style-type: none"> 4.1. Jika aktor menekan tombol "<i>Cancel</i>" maka sistem tidak akan melakukan eksekusi edit data
Post-Condition	Sistem menunjukkan tampilan data terbaru pada fitur list <i>order</i> masuk

4.5.9 Use case scenario Menghapus *order* masuk

Tabel 4.14 Tabel use case scenario Menghapus *order* masuk

Actor	Admin
Objective	Fungsi untuk menghapus data <i>order</i> masuk
Pre-Condition	Pengguna telah masuk ke dalam fitur <i>order</i> masuk
Main Flow	<ol style="list-style-type: none"> 1. Sistem menampilkan fitur list <i>order</i> masuk 2. Aktor memilih barang yang akan dihapus 3. Aktor menekan tombol hapus dalam tabel aksi kemudian muncul notif kemudian tekan tombol “Ok” 4. Sistem menampilkan data yang sudah ter-update
Alternative Flow	<ol style="list-style-type: none"> 3.1. Jika aktor menekan tombol “Cancel” maka sistem tidak akan melakukan eksekusi hapus data
Post-Condition	Sistem menunjukkan tampilan data terbaru pada fitur list <i>order</i> masuk

4.5.10 Use case scenario Menambahkan pengiriman

Tabel 4.15 Tabel use case scenario Menambahkan pengiriman

Actor	Admin
Objective	Fungsi untuk menambah data pengiriman
Pre-Condition	Aktor telah masuk ke dalam fitur tambah pengiriman
Main Flow	<ol style="list-style-type: none"> 1. Sistem menampilkan fitur tambah pengiriman 2. Aktor mengisi <i>form</i> yang meliputi no surat jalan, no kendaraan, nama pelanggan, nama barang, harga satuan, <i>quantity</i>, harga total, keterangan, tanggal penyerahan, status. 3. Aktor menekan tombol simpan untuk menyimpan data pengiriman 4. Sistem menampilkan data yang sudah ter-update
Alternative Flow	<ol style="list-style-type: none"> 2.1. Akan muncul notifikasi “This is a required field” yang menandakan form harus diisi, apabila form tidak diisi lengkap maka data tidak bisa disimpan
Post-Condition	Sistem menampilkan data terbaru dalam fitur list pengiriman

4.5.11 Use case scenario Mengubah pengiriman

Tabel 4.16 Tabel use case scenario Mengubah pengiriman

Actor	Admin
Objective	Fungsi untuk mengubah data pengiriman
Pre-Condition	Aktor telah masuk ke dalam fitur pengiriman
Main Flow	<ol style="list-style-type: none"> 1. Sistem menampilkan fitur pengiriman 2. Aktor menekan tombol <i>edit</i> 3. Tampilan <i>pop-up</i> yang berisi <i>form</i> yang harus diisi oleh pengguna meliputi no surat jalan, no kendaraan, nama pelanggan, nama barang, harga satuan, <i>quantity</i>, harga total, keterangan, tanggal penyerahan, status. 4. Sistem menampilkan data yang sudah ter-<i>update</i>
Alternative Flow	2.1. Jika Pengguna menekan tombol “ <i>Cancel</i> ” maka sistem tidak akan melakukan eksekusi edit data
Post-Condition	Sistem menunjukkan tampilan data terbaru dalam fitur list pengiriman

4.5.12 Use case scenario Menghapus pengiriman

Tabel 4.17 Tabel use case scenario Menghapus pengiriman

Actor	Admin
Objective	Fungsi untuk menghapus data pengiriman
Pre-Condition	Aktor telah masuk ke dalam fitur pengiriman
Main Flow	<ol style="list-style-type: none"> 1. Sistem menampilkan fitur pengiriman 2. Aktor memilih data yang akan dihapus 3. Aktor menekan tombol hapus dalam tabel aksi kemudian muncul notif kemudian tekan tombol “<i>OK</i>” 4. Sistem menampilkan data yang sudah ter-<i>update</i>
Alternative Flow	3.1. Jika aktor menekan tombol “ <i>Cancel</i> ” maka sistem tidak akan melakukan eksekusi hapus data
Post-Condition	Sistem menunjukkan tampilan data terbaru dalam fitur list pengiriman

4.5.13 Use case scenario Menambahkan pembelian barang

Tabel 4.18 Tabel use case scenario Menambahkan pembelian barang

Actor	Admin
Objective	Fungsi untuk menambah data pembelian barang
Pre-Condition	Aktor telah masuk ke dalam sistem informasi penjualan
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih fitur Stok gudang 2. Aktor memilih menu pada <i>drop-down</i> input pembelian barang 3. Sistem menampilkan <i>form</i> input pembelian barang 4. Aktor mengisi <i>form</i> yang meliputi no po, nama <i>supplier</i>, nama barang, harga satuan, <i>quantity</i>, harga total, keterangan, tanggal beli, status. 5. Aktor menekan tombol simpan untuk menyimpan data pengiriman
Alternative Flow	2.1. Jika data tidak berhasil disimpan maka akan muncul notifikasi " <i>This is a required field</i> " yang menandakan <i>form</i> harus diisi lengkap
Post-Condition	Sistem menampilkan data dalam fitur list pembelian

4.5.14 Use case scenario Melihat Stok Barang

Tabel 4.19 Tabel use case scenario Melihat stok barang

Actor	Admin
Objective	Fungsi untuk melihat data stok barang
Pre-Condition	Aktor telah masuk ke dalam sistem informasi penjualan
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih fitur stok gudang 2. Aktor memilih tombol list stok gudang 3. Sistem menampilkan data list stok barang
Alternative Flow	-
Post-Condition	Sistem menunjukkan tampilan data stok barang

4.5.15 Use case scenario Menambahkan supplier

Tabel 4.20 Tabel use case scenario Menambahkan supplier

Actor	Admin
Objective	Fungsi untuk menambah data <i>supplier</i>
Pre-Condition	Aktor telah masuk ke dalam fitur <i>supplier</i>
Main Flow	<ol style="list-style-type: none"> 1. Sistem menampilkan fitur <i>supplier</i> 2. Aktor mengisi <i>form</i> yang meliputi nama <i>supplier</i>, telepon <i>supplier</i>, alamat <i>supplier</i> 3. Aktor menekan tombol simpan untuk menyimpan data pengiriman 4. Sistem menampilkan data yang sudah ter-update
Alternative Flow	<ol style="list-style-type: none"> 2.1 Jika data tidak berhasil disimpan maka akan muncul notifikasi "<i>This is a required field</i>" yang menandakan <i>form</i> harus diisi lengkap
Post-Condition	Sistem menunjukkan tampilan data dalam fitur list pembelian

4.5.16 Use case scenario Mengubah supplier

Tabel 4.21 Tabel use case scenario Mengubah supplier

Actor	Admin
Objective	Fungsi untuk mengubah data <i>supplier</i>
Pre-Condition	Aktor telah masuk ke dalam fitur <i>supplier</i>
Main Flow	<ol style="list-style-type: none"> 1. Sistem menampilkan fitur <i>supplier</i> 2. Aktor menekan tombol <i>edit</i> 3. Tampilan <i>pop-up</i> yang berisi <i>form</i> yang harus diisi oleh admin meliputi nama <i>supplier</i>, telepon <i>supplier</i>, alamat <i>supplier</i>
Alternative Flow	<ol style="list-style-type: none"> 2.1 Jika aktor menekan tombol "<i>Cancel</i>" maka sistem tidak akan melakukan eksekusi hapus data 3.1. Jika data tidak berhasil disimpan maka akan muncul notifikasi "<i>Please fill out this field</i>" yang menandakan <i>form</i> harus diisi lengkap
Post-Condition	Sistem menunjukkan tampilan data dalam fitur <i>supplier</i>

4.5.17 Use case scenario Menghapus supplier

Tabel 4.22 Tabel use case scenario Menghapus supplier

Actor	Admin
Objective	Fungsi untuk menghapus data <i>supplier</i>
Pre-Condition	Aktor telah masuk ke dalam fitur <i>supplier</i>
Main Flow	<ol style="list-style-type: none"> 1. Sistem menampilkan fitur <i>supplier</i> 2. Aktor memilih data yang akan dihapus 3. Aktor menekan tombol hapus dalam tabel aksi kemudian muncul notif kemudian tekan tombol “Ok” 4. Sistem menampilkan data yang sudah ter-update
Alternative Flow	<ol style="list-style-type: none"> 3.1. Jika Pengguna menekan tombol “Cancel” maka sistem tidak akan melakukan eksekusi hapus data
Post-Condition	Sistem menunjukkan tampilan data dalam fitur list <i>supplier</i>

4.5.18 Use case scenario Menambahkan pelanggan

Tabel 4.23 Tabel use case scenario Menambahkan pelanggan

Actor	Admin
Objective	Fungsi untuk menambah data pelanggan
Pre-Condition	Aktor telah masuk ke dalam fitur pelanggan
Main Flow	<ol style="list-style-type: none"> 1. Sistem menampilkan fitur pembelian 2. Aktor mengisi <i>form</i> yang meliputi nama pelanggan, alamat pelanggan, telepon pelanggan, status 3. Aktor menekan tombol tambah untuk menyimpan data pengiriman 4. Jika berhasil menyimpan, maka tabel yang ada di bawah <i>form</i> akan otomatis ter-update
Alternative Flow	<ol style="list-style-type: none"> 2.1. Jika data tidak berhasil disimpan maka akan muncul notifikasi “This is a required field” yang menandakan <i>form</i> harus diisi lengkap
Post-Condition	Sistem menunjukkan tampilan data dalam fitur list pelanggan

4.5.19 Use case scenario Mengubah pelanggan

Tabel 4.24 Tabel use case scenario Mengubah pelanggan

Actor	Admin
Objective	Fungsi untuk mengubah data pelanggan
Pre-Condition	Aktor telah masuk ke dalam fitur pelangan
Main Flow	<ol style="list-style-type: none"> 1. Sistem menampilkan fitur pelangan 2. Aktor menekan tombol <i>edit</i> 3. Sistem menampilkan <i>pop-up</i> yang berisi <i>form</i> yang harus diisi oleh admin meliputi nama pelanggan, alamat pelanggan, telepon pelanggan, status
Alternative Flow	<ol style="list-style-type: none"> 2.1. Jika aktor menekan tombol “<i>Cancel</i>” maka sistem tidak akan melakukan eksekusi hapus data 2.3 Jika data tidak berhasil disimpan maka akan muncul notifikasi “<i>Please fill out this field</i>” yang menandakan <i>form</i> harus diisi lengkap
Post-Condition	Sistem menunjukkan tampilan data dalam fitur list pelanggan

4.5.20 Use case scenario Menghapus pelanggan

Tabel 4.25 Tabel use case scenario Menghapus pelanggan

Actor	Admin
Objective	Fungsi untuk menghapus data pelanggan
Pre-Condition	Aktor telah masuk ke dalam fitur pelangan
Main Flow	<ol style="list-style-type: none"> 1. Sistem menampilkan fitur pelangan 2. Aktor memilih data yang akan dihapus 3. Aktor menekan tombol hapus dalam tabel aksi kemudian muncul notif kemudian tekan tombol “<i>Ok</i>” 4. Sistem menampilkan data yang sudah ter-update
Alternative Flow	<ol style="list-style-type: none"> 3.1. Jika Pengguna menekan tombol “<i>Cancel</i>” maka sistem tidak akan melakukan eksekusi hapus data
Post-Condition	Sistem menunjukkan tampilan data dalam fitur list pelanggan

4.5.21 Use case scenario Logout

Tabel 4.26 Tabel use case scenario Logout

Actor	Admin
Objective	Fungsi untuk keluar dari sistem
Pre-Condition	Aktor memilih menu <i>logout</i>
Main Flow	<ol style="list-style-type: none">1. Aktor memilih menu <i>logout</i> pada menu navigasi2. Sistem akan mematikan <i>session</i> dan pengguna keluar dari sistem
Alternative Flow	-
Post-Condition	Pengguna keluar dari sistem

4.5.22 Use case scenario Login

Tabel 4.27 Tabel use case scenario Login

Actor	Admin
Objective	Fungsi untuk masuk ke dalam sistem
Pre-Condition	Aktor berada dalam halaman <i>login</i>
Main Flow	<ol style="list-style-type: none">1. Aktor memasukan <i>username</i> dan <i>password</i> dan menekan tombol <i>login</i>2. Sistem akan melakukan pengecekan data3. Jika data sesuai maka sistem akan menampilkan <i>dashboard</i> Sistem
Alternative Flow	3.1 Jika data yang dimasukkan salah maka akan menampilkan pesan, "Error! Username atau password salah"
Post-Condition	Pengguna masuk ke dalam sistem

4.5.23 Use case scenario Menambahkan user

Tabel 4.28 Tabel use case scenario Tambah user

Actor	Tamu
Objective	Fungsi untuk menambah <i>user</i> atau admin baru
Pre-Condition	Aktor dalam halaman <i>login</i>
Main Flow	<ol style="list-style-type: none"> 1. Aktor masuk halaman <i>login</i> 2. Aktor memilih tombol register 3. Aktor mengisi form registrasi yang meliputi nama, <i>username, password</i> 4. Aktor menekan tombol simpan
Alternative Flow	<ol style="list-style-type: none"> 3.1 Jika data tidak berhasil disimpan maka akan muncul notifikasi “<i>This is a required field</i>” yang menandakan form harus diisi lengkap 3.2 <i>Password</i> harus diisi minimal 4 karakter dan maksimal 30 karakter
Post-Condition	Sistem menampilkan halaman <i>login</i>

4.5.24 Use case scenario Melihat Pengiriman

Tabel 4.29 Tabel use case scenario Melihat data Pengiriman

Actor	Admin
Objective	Fungsi untuk mengakses fitur list pengiriman
Pre-Condition	Aktor telah masuk ke dalam sistem informasi penjualan
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih fitur Pengiriman 2. Sistem menampilkan <i>dropdown</i> menu pada fitur penawaran dan memilih fitur list pengiriman
Alternative Flow	-
Post-Condition	Sistem menampilkan data pada fitur halaman list pengiriman

4.5.25 Use case scenario Melihat Supplier

Tabel 4.30 Tabel use case scenario Mengelola data Supplier

Actor	Admin
Objective	Fungsi untuk mengakses fitur list supplier
Pre-Condition	Aktor telah masuk ke dalam sistem informasi penjualan
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih fitur supplier 2. Sistem menampilkan <i>dropdown</i> menu pada fitur supplier dan memilih fitur list supplier
Alternative Flow	-
Post-Condition	Sistem menampilkan data pada fitur halaman list supplier

4.5.26 Use case scenario Melihat Pelanggan

Tabel 4.31 Tabel use case scenario Melihat Pelanggan

Actor	Admin
Objective	Fungsi untuk mengakses fitur list pelanggan
Pre-Condition	Aktor telah masuk ke dalam sistem informasi penjualan
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih fitur Pembelian 2. Sistem menampilkan <i>dropdown</i> menu pada fitur pelanggan dan memilih fitur list pelanggan
Alternative Flow	-
Post-Condition	Sistem menampilkan data pada fitur halaman list pelanggan

4.5.27 Use case scenario Melihat Laporan Pengiriman

Tabel 4.32 Tabel use case scenario laporan pengiriman

Actor	Admin
Objective	Fungsi untuk mengakses melihat laporan pengiriman
Pre-Condition	Aktor telah masuk ke dalam sistem informasi penjualan
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih fitur pengiriman 2. Sistem menampilkan list pengiriman 3. Aktor menekan tombol tampil laporan
Alternative Flow	-
Post-Condition	Sistem menampilkan data pada laporan pengiriman

4.5.28 Use case scenario Melihat Laporan Pembelian

Tabel 4.33 Tabel *use case scenario* laporan pembelian

Actor	Admin
Objective	Fungsi untuk mengakses melihat laporan pembelian
Pre-Condition	Aktor telah masuk ke dalam sistem informasi penjualan
Main Flow	<ol style="list-style-type: none">1. Aktor memilih fitur pembelian2. Sistem menampilkan list pembelian3. Aktor menekan tombol tampil laporan
Alternative Flow	-
Post-Condition	Sistem menampilkan data pada laporan pembelian

4.5.29 Use case scenario Melihat Laporan Order Masuk

Tabel 4.34 Tabel *use case scenario* laporan *order* masuk

Actor	Admin
Objective	Fungsi untuk mengakses melihat laporan <i>order</i> masuk
Pre-Condition	Aktor telah masuk ke dalam sistem informasi penjualan
Main Flow	<ol style="list-style-type: none">1. Aktor memilih fitur <i>order</i> masuk2. Sistem menampilkan list <i>order</i> masuk3. Aktor menekan tombol tampil laporan
Alternative Flow	-
Post-Condition	Sistem menampilkan data pada laporan <i>order</i> masuk

4.5.30 Use case scenario Menambahkan Stok Barang

Tabel 4.35 Tabel *use case scenario* menambahkan stok barang

Actor	Admin
Objective	Fungsi untuk menambah data stok barang
Pre-Condition	Aktor telah masuk ke dalam sistem informasi penjualan
Main Flow	<ol style="list-style-type: none">1. Aktor memilih fitur Stok gudang2. Aktor memilih menu pada <i>drop-down</i> input pembelian barang3. Sistem menampilkan <i>form</i> input pembelian barang

Tabel 4.35 Tabel use case scenario menambahkan stok barang (lanjutan)

	<ol style="list-style-type: none">4. Aktor mengisi <i>form</i> yang meliputi nama <i>supplier</i>, nama barang, harga satuan, <i>quantity</i>, harga total, keterangan, tanggal pembelian.5. Aktor menekan tombol simpan untuk menyimpan data pengiriman
Alternative Flow	2.1. Jika data tidak berhasil disimpan maka akan muncul notifikasi " <i>This is a required field</i> " yang menandakan <i>form</i> harus diisi lengkap
Post-Condition	Sistem menampilkan form menambahkan stok barang

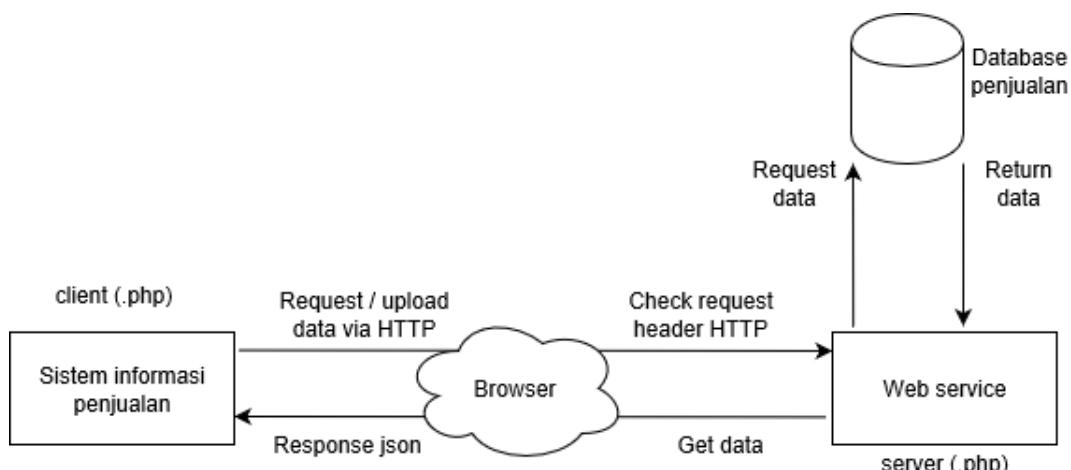
BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan

Tahap selanjutnya yaitu tahap analisis kebutuhan, tahap ini menjelaskan tentang apa saja perancangan yang dilakukan sebelum tahap implementasi pada pembangunan sistem informasi manajemen penjualan CV. Mitra Teknik berbasis web. Perancangan yang akan dilakukan diantaranya meliputi perancangan arsitektur sistem, pemodelan *class diagram*, pemodelan *sequence diagram*, *entity relation diagram*, perancangan pseudocode dan perancangan antarmuka. Pembahasan tahap ini akan dijelaskan pada sub-bab perancangan sistem diantaranya sebagai berikut.

5.1.1 Arsitektur Web Service

Pada bagian ini akan menjelaskan bagaimana komunikasi yang dilakukan oleh sistem informasi penjualan sesuai dengan arsitektur *REST web service*. Arsitektur sistem yang digunakan dapat dilihat pada gambar 5.1 berikut.



Gambar 5.1 Arsitektur Sistem Informasi Penjualan

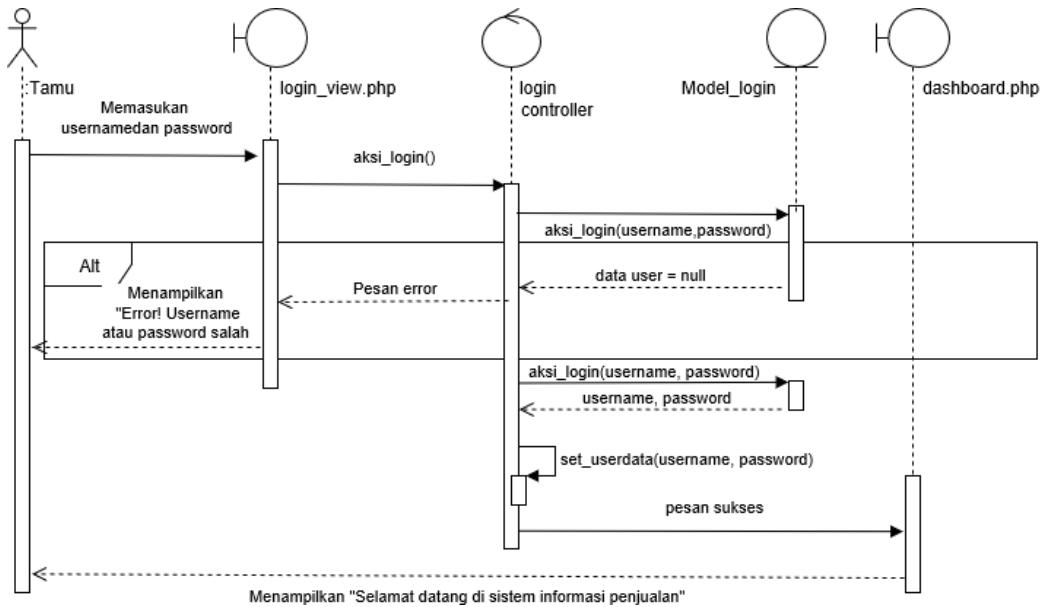
Pada bagian ini akan menggambarkan rangkaian yang telah dirancang dan diimplementasikan dengan benar. Pada gambar 5.1 implementasi *client* menggunakan bahasa pemrograman *web*, kemudian ketika mengirim *request* pada *server* menggunakan *library CURL*. *CURL* digunakan untuk memindai data dari atau ke sebuah *server* tanpa interaksi dari masing-masing *user*. Setelah sampai pada *server*, kemudian akses diperiksa dengan autentikasi dan *key*, jika tidak sesuai maka akses akan ditolak. Ketika akses berhasil maka akan melakukan akses *database* dan mengirim kembali melalui *response json*.

5.1.2 Pemodelan Sequence Diagram

Sequence diagram merupakan diagram yang menggambarkan hubungan antar objek dan alur kerja pada setiap fungsi sistem. *Sequence diagram* dapat dibuat berdasarkan *use case scenario* dan hasil pemodelan

class diagram yang telah dideskripsikan pada tahap sebelumnya, berikut hanya ada lima sampel *sequence diagram* yang ditampilkan yaitu *sequence diagram Login*, Tambah *Order Masuk*, Edit *Order Masuk*, Hapus *Order Masuk*, dan Tambah Pembelian Barang.

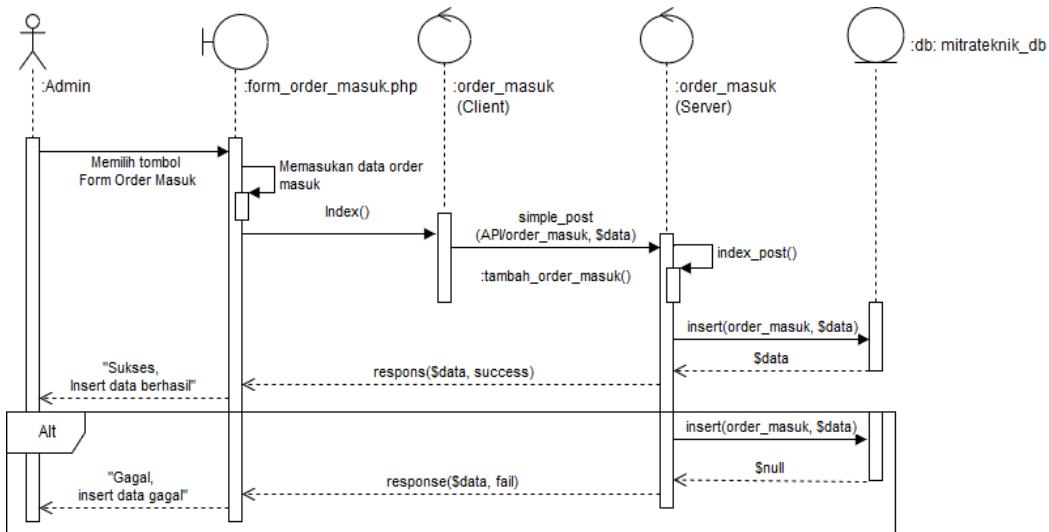
1. Sequence Diagram Login



Gambar 5.2 Sequence diagram Login

Pada gambar 5.2 menggambarkan interaksi antar objek yang saling berhubungan dalam fungsi *login*. Objek yang terlibat di dalamnya antara lain adalah aktor Tamu, *boundary login_view* dan *dashboard*, *control login* dan *Entity Model_login*. Tamu sebagai aktor tanpa otoritas akan melakukan *login* sebelum dapat akses sistem informasi penjualan dengan memasukkan data *username* dan *password*, kemudian *control login* menjalankan proses *login* dengan *method aksi_login()*. Kemudian data yang telah dimasukkan akan diperiksa dalam *entity Model_login* dengan menggunakan *method cek_login()*. Ketika data tidak ditemukan atau bernilai nol maka akan mengembalikan pesan *error*, namun ketika data ditemukan maka data akan dimasukkan ke dalam *session* dengan *method set_userdata (username,password)*. Setelah berhasil *login* maka akan diteruskan ke halaman *dashboard* sistem informasi penjualan.

2. Sequence Diagram Menambahkan Order Masuk

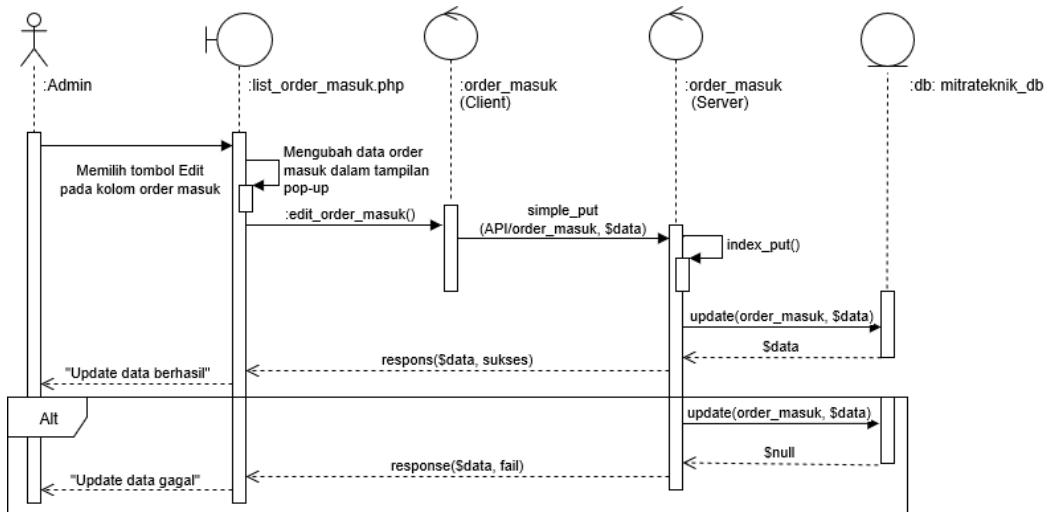


Gambar 5.3 Sequence diagram Menambahkan Order Masuk

Pada gambar 5.3 menggambarkan beberapa objek yang berinteraksi diantaranya adalah aktor admin, *boundary list_order_masuk*, *control order_masuk* pada *client* dan *server* kemudian *database* mitrateknik_db. Pertama aktor memilih menu penjualan kemudian memilih menu *order* masuk pada sub menu dan memilih tombol Tambah *Order* Masuk. Kemudian aktor memasukkan data pada *form pop-up*. Setelah mengisi *form* data dimasukkan ke dalam *array* dan dikirim ke *server* melalui *method simple_post* menuju *control order_masuk* pada *server*. Kemudian data yang masuk dikirim ke *database* mitrateknik_db. Jika berhasil menyimpan data maka akan menampilkan pesan *success* dan jika gagal menampilkan pesan *error*.

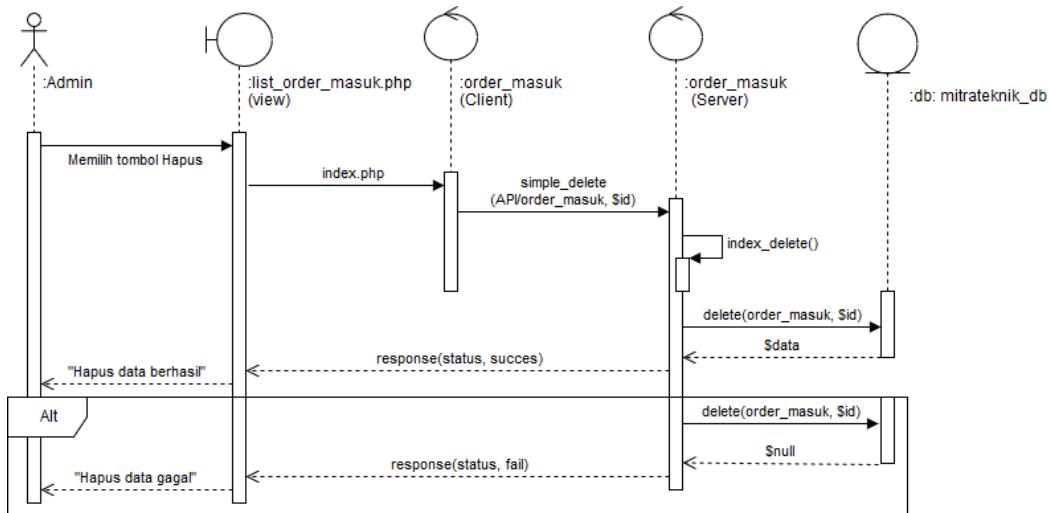
3. Sequence Diagram Mengubah Order Masuk

Pada gambar 5.4 menggambarkan beberapa objek yang berinteraksi diantaranya adalah aktor admin, *boundary list_order_masuk*, *control order_masuk* pada *client* dan *server* kemudian *database* mitrateknik_db. Pertama aktor memilih tombol *Edit* pada kolom di halaman *Order* Masuk. Kemudian aktor mengubah data pada data yang ditampilkan oleh *form pop-up*. Setelah mengisi *form* data dimasukkan ke dalam *array* dan dikirim ke *server* melalui *method simple_put* menuju *control order_masuk* pada *server*. Kemudian data yang masuk dikirim ke *database* mitrateknik_db. Jika berhasil menyimpan data maka akan menampilkan pesan sukses dan jika gagal menampilkan pesan *error*.



Gambar 5.4 Sequence diagram Mengubah Order Masuk

4. Sequence Diagram Hapus Order Masuk



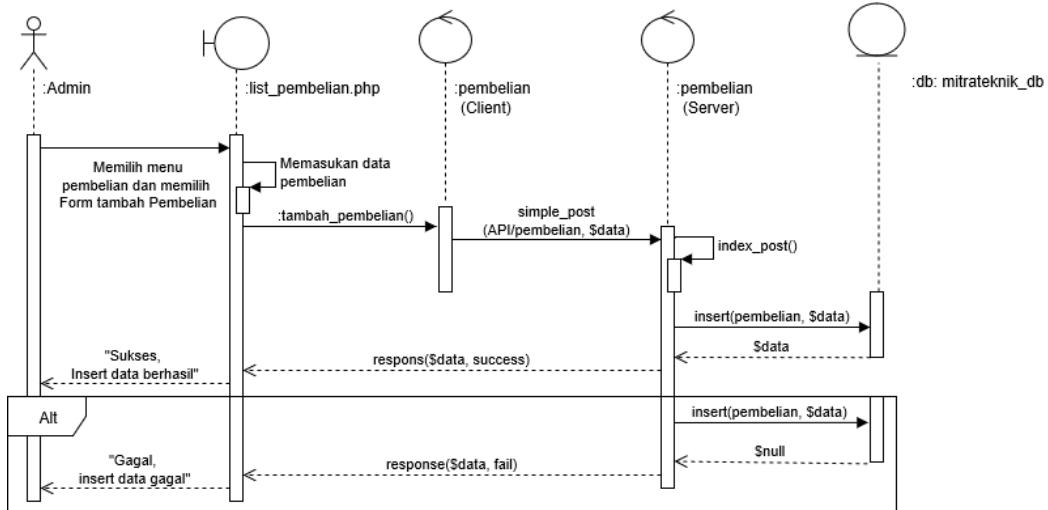
Gambar 5.5 Sequence diagram Menghapus Order Masuk

Pada gambar 5.5 menggambarkan beberapa objek yang saling berinteraksi diantaranya adalah aktor admin, boundary list_order_masuk, control order_masuk pada client dan pada server, dan database mitrateknik_db. Pertama aktor menekan tombol Hapus pada kolom order_masuk. Kemudian data yang akan dihapus dipilih berdasarkan *id* pada di control order_masuk pada client. *Id* order_masuk yang akan dihapus dikirim menuju control order_masuk pada server. Terakhir data dihapus pada database mitrateknik_db berdasarkan *id* yang diterima server.

5. Sequence Diagram Menambahkan Pembelian Barang

Pada gambar 5.6 di bawah menggambarkan beberapa objek yang berinteraksi diantaranya adalah aktor admin, boundary pembelian, control pembelian pada client dan server kemudian database mitrateknik_db. Pertama aktor memilih menu Pembelian dan memilih tombol Tambah Pembelian.

Kemudian aktor memasukkan data pada *form pop-up*. Setelah mengisi *form* data dimasukkan ke dalam *array* dan dikirim ke *server* melalui *method simple_post* menuju *control pembelian* pada *server*. Kemudian data yang masuk dikirim ke *database mitrateknik_db*. Jika berhasil menyimpan data maka akan menampilkan pesan *success* dan jika gagal menampilkan pesan *error*.



Gambar 5.6 Sequence diagram Menambahakan Pembelian Barang

5.1.3 Pemodelan *Class Diagram*

Pemodelan *class diagram* dapat digunakan dalam penerapan *Object Oriented Programming (OOP)*. Pada pemodelan implementasi sistem informasi penjualan memiliki beberapa *class* objek diantaranya *controller*, *model* dan *view*. Berikut contoh detail *class*:

Detail dari *class* gudang dapat dilihat pada tabel 5.1 dan 5.2 berikut.

Tabel 5.1 Class Gudang (Server)

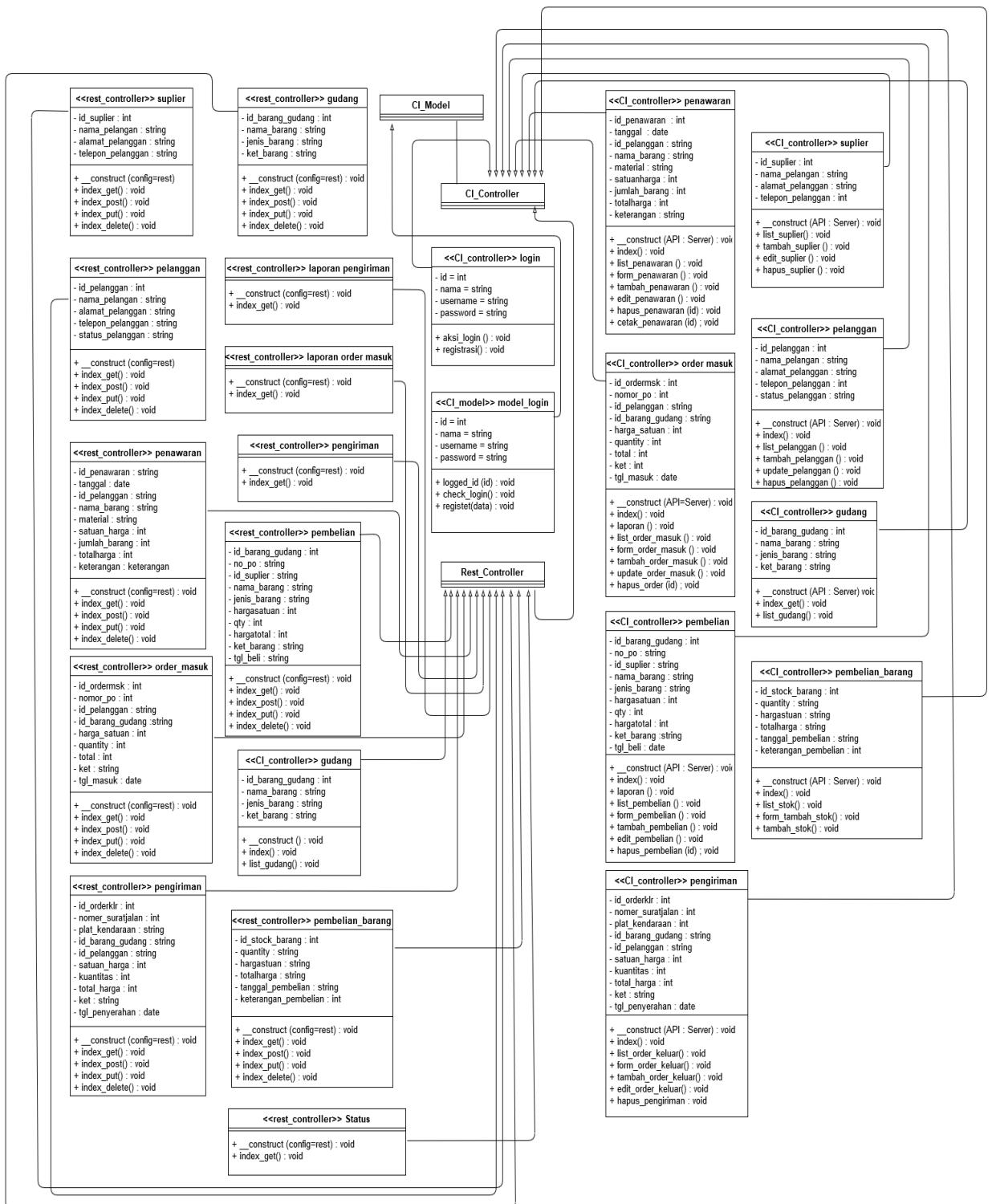
Nama Operasi	Visibility	Tipe
__construct(\$config = 'REST')	Public	Void
index_get()	Public	Void
index_post()	Public	Void
index_put()	Public	Void
index_delete()	Public	Void

Tabel 5.2 Class Gudang (Client)

Nama Operasi	Visibility	Tipe
__construct, \$this->API=..../Server/index.php	Public	Void
List_gudang()	Public	Void
Tambah_barang()	Public	Void
Edit_barang_gudang()	Public	Void
Hapus_barang()	Public	Void

Pada tabel 5.1 dan 5.2 di atas merupakan fungsi yang terdapat pada *class* gudang *client* dan *server*. *Class* gudang merupakan objek untuk mengolah data barang yang berisi rincian lengkap data barang berasal dari *class* gudang *client*. Objek gudang menurunkan sifat induknya yaitu *REST_Controller* untuk memanggil fungsi yang digunakan dalam mengolah data sehingga dapat menjalankan fungsi seperti *GET*, *POST*, *PUT*, dan *DELETE* dari *libraries REST*. Sedangkan objek gudang disisi *client* mengirim data ke *server* dan diolah oleh *class* gudang *server* kemudain disimpan dalam *database*. Objek lain yang menyerupai dengan fungsi dari *class* gudang yaitu *Order* keluar, *Order* masuk, Pelanggan, Pembelian, Penawaran, dan *Supplier*. Meskipun fungsi operasi menyerupai, tetapi memiliki data yang berbeda. Pada objek tertentu memiliki hubungan agregasi dengan objek lain dikarenakan terdapat atribut yang harus dimiliki oleh objek lain ketika objek tersebut akan dibuat.

Pada tahap selanjutnya yaitu menggambarkan pemodelan *class diagram* pada sistem informasi penjualan. Pemodelan *class diagram* kali ini mengacu pada arsitektur program yakni *framework MVC*. Untuk *class Controller* pada *client* berfungsi untuk menginstansiasi model, memanggil *view* dan *class REST controller* pada sisi *server*, kemudian untuk *class view* berisi fungsi-fungsi untuk mengakses tampilan *view* pada sistem. Sedangkan untuk menjalankan *query Insert, Update, Delete, Select*, dioperasikan dalam *class REST server* yang nantinya dipanggil dalam *construct* pada setiap *class controller* disisi *client*. Selengkapnya dapat dilihat pada gambar 5.7.

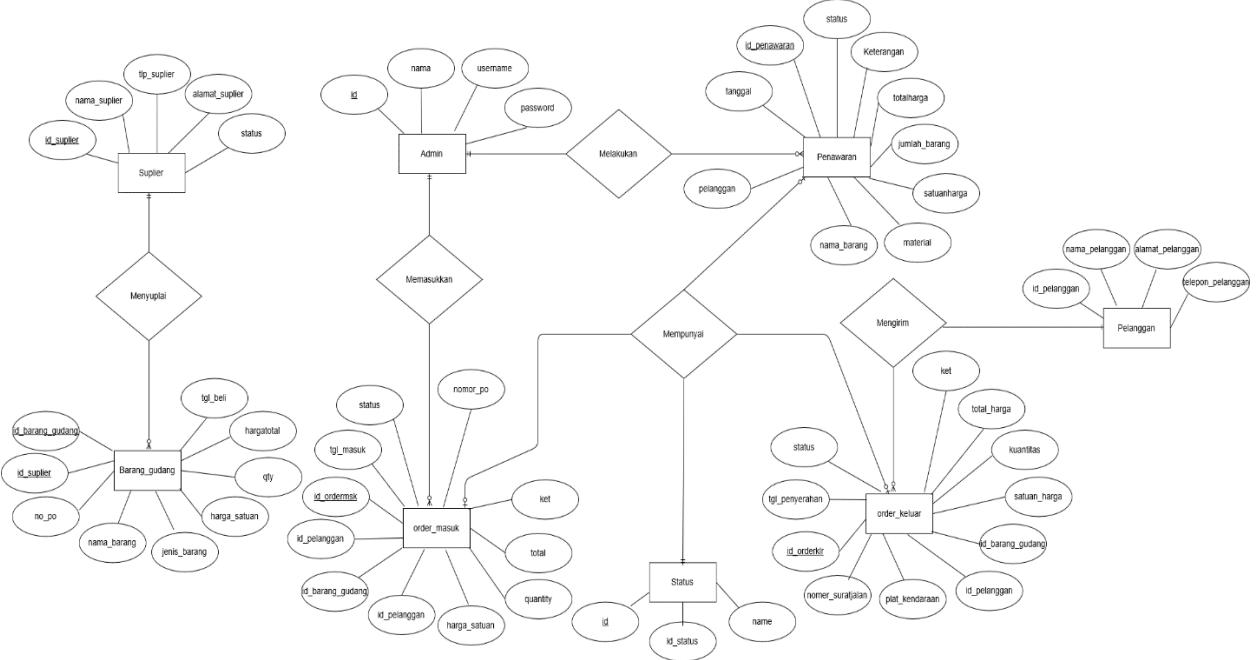


Gambar 5.7 Perancangan *class diagram* sistem informasi penjualan

5.1.4 Entity Relation Diagram

Entity Relation Diagram (ERD) digunakan untuk menggambarkan hubungan antara entitas yang ada pada sistem dan akan digunakan sebagai dasar dalam implementasi basis data. *Database* yang digunakan adalah *database*

mitrateknik_db yang berada pada *server*. Perancangan *ERD* menggunakan tipe *logical data model* yang dapat dilihat pada gambar 5.8 berikut.



Gambar 5.8 Entity Relationship Diagram (ERD)

Pada gambar 5.8 di atas menggambarkan hubungan antara entitas pada sistem penjualan, *Database* mitrateknik_db memiliki beberapa entitas yaitu admin, status, pembelian barang, *order_masuk*, *order_keluar*, penawaran, barang gudang, pelanggan, dan suplier. Entitas admin memiliki hubungan *one-to-one* dengan entitas barang gudang. Entitas barang gudang juga memiliki hubungan *one-to-one* dengan entitas *order_keluar*, dan beberapa entitas juga sama memiliki hubungan *one-to-one* dengan entitas yang lainnya.

5.1.5 Perancangan Pseudocode

Tahap selanjutnya yaitu menjelaskan pseudocode setiap operasi pada masing-masing *class*. Pseudocode pada setiap operasi menjelaskan bagaimana langkah-langkah logika untuk menjalankan sebuah fungsi untuk menghasilkan keluaran. Dalam perancangan pseudocode direpresentasikan dalam bentuk *pseudocode*. Akan tetapi hanya ada lima *class* yang dijelaskan sebagai sampel yaitu operasi *aksi_login()* pada *class LoginController*, *tambah_order_masuk* pada *class Order_masukController*, *index_post()* pada *class Order_masukController*, *index_put()* pada *class PembelianController*, dan *Edit_pembelian()* pada *class PembelianController*. Masing-masing pseudocode pada setiap *class* dapat dijelaskan sebagai berikut.

1. Pseudocode fungsi *aksi_login()* kelas *Login CI_Controller*

Penjelasan pseudocode fungsi *aksi_login()* dapat dilihat pada tabel 5.3 berikut.

Tabel 5.3 Pseudocode *aksi_login()*

1	Function <i>aksi_login()</i> start
2	
3	Form validation required, username
4	Form validation required, password
5	Form validation alert, message
6	If Form validation==TRUE then
7	
8	username = input POST (username)
9	password = input POST (password)
10	
11	cek = model_login-> <i>aksi_login</i> (username, password)
12	if (checking != FALSE) then
13	checking = apps->row_array then
14	session_data = array then
15	id=>apps->id
16	nama=>apps->nama
17	username=>apps->username
18	password=>apps->password
19	session->set_userdata(session_data)
20	Echo base_url(dashboard)
21	Else
22	Print "Erorr, password atau username salah!"
23	Else
24	load->view(fornt_end/login_view)
25	End if
26	End if

Pada tabel 5.3 merupakan pseudocode ketika sistem menjalankan proses *login* atau melakukan pemeriksaan otoritas pengguna ketika akan masuk ke dalam sistem. Fungsi berjalan ketika pengguna menekan tombol *login*, fungsi akan melakukan cek data masukan *username* dan *password* dengan melakukan inisialisasi variabel *cek* dan mengirim data ke *model* untuk melakukan pemeriksaan pada *database*. Jika data tidak bernilai nol maka data dimasukkan ke data *session* dan dapat melanjutkan ke dalam sistem dan menampilkan pesan sukses.

2. Pseudocode fungsi *tambah_order_masuk()* class *Order_masuk CI_Controller*

Penjelasan pseudocode fungsi *tambah_order_masuk()* dapat dilihat pada tabel 5.4 berikut.

Tabel 5.4 Pseudocode Tambah_order_masuk CI_Controller

1	Function tambah_order_masuk() start
2	data=array(
3	id_ordermsk = input post (no_ordermsk),
4	nomor_po = input post (nomor_po),
5	id_pelanggan = input post (id_pelanggan),
6	harga_satuan = input post (harga_satuan),
7	quantity = input post (quantity),
8	total = input post (total),
9	ket = input post (ket),
10	tgl_masuk = input post (tgl_masuk))
11	insert = curl simple post(API./order_masuk, data)
12	if(insert) then
13	session setflashdata(sukses,insert data sukses)
14	else
15	session setflashdata(gagal,insert data gagal)
16	end if
17	redirect(order_masuk)
18	end if
19	
20	end

Pada tabel 5.4 merupakan pseudocode ketika sistem menjalankan proses tambah *order* masuk pada *class order* masuk. Ketika pengguna telah selesai memasukkan data pada *form* tambah *order* masuk maka proses ini akan dijalankan. Data yang telah dimasukkan disimpan pada variabel data, kemudian variabel baru dengan nama *insert* dibuat untuk mengirim data ke *service*. Jika variabel *insert* berhasil mengirim data dan menerima *response success* maka akan menampilkan pesan sukses, jika mendapat *response error* maka akan menampilkan pesan gagal.

3. Pseudocode fungsi *index_post()* class Order_masuk RESTController

Penjelasan pseudocode fungsi *index_post()* dapat dilihat pada tabel 5.5 berikut.

Tabel 5.5 Pseudocode *index_post()* class Order_masuk RESTController

1	Function index_post() start
2	data=array(
3	id_ordermsk = post (no_ordermsk),
4	nomor_po = post (nomor_po),
5	id_pelanggan = post (id_pelanggan),
6	harga_satuan = post (harga_satuan),
7	quantity = post (quantity),
8	total = post (total),
9	ket = post (ket),
10	tgl_masuk = post (tgl_masuk))
11	insert = db insert (order_masuk, data)
12	if(insert) then
13	response(data, 200)
14	else
15	response(array(status = fail, 502)) end if

Pada tabel 5.5 merupakan pseudocode dari proses *post* pada *Web Service* ketika melakukan tambah *order* masuk. Pada saat *client* mengirim data dengan memanggil *method post* maka fungsi ini akan dijalankan. Pada inisialisasi variabel data *service*, data dari *client* dimasukkan dalam bentuk *array*. Kemudian data dari klien yang telah dikirim dimasukkan ke dalam variabel data. Setelah mengecek *id* pada *database* maka menjalankan variabel *insert*, jika berhasil akan mengirim *response* berupa data yang berhasil diperbarui namun jika gagal akan mengirim *response* pesan status *fail*.

4. Pseudocode fungsi *index_put()* class *Order_masuk RESTController*

Pada tabel 5.6 merupakan pseudocode dari proses *put* pada *Web Service* ketika melakukan tambah *order* masuk. Pada saat *client* mengirim data dengan memanggil *method put* maka fungsi ini akan dijalankan. Pada inisialisasi variabel data *service*, data dari *client* dimasukkan dalam bentuk *array*. Kemudian data dari klien yang telah dikirim dimasukkan ke dalam variabel data. Setelah mengecek *id* dan status pada *database* kemudian data dimasukkan menyesuaikan kondisi status, jika berhasil akan mengirim *response* berupa data yang berhasil diperbarui, namun jika gagal akan mengirim *response* pesan status *fail*.

Penjelasan pseudocode fungsi *index_put()* dapat dilihat pada tabel 5.6 berikut

Tabel 5.6 Pseudocode *index_put()* class *Order_masuk RESTController*

1	Function index_put() start
2	id = PUT (id_ordermsk)
3	stat = PUT (status)
4	If (stat == 3) then
5	data1=array(
6	id_ordermsk = put (no_ordermsk),
7	nomor_po = put (nomor_po),
8	id_pelanggan = put (id_pelanggan),
9	harga_satuan = put (harga_satuan),
10	quantity = put (quantity),
11	total = put (total),
12	ket = put (ket),
13	tgl_masuk = put (tgl_masuk),
14	status = put (status))
15	db where (id_ordermsk, id)
16	update = db update (order_masuk, data)
17	response (data1, 200)
18	else if (stat = 4) then
19	data2=array(
20	id_ordermsk = put (no_ordermsk),
21	nomor_po = put (nomor_po),
22	id_pelanggan = put (id_pelanggan),
23	harga_satuan = put (harga_satuan),
24	quantity = put (quantity),
25	total = put (total),
26	ket = put (ket),

Tabel 5.6 Pseudocode *index_put()* class Order_masuk RESTController (lanjutan)

27	tgl_masuk = put (tgl_masuk), status = put (status))
28	data3 = array(nomer_suratjalan = 0, plat_kendaraan = belum ada, id_barang_gudang = put (id_barang_gudang), id_pelanggan = put (id_pelanggan), satuan_harga = 0, kuantitas = put (quantity), total_harga = 0, ket = put (ket), tgl_penyerahan = put (tgl_masuk), status = put (status))
40	db where (id_ordermsk, id) update = db update (order_masuk, data2) m = insertOrderkeluar (data3) response (data2, 200)
44	else response(array(status = fail, 502))
46	end if
47	end if
48	end

5. Pseudocode fungsi *edit_pembelian()* class Pembelian Ci_Controller

Penjelasan pseudocode fungsi *edit_pembelian()* dapat dilihat pada tabel 5.7.

Tabel 5.7 Pseudocode *edit_pembelian()* class Pembelian Ci_Controller

1	Function edit_pembelian() start
2	data=array(id_barang_gudang = input post (id_barang_gudang), no_po = input post (no_po), id_suplier = input post (id_suplier), nama_barang = input post (nama_barang), jenis_barang = input post (jenis_barang), harga_satuan = input post (harga_satuan), qty = input post (qty), harga_total = input post (harga_total), ket_barang = input post (ket_barang), tgl_beli = input post (tgl_beli), status_barang = input post (status_barang))
14	update = curl simple put(API./pembelian, data)
15	if(update) then
16	session set flashdata(sukses,insert data sukses)
17	else
18	session set flashdata(gagal,insert data gagal)

Tabel 5.7 Pseudocode *edit_pembelian()* class Pembelian Ci_Controller (lanjutan)

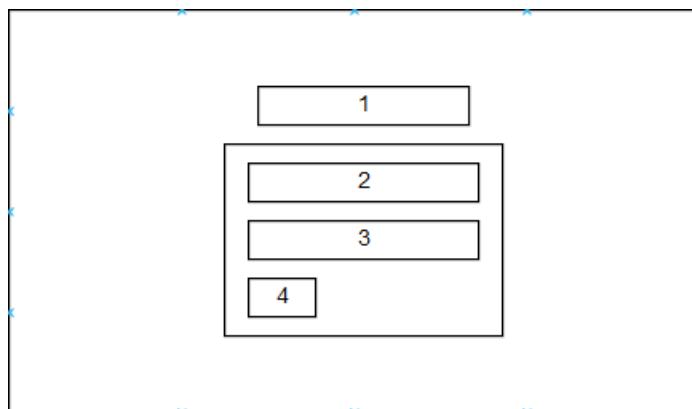
19	end if
20	redirect(pembelian)
21	end if
22	end

Pada tabel 5.7 merupakan pseudocode ketika sistem menjalankan proses *edit order* masuk pada *class order* masuk. Ketika pengguna telah selesai memasukkan data pada *form edit order* masuk maka proses ini akan dijalankan. Data yang telah dimasukkan disimpan pada variabel data, kemudian variabel baru dengan nama *update* dibuat untuk mengirim data ke *service*. Jika variabel *update* berhasil mengirim data dan menerima *response success* maka akan menampilkan pesan sukses, jika mendapat *response error* maka akan menampilkan pesan gagal.

5.1.6 Perancangan Antarmuka

Perancangan antarmuka merupakan perancangan tampilan *web* pada sistem yang akan dibuat. Perancangan antarmuka pada umumnya hanya dinotasikan dengan tulisan dan gambar dua dimensi seperti persegi maupun persegi panjang. Berikut adalah sebagian dari rancangan antarmuka sistem informasi penjualan.

1. Antarmuka *Login*



Gambar 5.9 Antarmuka *login*

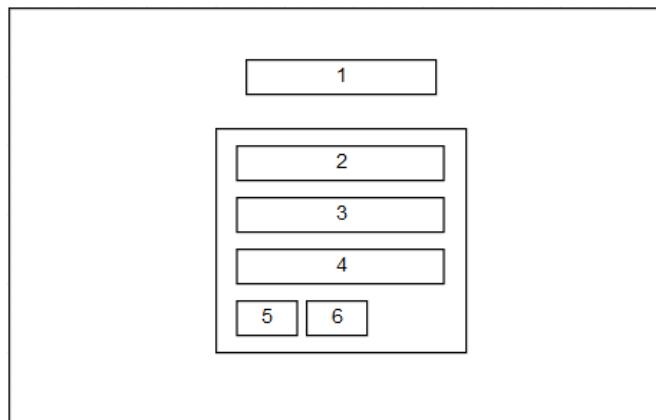
Pada gambar 5.9 adalah antarmuka *login*, dalam gambar antarmuka terdapat angka yang memiliki keterangan sebagai berikut:

- Angka 1: Nama perusahaan CV. Mitra Teknik
- Angka 2: *Field* untuk memasukkan *username*
- Angka 3: *Field* untuk memasukkan *password*
- Angka 4: Merupakan tombol *sign-in*

2. Antarmuka Menambahkan user

Pada gambar 5.10 adalah gambar antarmuka menambahkan user, dalam gambar antarmuka registrasi ini terdapat nomor dan keterangan sebagai berikut:

- Angka 1: Nama perusahaan CV. Mitra Teknik
- Angka 2: *Field* untuk memasukkan nama
- Angka 3: *Field* untuk memasukkan *username*
- Angka 4: *Field* untuk memasukkan password
- Angka 5: Merupakan tombol register
- Angka 6: Merupakan tombol *back*

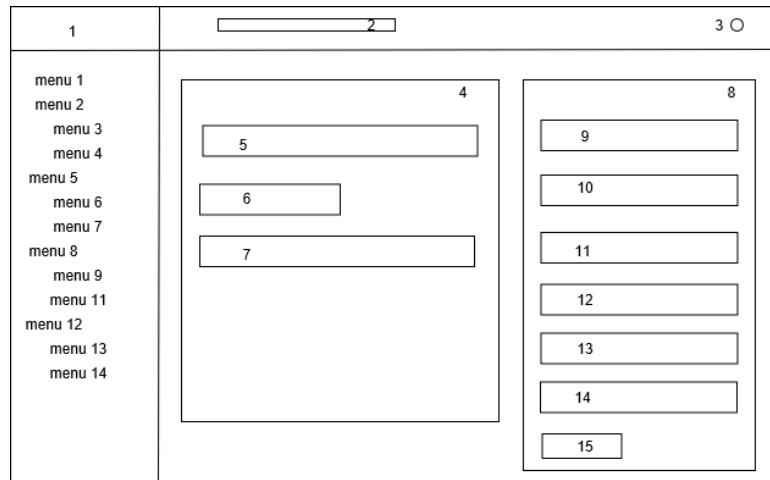


Gambar 5.10 Antarmuka menambahkan user

3. Antarmuka Menambahkan penawaran

Pada gambar 5.11 adalah gambar antarmuka menambahkan penawaran, dalam gambar antarmuka terdapat nomor dengan keterangan sebagai berikut:

- Angka 1: Nama perusahaan CV. Mitra Teknik
- Angka 2: Kolom pencarian
- Angka 3: Tombol navigasi admin
- Menu 1-14: Merupakan menu *sidebar* sistem
- Angka 4: *Form* penawaran
- Angka 5: *Field* untuk memasukkan nama perusahaan
- Angka 6: *Field* untuk memasukkan tanggal
- Angka 7: *Field* untuk memasukkan keterangan barang
- Angka 8: *Form* penawaran
- Angka 9: *Field* untuk memasukkan nama barang
- Angka 10: *Field* untuk memasukkan material
- Angka 11: *Field* untuk memasukkan harga satuan
- Angka 12: *Field* untuk memasukkan *quantity*
- Angka 13: *Field* untuk menampilkan harga total
- Angka 15: Merupakan tombol simpan



Gambar 5.11 Antarmuka menambahkan penawaran

4. Antarmuka List Penawaran

Pada gambar 5.12 adalah gambar antarmuka list penawaran, dalam gambar antarmuka terdapat nomor dengan keterangan sebagai berikut:

Angka 1: Nama perusahaan CV. Mitra Teknik

Angka 2: Kolom pencarian

Angka 3: Nama tabel

Angka 4: Tombol navigasi admin

Menu 1-7: Merupakan menu *sidebar* sistem

Angka 5: Kolom untuk menampilkan nomor urut

Angka 6: Kolom untuk menampilkan data tanggal

Angka 7: Kolom untuk menampilkan data nama pelanggan

Angka 8: Kolom untuk menampilkan data nama

Angka 9: Kolom untuk menampilkan data harga

Angka 10: Kolom untuk menampilkan data keterangan

Angka 11: Kolom aksi berisi tombol cetak, *edit*, hapus

Menu 1		3					
Menu 2							
Menu 3							
Sub Menu							
Sub Menu							
Menu 4	5	6	7	8	9	10	11
Menu 5							
Menu 6							
Menu 7							

Gambar 5.12 Antarmuka list penawaran

5. Antarmuka Menambahkan *Order* Masuk

1	2	3 O
menu 1 menu 2 menu 3 menu 4 menu 5 menu 6 menu 7 menu 8 menu 9 menu 11 menu 12 menu 13 menu 14	4 5 6 7 8	9 10 11 12 13 14

Gambar 5.13 Antarmuka menambahkan *order* masuk

Pada gambar 5.13 adalah gambar antarmuka menambahkan *order* masuk, dalam gambar antarmuka terdapat nomor dengan keterangan sebagai berikut:

- Angka 1: Nama perusahaan CV. Mitra Teknik
- Angka 2: Kolom pencarian
- Angka 3: Tombol navigasi admin
- Menu 1-14: Merupakan menu *sidebar* sistem
- Angka 4: *Form* order masuk
- Angka 5: *Field* untuk memasukkan no purchase order
- Angka 6: *Field* untuk memasukkan nama perusahaan
- Angka 7: *Field* untuk memasukkan tanggal
- Angka 8: *Field* untuk memasukkan keterangan barang
- Angka 9: *Form* order masuk
- Angka 10: *Field* untuk memasukkan nama barang
- Angka 11: *Field* untuk memasukkan harga satuan
- Angka 12: *Field* untuk memasukkan *quantity*
- Angka 13: *Field* untuk menampilkan harga total
- Angka 14: Merupakan tombol simpan

6. Antarmuka List Order Masuk

1	2	3 O
menu 1 menu 2 menu 3 menu 4 menu 5 menu 6 menu 7 menu 8 menu 9 menu 11 menu 12 menu 13 menu 14	4	5 6 7 8 9 10

Gambar 5.14 Antarmuka list order masuk

Pada gambar 5.14 adalah gambar antarmuka list order masuk, dalam gambar antarmuka terdapat nomor dengan keterangan sebagai berikut:

- Angka 1: Nama perusahaan CV. Mitra Teknik
- Angka 2: Kolom pencarian
- Angka 3: Tombol navigasi admin
- Angka 4: Tombol tampil laporan
- Menu 1-14: Merupakan menu *sidebar* sistem
- Angka 5: Kolom untuk menampilkan nomor urut
- Angka 6: Kolom untuk menampilkan nomor po
- Angka 7: Kolom untuk menampilkan data nama pelanggan
- Angka 8: Kolom untuk menampilkan data nama barang
- Angka 9: Kolom untuk menampilkan data tanggal masuk
- Angka 10: Kolom aksi berisi tombol detail, *edit*, *hapus*

7. Antarmuka Menambahkan Pembelian Barang

1	2	3 O
menu 1 menu 2 menu 3 menu 4 menu 5 menu 6 menu 7 menu 8 menu 9 menu 11 menu 12 menu 13 menu 14	4 5 6 7 8	9 10 11 12 13 14 15

Gambar 5.15 Antarmuka menambahkan pembelian barang

Pada gambar 5.15 adalah gambar antarmuka menambahkan pembelian, dalam gambar antarmuka terdapat nomor dengan keterangan sebagai berikut:

- Angka 1: Nama perusahaan CV. Mitra Teknik
- Angka 2: Kolom pencarian
- Angka 3: Tombol navigasi admin
- Menu 1-14: Merupakan menu *sidebar* sistem
- Angka 4: *Form* pembelian
- Angka 5: *Field* untuk memasukkan no purchase order
- Angka 6: *Field* untuk memasukkan nama supplier
- Angka 7: *Field* untuk memasukkan tanggal
- Angka 8: *Field* untuk memasukkan keterangan barang
- Angka 9: *Form* pembelian
- Angka 10: *Field* untuk memasukkan nama barang
- Angka 11: *Field* untuk memasukkan jenis barang
- Angka 12: *Field* untuk memasukkan harga satuan
- Angka 13: *Field* untuk memasukkan *quantity*
- Angka 14: *Field* untuk menampilkan harga total
- Angka 15: Merupakan tombol simpan

8. Antarmuka List Stok Barang

Pada gambar 5.16 adalah gambar antarmuka list stok barang, dalam gambar antarmuka terdapat nomor dengan keterangan sebagai berikut:

- Angka 1: Nama perusahaan CV. Mitra Teknik
- Angka 2: Kolom pencarian
- Angka 3: Nama tabel
- Angka 4: Tombol tampil laporan
- Menu 1-14: Merupakan menu *sidebar* sistem
- Angka 5: Tombol navigasi admin
- Angka 6: Kolom untuk menampilkan nomor urut
- Angka 7: Kolom untuk menampilkan tanggal pembelian
- Angka 8: Kolom untuk menampilkan data nama barang
- Angka 9: Kolom untuk menampilkan data *quantity*
- Angka 10: Kolom untuk menampilkan data harga

1	2	4 O					
Menu 1	<input type="text"/> 3						
Menu 2	<input type="text"/> 5						
Menu 3 Sub Sub							
Menu 4	<table border="1" style="width: 100%;"><tr><td style="width: 20%;">6</td><td style="width: 20%;">7</td><td style="width: 20%;">8</td><td style="width: 20%;">9</td><td style="width: 20%;">10</td></tr></table>	6	7	8	9	10	
6	7	8	9	10			
Menu 5							
Menu 6							
Menu 7							

Gambar 5.16 Antarmuka list stok barang

9. Antarmuka Menambahkan Pengiriman

1	2	3 O																				
menu 1 menu 2 menu 3 menu 4 menu 5 menu 6 menu 7 menu 8 menu 9 menu 11 menu 12 menu 13 menu 14 Menu Menu Menu	<table border="1" style="width: 100%;"><tr><td style="width: 50%;">4</td><td style="width: 50%;">5</td></tr><tr><td>6</td><td></td></tr><tr><td>7</td><td></td></tr><tr><td>9</td><td></td></tr><tr><td>10</td><td></td></tr></table>	4	5	6		7		9		10		<table border="1" style="width: 100%;"><tr><td style="width: 50%;">11</td><td style="width: 50%;">12</td></tr><tr><td>13</td><td></td></tr><tr><td>14</td><td></td></tr><tr><td>15</td><td></td></tr><tr><td>16</td><td></td></tr></table>	11	12	13		14		15		16	
4	5																					
6																						
7																						
9																						
10																						
11	12																					
13																						
14																						
15																						
16																						

Gambar 5.17 Antarmuka menambahkan pengiriman

Pada gambar 5.17 adalah gambar antarmuka menambahkan pengiriman, dalam gambar antarmuka terdapat nomor dengan keterangan sebagai berikut:

- Angka 1: Nama perusahaan CV. Mitra Teknik
- Angka 2: Kolom pencarian
- Angka 3: Tombol navigasi admin
- Menu 1-14: Merupakan menu *sidebar* sistem
- Angka 4: *Form* pengiriman
- Angka 5: *Field* untuk memasukkan no kendaraan
- Angka 6: *Field* untuk memasukkan no surat jalan
- Angka 7: *Field* untuk memasukkan nama perusahaan
- Angka 9: *Field* untuk memasukkan tanggal pengiriman
- Angka 10: *Field* untuk memasukkan keterangan barang
- Angka 11: *Form* pengiriman
- Angka 12: *Field* untuk memasukkan nama barang
- Angka 13: *Field* untuk memasukkan harga satuan
- Angka 14: *Field* untuk memasukkan *quantity*
- Angka 15: *Field* untuk menampilkan harga total
- Angka 16: Merupakan tombol simpan

10. Antarmuka List Pengiriman

Pada gambar 5.18 adalah gambar antarmuka list pengiriman, dalam gambar antarmuka terdapat nomor dengan keterangan sebagai berikut:

- Angka 1: Nama perusahaan CV. Mitra Teknik
- Angka 2: Kolom pencarian
- Angka 3: Nama tabel
- Angka 4: Tombol tampil laporan
- Menu 1-14: Merupakan menu *sidebar* sistem
- Angka 5: Tombol navigasi admin
- Angka 6: Kolom untuk menampilkan nomor urut
- Angka 7: Kolom untuk menampilkan nomor surat jalan
- Angka 8: Kolom untuk menampilkan data nomor kendaraan
- Angka 9: Kolom untuk menampilkan data nama perusahaan
- Angka 10: Kolom untuk menampilkan data nama barang
- Angka 11: Kolom untuk menampilkan data tanggal pengiriman
- Angka 12: Kolom aksi berisi tombol detail, *edit*, *hapus*

1	2	5 O
Menu 1	3	
Menu 2	4	
Menu 3		
Sub Menu		
Sub Menu		
Menu 4	6 7 8 9 10 11 12	
Menu 5		
Menu 6		
Menu 7		

Gambar 5.18 Antarmuka List Pengiriman

11. Antarmuka Menambahkan *Supplier*

1	2	3 O
menu 1 menu 2 menu 3 menu 4 menu 5 menu 6 menu 7 menu 8 menu 9 menu 11 menu 12 menu 13 menu 14	4	6 7 8 9 11 10
5		

Gambar 5.19 Antarmuka menambahkan *supplier*

Pada gambar 5.19 adalah gambar antarmuka menambahkan *supplier*, dalam gambar antarmuka terdapat nomor dengan keterangan sebagai berikut:

- Angka 1: Nama perusahaan CV. Mitra Teknik
- Angka 2: Kolom pencarian
- Angka 3: Tombol navigasi admin
- Menu 1-14: Merupakan menu *sidebar* sistem
- Angka 4: Merupakan tombol tambah *supplier*
- Angka 5: Tabel menampilkan data *supplier*
- Angka 6: *Form pop-up* tambah *supplier*
- Angka 7: *Field* untuk memasukkan nama *supplier*
- Angka 8: *Field* untuk memasukkan no telepon
- Angka 9: *Field* untuk memasukkan alamat
- Angka 10: Merupakan tombol simpan
- Angka 11: Merupakan tombol *close*

12. Antarmuka List *Supplier*

Pada gambar 5.20 adalah gambar antarmuka list *supplier*, dalam gambar antarmuka terdapat nomor dengan keterangan sebagai berikut:

- Angka 1: Nama perusahaan CV. Mitra Teknik
- Angka 2: Kolom pencarian
- Angka 3: Tombol untuk tambah pelanggan
- Angka 4: Tombol navigasi admin
- Menu 1-7: Merupakan menu *sidebar* sistem
- Angka 5: Kolom untuk menampilkan nomor urut
- Angka 6: Kolom untuk menampilkan nama *supplier*
- Angka 7: Kolom untuk menampilkan alamat
- Angka 8: Kolom untuk menampilkan no telepon
- Angka 9: Kolom aksi berisi tombol *edit*, *hapus*

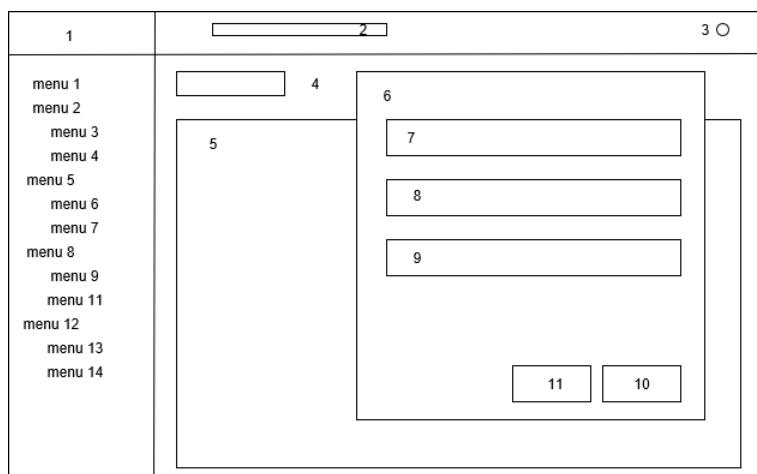
1	<input type="text"/> 2	4 <input type="radio"/>
Menu 1	<input type="text"/> 3	
Menu 2		
Menu 3		
Sub Menu		
Sub Menu		
Menu 4		
Menu 5		
Menu 6		
Menu 7		
	5 6 7 8 9	

Gambar 5.20 Antarmuka list *supplier*

13. Antarmuka Menambahkan Pelanggan

Pada gambar 5.21 adalah gambar antarmuka menambahkan pelanggan, dalam gambar antarmuka terdapat nomor dengan keterangan sebagai berikut:

- Angka 1: Nama perusahaan CV. Mitra Teknik
- Angka 2: Kolom pencarian
- Angka 3: Tombol navigasi admin
- Menu 1-14: Merupakan menu *sidebar* sistem
- Angka 4: Merupakan tombol tambah pelanggan
- Angka 5: Tabel menampilkan data *supplier*
- Angka 6: *Form pop-up* tambah *supplier*
- Angka 7: *Field* untuk memasukkan nama *supplier*
- Angka 8: *Field* untuk memasukkan no telepon
- Angka 9: *Field* untuk memasukkan alamat
- Angka 10: Merupakan tombol simpan
- Angka 11: Merupakan tombol *close*



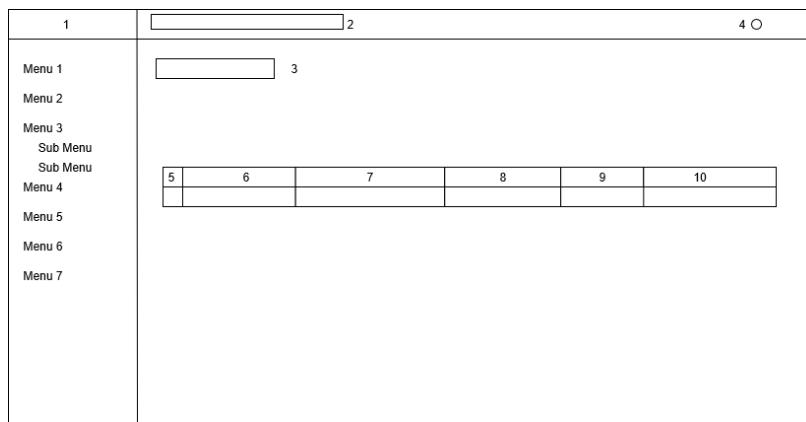
Gambar 5.21 Antarmuka menambahkan pelanggan

14. Antarmuka List Pelanggan

Pada gambar 5.22 adalah gambar antarmuka list pelanggan, dalam gambar antarmuka terdapat nomor dengan keterangan sebagai berikut:

- Angka 1: Nama perusahaan CV. Mitra Teknik
- Angka 2: Kolom pencarian
- Angka 3: Tombol untuk tambah pelanggan
- Angka 4: Tombol navigasi admin
- Menu 1-7: Merupakan menu *sidebar* sistem
- Angka 5: Kolom untuk menampilkan nomor urut
- Angka 6: Kolom untuk menampilkan nama pelanggan
- Angka 7: Kolom untuk menampilkan alamat
- Angka 8: Kolom untuk menampilkan no telepon

Angka 9: Kolom untuk status pelanggan
Angka 10: Kolom aksi berisi tombol edit, hapus

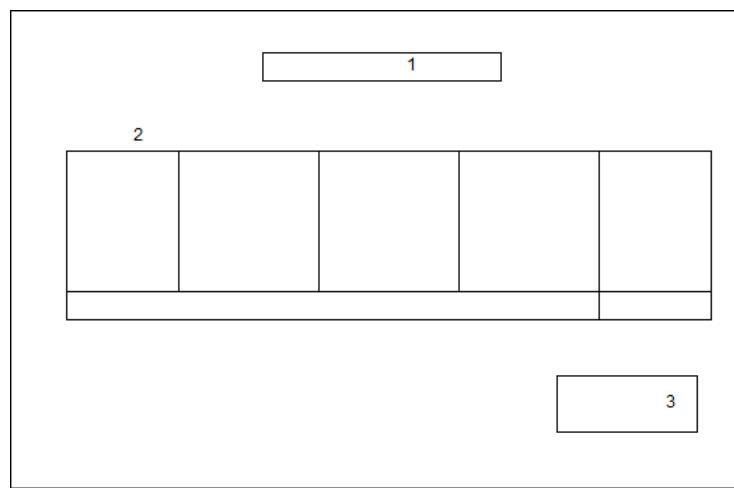


Gambar 5.22 Antarmuka list pelanggan

15. Antarmuka Menampilkan Laporan

Pada gambar 5.23 adalah gambar antarmuka laporan. Dalam rancangan antarmuka ini memiliki desain yang sama meliputi laporan pengiriman, laporan order masuk, laporan pembelian. Dalam gambar antarmuka terdapat nomor dengan keterangan sebagai berikut.

- Angka 1: Nama judul laporan
- Angka 2: Tabel untuk menampilkan data laporan
- Angka 3: Merupakan letak untuk tanda tangan



Gambar 5.23 Antarmuka menampilkan laporan

5.2 Implementasi

Setelah menyelesaikan tahap perancangan sistem maka selanjutnya adalah tahap implementasi. Perancangan yang telah dibuat akan digunakan sebagai dasar untuk melakukan proses tahap implementasi. Pada tahap

implementasi akan menjelaskan spesifikasi sistem perangkat keras dan perangkat lunak, implementasi arsitektur, implementasi pseudocode, implementasi basis data, dan implementasi antarmuka. Untuk penjelasan selengkapnya dapat dilihat pada bagian berikut.

5.2.1 Spesifikasi Sistem

Pada bagian ini akan menjelaskan spesifikasi dari perangkat keras dan perangkat lunak yang digunakan dalam tahap implementasi sistem.

5.2.1.1 Spesifikasi Perangkat Keras

Berikut spesifikasi perangkat keras yang digunakan dalam tahap implementasi sistem, dapat dilihat pada tabel 5.8.

Tabel 5.8 Tabel Spesifikasi Perangkat Keras

Nama Komponen	Keterangan
Harddisk	1 TB
Processor	Intel(R) Core(TM) i5-7200U CPU @ 2.71 GHz
RAM	4 GB DDR4
Graphic Card	Intel HD Graphics 520 & Nvidia GeForce GT930MX

5.2.1.2 Spesifikasi perangkat lunak

Berikut spesifikasi perangkat keras yang digunakan dalam tahap implementasi sistem, dapat dilihat pada tabel 5.9.

Tabel 5.9 Tabel Spesifikasi Perangkat Lunak

Hardware	Keterangan
Sistem Operasi	Windows 10 Professional 64-bit
Tools	Sublime Text 3
Bahasa Pemrograman	PHP 7, HTML 5, CSS 3, Javascript
DBMS	MySQL phpMyadmin 4.8.3
Server	localhost (XAMPP) 3.2.2
Framework	CodeIgniter 3.1.6
Browser	Google Chrome 70.0.3538.110 & Internet Explore 42.17134.1.0
Service library	RESTful Web Service
Library client	CURL

5.2.2 Implementasi Class

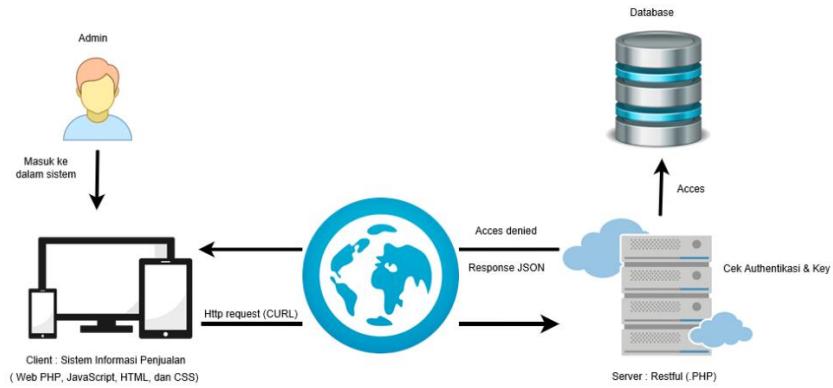
Implementasi *class* dilakukan mengacu pada hasil perancangan *class diagram* sebelumnya. Setiap *class* yang terdapat pada *class diagram*, akan diimplementasikan ke dalam file dengan ekstensi (.php). Berikut daftar implementasi *class* dapat diperhatikan pada tabel 5.10 berikut :

Tabel 5.10 Implementasi Class

No	Nama Class	Nama File Server
1	Gudang	Server/index.php/Gudang.php
2	Order keluar	Server/index.php/Order_keluar.php
3	Order masuk	Server/index.php/Order_masuk.php
4	Pelanggan	Server/index.php/Pelanggan.php
5	Pembelian	Server/index.php/Pembelian.php
6	Penawaran	Server/index.php/Penawaran.php
7	Suplier	Server/index.php/Suplier.php
8	Laporan order masuk	Server/index.php/Laporan_ordermsk.php
9	Laporan pembelian	Server/index.php/Laporan_pembelian.php
10	Laporan pengiriman	Server/index.php/Laporan_pengiriman.php
11	Pembelian barang	Server/index.php/Pembelian_barang.php
12	Status	Server/index.php/Status.php
No	Nama Class	Nama File Client
1	Gudang	ClientMT/index.php/Gudang.php
2	Order keluar	ClientMT/index.php/Order_keluar.php
3	Order masuk	ClientMT/index.php/Order_masuk.php
4	Pelanggan	ClientMT/index.php/Pelanggan.php
5	Pembelian	ClientMT/index.php/Pembelian.php
6	Penawaran	ClientMT/index.php/Penawaran.php
7	Suplier	ClientMT/index.php/Suplier.php
8	Pembelian barang	ClientMT/index.php/Pembelian_barang.php

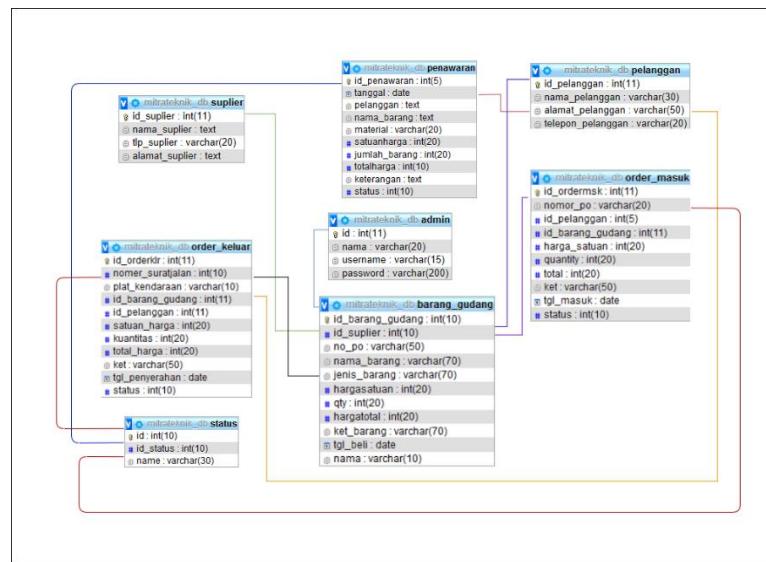
5.2.3 Implementasi Arsitektur

Pada bagian ini akan menggambarkan rangkaian yang telah dirancang dan diimplementasikan dengan benar. Pada gambar 5.24 implementasi *client* menggunakan bahasa pemrograman *web*, kemudian ketika mengirim *request* pada *server* menggunakan *library CURL*. *CURL* digunakan untuk memindai data dari atau ke sebuah *server* tanpa interaksi dari masing-masing *user*. Setelah sampai pada *server*, kemudian akses diperiksa dengan autentikasi dan *key*, jika tidak sesuai maka akses akan ditolak. Ketika akses berhasil maka akan melakukan akses *database* dan mengirim kembali melalui *response json*.



Gambar 5.24 Implementasi Arsitektur

5.2.4 Implementasi Basis Data



Gambar 5.25 Implementasi Basis Data

Pada gambar 5.25 menggambarkan implementasi basis data yang digunakan pada sistem informasi informasi penjualan. Dalam implementasinya sistem ini menggunakan total 7 tabel database.

5.2.5 Implementasi Pseudocode

Pada bagian ini akan menjelaskan mengenai implementasi dari perancangan pseudocode pada tahap perancangan sebelumnya.

1. Implementasi pada fungsi aksi_login()

Implementasi fungsi aksi_login dapat dilihat pada tabel 5.11.

Tabel 5.11 Implementasi fungsi aksi_login()

1	<pre>public function aksi_login(){</pre>
2	<pre> \$cek = \$this->model_login->logged_id();</pre>
3	<pre> if (\$this->model_login->logged_id())</pre>
4	<pre> {</pre>
5	<pre> redirect ('dashboard');</pre>
6	<pre> }else{</pre>
7	<pre> \$this->form_validation->set_rules</pre>
8	<pre> ('username', 'username', 'required');</pre>
9	<pre> \$this->form_validation->set_rules</pre>
10	<pre> ('password', 'password', 'required');</pre>
11	<pre> \$this->form_validation->set_message</pre>
12	<pre>('required', '<div class="alert alert-</pre>
13	<pre><div class="header"><i class="fa fa-exclamation-</pre>
14	<pre>circle"></i> {field} harus diisi</div></div>');</pre>
15	<pre>if (\$this->form_validation->run() == TRUE) {</pre>
16	<pre> \$username = \$this->input->post('username', TRUE);</pre>
17	<pre> \$password = MD5(\$this->input->post('password', TRUE));</pre>
18	<pre> \$checking = \$this->model_login->check_login</pre>
19	<pre>('admin', array('username'=>\$username),</pre>
20	<pre>array('password'=>\$password));</pre>
21	<pre> if (\$checking != FALSE) {</pre>
22	<pre> foreach (\$checking as \$apps) {</pre>
23	<pre> \$session_data = array(</pre>
24	<pre> 'id' => \$apps->id,</pre>
25	<pre> 'nama' => \$apps->nama,</pre>
26	<pre> 'username' => \$apps->username,</pre>
27	<pre> 'password' => \$apps->password);</pre>
28	<pre>\$this->session->set_userdata(\$session_data);</pre>
29	<pre> echo "<script>alert('Selamat datang di halaman system</pre>
30	<pre>informasi penjualan');</script>";</pre>
41	<pre> echo "<script>window.location.href = '" . base_url() .</pre>
42	<pre>"dashboard';</script>";</pre>
43	<pre> helper_log("login", "masuk ke dalam sistem");</pre>
44	<pre> }</pre>
45	<pre>}else{</pre>
46	<pre> echo "<script>alert('Error!, Username atau Password</pre>
47	<pre>salah');</script>";</pre>
48	<pre> \$this->load->view('main/login_view');</pre>
49	<pre>}</pre>

Tabel 5.11 Implementasi fungsi aksi_login() (lanjutan)

```
50     }else{
51         $this->load->view('main/login_view');
52     }
53 }
54 }
55 }
```

2. Implementasi fungsi Tambah_Order_Masuk()

Implementasi fungsi Tambah_Order_Masuk *CI_Controller* dapat dilihat pada tabel 5.12.

Tabel 5.12 Implementasi fungsi Tambah_Order_Masuk

```
1 function tambah_order_masuk(){
2     $data = array(
3         'id_ordermsk'      => $this->input->post('id_ordermsk'),
4         'nomor_po'        => $this->input->post('nomor_po'),
5         'id_pelanggan'   => $this->input->post('id_pelanggan'),
6         'id_barang_gudang'=> $this->input->post('id_barang_gudang'),
7         'harga_satuan'    => $this->input->post('harga_satuan'),
8         'quantity'        => $this->input->post('quantity'),
9         'total'            => $this->input->post('total'),
10        'ket'              => $this->input->post('ket'),
11        'tgl_masuk'       => $this->input->post('tgl_masuk'));
12
13        $insert = $this->curl->simple_post($this->API.'/order_masuk',
14        $data);
15
16        if($insert) {
17            $this->session->set_flashdata('Csukses','insert data sukses');
18        }else{
19            $this->session->set_flashdata('Cgagal','insert data gagal');
20        }
21        redirect ('order_masuk'); }
```

3. Implementasi fungsi index_post()

Implementasi fungsi *index_post()* pada *Order_Masuk REST_Controller* dapat dilihat pada tabel 5.13.

Tabel 5.13 Implementasi fungsi *index_post()*

```
1 function index_post() {  
2     $data = array(  
3         'id_ordermsk'      => $this->post('id_ordermsk'),  
4         'nomor_po'          => $this->post('nomor_po'),  
5         'id_pelanggan'      => $this->post('id_pelanggan'),  
6         'id_barang_gudang'   => $this->post('id_barang_gudang'),  
7         'harga_satuan'       => $this->post('harga_satuan'),  
8         'quantity'           => $this->post('quantity'),  
9         'total'               => $this->post('total'),  
10        'ket'                 => $this->post('ket'),  
11        'tgl_masuk'          => $this->post('tgl_masuk'));  
12        $insert = $this->db->insert('order_masuk', $data);  
13        if ($insert) {  
14            $this->response($data, 200);  
15        } else {  
16            $this->response(array('status' => 'fail', 502));  
17        }  
18    }
```

4. Implementasi fungsi *index_put()*

Implementasi fungsi *index_put()* pada *Order_Masuk REST_Controller* dapat dilihat pada tabel 5.14.

Tabel 5.14 Implementasi fungsi *index_put()*

```
1 function index_put() {  
2     $id = $this->put('id_ordermsk');  
3     $stat = $this->put('status');  
4     if($stat == 3){  
5         $data1 = array(  
6             'id_ordermsk'      => $this->put('id_ordermsk'),  
7             'nomor_po'          => $this->put('nomor_po'),  
8             'id_pelanggan'      => $this->put('id_pelanggan'),  
9             'id_barang_gudang'   => $this->put('id_barang_gudang'),  
10            'harga_satuan'       => $this->put('harga_satuan'),  
11            'quantity'           => $this->put('quantity'),  
12            'total'               => $this->put('total'),  
13            'ket'                 => $this->put('ket'),  
14            'tgl_masuk'          => $this->put('tgl_masuk'),  
15            'status'              => $this->put('status')
```

Tabel 5.14 Implementasi fungsi *Index_put()* (lanjutan)

```

17      );
18      $this->db->where('id_ordermsk', $id);
19      $update = $this->db->update('order_masuk', $data1);
20      $this->response($data1,200);
21      }elseif($stat == 4){
22          $data2 = array(
23              'id_ordermsk'      => $this->put('id_ordermsk'),
24              'nomor_po'         => $this->put('nomor_po'),
25              'id_pelanggan'    => $this->put('id_pelanggan'),
26              'id_barang_gudang' => $this->put('id_barang_gudang'),
27              'harga_satuan'     => $this->put('harga_satuan'),
28              'quantity'         => $this->put('quantity'),
29              'total'             => $this->put('total'),
30              'ket'               => $this->put('ket'),
31              'tgl_masuk'        => $this->put('tgl_masuk'),
32              'status'            => $this->put('status'));
33          $data3 = array(
34              'nomer_suratjalan' => 0,
35              'plat_kendaraan'   => 'belum ada',
36              'id_barang_gudang' => $this-
37 >put('id_barang_gudang'),
38              'id_pelanggan'     =>$this->put('id_pelanggan'),
39              'satuan_harga'     => 0,
40              'kuantitas'         =>$this->put('quantity'),
41              'total_harga'       =>0,
42              'ket'               =>$this->put('ket'),
43              'tgl_penyerahan'   =>$this->put('tgl_masuk'),
44              'status'            =>$this->put('status'));

45      $this->db->where('id_ordermsk', $id);
46      $update = $this->db->update('order_masuk', $data2);
47      $m = $this->coba->insertOrderkeluar($data3);
48      $this->response($data2,200);
49      }else{
50          $this->response(array('status' => 'fail', 502));
51      }
52  }
53

```

5. Implementasi fungsi *Edit_Pembelian()*

Implementasi fungsi *Edit_Pembelian()* pada dapat dilihat pada tabel 5.15.

Tabel 5.15 Implementasi fungsi *Edit_Pembelian()*

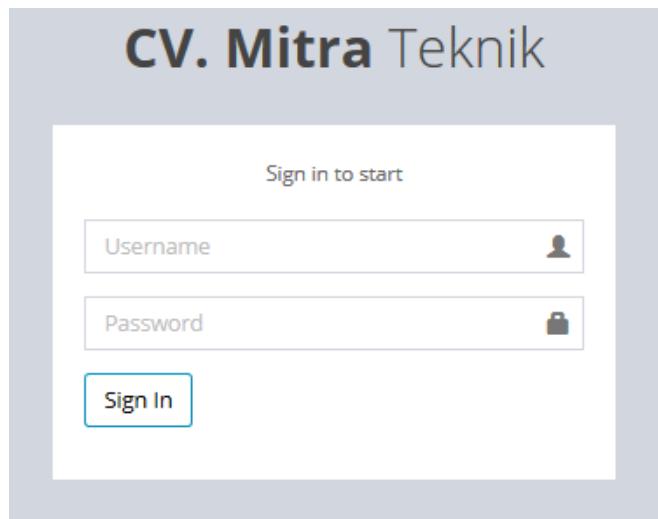
```
1 function edit_pembelian(){
2     $data = array(
3         'id_barang_gudang' => $this->input->post('id_barang_gudang'),
4         'no_po'           => $this->input->post('no_po'),
5         'id_suplier'      => $this->input->post('id_suplier'),
6         'nama_barang'     => $this->input->post('nama_barang'),
7         'jenis_barang'    => $this->input->post('jenis_barang'),
8         'ket_barang'       => $this->input->post('ket_barang'),
9         'tgl_beli'         => $this->input->post('tgl_beli'),
10        'status_barang'   => $this->input->post('status_barang'));
11
12        $update = $this->curl->simple_put($this->API.'/pembelian',
13        $data, array(CURLOPT_BUFFERSIZE => 90));
14        if($update) {
15            $this->session->set_flashdata('Update data berhasil');
16        }else{
17            $this->session->set_flashdata('Update data gagal');
18        }
19        redirect('pembelian');
20    }
```

5.2.6 Implementasi Antarmuka

Pada proses implementasi antarmuka mengacu pada hasil perancangan antarmuka pada tahap sebelumnya. Masing-masing implementasi dari hasil perancangan antarmuka yang telah dirancang diantaranya dapat dilihat pada bagian berikut.

1. Tampilan antarmuka *Login*

Hasil implementasi antarmuka *Login* dapat dilihat pada gambar 5.26 berikut.



Gambar 5.26 Antarmuka *login*

2. Tampilan antarmuka Menambahkan *user*

Hasil implementasi antarmuka menambahkan *user* dapat dilihat pada gambar 5.27 berikut.

A screenshot of a web-based user registration interface. At the top center, it says "CV. Mitra Teknik". Below that, there is a "Register Your Account" message. There are three input fields: "Nama" (with placeholder "Nama Lengkap"), "Username" (with placeholder "Name Pengguna"), and "Password" (with placeholder "Kata Sandi"). At the bottom of the form are two red buttons: "REGISTER" and "BACK".

Gambar 5.27 Antarmuka menambahkan *user*

3. Tampilan antarmuka Menambahkan Penawaran

Hasil implementasi antarmuka menambahkan penawaran dapat dilihat pada gambar 5.28 berikut.

Gambar 5.28 Antarmuka menambahkan penawaran

4. Tampilan antarmuka List Penawaran

Hasil implementasi antarmuka list penawaran dapat dilihat pada gambar 5.29 berikut.

NO	TANGGAL	NAMA PELANGGAN	NAMA BARANG	HARGA (PCS)	KETERANGAN	STATUS	AKSI
1	08 February 2019	PT. Multi Manao Indonesia	Material besi	Rp. 120.000	baru	Disetujui, ke tahap 2	<button>Cetak</button> <button>Edit</button>
2	01 January 1970	PT. Multi Manao Indonesia	Sepatu baru	Rp. 12.000	Pembuatan baru	Disetujui, ke tahap 2	<button>Cetak</button> <button>Edit</button>
3	26 March 2019	PT. Ecco Indonesia	Grinding	Rp. 12.000	Pembuatan baru	Disetujui, ke tahap 2	<button>Cetak</button> <button>Edit</button>
4	26 March 2019	PT. Mustika Light Metal	Fanbolt	Rp. 12.000	Pembuatan baru	Disetujui, ke tahap 2	<button>Cetak</button> <button>Edit</button>

Gambar 5.29 Antarmuka list penawaran

5. Tampilan antarmuka List Order Masuk

Hasil implementasi antarmuka list *order masuk* dapat dilihat pada gambar 5.30 berikut.

NO	NOMOR PO	NAMA PELANGGAN	BARANG	TANGGAL MASUK	STATUS	AKSI
1	13441122	PT. Japfa Comfeed Indonesia	Baut ukuran M14	02 March 2019	Pengerjaan selesai, ke tahap 4	[Detail] [Edit] [Hapus]
2	23123	PT. Yanaprima Hastapersada	Baut ukuran M14	25 June 2019	Pengerjaan selesai, ke tahap 4	[Detail] [Edit] [Hapus]
3	231231	PT. Japfa Comfeed	Baut Ukuran M18	26 June 2019	Pengerjaan selesai, ke tahap 4	[Detail] [Edit] [Hapus]

Gambar 5.30 Antarmuka list *order* masuk

6. Tampilan antarmuka Menambahkan *Order* Masuk

Hasil implementasi antarmuka menambahkan *order* masuk dapat dilihat pada gambar 5.31 berikut.

Gambar 5.31 Antarmuka menambahkan *order* masuk

7. Tampilan antarmuka Menambahkan Pembelian Barang

Hasil implementasi antarmuka menambahkan pembelian dapat dilihat pada gambar 5.32 berikut.

Company Form

Input

No Purchase Order
[Text Input]

Nama Supplier
[Select Box] - Pilih -

Tanggal Pembelian
[Text Input]

Keterangan Barang
[Text Input]

Barang Form

Nama Barang
[Text Input]

Jenis Barang
[Text Input]

Harga Satuan
Rp [Text Input] | [Text Input]

Quantity
1

Total Harga
0

SIMPAN

Gambar 5.32 Antarmuka menambahkan pembelian barang

8. Tampilan antarmuka List Stok Barang

Hasil implementasi antarmuka list stok barang dapat dilihat pada gambar 5.33 berikut.

Data Pembelian

Tampil Laporan

NO	TANGGAL PEMBELIAN	BARANG	QUANTITY	HARGA (PCS)
1	26 February 2019	Fan belt	1	Rp. 12.000
2	02 February 2019	Baut ukuran M14	0	Rp. 12.000
3	02 February 2019	Motabor	9	Rp. 12.000
4	03 April 2019	Pipa	2	Rp. 23.000
5	10 April 2019	Roll Rubber	0	Rp. 24.500
6	09 April 2019	Baut Ukuran M16	3	Rp. 2.349.000

Gambar 5.33 Antarmuka list stok barang

9. Tampilan antarmuka Menambahkan Pengiriman

Hasil implementasi antarmuka menambahkan pengiriman dapat dilihat pada gambar 5.34 berikut.

Gambar 5.34 Antarmuka menambahkan pengiriman

10. Tampilan antarmuka List Pengiriman

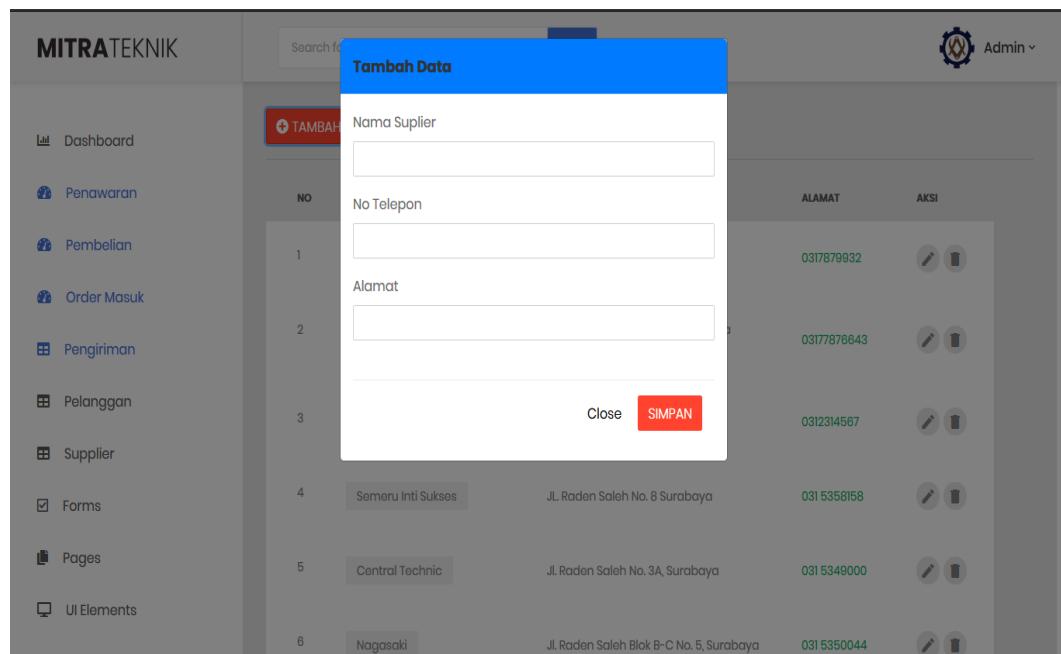
Hasil implementasi antarmuka list pengiriman dapat dilihat pada gambar 5.35 berikut.

NO	NOMOR SURAT JALAN	NOMOR KENDARAAN	NAMA PERUSAHAAN	BARANG	TANGGAL PENGIRIMAN	STATUS	AKSI
1	100	belum ada	PT. Forindo Prima Perkasa	Pipa	26 June 2019	finish	Detail Edit
2	23441	W 2712 WB	PT. Indospring Tbk	Fan belt	17 July 2019	finish	Detail Edit
3	232132	W 2712 WB	PT. Ecco Indonesia	Conveyor	20 August 2019	finish	Detail Edit

Gambar 5.35 Antarmuka list pengiriman

11. Tampilan antarmuka Menambahkan *Supplier*

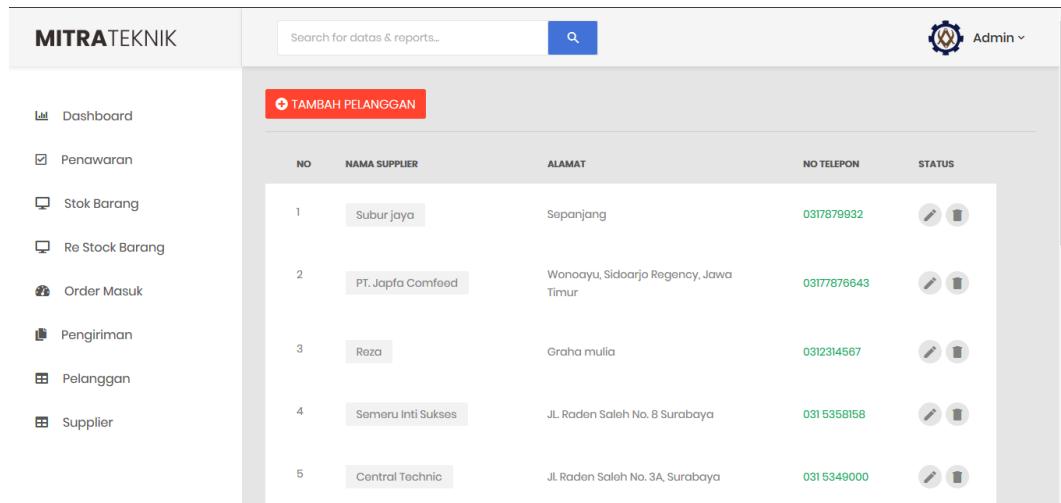
Hasil implementasi antarmuka menambahkan *supplier* dapat dilihat pada gambar 5.36 berikut.



Gambar 5.36 Antarmuka menambahkan *supplier*

12. Tampilan antarmuka List *Supplier*

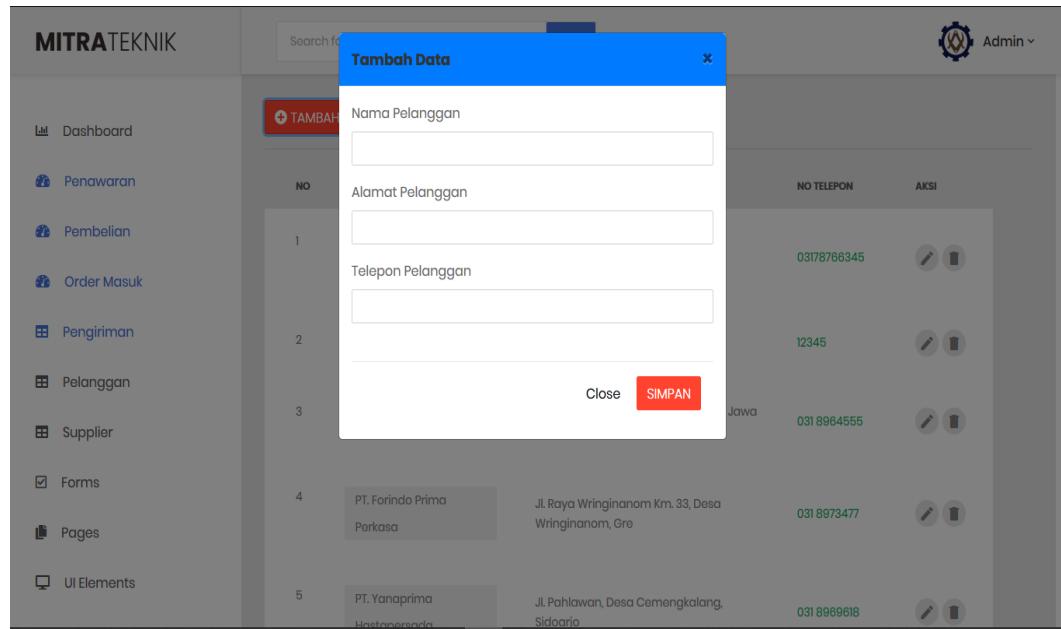
Hasil implementasi antarmuka list *supplier* dapat dilihat pada gambar 5.37 berikut.



Gambar 5.37 Antarmuka list *supplier*

13. Tampilan antarmuka Menambahkan Pelanggan

Hasil implementasi antarmuka menambahkan pelanggan dapat dilihat pada gambar 5.38 berikut.



Gambar 5.38 Antarmuka menambahkan pelanggan

14. Tampilan antarmuka List Pelanggan

Hasil implementasi antarmuka list pelanggan dapat dilihat pada gambar 5.39 berikut.

NO	NAMA PELANGGAN	ALAMAT	NO TELEPON	STATUS	AKSI
1	PT. Multi Manao Indonesia	Driyorejo - Gresik	03178766345	Aktif	
2	PT. Japfa Comfeed	Sawunggaling - Sidoarjo	031 99716930	Aktif	
3	PT. Ecco Indonesia	Jl. Raya Bligo No.17 Candi - Sidoarjo Jawa Timur.	031 8964555	Aktif	
4	PT. Forindo Prima Perkasa	Jl. Raya Wringinanom Km. 33, Gresik	031 8973477	Aktif	

Gambar 5.39 Antarmuka list pelanggan

15. Tampilan antarmuka menampilkan laporan

Hasil implementasi antarmuka menampilkan laporan dapat dilihat pada gambar 5.40 berikut.

LAPORAN PEMBELIAN

No Purchase Order	Nama Suplier	Nama Barang	Tanggal Pembelian	Harga Total
26022019667	Aneka Logam Sejahtera	Fan belt	2019-02-26	Rp. 24.000
26122019667	Tk Besi Cahaya Agung	Matabor	2019-02-02	Rp. 24.000
26122019667	Tk Besi Cahaya Agung	Matabor	2019-02-02	Rp. 24.000
887211	Central Technic	Pipa	2019-04-03	Rp. 46.000
323221	Sumber Usaha	Roll Rubber	2019-04-10	Rp. 49.000
211311	PT. Japfa Comfeed	Roll rubber	2019-04-09	Rp. 7.047.000
34455	Central Technic	rubber	2019-05-06	Rp. 155.000
78882	Central Technic	Rubber	2019-05-09	Rp. 90.000
123321	AJBS	Baut	0000-00-00	Rp. 13.500
12345	Nagasaki	Rubber	2019-05-15	Rp. 240.000
12300121	PT. Japfa Comfeed	Conveyor	2019-07-24	Rp. 24.000
12300121	PT. Japfa Comfeed	Conveyor	2019-07-23	Rp. 24.000
1607201901	Central Technic	Gerinda	2019-07-16	Rp. 8.250.000
20190717001	Nagasaki	Gerinda	2019-07-17	Rp. 4.700.000
20190717001	Central Technic	Gerinda	2019-07-17	Rp. 2.900.000
Total				Rp. 23.610.500

Sidoarjo, 22-Jul-2019

(reza)

Gambar 5.40 Antarmuka menampilkan laporan

BAB 6 PENGUJIAN DAN ANALISIS

6.1 Pengujian *White-box*

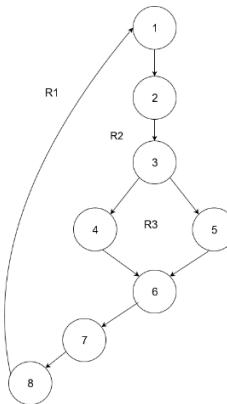
Pada tahap pengujian *white-box* yang dilakukan dalam penelitian ini hanya berfokus pada pengujian antar unit dengan tipe pengujian yang digunakan yaitu *path coverage* atau yang biasa disebut dengan *basis path testing*. Pengujian dilakukan terhadap tiga fungsi pada sistem informasi penjualan CV. Mitra Teknik yaitu tambah_order_masuk pada *class order_masuk*, Edit_Pembelian pada *class pembelian*, dan Tambah_Pembelian pada *class pembelian*. Tambah_order_masuk merupakan *method* untuk mengirim permintaan tambah *order* masuk dari *class order_masuk* pada *client* untuk dikirim menuju *class order_masuk* pada bagian *server*. Edit_Pembelian merupakan *method* dari *class pembelian* yang berfungsi untuk mengirim permintaan dari *class pembelian* pada bagian *client* dikirim menuju *class pembelian* bagian *server*. Hapus_order_masuk merupakan *method* dari *class order_masuk* yang berfungsi untuk mengirim permintaan dari *class order_masuk* bagian *client* untuk dikirim menuju *class order_masuk* bagian *server*.

6.1.1 Pengujian Fungsi Tambah_order_masuk

Tabel 6.1 Algoritme fungsi tambah_order_masuk

1	function tambah_order_masuk 1
2	data = array(2
3	id_ordermsk => input post(id_ordermsk), 2
4	nomor_po => input post(nomor_po), 2
5	id_pelanggan => input post(id_pelanggan), 2
6	id_barang_gudang => input post(id_barang_gudang), 2
7	harga_satuan => input post(harga_satuan), 2
8	quantity => input post(quantity), 2
9	total => input post(total), 2
10	ket => input post(ket), 2
11	tgl_masuk => input post(tgl_masuk)) 2
12	
13	insert = curl simple_post(API.'/order_masuk', \$data); 2
14	if (insert) then 3
15	session set_flashdata('Csukses','insert data sukses'); 4
16	else 5
17	session set_flashdata('Cgagal','insert data gagal'); 5
18	end if 6
19	redirect(order_masuk); 7
20	end if 8

Pada tabel 6.1 merupakan algoritme dari fungsi tambah_order_masuk untuk menjelaskan alur kode baris program sebelum dilakukan pengujian menggunakan *flowgraph*. *Flowgraph* digunakan untuk melihat alur kode program perbaris apakah sudah berjalan dengan baik atau belum. *Flowgraph* fungsi *create_func* dapat dilihat pada gambar 6.1 berikut.



Gambar 6.1 Flowgraph tambah_order_masuk

Pada gambar 6.1 merupakan *flowgraph* dari fungsi tambah_order_masuk. Pada *flowgraph* tersebut terdapat tiga region, delapan *node* dan sembilan *edge* yang digunakan untuk menghitung *cyclomatic complexity*. Untuk menghitung *cyclomatic complexity* (mencari nilai $V(G)$) dapat dilihat pada bagian berikut.

Perhitungan $V(G)$:

- $V(G) = \text{Total Ruang (R), Total } R = 3$
- $V(G) = \text{Edge (Jumlah Panah)} - \text{Node (Jumlah Node)} + 2 = 9 - 8 + 2 = 3$

Jalur Independen :

R1: 1-8

R2: 1-2-3-4-6-7-8

R3: 1-2-3-5-6-7-8

Tabel 6.2 Pengujian unit fungsi Tambah_order_masuk

No	Jalur	Data	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	1-8	<i>Function tambah_order_masuk = Start</i>	Sistem memanggil method tambah_order_masuk untuk memulai proses, data dimasukkan ke dalam variabel data, kemudian mengirim data ke service.	<i>Method tambah_order_masuk berjalan dan mengirim data ke service</i>	<i>Valid</i>
2	1-2-3-4-6-7-8	<i>Function tambah_order_masuk = insert</i>	Sistem menampilkan pesan <i>insert data sukses</i>	Sistem menampilkan pesan <i>sukses berhasil</i>	<i>Valid</i>

Tabel 6.2 Pengujian unit fungsi Tambah_order_masuk (lanjutan)

				menambahkan data	
3	1-2-3-5-6-7-8	Function tambah_order_masuk = !insert	Sistem menampilkan insert data gagal	Sistem menampilkan pesan gagal menambahkan data	Valid

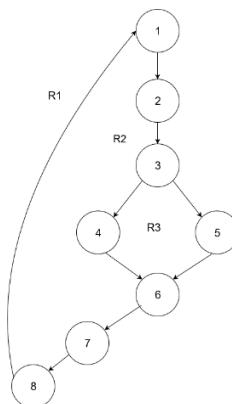
6.1.2 Pengujian Fungsi *Edit_pembelian*

Tabel 6.3 Algoritme fungsi *Edit_pembelian*

```

1 function edit_pembelian 1
2   data = array( 2
3     id_barang_gudang      => input post (id_barang_gudang), 2
4     no_po                 => input post (no_po), 2
5     id_suplier            => input post (id_suplier), 2
6     nama_barang           => input post (nama_barang), 2
7     jenis_barang          => input post (jenis_barang), 2
8     hargasatuan          => input post (harga_satuan), 2
9     qty                   => input post (qty), 2
10    hargatotal            => input post (hargatotal), 2
11    ket_barang             => input post (ket_barang) , 2
12    tgl_beli               => input post (tgl_beli)) 2
13    update = curl->simple_put(API.'/pembelian', data); 2
14  if(update) then 2
15    session set_flashdata('Update data berhasil'); 3
16  else 4
17    Session set_flashdata('Update data gagal'); 4
18  end if 6
19  redirect('order_masuk'); 7
20 end if 8

```



Gambar 6.2 Flowgraph *edit_pembelian*

Pada gambar 6.2 merupakan *flowgraph* dari fungsi *edit_pembelian*. Pada *flowgraph* tersebut terdapat tiga region, delapan *node* dan sembilan *edge* yang digunakan untuk menghitung *cyclomatic complexity*. Untuk menghitung *cyclomatic complexity* (mencari nilai $V(G)$) dapat dilihat pada bagian berikut.

Perhitungan $V(G)$:

- $V(G) = \text{Total Ruang (R)}, \text{Total R} = 3$
- $V(G) = \text{Edge (Jumlah Panah)} - \text{Node (Jumlah Node)} + 2$
 $= 9 - 8 + 2 = 3$

Jalur Independen :

R1: 1-8

R2: 1-2-3-4-6-7-8

R3: 1-2-3-5-6-7-8

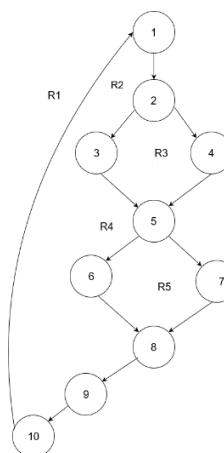
Tabel 6.4 Pengujian unit fungsi *edit_pembelian*

No	Jalur	Data	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	1-8	<i>Function edit_pembelian = Start</i>	Sistem memanggil <i>method edit_pembelian</i> untuk memulai proses, data dimasukkan ke dalam variabel data, kemudian mengirim data ke <i>service</i> .	<i>Method edit_pembelian berjalan dan mengirim data ke service</i>	<i>Valid</i>
2	1-2-3-5-6	<i>Function edit_pembelian = update</i>	Sistem menampilkan pesan <i>update data berhasil</i>	Sistem menampilkan pesan sukses berhasil mengubah data	<i>Valid</i>
3	1-2-4-5-6	<i>Function edit_pembelian = !update</i>	Sistem menampilkan pesan <i>update data gagal</i>	Sistem menampilkan pesan gagal mengubah data	<i>Valid</i>

6.1.3 Pengujian Fungsi Hapus_order_masuk

Tabel 6.5 Algoritme fungsi hapus_order_masuk

1	<code>function hapus_order_masuk (id) 1</code>
2	<code> if (empty (id)) 2</code>
3	<code> redirect(order_masuk) 3</code>
4	<code> else 4</code>
5	<code> delete = curl simple_delete (API./order_masuk, array(id_ordermsk'=> id));</code>
6	
7	<code> if (delete) 5</code>
8	<code> session set_flashdata(sukses, Delete Data Berhasil) 6</code>
9	<code> else if 7</code>
10	<code> session set_flashdata(gagal, Delete Data Gagal) 7</code>
11	<code> redirect(order_masuk) 8</code>
12	<code> end if 9</code>
13	<code>end if 10</code>



Gambar 6.3 Flowgraph hapus_order_masuk

Pada gambar 6.3 merupakan *flowgraph* dari fungsi hapus_order_masuk. Pada *flowgraph* tersebut terdapat tiga region, delapan *node* dan sembilan *edge* yang digunakan untuk menghitung *cyclomatic complexity*. Untuk menghitung *cyclomatic complexity* (mencari nilai $V(G)$) dapat dilihat pada bagian berikut.

Perhitungan $V(G)$:

- $V(G) = \text{Total Ruang (R), Total } R = 5$
- $V(G) = \text{Edge} - \text{Node} + 2$
 $= 11 - 10 + 2 = 3$

Jalur Independen :

R1: 1-11

R2: 1-2-3-5-6-7-8-9-10

R3: 1-2-3-4-6-7-8-9-10

Jalur independen lanjutan :

R4: 1-2-3-4-5-6-8-9-10

R5: 1-2-3-4-5-6-7-9-10

Tabel 6.6 Algoritme fungsi hapus_order_masuk

No	Jalur	Data	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	1-10	<i>Function hapus_order_masuk = Start</i>	Sistem memanggil <i>method hapus_order_masuk</i> untuk memulai proses	<i>Method hapus_order_masuk</i> berjalan	<i>Valid</i>
2	1-2-3-5-6-7-8-9-10	<i>Function hapus_order_masuk (id), id != empty</i>	Parameter <i>id</i> dicek apakah tersedia atau tidak	Paramater <i>id</i> terdefinisi dan menjalankan proses selanjutnya	<i>Valid</i>
3	1-2-3-4-6-7-8-9-10	<i>Function hapus_order_masuk (id), id = empty</i>	Parameter <i>id</i> dicek apakah tersedia atau tidak	Paramater <i>id</i> tidak tersedia dan menjalankan proses selanjutnya	<i>Valid</i>
4	1-2-3-4-5-6-8-9-10	<i>Function hapus_order_masuk (id) = delete</i>	Sistem menampilkan pesan <i>delete</i> data berhasil	Sistem menampilkan pesan sukses dan berhasil menghapus data	<i>Valid</i>
5	1-2-3-4-5-6-7-9-10	<i>Function hapus_order_masuk (id) = !delete</i>	Sistem menampilkan pesan <i>delete</i> data gagal	Sistem menampilkan pesan gagal menghapus data	<i>Valid</i>

6.2 Pengujian *Black-box*

Pengujian *black-box* dilakukan untuk mengetahui apakah program berjalan dengan benar dilihat dari proses *input* dan *output*. Untuk pengujian fungsional dilakukan dengan tipe pengujian yaitu pengujian validasi atau dengan kasus uji. Pengujian non-fungsional dilakukan untuk mengetahui hasil dari pengujian kebutuhan non-fungsional.

6.2.1 Pengujian Fungsional

Pengujian kebutuhan fungsional dilakukan untuk mengetahui hasil pengujian dengan cara kasus uji, dan diharapkan *output* berjalan sesuai dengan *input* dalam program. Berikut tabel kebutuhan fungsional.

Tabel 6.7 Pengujian validasi kebutuhan fungsional

No	Nama Kebutuhan	Kasus Uji	Hasil yang diharapkan	Hasil yang diberikan	Status
1	<i>Login</i>	<i>User</i> memasukkan: Kode username = “1234” dan password = “1234”	Sistem menampilkan pesan <i>error</i> “Error! Username atau password salah” dan menampilkan halaman <i>login</i>	Berhasil <i>login</i> sesuai dengan otoritas dan menampilkan <i>dashboard</i>	<i>Valid</i>
2	Menambahkan Penawaran	<i>User</i> memasukkan data pada <i>form</i> sesuai kolom <i>input</i>	Berhasil menyimpan data Penawaran ke dalam <i>database</i>	Data berhasil disimpan ke dalam <i>database</i> dan sistem menampilkan data dalam fitur Penawaran	<i>Valid</i>
3	Mencetak Penawaran	<i>User</i> menekan tombol cetak pada tabel daftar penawaran	Sistem berhasil menampilkan <i>form</i> penawaran dalam format PDF	Data <i>input</i> berhasil di <i>generate</i> ke dalam format PDF	<i>Valid</i>
4	Menghapus Penawaran	<i>User</i> menekan tombol Hapus pada tabel daftar penawaran	Sistem akan menampilkan peringatan dan menghapus data jika permintaan dilanjutkan	Sistem menghapus data jika permintaan dilanjutkan dan menampilkan pesan berhasil hapus data	<i>Valid</i>
5	Menambahkan Order Masuk	<i>User</i> menekan tombol Tambah Order Masuk dalam yang terdapat di dalam fitur	Sistem akan menampilkan <i>form</i> untuk <i>input</i> data <i>order</i> masuk	Data berhasil disimpan ke dalam <i>database</i> dan sistem menampilkan data dalam fitur <i>order</i> masuk	<i>Valid</i>

Tabel 6.7 Pengujian validasi kebutuhan fungsional (lanjutan)

No	Nama Kebutuhan	Kasus Uji	Hasil yang diharapkan	Hasil yang diberikan	Status
6	Mengubah <i>Order</i> Masuk	<i>User</i> melakukan perubahan data <i>Order</i> masuk sesuai dengan form yang ditampilkan	Data yang diubah akan disimpan dalam <i>database</i>	Data berhasil dirubah dan disimpan kedalam <i>database</i>	Valid
7	Menghapus <i>Order</i> Masuk	<i>User</i> menekan tombol hapus pada tabel daftar <i>Order</i> Masuk	Sistem akan menghapus data yang dipilih oleh <i>user</i> dalam fitur <i>Order</i> Masuk	Sistem menghapus data <i>order</i> masuk sesuai yang telah dipilih oleh <i>user</i>	Valid
8	Menambahkan Pengiriman	<i>User</i> menekan tombol Tambah Pengiriman dalam yang terdapat di dalam fitur	Sistem akan menampilkan form untuk <i>input</i> data Pengiriman	Data berhasil disimpan ke dalam <i>database</i> dan sistem menampilkan data dalam fitur Pengiriman	Valid
9	Mengubah Pengiriman	<i>User</i> melakukan perubahan data pengiriman sesuai dengan form yang ditampilkan	Data yang diubah akan disimpan dalam <i>database</i>	Data <i>input</i> berhasil disimpan dalam <i>database</i>	Valid
10	Menghapus Pengiriman	<i>User</i> menekan tombol Hapus pada tabel daftar pengiriman	Sistem akan menampilkan peringatan dan menghapus data jika permintaan dilanjutkan	Sistem menghapus data jika permintaan dilanjutkan dan menampilkan pesan berhasil hapus data	Valid
11	Menambahkan Pembelian barang	<i>User</i> menekan tombol input Pembelian dalam yang terdapat di dalam fitur stok gudang	Sistem akan menampilkan form untuk <i>input</i> data Pembelian barang	Data berhasil disimpan ke dalam <i>database</i> dan sistem menampilkan pesan berhasil	Valid

Tabel 6.7 Pengujian validasi kebutuhan fungsional (lanjutan)

No	Nama Kebutuhan	Kasus Uji	Hasil yang diharapkan	Hasil yang diberikan	Status
12	Menambahkan stok barang	<i>User</i> menekan tombol input Pembelian barang dalam yang terdapat di dalam fitur stok barang	Sistem akan menampilkan <i>form</i> untuk <i>input</i> data stok barang	Data berhasil disimpan ke dalam <i>database</i> dan sistem menampilkan pesan berhasil	Valid
13	Menambahkan <i>Supplier</i>	<i>User</i> menekan tombol Tambah <i>Supplier</i> dalam yang terdapat di dalam fitur	Sistem akan menampilkan <i>pop-up form</i> untuk <i>input</i> data <i>Supplier</i>	Data berhasil disimpan ke dalam <i>database</i> dan sistem menampilkan data dalam fitur <i>Supplier</i>	Valid
14	Mengubah <i>Supplier</i>	<i>User</i> melakukan perubahan data <i>Supplier</i> sesuai dengan form yang ditampilkan	Data yang diubah akan disimpan dalam <i>database</i>	Data berhasil dirubah dan disimpan kedalam <i>database</i>	Valid
15	Menghapus <i>Supplier</i>	<i>User</i> menekan tombol Hapus pada tabel daftar <i>Supplier</i>	Sistem akan menampilkan peringatan dan menghapus data jika permintaan dilanjutkan	Sistem menghapus data jika permintaan dilanjutkan dan menampilkan pesan berhasil hapus data	Valid
16	Menambahkan Pelanggan	<i>User</i> menekan tombol Tambah Pelanggan dalam yang terdapat di dalam fitur	Sistem akan menampilkan <i>pop-up form</i> untuk <i>input</i> data Pelanggan	Data berhasil disimpan ke dalam <i>database</i> dan sistem menampilkan data dalam fitur Pelanggan	Valid
17	Mengubah Pelanggan	<i>User</i> melakukan perubahan data Pelanggan sesuai dengan form yang ditampilkan	Data yang diubah disimpan dalam <i>database</i>	Data berhasil dirubah dan disimpan kedalam <i>database</i>	Valid

Tabel 6.7 Pengujian validasi kebutuhan fungsional (lanjutan)

No	Nama Kebutuhan	Kasus Uji	Hasil yang diharapkan	Hasil yang diberikan	Status
18	Menghapus Pelanggan	<i>User</i> menekan tombol hapus pada tabel daftar Pelanggan	Sistem akan menghapus data yang dipilih oleh <i>user</i> dalam tabel Pelanggan	Sistem menghapus data Pelanggan sesuai yang telah dipilih oleh <i>user</i>	Valid
19	<i>Logout</i>	<i>User</i> memilih menu <i>logout</i>	Sistem menampilkan peringatan, jika <i>user</i> memilih OK maka <i>user</i> keluar dari sistem	Sistem mematikan <i>session</i> jika <i>user</i> yakin ingin keluar dari sistem	Valid
20	Mengubah Penawaran	<i>User</i> melakukan perubahan data Pelanggan sesuai dengan form yang ditampilkan	Data yang diubah disimpan dalam <i>database</i>	Data berhasil dirubah dan disimpan kedalam <i>database</i>	Valid
21	Menambahkan <i>User</i>	<i>User</i> melakukan penambahan untuk pengguna baru	<i>User</i> baru tersimpan dalam sistem	<i>User</i> baru dapat <i>login</i> kedalam sistem	Valid
22	Melihat stok gudang	<i>User</i> menekan menu list stok gudang	Sistem menampilkan list barang	Data barang berhasil ditampilkan	Valid
23	Melihat penawaran	<i>User</i> menekan tombol list penawaran	Sistem menampilkan list penawaran	Data penawaran berhasil ditampilkan	Valid
24	Melihat order masuk	<i>User</i> menekan tombol list order masuk	Sistem menampilkan list order masuk	Data order masuk berhasil ditampilkan	Valid
25	Melihat pengiriman	<i>User</i> menekan tombol list pengiriman	Sistem menampilkan list pengiriman	Data pengiriman berhasil ditampilkan	Valid
26	Melihat pelanggan	<i>User</i> menekan tombol list pelanggan	Sistem menampilkan list pelanggan	Data pelanggan berhasil ditampilkan	Valid

Tabel 6.7 Pengujian validasi kebutuhan fungsional (lanjutan)

No	Nama Kebutuhan	Kasus Uji	Hasil yang diharapkan	Hasil yang diberikan	Status
27	Melihat <i>supplier</i>	<i>User</i> menekan tombol hapus pada tabel daftar Pelanggan	Sistem akan menghapus data yang dipilih oleh <i>user</i> dalam tabel Pelanggan	Sistem menghapus data Pelanggan sesuai yang telah dipilih oleh <i>user</i>	<i>Valid</i>
28	Melihat laporan <i>order</i> masuk	<i>User</i> menekan tombol tampil laporan	Sistem menampilkan data laporan <i>order</i> masuk	Sistem berhasil menampilkan laporan <i>order</i> masuk	<i>Valid</i>
29	Melihat laporan pengiriman	<i>User</i> menekan tombol tampil laporan	Sistem menampilkan data laporan pengiriman	Sistem berhasil menampilkan data laporan pengiriman	<i>Valid</i>
30	Melihat laporan pembelian	<i>User</i> menekan tombol tampil laporan pembelian	Sistem menampilkan data laporan laporan	Sistem berhasil menampilkan data laporan pembelian	<i>Valid</i>

Pada tabel 6.7 di atas merupakan pengujian validasi menggunakan kasus uji untuk menguji sistem berjalan dengan baik menurut proses *input* dan *output*. Total pengujian yang dicantumkan pada tabel di atas didapatkan sebanyak 30 kali dan mendapatkan status *valid*.

6.2.2 Pengujian Kebutuhan Non-Fungsional

Pengujian non-fungsional dilakukan untuk mengetahui hasil dari pengujian non-fungsional yang telah dilakukan. Untuk pengujian non-fungsional ini menggunakan pengujian *security* dan *performance*. Berikut detail pengujian yang terdapat pada tabel 6.8.

Tabel 6.8 Pengujian kebutuhan non-fungsional

No	Kebutuhan Non-Fungsional	Kasus Uji	Hasil yang diharapkan	Hasil yang diberikan	Status
1	<i>Security</i>	Menambah data “Nama : Reza , <i>Username</i> : Admin, <i>Password</i> : Admin	Data <i>password</i> yang telah dimasukkan dan dienkripsi	Sistem berhasil menyimpan <i>password</i> dalam <i>database</i> yang telah dienkripsi dengan metode <i>hashing MD 5</i>	<i>Valid</i>
2	<i>Performance</i>	1. Memasukkan data pada fitur penawaran 2. Menampilkan fitur list penawaran 3. Memasukkan data pada fitur pembelian barang 4. Menampilkan fitur list stok barang 5. Memasukkan data pada fitur order masuk	Sistem memberikan waktu respon maksimal 1000 <i>milisecond</i> Sistem memberikan waktu respon maksimal 1000 <i>milisecond</i>	Sistem berhasil memberikan respon 171 <i>milisecond</i> Sistem berhasil memberikan respon 185 <i>milisecond</i> Sistem berhasil memberikan respon 236 <i>milisecond</i> Sistem berhasil memberikan respon 338 <i>milisecond</i> Sistem berhasil memberikan respon 230 <i>milisecond</i>	<i>Valid</i> <i>Valid</i> <i>Valid</i> <i>Valid</i> <i>Valid</i>

Tabel 6.8 adalah hasil pengujian *black-box* pada kebutuhan non-fungsional. Untuk melihat pengujian *security* yang menggunakan metode *hashing* MD5 pada pengujian non-fungsional dapat dilihat di gambar berikut.

1. Pengujian *Security*

Berikut adalah *screenshot* hasil dari metode *hash MD5* yang tersimpan dalam *database mitrateknik_db*

mitrateknik_db

	New	Show all	Number of rows:	25	Filter rows:	Search this table
<input checked="" type="checkbox"/>	admin	<input type="checkbox"/>				
<input checked="" type="checkbox"/>	barang_gudang	<input type="checkbox"/>				
<input checked="" type="checkbox"/>	log_tabel	<input type="checkbox"/>				
<input checked="" type="checkbox"/>	order_keluar	<input type="checkbox"/>				
<input checked="" type="checkbox"/>	order_masuk	<input type="checkbox"/>				
<input checked="" type="checkbox"/>	pelanggan	<input type="checkbox"/>				
<input checked="" type="checkbox"/>	penawaran	<input type="checkbox"/>				
<input checked="" type="checkbox"/>	suplier	<input type="checkbox"/>				

+ Options

	id	nama	username	password
<input type="checkbox"/>	1	reza	admin	21232f297a57a5a743894a0e4a801fc3

Edit Copy Delete Check all With selected: Edit Copy Delete Export

Gambar 6.4 Hasil Pengujian *Security*

2. Pengujian *Performance*

Untuk melihat pengujian non-fungsional yang sudah di jelaskan pada tabel 6.8. Pada sisi *performance* dapat diketahui hasilnya dalam *inspect element* yang ada dalam *browser*. Berikut adalah *screenshot* hasil dari pengujian non-fungsional untuk sisi *performance*. Hasil pengujian dapat dilihat pada gambar 6.5.

Status	Method	Domain	File	Cause	Type	Transferred	Size	0 ms	159 ms
200	GET	localhost	form_penawaran	document	html	30.18 KB	29.82 KB	<div style="width: 100%;"> </div>	<div style="width: 100%; background-color: #2e7131; color: white;">171 ms</div>
Status	Method	Domain	File	Cause	Type	Transferred	Size	0 ms	171 ms
200	GET	localhost	list_penawaran	document	html	81.45 KB	81.09 KB	<div style="width: 100%;"> </div>	<div style="width: 100%; background-color: #2e7131; color: white;">185 ms</div>
Status	Method	Domain	File	Cause	Type	Transferred	Size	0 ms	186 ms
200	GET	localhost	form_pembelian	document	html	30.11 KB	29.75 KB	<div style="width: 100%;"> </div>	<div style="width: 100%; background-color: #2e7131; color: white;">236 ms</div>
Status	Method	Domain	File	Cause	Type	Transferred	Size	0 ms	190 ms
200	GET	localhost	list_pembelian	document	html	122.35 KB	121.99 KB	<div style="width: 100%;"> </div>	<div style="width: 100%; background-color: #2e7131; color: white;">338 ms</div>
Status	Method	Domain	File	Cause	Type	Transferred	Size	0 ms	339 ms
200	GET	localhost	form_order_masuk	document	html	30.84 KB	30.48 KB	<div style="width: 100%;"> </div>	<div style="width: 100%; background-color: #2e7131; color: white;">230 ms</div>

Gambar 6.5 Hasil Pengujian *Performance*

6.3 Pengujian *User Acceptance Testing* (UAT)

Pada tahap pengujian selanjutnya yaitu pengujian *User Acceptance Testing* (UAT). Pengujian ini dilakukan untuk memastikan bahwa sistem yang dibuat dapat dipahami oleh pengguna. Pengujian dilakukan dengan cara *user* menjalankan sistem dan menguji beberapa fungsi yang ada dalam sistem. Untuk kriteria penilaian dilakukan sesuai dengan langkah-langkah yang disusun oleh penulis yang ditujukan langsung kepada validator

Berikut analisis pengujian yang berdasarkan dari *form* kuisioner *User Acceptance Testing* dari total 4 validator.

Keterangan tabel :

SA = *Strongly Agree*, **A** = *Agree*, **N** = *Neutral*, **D** = *Disagree*, **SD**= *Strongly Disagree*.

Tabel 6.9 Data tabel penilaian untuk pengguna

No	Case / Proses	Penilaian					Total
		SAx5	Ax4	Nx3	Dx2	SDx1	
1	Sistem informasi penjualan dinilai dapat membantu kinerja pengguna sistem menjadi lebih efektif	10	8	-	-	-	18
2	Sistem ini membantu pengguna menghemat waktu dalam proses penyimpanan data transaksi	5	12	-	-	-	17
3	Sistem mudah untuk digunakan	10	8	-	-	-	18
4	Fitur dalam sistem yang mudah dipahami dan <i>user friendly</i>	10	8	-	-	-	18
5	Pengguna senang menggunakan sistem penjualan ini karena mudah dan cepat untuk melakukan <i>input</i> atau mencari data transaksi yang telah disimpan	5	8	3	-	-	16

Tabel 6.10 Data tabel penilaian fitur penawaran

No	Case / Proses	Penilaian					Total
		SAx5	Ax4	Nx3	Dx2	SDx1	
1	Tombol untuk akses fitur penawaran	15	4	-	-	-	19
2	Kemudahan proses memasukkan data dalam <i>form</i> penawaran	5	12	-	-	-	17
3	Tampilan tabel data penawaran menarik	5	8	3	-	-	16
4	Terdapat tombol <i>edit</i> , dan <i>hapus</i> untuk mengelola data penawaran		12	3	-	-	15
5	Tombol cetak untuk mencetak laporan penawaran sesuai dengan data yang dipilih	15	4	-	-	-	19

Tabel 6.11 Data tabel penilaian fitur *order* masuk

No	Case / Proses	Penilaian					Total
		SAx5	Ax4	Nx3	Dx2	SDx1	
1	Tombol untuk akses fitur <i>order</i> masuk	5	12	-	-	-	17
2	Kemudahan proses memasukkan data dalam <i>form order</i> masuk	5	8	1	-	-	14
3	Tampilan tabel data <i>order</i> masuk menarik	10	8	-	-	-	18
4	Terdapat tombol <i>edit</i> , dan <i>hapus</i> untuk mengelola data <i>order</i> masuk	10	8	-	-	-	18
5	Tombol detail untuk melihat detail data <i>order</i> masuk yang telah dimasukkan oleh pengguna	15	4	-	-	-	19
6	Terdapat tombol tampil laporan untuk menampilkan laporan <i>order</i> masuk	10	8	-	-	-	18

Tabel 6.12 Data tabel penilaian fitur pengiriman

No	Case / Proses	Penilaian					Total
		SAx5	Ax4	Nx3	Dx2	SDx1	
1	Tombol untuk akses fitur pengiriman	5	12	-	-	-	17
2	Kemudahan proses memasukkan data dalam <i>form pengiriman</i>	10	8	-	-	-	18
3	Tampilan tabel data pengiriman menarik	10	8	-	-	-	18
4	Terdapat tombol <i>edit</i> , dan <i>hapus</i> untuk mengelola data pengiriman	10	8	-	-	-	18
5	Tombol detail untuk melihat detail data pengiriman yang telah dimasukkan oleh pengguna	5	12	-	-	-	17
6	Terdapat tombol tampil laporan untuk menampilkan laporan pengiriman	5	8	3	-	-	16

Tabel 6.13 Data tabel penilaian fitur pembelian

No	Case / Proses	Penilaian					Total
		SAx5	Ax4	Nx3	Dx2	SDx1	
1	Tombol untuk akses fitur pembelian	-	12	3	-	-	15
2	Kemudahan proses memasukkan data dalam form pembelian	5	8	1	-	-	14
3	Tampilan tabel data pembelian menarik	10	8	-	-	-	18
4	Terdapat tombol <i>edit</i> , dan hapus untuk mengelola data pembelian	5	12	-	-	-	17
5	Tombol detail untuk melihat detail data pembelian yang telah dimasukkan oleh pengguna	10	8	-	-	-	18
6	Terdapat tombol tampil laporan untuk menampilkan laporan pembelian	10	8	-	-	-	18

Tabel 6.14 Data tabel penilaian fitur pelanggan

No	Case / Proses	Penilaian					Total
		SAx5	Ax4	Nx3	Dx2	SDx1	
1	Tombol untuk akses fitur pelanggan	10	8	-	-	-	18
2	Kemudahan proses memasukkan data pelanggan dalam bentuk <i>form pop-up</i>	-	16	-	-	-	16
3	Tampilan tabel data pelanggan menarik	5	8	3	-	-	16
4	Terdapat tombol <i>edit</i> , dan hapus untuk mengelola data pelanggan	10	8	-	-	-	18

Tabel 6.15 Data tabel penilaian fitur *supplier*

No	Case / Proses	Penilaian					Total
		SAx5	Ax4	Nx3	Dx2	SDx1	
1	Tombol untuk akses fitur <i>supplier</i>	5	12	-	-	-	17
2	Kemudahan proses memasukkan data <i>supplier</i> dalam bentuk <i>form pop-up</i>	10	8	-	-	-	18
3	Tampilan tabel data <i>supplier</i> menarik	5	12	-	-	-	17
4	Terdapat tombol <i>edit</i> , dan <i>hapus</i> untuk mengelola data <i>supplier</i>	5	12	-	-	-	17

Tabel 6.16 Data tabel penilaian fitur *login*

No	Case / Proses	Penilaian					Total
		SAx5	Ax4	Nx3	Dx2	SDx1	
1	Fitur <i>login</i> untuk proses masuk ke dalam sistem	10	8	-	-	-	18
2	Muncul pesan <i>error</i> ketika salah memasukkan <i>username</i> dan <i>password</i>	10	8	-	-	-	18
3	Tampilan fitur <i>login</i> sistem menarik	10	8	-	-	-	18

1. Analisis dari tabel penilaian pengguna

Keterangan : Nilai rata-rata = Nilai / Keterangan Penilaian

Persentase = Nilai rata-rata / validator x 100

Tabel 6.17 Penilaian pengguna

No	Jumlah Validator	Nilai rata-rata	Persentase
1	4	18/5=3.6	3.6/4 x 100 = 90%
2	4	18/5=3.4	3.4/4 x 100 = 85%
3	4	18/5=3.6	3.6/4 x 100 = 90%
4	4	18/5=3.6	3.6/4 x 100 = 85%
5	4	16/5=3.2	3.4/4 x 100 = 80%

2. Analisis dari tabel fitur penawaran

Tabel 6.18 Penilaian fitur penawaran

No	Jumlah Validator	Nilai rata-rata	Persentase
1	4	18/5=3.8	3.8/4 x 100 = 95%
2	4	17/5=3.4	3.8/4 x 100 = 95%
3	4	16/5=3.2	3.2/4 x 100 = 80%
4	4	15/5=3	3/4 x 100 = 75%
5	4	16/5=3.2	3.4/4 x 100 = 80%

3. Analisis dari tabel fitur order masuk.

Tabel 6.19 Penilaian fitur order masuk

No	Jumlah Validator	Nilai rata-rata	Persentase
1	4	17/5=3.4	3.4/4 x 100 = 85%
2	4	16/5=3.2	3.2/4 x 100 = 80%
3	4	18/5=3.6	3.6/4 x 100 = 90%
4	4	18/5=3.6	3.6/4 x 100 = 90%
5	4	19/5=3.8	3.8/4 x 100 = 95%
6	4	18/5=3.6	3.6/4 x 100 = 90%

4. Analisis dari tabel fitur pengiriman.

Tabel 6.20 Penilaian fitur pengiriman

No	Jumlah Validator	Nilai rata-rata	Persentase
1	4	17/5=3.4	3.4/4 x 100 = 85%
2	4	18/5=3.6	3.6/4 x 100 = 90%
3	4	18/5=3.6	3.6/4 x 100 = 90%
4	4	18/5=3.6	3.6/4 x 100 = 90%
5	4	17/5=3.4	3.4/4 x 100 = 85%
6	4	18/5=3.2	3.2/4 x 100 = 80%

5. Analisis dari tabel fitur pembelian.

Tabel 6.21 Penilaian fitur pembelian

No	Jumlah Validator	Nilai rata-rata	Persentase
1	4	15/5=3	$3/4 \times 100 = 75\%$
2	4	16/5=3.2	$3.2/4 \times 100 = 80\%$
3	4	18/5=3.6	$3.6/4 \times 100 = 90\%$
4	4	17/5=3.4	$3.4/4 \times 100 = 85\%$
5	4	18/5=3.6	$3.6/4 \times 100 = 90\%$
6	4	18/5=3.6	$3.6/4 \times 100 = 90\%$

6. Analisis dari tabel fitur pelanggan

Tabel 6.22 Penilaian fitur pelanggan

No	Jumlah Validator	Nilai rata-rata	Persentase
1	4	18/5=3.6	$3.6/4 \times 100 = 90\%$
2	4	16/5=3.2	$3.2/4 \times 100 = 80\%$
3	4	16/5=3.2	$3.2/4 \times 100 = 80\%$
4	4	18/5=3.6	$3.6/4 \times 100 = 90\%$

7. Analisis dari tabel fitur *Supplier*

Tabel 6.23 Penilaian fitur *supplier*

No	Jumlah Validator	Nilai rata-rata	Persentase
1	4	17/5=3.4	$3.4/4 \times 100 = 85\%$
2	4	18/5=3.6	$3.6/4 \times 100 = 90\%$
3	4	17/5=3.4	$3.4/4 \times 100 = 85\%$
4	4	17/5=3.4	$3.4/4 \times 100 = 95\%$

8. Analisis dari tabel fitur *login*

Tabel 6.24 Penilaian fitur *login*

No	Jumlah Validator	Nilai rata-rata	Persentase
1	4	18/5=3.6	3.6/4 x 100 = 90%
2	4	18/5=3.6	3.6/4 x 100 = 90%
3	4	18/5=3.6	3.6/4 x 100 = 90%

Berdasarkan tabel diatas menunjukkan bahwa sistem informasi penjualan ini sudah bisa dijalankan dengan baik. Dapat dilihat dari penilaian yang diberikan kepada validator yang menunjukkan skala penilaian (%) di angka 75% – 100% yang menunjukkan skala jawaban bernilai *agree* – *strongly agree* yang artinya sistem mudah digunakan dan dapat diterima oleh pengguna. Untuk kriteria penilaian dilakukan sesuai dengan langkah-langkah yang disusun oleh penulis yang ditujukan langsung kepada validator.

6.4 Analisis Pengujian

Berdasarkan hasil pengujian *white-box* pada pengujian unit atau *cyclomatic complexity* didapatkan pengujian tiga fungsi yaitu tambah_order_masuk, Edit_order_masuk, dan Hapus_order_masuk. Dari pengujian yang dilakukan pada semua jalur independen yang diuji mendapat tingkat keberhasilan pengujian *white-box* 100% dan dinyatakan *valid*.

Sedangkan pengujian *black-box* menggunakan kasus uji pada fungsional yang terdapat pada tabel 6.5 yang terdapat 30 total kasus yang diuji dan tidak terdapat kesalahan maka dipastikan mendapat data *valid*, dapat dilihat dari perhitungan berikut.

$$\frac{30}{30} \times 100\% = 100\%$$

Berdasarkan rata-rata presentase keberhasilan pengujian fungsional yang didapatkan adalah lebih dari 90%, maka setelah dilakukan konversi ke dalam predikat seperti pada tabel 2.1 maka hasil pengujian fungsional sistem dapat dinyatakan dalam predikat sangat layak.

Untuk pengujian *User Acceptance Testing* (UAT) didapatkan rata-rata nilai diatas 75% diskala *agree*- *strongly agree* dari 4 validator yang menguji sistem menurut tabel skala penilaian 2.2 maka hasil pengujian sistem dinyatakan layak dan sudah berjalan dengan baik.

BAB 7 PENUTUP

7.1 Kesimpulan

Kesimpulan yang diperoleh dari hasil penelitian “Pembangunan Sistem Informasi Penjualan Pada CV. Mitra Teknik Berbasis Web” yang dapat ditarik kesimpulan sebagai berikut.

1. Dari hasil analisis kebutuhan, terdapat 30 kebutuhan fungsional. Dari 30 yang dihasilkan oleh kebutuhan fungsional terdapat 8 kebutuhan fungsional utama, yakni kebutuhan fungsional untuk Melihat Data Penawaran, Melihat Data Pengiriman, Melihat data stok barang, Melihat Data *Order* Masuk, Melihat Data *Supplier*, Melihat Data Pelanggan, dan Menambahkan stok barang yang digunakan untuk sistem manajemen penjualan CV. Mitra Teknik.
2. Dari hasil perancangan, menghasilkan 22 *class* yang terdapat dalam komponen *Model*, *View*, dan *Controller*. 8 tabel dari hasil perancangan basis data, dan perancangan antarmuka menjadi acuan untuk membuat antarmuka sistem, Pada tahap implementasi menghasilkan beberapa fitur penunjang untuk sistem manajemen penjualan.
3. Proses pengujian yang dilakukan pada pembangunan sistem informasi yaitu menggunakan pengujian *white-box testing* dan *black-box testing* dalam sistem informasi penjualan tidak ditemukan masalah pada setiap fungsi sistem. Untuk pengujian *User Acceptance Testing* (UAT) terdapat dua validator untuk menguji kinerja sistem, dari hasil pengujian yang dilakukan oleh validator didapatkan 8 tabel proses *testing* dan menunjukkan skala penilaian (%) di angka lebih dari 75% yang menyatakan sistem sudah mampu untuk dipahami dan digunakan oleh pengguna.

7.2 Saran

Adapun saran-saran dalam penelitian berikut:

1. Pembangunan sistem informasi penjualan ini berbasis *website* dan menggunakan teknologi *web service*. *Web service* yang diterapkan dapat dimanfaatkan untuk pembangunan sistem secara *multi-platform*.
2. Sistem yang dibangun bertujuan untuk memenuhi kebutuhan yang bersinggungan langsung dengan transaksi penjualan dan diharapkan sistem ini dapat dikembangkan dan bisa dimanfaatkan oleh bagian lain seperti contoh merekap data keuangan, manajemen ekspor-impor, dan manajemen karyawan perusahaan secara lengkap dan detail.

DAFTAR REFERENSI

- Abdullah, Thamrin & Tantri F., 2016. *Manajemen Pemasaran*. Depok: PT. Raja Grafindo Persada.
- Alexander F.K. Sibero. 2013. *Web programming power pack*. Yogyakarta: MediaKom.
- Bambang, Hariyanto. 2004. *Sistem Manajemen Basis Data: Pemodelan, Perancangan, dan Terapannya*. Bandung: Informatika Bandung.
- Basu, Swastha. 2001. *Manajemen Pemasaran Modern*. Yogyakarta: BPFE.
- BPMN, 2008. *The Business Process Modeling Notation; Poket Handbook by Briol Patrice*, First Edition ISBN. 978-1-4092-0299-8.
- Brittenham, Peter. 2002. *An overview of the Web Services Inspection Language*, [Online] Tersedia di: <<HTTP://www.ibm.com/developerworks/library/ws-wsilover/>>. [diakses 10 Mei 2018].
- Burbeck, Steven. 1992. *Application Programmings in Smalltalk's 80™: How To Use MVC*, (Online). <HTTPPs://www.researchgate.net/publication/238719652_Applications_programming_in_smalltalk-80_how_to_use_model-view-controller_mvc>. [diakses 21 Mei 2018].
- Chairul, Marom. 2002. *Sistem Akuntansi Perusahaan Dagang*, Jakarta: Penerbit Grasindo, Edisi ke-2.
- Harjunawati, Sri. 2016. *Sistem Informasi Akuntansi Penjualan Berbasis Waterfall Model Untuk Perusahaan Dagang*, Jakarta: AMIK BSI, Vol 1 No. 2,
- Jogiyanto, H.M., 2005. *Analisa dan Desain Sistem Informasi Pendekatan Terstruktur Teori dan Praktik Aplikasi Bisnis*. Yogyakarta: ANDI.
- Kreger, H., 2001. *Web Service Conceptual Architecture*. IBM Software Group.
- Lereng, S. F., & Fugelseth, L., 2004. *Testing of Web-based Systems*. Norwegia: NTNU.
- Manes, A.T., 2001. *Introduction to Web-services*. [Online] Tersedia di: <<HTTP://www.systinet.com>> [Diakses 20 Juli 2018].
- Munir, Rinaldi. 2006. *Diktat Kuliah IF2153 Matematika Diskrit*. Program Studi Teknik Informatika, Institut Teknologi Bandung.
- Nielsen, J., 2010. *Website Response Times*. [Online] Tersedia di <<HTTPPs://www.nngroup.com/articles/web site-response-times/>> [Diakses 13 Juli 2018]
- Pressman R. S. dan Maxim, B. R., 2015. *Software Engineering A Practitioner's Approach Eighth Edition*. New York: McGraw-Hill Education.

- Putra, Rizqi A. S., 2016. *Aplikasi Sistem Pakar Untuk Membantu Guru Dalam Memilih Jenis Media Pembelajaran*. Yogyakarta: Universitas Negeri Yogyakarta.
- Raharjo, Budi, dkk., 2012. *Modul Pemrograman Web HTML, PHP & MySQL*, Bandung: Modula.
- Richardus, Eko Indrajit. 2014. *Manajemen dan Organisasi Tata Kelola Teknologi Informasi*: Aptikom.
- Risnandar, 2010. *Integrasi Teknologi Informasi dan Supply Chain Management (Studi Kasus : PT X, West Java)*. Bandung: Politeknik Telkom.
- Ristono, Agus. 2009. *Manajemen persediaan edisi 1*. Yogyakarta : Graha Ilmu
- Rosa A.S & Salahuddin M., 2011. *Modul Pembelajaran Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Modula.
- Salahuddin. M., 2011. *Modul pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Modula.
- Siswoutomo, Wiwit. 2004. *Intermediate Accounting*. Terbitan: Elex Media Komputindo.
- Sommerville. I., 2011. *Software engineering Ninth Edition*. Massachusetts: Pearson Education.
- Sunarto. 2018. *Alur Proses Bisnis Perusahaan CV. Mitra Teknik*. Diwawancara oleh Mohamad Reza Mawardi. Sidoarjo, 23 Desember 2018.
- Susanto, F. N. R. d. A., 2017. *Implementasi RESTful Web Service untuk Sistem Penghitungan Suara Secara Cepat pada Pilkada*. EKSPLORA INFORMATIKA, Volume 6, pp. 159-168.
- Qomaruddin M., 2014. *User Acceptance Test of Computer-Assisted Problem-Based Learning Assessment Tool (CAPBLAT)*. Journal of Education and Learning. Vol.8 (1) 71-77.
- Wagner, Tony. 2008. *The Global Achievement Gap*. New York: Basic Books.
- Winardi, 2001. *Motivasi dan Pemotivasi dalam Manajemen*. Jakarta: Raja Grafindo Persada.

LAMPIRAN Kuisioner *User Acceptance Testing (UAT)*

Kuisioner *User Acceptance Testing (UAT)*

<i>Company Name</i>	: CV. Mitra Teknik
<i>Consultan Name</i>	: -
<i>Program Name</i>	: Sistem Informasi Penjualan
<i>Procces Name</i>	: Testing
<i>Version Acces</i>	: Web
<i>User category</i>	: All user
<i>UAT Date</i>	: 02 Juli 2014
<i>Description Procces</i>	: Pengujian pada sistem ketika digunakan oleh pengguna
<i>Validator</i>	: Sunarto

Description :

- SA = *Strongly Aggre*
- A = *Aggre*
- N = *Neutral*
- D = *Disagree*
- SD = *Strongly Disagree*

Tabel Kuisioner

No	Case / Proses	Penilaian				
		SA	A	N	D	SD
1	Sistem informasi penjualan dinilai dapat membantu kinerja pengguna sistem menjadi lebih efektif	✓				
2	Sistem ini membantu pengguna menghemat waktu dalam proses penyimpanan data transaksi	✓				
3	Sistem mudah untuk digunakan		✓			
4	Fitur dalam sistem yang mudah dipahami dan <i>user friendly</i>		✓			
5	Pengguna senang menggunakan sistem penjualan ini karena mudah dan cepat untuk melakukan <i>input</i> atau mencari data transaksi yang telah disimpan		✓			

LAMPIRAN

Tabel kuisioner fitur Pelanggan

No	Case / Proses	Penilaian				
		SA	A	N	D	SD
1	Tombol untuk akses fitur pelanggan		✓			
2	Kemudahan proses memasukkan data pelanggan dalam bentuk <i>form pop-up</i>	✓		*		
3	Tampilan tabel data pelanggan menarik			✓		
4	Terdapat tombol <i>edit</i> , dan <i>hapus</i> untuk mengelola data pelanggan		✓			

Tabel kuisioner fitur *Supplier*

No	Case / Proses	Penilaian				
		SA	A	N	D	SD
1	Tombol untuk akses fitur <i>supplier</i>		✓			
2	Kemudahan proses memasukkan data <i>supplier</i> dalam bentuk <i>form pop-up</i>		✓			
3	Tampilan tabel data <i>supplier</i> menarik		✓			
4	Terdapat tombol <i>edit</i> , dan <i>hapus</i> untuk mengelola data <i>supplier</i>		✓			

Tabel kuisioner fitur *Login*

No	Case / Proses	Penilaian				
		SA	A	N	D	SD
1	Fitur <i>login</i> untuk proses masuk ke dalam sistem	✓				
2	Muncul pesan <i>error</i> ketika salah memasukkan <i>username</i> dan <i>password</i>		✓			
3	Tampilan fitur <i>login</i> sistem menarik		✓			

LAMPIRAN

Tabel kuisioner fitur Pengiriman

No	Case / Proses	Penilaian				
		SA	A	N	D	SD
1	Tombol untuk akses fitur pengiriman	✓				
2	Kemudahan proses memasukkan data dalam form pengiriman		✓	•		
3	Tampilan tabel data pengiriman menarik		✓			
4	Terdapat tombol edit, dan hapus untuk mengelola data pengiriman		✓			
5	Tombol detail untuk melihat detail data pengiriman yang telah dimasukkan oleh pengguna	✓				
6	Terdapat tombol tampil laporan untuk menampilkan laporan pengiriman	✓				

Tabel kuisioner fitur Pembelian

No	Case / Proses	Penilaian				
		SA	A	N	D	SD
1	Tombol untuk akses fitur pembelian		✓			
2	Kemudahan proses memasukkan data dalam form pembelian		✓			
3	Tampilan tabel data pembelian menarik		✓			
4	Terdapat tombol edit, dan hapus untuk mengelola data pembelian		✓			
5	Tombol detail untuk melihat detail data pembelian yang telah dimasukkan oleh pengguna	✓				
6	Terdapat tombol tampil laporan untuk menampilkan laporan pembelian	✓				

LAMPIRAN

Tabel kuisioner fitur Penawaran

No	Case / Proses	Penilaian				
		SA	A	N	D	SD
1	Tombol untuk akses fitur penawaran	✓				
2	Kemudahan proses memasukkan data dalam <i>form</i> penawaran		✓	*		
3	Tampilan tabel data penawaran menarik		✓			
4	Terdapat tombol <i>edit</i> , dan hapus untuk mengelola data penawaran			✓		
5	Tombol cetak untuk mencetak laporan penawaran sesuai dengan data yang dipilih	✓				

Tabel kuisioner fitur Order Masuk

No	Case / Proses	Penilaian				
		SA	A	N	D	SD
1	Tombol untuk akses fitur <i>order masuk</i>	✓				
2	Kemudahan proses memasukkan data dalam <i>form</i> <i>order masuk</i>		✓			
3	Tampilan tabel data <i>order masuk</i> menarik		✓			
4	Terdapat tombol <i>edit</i> , dan hapus untuk mengelola data <i>order masuk</i>		✓			
5	Tombol detail untuk melihat detail data <i>order masuk</i> yang telah dimasukkan oleh pengguna		✓			
6	Terdapat tombol tampil laporan untuk menampilkan laporan <i>order masuk</i>	✓				