

**PENGEMBANGAN SISTEM MANAJEMEN LAYANAN JASA
SERVIS ALAT ELEKTRONIK RUMAH TANGGA
(STUDI KASUS: UKM BALI TEKINDO JAYA)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Ni Luh Irma Arini

NIM: 155150200111112



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

PENGESAHAN

PENGEMBANGAN SISTEM MANAJEMEN LAYANAN JASA SERVIS ALAT ELEKTRONIK
RUMAH TANGGA (STUDI KASUS: UKM BALI TEKINDO JAYA)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Ni Luh Irma Arini
NIM: 155150200111112

Skripsi ini telah diuji dan dinyatakan lulus pada
4 Juli 2019

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Bayu Priyambadha, S.Kom. M.Kom.
NIP: 19820909 200812 1 004

Dosen Pembimbing 2



Ir. Heru Nurwasito, M.Kom.
NIP: 19650402 199002 1 001

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 referensi.

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, 20 Juni 2019



Ni Luh Irma Arini

NIM: 155150200111112

PRAKATA

Puji syukur penulis panjatkan kehadiran Tuhan Yang Maha Esa, Ida Sang Hyang Widhi Wasa, yang telah melimpahkan asung kertha wara nugraha-Nya sehingga dokumen skripsi yang berjudul “Pengembangan Sistem Manajemen Layanan Jasa Servis Alat Elektronik Rumah Tangga” ini dapat terselesaikan.

Penulis menyadari bahwa laporan ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Bayu Priyambadha, S.Kom., M.Kom., selaku dosen pembimbing pertama skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan laporan ini.
2. Bapak Ir. Heru Nurwasito, M.Kom., selaku dosen pembimbing kedua skripsi yang telah membimbing terutama penulisan yang baik dalam menyelesaikan laporan.
3. Bapak Agus Wahyu Widodo, S.T, M.Cs., selaku ketua Program Studi Teknik Informatika.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku ketua Jurusan Teknik Informatika.
5. Bapak Kadek Armita, selaku kepala UKM Bali Tekindo Jaya yang bersedia menjadi topik penelitian penulis dan memberikan informasi yang penulis butuhkan dengan sangat lengkap.
6. Ayahanda dan Ibunda juga seluruh keluarga serta orang-orang terdekat atas segala nasehat, kasih sayang, perhatian dan kesabarannya dalam mendengarkan segala keluh kesah penulis, pun yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya laporan ini.
7. Seluruh civitas akademika Teknik Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penyelesaian laporan PKL ini.

Penulis menyadari bahwa dalam penyusunan laporan ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap laporan skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 20 Juni 2019

Penulis

irmarini15@gmail.com

ABSTRAK

Ni Luh Irma Arini, Pengembangan Sistem Manajemen Layanan Jasa Servis Alat Elektronik Rumah Tangga (Studi Kasus: UKM Bali Tekindo Jaya)

Pembimbing: Bayu Priyambadha, S.Kom., M.Kom. dan Ir. Heru Nurwarsito, M.Kom.

Alat-alat elektronik merupakan suatu hal yang sudah pasti ditemui dalam suatu rumah tangga. Bahkan, kegunaan alat elektronik mulai bergeser fungsi menjadi kebutuhan primer di beberapa pihak terutama dalam membantu memenuhi kebutuhan sehari-hari. UKM Bali Tekindo Jaya merupakan salah satu usaha kecil menengah yang menawarkan jasa perbaikan maupun pemasangan alat-alat elektronik dalam rumah tangga seperti pendingin ruangan, lemari pendingin, penanak nasi, mesin cuci, pemanas air, dan sebagainya. Seiring berkembangnya usaha, jumlah pelanggan yang dimiliki UKM kini menjadi bertambah sehingga berdampak juga pada penambahan jumlah pekerja untuk dapat menangani seluruh permintaan servis. Peningkatan kualitas layanan terhadap pelanggan tentunya perlu didukung dengan manajemen usaha yang baik dan efektif. Sehingga diperlukan sebuah sistem manajemen yang dapat membantu memaksimalkan layanan usaha di UKM Bali Tekindo Jaya. Guna membantu mengelola manajemen usaha UKM Bali Tekindo Jaya, skripsi ini membahas tentang pengembangan sebuah sistem manajemen berbasis web yang menggunakan *framework CodeIgniter* dengan bahasa PHP serta MySQL sebagai basis data. Seperti namanya, sistem ini berfokus pada manajemen usaha dilengkapi dengan fitur *booking* servis, agenda, laporan, penilaian, testimoni, juga notifikasi menggunakan *SMS Gateway*. Perancangan dilakukan sedemikian rupa hingga diperoleh tiga pengguna dalam sistem manajemen ini yaitu dari sisi pelanggan, pekerja lapangan, dan pemilik usaha sebagai admin. Hasil pengujian yang dilakukan dengan metode kotak hitam (*blackbox*) dan kotak putih (*whitebox*) terhadap web menunjukkan seluruh fungsional sistem dapat dijalankan dengan baik yaitu berstatus valid di semua kasus uji. Adapun pengujian usabilitas menunjukkan sistem yang dibangun sesuai dengan ekspektasi pengguna. Sementara pengujian kompatibilitas yang dilakukan terhadap berbagai web browser menunjukkan sistem manajemen yang telah dikembangkan tidak memiliki masalah yang berarti ketika dijalankan pada berbagai peramban.

Kata kunci: sistem manajemen, pengembangan web, *CodeIgniter*, *SMS Gateway*, pengujian fungsional, pengujian usabilitas, pengujian kompatibilitas

ABSTRACT

Ni Luh Irma Arini, Development of Household Electronics Repair Services Management Systems (Case Study: Bali Tekindo Jaya SMEs)

Supervisors: Bayu Priyambadha, S.Kom., M.Kom. and Ir. Heru Nurwarsito, M.Kom.

Electronic appliances is one of the things that is definitely found in a household. In fact, the usefulness of electronic appliance began to shift their functions into primary needs in some parties, especially in helping to meet their daily needs. Bali Tekindo Jaya is one of the small and medium-sized businesses that offers repair and installation services in electronic appliances such as air conditioners, refrigerators, rice cookers, washing machines, water heaters, and so on. As the business grows, the number of customers owned by SMEs has now increased so that it has an impact on the increase in the number of workers to be able to handle all service requests. Improving the quality of service to customers certainly needs to be supported by good and effective business management. So, a management system is needed that can help maximize business services in Bali Tekindo Jaya SMEs. To help manage the business management of Bali Tekindo Jaya SMEs, this paper discusses the development of a website-based management system that uses the CodeIgniter framework with PHP languages and MySQL as the database. As the name suggests, this system focuses on business management equipped with service booking features, agendas, reports, assessments, testimonials, and notifications using the SMS Gateway. The design was carried out in such a way as to obtain three users in this management system, namely from the customer side, field workers, and business owners as admin. The test results carried out by the black box and white box method on the website show that all functional systems can be run properly, which is valid in all test cases. Usability testing shows that the system is built according to user expectations. While compatibility testing carried out on various web browsers shows that the management system that has been developed does not have significant problems when run on various browsers.

Keywords: management system, website development, CodeIgniter, SMS Gateway, functional testing, usability testing, compatibility testing

DAFTAR ISI

HALAMAN JUDUL.....	i
PENGESAHAN	ii
PERNYATAAN ORISINALITAS.....	iii
PRAKATA.....	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL	xii
DAFTAR GAMBAR	xv
DAFTAR LAMPIRAN	xviii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Tujuan.....	3
1.4 Manfaat	3
1.5 Batasan Masalah	3
1.6 Sistematika Pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN.....	6
2.1 Penelitian Terdahulu	6
2.2 SERVQUAL.....	9
2.3 HTML	9
2.4 PHP	10
2.5 MySQL.....	10
2.6 Konsep <i>Model, View, dan Controller (MVC)</i>	11
2.7 <i>CodeIgniter</i>	12
2.8 <i>SMS Gateway</i>	12
2.9 <i>Software Development Process</i>	13
2.10 <i>Business Process Model and Notation</i>	15
2.11 <i>Usecase Diagram</i>	17
2.12 <i>Sequence Diagram</i>	18

2.13 Class Diagram	19
2.14 White-Box Testing	21
2.15 Black-Box Testing	23
2.16 Usability Testing	24
2.17 Compatibility Testing	24
BAB 3 METODOLOGI	26
3.1 Identifikasi Masalah di Lapangan	26
3.2 Studi Literatur	27
3.3 Analisis Kebutuhan	27
3.4 Perancangan	28
3.5 Implementasi	28
3.6 Pengujian dan Analisis	29
3.7 Penarikan Kesimpulan	29
BAB 4 REKAYASA KEBUTUHAN	30
4.1 Gambaran Umum Sistem	30
4.2 Identifikasi Aktor	32
4.3 Fungsi Utama Sistem	33
4.4 Usecase Diagram	35
4.5 Usecase Scenario	36
BAB 5 PERANCANGAN DAN IMPLEMENTASI	44
5.1 Sequence Diagram	44
5.1.1 Sequence Diagram Ubah Booking	44
5.1.2 Sequence Diagram Beri Penilaian	45
5.1.3 Sequence Diagram Tambah Agenda	45
5.2 Class Diagram	45
5.3 Rancangan Basis Data	47
5.4 Database Relational	48
5.5 Rancangan Algoritme	48
5.5.1 Algoritme Membuat Booking Baru	49
5.5.2 Algoritme Menghapus Data Booking	49
5.5.3 Algoritme Menghitung Rata-rata Rating	50
5.5.4 Algoritme Menghitung Rata-rata Nilai Kuesioner	50

5.5.5	Algoritme Menampilkan Data Penilaian	51
5.6	Rancangan Antarmuka	52
5.6.1	Rancangan Antarmuka Form Pendaftaran.....	52
5.6.2	Rancangan Antarmuka Form <i>Login</i>	53
5.6.3	Rancangan Antarmuka Pelanggan.....	53
5.6.4	Rancangan Antarmuka Admin.....	54
5.6.5	Rancangan Antarmuka Pekerja	54
5.6.6	Rancangan Antarmuka Lihat <i>Booking</i> – Pelanggan	55
5.6.7	Rancangan Antarmuka Tambah <i>Booking</i> – Pelanggan	55
5.6.8	Rancangan Antarmuka Ubah <i>Booking</i> – Pelanggan	56
5.6.9	Rancangan Antarmuka Lihat Testimoni – Pelanggan	57
5.6.10	Rancangan Antarmuka Tambah Testimoni – Pelanggan	57
5.6.11	Rancangan Antarmuka Lihat <i>Booking</i> – Admin	58
5.6.12	Rancangan Antarmuka Lihat Agenda – Admin	58
5.6.13	Rancangan Antarmuka Tambah Agenda – Admin	59
5.6.14	Rancangan Antarmuka Ubah Agenda - Admin	59
5.6.15	Rancangan Antarmuka Lihat Laporan – Admin	60
5.6.16	Rancangan Antarmuka Ubah Laporan – Admin	60
5.6.17	Rancangan Antarmuka Lihat Penilaian – Admin	61
5.6.18	Rancangan Antarmuka Lihat Testimoni – Admin	61
5.6.19	Rancangan Antarmuka Lihat Agenda – Pekerja	62
5.6.20	Rancangan Antarmuka Lihat Laporan – Pekerja	62
5.6.21	Rancangan Antarmuka Tambah Laporan – Pekerja	63
5.6.22	Rancangan Antarmuka Ubah Laporan - Pekerja	63
5.6.23	Rancangan Antarmuka Form Penilaian.....	64
5.6.24	Rancangan Antarmuka Status SMS Notifikasi	64
5.7	Implementasi Kode Program	65
5.7.1	Fungsi Membuat <i>Booking</i> Baru	65
5.7.2	Fungsi Menghapus Data <i>Booking</i>	65
5.7.3	Fungsi Menghitung Rata-rata Rating	66
5.7.4	Fungsi Menghitung Rata-rata Nilai Kuesioner	66
5.7.5	Fungsi Menampilkan Data Penilaian	67

5.8 Implementasi Antarmuka	68
5.8.1 Implementasi Antarmuka Form Pendaftaran	68
5.8.2 Implementasi Antarmuka Form <i>Login</i>	69
5.8.3 Implementasi Antarmuka Lihat <i>Booking</i> – Pelanggan.....	70
5.8.4 Implementasi Antarmuka Tambah <i>Booking</i> – Pelanggan.....	70
5.8.5 Implementasi Antarmuka Ubah <i>Booking</i> – Pelanggan	71
5.8.6 Implementasi Antarmuka Lihat Testimoni – Pelanggan.....	71
5.8.7 Implementasi Antarmuka Tambah Testimoni – Pelanggan.....	72
5.8.8 Implementasi Antarmuka Lihat <i>Booking</i> – Admin.....	72
5.8.9 Implementasi Antarmuka Lihat Agenda – Admin	73
5.8.10 Implementasi Antarmuka Tambah Agenda – Admin	73
5.8.11 Implementasi Antarmuka Ubah Agenda – Admin.....	74
5.8.12 Implementasi Antarmuka Lihat Laporan – Admin.....	74
5.8.13 Implementasi Antarmuka Ubah Laporan – Admin	75
5.8.14 Implementasi Antarmuka Lihat Penilaian – Admin	75
5.8.15 Implementasi Antarmuka Lihat Testimoni – Admin.....	76
5.8.16 Implementasi Antarmuka Lihat Agenda – Pekerja	76
5.8.17 Implementasi Antarmuka Lihat Laporan – Pekerja	77
5.8.18 Implementasi Antarmuka Tambah Laporan – Pekerja	77
5.8.19 Implementasi Antarmuka Ubah Laporan – Pekerja	78
5.8.20 Implementasi Antarmuka Form Penilaian	78
5.8.21 Implementasi Antarmuka Status SMS Notifikasi.....	79
BAB 6 PENGUJIAN DAN ANALISIS	80
6.1 Pengujian <i>Blackbox</i>	80
6.1.1 Kasus Uji Fungsional <i>Login</i>	80
6.1.2 Kasus Uji Fungsional <i>Logout</i>	80
6.1.3 Kasus Uji Fungsional Pendaftaran	81
6.1.4 Kasus Uji Fungsional Melihat Data <i>Booking</i>	81
6.1.5 Kasus Uji Fungsional Membuat <i>Booking</i> Baru.....	82
6.1.6 Kasus Uji Fungsional Mengubah Data <i>Booking</i>	82
6.1.7 Kasus Uji Fungsional Menghapus Data <i>Booking</i>	83

6.1.8 Kasus Uji Fungsional Memberi Penilaian	83
6.1.9 Kasus Uji Fungsional Kirim Notifikasi	84
6.1.10 Kasus Uji Fungsional Membuat Testimoni Baru	85
6.1.11 Kasus Uji Fungsional Menghapus Testimoni	85
6.1.12 Kasus Uji Fungsional Melihat Agenda	86
6.1.13 Kasus Uji Fungsional Menambahkan Agenda Baru	86
6.1.14 Kasus Uji Fungsional Mengubah Agenda	87
6.1.15 Kasus Uji Fungsional Menghapus Agenda.....	87
6.1.16 Kasus Uji Fungsional Melihat Laporan	88
6.1.17 Kasus Uji Fungsional Menambahkan Laporan Baru	88
6.1.18 Kasus Uji Fungsional Mengubah Laporan	89
6.1.19 Kasus Uji Fungsional Melihat Penilaian	90
6.2 Pengujian <i>Whitebox</i>	90
6.2.1 Algoritme Membuat <i>Booking</i> Baru.....	90
6.2.2 Algoritme Menghapus Data <i>Booking</i>	92
6.2.3 Algoritme Menghitung Rata-rata Rating.....	93
6.2.4 Algoritme Menghitung Rata-rata Nilai Kuesioner	95
6.2.5 Algoritme Menampilkan Data Penilaian	96
6.3 Pengujian <i>Usability</i>	98
6.4 Pengujian <i>Compatibility</i>	101
BAB 7 PENUTUP	103
7.1 Kesimpulan	103
7.2 Saran.....	104
DAFTAR REFERENSI	105
LAMPIRAN A HASIL WAWANCARA	107
A.1 Wawancara dengan Kepala UKM Bali Tekindo Jaya	107
LAMPIRAN B PERTANYAAN KUESIONER <i>USABILITY</i>	109
B.1 Pertanyaan Kuesioner <i>Usability</i>	109
LAMPIRAN C Dokumen pendukung	110
C.1 Nota Tagihan Kerja	110
C.2 Laporan Harian Pekerja UKM	110
C.3 Rekap Laporan oleh Kepala UKM.....	111

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu.....	6
Tabel 2.2 Komponen BPMN	16
Tabel 2.3 <i>Komponen Usecase Diagram</i>	17
Tabel 2.4 <i>Komponen Sequence Diagram</i>	18
Tabel 2.5 Versi <i>Browser</i> pada Pengujian <i>Compatibility</i>	25
Tabel 2.6 Indikator Permasalahan <i>Compatibility</i>	25
Tabel 4.1 Identifikasi Aktor dalam Sistem.....	32
Tabel 4.2 Kebutuhan Fungsional Sistem	33
Tabel 4.3 Kebutuhan Non Fungsional	35
Tabel 4.4 <i>Usecase</i> Scenario Login	36
Tabel 4.5 <i>Usecase</i> Scenario Logout	37
Tabel 4.6 <i>Usecase</i> Scenario Daftar	37
Tabel 4.7 <i>Usecase</i> Scenario <i>Booking</i> Servis Baru	37
Tabel 4.8 <i>Usecase</i> Scenario Lihat <i>Booking</i>	38
Tabel 4.9 <i>Usecase</i> Scenario Ubah <i>Booking</i> Servis	38
Tabel 4.10 <i>Usecase</i> Scenario Batal <i>Booking</i>	39
Tabel 4.11 <i>Usecase</i> Scenario Beri Penilaian	39
Tabel 4.12 <i>Usecase</i> Scenario Terima Notifikasi.....	39
Tabel 4.13 <i>Usecase</i> Scenario Testimoni Baru.....	40
Tabel 4.14 <i>Usecase</i> Scenario Hapus Testimoni	40
Tabel 4.15 <i>Usecase</i> Scenario Lihat Agenda	40
Tabel 4.16 <i>Usecase</i> Scenario Tambah Agenda	41
Tabel 4.17 <i>Usecase</i> Scenario Ubah Agenda	41
Tabel 4.18 <i>Usecase</i> Scenario Hapus Agenda	42
Tabel 4.19 <i>Usecase</i> Scenario Lihat Laporan	42
Tabel 4.20 <i>Usecase</i> Scenario Tambah Laporan	42
Tabel 4.21 <i>Usecase</i> Scenario Ubah Laporan	43
Tabel 4.22 <i>Usecase</i> Scenario Lihat Penilaian	43
Tabel 5.1 Algoritme Membuat Data <i>Booking</i> Baru	49
Tabel 5.2 Algoritme Menghapus Data <i>Booking</i>	49

Tabel 5.3 Algoritme Menghitung Rata-rata Rating	50
Tabel 5.4 Algoritme Menghitung Rata-rata Nilai Kuesioner	50
Tabel 5.5 Algoritme Menampilkan Data Penilaian	51
Tabel 5.6 Implementasi Fungsi Membuat <i>Booking</i> Baru	65
Tabel 5.7 Implementasi Fungsi Menghapus Data <i>Booking</i>	65
Tabel 5.8 Implementasi Fungsi Menghitung Rata-rata Rating	66
Tabel 5.9 Implementasi Fungsi Menghitung Rata-rata Nilai Kuesioner	66
Tabel 5.10 Implementasi Fungsi Menampilkan Data Penilaian	67
Tabel 6.1 Kasus Uji Fungsional <i>Login</i>	80
Tabel 6.2 Kasus Uji Fungsional <i>Logout</i>	80
Tabel 6.3 Kasus Uji Fungsional Pendaftaran	81
Tabel 6.4 Kasus Uji Fungsional Lihat <i>Booking</i>	81
Tabel 6.5 Kasus Uji Fungsional <i>Booking</i> Baru	82
Tabel 6.6 Kasus Uji Fungsional Ubah <i>Booking</i>	82
Tabel 6.7 Kasus Uji Fungsional Batal <i>Booking</i>	83
Tabel 6.8 Kasus Uji Fungsional Beri Penilaian	83
Tabel 6.9 Kasus Uji Fungsional Kirim Notifikasi	84
Tabel 6.10 Kasus Uji Fungsional Beri Testimoni	85
Tabel 6.11 Kasus Uji Fungsional Hapus Testimoni	85
Tabel 6.12 Kasus Uji Fungsional Lihat Agenda	86
Tabel 6.13 Kasus Uji Fungsional Tambah Agenda	86
Tabel 6.14 Kasus Uji Fungsional Ubah Agenda	87
Tabel 6.15 Kasus Uji Fungsional Hapus Agenda	87
Tabel 6.16 Kasus Uji Fungsional Lihat Laporan	88
Tabel 6.17 Kasus Uji Fungsional Tambah Laporan	88
Tabel 6.18 Kasus Uji Fungsional Ubah Laporan	89
Tabel 6.19 Kasus Uji Fungsional Lihat Penilaian	90
Tabel 6.20 Pengujian Algoritme Membuat Data <i>Booking</i> Baru	90
Tabel 6.21 Kasus Uji Fungsi Membuat Data <i>Booking</i> Baru	91
Tabel 6.22 Pengujian Algoritme Menghapus Data <i>Booking</i>	92
Tabel 6.23 Kasus Uji Fungsi Menghapus Data <i>Booking</i>	93
Tabel 6.24 Pengujian Algoritme Menghitung Rata-rata Rating	94

Tabel 6.25 Kasus Uji Fungsi Menghitung Rata-rata Rating	94
Tabel 6.26 Pengujian Algoritme Menghitung Rata-rata Nilai Kuesioner	95
Tabel 6.27 Kasus Uji Fungsi Menghitung Rata-rata Nilai Kuesioner.....	96
Tabel 6.28 Pengujian Algoritme Menampilkan Data Penilaian.....	96
Tabel 6.29 Kasus Uji Fungsi Menampilkan Data Penilaian	97
Tabel 6.30 Skala <i>Likert 5 Level of Agreement</i>	99
Tabel 6.31 Interpretasi Skor <i>Likert</i>	99
Tabel 6.32 Daftar Pernyataan Kuesioner	100
Tabel 6.34 Indikator Permasalahan <i>Compatibility</i>	102

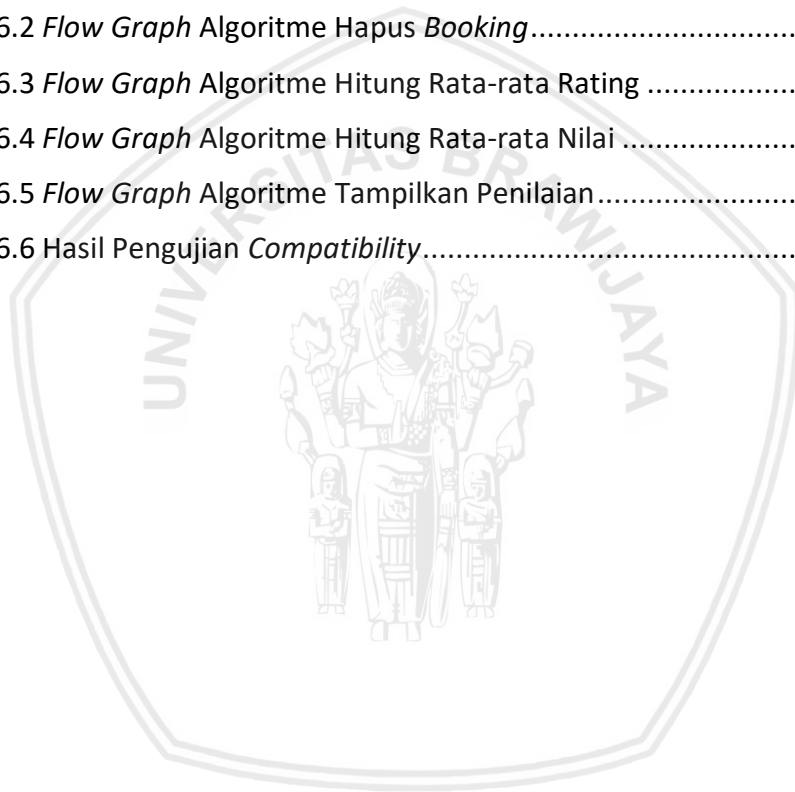


DAFTAR GAMBAR

Gambar 2.1 Alur Konsep MVC.....	11
Gambar 2.2 <i>Software Development Process</i>	13
Gambar 2.3 <i>Software Development Process Flow</i>	14
Gambar 2.4 Contoh BPMN	15
Gambar 2.5 Contoh <i>Usecase Diagram</i>	17
Gambar 2.6 Contoh <i>Sequence Diagram</i>	19
Gambar 2.7 Contoh <i>Class Diagram</i>	20
Gambar 2.8 <i>Class Diagram Detail</i>	20
Gambar 2.9 <i>White-box Testing</i>	21
Gambar 2.10 Struktur <i>Flow Graph</i>	22
Gambar 2.11 Contoh <i>Flow Graph</i>	22
Gambar 2.12 <i>Predicate Node</i> dalam <i>Flow Graph</i>	23
Gambar 2.13 <i>Black-box Testing</i>	23
Gambar 3.1 Alur Metode Penelitian.....	26
Gambar 4.1 BPMN (<i>as is</i>) UKM Bali Tekindo Jaya	31
Gambar 4.2 BPMN (<i>to be</i>) UKM Bali Tekindo Jaya	32
Gambar 4.3 Aturan Penomoran Kebutuhan Sistem.....	33
Gambar 4.4 <i>Usecase Diagram</i> Sistem.....	35
Gambar 5.1 <i>Sequence Diagram</i> Ubah <i>Booking</i>	44
Gambar 5.2 <i>Sequence Diagram</i> Isi Kuesioner dan Terima Notifikasi	45
Gambar 5.3 <i>Sequence Diagram</i> Tambah Agenda.....	45
Gambar 5.4 <i>Class Diagram</i>	46
Gambar 5.5 Entity Relationship <i>Diagram</i>	47
Gambar 5.6 Skema Relasi Basis Data	48
Gambar 5.7 Rancangan antarmuka Form Pendaftaran	52
Gambar 5.8 Rancangan Antarmuka Form <i>Login</i>	53
Gambar 5.9 Rancangan Antarmuka Pelanggan.....	53
Gambar 5.10 Rancangan Antarmuka Admin.....	54
Gambar 5.11 Rancangan Antarmuka Pekerja	54
Gambar 5.12 Rancangan Antarmuka Lihat <i>Booking</i> – Pelanggan	55

Gambar 5.13 Rancangan Antarmuka Tambah <i>Booking</i> – Pelanggan	55
Gambar 5.14 Rancangan Antarmuka Ubah <i>Booking</i> – Pelanggan	56
Gambar 5.15 Rancangan Antarmuka Lihat Testimoni – Pelanggan	57
Gambar 5.16 Rancangan Antarmuka Tambah Testimoni – Pelanggan	57
Gambar 5.17 Rancangan Antarmuka Lihat <i>Booking</i> – Admin	58
Gambar 5.18 Rancangan Antarmuka Lihat Agenda – Admin.....	58
Gambar 5.19 Rancangan Antarmuka Tambah Agenda – Admin.....	59
Gambar 5.20 Rancangan Antarmuka Ubah Agenda – Admin	59
Gambar 5.21 Rancangan Antarmuka Lihat Laporan – Admin.....	60
Gambar 5.22 Rancangan Antarmuka Ubah Laporan – Admin	60
Gambar 5.23 Rancangan Antarmuka Lihat Penilaian – Admin	61
Gambar 5.24 Rancangan Antarmuka Lihat Testimoni – Admin	61
Gambar 5.25 Rancangan Antarmuka Lihat Agenda – Pekerja	62
Gambar 5.26 Rancangan Antarmuka Lihat Laporan – Pekerja	62
Gambar 5.27 Rancangan Antarmuka Tambah Laporan – Pekerja	63
Gambar 5.28 Rancangan Antarmuka Ubah Laporan – Pekerja.....	63
Gambar 5.29 Rancangan Antarmuka Form Penilaian.....	64
Gambar 5.30 Rancangan Antarmuka Status SMS Notifikasi	64
Gambar 5.31 Implementasi Antarmuka Form Pendaftaran	68
Gambar 5.32 Implementasi Antarmuka Form <i>Login</i>	69
Gambar 5.33 Implementasi Antarmuka Lihat <i>Booking</i> – Pelanggan.....	70
Gambar 5.34 Implementasi Antarmuka Tambah <i>Booking</i> – Pelanggan.....	70
Gambar 5.35 Implementasi Antarmuka Ubah <i>Booking</i> – Pelanggan	71
Gambar 5.36 Implementasi Antarmuka Lihat Testimoni – Pelanggan.....	71
Gambar 5.37 Implementasi Antarmuka Tambah Testimoni – Pelanggan.....	72
Gambar 5.38 Implementasi Antarmuka Lihat <i>Booking</i> – Admin.....	72
Gambar 5.39 Implementasi Antarmuka Lihat Agenda – Admin	73
Gambar 5.40 Implementasi Antarmuka Tambah Agenda – Admin	73
Gambar 5.41 Implementasi Antarmuka Ubah Agenda – Admin.....	74
Gambar 5.42 Implementasi Antarmuka Lihat Laporan – Admin.....	74
Gambar 5.43 Implementasi Antarmuka Ubah Laporan – Admin.....	75
Gambar 5.44 Implementasi Antarmuka Lihat Penilaian – Admin	75

Gambar 5.45 Implementasi Antarmuka Lihat Testimoni – Admin	76
Gambar 5.46 Implementasi Antarmuka Lihat Agenda – Pekerja	76
Gambar 5.47 Implementasi Antarmuka Lihat Laporan – Pekerja	77
Gambar 5.48 Implementasi Antarmuka Tambah Laporan – Pekerja	77
Gambar 5.49 Implementasi Antarmuka Ubah Laporan – Pekerja	78
Gambar 5.50 Implementasi Antarmuka Form Penilaian	78
Gambar 5.51 Implementasi Antarmuka Status SMS Notifikasi.....	79
Gambar 6.1 <i>Flow Graph</i> Algoritme <i>Booking</i> Baru	91
Gambar 6.2 <i>Flow Graph</i> Algoritme Hapus <i>Booking</i>	92
Gambar 6.3 <i>Flow Graph</i> Algoritme Hitung Rata-rata Rating	94
Gambar 6.4 <i>Flow Graph</i> Algoritme Hitung Rata-rata Nilai	96
Gambar 6.5 <i>Flow Graph</i> Algoritme Tampilkan Penilaian	97
Gambar 6.6 Hasil Pengujian <i>Compatibility</i>	101



DAFTAR LAMPIRAN

LAMPIRAN A HASIL WAWANCARA	107
A.1 Wawancara dengan Kepala UKM Bali Tekindo Jaya.....	107
LAMPIRAN B PERTANYAAN KUESIONER <i>USABILITY</i>	109
B.1 Pertanyaan Kuesioner <i>Usability</i>	109
LAMPIRAN C DOKUMEN PENDUKUNG	110
C.1 Nota Tagihan Kerja.....	110
C.2 Laporan Harian Pekerja UKM	110
C.3 Rekap Laporan oleh kepala UKM.....	111



BAB 1 PENDAHULUAN

Pendahuluan adalah Bab awal dalam skripsi ini mengenai sebab dipilihnya topik sistem layanan jasa servis sehingga memperoleh beberapa rumusan masalah yang akan diselesaikan berikut pendefinisian tujuan dan manfaat yang didapatkan namun tetap memiliki batasan permasalahan serta penjelasan singkat tentang rancangan dokumen skripsi yang menjabarkan sistematika penulisan.

1.1 Latar Belakang

Alat-alat elektronik merupakan suatu hal yang sudah pasti ditemui dalam suatu rumah tangga. Bahkan, di masa kini kegunaan alat elektronik mulai bergeser fungsi menjadi kebutuhan primer terutama dalam membantu memenuhi kebutuhan sehari-hari. Tak hanya karena kegunaannya yang sangat dibutuhkan, alat elektronik juga menjadi sumber mata pencaharian pokok bagi sebagian orang yang menggeluti bidang jasa perbaikan alat-alat elektronik. Seperti kutipan Agung (2017) dalam *Journal of Business Studies* yang menjabarkan pernyataan Philip Kotler bahwa jasa merupakan serangkaian aktivitas yang bertempat pada interaksi kontak seseorang atau mesin dan menghasilkan kepuasan konsumen. Hal tersebut berarti kepuasan pengguna jasa merupakan tujuan dari penyedia jasa.

UKM Bali Tekindo Jaya merupakan salah satu usaha kecil menengah yang menawarkan jasa perbaikan maupun pemasangan alat-alat elektronik dalam rumah tangga seperti pendingin ruangan, lemari pendingin, penanak nasi, mesin cuci, pemanas air, dan sebagainya. Selain perbaikan, usaha ini juga menyediakan jasa instalasi atau pemasangan alat elektronik baru. Alih-alih pelanggan yang datang ke lapak usaha melainkan para pekerja UKM yang datang langsung dari rumah ke rumah dalam melayani pelanggan. Hal inilah yang menjadi nilai tambah pelanggan memilih jasa UKM Bali Tekindo Jaya.

Seiring berkembangnya usaha, permintaan yang semakin banyak menyebabkan manajemen perusahaan dalam hal pendataan permintaan, agenda pekerjaan, penilaian pekerja, dan pelaporan pekerjaan servis serta tingkat pengawasan dari kepala UKM semakin menurun. Permintaan servis yang dilakukan pelanggan melalui telepon seringkali tidak terorganisir dengan baik berkaitan dengan antrian servis yang tidak selalu berurutan maupun permintaan servis yang tak jarang menjadi terlupakan sebab banyaknya permintaan. Selama ini, pelaporan pekerjaan harian usaha terdiri dari dua pembukuan yaitu oleh pekerja yang berisikan laporan harian masing-masing pekerja dan oleh kepala UKM yang berisikan laporan seluruh pekerjaan beserta biaya. Berdasarkan hal tersebut terdapat duplikasi laporan karena kepala UKM harus menyalin laporan pekerja ke dalam laporan pribadi kepala UKM.

Pekerja yang setiap harinya berkeliling ke rumah-rumah pelanggan untuk memberikan servis terbilang kurang pengawasan karena kepala UKM tidak mengetahui dengan pasti bagaimana performa pekerja di lapangan. Kepala UKM hanya mengetahui performa pekerja secara subjektif yang dilihat ketika sedang bekerja bersama. Performa pekerja tentunya tak hanya dinilai oleh kepala UKM namun juga oleh pelanggan sebagai penerima jasa. Terdapat lima dimensi penilaian penentu kualitas jasa yang tertulis dalam jurnal *Retailing* oleh Parasuraman (1998) diantaranya adalah *tangibles* (bukti fisik), *reliability* (kehandalan), *responsiveness* (daya tanggap), *assurance* (jaminan), dan *empathy* (perhatian). Kelima dimensi ini dinamakan model SERVQUAL yang patut diperhatikan pekerja lapangan saat berada di tempat pelanggan.

Sehari-hari, pekerja akan mendapatkan agenda kerja berupa alamat pelanggan dan jenis pekerjaan yang harus diselesaikan. Jumlah pekerjaan dalam agenda setiap harinya tidak menentu bergantung kepada permintaan pelanggan. Jika agenda yang diperoleh dapat terselesaikan sebelum jam istirahat artinya pekerja akan menganggur sampai waktunya pulang. Untuk menghindari hal tersebut, kepala UKM biasanya akan memberikan agenda tambahan kepada pekerja namun baru menyampaikannya setelah pekerja melapor agenda telah diselesaikan. Jika pekerja tidak melapor dan pekerja tidak datang ke kantor setelah menyelesaikan seluruh pekerjaan, kepala UKM tidak akan memberikan agenda tambahan. Hal ini menyebabkan penting untuk kepala UKM mengetahui status pekerja dan waktu pekerjaan selesai untuk memaksimalkan layanan.

Peningkatan kualitas layanan terhadap pelanggan tentunya perlu didukung dengan manajemen usaha yang baik dan efektif. Jika pelayanan terhadap pelanggan dapat dilakukan secara maksimal tentunya akan berdampak pada keuntungan usaha. Berdasarkan permasalahan tersebut, perlu dibangun sebuah sistem yang mampu membantu kepala UKM dalam manajemen pekerjaan seperti pelaporan pekerjaan, memaksimalkan agenda pekerjaan, serta perekaman data ulasan pelanggan. Permasalahan inilah yang mendorong minat penulis untuk mengembangkan "Sistem Manajemen Layanan Servis Alat Elektronik Rumah Tangga untuk UKM Bali Tekindo Jaya".

Sebuah sistem manajemen yang dapat membantu UKM Bali Tekindo Jaya dalam menjalankan usaha yang lebih efisien adalah sistem yang mampu (1) mengatur agenda pekerjaan sesuai permintaan pelanggan; (2) mewadahi tempat pelanggan melakukan pemesanan layanan jasa servis secara online; (3) memaksimalkan pekerjaan lapangan dengan laporan penyelesaian yang memastikan waktu; (4) merekam penilaian pelanggan terhadap pekerja lapangan juga mewadahi ulasan pelanggan di kemudian hari; serta (5) mengelola data pekerjaan yang diinputkan pekerja menjadi laporan. Untuk mewujudkan sistem tersebut, seluruh pengguna perlu mengaksesnya dimana dan kapan saja melalui berbagai perangkat yang dimiliki. Penggunaan sistem tersebut pun tentunya harus mampu memenuhi keinginan pengguna dengan mempermudah proses yang berjalan selama ini. Sehingga diharapkan sistem manajemen yang dibangun dapat membantu memaksimalkan layanan usaha di UKM Bali Tekindo Jaya.

1.2 Rumusan Masalah

Berdasarkan penjelasan dalam latar belakang permasalahan, berikut adalah rumusan masalah yang akan dibahas:

1. Bagaimana hasil dari rekayasa kebutuhan sistem manajemen UKM yang dikembangkan?
2. Bagaimana hasil merancang sistem manajemen UKM berdasarkan kebutuhan yang diperoleh?
3. Bagaimana hasil implementasi dari rancangan sistem untuk manajemen UKM?
4. Bagaimana hasil pengujian sistem manajemen UKM yang telah diimplementasikan?

1.3 Tujuan

Berdasarkan permasalahan yang didefinisikan, berikut tujuan dan hasil yang hendak dicapai dalam penelitian ini:

1. Untuk memberikan spesifikasi kebutuhan sistem yang diperlukan UKM dalam manajemen layanan jasa servis.
2. Untuk merancang sistem yang dapat digunakan oleh UKM dalam manajemen layanan jasa servis.
3. Untuk mengimplementasikan rancangan dari sistem agar dapat dimanfaatkan dalam manajemen layanan jasa servis.
4. Untuk menguji sistem yang telah dikembangkan dan menganalisis hasil pengujian sistem.

1.4 Manfaat

Berdasarkan rumusan masalah yang telah diperoleh, berikut manfaat yang dapat diperoleh dari terselesaikannya penelitian ini:

1. Mendefinisikan spesifikasi kebutuhan sistem yang diperlukan UKM dalam manajemen layanan jasa servis.
2. Memberikan rancangan sistem berupa pemodelan sistem yang dapat digunakan oleh UKM dalam manajemen layanan jasa servis.
3. Mengimplementasikan rancangan sistem menjadi sebuah sistem yang dapat dioperasikan sebagai manajemen layanan jasa servis.
4. Memberikan hasil pengujian sistem yang telah dibuat serta menganalisis hasil pengujian untuk memastikan kesesuaian dengan kebutuhan yang telah didefinisikan.

1.5 Batasan Masalah

Masalah yang diangkat terkait pengembangan sistem manajemen jasa servis alat elektronik pada UKM Bali Tekindo Jaya hanya mencakup permasalahan pekerjaan terkait jasa servis, tidak termasuk penjualan baik penjualan suku cadang maupun alat elektronik itu sendiri. Sistem yang dikembangkan hanya

terbatas pada lingkungan UKM Bali Tekindo Jaya dan pelanggannya sebagai pengguna dalam permintaan servis, agenda kerja, memaksimalkan pekerjaan harian, pelaporan pekerjaan harian, serta pencatatan ulasan pelanggan. Sistem yang dikembangkan tidak mencakup mekanisme pembayaran servis.

Selain itu, sistem manajemen yang dikembangkan terbatas pada seluruh pemesanan jasa servis yang dilakukan melalui web dengan akun pelanggan sebagai pemesan. Sistem manajemen memiliki alur yang dimulai dari proses *booking* sampai pelaporan servis. Pengguna yang tidak memiliki hak akses pelanggan tidak dapat membuat *booking* servis.

1.6 Sistematika Pembahasan

Sistematika pembahasan adalah bagian yang memuat rancangan bab pada penelitian ini, diantaranya:

1. Bab 1: Pendahuluan

Bagian memuat gambaran umum penelitian yang akan dikerjakan. Gambaran umum ini dimulai dengan penjelasan latar belakang yang mendasari pemilihan topik hingga diperoleh rumusan masalah, tujuan yang menggambarkan hasil yang ingin dicapai, manfaat yang diperoleh dari pengembangan sistem, serta penjelasan mengenai batasan permasalahan yang dibahas serta sistematika pembahasan untuk memudahkan mengetahui keseluruhan bagian dokumen skripsi ini.

2. Bab 2: Landasan Kepustakaan

Bagian ini berisikan penjabaran konsep yang terkait dengan penelitian yang diangkat dalam skripsi ini. Penjabaran tersebut sesuai dengan literatur dan fakta yang mendukung.

3. Bab 3: Metodologi

Bagian metodologi penelitian menjabarkan metode yang digunakan penulis dalam meneliti permasalahan dan mengembangkan sistem. Metode tersebut terdiri dari metode penggalan kebutuhan penelitian dan metode pengembangan sistem manajemen.

4. Bab 4: Rekayasa Kebutuhan

Rekayasa kebutuhan sistem berisikan analisa kebutuhan dari para *stakeholder* yang berperan sebagai pengguna sistem nantinya. Kebutuhan pengguna sistem didefinisikan menggunakan kalimat kebutuhan. Kalimat-kalimat yang telah dituliskan menjadi dasar memodelkan kebutuhan dengan *diagram-diagram*.

5. Bab 5: Perancangan dan Implementasi

Perancangan sistem dibuat dengan memodelkan sistem sebelum diimplementasikan berdasarkan kebutuhan pengguna yang telah diperoleh. Pemodelan rancangan digambarkan dengan *diagram* yang menjelaskan alur

sistem, susunan *class* dan relasinya, rancangan antarmuka maupun rancangan basis data sistem. Sehingga rancangan yang telah dibuat sebelumnya dapat diimplementasikan dengan menampilkan kode program maupun tampilan sistem.

6. Bab 6: Pengujian dan Analisis

Hasil implementasi sistem akan diuji untuk memastikan segala kebutuhan terpenuhi dan sistem dapat berjalan sesuai kebutuhan yang dijabarkan pada analisis kebutuhan. Pengujian dilakukan secara algoritme maupun kesesuaian fungsi yang didefinisikan.

7. Bab 7: Penutup

Bab terakhir dari penulisan skripsi ini berisikan kesimpulan dari seluruh rangkaian penelitian yang menjawab rumusan permasalahan pada bab pertama. Tentunya terdapat saran untuk penelitian selanjutnya guna menyempurnakan penelitian ini.



BAB 2 LANDASAN KEPUSTAKAAN

Landasan Kepustakaan adalah bagian dalam dokumen penelitian skripsi yang mengandung pembahasan mengenai konsep maupun teori berkaitan dengan topik pengembangan sistem manajemen layanan jasa servis. Konsep-konsep yang dijabarkan tentunya didukung dengan literatur terdahulu. Konsep tersebut dijabarkan baik dalam bentuk penjelasan teori atau pengertian, metode, maupun model yang nantinya dapat menyelesaikan permasalahan sesuai tema yang diangkat dalam penelitian ini.

2.1 Penelitian Terdahulu

Dalam menyelesaikan penelitian ini, penulis telah melakukan studi literatur terkait penelitian serupa yang pernah dilakukan sebelumnya. Banyak penelitian sebelumnya yang mengangkat topik terkait manajemen layanan jasa, namun belum ada layanan jasa perbaikan alat elektronik. Sejauh ini, layanan jasa yang diangkat dalam penelitian sebelumnya adalah perbaikan kendaraan, jasa penyewaan barang, jasa transportasi, penyedia layanan internet, dll. Selain itu, dari penelitian terkait sistem manajemen perusahaan tidak sekaligus mencakup mekanisme penilaian atau ulasan dari pelanggan terhadap layanan yang diberikan. Maka dari itu, penulis telah menggabungkan beberapa referensi dan mengelompokkannya menjadi dua bagian yaitu penelitian terkait website manajemen usaha dan penilaian kepuasan pelanggan yang akan menjadi acuan dalam membangun sistem manajemen jasa servis bagi UKM Bali Tekindo Jaya.

Tabel 2.1 Penelitian Terdahulu

No.	Penulis	Judul	Perbedaan	
			Penelitian Sebelumnya	Penelitian Penulis
1.	Winardi, A. dkk. (2017)	Rancang Bangun Sistem Informasi Manajemen Bengkel	Merancang sistem informasi manajemen bengkel dengan fitur transaksi penjualan barang, layanan servis, pemesanan barang, dan pelaporan.	Merancang sistem informasi manajemen layanan servis alat elektronik tanpa transaksi penjualan barang.
2.	Alfathin, A.H. dkk. (2018)	Perancangan SMS Gateway sebagai Reminder Pembayaran Tagihan Layanan	Merancang sistem yang berfokus pada reminder melalui SMS Gateway, dengan data pembayaran pelanggan saja.	Menggunakan SMS Gateway secara otomatis sebagai notifikasi terselesaikannya pekerjaan servis.

		Internet di Muna Net Media		
3.	Osman, M.N. dkk. (2017)	<i>Online Car Rental System using Web-Based and SMS Technology</i>	Merancang sistem persewaan kendaraan online yang berfokus pada pemberitahuan untuk pelanggan terkait <i>booking</i> status dan ketersediaan kendaraan.	Menggunakan <i>SMS Gateway</i> secara otomatis sebagai notifikasi terselesaikannya pekerjaan servis.
4.	Parasuraman, A. (1988)	<i>SERVQUAL: A Multiple-Item Scale for Measuring Consumer Perceptions of Service Quality</i>	Menjelaskan konsep SERVQUAL untuk dapat melakukan penilaian terhadap kinerja dengan lima parameter penilaian.	Menerapkan konsep SERVQUAL untuk digunakan sebagai acuan kuesioner penilaian kinerja pekerja oleh pelanggan pada fungsi beri penilaian.
5.	Izogo, E.E. (2015)	<i>Service Quality, Customer Satisfaction and Loyalty in Automobile Repair Service Sector</i>	Menerapkan metode SERVQUAL untuk mengetahui kepuasan pelanggan bengkel membuktikan hubungan dimensi kualitas layanan yang positif terhadap loyalitas pelanggan.	Menerapkan konsep SERVQUAL untuk digunakan sebagai acuan kuesioner penilaian kinerja pekerja oleh pelanggan pada fungsi beri penilaian.

2.1.1 Penelitian Terkait Web Manajemen Usaha

Dalam *Indonesian Journal of Software Engineering (IJSE)* Volume 3, terdapat sebuah tulisan yang sangat relevan dengan sistem yang ingin penulis bangun dengan topik manajemen bengkel. Sistem manajemen tersebut dibangun berupa website untuk CV. Anugrah. Sebelumnya CV. Anugrah menggunakan sistem yang masih manual, mulai dari proses transaksi penjualan, proses pelayanan servis, proses pemesanan, pembelian dan penerimaan barang serta pelaporan sehingga sering terjadi ketidakakuratan dalam pencatatan data dan pengolahan informasi

Winardi (2017). Untuk itulah proses bisnis bengkel tersebut menjadi lebih cepat dan mudah ketika dijalankan dengan website.

Sementara itu, salah satu penelitian terkait sistem informasi manajemen jasa penyedia layanan internet menyoroti tentang pengingat berupa pemberitahuan pembayaran tagihan pelanggan melalui layanan *Short Message Service* (SMS). SMS merupakan salah satu fitur pada handphone yang pasti digunakan oleh pengguna (user), baik untuk mengirim, maupun untuk menerima SMS (Alfathin et al, 2018). Sesuai *flowchart* sistem pada penelitian tersebut, SMS dilakukan secara otomatis dengan mengambil data pembayaran dari *database* tagihan pelanggan. Hal ini membuat penulis mempertimbangkan penggunaan layanan SMS untuk laporan pemberitahuan penyelesaian pekerjaan yang otomatis dikirimkan kepada kepala UKM dalam upaya memaksimalkan pekerjaan harian UKM Bali Tekindo Jaya.

Penggunaan web yang terintegrasi dengan layanan SMS juga terdapat dalam jurnal *Computing Research & Innovation* (CRINN) dengan topik *Car Rental*. *The integration of web-based and SMS technology in the car rental agencies is the best way to take the advantages of today technology, in order to enhance the productivity and efficiency of organization. Thus, there is no delay in the availability of any information, whether needed, can be captured very quickly and easily* (Osman et al, 2017). Layanan SMS dapat membantu interaksi dengan cepat ke pelanggan karena sudah pasti seluruh pengguna *handphone* dapat setidaknya menerima SMS.

2.1.2 Penelitian Terkait Penilaian Kepuasan Pelanggan

Kepuasan pelanggan dapat diukur dari penilaiannya terhadap pekerja. Namun, untuk menilai pekerja diperlukan parameter yang dapat dinilai dengan mengisi sejumlah pertanyaan yang bersifat tertutup karena tidak semua orang suka menulis ulasan. Terdapat lima dimensi kualitas pelayanan dalam *Journal of Retailing* yang dinamakan SERVQUAL (Parasuraman et al, 1988). Adapun kelima dimensi tersebut antara lain (1) *tangibles*, yaitu bukti fisik secara langsung. Bukti yang dapat dilihat oleh pelanggan penerima jasa servis dalam hal ini adalah pekerja yang datang dengan peralatannya ketika hendak memberikan layanan jasa. (2) *reliability* atau keandalan, yaitu kemampuan pekerja dalam menyelesaikan permasalahan (terkait kerusakan barang elektronik yang akan diperbaiki). (3) *responsiveness* atau tanggap, yaitu penilaian sikap pekerja yang cepat dan tepat dalam memberikan bantuan sesuai permintaan pelanggan. (4) *assurance* atau jaminan, jaminan pekerja dalam hal ini adalah memberikan kepercayaan pelanggan kepada perusahaan (berkaitan juga dengan jaminan rasa aman saat pekerja masuk ke rumah pelanggan). dan (5) *empathy*, yaitu perhatian tulus yang diberikan pekerja kepada pelanggan untuk memahami keinginan pelanggan. Kelima dimensi inilah yang penulis jadikan acuan dalam menyusun pertanyaan penilaian pekerja oleh pelanggan.

Mengacu pada lima dimensi penilaian kepuasan pelanggan (SERVQUAL) oleh Parasuraman, salah satu tulisan dalam *International Journal of Quality & Reliability Management* telah menerapkannya untuk mengetahui tingkat kepuasan pelanggan bengkel. Penelitian tersebut membuktikan hubungan dimensi kualitas layanan yang positif terhadap loyalitas pelanggan. *The SERVQUAL scale is a valid and reliable scale for measuring automobile service quality and can serve as a good starting point for scale construction* (Izogo & Ogba, 2014). Walaupun SERVQUAL telah diperdebatkan dengan metode baru yang lebih umum, namun setelah dua dekade pun SERVQUAL masih terbilang andal dan tetap menjadi acuan dalam penyusunan parameter penilaian kualitas layanan khususnya layanan jasa.

2.2 SERVQUAL

Servqual merupakan sebuah metode yang disampaikan oleh Parasuraman terkait pengukuran kualitas jasa. Metode tersebut digunakan dalam penyusunan kuesioner untuk mengukur kualitas layanan terhadap pelanggan. dengan adanya kuesioner SERVQUAL, dapat diketahui seberapa celah (gap) yang ada di antara persepsi pelanggan dan ekspektasi pelanggan terhadap suatu perusahaan jasa. Kuesioner SERVQUAL dapat diubah-ubah atau disesuaikan agar cocok dengan industri jasa yang berbeda-beda pula. Industri jasa tersebut dapat berarti perbankan, restoran, jasa servis, atau perusahaan telekomunikasi. SERVQUAL memiliki lima dimensi yang dapat diukur dalam hal layanan jasa servis, yaitu:

- *tangibles*, yaitu bukti fisik secara langsung. Bukti yang dapat dilihat oleh pelanggan penerima jasa servis dalam hal ini adalah pekerja yang datang dengan peralatannya ketika hendak memberikan layanan jasa.
- *reliability* atau keandalan, yaitu kemampuan pekerja dalam menyelesaikan permasalahan (terkait kerusakan barang elektronik yang akan diperbaiki).
- *responsiveness* atau tanggap, yaitu penilaian sikap pekerja yang cepat dan tepat dalam memberikan bantuan sesuai permintaan pelanggan.
- *assurance* atau jaminan, jaminan pekerja dalam hal ini adalah memberikan kepercayaan pelanggan kepada perusahaan (berkaitan juga dengan jaminan rasa aman saat pekerja masuk ke rumah pelanggan).
- *empathy*, yaitu perhatian tulus yang diberikan pekerja kepada pelanggan untuk memahami keinginan pelanggan.

2.3 HTML

HTML merupakan singkatan dari *Hypertext Markup Language*, dimana HTML adalah bahasa yang digunakan untuk membuat halaman web. HTML dibuat oleh Tim Berners-Lee dalam kolaborasinya dengan Robert Caillau ketika mereka bekerja di CERN-Institusi penelitian ahasa fisika di Geneva. HTML merupakan subset dari SGML (*Standard Generalized Markup language*) yang diperkenalkan oleh *International Standards* sebagai ISO pada 1986. Sejak saat itu, orang-orang

membuat dokumen dengan SGML dan menuliskan perintah (*tags* HTML) yang dapat bekerja untuk menampilkan isi dokumen dalam aturan tertentu.

Tujuan utama HTML adalah menampilkan data atau konten dalam cara yang memuaskan (Mercer, 2002). Dokumen yang dapat ditampilkan dengan HTML tak hanya berupa teks. Kini, HTML dapat menampilkan bentuk lain seperti gambar, video bahkan dalam format suara bukan hanya visual. Karakteristik tampilan HTML dapat juga ditambah maupun ditingkatkan dengan penggunaan CSS (*Cascading Style Sheet*) untuk mempercantik tampilan *website*.

2.4 PHP

PHP yang juga ditulis sebagai *PHP: Hypertext Preprocessor* adalah sebuah bahasa yang digunakan untuk pembuatan maupun pengembangan web. PHP pertama kali diciptakan oleh Rasmus Lerdorf pada 1994. PHP adalah bahasa yang fleksibel karena dapat digunakan bersamaan dengan HTML (Nixon, 2012). Pada mulanya PHP hanya didesain untuk pembuatan web personal jika dilihat dari namanya terdahulu yaitu *Personal Home Page*. Namun, karena perkembangannya kini PHP tak hanya digunakan untuk membuat *website* sederhana namun juga *website* yang memiliki fungsionalitas kompleks.

PHP dikenal sebagai bahasa pemrograman *server-side* karena PHP diproses pada server (Andre, 2014). Bahasa pemrograman yang bersifat *server-side* seperti PHP membuat pengolahan datanya dilakukan oleh server. Untuk itu, setiap kali halaman web dikunjungi oleh pengguna maka server atau penyedia akan mengirimkan data yang diminta oleh pengguna. Data yang diambil dari *database* tersebut nantinya akan ditampilkan pada halaman web.

2.5 MySQL

MySQL adalah *multiuser database* yang menggunakan bahasa *Structured Query Language* atau SQL (Bimo, 2002). *Multiuser* dapat berarti pengguna dalam jumlah sangat banyak dapat mengakses *database* MySQL dalam satu waktu. MySQL sebagai salah satu aplikasi dari RDBMS (*Relational Database Management System*) bersifat gratis dan *open source*. RDBMS merupakan program komputer yang dirancang untuk mengatur basis data sebagai sekumpulan data yang tersimpan secara terstruktur. Walaupun MySQL bersifat gratis namun ribuan *programmer* dari seluruh dunia telah mendukung perkembangannya sehingga menjadi aplikasi RDBMS yang lengkap dan cepat (Andre, 2017).

Beberapa keunggulan dari MySQL adalah MySQL memiliki kecepatan yang baik. Sebuah studi dari *eWeek* 2002 telah membandingkan performa kecepatan MySQL dengan RDBMS lainnya seperti *Microsoft SQL Server 2000*, *IBM DB2*, *Oracle 9i* dan *Sybase* yang hasilnya kecepatan MySQL lebih tinggi. Dari sisi keandalan (*reliability*), MySQL dapat diandalkan karena sifatnya yang *open source* jadi setiap orang dapat berkontribusi memeriksa *bug* dan melakukan pembenahan-pembenahan yang dirasa perlu. MySQL yang mendukung sistem

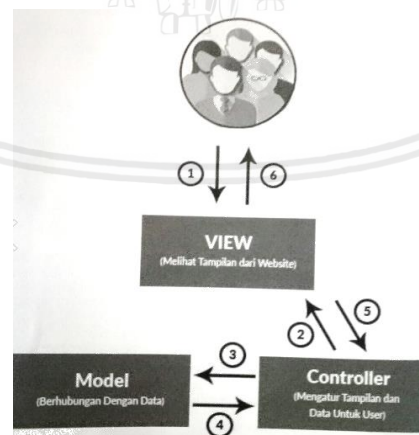
multi-processor juga menjadikan nilai tambah dengan memproses data yang sangat besar tanpa ada penurunan performa. Jika dilihat dari segi portabilitasnya, MySQL dapat berpindah dari satu sistem ke sistem lainnya seperti sistem operasi dengan mudah.

2.6 Konsep *Model, View, dan Controller* (MVC)

Konsep *Model, View, dan Controller* merupakan pola desain yang biasanya digunakan untuk mengembangkan perangkat lunak berorientasi objek dengan Bahasa PHP. Ketiga komponen ini tentunya memiliki tugas masing-masing dan saling berhubungan satu sama lain. Tugas *model, view, dan controller* dijelaskan lebih lanjut sebagai berikut (David, 2017):

- Model*, setiap objek dapat menjadi model. *Model* merupakan komponen yang berhubungan langsung dengan *database* untuk memanipulasi data yaitu melakukan operasi *create, read, update, dan delete*. Namun, *model* tidak dapat langsung berhubungan dengan *view*.
- View*, komponen yang mengelola *presentation logic*. Dalam aplikasi web, *view* biasanya berupa file template HTML yang diatur oleh *controller*. *View* juga berfungsi untuk menerima dan merepresentasikan data kepada pengguna.
- Controller*, komponen yang mengantarkan *request* dari pengguna lalu memerintahkan *model* dan/atau *view* untuk melakukan perubahan yang sesuai. *Controller* mengatur hubungan antara *model* dan *view* serta menentukan apa yang akan diproses oleh aplikasi.

Berikut adalah alur komunikasi dalam konsep MVC :



Gambar 2.1 Alur Konsep MVC

Sumber: David (2017)

2.7 CodeIgniter

CodeIgniter merupakan sebuah *framework* berbasis PHP yang kuat dengan *footprint* yang sangat kecil, dibangun untuk pengembang yang membutuhkan *toolkit* sederhana dan elegan untuk membuat aplikasi web dengan fitur lengkap. (David, 2017). Rick Ellis, seorang musisi rock yang beralih profesi menjadi *programmer* merupakan penulis *CodeIgniter* pertama kali. Riset kecil-kecilannya menghasilkan suatu *framework* yang ringan namun memenuhi fitur umum aplikasi. Akan tetapi, *CodeIgniter* dimiliki *British Columbia Institute of Technology* (BCIT) sejak 2014.

Beberapa kelebihan *CodeIgniter* sebagai *framework* yang populer saat ini adalah:

- a. Mudah dipelajari. Terdapat berbagai forum diskusi mengenai *framework CodeIgniter* yang dapat memecahkan berbagai masalah terutama untuk pengembang baru
- b. Ringan. Ukuran *file download CodeIgniter* tergolong sangat kecil yaitu hanya sekitar 2MB
- c. Memiliki dokumentasi yang lengkap
- d. Sedikit Konfigurasi. Saat memulai menggunakan *CodeIgniter*, konfigurasi awal yang diperlukan hanya mengganti beberapa bagian di bagian *application/config* yang tentunya tetap dapat berjalan meski tanpa melakukan konfigurasi apapun.
- e. Mendukung *database-database* paling umum termasuk MySQL

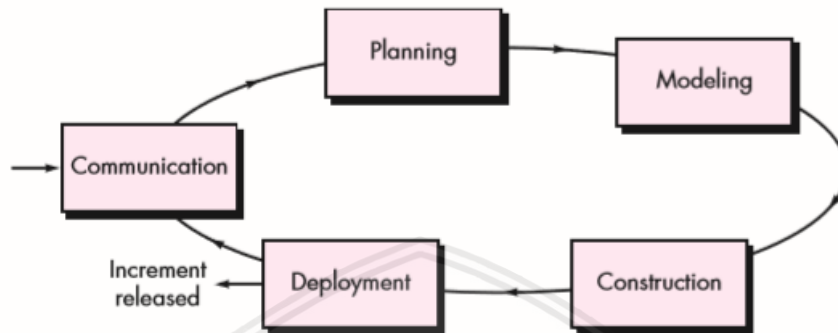
2.8 SMS Gateway

SMS atau *Short Messages Services* merupakan suatu layanan yang memungkinkan kita untuk bertukar informasi berupa pesan singkat melalui jaringan nirkabel telepon selular. Istilah *Gateway* dilihat dari artinya merupakan sebuah gerbang. Namun, pada dunia computer, *gateway* dapat diartikan sebagai jembatan penghubung antara satu sistem dengan sistem yang lain sehingga dapat terjadi pertukaran data antar sistem tersebut (Alfathin, 2018).

Kini layanan SMS tak hanya dapat digunakan melalui media telepon selular. Adanya *SMS Gateway* memungkinkan pengguna dapat mengirim, menerima, bahkan mengolah SMS melalui komputer dan perangkat lunak. Sehingga dalam dunia bisnis terutama yang berhubungan dengan pelanggan, SMS merupakan hal yang paling dapat diandalkan untuk mengirim informasi karena sudah pasti seluruh pengguna yang memiliki nomor telepon dan telepon selular pasti dapat menerima pesan berupa SMS.

2.9 Software Development Process

Dalam *software development*, untuk dapat mengembangkan perangkat lunak yang berkualitas tentunya melalui proses dan metode. Menurut Pressman (2010), pengembangan perangkat lunak dapat dilakukan dengan melalui lima aktifitas yang disebut “process framework”, yaitu:

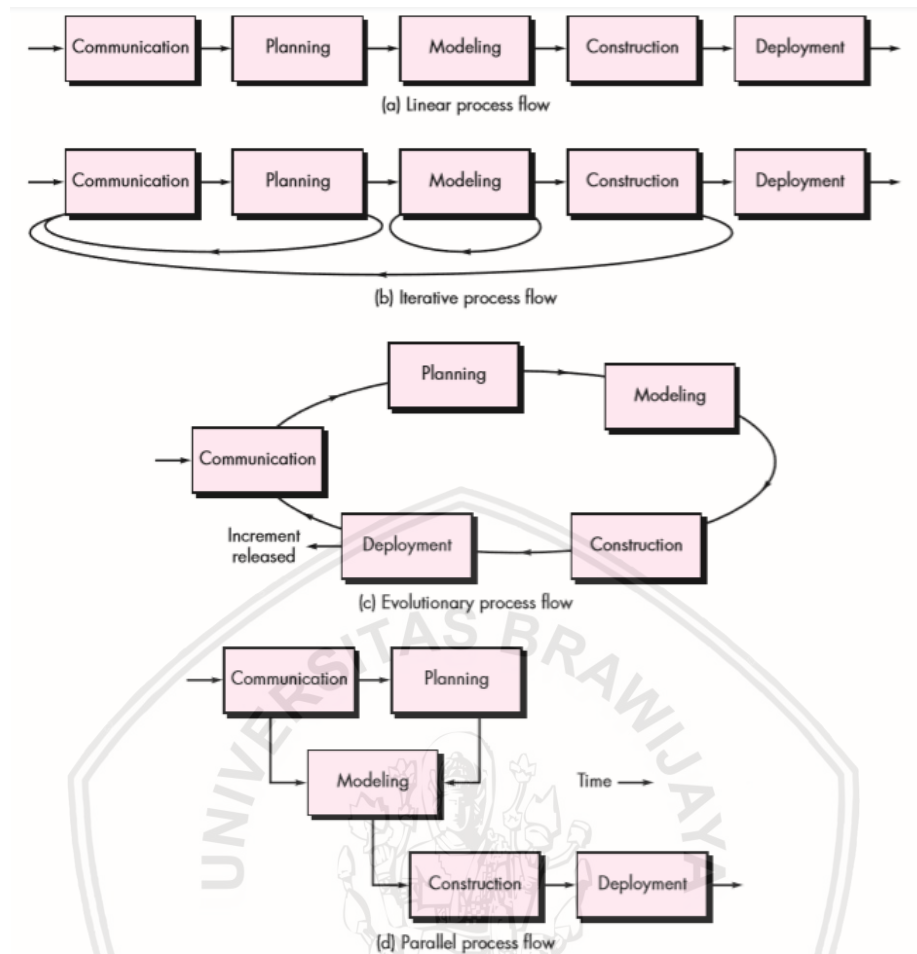


Gambar 2.2 Software Development Process

Sumber: Pressman (2010)

Proses dimulai dari *Communication* yang merupakan tahap terpenting untuk mengomunikasikan masalah dan berkolaborasi dengan konsumen atau para *stakeholder*. Tujuannya adalah memahami keinginan *stakeholder* terhadap *project* dan menggali kebutuhan yang dapat membantu mendefinisikan fitur-fitur maupun fungsional perangkat lunak nantinya. Proses kedua adalah *Planning*, yaitu perencanaan yang dapat menjadi acuan dalam pengembangan software nantinya.

Ketika aktivitas *planning* telah dijalankan, maka mulai memasuki tahap implementasi. Implementasi dalam lima *process framework* yang didefinisikan Pressman dimulai dengan *Modeling*. *Modeling* sangat erat kaitannya dengan pembuatan sketsa. Sketsa *project* dibuat untuk memahami gambaran secara keseluruhan. Sketsa yang dibuat akan diperbaiki seiring waktu sehingga menjadi lebih rinci untuk memahami lebih baik tentang masalah dan solusi pemecahannya. Proses berikutnya adalah *Constructions* yang merupakan aktivitas untuk mewujudkan model yang telah didefinisikan sebelumnya. *Constructions* mengombinasikan *code* dan *testing* untuk mengetahui *error* yang muncul. Proses terakhir adalah *Deployment*, dimana perangkat lunak diserahkan kepada *customer* yang mengevaluasi proyek dan memberikan *feedback* berdasarkan evaluasinya. Namun, proses deployment pada penelitian ini belum dilakukan sehingga *process framework* tersebut berakhir pada tahap *construction*. Kelima proses tersebut dapat diterapkan secara berulang hingga memperoleh proyek yang diinginkan (dalam beberapa jenis SDLC). Tentunya *process framework* ini dapat dilakukan secara *linear*, *iterative*, *evolutionary*, maupun *parallel flow*.



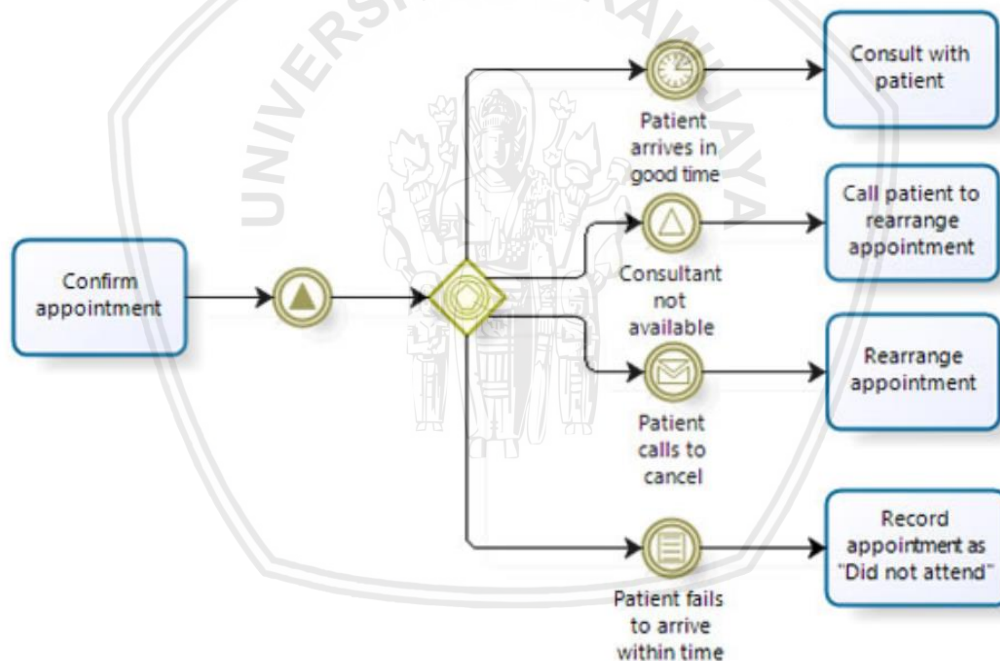
Gambar 2.3 Software Development Process Flow

Sumber: Pressman (2010)

Terdapat beberapa model pengembangan software yang dapat dipilih oleh pengembang sesuai dengan kebutuhan menurut Pressman (2010). Model *waterfall* atau *linier process flow* adalah salah satu model tradisional yang hanya dapat dipilih ketika kebutuhan sudah dipastikan di awal tanpa ada perubahan kebutuhan di tengah pengembangan sistem. Selain *waterfall* terdapat model *spiral* atau *iterative process flow* yang memperbolehkan proses pengembangan dilakukan berulang kali hingga iterasi tertentu dimana sistem yang diinginkan selesai dikembangkan. Jika *iterative process flow* melakukan perulangan di tiap tahapan dan dapat kembali ke tahap sebelumnya, maka model *revolutionary* melakukan perulangan per seluruh tahapan. Jadi, setelah seluruh tahapan dilalui, dapat berulang lagi dari awal berurutan sampai tahap terakhir. Model yang dipilih oleh penulis adalah model *waterfall* atau *linier process flow* karena seluruh kebutuhan *stakeholder* telah didefinisikan di awal dan tahap selanjutnya hanya akan dilakukan ketika tahap sebelumnya telah selesai. Selain itu model terakhir dalam pengembangan *software* adalah *parallel process flow* yang memungkinkan pengembangan sistem jangka pendek dapat dilakukan dimana satu tahapan dapat dilakukan berbarengan dengan tahap lain atau secara paralel.

2.10 Bussiness Process Model and Notation

Bussiness Process Model and Notation atau BPMN adalah sebuah standar yang digunakan untuk menggambarkan proses bisnis suatu sistem. Notasi ini diperkenalkan pertama kali oleh BPMI (*Business Process Management Institute*). Ketika BPMN diperkenalkan pada 2001, tidak terlihat perbedaan yang mendalam dengan notasi-notasi yang lain yang biasa digunakan untuk menggambarkan proses bisnis sebuah sistem. Namun, BPMN sebenarnya berkontribusi terhadap beberapa inovasi yang sangat berguna dalam seni memodelkan proses (Ditono, 2017). Inovasi yang dimaksud adalah jika selama ini pembuat model proses bisnis mengesampingkan *exception flows* seperti misalnya pelanggan membatalkan order, pasien melewatkan janji pertemuan, komponen yang hilang, atau stok material yang habis maka BPMN dapat mengatasinya. Pada BPMN, beberapa proses dapat diproses secara langsung dari *start* sampai *finish* tanpa dipengaruhi *event* lain. Dengan kata lain, proses-proses pada BPMN dapat digambarkan sebagai proses paralel dalam kondisi tertentu. Berikut adalah gambaran BPMN:



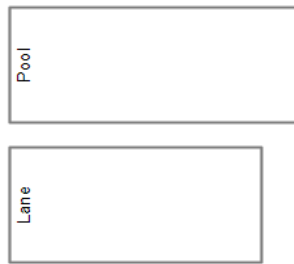
Gambar 2.4 Contoh BPMN

Sumber: www.bpmi-institute.org

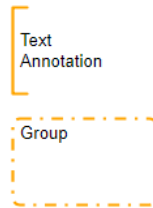
Dalam penggambarannya, BPMN memiliki banyak notasi untuk menggambarkan setiap proses seperti *gateways* yang disediakan untuk keperluan proses yang berjalan paralel, *exclusive* (hanya salah satu yang akan dijalankan), *inclusive* (proses yang dijalankan mengextend proses lain) dan *event*. Notasi selengkapnya dapat dilihat pada tabel berikut:

Tabel 2.2 Komponen BPMN

Participants



Artifacts



Gateways



Type	Start			Intermediate				End
	Normal	Event Subprocess	Event Subprocess non-interrupt	catch	boundary	boundary non-interrupt	throw	
None								
Message								
Timer								
Conditional								
Link								
Signal								
Error								
Escalation								
Termination								
Compensation								
Cancel								
Multiple								
Multiple Parallel								

Sumber: www.camunda.com

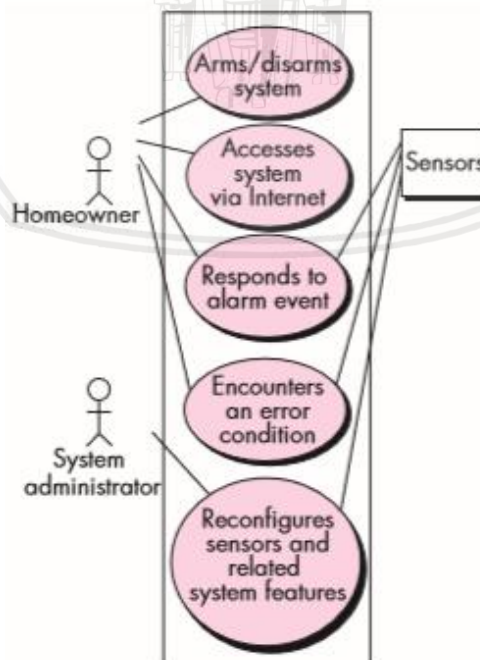
2.11 Usecase Diagram

Usecase Diagram adalah sebuah *diagram* yang dapat menggambarkan fungsionalitas dan fitur sistem dilihat dari perspektif pengguna (Pressman, 2010). Fungsionalitas berarti hal-hal yang dapat dilakukan oleh sistem. Gambaran singkat yang diberikan dituangkan dalam bentuk *usecase* dengan penggunaan kata kerja yang diikuti oleh aktor-aktor seperti pengguna dan sistem lain yang dapat melakukan pekerjaan sesuai fungsionalitas sistem. Selain komponen aktor dan *usecase*, terdapat hubungan keduanya yang dinamakan relasi. Relasi yang berupa garis tersebut dapat digambarkan antar aktor maupun antara aktor dan *usecase*. Penjelasan dari komponen-komponen *usecase diagram* dapat dilihat pada tabel berikut:

Tabel 2.3 Komponen Usecase Diagram

Komponen	Keterangan
	Aktor
	<i>Usecase</i>
	<i>Association Relationship</i>
	<i>Include and Extends Relationship</i>
	<i>Generalization Relationship</i>

Secara sederhana, ilustrasi *usecase diagram* digambarkan sebagai berikut:



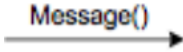
Gambar 2.5 Contoh Usecase Diagram

Sumber : Pressman (2010)

2.12 Sequence Diagram

Sequence Diagram adalah diagram yang digunakan untuk menggambarkan komunikasi dinamis antar objek selama eksekusi tugas (Pressman, 2010). Seperti namanya, *sequence* menunjukkan urutan maupun rangkaian pesan yang dikirim antara objek satu dengan yang lain. Pertukaran pesan tersebut juga dapat dikatakan sebagai interaksi antar objek. Penamaan objek pada *sequence diagram* ini pun terdapat beberapa cara, diantaranya adalah hanya menamai dengan nama objek, hanya menamai dengan nama *class*, dan menamai dengan nama objek dan *class*. Sementara pesan yang dikirim dapat berupa *method*. Banyaknya objek pada *sequence diagram* biasanya sesuai dengan banyaknya *usecase* pada *usecase diagram*. Beberapa komponen yang terdapat pada *sequence diagram* antara lain:

Tabel 2.4 Komponen *Sequence Diagram*

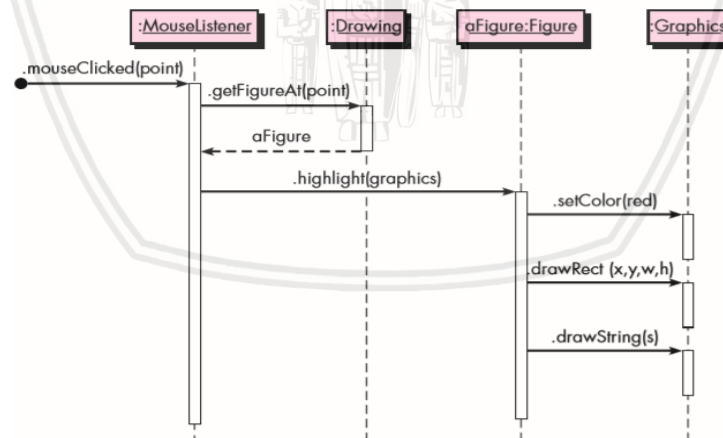
Komponen	Keterangan
	<i>Boundary</i>
	<i>Entity</i>
	<i>Controller</i>
	<i>Actor</i>
	<i>Lifeline</i>
	<i>Activation Boxes</i>
	<i>Message</i>

Sumber: www.creately.com

- *Boundary* :
lingkungan sebagai jembatan antara sistem dan dunia luar. *Boundary* dapat berupa *interface system* seperti layar atau halaman - halaman sistem.
- Entitas :
entitas dapat berupa kelas - kelas yang sesuai dengan *usecase* pada *usecase diagram*.

- **Controller** :
penghubung antara *boundary* dan entitas untuk memastikan aliran pesan terkontrol.
- **Aktor** :
komponen aktor yang digambarkan sama dengan aktor pada *usecase diagram* ini mewakili pengguna yang berinteraksi dengan sistem.
- **Lifeline** :
komponen yang membentuk garis putus-putus sebagai garis hidup sebuah objek. Garis hidup disini berarti objek tersebut dapat aktif melakukan sesuatu.
- **Activation boxes** :
komponen berupa kotak yang menempel pada *lifeline*. Ketika terdapat *activation boxes* pada *lifeline*, maka objek tersebut sedang aktif melakukan tugas.
- **Message** :
pesan yang menunjukkan komunikasi yang terjadi antar objek berisi informasi maupun pemanggilan aktivitas.

Berikut ilustrasi dari *sequence diagram* :



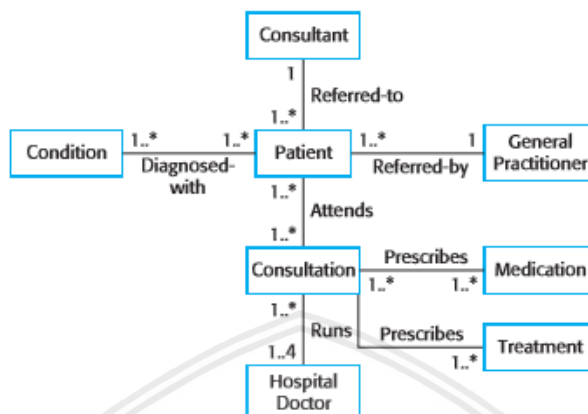
Gambar 2.6 Contoh Sequence Diagram

Sumber : Pressman (2010)

2.13 Class Diagram

Class Diagram digunakan ketika mengembangkan model sistem *object oriented* untuk menampilkan *class-class* dalam sistem dan hubungan antar *class* (Somerville, 2011). Hubungan atau asosiasi yang didefinisikan antar *class* memberikan konsekuensi bahwa setiap *class* yang berhubungan harus memiliki

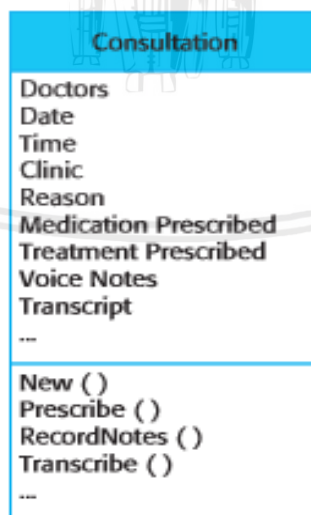
beberapa pengetahuan tentang *class* yang berasosiasi dengannya. Ketika membangun sebuah *class diagram* dalam proses pengembangan *software*, objek-objek yang ada merepresentasikan sesuatu di dunia nyata yang berkaitan dengan sistem. Berikut adalah contoh dari penggambaran *class diagram* :



Gambar 2.7 Contoh Class Diagram

Sumber: Sommerville (2011)

Gambar *class diagram* tersebut hanya menggambarkan nama *class* dan asosiasinya. Hal inilah yang dimaksud penggambaran *class* berdasarkan objek di dunia nyata yaitu *Patient*, *Condition*, *Consultant*, dll. Sementara hubungan atau asosiasi yang digambarkan adalah 1...* yang berarti *one-to-many* misalnya antara *patient* dan *condition* memiliki hubungan 1...* yang dapat diartikan seorang pasien dapat didiagnosa dengan banyak kondisi. Jika dirinci lagi, satu *class* dalam *class diagram* dapat digambarkan sebagai berikut :



Gambar 2.8 Class Diagram Detail

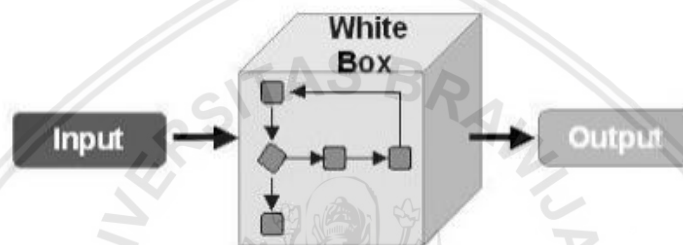
Sumber: Sommerville (2011)

repository.ub.ac.id

Dalam *class diagram*, *class consultation* memiliki asosiasi dengan beberapa *class* lain. Untuk itulah dalam *class Consultation*, terdapat pengetahuan tentang *class-class* yang berasosiasi dengannya.

2.14 White-Box Testing

Pengujian *white-box* atau terkadang disebut juga *glass-box testing* merupakan salah satu metode pengujian perangkat lunak. Menurut Pressman (2010), kasus pengujian yang dapat diperoleh dengan metode *white-box testing* antara lain (1) menjamin bahwa semua jalur independen dalam modul telah dilakukan setidaknya sekali; (2) menjalankan semua keputusan logis pada sisi yang benar dan salah; (3) menjalankan semua perulangan pada batas-batasnya dan dalam batas operasionalnya; serta (4) melaksanakan struktur data internal untuk memastikan validitasnya.

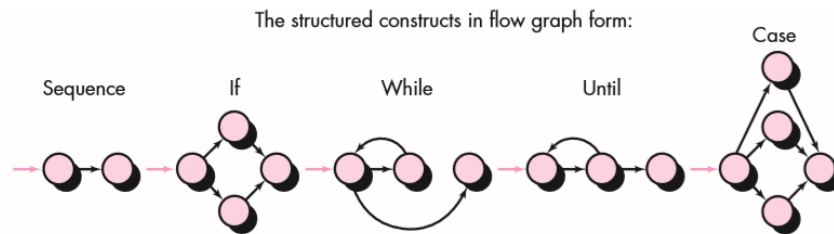


Gambar 2.9 White-box Testing

Sumber: www.medium.com

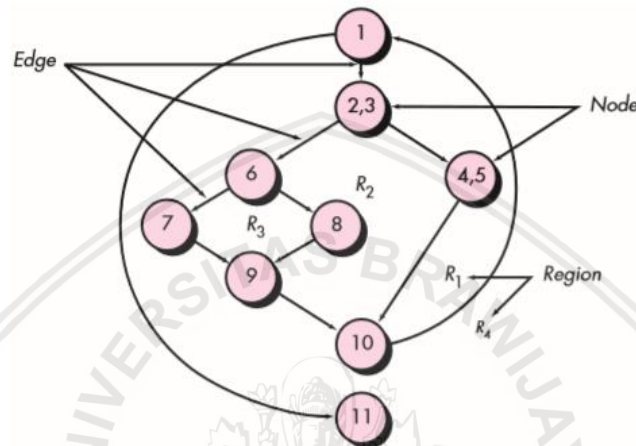
Teknik pengujian *white-box* dapat dilakukan dengan beberapa cara yang dijelaskan lebih lanjut oleh Pressman (2010). *Basis path testing* yang merupakan salah satu dari sejumlah teknik untuk pengujian struktur kontrol (*control structure testing*) adalah metode pengujian yang memungkinkan perancang kasus uji untuk mendapatkan ukuran kompleksitas dari desain prosedural dan menggunakan ukuran tersebut sebagai panduan untuk mendefinisikan set dasar jalur eksekusi (Pressman, 2010). Pengujian juga dapat dilakukan di tingkat unit atau blok kode (*unit testing*), integrasi beberapa unit (*integration testing*), keseluruhan sistem (*system testing*), serta pengujian beta oleh pengguna sebenarnya (*acceptance testing*). Namun, dalam penelitian ini penulis hanya melakukan pengujian di tingkat unit yaitu pengujian terhadap method.

Dalam teknik *basis path testing*, terdapat sebuah notasi untuk merepresentasikan aliran kontrol yang dinamakan *flow graph*. *Flow graph* ini dapat menggambarkan satu atau lebih kalimat prosedural. *Flow graph* mirip dengan *flowchart* namun beberapa kalimat yang menggambarkan proses dapat digabung menjadi satu *node* dalam *flow graph*. *Flow graph* terdiri dari *node* (merepresentasikan kalimat prosedural), *edge* atau *link* (merepresentasikan aliran kontrol), dan *region* (area yang dibatasi oleh *node* dan *edge* juga area diluar graf).



Gambar 2.10 Struktur Flow Graph

Sumber: Pressman (2010)

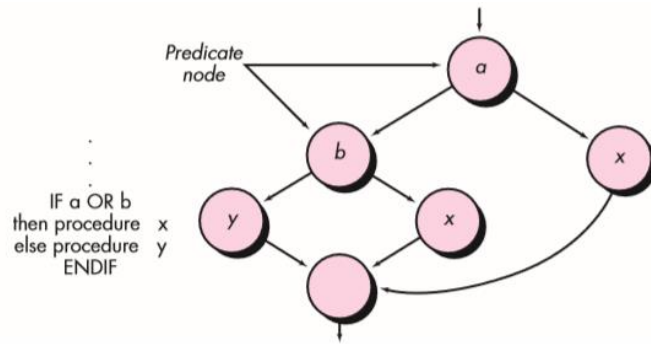


Gambar 2.11 Contoh Flow Graph

Sumber: Pressman (2010)

Flow graph biasanya menggambarkan aliran kontrol sebuah unit yaitu *method* dalam program perangkat lunak. Ketika *flow graph* telah digambarkan dan seluruh komponen telah diketahui, nilai *cyclomatic complexity* yang menunjukkan jumlah jalur *testing* dapat dihitung melalui tiga cara:

1. Banyaknya *region* dalam *flow graph* sama dengan nilai *cyclomatic complexity* atau $V(G) = R$
2. Perhitungan berdasarkan jumlah *node* dan *edge* menunjukkan nilai *cyclomatic complexity*, yaitu $V(G) = E - N + 2$
3. *Cyclomatic complexity* dihitung berdasarkan nilai P dimana P adalah predikat *node* (*node* yang memiliki *edge* mengarah keluar lebih dari satu) yaitu $V(G) = P + 1$



Gambar 2.12 Predicate Node dalam Flow Graph

Sumber: Pressman (2010)

Nilai $V(G)$ dapat menjadi acuan untuk memperoleh banyaknya jalur pengujian perangkat lunak.

2.15 Black-Box Testing

Black-Box Testing juga disebut *behavioral testing* merupakan pengujian yang berfokus pada pengujian fungsional sistem. Pengujian *black-box* mencoba menemukan kesalahan dalam kategori berikut: (1) fungsi salah atau hilang; (2) kesalahan antarmuka; (3) kesalahan dalam struktur data atau akses basis data eksternal; (4) kesalahan perilaku atau kinerja; dan (5) kesalahan inisialisasi dan terminasi (Pressman, 2010). Untuk itu, *black-box testing* dilakukan untuk menguji validitas fungsional, perilaku kinerja sistem, *test case input*, batas kelas data, kecepatan dan volume data yang dapat ditoleransi sistem, dan pengaruh kombinasi data tertentu terhadap operasi sistem.



Gambar 2.13 Black-box Testing

Sumber: www.medium.com

Salah satu metode pengujian yang tergolong ke Black-box testing adalah pengujian validasi. Validasi merupakan konfirmasi melalui pengujian dan penyediaan bukti objektif bahwa persyaratan tertentu untuk suatu maksud khusus dipenuhi (Standar ISO 9000:2005). Sehingga melalui pengujian validasi dalam *Black-box testing*, penguji dapat mengetahui seluruh kebutuhan yang didefinisikan valid (dapat berjalan dengan baik sesuai yang didefinisikan) atau sebaliknya.

2.16 Usability Testing

Usability atau daya guna merupakan sebuah kualitas atribut yang menilai seberapa mudah sistem digunakan khususnya pada user interface. Terdapat lima elemen usability yang terpenting oleh Nielsen, antara lain:

- Learnability, penilaian terhadap seberapa bisa pengguna menggunakan fungsi-fungsi dasar yang ada pada aplikasi saat pertama kali menggunakan.
- Efficiency, penilaian terhadap seberapa cepat pengguna mempelajari cara pengoperasian sistem setelah melihat desain aplikasi yang dibuat.
- Memorability, penilaian terhadap seberapa mudah pengguna mengingat fungsi dalam sistem setelah tidak menggunakannya selama periode tertentu.
- Errors, penilaian terhadap kesalahan yang dilakukan pengguna saat mengoperasikan sistem dan bagaimana pengguna dapat memperbaiki kesalahan.
- Satisfaction, penilaian terhadap seberapa puas pengguna menggunakan sistem dan merasa nyaman terhadapnya.

Pengujian usabilitas ini dapat dilakukan dengan memberikan pertanyaan-pertanyaan kepada pengguna sesuai dengan kelima elemen usabilitas sebagai acuan pembuatan kuesioner. Usability cukup penting untuk mengetahui respon pengguna terhadap sistem yang telah dikembangkan. Jika suatu sistem sulit digunakan, maka orang-orang akan meninggalkannya. Agar hal tersebut tidak terjadi, sebuah sistem harus memiliki usabilitas yang baik.

2.17 Compatibility Testing

Compatibility, seperti namanya, berarti kesesuaian. Compatibility testing merupakan pengujian kesesuaian sistem dengan lingkungannya. Lingkungan dalam hal ini dapat menjadi apa saja seperti perangkat keras, sistem operasi, aplikasi, maupun jaringan yang berbeda. Sehingga dapat disimpulkan pengujian kompatibilitas bertujuan untuk menguji apakah sistem dapat berjalan pada lingkungan yang berbeda atau sebaliknya. Semakin banyak lingkungan tempat sistem dapat berjalan artinya sistem yang telah dikembangkan semakin baik.

Dalam penelitian ini dilakukan uji kompatibilitas pada aplikasi yaitu browser sebagai tempat sistem dapat dijalankan. Pengujian kompatibilitas ini menggunakan tools SortSite. Untuk itu terdapat beberapa *browser* yang digunakan dalam pengujian ini antara lain:

Tabel 2.5 Versi Browser pada Pengujian *Compatibility*

No.	Web Browser	Versi
1	Internet Explorer	11
2	Microsoft Edge	18
3	Firefox	66
4	Safari	12
5	Opera	60
6	Chrome	74
7	iOS	<10, 10, 11 dan 12
8	Android	<3, 3, dan 4

Dalam melakukan pengujian, terdapat indikator hasil pengujian *compatibility* oleh SortSite yang dijabarkan dalam table berikut:

Tabel 2.6 Indikator Permasalahan *Compatibility*

Simbol	Keterangan
	Tidak terdapat kendala pada <i>browser</i>
	Minor issues atau terdapat masalah ringan pada konten atau permasalahan performa
	Major issues atau terdapat masalah mayor pada konten atau permasalahan performa
	Critical issues merupakan permasalahan yang fatal karena terdapat konten atau fungsi yang hilang pada <i>browser</i>

BAB 3 METODOLOGI

Pada Bab Metodologi akan dibahas mengenai langkah-langkah dalam melakukan penelitian dimulai dari identifikasi masalah yang dilanjutkan dengan studi literatur, analisis kebutuhan dari *stakeholder*, perancangan hingga implementasi sistem serta pengujian sistem. Kemudian kesimpulan dan saran akan digunakan sebagai catatan implementasi sistem lebih lanjut ataupun sistem yang serupa di kemudian hari. Alur proses pengembangan sistem dilakukan dengan metode *linear process flow* atau *waterfall* yang disesuaikan dengan alur penelitian skripsi berikut:



Gambar 3.1 Alur Metode Penelitian

3.1 Identifikasi Masalah di Lapangan

Dalam proses mengidentifikasi masalah di lapangan, penulis menemukan beberapa kekurangan dari proses bisnis yang selama ini dijalani secara manual baik oleh kepala UKM maupun pekerja lapangan. Beberapa permasalahan diantaranya diungkapkan sendiri oleh kepala UKM sebagai pengelola usaha maupun kendala yang dialami pekerja itu sendiri. Dari seluruh proses yang terjadi sehari-hari, kendala yang paling terlihat adalah pencatatan laporan pekerjaan harian dan jadwal pekerjaan yang tidak terorganisir dengan baik sehingga berimbas kepada rekap penghasilan usaha. Agenda pekerjaan yang diberikan setiap pagi oleh kepala UKM pun sering kali tidak terselesaikan dalam satu hari sehingga perlu dimaksimalkan. Pelayanan yang maksimal tentunya berpengaruh kepada berkurangnya keluhan dari pelanggan serta pencatatan keluhan pelanggan menjadi suatu hal yang penting dalam memelihara usaha mengingat usaha ini bergerak dominan di bidang jasa.

3.2 Studi Literatur

Pada tahap studi literatur menjelaskan tentang dasar teori yang digunakan sebagai penunjang dalam pengembangan sistem manajemen UKM Bali Tekindo Jaya. Sejauh ini penulis telah mempelajari literatur terkait permasalahan yang diangkat dalam skripsi ini. Teori-teori pendukung tersebut telah dijelaskan dalam Bab Tinjauan Pustaka, antara lain:

- HTML
- PHP
- MySQL
- Konsep MVC
- *CodeIgniter*
- *SMS Gateway*
- *Software Development Process*
- *Business Process Model and Notation*
- *Usecase Diagram*
- *Sequence Diagram*
- *Class Diagram*
- *White-Box Testing*
- *Black-Box Testing*
- *Usability Testing*
- *Compatibility Testing*

3.3 Analisis Kebutuhan

Analisis kebutuhan yang masuk pada tahap *Communication* dalam *Linear Process Flow* terkait kebutuhan sistem oleh pemangku kepentingan yang diperoleh dengan melakukan wawancara di lapangan menghasilkan daftar kebutuhan yang diutarakan para *stakeholder* yaitu kepala UKM dan para pekerja lapangan yang berhubungan langsung dengan pelanggan. Analisis kebutuhan ini dilakukan dengan menggali informasi terkait proses yang selama ini berjalan kemudian merangkum kendala yang selama ini ada untuk merancang sistem manajemen yang lebih baik.

Kebutuhan sistem yang telah dirangkum akan dijadikan acuan dalam perencanaan pembuatan sistem baru yang diselaraskan dengan proses yang selama ini berjalan. Kebutuhan-kebutuhan yang terekam ditulis menjadi daftar kebutuhan yang dipastikan kembali tidak ada yang konflik baik dari pihak perancang maupun para pengguna sistem. Fase penulisan daftar kebutuhan ini dilakukan berkali-kali hingga pengguna sistem merasa cukup bahwa kebutuhan yang telah diutarakan dimengerti oleh perancang sistem. Serta tidak ada kesalahpahaman antara kedua pihak dalam perancangan sistem. Sehingga kebutuhan tersebut dapat dispesifikasikan serta dimodelkan dengan kebutuhan *usecase* dan *scenario*-nya.

3.4 Perancangan

Tahap perancangan yang termasuk ke dalam *Planning* dan *Modelling* dalam *Linear Process Flow* berangkat dari seluruh kebutuhan yang telah didefinisikan pada tahap analisis kebutuhan. Perancangan mulai dilakukan dengan membuat diagram-diagram untuk memudahkan visualisasi gambaran sistem dan lebih mudah dimengerti oleh *stakeholder*. *Diagram-diagram* tersebut antara lain *usecase diagram* sebagai *diagram* yang menggambarkan fungsionalitas dan fitur sistem dilihat dari perspektif pengguna, *sequence diagram* sebagai *diagram* yang menggambarkan komunikasi dinamis antar objek selama eksekusi tugas, serta *class diagram* sebagai *diagram* yang menampilkan *class-class* dalam sistem dan hubungan antar *class*. Tentunya seluruh fungsi dalam perancangan ini diikuti dengan penjelasan skenario yang jelas.

Diagram-diagram yang telah dibuat dapat menjadi acuan dalam perancangan algoritme sistem. Sementara perancangan antarmuka yang digambarkan nantinya mengandung fungsionalitas-fungsionalitas sistem. Tersedianya rancangan antarmuka sistem akan memudahkan pembaca mengetahui apa saja yang nantinya dapat dilakukan oleh sistem dengan melihat menu-menu yang tersedia maupun form-form yang disuguhkan sebelum benar-benar diimplementasikan dalam kode program.

3.5 Implementasi

Implementasi sistem yang masuk ke dalam tahap *Construction* pada *Linear Process Flow* merupakan kode program yang dibuat berdasarkan rancangan sistem sebelumnya. Implementasi dibuat sesuai dengan hasil purwarupa sistem yang telah disetujui oleh *stakeholder*. Dalam implementasi, seluruh fungsi yang telah didefinisikan pada analisis kebutuhan harus tersedia dalam program aplikasi dan dapat dijalankan dengan baik. Implementasi rancangan sistem dibuat dalam bentuk web yang diprogram dengan *CodeIgniter* dengan tambahan *SMS Gateway* sebagai media untuk mengirim notifikasi sistem ke pelanggan dan notifikasi ke admin sebagai aktor sistem. Hasil implementasi kode program nantinya akan diuji dengan beberapa metode pengujian.

Sistem yang diimplementasikan memiliki beberapa fitur yang dapat dijalankan masing-masing pengguna. Fitur *booking*, penilaian kinerja, dan testimoni dapat digunakan oleh pelanggan, fitur agenda kerja dan laporan pekerjaan dapat diakses oleh pekerja, sedangkan admin dapat mengakses mulai dari *booking*, testimoni, agenda kerja, pelaporan, hingga penilaian kinerja oleh pelanggan. Fitur tersebut tentunya memiliki beberapa fungsi di dalamnya yang akan diperjelas dalam Bab Perancangan dan Implementasi.

3.6 Pengujian dan Analisis

Pengujian rancangan sistem , masih dalam tahap *Construction* pada *Linear Process Flow*, dilakukan untuk memastikan hal-hal yang telah didefinisikan dalam kebutuhan sistem terdapat pada hasil implementasi program aplikasi. Pengujian sistem bertujuan menemukan kesalahan-kesalahan yang terjadi pada sistem sehingga dapat dilakukan perbaikan. Pengujian perangkat lunak memiliki sejumlah sasaran penting, yaitu pengujian dilakukan untuk dapat menemukan kesalahan-kesalahan pada sistem, menemukan kesalahan yang belum pernah ditemukan sebelumnya, dan memberikan kasus uji yang mempunyai probabilitas tinggi untuk menemukan kesalahan. Untuk itu, pengujian hasil implementasi program aplikasi akan dilakukan dengan *white-box testing* untuk setiap *method* agar dapat diukur kompleksitasnya pada tingkat unit (unit testing) dan *black-box testing* dalam memastikan seluruh fungsi berjalan sesuai kebutuhan sistem yang telah didefinisikan di analisis kebutuhan melalui jenis pengujian validasi. Dalam *white-box testing* digunakan teknik *Basis Path Testing* untuk mendapatkan ukuran kompleksitas struktural dari kode program. Selain itu, terdapat pengujian usabilitas untuk memperoleh respon pengguna terhadap penggunaan sistem serta kompatibilitas untuk memastikan sistem dapat berjalan di lingkungan browser yang berbeda. Tentunya seluruh pengujian yang dilakukan berdasarkan kasus uji yang telah direncanakan. Kemudian, hasil pengujian akan dianalisis untuk dapat mengetahui capaian maupun kekurangan sistem yang telah dibangun.

3.7 Penarikan Kesimpulan

Tentunya dari pengujian sistem yang telah dibuat akan menghasilkan sebuah kesimpulan terkait hasil pengujian tersebut. Hasil perancangan yang telah dilakukan tidak menutup kemungkinan memiliki kekurangan. Hal-hal yang telah dicapai dan hal-hal yang belum atau kurang dalam pengembangan sistem tersebut akan dipaparkan dalam kesimpulan. Demi pengembangan sistem yang lebih baik di kemudian hari dan demi perancangan yang lebih baik terhadap sistem yang serupa perlu dilakukan penarikan kesimpulan dengan saran untuk memastikan perancangan yang dibuat sesuai harapan.

BAB 4 REKAYASA KEBUTUHAN

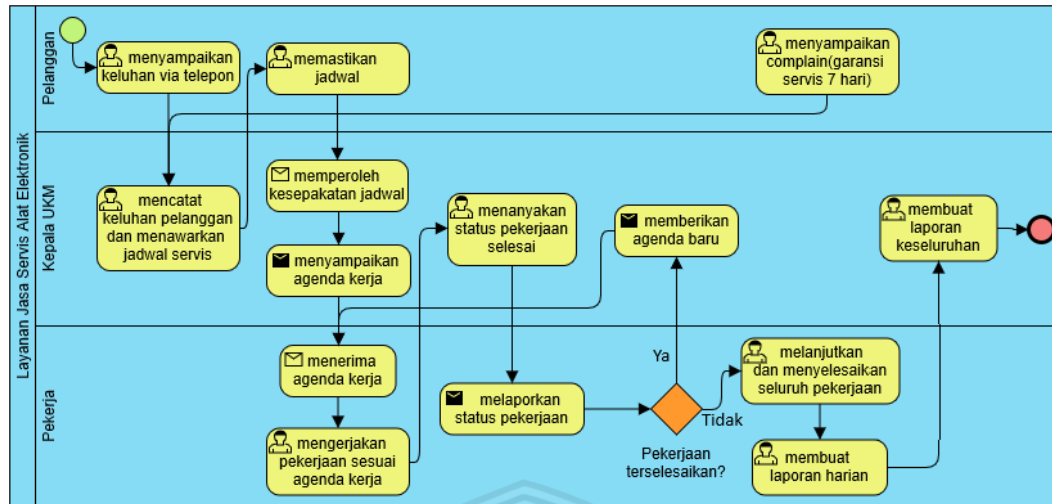
Pada Bab Rekayasa Kebutuhan akan dibahas mengenai proses penggalian kebutuhan oleh para *stakeholder* baik melalui wawancara maupun pengamatan langsung di lapangan. Rekayasa kebutuhan tersebut mencakup deskripsi umum sistem, identifikasi aktor yang berperan dalam sistem hingga kebutuhan perangkat lunak baik secara fungsional maupun non fungsional berikut spesifikasinya. Seluruh kebutuhan yang telah direkam akan dimodelkan dengan *usecase diagram* dan *usecase* .

4.1 Gambaran Umum Sistem

Selama ini, manajemen usaha di UKM Bali Tekindo Jaya dilakukan oleh kepala UKM secara manual. Dimulai dari pemesanan layanan jasa servis yang dilakukan pelanggan melalui sambungan telepon, pelanggan berkomunikasi langsung dengan kepala UKM dan membuat janji servis yang disepakati. Dalam menentukan waktu servis pelanggan, tak jarang kepala UKM membuat jadwal servis yang bertabrakan dengan pelanggan lain karena harus mengingat antrian servis lainnya. Hal ini menimbulkan kekecewaan pelanggan dan mempengaruhi loyalitas pelanggan terhadap UKM.

Antrian servis yang dipesan oleh pelanggan diagendakan kepada pekerja lapangan setiap pagi sebelum pekerja berangkat ke rumah-rumah pelanggan. Agenda ini disampaikan kepala UKM secara lisan sehingga pekerja pun wajib mengingat seluruh agenda yang disampaikan seperti nama pelanggan, alamat hingga keluhan servis pelanggan. Terkadang terjadi kesalahpahaman keluhan yang disampaikan oleh kepala UKM maupun kesalahpahaman pekerja dalam memahami keluhan tersebut sehingga layanan yang diberikan kepada pelanggan tidak tepat sasaran.

Pelaporan pekerjaan di UKM Bali Tekindo Jaya dibuat berdasarkan pekerjaan harian pekerja. Masing-masing pekerja memiliki satu buku laporan yang diisi setiap hari. Buku laporan masing-masing pekerja disalin oleh kepala UKM ke buku besar usaha milik kepala UKM. Hal ini menyebabkan terdapat duplikat laporan. Sementara itu, tidak terdapat sistem penilaian kinerja secara khusus kepada pekerja UKM. Performa kerja setiap pekerja hanya berdasarkan asumsi atau subjektif kepala UKM. Adapun ulasan pelanggan terkait servis disampaikan secara mulut ke mulut yang mendatangkan pelanggan baru bagi UKM. Penggambaran sistem manajemen UKM yang selama ini berjalan dapat dilihat pada diagram berikut:



Gambar 4.1 BPMN (as is) UKM Bali Tekindo Jaya

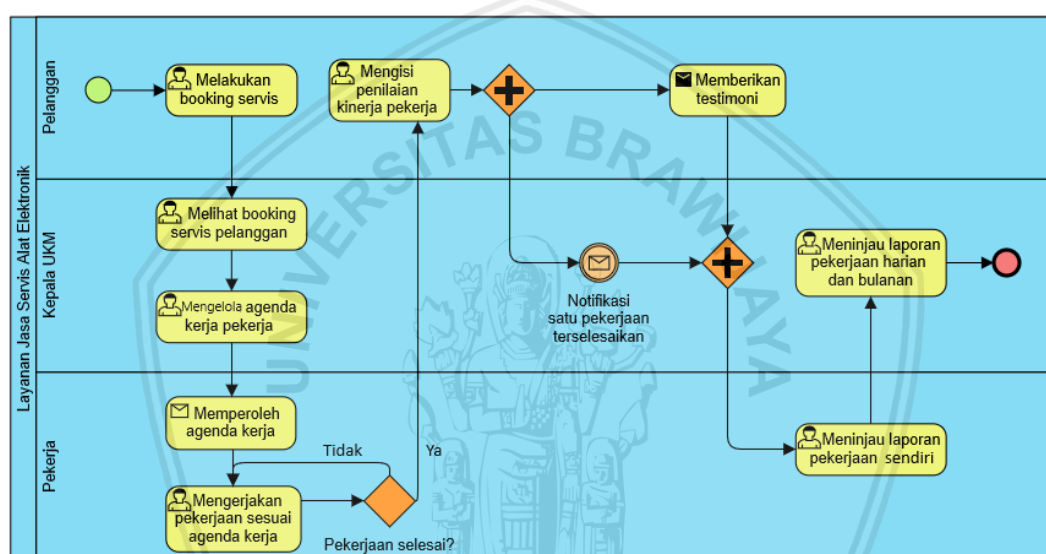
Sistem manajemen layanan jasa servis alat elektronik rumah tangga di UKM Bali Tekindo Jaya merupakan sistem untuk membantu mengelola manajemen usaha. Pengelolaan usaha yang selama ini dilakukan secara manual oleh kepala UKM dapat digantikan dengan hadirnya sistem manajemen ini. Sistem manajemen ini dapat memudahkan kepala UKM untuk mengatur agenda pekerjaan sehari-hari, memonitor kinerja pekerja di lapangan, mencatat keluhan dan ulasan pelanggan serta memperoleh laporan pekerjaan bulanan. Pelanggan setia UKM juga dapat melakukan *booking* layanan dan memberikan testimoni bagi UKM.

Pengguna sistem layanan servis elektronik ini tentunya dari pihak kepala UKM sebagai administrator yang dapat mengakses seluruh data maupun informasi pada sistem. Pekerja UKM adalah pengguna layanan berikutnya untuk dapat melihat agenda pekerjaan yang diberikan padanya. Selain itu, pekerja UKM juga dapat mengelola laporan kerja yang telah diselesaikan agar tercatat ke dalam sistem. Sementara dari sisi pelanggan, sistem manajemen ini dapat membantu pelanggan melakukan *booking* servis dan menuliskan testimoni. Untuk dapat masuk dan mengakses fitur-fitur layanan sistem, tentunya masing-masing pengguna sistem diharuskan untuk memasukkan *username* dan *password* yang diperoleh dengan melakukan daftar ke sistem.

Dalam pemesanan layanan servis alat elektronik, setelah berhasil login ke dalam sistem, pelanggan dapat memesan layanan dengan memasukkan data pelanggan yang terkait dengan tempat dan waktu layanan. Layanan yang dipesan adalah jasa perbaikan alat elektronik di rumah pelanggan. Data pesanan pelanggan yang dikelompokkan sesuai tanggal pemesanan dapat dilihat oleh pekerja UKM sebagai agenda kerja harian. Setiap satu pekerjaan terselesaikan, pekerja harus meminta pelanggan untuk mengisi penilaian kinerja berupa kuesioner penilaian. Penilaian kinerja yang telah disubmit oleh pelanggan menjadi acuan waktu terselesaikannya pekerjaan sekaligus pemberitahuan kepada kepala UKM bahwa satu pekerjaan telah selesai.

Pemberitahuan penyelesaian pekerjaan otomatis dikirim oleh sistem berupa pesan singkat kepada kepala UKM. Kepala UKM menerima pemberitahuan untuk mengetahui waktu pekerja menyelesaikan pekerjaan sehingga dapat memadatkan pekerjaan yang diambil dalam satu hari sehingga performa pekerja sesuai dengan upah yang dibayarkan.

Seluruh pekerjaan servis yang tercatat dalam sistem oleh masing-masing pekerja dapat terangkum menjadi satu laporan. Laporan yang terekam dapat dikelola oleh kepala UKM untuk ditambahkan data pendapatan. Laporan juga dapat dikelompokkan setiap bulan. Berdasarkan laporan yang ada, kepala UKM dapat mengevaluasi perkembangan usaha dan meningkatkan hal yang perlu ditingkatkan untuk dapat tetap bersaing dengan pebisnis lainnya. Penggambaran sistem yang akan dikembangkan dapat dilihat pada *diagram* berikut:



Gambar 4.2 BPMN (*to be*) UKM Bali Tekindo Jaya

4.2 Identifikasi Aktor

Pada bagian identifikasi aktor akan dijelaskan tentang peran aktor dalam menggunakan sistem. Berikut adalah peran aktor dalam sistem manajemen layanan jasa UKM:

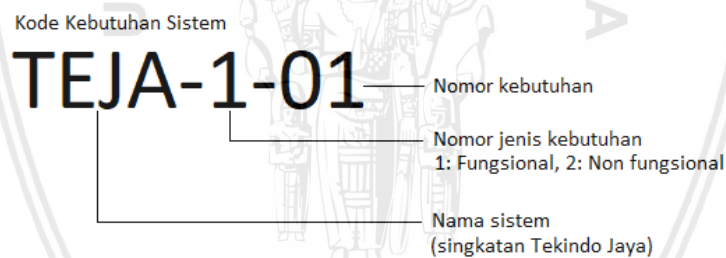
Tabel 4.1 Identifikasi Aktor dalam Sistem

No.	Aktor	Deskripsi
1.	Pelanggan	Pelanggan merupakan aktor yang berperan dalam sistem untuk melakukan <i>booking</i> servis, memberi penilaian terhadap pekerja, memberi testimoni, dan menerima notifikasi dari sistem. Untuk dapat menggunakan layanan sistem, pelanggan perlu melakukan pendaftaran serta login menggunakan <i>username</i> dan <i>password</i> .

2.	Pekerja	Pekerja berperan sebagai aktor yang bekerja langsung di lapangan. Dalam sistem, pekerja dapat melihat agenda kerja yang telah dijadwalkan untuknya serta melaporkan pekerjaan yang telah dikerjakan. Pekerja tentu memiliki <i>username</i> dan <i>password</i> untuk dapat masuk ke sistem.
3.	Admin	Admin dalam sistem memegang peran manajemen seperti mengatur agenda kerja untuk pekerja dan mengelola laporan kerja. Selain itu, Admin juga dapat menerima notifikasi jika suatu pekerjaan telah diselesaikan oleh pekerja. Admin juga memiliki <i>username</i> dan <i>password</i> untuk dapat menggunakan sistem.

4.3 Fungsi Utama Sistem

Berdasarkan penggalan informasi tentang proses bisnis dan kebutuhan pengguna, fungsi utama sistem manajemen layanan jasa servis UKM dapat dijabarkan dengan ketentuan penomoran sebagai berikut:



Gambar 4.3 Aturan Penomoran Kebutuhan Sistem

Adapun kebutuhan sistem baik fungsional maupun non fungsional akan dijabarkan mengikuti penomoran tersebut dalam table berikut:

Tabel 4.2 Kebutuhan Fungsional Sistem

No.	Fungsional	Kalimat Kebutuhan
TEJA-1-01	Login	Pengguna dapat masuk ke sistem dengan memasukkan <i>username</i> dan <i>password</i> .
TEJA-1-02	Logout	Pengguna yang sedang login dapat keluar dari sistem dengan melakukan <i>logout</i> .
TEJA-1-03	Daftar	Pelanggan dapat melakukan pendaftaran sistem dengan memasukkan email, <i>username</i> , <i>password</i> , nama lengkap, nomor telepon, dan alamat.

TEJA-1-04	Lihat <i>Booking</i>	Pelanggan dan admin dapat melihat data riwayat <i>booking</i> baik yang sudah dikerjakan maupun belum.
TEJA-1-05	<i>Booking</i> Baru	Pelanggan dapat membuat <i>booking</i> baru melalui form <i>booking</i> layanan servis.
TEJA-1-06	Ubah <i>Booking</i>	Pelanggan dapat mengubah data <i>booking</i> yang telah dibuat sebelumnya.
TEJA-1-07	Batal <i>Booking</i>	Pelanggan dapat melakukan pembatalan <i>booking</i> dengan menghapus <i>booking</i> yang telah dibuat sebelumnya.
TEJA-1-08	Beri Penilaian	Pelanggan dapat memberi penilaian terhadap pekerja dengan rentang 1 sampai 5 bintang melalui agenda pekerja yang telah diselesaikan dan mengisi kuesioner penilaian pekerja
TEJA-1-09	Kirim Notifikasi	Sistem harus dapat secara otomatis mengirimkan notifikasi berupa SMS kepada admin yang berisi laporan pekerjaan telah terselesaikan.
TEJA-1-10	Testimoni Baru	Pelanggan dapat memberikan testimoni baru setelah servis diselesaikan oleh pekerja.
TEJA-1-11	Hapus Testimoni	Pelanggan dapat menghapus testimoni yang telah dibuat sebelumnya.
TEJA-1-12	Lihat Agenda	Admin dan pekerja dapat melihat daftar agenda pekerjaan.
TEJA-1-13	Tambah Agenda	Admin dapat menambahkan agenda baru untuk pekerja dengan memasukkan nama pelanggan, alamat, nomor telepon, dan keluhan yang diambil dari data <i>booking</i> pelanggan.
TEJA-1-14	Ubah Agenda	Admin dapat melakukan perubahan data pada agenda kerja yang telah dibuat sebelumnya.
TEJA-1-15	Hapus Agenda	Admin dapat menghapus agenda kerja yang telah dibuat sebelumnya.
TEJA-1-16	Lihat Laporan	Admin dan pekerja dapat melihat laporan pekerjaan yang telah diselesaikan.
TEJA-1-17	Tambah Laporan	Pekerja dapat membuat laporan baru dengan mengisi data laporan pekerjaan berupa tanggal, nama pelanggan, alamat, uraian pekerjaan, dan keterangan yang diambil dari data agenda pekerjaan.
TEJA-1-18	Ubah Laporan	Admin dan pekerja dapat melakukan perubahan data pada laporan pekerjaan

		yang telah dibuat sebelumnya.
TEJA-1-19	Tampilkan Penilaian	Admin dapat melihat data penilaian kinerja yaitu rata-rata penilaian pekerja berdasarkan rating dan nilai hasil kuesioner.

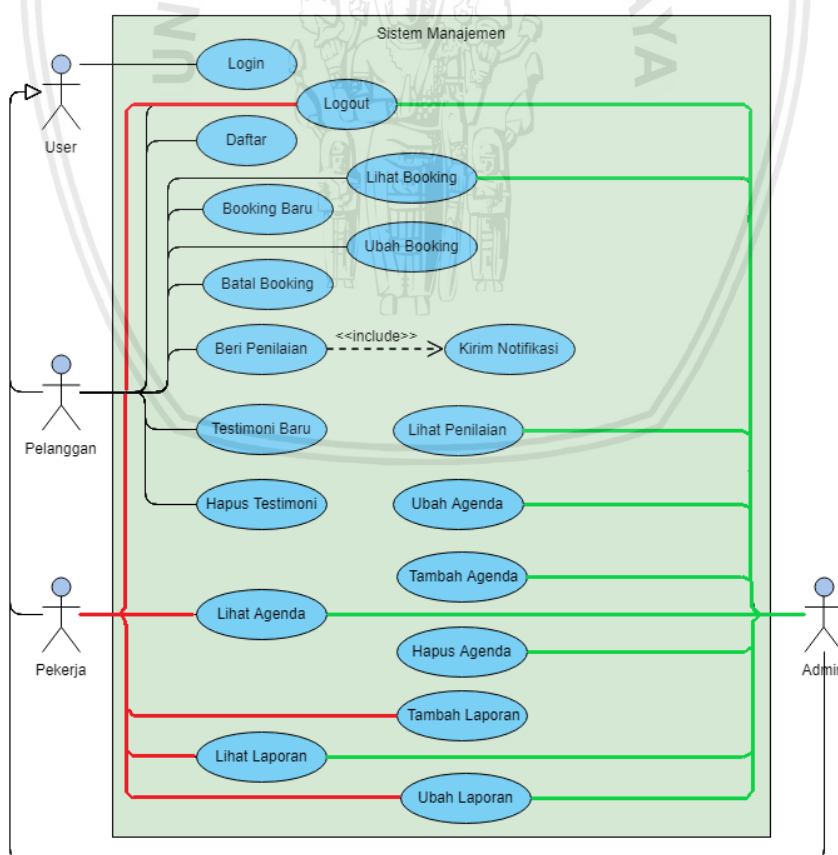
Sementara itu, kebutuhan non fungsional sistem dapat dijabarkan melalui tabel berikut:

Tabel 4.3 Kebutuhan Non Fungsional

No.	Fungsional	Kalimat Kebutuhan
TEJA-2-01	<i>Usability</i>	Daya guna sistem yang dikembangkan harus sesuai dengan pengguna.
TEJA-2-02	<i>Compatibility</i>	Sistem berbasis web harus dapat diakses dengan berbagai <i>browser</i> .

4.4 Usecase Diagram

Usecase diagram yang menggambarkan peran aktor dalam menggunakan sistem pada sistem manajemen layanan UKM, dapat dilihat pada *diagram* berikut:



Gambar 4.4 Usecase Diagram Sistem

Pada *usecase diagram* yang digambarkan, terlihat seluruh user yang merupakan generalisasi dari admin, pekerja, dan pelanggan dapat melakukan fungsi login dan *logout* sistem namun hanya pelanggan yang dapat melakukan daftar untuk membuat akun baru di dalam sistem. Layanan yang dapat digunakan pelanggan dalam sistem adalah melakukan *booking* servis yaitu membuat *booking* baru, melihat data *booking*, mengubah data *booking* yang telah dibuat serta membatalkan *booking* servis. Pelanggan juga dapat memberi penilaian pekerja di lapangan yang mengharuskannya untuk mengisi kuesioner. Selain itu, notifikasi terselesaikannya pekerjaan hanya dapat diterima jika pelanggan telah memberi penilaian. Berkaitan dengan loyalitas pelanggan, pelanggan sebagai pengguna sistem juga dapat memberikan testimoni di sistem.

Sementara itu, admin sebagai pengguna dapat mengatur agenda seperti melihat agenda, menambah agenda baru, menghapus agenda serta mengubah agenda yang ada. Agenda yang diatur oleh admin (dalam hal ini kepala UKM) dapat dilihat oleh pekerja tanpa dapat dimodifikasi oleh pekerja. Fitur laporan pekerjaan dapat digunakan oleh pekerja untuk menambahkan laporan pekerjaan, mengubah laporan pekerjaan, maupun hanya sekedar untuk melihat laporan pekerjaan. Admin sebagai pengguna pun dapat menerima notifikasi bahwa pekerja telah selesai melakukan pekerjaan ditandai dengan pemberian penilaian oleh pelanggan.

4.5 Usecase Scenario

Usecase scenario merupakan gambaran alur suatu *usecase* untuk menjalankan sistem. Dalam *diagram usecase* yang telah dibuat, scenario yang dapat digambarkan, antara lain:

Tabel 4.4 Usecase Scenario Login

Flow of Events - Login	
<i>Objective</i>	Pengguna masuk ke halaman sistem
<i>Actor</i>	User
<i>Pre-condition</i>	Pengguna membuka halaman sistem yang menampilkan form login
<i>Main Flow</i>	1. User memasukkan username 2. User memasukkan password 3. User menekan tombol masuk 4. Sistem menampilkan halaman utama sistem
<i>Alternative Flow</i>	1. Sistem kembali menampilkan form login dengan peringatan username dan/atau password salah 2. Sistem menampilkan peringatan username dan/atau password belum diisi
<i>Post-condition</i>	Pengguna berhasil masuk ke sistem yang menampilkan halaman utama

Tabel 4.5 Usecase Scenario Logout

Flow of Events - Logout	
<i>Objective</i>	Pengguna dapat keluar dari sistem
<i>Actor</i>	Pelanggan, Pekerja, Admin
<i>Pre-condition</i>	Pengguna masuk ke sistem dan sedang melakukan aktivitas di dalam sistem
<i>Main Flow</i>	1. Pelanggan, Pekerja, Admin menekan tombol logout 2. Pelanggan, Pekerja, Admin keluar dari sistem
<i>Alternative Flow</i>	-
<i>Post-condition</i>	Sistem menampilkan halaman website dengan tombol login atau daftar

Tabel 4.6 Usecase Scenario Daftar

Flow of Events - Daftar	
<i>Objective</i>	Pengguna memiliki akun untuk dapat masuk ke sistem
<i>Actor</i>	User
<i>Pre-condition</i>	Pengguna membuka halaman sistem yang menampilkan halaman depan website
<i>Main Flow</i>	1. User memasukkan email, username, password, nama lengkap, nomor telepon, alamat 2. User menekan tombol daftar 3. Sistem menampilkan pesan berhasil membuat akun
<i>Alternative Flow</i>	1. Sistem menampilkan pesan field kosong 2. Sistem menampilkan peringatan username sudah terdaftar 3. Sistem menampilkan peringatan password belum memenuhi kriteria 4. Sistem menampilkan peringatan jika email yang diisikan tidak sesuai format email, alamat dan telepon tidak sesuai format
<i>Post-condition</i>	Pengguna berhasil mendaftar ke sistem dan telah memiliki akun

Tabel 4.7 Usecase Scenario Booking Servis Baru

Flow of Events – Booking Servis Baru	
<i>Objective</i>	Pengguna dapat melakukan booking servis baru
<i>Actor</i>	Pelanggan
<i>Pre-condition</i>	Pengguna masuk ke sistem yang menampilkan halaman utama dengan menu <i>booking</i> servis
<i>Main Flow</i>	1. Pelanggan mengisikan data nama pemesan,

	alamat servis, dan keluhan servis 2. Pelanggan memilih buat <i>booking</i> baru 3. Sistem otomatis memberi status <i>booking</i> menjadi “booked” 4. Sistem menampilkan pesan data <i>booking</i> telah direkam
<i>Alternative Flow</i>	1. Sistem menampilkan pesan field kosong
<i>Post-condition</i>	Data <i>booking</i> pelanggan telah berhasil tersimpan

Tabel 4.8 Usecase Scenario Lihat Booking

Flow of Events – Lihat Booking	
<i>Objective</i>	Pengguna dapat melihat data <i>booking</i>
<i>Actor</i>	Pelanggan, Admin
<i>Pre-condition</i>	Pengguna masuk ke sistem yang menampilkan <i>booking</i>
<i>Main Flow</i>	Pelanggan dan Admin dapat melihat daftar <i>booking</i> yang telah dibuat sebelumnya
<i>Alternative Flow</i>	-
<i>Post-condition</i>	Data <i>booking</i> berhasil ditampilkan oleh sistem yang dilihat oleh pengguna

Tabel 4.9 Usecase Scenario Ubah Booking Servis

Flow of Events – Ubah Booking Servis	
<i>Objective</i>	Pengguna dapat melakukan perubahan pada data <i>booking</i> yang telah dibuat sebelumnya
<i>Actor</i>	Pelanggan
<i>Pre-condition</i>	Sistem menampilkan data <i>booking</i> pelanggan yang telah dibuat sebelumnya
<i>Main Flow</i>	1. Pelanggan memilih pilihan ubah data <i>booking</i> servis 2. Pelanggan melakukan perubahan data nama pemesan, alamat servis, maupun keluhan servis. 3. Pelanggan menyimpan perubahan dengan memilih tombol simpan
<i>Alternative Flow</i>	Sistem melakukan perubahan data agenda jika data <i>booking</i> yang diubah sudah masuk ke agenda kerja
<i>Post-condition</i>	Perubahan data <i>booking</i> pelanggan telah berhasil tersimpan

Tabel 4.10 Usecase Scenario Batal Booking

Flow of Events – Batal Booking	
<i>Objective</i>	Pengguna dapat melakukan pembatalan <i>booking</i> servis yang telah dibuat sebelumnya
<i>Actor</i>	Pelanggan
<i>Pre-condition</i>	Sistem menampilkan data <i>booking</i> pelanggan yang telah dibuat sebelumnya
<i>Main Flow</i>	1. Pelanggan memilih pilihan batalkan data <i>booking</i> servis 2. Sistem menampilkan peringatan konfirmasi pembatalan 3. Pelanggan melakukan konfirmasi pembatalan <i>booking</i> servis
<i>Alternative Flow</i>	Sistem menghapus data agenda jika data booking yang dihapus sudah masuk ke agenda kerja
<i>Post-condition</i>	Pembatalan <i>booking</i> berhasil dilakukan dan data <i>booking</i> terhapus dari sistem

Tabel 4.11 Usecase Scenario Beri Penilaian

Flow of Events – Beri Penilaian	
<i>Objective</i>	Pengguna dapat memberikan penilaian terhadap kinerja pekerja di lapangan
<i>Actor</i>	Pelanggan
<i>Pre-condition</i>	Sistem menampilkan form penilaian kinerja dari halaman agenda pekerja
<i>Main Flow</i>	1. Pelanggan mengisi rentang penilaian dari 1 sampai 5 bintang 2. Pelanggan mengisi kuesioner penilaian pekerja 3. Pelanggan memilih kirim penilaian 4. Sistem otomatis mengubah status <i>booking</i> dari “process” menjadi “done” 5. Sistem menampilkan pesan berhasil menyimpan penilaian
<i>Alternative Flow</i>	-
<i>Post-condition</i>	Penilaian pekerja berhasil direkam oleh sistem

Tabel 4.12 Usecase Scenario Terima Notifikasi

Flow of Events – Terima Notifikasi	
<i>Objective</i>	Pengguna dapat menerima notifikasi berupa SMS
<i>Actor</i>	Admin
<i>Pre-condition</i>	Penilaian dan kuesioner telah diisi oleh pelanggan dan menekan tombol simpan
<i>Main Flow</i>	1. Sistem mengirim SMS laporan bahwa pekerja

	telah menyelesaikan pekerjaan sesuai agenda kepada Admin
<i>Alternative Flow</i>	-
<i>Post-condition</i>	Notifikasi berupa SMS berhasil diterima Admin dan Pelanggan

Tabel 4.13 Usecase Scenario Testimoni Baru

Flow of Events – Beri Testimoni	
<i>Objective</i>	Pengguna dapat menambahkan testimoni baru
<i>Actor</i>	Pelanggan
<i>Pre-condition</i>	Pengguna masuk ke sistem yang menampilkan halaman daftar booking yang belum diulas
<i>Main Flow</i>	1. Pelanggan memilih pilihan beri testimoni 2. Pelanggan mengisi testimoni 3. Pelanggan memilih simpan testimoni 4. Sistem otomatis mengubah status <i>booking</i> dari “done” menjadi “reviewed” 5. Sistem menampilkan pesan testimoni telah berhasil disimpan dan ditampilkan ke halaman
<i>Alternative Flow</i>	-
<i>Post-condition</i>	Testimoni pelanggan telah berhasil tersimpan dan ditampilkan oleh sistem

Tabel 4.14 Usecase Scenario Hapus Testimoni

Flow of Events – Hapus Testimoni	
<i>Objective</i>	Pengguna dapat menghapus testimoni yang telah dibuat sebelumnya
<i>Actor</i>	Pelanggan
<i>Pre-condition</i>	Sistem menampilkan testimoni pelanggan yang telah dibuat sebelumnya
<i>Main Flow</i>	1. Pelanggan memilih pilihan hapus testimoni 2. Sistem otomatis mengubah status <i>booking</i> dari “reviewed” kembali ke “done” 3. Sistem menghapus testimoni
<i>Alternative Flow</i>	-
<i>Post-condition</i>	Testimoni berhasil terhapus dari sistem

Tabel 4.15 Usecase Scenario Lihat Agenda

Flow of Events – Lihat Agenda	
<i>Objective</i>	Pengguna dapat melihat daftar agenda
<i>Actor</i>	Admin, Pekerja
<i>Pre-condition</i>	Pengguna masuk ke sistem yang menampilkan

	agenda pekerjaan
<i>Main Flow</i>	Admin dan Pekerja dapat melihat detail pekerjaan yang tertulis dalam agenda kerja
<i>Alternative Flow</i>	-
<i>Post-condition</i>	Agenda pekerjaan hari ini berhasil ditampilkan oleh sistem yang dilihat oleh pengguna

Tabel 4.16 Usecase Scenario Tambah Agenda

Flow of Events – Tambah Agenda	
<i>Objective</i>	Pengguna dapat menambahkan agenda
<i>Actor</i>	Admin
<i>Pre-condition</i>	Pengguna masuk ke sistem yang menampilkan agenda pekerjaan
<i>Main Flow</i>	1. Admin memilih pilihan masukkan antrian <i>booking</i> ke agenda kerja 2. Admin mengisi data agenda pekerjaan berupa nama pelanggan, alamat, nomor telepon, dan keluhan yang diambil dari data <i>booking</i> pelanggan 3. Admin menyimpan data agenda kerja 4. Sistem otomatis mengubah status booking dari “booked” menjadi “process” 5. Sistem merekam data agenda baru
<i>Alternative Flow</i>	-
<i>Post-condition</i>	Agenda pekerjaan hari ini berhasil ditambahkan ke sistem

Tabel 4.17 Usecase Scenario Ubah Agenda

Flow of Events – Ubah Agenda	
<i>Objective</i>	Pengguna dapat melakukan perubahan terhadap agenda
<i>Actor</i>	Admin
<i>Pre-condition</i>	Pengguna masuk ke sistem yang menampilkan agenda pekerjaan
<i>Main Flow</i>	1. Admin memilih pilihan ubah agenda kerja yang telah dibuat sebelumnya 2. Admin melakukan perubahan data agenda pekerjaan berupa nama pelanggan, alamat, nomor telepon, dan/atau keluhan 3. Admin menyimpan perubahan data agenda kerja 4. Sistem merekam data perubahan agenda
<i>Alternative Flow</i>	-

<i>Post-condition</i>	Agenda pekerjaan yang telah dibuat sebelumnya berhasil diubah
-----------------------	---

Tabel 4.18 Usecase Scenario Hapus Agenda

Flow of Events – Hapus Agenda	
<i>Objective</i>	Pengguna dapat melakukan penghapusan agenda
<i>Actor</i>	Admin
<i>Pre-condition</i>	Pengguna masuk ke sistem yang menampilkan agenda pekerjaan
<i>Main Flow</i>	1. Admin memilih pilihan hapus agenda kerja yang telah dibuat sebelumnya 2. Sistem menampilkan peringatan konfirmasi penghapusan agenda kerja 3. Admin melakukan konfirmasi penghapusan agenda kerja 4. Sistem otomatis mengubah status booking dari “process” kembali menjadi “booked” 5. Sistem menghapus data agenda
<i>Alternative Flow</i>	Pengguna batal menghapus agenda kerja melalui konfirmasi batal
<i>Post-condition</i>	Agenda pekerjaan yang telah dibuat sebelumnya berhasil dihapus

Tabel 4.19 Usecase Scenario Lihat Laporan

Flow of Events – Lihat Laporan	
<i>Objective</i>	Pengguna dapat melihat laporan pekerjaan
<i>Actor</i>	Admin, Pekerja
<i>Pre-condition</i>	Pengguna masuk ke sistem yang menampilkan laporan pekerjaan
<i>Main Flow</i>	Admin dan Pekerja dapat melihat detail pekerjaan yang tertulis dalam laporan pekerjaan
<i>Alternative Flow</i>	-
<i>Post-condition</i>	Laporan pekerjaan berhasil ditampilkan oleh sistem yang dilihat oleh pengguna

Tabel 4.20 Usecase Scenario Tambah Laporan

Flow of Events – Tambah Laporan	
<i>Objective</i>	Pengguna dapat menambahkan laporan pekerjaan baru
<i>Actor</i>	Pekerja
<i>Pre-condition</i>	Pengguna masuk ke sistem yang menampilkan laporan pekerjaan

<i>Main Flow</i>	1. Pekerja memilih pilihan buat laporan baru 2. Pekerja mengisi data laporan pekerjaan berupa tanggal, nama pelanggan, alamat, uraian pekerjaan, dan keterangan yang diambil dari data agenda 3. Pekerja menyimpan data laporan pekerjaan 4. Sistem otomatis mengubah status agenda dari "done" menjadi "reported"
<i>Alternative Flow</i>	-
<i>Post-condition</i>	Laporan pekerjaan berhasil ditambahkan ke sistem

Tabel 4.21 Usecase Scenario Ubah Laporan

Flow of Events – Laporkan Pekerjaan (Ubah Laporan)	
<i>Objective</i>	Pengguna dapat melakukan perubahan terhadap laporan pekerjaan yang telah dibuat sebelumnya
<i>Actor</i>	Admin, Pekerja
<i>Pre-condition</i>	Pengguna masuk ke sistem yang menampilkan laporan pekerjaan
<i>Main Flow</i>	1. Admin dan Pekerja memilih pilihan ubah laporan yang telah dibuat sebelumnya 2. Admin dan Pekerja melakukan perubahan data laporan pekerjaan berupa tanggal, nama pelanggan, alamat, uraian pekerjaan, dan keterangan penggunaan bahan tambahan 3. Admin dan Pekerja menyimpan perubahan laporan pekerjaan
<i>Alternative Flow</i>	Admin dapat melakukan perubahan data laporan dan data biaya
<i>Post-condition</i>	Laporan pekerjaan yang telah dibuat sebelumnya berhasil diubah

Tabel 4.22 Usecase Scenario Lihat Penilaian

Flow of Events – Lihat Penilaian	
<i>Objective</i>	Pengguna dapat melihat penilaian pekerja
<i>Actor</i>	Admin
<i>Pre-condition</i>	Pengguna masuk ke sistem yang menampilkan penilaian
<i>Main Flow</i>	1. Sistem dapat menampilkan penilaian setiap pekerja yang telah dihitung rata-ratanya 2. Admin dapat melihat seluruh data penilaian pekerja
<i>Alternative Flow</i>	-
<i>Post-condition</i>	Penilaian pekerja berhasil ditampilkan sistem

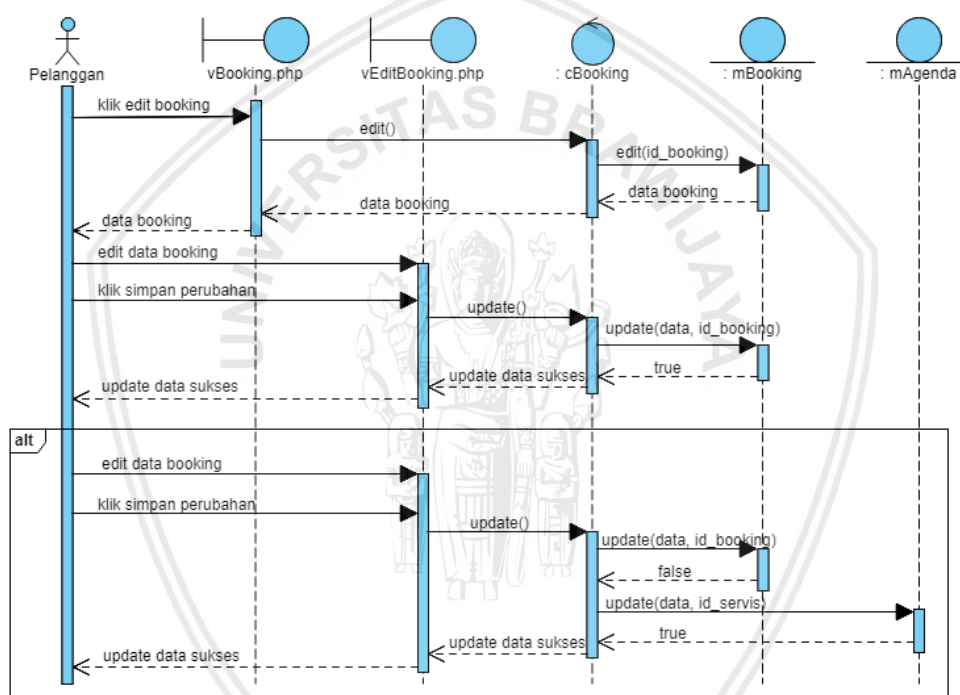
BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bagian perancangan menggambarkan rancangan sistem yang nantinya akan diimplementasikan dalam bentuk kode program. Perancangan tersebut terdiri dari *sequence diagram*, *class diagram*, rancangan algoritme, serta rancangan antarmuka. Hasil rancangan yang telah dibuat akan diimplementasikan menjadi sebuah sistem yang utuh.

5.1 Sequence Diagram

Dalam *sequence diagram* telah digambarkan rangkaian penyampaian pesan yang berurutan antara aktor dengan *boundary*, *controller*, maupun *entity* sebagai berikut:

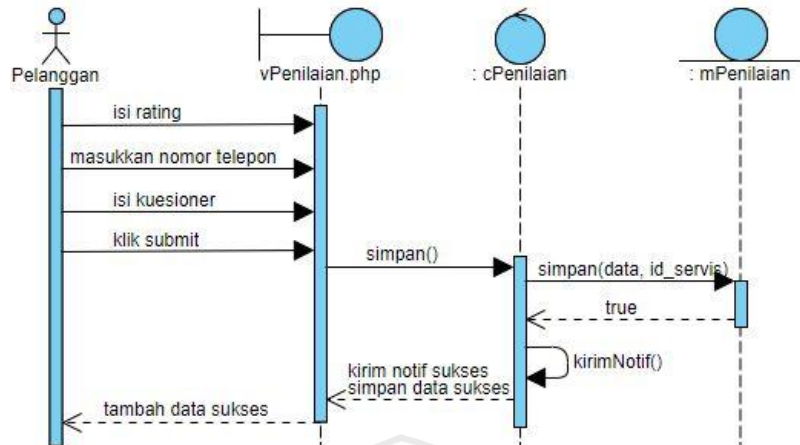
5.1.1 Sequence Diagram Ubah Booking



Gambar 5.1 Sequence Diagram Ubah Booking

Pelanggan dapat mengubah data *booking* yang telah dibuat sebelumnya dengan memilih ubah *booking*. View daftar *booking* sebagai *boundary* akan menampilkan daftar *booking* yang ada. Untuk memperoleh data *booking*, *controller booking* akan memanggil *method* *edit(id_booking)* sesuai parameter id kemudian *class mBooking* akan melakukan koneksi ke *database* dengan *query select*. Ketika detail *booking* yang diperoleh telah ditampilkan pada *boundary* edit *booking*, pelanggan dapat mengubah isian data dan menyimpan perubahan yang telah dilakukan. Perubahan tersebut akan diteruskan kepada *controller* dengan memanggil *method* *update(data, id_booking)* agar dapat dilakukan *query update* oleh *class mBooking* sesuai data yang diubah dan *id_booking* yang diubah.

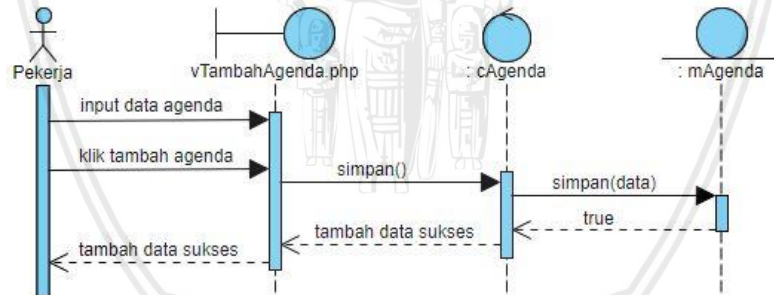
5.1.2 Sequence Diagram Beri Penilaian



Gambar 5.2 Sequence Diagram Isi Kuesioner dan Terima Notifikasi

Sequence *diagram* dari fungsi beri penilaian dan kirim notifikasi digambarkan menjadi satu karena notifikasi hanya dapat dikirim oleh sistem begitu penilaian diisi. Ketika data isian kuesioner dikirim berupa data bobot penilaian masing-masing pertanyaan, *controller* meneruskan ke *class* mPenilaian dengan memanggil *method* *simpan()*. Model akan meneruskan data dengan melakukan *query* insert kepada *database* sehingga data penilaian berhasil disimpan. Ketika data penilaian berhasil disimpan, *controller* kembali memanggil *method* *kirimNotif()* untuk mengirim notifikasi kepada admin.

5.1.3 Sequence Diagram Tambah Agenda

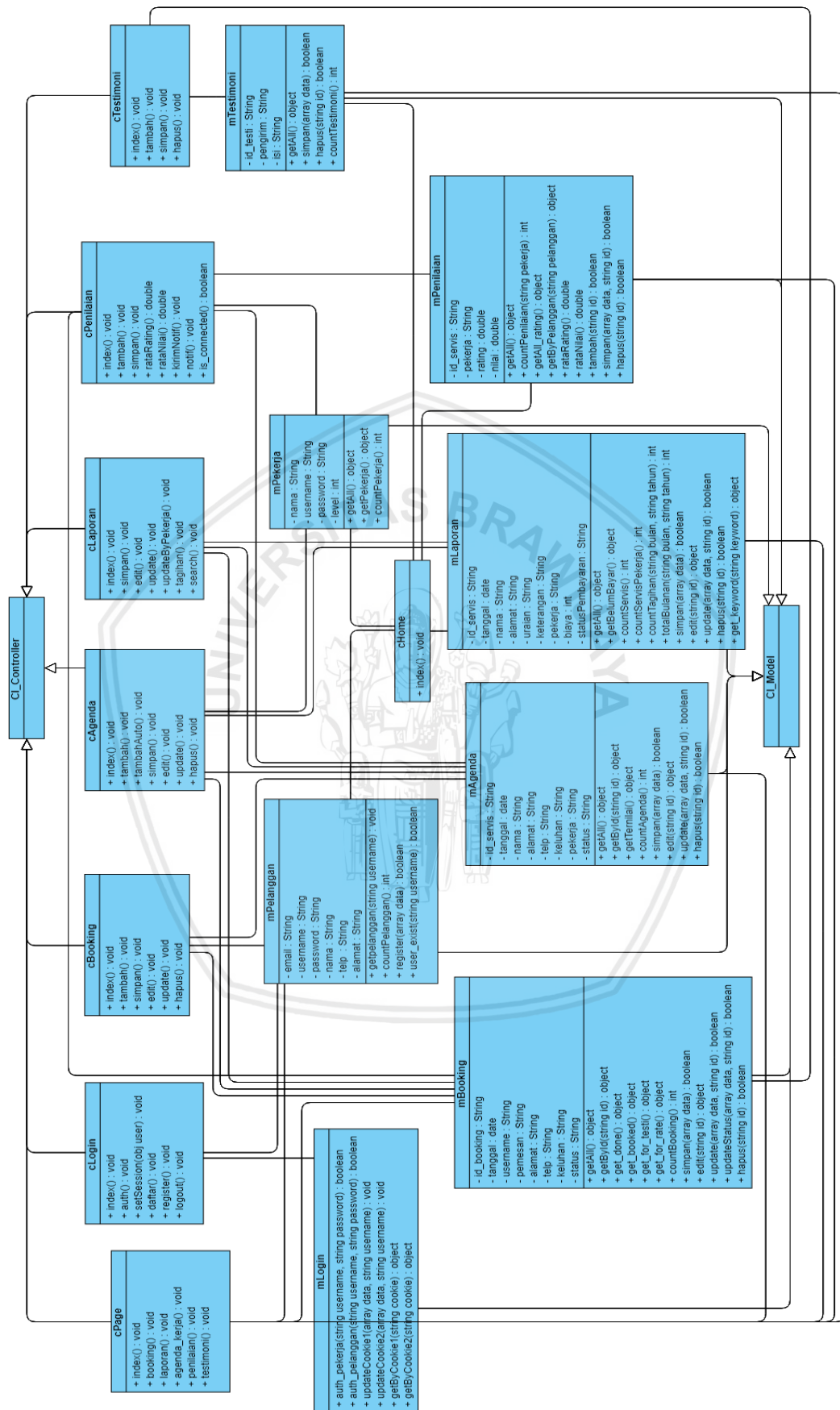


Gambar 5.3 Sequence Diagram Tambah Agenda

Agenda baru dapat ditambahkan dengan menginputkan data agenda dan memilih menu tambah agenda pada halaman agenda. Data pilihan pada *boundary view* akan diteruskan ke *controller* agenda dengan memanggil *method* *simpan()*. *Controller* agenda berhubungan langsung dengan mAgenda untuk dapat mengakses *database* dengan perintah *insert* untuk menyimpan data agenda.

5.2 Class Diagram

Class diagram pada perancangan sistem manajemen layanan jasa servis UKM terdiri dari delapan kelas utama dan lima *class* pengelola yang mengandung atribut dan operasi. Berikut gambaran *class diagram* yang dirancang:



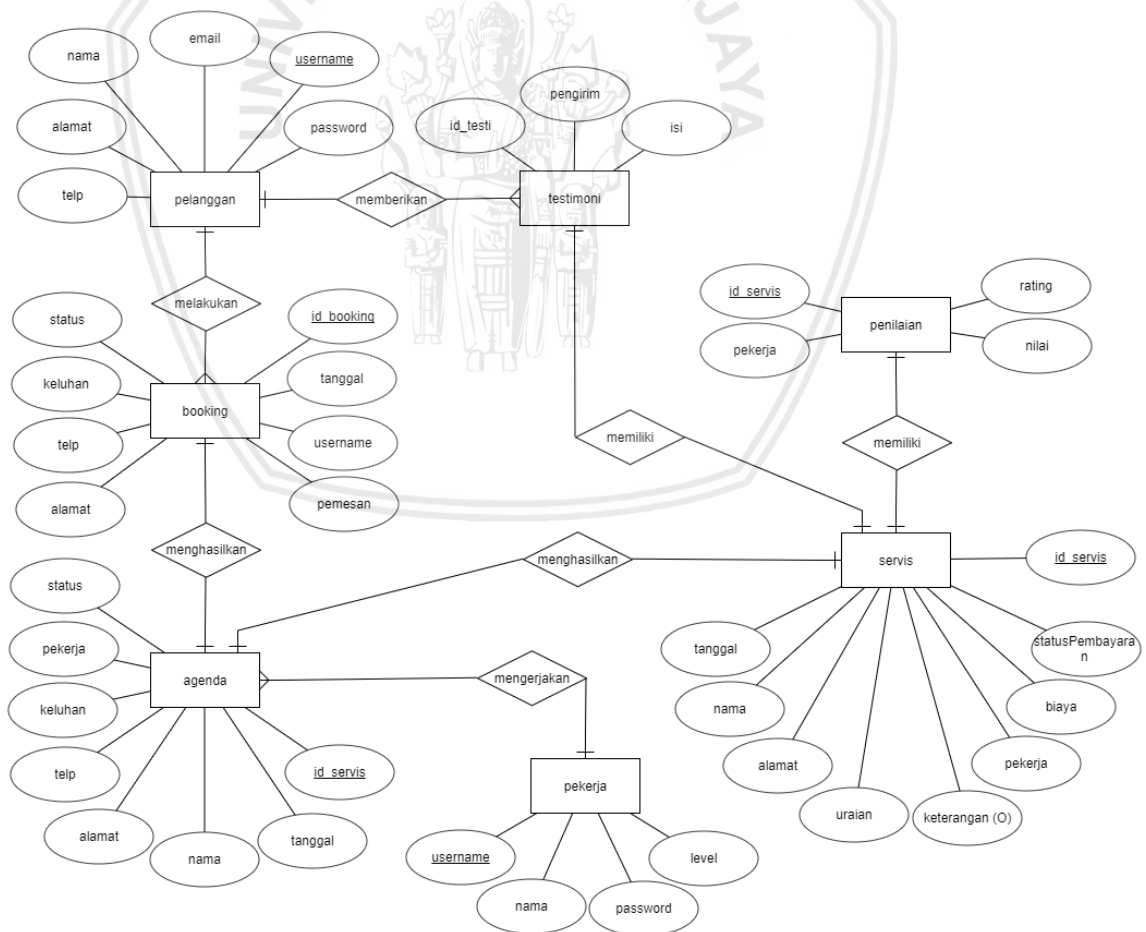
Gambar 5.4 Class Diagram

Class Diagram tersebut terdiri dari *class controller* dan model yang mengatur pertukaran data ke *database*. Terdapat *class model* dan *controller* yang memiliki hubungan asosiasi karena *class controller* memanggil *method* dalam *class model* untuk mengakses *database*. Sementara hasil kembalian dari *method* dalam *class model* akan diolah kembali oleh *controller* untuk nantinya dapat ditampilkan melalui *class view*. Tentunya seluruh *class controller* termasuk ke dalam *CI_Controller* dan seluruh *class model* termasuk ke dalam *CI_Model*.

Adapun hubungan antar *class* yang terjadi pada *class diagram* tersebut adalah *class cBooking* yang berhubungan dengan *class mBooking*, *mPelanggan* dan *mAgenda*, *cAgenda* yang berhubungan dengan *class mAgenda*, *mBooking*, *mLaporan* dan *mPekerja*, *cLaporan* yang berhubungan dengan *class mBooking*, *mAgenda*, *mLaporan* dan *mPenilaian*, *cPenilaian* yang berhubungan dengan *mPenilaian*, *mPekerja* dan *mLaporan*, serta *class cTestimoni* yang berhubungan dengan *class mTestimoni* dan *mBooking*.

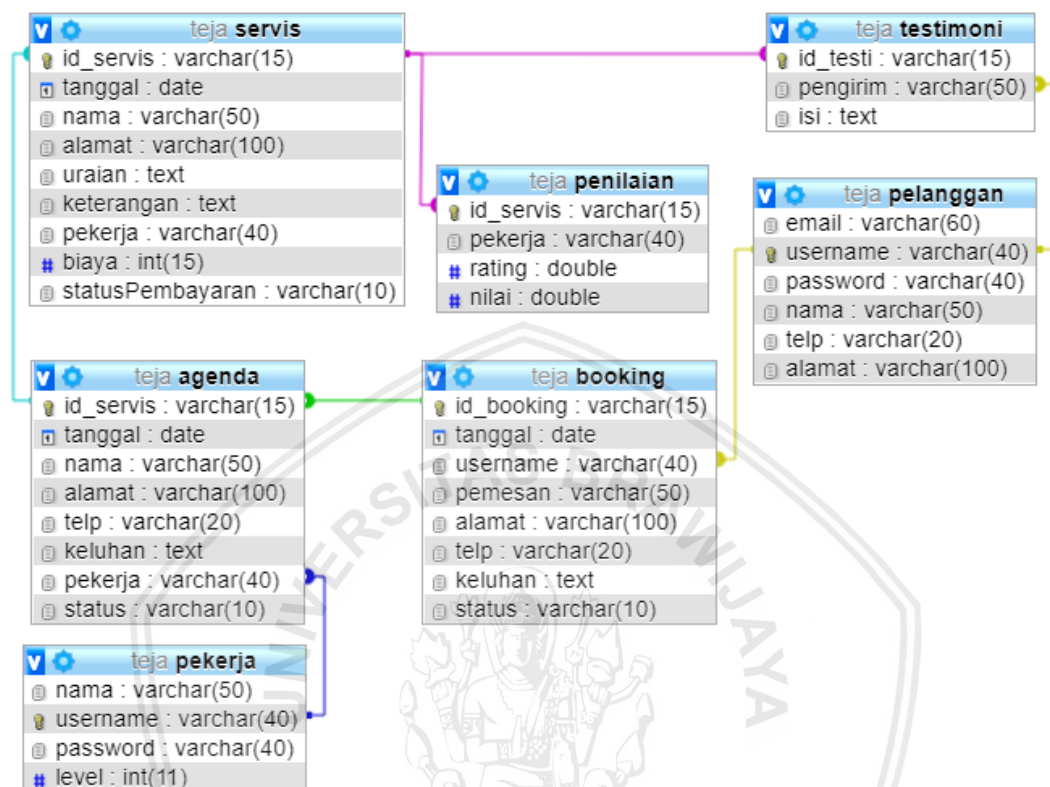
5.3 Rancangan Basis Data

Perancangan basis data dalam web UKM Bali Tekindo Jaya dapat digambarkan sesuai dengan *Entity Relationship Diagram* (ERD) berikut:



Gambar 5.5 Entity Relationship Diagram

Berikut merupakan skema relasi basis data yang digunakan dalam pembuatan web UKM Bali Tekindo Java:



Gambar 5.6 Skema Relasi Basis Data

Dalam *database* yang diberi nama ‘teja’, memiliki delapan buah tabel yang saling berelasi seperti pada gambar 5.6. Adapun tabel pekerja memiliki atribut atau kolom ‘*username*’ sebagai primary key sama halnya dengan tabel pelanggan juga memiliki atribut ‘*username*’ sebagai primary key. Di sisi lain, tabel ‘*booking*’ dalam atribut ‘*username*’ merujuk kepada tabel ‘pelanggan’ dengan atribut sama yang mana hal ini disebut sebagai foreign key.

Perujukan berupa foreign key yang menciptakan relasi antar tabel pun terjadi pada tabel 'servis' ke 'agenda', 'agenda' ke 'booking', 'agenda' ke 'pekerja', 'penilaian' ke 'pekerja' serta 'testimoni' ke 'pelanggan'. Relasi-relasi tersebut dapat terjadi karena atribut dari tabel perujuk menggunakan atribut dari tabel lain yang dirujuk sehingga data yang dihasilkan dari masing-masing tabel sesuai dengan data di tabel lain.

5.5 Rancangan Algoritme

Perancangan algoritme dalam web UKM Bali Tekindo Jaya dapat dilihat dalam cuplikan *pseudocode* berikut:

5.5.1 Algoritme Membuat *Booking* Baru

Algoritme dalam fungsi simpan ini akan diletakkan dalam *class controller booking* yang digunakan untuk menyimpan data *booking* baru oleh pelanggan ke *database* program.

Tabel 5.1 Algoritme Membuat Data *Booking* Baru

Algoritme 1: Fungsi Membuat Data Booking Baru	
1	FUNCTION simpan
2	BEGIN
3	SET id = date dengan format ymdHis;
4	SET data = array(
5	'id_booking' = id,
6	'tanggal' = ambil masukan tanggal di form input,
7	'username' = ambil session user,
8	'pemesan' = ambil masukan nama di form input,
9	'alamat' = ambil masukan alamat di form input
10	'telp' = ambil masukan telepon di form input
11	'keluhan' = ambil masukan keluhan di form input
12	);
13	Panggil fungsi simpan dalam class model
14	Pesan "Data berhasil disimpan";
15	Tampilkan data booking ke halaman view;
16	END

Ketika pelanggan mengisi data *booking* baru pada form *booking*, data isian pelanggan akan dimasukkan ke array. Array tersebut kemudian dimasukkan ke *database* dengan perintah sesuai *query* MySQL. Hasil *query* lalu dikembalikan baik jika berhasil maupun gagal. Data *booking* yang berhasil disimpan akan ditampilkan ke halaman *view*.

5.5.2 Algoritme Menghapus Data *Booking*

Algoritme dalam fungsi hapus ini akan diletakkan dalam *class controller booking* yang digunakan untuk menghapus data *booking* oleh pelanggan.

Tabel 5.2 Algoritme Menghapus Data *Booking*

Algoritme 2: Fungsi Menghapus Data Booking	
1	FUNCTION hapus
2	BEGIN
3	SET id_booking = ambil dari view booking;
4	SET array of data_booking = DB SELECT dari table booking where
5	id_booking = id_booking;
6	SET status = data_booking->status;
7	IF status == "booked"
8	THEN DB DELETE dari table booking where id_booking=id_booking;
9	ELSEIF status == "process"
10	THEN DB DELETE dari table agenda where id_servis=id_booking;
11	DB DELETE dari table booking where id_booking = id_booking;
12	ELSE
13	THEN Pesan "Anda tidak dapat menghapus data booking ini"
14	ENDIF
15	END

Data *booking* pelanggan dapat dihapus berdasarkan *id_booking*. Jika pelanggan ingin menghapus data *booking* maka harus diperhatikan terlebih dahulu status *booking* tersebut. Jika status *booking* adalah "booked" berarti data *booking* tersebut belum masuk ke agenda pekerjaan sehingga dapat langsung dihapus. Namun jika status *booking* adalah "process" maka data agenda juga

harus dihapus. Lain halnya jika status *booking* bukan keduanya, maka tidak bisa dihapus.

5.5.3 Algoritme Menghitung Rata-rata Rating

Algoritme dalam fungsi *rataRating* ini akan diletakkan dalam *class controller* penilaian yang digunakan untuk menghitung nilai rata-rata rating yang diperoleh pekerja.

Tabel 5.3 Algoritme Menghitung Rata-rata Rating

Algoritme 3: Fungsi Menghitung Rata-rata Rating	
1	FUNCTION rataRating
2	BEGIN
3	SET avg_rating = array();
4	SET array of data_pekerja = DB SELECT data seluruh pekerja;
5	SET itr = 0;
6	FOREACH(data_pekerja)
7	SET pekerja = data_pekerja->username;
8	SET rating = rata-rata rating where username pekerja = pekerja;
9	avg_rating[itr] = rating;
10	itr++;
11	END
12	return avg_rating;
13	END

Rata-rata rating dihitung dengan cara mengambil data rating setiap pekerja sesuai *username*. Lalu setiap data ditambahkan dan dihitung rata-ratanya sehingga diperoleh data rata-rata sesuai pekerja. Perhitungan ini dilakukan melalui perulangan.

5.5.4 Algoritme Menghitung Rata-rata Nilai Kuesioner

Algoritme dalam fungsi *rataNilai* ini akan diletakkan dalam *class controller* penilaian yang digunakan untuk menghitung nilai rata-rata nilai hasil pengisian kuesioner yang diperoleh pekerja.

Tabel 5.4 Algoritme Menghitung Rata-rata Nilai Kuesioner

Algoritme 4: Fungsi Menghitung Rata-rata Nilai Kuesioner	
1	FUNCTION rataNilai
2	BEGIN
3	SET avg_nilai = array();
4	SET max = 25;
5	SET array of data pekerja = DB SELECT dari table pekerja where
6	level = 2;
7	SET itr = 0;
8	FOREACH(data pekerja) THEN
9	SET pekerja = data pekerja -> username;
10	SET nilai = rata-rata nilai where username pekerja = pekerja;
11	SET countPekerja = banyaknya data penilaian pekerja where
12	username pekerja = pekerja;
13	IF(countPekerja = 0)
14	THEN avg_nilai[itr] = 0;
15	ELSE
16	THEN avg_nilai[itr] = (nilai/max)*100;
17	ENDIF
18	itr++;
19	ENDFOR
20	return avg_nilai;
21	END

Rata-rata nilai akan dihitung saat seorang pekerja mendapatkan penilaian dari banyak pelanggan melalui agenda yang telah diselesaikan oleh pekerja tersebut. Seluruh pertanyaan kuesioner mencakup 5 pertanyaan dengan rentang nilai 1 untuk sangat kurang sampai 5 untuk sangat baik sehingga nilai maksimal pengisian kuesioner penilaian adalah 25. Kemudian, melalui perulangan, masing-masing pekerja dihitung banyak penilaian yang diperoleh dan disimpan dalam variabel `countPekerja`. Sementara itu, seluruh nilai rata-rata penilaian seorang pekerja disimpan dalam variabel `nilai`. Jika nilai `countPekerja` adalah 0 atau pekerja tersebut belum pernah diberi nilai yang mana berarti pekerja tersebut belum menyelesaikan satu agenda sama sekali, maka nilai `avg_nilai` atau rata-rata nilai diisi dengan 0. Lain hal saat pekerja memiliki penilaian karena telah menyelesaikan pekerjaan maka `avg_nilai`-nya adalah rata-rata nilai yang diperoleh dibagi dengan nilai maksimal kemudian dikalikan 100 untuk data 100%. Demikian perulangan dilakukan dengan menaikkan nilai variabel `itr` agar data dalam array terus bertambah sesuai banyaknya pekerja.

5.5.5 Algoritme Menampilkan Data Penilaian

Algoritme dalam fungsi `index` ini akan diletakkan dalam *class controller* penilaian yang digunakan untuk menampilkan data penilaian baik kepada admin maupun form penilaian untuk akses pekerja.

Tabel 5.5 Algoritme Menampilkan Data Penilaian

Algoritme 5: Fungsi Menampilkan Data Penilaian	
1	FUNCTION <code>index</code>
2	BEGIN
3	IF <code>session user masuk != true</code>
4	THEN <code>redirect to halaman login;</code>
5	ELSE
6	IF <code>session user akses = 1 //sebagai admin</code>
7	THEN SET <code>data = array(</code>
8	<code>'data_penilaian' = GET semua data penilaian,</code>
9	<code>'data_pekerja' = GET semua data pekerja,</code>
10	<code>'data_rating' = hitung rata-rata rating,</code>
11	<code>'data_nilai' = hitung rata-rata nilai,</code>
12	);
13	tampilkan array data dalam view admin
14	ELSEIF <code>session user akses = 2 //sebagai pekerja</code>
15	IF <code>servis belum diberi nilai</code>
16	THEN GET <code>id_servis</code>
17	SET array of <code>data_penilaian = DB SELECT data penilaian</code>
18	<code>where id_servis = id_servis;</code>
19	tampilkan form penilaian servis dalam view pekerja
20	ELSE
21	<code>redirect to controller agenda;</code>
22	pesan "Anda tidak bisa mengakses form penilaian"
23	ENDIF
24	ENDIF
25	ENDIF
26	END

Saat pengguna yang mengakses halaman penilaian belum *login*, pengguna akan diarahkan untuk melakukan *login* terlebih dahulu. Namun jika pengguna sudah *login*, akan dicek *session* akses pengguna dalam *if*. Jika pengguna memiliki hak akses 1 yaitu admin, halaman yang tampil adalah halaman yang berisi data penilaian seluruh pekerja. Lain halnya jika pengguna memiliki hak akses 2 yaitu

pekerja, dilakukan pengecekan lagi terhadap maksud pekerja tersebut mengakses halaman penilaian. Jika untuk mengakses form isian penilaian setelah menyelesaikan pekerjaan, akan diambil `id_servis` dari pekerjaan yang belum dinilai tersebut sehingga dapat menampilkan form isi penilaian dan kuesioner. Namun jika tidak ada pekerjaan yang belum dinilai, maka pekerja tidak dapat mengakses *controller* penilaian dan diarahkan ke *controller* agenda.

5.6 Rancangan Antarmuka

Perancangan antarmuka dalam web UKM Bali Tekindo Jaya secara garis besar menampilkan tiga jenis menu yang berbeda-beda sesuai user. Rancangan antarmuka yang dibuat tentunya mempertimbangkan beberapa prinsip daya guna menurut Jacob Nielsen yaitu (1) mampu menyediakan informasi berupa status, (2) memberikan informasi yang mudah dipahami seperti bahasa sehari-hari, (3) memberi kebebasan dan kemudahan pengguna dalam menggunakan antarmuka, (4) konsisten, (5) pencegahan *error*, (6) mengurangi ingatan pengguna, (7) fleksibel dan efisien, (8) desain dengan estetika dan minimalis, (9) membantu mengenal dan menangani *error*, serta (10) menyediakan dokumentasi dan bantuan. Beberapa dari prinsip tersebut telah diterapkan dalam web Teja seperti tampilan yang sederhana dan bersifat natural, menyediakan informasi singkat mungkin untuk mengurangi beban ingatan pengguna dengan penggunaan navigasi menu, tata letak perintah yang konsisten, pesan kesalahan yang baik, dan mencegah kesalahan terjadi yang dapat dilihat pada penggunaan form validasi.

5.6.1 Rancangan Antarmuka Form Pendaftaran



Form Pendaftaran

Nama

Username

Email

Password

Retype Password

Telepon

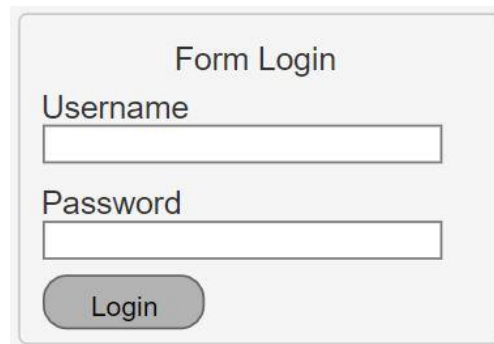
Alamat

Daftar

Gambar 5.7 Rancangan antarmuka Form Pendaftaran

Rancangan antarmuka form pendaftaran berisikan data pendaftaran untuk seorang pelanggan baru. Masing-masing isian seperti nama, *username*, email, *password*, telepon dan alamat harus diisi guna mempercepat proses *booking* servis. Terdapat pula tombol Daftar di sebelah kiri bawah untuk menyimpan data pendaftaran pelanggan baru. Rancangan ini didesain memenuhi prinsip daya guna yaitu dialog yang sederhana hanya menampilkan perintah yang diperlukan saja serta bahasa yang digunakan tergolong natural.

5.6.2 Rancangan Antarmuka Form *Login*

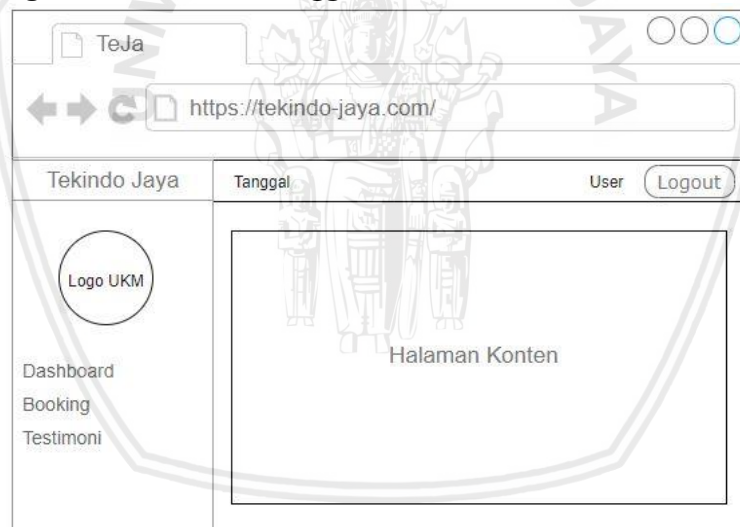


The image shows a simple login form titled "Form Login". It contains two input fields: "Username" and "Password". Below the password field is a "Login" button.

Gambar 5.8 Rancangan Antarmuka Form *Login*

Form *login* dapat dikatakan sebagai perantara masuknya pengguna ke sistem. Pengguna dapat masuk ke sistem hanya dengan mengisi field *username* dan *password* yang telah diinputkan ke pendaftaran sebelumnya. Adapun tombol *Login* berfungsi melakukan pengecekan pengguna agar dapat masuk ke sistem atau menampilkan pesan error jika *username* atau *password* pengguna tidak ditemukan.

5.6.3 Rancangan Antarmuka Pelanggan



Gambar 5.9 Rancangan Antarmuka Pelanggan

Ketika user melakukan *login* sebagai pelanggan, maka menu yang muncul atau menu yang dapat diakses hanya halaman *Dashboard*, *Booking*, dan *Testimoni*. Halaman-halaman tersebut sesuai dengan *usecase* pelanggan yang hanya dapat mengakses fitur *booking* dan *testimoni*. Terdapat penggunaan bar navigasi di sebelah kiri untuk memudahkan pengguna dalam memilih menu layanan yang tersedia. Hal ini tentunya dipertimbangkan guna memenuhi prinsip daya guna yaitu mengurangi beban ingatan pengguna alih-alih meletakkan seluruh layanan dalam satu halaman.

5.6.4 Rancangan Antarmuka Admin



Gambar 5.10 Rancangan Antarmuka Admin

Ketika user melakukan *login* sebagai pekerja level 1 artinya user tersebut memiliki hak akses admin. Menu yang muncul atau menu yang dapat diakses oleh admin adalah halaman Dashboard, Agenda, Penilaian, Laporan dan Testimoni. Halaman-halaman tersebut sesuai dengan *usecase* admin yang memiliki hak akses tertinggi dalam sistem. Sama seperti desain antarmuka pelanggan pada sub-bab sebelumnya, halaman admin juga memuat bar navigasi yang membantu mengurangi beban ingatan pengguna. Selain itu, bar navigasi di setiap halaman juga menunjukkan konsistensi desain antarmuka.

5.6.5 Rancangan Antarmuka Pekerja



Gambar 5.11 Rancangan Antarmuka Pekerja

Ketika user melakukan *login* sebagai pekerja level 2 artinya user tersebut memiliki hak akses pekerja. Menu yang muncul atau menu yang dapat diakses oleh pekerja hanya halaman Dashboard, Agenda, dan Laporan. Halaman-halaman tersebut sesuai dengan *usecase* pekerja yang hanya dapat mengakses

beberapa fitur saja. Tombol *Logout* yang selalu terletak di kanan atas pada setiap halaman juga menunjukkan konsistensi desain sehingga pengguna mudah beradaptasi dengan tata letak perintah web TeJa seperti aplikasi-aplikasi lain pada umumnya.

5.6.6 Rancangan Antarmuka Lihat *Booking* – Pelanggan

Tanggal	Nama	Alamat	Keluhan	Status	Opsi

Gambar 5.12 Rancangan Antarmuka Lihat *Booking* – Pelanggan

Pengguna yang masuk sebagai pelanggan memperoleh layanan *Booking* dan *Testimoni*. Dalam menu *Booking*, pelanggan dapat melihat data *booking* yang pernah dilakukan ditampilkan berupa tabel data dengan opsi edit maupun hapus di masing-masing barisnya. Adapun opsi edit berguna untuk mengarahkan pelanggan ke form edit *booking* dengan isian yang diambil dari data *booking* sebelumnya dan opsi hapus berguna untuk menghapus *booking* yang pernah dibuat. Sementara itu, tombol tambah *booking* berguna untuk mengarahkan pelanggan ke form tambah *booking* baru.

5.6.7 Rancangan Antarmuka Tambah *Booking* – Pelanggan

Gambar 5.13 Rancangan Antarmuka Tambah *Booking* – Pelanggan

Jika pelanggan memilih tombol Tambah *Booking* di menu tampilan *booking* maka akan masuk ke form tambah *booking*. Terdapat isian yang harus diisi seperti tanggal, nama, alamat, telepon dan keluhan yang otomatis diambil dari data pelanggan namun tetap dapat diubah untuk keperluan *booking*. Sementara tombol simpan berguna untuk menyimpan data *booking* ke *database* serta mengarahkan kembali pelanggan ke tampilan data *booking* dengan pesan penyimpanan data berhasil. Sama seperti desain halaman daftar dan halaman *login*, halaman setiap form dalam web TeJa juga memiliki tombol yang terdapat di bagian kiri bawah. Hal ini menunjukkan konsistensi desain agar memudahkan pengguna beradaptasi dengan sistem.

5.6.8 Rancangan Antarmuka Ubah *Booking* – Pelanggan

Gambar 5.14 Rancangan Antarmuka Ubah *Booking* – Pelanggan

Form edit *booking* muncul jika opsi edit di masing-masing baris data *booking* dipilih. Form edit *booking* menampilkan isian data *booking* sesuai data *booking* yang dipilih untuk diedit. Seluruh data *booking* yang pernah diisi seperti tanggal, nama, alamat, telepon dan keluhan dapat diubah kecuali ID *booking*. Tombol simpan berfungsi menyimpan data perubahan *booking* yang dilakukan.

5.6.9 Rancangan Antarmuka Lihat Testimoni – Pelanggan

Gambar 5.15 Rancangan Antarmuka Lihat Testimoni – Pelanggan

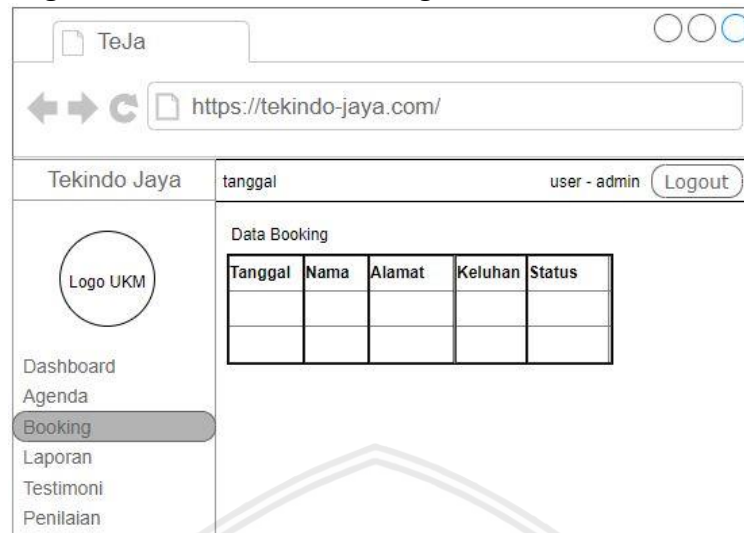
Menu testimoni pelanggan berisikan data riwayat *booking* dan data testimoni. Testimoni hanya dapat diberikan jika status *booking* sudah selesai. Data *booking* yang sudah selesai namun belum diberi testimoni akan ditampilkan pada tabel riwayat *booking* maka terdapat opsi Beri Testimoni. Pada tabel data testimoni terdapat opsi hapus untuk menghapus testimoni yang pernah dibuat.

5.6.10 Rancangan Antarmuka Tambah Testimoni – Pelanggan

Gambar 5.16 Rancangan Antarmuka Tambah Testimoni – Pelanggan

Halaman pengisian testimoni hanya berisi isian testimoni dengan ID yang diambil dari data *booking* dan tidak dapat diubah. Isian testimoni berupa textarea memudahkan pelanggan mengisi banyak kata maupun kalimat terkait testimoni yang dirasakan pasca layanan servis yang diperoleh. Adapun tombol Tambah berguna untuk menambahkan data isian testimoni ke *database* sistem.

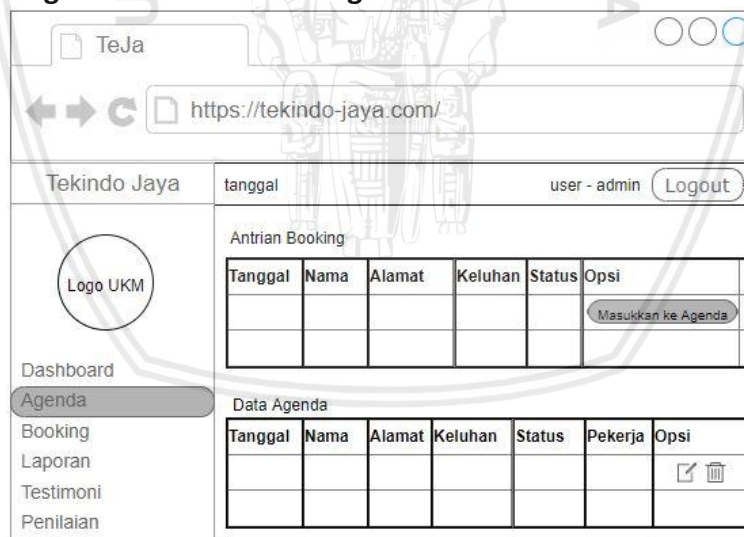
5.6.11 Rancangan Antarmuka Lihat *Booking* – Admin



Gambar 5.17 Rancangan Antarmuka Lihat *Booking* – Admin

Sebagai admin, pengguna pun dapat melihat data seluruh *booking* melalui menu *booking*. Data seluruh *booking* ditampilkan dalam bentuk tabel namun tidak ada opsi apapun karena admin tidak dapat melakukan pengelolaan data *booking*.

5.6.12 Rancangan Antarmuka Lihat *Agenda* – Admin



Gambar 5.18 Rancangan Antarmuka Lihat *Agenda* – Admin

Admin dapat melihat agenda yang belum terselesaikan berikut opsi untuk mengedit dan menghapus data agenda. Data agenda yang ditampilkan tentunya berasal dari *booking* sehingga jika ingin menambahkan agenda baru, terdapat tabel antrian *booking* yang belum dikerjakan. Antrian *booking* yang ditampilkan memiliki opsi untuk admin dapat menambahkannya ke agenda melalui form tambah data agenda yang diarahkan saat opsi dipilih.

5.6.13 Rancangan Antarmuka Tambah Agenda – Admin

Gambar 5.19 Rancangan Antarmuka Tambah Agenda – Admin

Data *booking* yang ingin ditambahkan ke agenda akan ditampilkan secara detail di form tambah agenda. Terdapat isian tanggal, nama, alamat, telepon, dan keluhan yang sudah otomatis diisi berdasarkan data *booking* namun tetap dapat diubah sesuai kebutuhan. Sementara isian pekerja berupa combobox adalah pilihan nama pekerja yang ditugaskan oleh admin mengerjakan agenda tersebut. Adapun tombol simpan berfungsi untuk menyimpan data agenda baru dan mengarahkan kembali ke menu agenda.

5.6.14 Rancangan Antarmuka Ubah Agenda - Admin

Gambar 5.20 Rancangan Antarmuka Ubah Agenda – Admin

Ketika admin memilih opsi edit pada salah satu baris agenda dalam tabel, admin akan diarahkan ke form edit agenda. Terdapat isian tanggal, nama, alamat, telepon, keluhan dan pekerja yang dapat diubah namun tidak pada ID servis yang diperoleh dari ID *booking*. Sementara tombol simpan akan menyimpan perubahan data agenda dan mengembalikan admin ke menu tampilan agenda.

5.6.15 Rancangan Antarmuka Lihat Laporan – Admin

Tanggal	Nama	Alamat	Keluhan	Pekerja	Biaya	Status	Opsi

Gambar 5.21 Rancangan Antarmuka Lihat Laporan – Admin

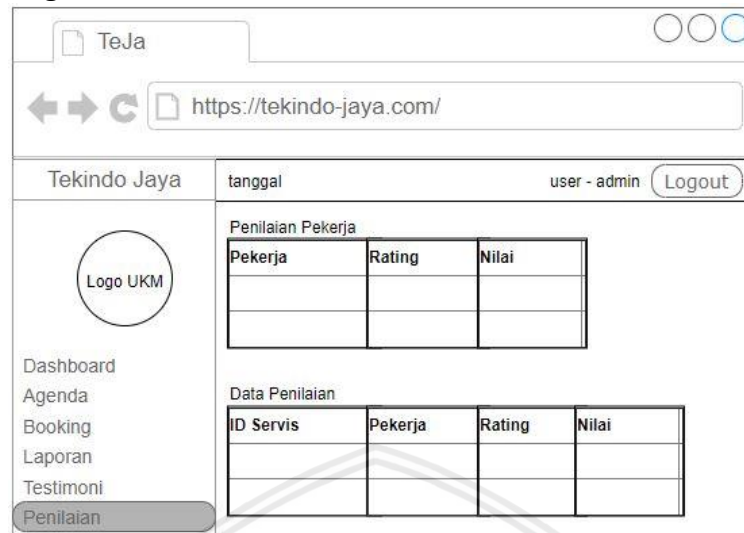
Dalam menu laporan, admin dapat melihat tampilan data laporan berupa tabel yang hanya memiliki opsi edit. Ketika opsi edit dipilih, admin akan diarahkan ke form edit laporan.

5.6.16 Rancangan Antarmuka Ubah Laporan – Admin

Gambar 5.22 Rancangan Antarmuka Ubah Laporan – Admin

Form edit laporan muncul ketika opsi edit laporan dipilih serta secara otomatis menampilkan detail data laporan ke masing-masing field input. Terdapat field yang dapat diubah seperti tanggal, nama, alamat, uraian, keterangan, pekerja, biaya, dan status pembayaran namun tidak untuk ID servis. Adapun tombol simpan berfungsi untuk menyimpan data laporan yang telah diubah dan mengembalikan admin ke menu tampilan laporan.

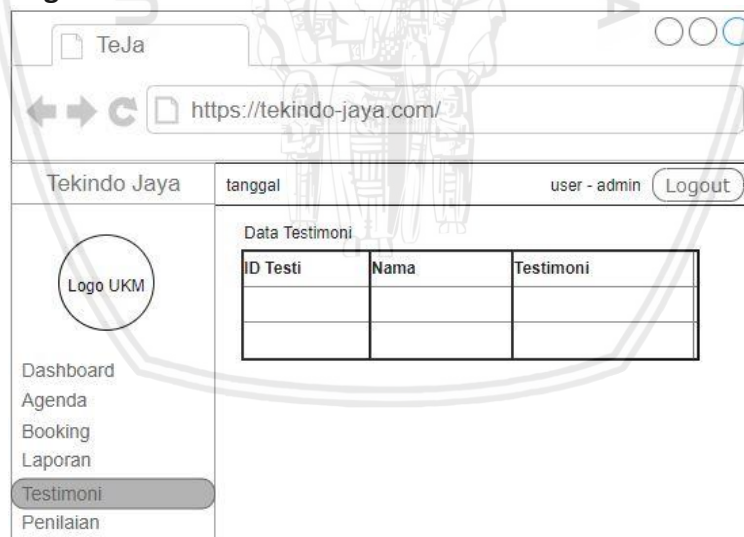
5.6.17 Rancangan Antarmuka Lihat Penilaian – Admin



Gambar 5.23 Rancangan Antarmuka Lihat Penilaian – Admin

Admin dapat melihat penilaian pekerja yang diberikan oleh pelanggan. Tampilan penilaian pekerja terdiri dari 2 tabel yaitu tabel yang memuat rata-rata perolehan nilai pekerja dan tabel yang memuat seluruh data penilaian pekerja. Tidak terdapat opsi apapun pada menu penilaian dari sisi admin.

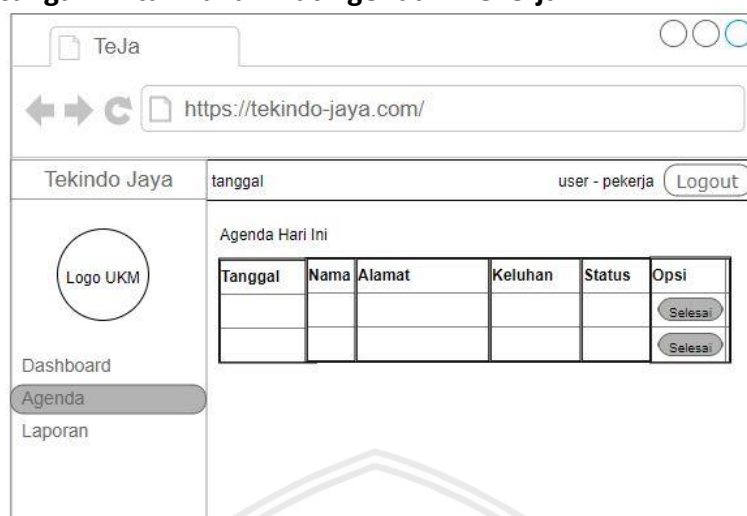
5.6.18 Rancangan Antarmuka Lihat Testimoni – Admin



Gambar 5.24 Rancangan Antarmuka Lihat Testimoni – Admin

Menu testimoni dari sisi admin hanya menampilkan data testimony berbentuk tabel yang berisi data ID testi, nama pelanggan yang memberikan testimoni, dan data testimoni itu sendiri. Tidak terdapat opsi apapun dalam tabel testimoni karena admin tidak berhak melakukan pengelolaan testimoni.

5.6.19 Rancangan Antarmuka Lihat Agenda – Pekerja



Gambar 5.25 Rancangan Antarmuka Lihat Agenda – Pekerja

Pekerja dapat melihat agenda yang ditugaskan kepadanya melalui menu agenda. Pada menu agenda pekerja yang ditampilkan hanya data agenda berupa tabel dengan kolom tanggal, nama, alamat, keluhan, dan status. Alih-alih melakukan pengelolaan agenda kerja, pekerja hanya bias memilih opsi selesai jika agenda kerja sudah diselesaikan. Opsi selesai juga berfungsi mengarahkan pekerja ke form tambah laporan untuk membuat laporan pekerjaan baru.

5.6.20 Rancangan Antarmuka Lihat Laporan – Pekerja



Gambar 5.26 Rancangan Antarmuka Lihat Laporan – Pekerja

Pekerja dapat melihat data laporan yang telah dikerjakan melalui menu laporan. Tabel laporan yang ditampilkan hanya memuat laporan yang diselesaikan pekerja itu sendiri. Terdapat opsi edit laporan yang mengarahkan pekerja ke form edit laporan dan melakukan perubahan data laporan jika diperlukan.

5.6.21 Rancangan Antarmuka Tambah Laporan – Pekerja

Gambar 5.27 Rancangan Antarmuka Tambah Laporan – Pekerja

Form tambah laporan memunculkan detail laporan yang otomatis dimuat dari agenda namun tetap dapat diubah oleh pekerja seperti tanggal, nama, alamat, uraian, dan keterangan. Sementara isian ID servis tidak dapat diubah karena bersifat *readonly*. Seluruh isian tambah laporan akan disimpan ketika pekerja memilih tombol simpan yang sekaligus mengarahkan pekerja ke form penilaian pekerja.

5.6.22 Rancangan Antarmuka Ubah Laporan - Pekerja

Gambar 5.28 Rancangan Antarmuka Ubah Laporan – Pekerja

Laporan pekerjaan dapat diubah dengan memilih opsi edit sesuai baris yang ingin diedit pada tabel tampilan data laporan dari sisi pekerja. Dalam form edit laporan terdapat isian yang dapat diubah oleh pekerja seperti tanggal, nama, alamat, uraian, dan keterangan. Adapun tombol simpan berguna untuk menyimpan data perubahan laporan dan mengarahkan kembali ke halaman tampilan data laporan pekerja.

5.6.23 Rancangan Antarmuka Form Penilaian

Gambar 5.29 Rancangan Antarmuka Form Penilaian

Form penilaian yang muncul ketika pekerja menyelesaikan pekerjaan memang dapat diakses dari sisi pekerja namun harus diisi oleh pelanggan karena membutuhkan nomor telepon untuk verifikasi pengisian penilaian. Form penilaian terdiri dari 2 jenis isian yaitu penilaian berupa bintang dengan rentang 5 bintang dan isian kuesioner singkat dengan 5 pilihan jawaban yang ditentukan. Terdapat tombol kirim penilaian yang berfungsi menyimpan data penilaian pekerja dan mengarahkan ke halaman status notifikasi yang akan dikirimkan berupa SMS ke pelanggan dan kepala UKM.

5.6.24 Rancangan Antarmuka Status SMS Notifikasi

Gambar 5.30 Rancangan Antarmuka Status SMS Notifikasi

Halaman yang menampilkan status notifikasi SMS untuk admin dan pelanggan hanya berisi pesan status pengiriman notifikasi dan sebuah tombol yang mengisyaratkan selesainya sebuah agenda maupun penilaian dengan pesan "Terimakasih". Tombol tersebut dapat mengarahkan pekerja kembali ke menu agenda pekerjaan yang memunculkan pesan bahwa sebuah agenda telah terselesaikan.

5.7 Implementasi Kode Program

Kode program diimplementasikan dengan *framework CodeIgniter* dan Bahasa php yang tentunya banyak perintah yang dibuat dengan *library CodeIgniter* sehingga mempersingkat penulisan beberapa *method* dengan hanya melakukan pemanggilan saja.

5.7.1 Fungsi Membuat *Booking* Baru

Fungsi simpan ini terletak dalam *class controller booking* yang digunakan untuk menyimpan data *booking* baru oleh pelanggan ke *database* program.

Tabel 5.6 Implementasi Fungsi Membuat *Booking* Baru

Implementasi 1: Fungsi Membuat Data Booking Baru	
1	public function simpan(){
2	\$id = date("ymdHis");
3	\$data = array(
4	'id_booking' => \$id,
5	'tanggal' => \$this->input->post("tanggal"),
6	'username' => \$this->session->userdata('ses_id'),
7	'pemesan' => \$this->input->post("nama"),
8	'alamat' => \$this->input->post("alamat"),
9	'telp' => \$this->input->post("telp"),
10	'keluhan' => \$this->input->post("keluhan"),
11	);
12	\$this->mBooking->simpan(\$data);
13	\$this->session->set_flashdata('notif', '<div class="alert alert-
14	success alert-dismissible"> Success! data berhasil disimpan.
15	</div>');
16	//redirect
17	redirect('cBooking');
18	}

Ketika pelanggan mengisi data *booking* baru pada form *booking*, data isian pelanggan akan dimasukkan ke array. Array tersebut kemudian dimasukkan ke *database* dengan perintah sesuai *query* MySQL yang dilakukan oleh model *mBooking*. Hasil *query* lalu dikembalikan baik jika berhasil maupun gagal. Data *booking* yang berhasil disimpan akan ditampilkan ke halaman *view* dengan pesan sesuai isi pada *flashdata*.

5.7.2 Fungsi Menghapus Data *Booking*

Fungsi hapus ini terletak dalam *class controller booking* yang digunakan untuk menghapus data *booking* oleh pelanggan.

Tabel 5.7 Implementasi Fungsi Menghapus Data *Booking*

Implementasi 2: Fungsi Menghapus Data Booking	
1	public function hapus(){
2	\$id['id_booking'] = \$this->uri->segment(3);
3	\$id2['id_servis'] = \$this->uri->segment(3);
4	\$data['data_booking'] = \$this->mBooking->getById(implode(\$id));
5	\$status = \$data['data_booking']->status;
6	
7	if (\$status == "booked"){
8	\$this->mBooking->hapus(\$id);
9	}
10	elseif (\$status == "process"){
11	\$this->mAgenda->hapus(\$id2);
12	\$this->mBooking->hapus(\$id);
13	}else{

```

14     $this->session->set_flashdata('notif', '<div class="alert
15 alert-success alert-dismissible"> Anda tidak dapat menghapus data
16 riwayat booking.</div>');
17     }
18     //redirect
19     redirect('cBooking');
20 }

```

Data *booking* pelanggan dapat dihapus berdasarkan *id_booking*. Jika pelanggan ingin menghapus data *booking* maka harus diperhatikan terlebih dahulu status *booking* tersebut. Jika status *booking* adalah “booked” berarti data *booking* tersebut belum masuk ke agenda pekerjaan sehingga dapat langsung dihapus. Namun jika status *booking* adalah “process” maka data agenda juga harus dihapus. Lain halnya jika status *booking* bukan keduanya, maka tidak bisa dihapus.

5.7.3 Fungsi Menghitung Rata-rata Rating

Fungsi *rataRating* ini terletak dalam *class controller* penilaian yang digunakan untuk menghitung nilai rata-rata rating yang diperoleh pekerja.

Tabel 5.8 Implementasi Fungsi Menghitung Rata-rata Rating

Implementasi 3: Fungsi Menghitung Rata-rata Rating

```

1 public function rataRating() {
2     $avg_rating = array();
3     $data['data_pekerja'] = $this->mPekerja->getPekerja();
4     $itr = 0;
5     foreach($data['data_pekerja'] as $hasil){
6         $pekerja = $hasil->username;
7         $rating = $this->mPenilaian->rataRating($pekerja);
8         $avg_rating[$itr] = $rating;
9         $itr++;
10    }
11    return $avg_rating;
12 }

```

Rata-rata rating dihitung dengan cara mengambil data rating setiap pekerja sesuai *username*. Lalu setiap data ditambahkan dan dihitung rata-ratanya melalui *class* model *mPenilaian* sehingga diperoleh data rata-rata sesuai pekerja. Perhitungan ini dilakukan melalui perulangan dan mengembalikan nilai *avg_rating* sebagai rata-rata keseluruhan data.

5.7.4 Fungsi Menghitung Rata-rata Nilai Kuesioner

Fungsi *rataNilai* ini terletak dalam *class controller* penilaian yang digunakan untuk menghitung nilai rata-rata nilai hasil pengisian kuesioner yang diperoleh pekerja.

Tabel 5.9 Implementasi Fungsi Menghitung Rata-rata Nilai Kuesioner

Implementasi 4: Fungsi Menghitung Rata-rata Nilai Kuesioner

```

1 public function rataNilai() {
2     $avg_nilai = array();
3     $max = 25;
4     $data['data_pekerja'] = $this->mPekerja->getPekerja();
5     $itr = 0;
6     foreach($data['data_pekerja'] as $hasil){
7         $pekerja = $hasil->username;
8         $nilai = $this->mPenilaian->rataNilai($pekerja);
9         $countPekerja = $this->mPenilaian->countPenilaian($pekerja);
10    if($countPekerja == 0){

```

```

11         $avg_nilai[$itr] = 0;
12     }else{
13         $avg_nilai[$itr] = ($nilai/$max)*100;
14     }
15     $itr++;
16 }
17 return $avg_nilai;
18 }

```

Rata-rata nilai akan dihitung saat seorang pekerja mendapatkan penilaian dari banyak pelanggan melalui agenda yang telah diselesaikan oleh pekerja tersebut. Seluruh pertanyaan kuesioner mencakup 5 pertanyaan dengan rentang nilai 1 untuk sangat kurang sampai 5 untuk sangat baik sehingga nilai maksimal pengisian kuesioner penilaian adalah 25. Kemudian, melalui perulangan, masing-masing pekerja dihitung banyak penilaian yang diperoleh dan disimpan dalam variable countPekerja. Sementara itu, seluruh nilai rata-rata penilaian seorang pekerja disimpan dalam variable nilai. Jika nilai countPekerja adalah 0 atau pekerja tersebut belum pernah diberi nilai yang mana berarti pekerja tersebut belum menyelesaikan satu agenda sama sekali, maka nilai avg_nilai atau ratarata nilai diisi dengan 0. Lain hal saat pekerja memiliki penilaian karena telah menyelesaikan pekerjaan maka avg_nilai-nya adalah rata-rata nilai yang diperoleh dibagi dengan nilai maksimal kemudian dikalikan 100 untuk data 100%. Demikian perulangan dilakukan dengan menaikkan nilai variable itr agar data dalam array terus bertambah sesuai banyaknya pekerja.

5.7.5 Fungsi Menampilkan Data Penilaian

Fungsi index ini terletak dalam *class controller* penilaian yang digunakan untuk menampilkan data penilaian baik kepada admin maupun form penilaian untuk akses pekerja.

Tabel 5.10 Implementasi Fungsi Menampilkan Data Penilaian

Implementasi 5: Fungsi Menampilkan Data Penilaian	
1	public function index(){
2	if(\$this->session->userdata('masuk') != TRUE){
3	redirect('cLogin');
4	}else{
5	if (\$this->session->userdata('akses')==1){
6	\$data = array(
7	'title' => 'Data Penilaian',
8	'data_penilaian' => \$this->mPenilaian->getAll(),
9	'data_pekerja' => \$this->mPekerja->getPekerja(),
10	'data_rating' => \$this->rataRating(),
11	'data_nilai' => \$this->rataNilai(),
12	);
13	\$this->load->view('admin\vPenilaian', \$data);
14	}elseif (\$this->session->userdata('akses')==2) {
15	if(\$this->session->userdata('id_penilaian')) {
16	\$id = \$this->session->userdata('id_penilaian');
17	\$data = array(
18	'title' => 'Data Penilaian',
19	'data_penilaian' => \$this->mPenilaian->getById(\$id),
20	);
21	\$this->load->view('pekerja\vPenilaian', \$data);
22	}else{
23	\$this->session->set_flashdata('notif', '<div class="alert
24	alert-success alert-dismissible"> Anda tidak bisa mengakses Form
25	Penilaian.</div>');
26	redirect('cAgenda');

```

27     }
28     }else{
29         $this->session->set_flashdata('notif', '<div class="alert
30 alert-success alert-dismissible"> Anda tidak berhak mengakses
31 penilaian.</div>');
32         redirect('cPage');
33     }
34 }
35 }

```

Saat pengguna yang mengakses halaman penilaian belum *login*, pengguna akan diarahkan untuk melakukan *login* terlebih dahulu. Namun jika pengguna sudah *login*, akan dicek session akses pengguna dalam if. Jika pengguna memiliki hak akses 1 yaitu admin, halaman yang tampil adalah halaman yang berisi data penilaian seluruh pekerja. Lain halnya jika pengguna memiliki hak akses 2 yaitu pekerja, dilakukan pengecekan lagi terhadap maksud pekerja tersebut mengakses halaman penilaian. Jika untuk mengakses form isian penilaian setelah menyelesaikan pekerjaan, akan diambil id_servis dari pekerjaan yang belum dinilai tersebut sehingga dapat menampilkan form isi penilaian dan kuesioner.

5.8 Implementasi Antarmuka

5.8.1 Implementasi Antarmuka Form Pendaftaran

Daftar Teja
(Daftarkan akun baru)

Pengguna baru? Daftar sekarang!

Sudah memiliki akun? [Login disini](#)

Gambar 5.31 Implementasi Antarmuka Form Pendaftaran

Implementasi antarmuka form pendaftaran berisikan data pendaftaran untuk seorang pelanggan baru. Masing-masing isian seperti nama, *username*, email, *password*, telepon dan alamat harus diisi guna mempercepat proses *booking* servis. Terdapat pula tombol Daftar di sebelah kiri bawah untuk menyimpan data pendaftaran pelanggan baru. Sementara itu, jika pengguna sudah memiliki akun dapat melakukan *login* dengan memilih link *Login* disini.

5.8.2 Implementasi Antarmuka Form Login

Selamat Datang di Teja

(Silakan masuk untuk mendapatkan layanan)



Masukkan data Login

Username

Password

☐ Remember Me

Login

Belum memiliki akun? [klik disini](#)

Gambar 5.32 Implementasi Antarmuka Form Login

Form *login* dapat dikatakan sebagai perantara masuknya pengguna ke sistem. Pengguna dapat masuk ke sistem hanya dengan mengisi field *username* dan *password* yang telah diinputkan ke pendaftaran sebelumnya. Adapun tombol *Login* berfungsi melakukan pengecekan pengguna agar dapat masuk ke sistem atau menampilkan pesan *error* jika *username* atau *password* pengguna tidak ditemukan. Terdapat pula link untuk mengarahkan pengguna ke form pendaftaran jika pengguna belum memiliki akun.

5.8.3 Implementasi Antarmuka Lihat *Booking* – Pelanggan

No.	Tanggal	Nama Pelanggan	Alamat	Telepon	Keluhan	Status	Options
1	2019-05-23	Irma Arini	Jl. Veteran No 1 Malang	081234567890	Cuci AC	booked	Edit Hapus

No.	Tanggal	Nama Pelanggan	Alamat	Telepon	Keluhan	Status
1	2019-05-20	Irma Arini	Jl. Veteran No 1 Malang	081234567890	ac mati	done
2	2019-05-17	Irma Arini	Jl. Veteran No 1 Malang	081234567890	AC mati	done
3	2019-05-14	Irma Arini	Jl. Veteran No 1 Malang	081234567890	Keran bocor	done
4	2019-05-15	Irma Arini	Jl. Veteran No 1 Malang	081234567890	AC mati	reviewed

Gambar 5.33 Implementasi Antarmuka Lihat *Booking* – Pelanggan

Pengguna yang masuk sebagai pelanggan memperoleh layanan *Booking* dan Testimoni. Dalam menu *Booking*, pelanggan dapat melihat data *booking* yang pernah dilakukan ditampilkan berupa tabel data dengan opsi edit maupun hapus di masing-masing barisnya. Adapun opsi edit berguna untuk mengarahkan pelanggan ke form edit *booking* dengan isian yang diambil dari data *booking* sebelumnya dan opsi hapus berguna untuk menghapus *booking* yang pernah dibuat. Sementara itu, tombol tambah *booking* berguna untuk mengarahkan pelanggan ke form tambah *booking* baru.

5.8.4 Implementasi Antarmuka Tambah *Booking* – Pelanggan

Gambar 5.34 Implementasi Antarmuka Tambah *Booking* – Pelanggan

Jika pelanggan memilih tombol Tambah *Booking* di menu tampilan *booking* maka akan masuk ke form tambah *booking*. Terdapat isian yang harus diisi seperti tanggal, nama, alamat, telepon dan keluhan yang otomatis diambil dari data pelanggan namun tetap dapat diubah untuk keperluan *booking*. Sementara tombol simpan berguna untuk menyimpan data *booking* ke *database* serta mengarahkan kembali pelanggan ke tampilan data *booking* dengan pesan penyimpanan data berhasil.

5.8.5 Implementasi Antarmuka Ubah *Booking* – Pelanggan

Gambar 5.35 Implementasi Antarmuka Ubah *Booking* – Pelanggan

Form edit *booking* muncul jika opsi edit di masing-masing baris data *booking* dipilih. Form edit *booking* menampilkan isian data *booking* sesuai data *booking* yang dipilih untuk diedit. Seluruh data *booking* yang pernah diisikan seperti tanggal, nama, alamat, telepon dan keluhan dapat diubah kecuali ID *booking*. Tombol simpan berfungsi menyimpan data perubahan *booking* yang dilakukan.

5.8.6 Implementasi Antarmuka Lihat Testimoni – Pelanggan

No.	Tanggal	Nama Pelanggan	Alamat	Telepon	Keluhan	Status	Options
1	2019-05-20	Irma Arini	Jl. Veteran No 1 Malang	081234567890	ac mati	done	Beri Testimoni
2	2019-05-17	Irma Arini	Jl. Veteran No 1 Malang	081234567890	AC mati	done	Beri Testimoni
3	2019-05-14	Irma Arini	Jl. Veteran No 1 Malang	081234567890	Keran bocor	done	Beri Testimoni

No.	ID Testi	Username	Testimoni	Options
1	190513202443	Irma	Staf ramah, pelayanan OK	Hapus

Gambar 5.36 Implementasi Antarmuka Lihat Testimoni – Pelanggan

Menu testimoni pelanggan berisikan data riwayat *booking* dan data testimoni. Testimoni hanya dapat diberikan jika status *booking* sudah selesai. Data *booking* yang sudah selesai namun belum diberi testimoni akan ditampilkan pada tabel riwayat *booking* maka terdapat opsi Beri Testimoni. Pada tabel data testimoni terdapat opsi hapus untuk menghapus testimoni yang pernah dibuat.

5.8.7 Implementasi Antarmuka Tambah Testimoni – Pelanggan

Gambar 5.37 Implementasi Antarmuka Tambah Testimoni – Pelanggan

Halaman pengisian testimoni hanya berisi isian testimoni dengan ID yang diambil dari data *booking* dan tidak dapat diubah. Isian testimoni berupa textarea memudahkan pelanggan mengisi banyak kata maupun kalimat terkait testimoni yang dirasakan pasca layanan servis yang diperoleh. Adapun tombol Tambah berguna untuk menambahkan data isian testimoni ke *database* sistem.

5.8.8 Implementasi Antarmuka Lihat *Booking* – Admin

No.	Tanggal	Nama Pelanggan	Alamat	Telepon	Keluhan	Status
1	2019-05-06	Diana Rani	Jl. Merpati Putih 201	081337672727	Kompor meledak	reviewed
2	2019-05-06	Irma Arini	Jl. Veteran No 1 Malang	081234567890	Saklar lampu rusak	reviewed
3	2019-05-06	Irma Arini	Villa Indah, Jl. Raya Bukit Berbunga	081234567890	Pasang CCTV teras	done
4	2019-05-08	Irma Arini	Jl. Veteran No 1 Malang	081234567890	Mesin cuci rusak	process
5	2019-05-08	Dewi Ariani	Jl. Kebun Mangga	081234567890	Ganti lampu kolam renang	booked
6	2019-05-09	Dewi Ariani	Jl. Kebun Mangga	081234567890	Kulkas tidak dingin	booked
7	2019-05-10	Irma Arini	Jl. Veteran No 1 Malang	081234567890	Cuci AC	booked

Gambar 5.38 Implementasi Antarmuka Lihat *Booking* – Admin

Sebagai admin, pengguna pun dapat melihat data seluruh *booking* melalui menu *booking*. Data seluruh *booking* ditampilkan dalam bentuk tabel namun tidak ada opsi apapun karena admin tidak dapat melakukan pengelolaan data *booking*.

5.8.9 Implementasi Antarmuka Lihat Agenda – Admin

No.	Tanggal	Nama Pelanggan	Alamat	Telepon	Keluhan	Status	Options
1	2019-05-08	Dewi Ariani	Jl Kebun Mangga	081234567890	Ganti lampu kolam renang	booked	Tambahkan ke Agenda
2	2019-05-09	Dewi Ariani	Jl Kebun Mangga	081234567890	Kulkas tidak dingin	booked	Tambahkan ke Agenda
3	2019-05-10	Irma Arini	Jl. Veteran No 1 Malang	081234567890	Cuci AC	booked	Tambahkan ke Agenda

No.	ID Servis	Tanggal	Nama	Alamat	Telepon	Keluhan	Pekerja	Status	
1	190508090106	2019-05-08	Irma Arini	Jl. Veteran No 1 Malang	081234567890	Mesin cuci rusak	dodi	process	Edit Hapus

Gambar 5.39 Implementasi Antarmuka Lihat Agenda – Admin

Admin dapat melihat agenda yang belum terselesaikan berikut opsi untuk mengedit dan menghapus data agenda. Data agenda yang ditampilkan tentunya berasal dari *booking* sehingga jika ingin menambahkan agenda baru, terdapat tabel antrian *booking* yang belum dikerjakan. Antrian *booking* yang ditampilkan memiliki opsi untuk admin dapat menambahkannya ke agenda melalui form tambah data agenda yang diarahkan saat opsi dipilih.

5.8.10 Implementasi Antarmuka Tambah Agenda – Admin

Gambar 5.40 Implementasi Antarmuka Tambah Agenda – Admin

Data *booking* yang ingin ditambahkan ke agenda akan ditampilkan secara detail di form tambah agenda. Terdapat isian tanggal, nama, alamat, telepon, dan keluhan yang sudah otomatis diisi berdasarkan data *booking* namun tetap dapat diubah sesuai kebutuhan. Sementara isian pekerja berupa combobox adalah pilihan nama pekerja yang ditugaskan oleh admin mengerjakan agenda tersebut. Adapun tombol simpan berfungsi untuk menyimpan data agenda baru dan mengarahkan kembali ke menu agenda.

5.8.11 Implementasi Antarmuka Ubah Agenda – Admin

Gambar 5.41 Implementasi Antarmuka Ubah Agenda – Admin

Ketika admin memilih opsi edit pada salah satu baris agenda dalam tabel, admin akan diarahkan ke form edit agenda. Terdapat isian tanggal, nama, alamat, telepon, keluhan dan pekerja yang dapat dibuuh namun tidak pada ID servis yang diperoleh dari ID *booking*. Sementara tombol simpan akan menyimpan perubahan data agenda dan mengembalikan admin ke menu tampilan agenda.

5.8.12 Implementasi Antarmuka Lihat Laporan – Admin

No.	ID Servis	Tanggal	Nama	Alamat	Uraian	Keterangan	Pekerja	Biaya	Status
1	190506204918	2019-05-06	Irma Arini	Villa Indah, Jl Raya Bukit Berbunga	Pasang CCTV teras	-	dodi	0	belum
2	190506204833	2019-05-06	Irma Arini	Jl. Veteran No 1 Malang	Saklar lampu rusak	Ganti kabel lampu putus	dodi	80000	belum

No.	ID Servis	Tanggal	Nama	Alamat	Uraian	Keterangan	Pekerja	Biaya	Status
1	190506204740	2019-05-06	Diana Rani	Jl. Merdeka Puri 201	Kompor meledak	Ganti selang	budi	300000	lunas

Gambar 5.42 Implementasi Antarmuka Lihat Laporan – Admin

Dalam menu laporan, admin dapat melihat tampilan data laporan berupa tabel yang hanya memiliki opsi edit. Ketika opsi edit dipilih, admin akan diarahkan ke form edit laporan.

5.8.13 Implementasi Antarmuka Ubah Laporan – Admin

Form edit laporan muncul ketika opsi edit laporan dipilih serta secara otomatis menampilkan detail data laporan ke masing-masing field input. Terdapat field yang dapat diubah seperti tanggal, nama, alamat, uraian, keterangan, pekerja, biaya, dan status pembayaran namun tidak untuk ID servis. Adapun tombol simpan berfungsi untuk menyimpan data laporan yang telah diubah dan mengembalikan admin ke menu tampilan laporan.

Gambar 5.43 Implementasi Antarmuka Ubah Laporan – Admin

Form edit laporan muncul ketika opsi edit laporan dipilih serta secara otomatis menampilkan detail data laporan ke masing-masing field input. Terdapat field yang dapat diubah seperti tanggal, nama, alamat, uraian, keterangan, pekerja, biaya, dan status pembayaran namun tidak untuk ID servis. Adapun tombol simpan berfungsi untuk menyimpan data laporan yang telah diubah dan mengembalikan admin ke menu tampilan laporan.

5.8.14 Implementasi Antarmuka Lihat Penilaian – Admin

Penilaian Pekerja

No.	Pekerja	Rating	Nilai
1	agus	0	0%
2	budi	4	88%
3	dodi	4.7	88%

Data Penilaian dari Pelanggan

No.	ID Servis	Pekerja	Rating	Nilai
1	190508090106	dodi	4	20
2	190506204918	dodi	5	22
3	190506204833	dodi	5	24
4	190506204740	budi	4	22

Gambar 5.44 Implementasi Antarmuka Lihat Penilaian – Admin

Admin dapat melihat penilaian pekerja yang diberikan oleh pelanggan. Tampilan penilaian pekerja terdiri dari 2 tabel yaitu tabel yang memuat rata-rata perolehan nilai pekerja dan tabel yang memuat seluruh data penilaian pekerja. Tidak terdapat opsi apapun pada menu penilaian dari sisi admin.

5.8.15 Implementasi Antarmuka Lihat Testimoni – Admin

The screenshot shows the Tekindo Jaya Admin interface. The header includes the logo, the name 'Tekindo Jaya', the date 'Wednesday, 08 May 2019', and the user 'Andi Sudirman' with a 'Logout' button. The left sidebar contains navigation links: Dashboard, Booking, Agenda Kerja, Laporan, Penilaian Kinerja, and Testimoni. The main content area displays a table titled 'Data Testimoni' with the following data:

No.	ID Testi	Username	Testimoni
1	190506204918	Irma	Staf ramah, namun mohon tambah staf agar tidak lama antri servis
2	190506204833	Irma	Terimakasih Bali Tekindo Jaya, AC saya kembali normal
3	190506204740	dianarani10	puas, pelayanan berkualitas

At the bottom right, there is a watermark for 'UNIVERSITAS BRAWIJAYA' and a Windows activation notice: 'Activate Windows Go to Settings to activate Windows.'

Gambar 5.45 Implementasi Antarmuka Lihat Testimoni – Admin

Menu testimoni dari sisi admin hanya menampilkan data testimoni berbentuk tabel yang berisi data ID testi, nama pelanggan yang memberikan testimoni, dan data testimoni itu sendiri. Tidak terdapat opsi apapun dalam tabel testimoni karena admin tidak berhak melakukan pengelolaan testimoni.

5.8.16 Implementasi Antarmuka Lihat Agenda – Pekerja

The screenshot shows the Tekindo Jaya Worker interface. The header includes the logo, the name 'Tekindo Jaya', the date 'Wednesday, 08 May 2019', and the user 'Dodi Rahmadi' with a 'Logout' button. The left sidebar contains navigation links: Dashboard, Laporan, and Agenda Kerja. The main content area displays a table titled 'Agenda Pekerjaan' with the following data:

No.	ID Servis	Tanggal	Nama	Alamat	Telepon	Keluhan	Status	
1	190508190812	2019-05-10	Irma Arini	Jl. Veteran No 1 Malang	081234567890	Cuci AC	process	Selesai
2	190508090106	2019-05-08	Irma Arini	Jl. Veteran No 1 Malang	081234567890	Mesin cuci rusak	process	Selesai

At the bottom right, there is a watermark for 'UNIVERSITAS BRAWIJAYA'.

Gambar 5.46 Implementasi Antarmuka Lihat Agenda – Pekerja

Pekerja dapat melihat agenda yang ditugaskan kepadanya melalui menu agenda. Pada menu agenda pekerja yang ditampilkan hanya data agenda berupa tabel dengan kolom tanggal, nama, alamat, keluhan, dan status. Alih-alih melakukan pengelolaan agenda kerja, pekerja hanya bias memilih opsi selesai jika agenda kerja sudah diselesaikan. Opsi selesai juga berfungsi mengarahkan pekerja ke form tambah laporan untuk membuat laporan pekerjaan baru.

5.8.17 Implementasi Antarmuka Lihat Laporan – Pekerja

No.	ID Servis	Tanggal	Nama	Alamat	Uraian	Keterangan	
1	190506204833	2019-05-06	Irma Arini	Jl. Veteran No 1 Malang	Saklar lampu rusak	Ganti kabel lampu putus	Edit
2	190506204918	2019-05-06	Irma Arini	Villa Indah, Jl Raya Bukit Berbunga	Pasang CCTV teras	tambahan benahi plafon	Edit
3	190508090106	2019-05-08	Irma Arini	Jl. Veteran No 1 Malang	Mesin cuci rusak	-	Edit

Gambar 5.47 Implementasi Antarmuka Lihat Laporan – Pekerja

Pekerja dapat melihat data laporan yang telah dikerjakan melalui menu laporan. Tabel laporan yang ditampilkan hanya memuat laporan yang diselesaikan pekerja itu sendiri. Terdapat opsi edit laporan yang mengarahkan pekerja ke form edit laporan dan melakukan perubahan data laporan jika diperlukan.

5.8.18 Implementasi Antarmuka Tambah Laporan – Pekerja

Gambar 5.48 Implementasi Antarmuka Tambah Laporan – Pekerja

Form tambah laporan memunculkan detail laporan yang otomatis dimuat dari agenda namun tetap dapat diubah oleh pekerja seperti tanggal, nama, alamat, uraian, dan keterangan. Sementara isian ID servis tidak dapat diubah karena bersifat *readonly*. Seluruh isian tambah laporan akan disimpan ketika pekerja memilih tombol simpan yang sekaligus mengarahkan pekerja ke form penilaian pekerja.

5.8.19 Implementasi Antarmuka Ubah Laporan – Pekerja

Gambar 5.49 Implementasi Antarmuka Ubah Laporan – Pekerja

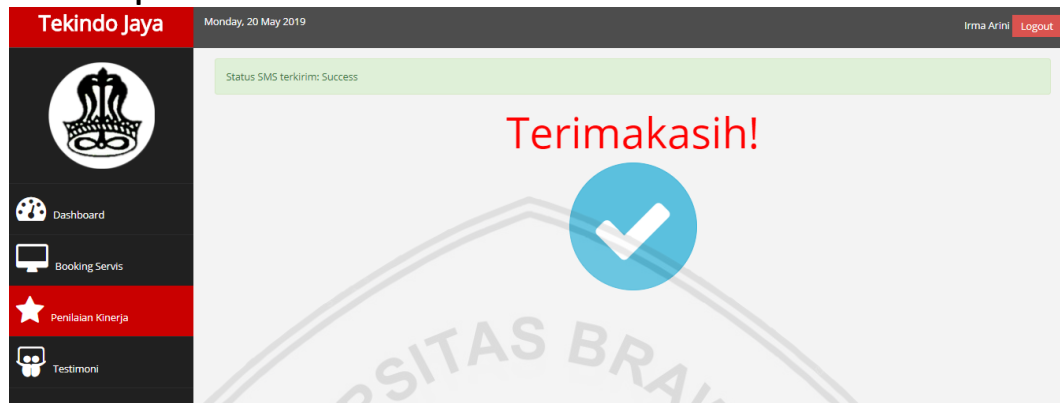
Laporan pekerjaan dapat diubah dengan memilih opsi edit sesuai baris yang ingin diedit pada tabel tampilan data laporan dari sisi pekerja. Dalam form edit laporan terdapat isian yang dapat diubah oleh pekerja seperti tanggal, nama, alamat, uraian, dan keterangan. Adapun tombol simpan berguna untuk menyimpan data perubahan laporan dan mengarahkan kembali ke halaman tampilan data laporan pekerja.

5.8.20 Implementasi Antarmuka Form Penilaian

Gambar 5.50 Implementasi Antarmuka Form Penilaian

Form penilaian muncul ketika pekerja menyelesaikan pekerjaan dan diisi oleh pelanggan. Form penilaian terdiri dari 2 jenis isian yaitu penilaian berupa bintang dengan rentang 5 bintang dan isian kuesioner singkat dengan 5 pilihan jawaban yang ditentukan. Terdapat tombol kirim penilaian yang berfungsi menyimpan data penilaian pekerja dan mengarahkan ke halaman status notifikasi yang akan dikirimkan berupa SMS ke kepala UKM.

5.8.21 Implementasi Antarmuka Status SMS Notifikasi



Gambar 5.51 Implementasi Antarmuka Status SMS Notifikasi

Halaman yang menampilkan status notifikasi SMS untuk admin hanya berisi pesan status pengiriman notifikasi dan sebuah tombol yang mengisyaratkan selesainya sebuah agenda maupun penilaian dengan pesan “Terimakasih”. Tombol tersebut dapat mengarahkan pelanggan kembali ke menu penilaian pekerja yang memunculkan pesan bahwa sebuah pekerjaan telah terselesaikan.

BAB 6 PENGUJIAN DAN ANALISIS

Bagian pengujian dan analisis memuat hasil pengujian sistem yang telah dikembangkan yaitu pengujian kotak hitam (*blackbox*) dan kotak putih (*whitebox*). Hasil pengujian dengan kedua metode tersebut kemudian akan dianalisis untuk menentukan capaian sistem pada analisis hasil pengujian.

6.1 Pengujian *Blackbox*

Dalam pengujian *blackbox* akan dilakukan pengujian validasi terhadap setiap kebutuhan yang telah didefinisikan dalam spesifikasi kebutuhan fungsional sistem. Terdapat rencana kasus uji yang menggambarkan perbandingan antara hasil yang diberikan sistem dengan hasil yang sebenarnya diharapkan pengguna. Berikut adalah kasus uji masing-masing kebutuhan sistem serta status validasi setiap fungsional sistem:

6.1.1 Kasus Uji Fungsional *Login*

Tabel 6.1 Kasus Uji Fungsional *Login*

Kasus Uji - <i>Login</i>	
Objek Uji	TEJA-1-01
Tujuan Pengujian	Memastikan pengguna masuk ke halaman sistem
Prosedur Uji	1. Pengguna memasukkan <i>username</i> "irmarini" 2. Pengguna memasukkan <i>password</i> "*****" 3. Pengguna menekan tombol <i>login</i> 4. Sistem menampilkan halaman utama sistem
Hasil yang Diharapkan	Pengguna dapat masuk ke sistem dengan memasukkan <i>username</i> dan <i>password</i>
Hasil Pengujian	Pengguna dapat masuk ke sistem
Status Validasi	Valid

Analisis:

Berdasarkan hasil rekayasa kebutuhan fungsional, fungsi *login* merupakan kebutuhan sistem dari sisi seluruh aktor baik admin, pelanggan, maupun pekerja. Pengujian kebutuhan fungsional *login* dilakukan dengan prosedur uji sesuai *usecase* hingga memperoleh status valid yang berarti kebutuhan fungsional *login* dapat diimplementasikan dan telah berjalan dengan baik pada sistem.

6.1.2 Kasus Uji Fungsional *Logout*

Tabel 6.2 Kasus Uji Fungsional *Logout*

Kasus Uji - <i>Logout</i>	
Objek Uji	TEJA-1-02
Tujuan Pengujian	Memastikan pengguna dapat keluar dari sistem
Prosedur Uji	1. Pengguna menekan tombol <i>logout</i> 2. Pengguna keluar dari sistem
Hasil yang Diharapkan	Pengguna yang sedang <i>login</i> dapat keluar dari sistem dengan melakukan <i>logout</i>

Hasil Pengujian	Pengguna dapat keluar dari sistem
Status Validasi	Valid

Analisis:

Seluruh pengguna yang telah *login* dapat melakukan *logout* dari sistem. Kebutuhan fungsional *logout* dilakukan dengan prosedur uji yang sesuai dengan *usecase logout* hingga memperoleh status valid. Status valid tersebut menunjukkan fungsional *login* dapat berjalan sesuai kebutuhan fungsional *logout* yang telah dijabarkan dalam rekayasa kebutuhan.

6.1.3 Kasus Uji Fungsional Pendaftaran

Tabel 6.3 Kasus Uji Fungsional Pendaftaran

Kasus Uji - Daftar	
Objek Uji	TEJA-1-03
Tujuan Pengujian	Memastikan pengguna memiliki akun untuk dapat masuk ke sistem
Prosedur Uji	1. Pengguna memasukkan email: irmarini15@gmail.com, <i>username</i>:irmarini, <i>password</i>: *****, nama lengkap: Irma Arini, nomor telepon: 081234567890, alamat: Jl Veteran 2. Pengguna menekan tombol daftar 3. Sistem menampilkan pesan berhasil membuat akun
Hasil yang Diharapkan	Pelanggan dapat melakukan pendaftaran sistem dengan memasukkan email, <i>username</i> , <i>password</i> , nama lengkap, nomor telepon, dan alamat.
Hasil Pengujian	Pelanggan dapat melakukan pendaftaran ke sistem
Status Validasi	Valid

Analisis:

Berdasarkan hasil rekayasa kebutuhan fungsional, fungsi daftar merupakan kebutuhan sistem dari sisi pelanggan sebagai aktor pengguna sistem. Pengujian kebutuhan fungsional daftar dilakukan dengan prosedur uji sesuai *usecase* daftar hingga memperoleh status valid yang berarti kebutuhan fungsional daftar dapat diimplementasikan dan telah berjalan dengan baik pada sistem.

6.1.4 Kasus Uji Fungsional Melihat Data *Booking*

Tabel 6.4 Kasus Uji Fungsional Lihat *Booking*

Kasus Uji – Lihat <i>Booking</i>	
Objek Uji	TEJA-1-04
Tujuan Pengujian	Memastikan pengguna dapat melihat data <i>booking</i>
Prosedur Uji	Pelanggan melihat data <i>booking</i> yang telah dibuat sebelumnya dengan memilih menu <i>booking</i>
Hasil yang Diharapkan	Pelanggan dan admin dapat melihat data riwayat <i>booking</i> baik yang sudah dikerjakan maupun belum.

Hasil Pengujian	Data riwayat <i>booking</i> berhasil ditampilkan
Status Validasi	Valid

Analisis:

Melihat data *booking* merupakan kebutuhan fungsional yang berasal dari sisi admin dan pelanggan. Prosedur uji fungsional melihat data *booking* telah dilakukan sesuai dengan *usecase* lihat *booking* sehingga menghasilkan hasil yang sesuai harapan dengan status valid. Status valid menunjukkan fungsional *booking* dapat berjalan dengan baik pada sistem.

6.1.5 Kasus Uji Fungsional Membuat *Booking* Baru

Tabel 6.5 Kasus Uji Fungsional *Booking* Baru

Kasus Uji – <i>Booking</i> Baru	
Objek Uji	TEJA-1-05
Tujuan Pengujian	Memastikan pelanggan dapat membuat <i>booking</i> servis baru
Prosedur Uji	1. Pelanggan mengisi data nama pemesan: Irma Arini, alamat servis: Jl Veteran, dan keluhan servis: AC mati 2. Pelanggan memilih simpan <i>booking</i> 3. Sistem menampilkan pesan data <i>booking</i> telah disimpan
Hasil yang Diharapkan	Pelanggan dapat membuat <i>booking</i> baru melalui form <i>booking</i> layanan servis.
Hasil Pengujian	Pelanggan dapat membuat <i>booking</i> baru
Status Validasi	Valid

Analisis:

Berdasarkan hasil rekayasa kebutuhan fungsional, fungsi *booking* baru merupakan kebutuhan sistem dari sisi pelanggan. Pengujian kebutuhan fungsional *booking* baru dilakukan dengan prosedur uji sesuai *usecase* hingga memperoleh status valid yang berarti kebutuhan fungsional *booking* baru dapat diimplementasikan dan telah berjalan dengan baik pada sistem.

6.1.6 Kasus Uji Fungsional Mengubah Data *Booking*

Tabel 6.6 Kasus Uji Fungsional Ubah *Booking*

Kasus Uji – Ubah <i>Booking</i>	
Objek Uji	TEJA-1-06
Tujuan Pengujian	Memastikan pelanggan dapat melakukan perubahan pada data <i>booking</i>
Prosedur Uji	1. Pelanggan memilih pilihan ubah data <i>booking</i> servis 2. Pelanggan melakukan perubahan data nama pemesan: Diana Rani, alamat servis: Jl Gunung Guntur, maupun keluhan servis: remote AC mati. 3. Pelanggan menyimpan perubahan dengan

	memilih tombol simpan
Hasil yang Diharapkan	Pelanggan dapat mengubah data <i>booking</i> yang telah dibuat sebelumnya.
Hasil Pengujian	Pelanggan dapat mengubah data <i>booking</i>
Status Validasi	Valid

Analisis:

Pengguna yang memiliki hak akses sebagai pelanggan memiliki kebutuhan fungsional untuk mengubah data *booking*. Kebutuhan fungsional ubah *booking* dilakukan dengan prosedur uji yang sesuai dengan *usecase* hingga memperoleh status valid. Status valid tersebut menunjukkan ubah *booking* dapat berjalan sesuai kebutuhan fungsional yang telah dijabarkan dalam rekayasa kebutuhan.

6.1.7 Kasus Uji Fungsional Menghapus Data *Booking*

Tabel 6.7 Kasus Uji Fungsional Batal *Booking*

Kasus Uji – Batal <i>Booking</i>	
Objek Uji	TEJA-1-07
Tujuan Pengujian	Memastikan pelanggan dapat melakukan pembatalan <i>booking</i> servis
Prosedur Uji	1. Pelanggan memilih pilihan hapus <i>booking</i> servis 2. Sistem menampilkan peringatan konfirmasi pembatalan 3. Pelanggan melakukan konfirmasi pembatalan <i>booking</i> servis
Hasil yang Diharapkan	Pelanggan dapat melakukan pembatalan <i>booking</i> dengan menghapus <i>booking</i> yang telah dibuat sebelumnya.
Hasil Pengujian	Pelanggan dapat menghapus <i>booking</i>
Status Validasi	Valid

Analisis:

Menghapus data *booking* merupakan kebutuhan fungsional yang berasal dari sisi pelanggan. Prosedur uji fungsional batal *booking* telah dilakukan sesuai dengan *usecase* batal *booking* sehingga menghasilkan hasil yang sesuai harapan dengan status valid. Status valid menunjukkan fungsional batal *booking* dapat berjalan sesuai harapan.

6.1.8 Kasus Uji Fungsional Memberi Penilaian

Tabel 6.8 Kasus Uji Fungsional Beri Penilaian

Kasus Uji – Beri Penilaian	
Objek Uji	TEJA-1-08
Tujuan Pengujian	Memastikan pelanggan dapat memberikan penilaian terhadap kinerja pekerja di lapangan
Prosedur Uji	1. Pelanggan mengisi rentang penilaian 4 bintang 2. Pelanggan mengisi jawaban dari pertanyaan kuesioner

	3. Pelanggan memilih kirim penilaian 4. Sistem menampilkan pesan berhasil menyimpan penilaian
Hasil yang Diharapkan	Pelanggan dapat memberi penilaian terhadap pekerja dengan rentang 1 sampai 5 bintang dan memberi jawaban kuesioner kinerja melalui agenda pekerja yang telah diselesaikan.
Hasil Pengujian	Pelanggan dapat memberi penilaian terhadap pekerja
Status Validasi	Valid

Analisis:

Berdasarkan hasil rekayasa kebutuhan fungsional, fungsi beri penilaian merupakan kebutuhan sistem dari sisi pelanggan. Pengujian kebutuhan fungsional beri penilaian dilakukan dengan prosedur uji sesuai *usecase* hingga memperoleh status valid yang berarti kebutuhan fungsional beri penilaian dapat diimplementasikan dan telah berjalan dengan baik pada sistem.

6.1.9 Kasus Uji Fungsional Kirim Notifikasi

Tabel 6.9 Kasus Uji Fungsional Kirim Notifikasi

Kasus Uji – Terima Notifikasi	
Objek Uji	TEJA-1-10
Tujuan Pengujian	Memastikan sistem dapat mengirim notifikasi berupa SMS kepada admin
Prosedur Uji	1. Sistem mengirim SMS laporan bahwa pekerja telah menyelesaikan pekerjaan sesuai agenda kepada Admin
Hasil yang Diharapkan	Sistem harus dapat secara otomatis mengirimkan notifikasi berupa SMS kepada admin yang berisi laporan pekerjaan telah terselesaikan.
Hasil Pengujian	Sistem dapat mengirimkan notifikasi secara otomatis berupa SMS
Status Validasi	Valid

Analisis:

Sistem dapat mengirim notifikasi otomatis kepada admin setelah pelanggan memberi penilaian kinerja pekerja. Kebutuhan fungsional kirim notifikasi dilakukan dengan prosedur uji yang sesuai dengan kebutuhan dalam bab rekayasa kebutuhan. Kasus uji yang dilaksanakan menghasilkan status valid yang berarti fungsi kirim notifikasi dapat berjalan sesuai penjabaran dalam kebutuhan fungsional.

6.1.10 Kasus Uji Fungsional Membuat Testimoni Baru

Tabel 6.10 Kasus Uji Fungsional Beri Testimoni

Kasus Uji – Beri Testimoni	
Objek Uji	TEJA-1-11
Tujuan Pengujian	Memastikan pelanggan dapat menambahkan testimoni baru
Prosedur Uji	1. Pengguna memilih pilihan beri testimoni 2. Pengguna mengisi testimoni: pelayanan ok, staf sopan 3. Pengguna memilih simpan testimoni 4. Sistem menampilkan pesan testimoni telah berhasil disimpan dan ditampilkan ke halaman
Hasil yang Diharapkan	Pelanggan dapat memberikan testimoni baru setelah servis diselesaikan oleh pekerja.
Hasil Pengujian	Pelanggan dapat memberikan testimoni servis
Status Validasi	Valid

Analisis:

Membuat testimoni baru merupakan kebutuhan fungsional yang berasal dari sisi pelanggan. Prosedur uji fungsional membuat testimoni baru telah dilakukan sesuai dengan *usecase* dalam bab rekayasa kebutuhan sehingga menghasilkan hasil yang sesuai harapan dengan status valid. Status valid tersebut menunjukkan bahwa fungsional buat testimoni baru dapat berjalan sesuai harapan.

6.1.11 Kasus Uji Fungsional Menghapus Testimoni

Tabel 6.11 Kasus Uji Fungsional Hapus Testimoni

Kasus Uji – Hapus Testimoni	
Objek Uji	TEJA-1-12
Tujuan Pengujian	Memastikan pelanggan dapat menghapus testimoni yang telah dibuat sebelumnya
Prosedur Uji	1. Pengguna memilih pilihan hapus testimoni 2. Sistem menghapus testimoni
Hasil yang Diharapkan	Pelanggan dapat menghapus testimoni yang telah dibuat sebelumnya.
Hasil Pengujian	Pelanggan dapat menghapus testimoni
Status Validasi	Valid

Analisis:

Berdasarkan hasil rekayasa kebutuhan fungsional, fungsi menghapus testimoni merupakan kebutuhan sistem dari sisi pelanggan. Pengujian kebutuhan fungsional hapus testimoni dilakukan dengan prosedur uji sesuai *usecase* hingga memperoleh status valid yang berarti kebutuhan fungsional hapus testimoni dapat diimplementasikan dan telah berjalan dengan baik pada sistem.

6.1.12 Kasus Uji Fungsional Melihat Agenda

Tabel 6.12 Kasus Uji Fungsional Lihat Agenda

Kasus Uji – Lihat Agenda	
Objek Uji	TEJA-1-13
Tujuan Pengujian	Memastikan admin dan pekerja dapat melihat daftar agenda
Prosedur Uji	Pengguna melihat detail agenda pekerjaan pada menu agenda
Hasil yang Diharapkan	Admin dan pekerja dapat melihat daftar agenda pekerjaan.
Hasil Pengujian	Admin dan pekerja dapat melihat daftar agenda pekerjaan.
Status Validasi	Valid

Analisis:

Pengguna yang memiliki hak akses sebagai pekerja dan admin memiliki kebutuhan fungsional untuk melihat agenda. Kebutuhan fungsional lihat agenda dilakukan dengan prosedur uji yang sesuai dengan *usecase* hingga memperoleh status valid. Status valid tersebut menunjukkan fungsi lihat agenda dapat berjalan sesuai kebutuhan fungsional yang telah dijabarkan dalam rekayasa kebutuhan.

6.1.13 Kasus Uji Fungsional Menambahkan Agenda Baru

Tabel 6.13 Kasus Uji Fungsional Tambah Agenda

Kasus Uji – Tambah Agenda	
Objek Uji	TEJA-1-14
Tujuan Pengujian	Memastikan admin dapat menambahkan agenda baru
Prosedur Uji	1. Pengguna memilih pilihan masukkan antrian <i>booking</i> ke agenda kerja 2. Pengguna mengisi data agenda pekerjaan berupa nama pelanggan: Irma Arini, alamat: Jl. Veteran, nomor telepon: 08123456890, dan keluhan: AC mati yang diambil dari data <i>booking</i> pelanggan 3. Pengguna menyimpan data agenda kerja
Hasil yang Diharapkan	Admin dapat menambahkan agenda baru untuk pekerja dengan memasukkan nama pelanggan, alamat, nomor telepon, dan keluhan yang diambil dari data <i>booking</i> pelanggan.
Hasil Pengujian	Admin dapat menambahkan agenda baru untuk pekerja
Status Validasi	Valid

Analisis:

Kasus uji menambahkan agenda baru diperuntukkan sebagai pengujian kebutuhan fungsional dari sisi admin. Prosedur uji yang dilakukan pun sesuai dengan *usecase* tambah agenda dalam bab rekayasa kebutuhan. Hasil yang diperoleh dalam pengujian tersebut adalah status valid yang menunjukkan bahwa fungsi menambahkan agenda baru dapat berjalan dengan baik dalam sistem yang telah diimplementasikan.

6.1.14 Kasus Uji Fungsional Mengubah Agenda

Tabel 6.14 Kasus Uji Fungsional Ubah Agenda

Kasus Uji – Ubah Agenda	
Objek Uji	TEJA-1-15
Tujuan Pengujian	Memastikan admin dapat melakukan perubahan terhadap agenda
Prosedur Uji	1. Pengguna memilih pilihan ubah agenda kerja yang telah dibuat sebelumnya 2. Pengguna melakukan perubahan data agenda pekerjaan berupa nama pelanggan: Diana Rani, alamat: Jl Gunung Guntur, nomor telepon: 081234567890, dan/atau keluhan: AC mati 3. Pengguna menyimpan perubahan data agenda kerja
Hasil yang Diharapkan	Admin dapat melakukan perubahan data pada agenda kerja yang telah dibuat sebelumnya.
Hasil Pengujian	Admin dapat melakukan perubahan data agenda kerja
Status Validasi	Valid

Analisis:

Mengubah data agenda merupakan kebutuhan fungsional yang berasal dari sisi admin. Prosedur uji fungsional mengubah data agenda telah dilakukan sesuai dengan *usecase* dalam bab rekayasa kebutuhan sehingga menghasilkan hasil yang sesuai harapan dengan status valid. Status valid tersebut menunjukkan bahwa fungsi mengubah data agenda dapat berjalan sesuai harapan.

6.1.15 Kasus Uji Fungsional Menghapus Agenda

Tabel 6.15 Kasus Uji Fungsional Hapus Agenda

Kasus Uji – Hapus Agenda	
Objek Uji	TEJA-1-16
Tujuan Pengujian	Memastikan admin dapat melakukan penghapusan agenda
Prosedur Uji	1. Pengguna memilih pilihan hapus agenda kerja yang telah dibuat sebelumnya 2. Sistem menampilkan peringatan konfirmasi penghapusan agenda kerja

	3. Pengguna melakukan konfirmasi penghapusan agenda kerja
Hasil yang Diharapkan	Admin dapat menghapus agenda kerja yang telah dibuat sebelumnya.
Hasil Pengujian	Admin dapat menghapus data agenda kerja
Status Validasi	Valid

Analisis:

Berdasarkan hasil rekayasa kebutuhan fungsional, fungsi menghapus agenda merupakan kebutuhan sistem dari sisi admin. Pengujian kebutuhan fungsional hapus agenda dilakukan dengan prosedur uji sesuai *usecase* hingga memperoleh status valid yang berarti kebutuhan fungsional hapus agenda dapat diimplementasikan dan telah berjalan dengan baik pada sistem.

6.1.16 Kasus Uji Fungsional Melihat Laporan

Tabel 6.16 Kasus Uji Fungsional Lihat Laporan

Kasus Uji – Lihat Laporan	
Objek Uji	TEJA-1-17
Tujuan Pengujian	Memastikan admin dan pekerja dapat melihat laporan pekerjaan
Prosedur Uji	Admin dan pekerja melihat laporan pekerjaan pada menu laporan
Hasil yang Diharapkan	Admin dan pekerja dapat melihat laporan pekerjaan yang telah diselesaikan.
Hasil Pengujian	Admin dan pekerja dapat melihat laporan pekerjaan.
Status Validasi	Valid

Analisis:

Pengguna yang memiliki hak akses sebagai pekerja dan admin memiliki kebutuhan fungsional untuk melihat laporan. Kebutuhan fungsional lihat laporan dilakukan dengan prosedur uji yang sesuai dengan *usecase* hingga memperoleh status valid. Status valid tersebut menunjukkan fungsi lihat laporan dapat berjalan sesuai kebutuhan fungsional yang telah dijabarkan dalam rekayasa kebutuhan.

6.1.17 Kasus Uji Fungsional Menambahkan Laporan Baru

Tabel 6.17 Kasus Uji Fungsional Tambah Laporan

Kasus Uji – Tambah Laporan	
Objek Uji	TEJA-1-18
Tujuan Pengujian	Memastikan pekerja dapat menambahkan laporan pekerjaan baru
Prosedur Uji	1. Pekerja memilih pilihan buat laporan baru 2. Pekerja mengisi data laporan pekerjaan berupa tanggal:2019-05-10, nama pelanggan: irma, alamat: Jl Veteran, uraian pekerjaan: AC mati, dan keterangan yang diambil dari data agenda

	3. Pekerja menyimpan data laporan pekerjaan
Hasil yang Diharapkan	Pekerja dapat membuat laporan baru dengan mengisi data laporan pekerjaan berupa tanggal, nama pelanggan, alamat, uraian pekerjaan, dan keterangan yang diambil dari data agenda pekerjaan.
Hasil Pengujian	Pekerja dapat membuat laporan baru
Status Validasi	Valid

Analisis:

Kasus uji menambahkan laporan baru diperuntukkan sebagai pengujian kebutuhan fungsional dari sisi pekerja. Prosedur uji yang dilakukan pun sesuai dengan *usecase* tambah laporan dalam bab rekayasa kebutuhan. Hasil yang diperoleh dalam pengujian tersebut adalah status valid yang menunjukkan bahwa fungsi menambahkan laporan baru dapat berjalan dengan baik dalam sistem yang telah diimplementasikan.

6.1.18 Kasus Uji Fungsional Mengubah Laporan

Tabel 6.18 Kasus Uji Fungsional Ubah Laporan

Kasus Uji – Ubah Laporan	
Objek Uji	TEJA-1-19
Tujuan Pengujian	Memastikan admin dan pekerja dapat melakukan perubahan terhadap laporan pekerjaan yang telah dibuat sebelumnya
Prosedur Uji	1. Pekerja memilih pilihan ubah laporan yang telah dibuat sebelumnya 2. Pekerja melakukan perubahan data laporan pekerjaan berupa tanggal: 2019-05-12 dan keterangan: ganti freon 3. Pekerja menyimpan perubahan laporan kerja
Hasil yang Diharapkan	Admin dan/atau pekerja dapat melakukan perubahan data pada laporan pekerjaan yang telah dibuat sebelumnya.
Hasil Pengujian	Admin dan pekerja dapat mengubah data laporan
Status Validasi	Valid

Analisis:

Mengubah data laporan merupakan kebutuhan fungsional yang berasal dari sisi admin. Prosedur uji fungsional mengubah data laporan telah dilakukan sesuai dengan *usecase* dalam bab rekayasa kebutuhan sehingga menghasilkan hasil yang sesuai harapan dengan status valid. Status valid tersebut menunjukkan bahwa fungsi mengubah data laporan dapat berjalan sesuai harapan.

6.1.19 Kasus Uji Fungsional Melihat Penilaian

Tabel 6.19 Kasus Uji Fungsional Lihat Penilaian

Kasus Uji – Lihat Penilaian	
Objek Uji	TEJA-1-20
Tujuan Pengujian	Memastikan admin dapat melihat penilaian pekerja
Prosedur Uji	1. Admin memilih menu penilaian 2. Admin dapat melihat seluruh data penilaian pekerja
Hasil yang Diharapkan	Admin dapat melihat data penilaian kinerja yaitu penilaian pekerja berdasarkan rating dan nilai hasil kuesioner.
Hasil Pengujian	Admin dapat melihat data penilaian kinerja pekerja
Status Validasi	Valid

Analisis:

Berdasarkan hasil rekayasa kebutuhan fungsional, fungsi melihat penilaian merupakan kebutuhan sistem dari sisi admin. Pengujian kebutuhan fungsional lihat penilaian dilakukan dengan prosedur uji sesuai *usecase* hingga memperoleh status valid yang berarti kebutuhan fungsional lihat penilaian dapat diimplementasikan dan telah berjalan dengan baik pada sistem.

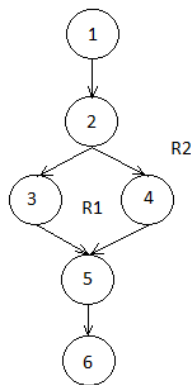
6.2 Pengujian *Whitebox*

6.2.1 Algoritme Membuat *Booking* Baru

Algoritme dalam fungsi simpan ini terletak dalam *class controller booking* yang digunakan untuk menyimpan data *booking* baru oleh pelanggan ke *database* program.

Tabel 6.20 Pengujian Algoritme Membuat Data *Booking* Baru

Algoritme 1: Fungsi Membuat Data Booking Baru	
1 FUNCTION simpan 2 BEGIN 3 SET id = date dengan format ymdHis; 4 SET data = array(5 'id_booking' = id, 6 'tanggal' = ambil masukan tanggal di form input, 7 'username' = ambil session user, 8 'pemesan' = ambil masukan nama di form input, 9 'alamat' = ambil masukan alamat di form input 10 'telp' = ambil masukan telepon di form input 11 'keluhan' = ambil masukan keluhan di form input 12); 13 SET query = DB INSERT array data ke table booking; 14 IF query = true ② 15 THEN return true; ③ 16 ELSE 17 THEN return false; ④ 18 ENDIF ⑤ 19 Pesan "Data berhasil disimpan"; 20 Tampilkan data booking ke halaman view; 21 END function ⑥	①



Nilai *cyclomatic Complexity*:

- $V(g) = E - N + 2 = 6 - 6 + 2 = 2$
- $V(g) = P + 1 = 1 + 1 = 2$
- $V(g) = R = 2$

Jalur independen:

- 1 – 2 – 3 – 5 – 6
- 1 – 2 – 4 – 5 – 6

Gambar 6.1 Flow Graph Algoritme Booking Baru

Berdasarkan *flowgraph* tersebut dapat dilihat algoritme *booking* baru memiliki nilai *cyclomatic complexity* 2 yang berarti terdapat 2 jalur independen. Jalur independen tersebut kemudian diuji dengan kasus uji yang dijabarkan pada tabel berikut:

Tabel 6.21 Kasus Uji Fungsi Membuat Data Booking Baru

No.	Kasus Uji	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
1.	- masukan data array adalah id_booking = 190510104812, tanggal = 2019-05-10, username=irma, pemesan=Irma Arini, alamat=Jl. Veteran No 1, telp=0812345678 90, keluhan=AC mati	- data dalam array berhasil disimpan ke <i>database</i> - sistem menampilkan penyimpanan data berhasil	- data dalam array berhasil disimpan ke <i>database</i> - sistem menampilkan penyimpanan data berhasil	Valid
2.	- masukan data array adalah 190510104812, tanggal = 20 april, username=irma, pemesan=Irma Arini, alamat=Jl. Veteran No 1, telp=0812345678 90, keluhan=AC mati	- penyimpanan data dalam array ke <i>database</i> gagal dilakukan - sistem menampilkan pesan error	- penyimpanan data dalam array ke <i>database</i> gagal dilakukan - sistem menampilkan pesan error	Valid

Analisis:

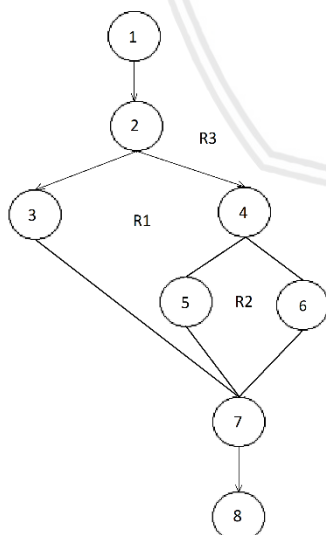
Algoritme *booking* baru merupakan rancangan algoritme berupa pseudocode yang diambil dari tahap perancangan. Rancangan algoritme yang telah diuji dengan metode *whitebox* tersebut menghasilkan nilai *cyclomatic complexity* 2. Nilai 2 tersebut berarti kompleksitas algoritme tergolong rendah sehingga mudah dalam pemeliharaan sistem kedepannya. Sementara itu, kedua jalur yang dihasilkan menjadi kasus uji dalam penerapannya untuk menguji sistem dan menghasilkan status valid. Pengujian tersebut menunjukkan masing-masing jalur dalam algoritme telah berjalan dengan benar sesuai keluaran yang diharapkan.

6.2.2 Algoritme Menghapus Data Booking

Algoritme dalam fungsi hapus ini terletak dalam *class controller booking* yang digunakan untuk menghapus data *booking* oleh pelanggan.

Tabel 6.22 Pengujian Algoritme Menghapus Data Booking

Algoritme 2: Fungsi Menghapus Data Booking	
1	FUNCTION hapus
2	BEGIN
3	SET id_booking = ambil dari view booking;
4	SET array of data_booking = DB SELECT dari table booking where
5	id_booking = id_booking;
6	SET status = data_booking->status;
7	IF status == "booked" ②
8	THEN DB DELETE dari table booking where id_booking=id_booking; ③
9	ELSEIF status == "process" ④
10	THEN DB DELETE dari table agenda where id_servis=id_booking; ⑤
11	DB DELETE dari table booking where id_booking = id_booking;
12	ELSE
13	THEN Pesan "Anda tidak dapat menghapus data booking ini" ⑥
14	ENDIF ⑦
15	END ⑧



Nilai *Cyclomatic Complexity*:

- $V(g) = E - N + 2 = 9 - 8 + 2 = 3$
- $V(g) = P + 1 = 2 + 1 = 3$
- $V(g) = R = 3$

Jalur independen:

- 1-2-3-7-8
- 1-2-4-5-7-8
- 1-2-6-7-8

Gambar 6.2 Flow Graph Algoritme Hapus Booking

Berdasarkan *flowgraph* tersebut dapat dilihat algoritme hapus *booking* memiliki nilai *cyclomatic complexity* 3 yang berarti terdapat 3 jalur independen. Jalur independen tersebut kemudian diuji dengan kasus uji yang dijabarkan pada tabel berikut:

Tabel 6.23 Kasus Uji Fungsi Menghapus Data *Booking*

No.	Kasus Uji	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
1.	- status <i>booking</i> = "booked"	- data <i>booking</i> dihapus dari <i>database</i>	- data <i>booking</i> dihapus dari <i>database</i>	Valid
2.	- status <i>booking</i> = "process"	- data <i>booking</i> sudah masuk ke agenda - data agenda dihapus terlebih dahulu - data <i>booking</i> dihapus dari <i>database</i>	- data <i>booking</i> sudah masuk ke agenda - data agenda dihapus terlebih dahulu - data <i>booking</i> dihapus dari <i>database</i>	Valid
3.	- status <i>booking</i> = "done" atau status <i>booking</i> = "reviewed" atau status <i>booking</i> diberi value lainnya	- data <i>booking</i> tidak dapat dihapus dari <i>database</i> - sistem menampilkan pesan gagal	- data <i>booking</i> tidak dapat dihapus dari <i>database</i> - sistem menampilkan pesan gagal	Valid

Analisis:

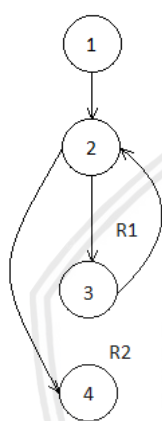
Algoritme menghapus data *booking* merupakan rancangan algoritme berupa pseudocode yang diambil dari tahap perancangan. Rancangan algoritme yang telah diuji dengan metode *whitebox* tersebut menghasilkan nilai *cyclomatic complexity* 3. Nilai 3 tersebut berarti kompleksitas algoritme tergolong rendah sehingga mudah dalam pemeliharaan sistem kedepannya. Sementara itu, ketiga jalur yang dihasilkan menjadi kasus uji dalam penerapannya untuk menguji sistem dan menghasilkan status valid. Pengujian tersebut menunjukkan masing-masing jalur dalam algoritme telah berjalan dengan benar sesuai keluaran yang diharapkan.

6.2.3 Algoritme Menghitung Rata-rata Rating

Algoritme dalam fungsi *rataRating* ini terletak dalam *class controller* penilaian yang digunakan untuk menghitung nilai rata-rata rating yang diperoleh pekerja.

Tabel 6.24 Pengujian Algoritme Menghitung Rata-rata Rating

Algoritme 3: Fungsi Menghitung Rata-rata Rating	
1	FUNCTION rataRating
2	BEGIN
3	SET avg_rating = array();
4	SET array of data_pekerja = DB SELECT data seluruh pekerja;
5	SET itr = 0;
6	FOREACH(data_pekerja) ②
7	SET pekerja = data_pekerja->username;
8	SET rating = rata-rata rating where username pekerja = pekerja;
9	avg_rating[itr] = rating;
10	itr++;
11	END
12	return avg_rating;
13	END



Nilai *cyclomatic Complexity*:

- $V(g) = E - N + 2 = 4 - 4 + 2 = 2$
- $V(g) = P + 1 = 1 + 1 = 2$
- $V(g) = R = 2$

Jalur independen:

- 1 - 2 - 4
- 1 - 2 - 3 - 4

Gambar 6.3 Flow Graph Algoritme Hitung Rata-rata Rating

Berdasarkan *flowgraph* tersebut dapat dilihat algoritme rataRating memiliki nilai *cyclomatic complexity* 2 yang berarti terdapat 2 jalur independen. Jalur independen tersebut kemudian diuji dengan kasus uji yang dijabarkan pada tabel berikut:

Tabel 6.25 Kasus Uji Fungsi Menghitung Rata-rata Rating

No.	Kasus Uji	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
1.	- array of data_pekerja = null	- sistem mengembalikan nilai avg_rating = null	- sistem mengembalikan nilai avg_rating = null	Valid
2.	- array of data_pekerja > 0 - avg_rating = 4.8	- sistem mengembalikan nilai avg_rating = 4.8	- sistem mengembalikan nilai avg_rating = 4.8	Valid

Analisis:

Algoritme menghitung rata-rata rating merupakan rancangan algoritme berupa *pseudocode* yang diambil dari tahap perancangan. Rancangan algoritme yang telah diuji dengan metode *whitebox* tersebut menghasilkan nilai *cyclomatic*

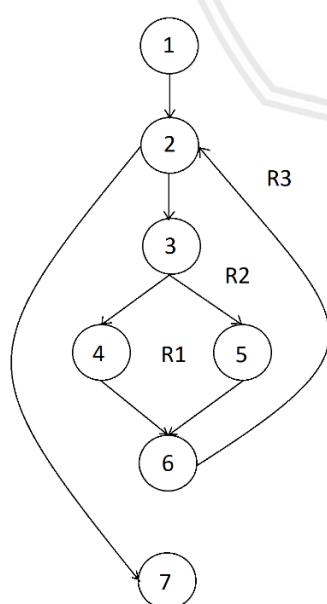
complexity 2. Nilai 2 tersebut berarti kompleksitas algoritme tergolong rendah sehingga mudah dalam pemeliharaan sistem kedepannya. Sementara itu, kedua jalur yang dihasilkan menjadi kasus uji dalam penerapannya untuk menguji sistem dan menghasilkan status valid. Pengujian tersebut menunjukkan masing-masing jalur dalam algoritme telah berjalan dengan benar sesuai keluaran yang diharapkan.

6.2.4 Algoritme Menghitung Rata-rata Nilai Kuesioner

Algoritme dalam fungsi *update* ini terletak dalam *class controller* agenda yang digunakan untuk menyimpan perubahan data agenda oleh admin ke *database* program.

Tabel 6.26 Pengujian Algoritme Menghitung Rata-rata Nilai Kuesioner

Algoritme 4: Fungsi Menghitung Rata-rata Nilai Kuesioner	
1	FUNCTION rataNilai
2	BEGIN
3	SET avg_nilai = array();
4	SET max = 25;
5	SET array of data pekerja = DB SELECT dari table pekerja where
6	level = 2;
7	SET itr = 0;
8	② FOREACH (data pekerja) THEN
9	SET pekerja = data pekerja -> username;
10	SET nilai = rata-rata nilai where username pekerja = pekerja;
11	SET countPekerja = banyaknya data penilaian pekerja where
12	username pekerja = pekerja;
13	IF (countPekerja = 0) ③
14	THEN avg_nilai[itr] = 0; ④
15	ELSE
16	THEN avg_nilai[itr] = (nilai/max)*100; ⑤
17	ENDIF ⑥
18	itr++;
19	ENDFOR ⑦
20	return avg_nilai;
21	END



Nilai cyclomatic Complexity:

- $V(g) = E - N + 2 = 8 - 7 + 2 = 3$
- $V(g) = P + 1 = 2 + 1 = 3$
- $V(g) = R = 3$

Jalur independen:

- 1-2-3-4-6-7
- 1-2-3-5-6-7
- 1-2-7

Gambar 6.4 Flow Graph Algoritme Hitung Rata-rata Nilai

Berdasarkan *flowgraph* tersebut dapat dilihat algoritme hitung rata-rata nilai memiliki nilai *cyclomatic complexity* 3 yang berarti terdapat 3 jalur independen. Jalur independen tersebut kemudian diuji dengan kasus uji yang dijabarkan pada tabel berikut:

Tabel 6.27 Kasus Uji Fungsi Menghitung Rata-rata Nilai Kuesioner

No.	Kasus Uji	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
1.	- array of data_pekerja = null	- sistem mengembalikan nilai avg_nilai = null	- sistem mengembalikan nilai avg_nilai = null	Valid
2.	- array of data_pekerja > 0 - nilai variable countPekerja = 0, nilai = 0	- sistem mengembalikan nilai avg_nilai = 0	- sistem mengembalikan nilai avg_nilai = 0	Valid
3.	- array of data_pekerja > 0 - nilai variable countPekerja = 5, nilai = 23	- sistem mengembalikan nilai avg_nilai = $(23/25)*100$, yaitu 92	- sistem mengembalikan nilai avg_nilai = $(23/25)*100$, yaitu 92	Valid

Analisis:

Algoritme menghitung rata-rata nilai kuesioner merupakan rancangan algoritme berupa *pseudocode* yang diambil dari tahap perancangan. Rancangan algoritme yang telah diuji dengan metode *whitebox* tersebut menghasilkan nilai *cyclomatic complexity* 3. Nilai 3 tersebut berarti kompleksitas algoritme tergolong rendah sehingga mudah dalam pemeliharaan sistem kedepannya. Sementara itu, ketiga jalur yang dihasilkan menjadi kasus uji dalam penerapannya untuk menguji sistem dan menghasilkan status valid. Pengujian tersebut menunjukkan masing-masing jalur dalam algoritme telah berjalan dengan benar sesuai keluaran yang diharapkan.

6.2.5 Algoritme Menampilkan Data Penilaian

Algoritme dalam fungsi index ini terletak dalam *class controller* penilaian yang digunakan untuk menampilkan data penilaian baik kepada admin maupun form penilaian untuk akses pekerja.

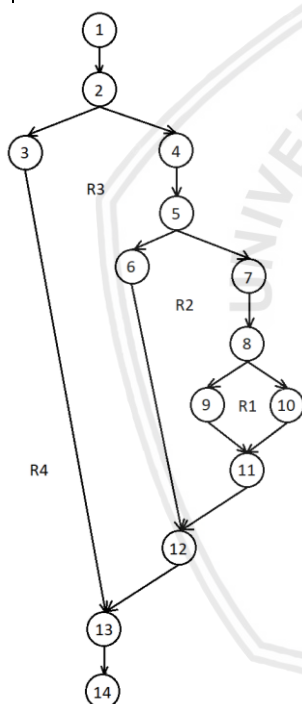
Tabel 6.28 Pengujian Algoritme Menampilkan Data Penilaian

Algoritme 5: Fungsi Menampilkan Data Penilaian	
1	FUNCTION index
2	BEGIN ①
3	IF session user masuk != true ②
4	THEN redirect to halaman login; ③
5	ELSE ④

```

6      IF session user akses = 1 //sebagai admin ⑤
7      THEN SET data = array(
8          'data_penilaian' = GET semua data penilaian,
9          'data_pekerja'   = GET semua data pekerja,
10         'data_rating'    = hitung rata-rata rating,
11         'data_nilai'     = hitung rata-rata nilai,
12     );
13     tampilkan array data dalam view admin
14 ELSEIF session user akses = 2 //sebagai pekerja ⑦
15     IF servis belum diberi nilai ⑧
16     THEN GET id_servis
17     SET array of data_penilaian = DB SELECT data penilaian
18     where id_servis = id_servis;
19     tampilkan form penilaian servis dalam view pekerja
20     ELSE
21     redirect to controller agenda;
22     pesan "Anda tidak bisa mengakses form penilaian" ⑩
23     ENDIF ⑪
24 ENDIF ⑫
25 ENDIF ⑬
26 END ⑭

```



Nilai *cyclomatic Complexity*:

$$- V(g) = E - N + 2 = 16 - 14 + 2 = 4$$

$$- V(g) = P + 1 = 3 + 1 = 4$$

$$- V(g) = R = 4$$

Jalur independen:

- 1 - 2 - 3 - 13 - 14

- 1 - 2 - 4 - 5 - 6 - 12 - 13 - 14

- 1 - 2 - 4 - 5 - 7 - 8 - 9 - 11 - 12 - 13 - 14

- 1 - 2 - 4 - 5 - 7 - 8 - 10 - 11 - 12 - 13 - 14

Gambar 6.5 Flow Graph Algoritme Tampilkan Penilaian

Berdasarkan *flowgraph* tersebut dapat dilihat algoritme tampilkan penilaian memiliki nilai *cyclomatic complexity* 4 yang berarti terdapat 4 jalur independen. Jalur independen tersebut kemudian diuji dengan kasus uji yang dijabarkan pada tabel berikut:

Tabel 6.29 Kasus Uji Fungsi Menampilkan Data Penilaian

No.	Kasus Uji	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
1.	- pengguna sedang tidak	- pengguna diarahkan ke	- pengguna diarahkan ke	Valid

	<i>login</i> , data session masuk = false	halaman <i>login</i>	halaman <i>login</i>	
2.	- data session akses pengguna = 1	- pengguna diarahkan ke halaman penilaian - yang memuat data dalam array	- pengguna diarahkan ke halaman penilaian - yang memuat data dalam array	Valid
3.	- data session akses pengguna = 2 - data id_servis = 190510111721	- pengguna diarahkan ke form pengisian penilaian sesuai id_servis	- pengguna diarahkan ke form pengisian penilaian	Valid
4.	- data session akses pengguna = 2 - data id_servis = null	- pengguna diarahkan ke <i>controller</i> agenda - sistem menampilkan pesan "Anda tidak dapat mengakses form penilaian"	- pengguna diarahkan ke <i>controller</i> agenda - sistem menampilkan pesan "Anda tidak dapat mengakses form penilaian"	Valid

Analisis:

Algoritme menampilkan data penilaian merupakan rancangan algoritme berupa *pseudocode* yang diambil dari tahap perancangan. Rancangan algoritme yang telah diuji dengan metode *whitebox* tersebut menghasilkan nilai *cyclomatic complexity* 4. Nilai 4 tersebut berarti kompleksitas algoritme tergolong rendah sehingga mudah dalam pemeliharaan sistem kedepannya. Sementara itu, keempat jalur yang dihasilkan menjadi kasus uji dalam penerapannya untuk menguji sistem dan menghasilkan status valid. Pengujian tersebut menunjukkan masing-masing jalur dalam algoritme telah berjalan dengan benar sesuai keluaran yang diharapkan.

6.3 Pengujian *Usability*

Pengujian usability merupakan pengujian yang dilakukan dengan memberikan kuesioner kepada pengguna dalam menjalankan web TeJa. Penulis telah melakukan pengujian terhadap pengguna yang terdiri dari 5 orang pekerja UKM dan 5 orang pelanggan. Berdasarkan hasil pengujian yang telah dilakukan, penulis mendapatkan total 10 responden dengan tujuan mengetahui kemampuan dari sistem manajemen TeJa yang sesuai ekspektasi pengguna. 10 orang yang telah menggunakan aplikasi ini menjawab bahwa mereka merasa senang dan mudah dalam penggunaan web TeJa. Desain yang sederhana

mempercepat pengguna dalam mengenali fitur-fitur dan mengerti penggunaannya.

Dalam pengujian usability, pernyataan kuesioner disesuaikan dengan lima aspek penting pengujian *usability* menurut Nielsen, yaitu *Learnability* (mudah dipelajari), *Efficiency* (penggunaan sumber daya), *Memorability* (mudah diingat), *Errors* (kesalahan), dan *Satisfaction* (kepuasan). Selain itu, penulis menambahkan pertanyaan terkait *effectivity* guna mengetahui apakah sistem yang dikembangkan cukup membantu dalam pelayanan jasa servis. Jawaban kuesioner pengguna kemudian dihitung menggunakan skala likert dengan 5 tingkat persetujuan dimulai dari “sangat tidak setuju” hingga “sangat setuju” hingga memperoleh hasil berupa skor kemampuan sistem informasi yang telah diuji. Adapun tingkatan persetujuan jawaban kuesioner dapat dilihat pada tabel berikut:

Tabel 6.30 Skala Likert 5 Level of Agreement

Level of Agreement	Keterangan
1	Sangat tidak setuju , menyatakan bahwa responden merasa aplikasi jauh dari ekspektasi berdasarkan pertanyaan yang diberikan.
2	Tidak setuju , menyatakan bahwa responden merasa tidak sependapat dengan pertanyaan yang diberikan.
3	Netral , menyatakan bahwa responden merasa tidak memiliki pandangan terkait pernyataan yang diberikan.
4	Setuju , menyatakan bahwa responden merasa pernyataan yang diberikan sesuai dengan pendapatnya.
5	Sangat setuju , menyatakan bahwa responden merasa sistem informasi yang dibuat memenuhi dan melebihi ekspektasi responden berdasarkan pertanyaan yang diberikan.

Tabel 6.31 Interpretasi Skor Likert

Skor Likert	Interpretasi Skor	Keterangan
1	0% - 19%	Sangat tidak memuaskan
2	20% - 39%	Tidak memuaskan
3	40% - 59%	Biasa
4	60% - 79%	Memuaskan
5	80% - 100%	Sangat memuaskan

Sementara berdasarkan lima aspek penting pengujian usabilitas, pernyataan dalam kuesioner adalah sebagai berikut:

Tabel 6.32 Daftar Pernyataan Kuesioner

No.	Aspek Usability	Pernyataan	Level of Agreement					Total Skor	Indeks (%)
			1	2	3	4	5		
1	Learnability	Menurut saya, cara penggunaan web TeJa mudah dipelajari.	0	0	1	2	7	46	92%
2	Efficiency	Menurut saya, hanya dengan melihat desain antarmuka web TeJa sebagai sistem informasi manajemen layanan jasa servis, saya langsung memahami fiturnya.	0	0	3	6	1	38	76%
3	Memorability	Menurut saya, cara menggunakan fitur web TeJa dalam layanan servis elektronik mudah diingat.	0	0	2	5	3	41	82%
4	Errors	Menurut saya, web TeJa memiliki desain antarmuka yang baik sehingga menghindari pengguna dari kesalahan maupun kekeliruan dalam menggunakan fitur.	0	0	1	7	2	41	82%
5	Satisfaction	Saya merasa web TeJa sesuai dengan ekspektasi saya dalam pelayanan jasa servis alat elektronik rumah tangga.	0	0	2	7	1	39	78%
6	Effectivity	Saya merasa web TeJa sangat membantu saya dalam menjalankan pekerjaan mulai dari	0	0	1	3	6	45	90%

		penjadwalan kerja hingga pelaporan.								
		Saya merasa web TeJa sangat membantu saya dalam melakukan pemesanan layanan servis elektronik hingga testimoni layanan.								
Rata-rata										83.3%

Analisis:

Pengujian *usability* yang telah dilakukan terhadap 5 orang pekerja dan 5 orang pelanggan menghasilkan rata-rata indeks 83.3% yang berarti sistem informasi TeJa sangat memuaskan pengguna dalam menjalankan sistem berdasarkan aspek-aspek pengujian. Hal ini juga menunjukkan bahwa sistem informasi TeJa sangat membantu pengguna dalam hal pelayanan jasa servis UKM Bali Tekindo Jaya. Selain itu, hasil pengujian *usability* menunjukkan rancangan sistem telah benar dibuat sesuai dengan ekspektasi pengguna.

6.4 Pengujian *Compatibility*

Pengujian *compatibility* web TeJa menggunakan perangkat lunak bantuan yaitu SortSite. Pengujian ini berkaitan dengan kebutuhan non fungsional sistem yaitu dapat diakses melalui berbagai web *browser*. Pengujian dilakukan dengan memasukkan alamat web TeJa ke halaman pengujian SortSite dan memilih menu Check untuk memulai pengecekan. SortSite akan memberikan keseluruhan hasil pengujian dengan hasil sebagai berikut:

Browser	IE	Edge	Firefox	Safari	Opera	Chrome	iOS			Android	
Version	11	18	66	12	60	74	≤ 10	11	12	≤ 3	4*
Critical Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Major Issues	✗	✗	✓	✗	✓	✓	✗	✓	✓	✗	✗
Minor Issues	✗		✗	✗	✓	✓		✓	✓		

Gambar 6.6 Hasil Pengujian *Compatibility*

Terdapat indikator hasil pengujian *compatibility* oleh SortSite yang dijabarkan dalam table berikut:

Tabel 6.33 Indikator Permasalahan *Compatibility*

Simbol	Keterangan
	Tidak terdapat kendala pada <i>browser</i>
	<i>Minor issues</i> atau terdapat masalah ringan pada konten atau permasalahan performa
	<i>Major issues</i> atau terdapat masalah mayor pada konten atau permasalahan performa
	<i>Critical issues</i> merupakan permasalahan yang fatal karena terdapat konten atau fungsi yang hilang pada <i>browser</i>

Analisis:

Hasil pengujian *compatibility* menggunakan perangkat lunak SortSite menunjukkan tidak terdapat *critical issues* pada seluruh web *browser* yang diuji yaitu Internet Explorer, Microsoft *Edge*, Firefox, Safari, Opera, Chrome, iOS dan Android. Hasil pengujian ini hanya menunjukkan terdapat mayor issues dan minor issues di beberapa web *browser* karena penggunaan tampilan bootstrap yang tidak didukung sebagian web *browser*. Namun masalah tersebut masih dapat dihindari jika pengguna menggunakan Opera maupun Chrome baik di perangkat PC maupun telepon pintar untuk mendapatkan tampilan web yang maksimal. Pengujian *compatibility* yang telah dilakukan menunjukkan sistem dapat berjalan dengan baik pada berbagai *browser*.

BAB 7 PENUTUP

Bagian terakhir dari penelitian ini merupakan penutup yang berisi kesimpulan hasil penelitian dalam hal ini adalah hasil pengembangan sistem manajemen UKM Bali Tekindo Jaya. Berdasarkan hasil kesimpulan akan dipaparkan juga saran untuk pengembangan sistem lebih lanjut maupun untuk pengembangan sistem yang serupa di kemudian hari.

7.1 Kesimpulan

Adapun hasil dari pengembangan sistem manajemen UKM Bali Tekindo Jaya, antara lain:

- a. Rekayasa kebutuhan fungsional sistem manajemen yang dikembangkan terdiri dari pengguna sebagai pelanggan, pekerja, dan admin yang memiliki kebutuhan fitur diantaranya pengelolaan *booking* servis, pengelolaan agenda pekerjaan, pengelolaan pelaporan pekerjaan, fitur penilaian pekerja, fitur testimoni, serta terdapat *SMS Gateway* sebagai notifikasi. Sementara rekayasa kebutuhan non fungsional sistem yaitu kompatibilitas agar sistem dapat diakses melalui berbagai *browser*. Seluruh kebutuhan fungsional tersebut telah diuji dan berjalan dengan baik dalam sistem yang telah dikembangkan.
- b. Berdasarkan kebutuhan yang diperoleh, perancangan sistem manajemen yang diberi nama TeJa tersebut digambarkan dengan diagram-diagram seperti *sequence diagram* yang menggambarkan komunikasi antar objek, *class diagram*, dan ERD sebagai diagram perancangan basis data. Diagram-diagram tersebut menjadi acuan dalam merancang algoritme berupa *pseudocode* yang telah diuji memiliki kompleksitas algoritme rendah. Adapun rancangan antarmuka sistem telah sesuai dengan ekspektasi pengguna yang ditunjukkan dengan pengujian usabilitas dengan pemberian kuesioner.
- c. Implementasi hasil rancangan sistem manajemen telah dilakukan dengan menggunakan *framework CodeIgniter* yang menerapkan konsep MVC dan bahasa pemrograman PHP. Implementasi antarmuka didesain berdasarkan rancangan antarmuka yang menggunakan *Bootstrap*. Seluruh implementasi sistem telah berjalan sesuai kebutuhan yang ditunjukkan berdasarkan pengujian validasi.
- d. Pengujian terhadap sistem dilakukan dengan beberapa metode yaitu kotak hitam (*black-box*) yang menghasilkan status valid terhadap seluruh kasus uji di setiap kebutuhan fungsional. Sementara pengujian kotak putih (*white-box*) terhadap lima algoritme menghasilkan *cyclomatic complexity* yang rendah. Adapun hasil pengujian usabilitas menunjukkan pengguna merasa puas dalam menggunakan sistem dan pengujian kompatibilitas tidak menunjukkan masalah yang berarti pada tampilan maupun pengoperasian sistem di berbagai *browser*.

7.2 Saran

Pengembangan sistem yang telah dilakukan tentunya belum sempurna. Untuk itu terdapat beberapa saran yang dapat penulis sampaikan untuk pengembangan sistem lebih lanjut maupun untuk pengembangan sistem yang serupa di kemudian hari:

- a. Dalam kaitannya dengan bidang usaha yang lebih luas lagi, diperlukan fitur untuk mengelola data pekerja dan data pelanggan agar pengguna sebagai admin dapat melakukan penyimpanan, perubahan maupun pengurangan informasi terkait pekerja.
- b. Selain pekerjaan servis, UKM Bali Tekindo Jaya juga sesekali menangani pembelian barang-barang elektronik pelanggan. Di masa depan diharapkan dapat ditambahkan fitur pemesanan dan transaksi barang elektronik maupun suku cadang sehingga dapat membantu perkembangan UKM.
- c. Sistem yang dikembangkan ini hanya menangani dan mengelola servis yang dipesan melalui web. Penulis berharap suatu saat dapat dilakukan penambahan fungsi untuk menangani pelanggan yang melakukan pemesanan servis secara *offline* agar pemesanan servis baik *offline* maupun melalui web dapat direkam sehingga data laporan servis tersinkronisasi.



DAFTAR REFERENSI

- Alfathin, A.H., Hanafi, M., & Agung, N. 2018. *Perancangan SMS Gateway sebagai Reminder Pembayaran Tagihan Layanan Internet di Muna Net Media*. Jurnal Komtika. 2(1). P 1-9. [e-journal] Tersedia melalui: <journal.ummgl.ac.id> [Diakses 15 Januari 2019]
- Andre. 2014. *Tutorial Belajar PHP Part 1: Pengertian dan Fungsi PHP dalam Pemrograman Web*, [online] Tersedia di: <<https://www.duniaikom.com/pengertian-dan-fungsi-php-dalam-pemrograman-web/>> [Diakses 10 September 2018]
- Andre. 2017. *Tutorial Belajar MySQL Part 1: Pengertian MySQL dan Kelebihan MySQL*, [online] Tersedia di: <<https://www.duniaikom.com/tutorial-mysql-alasan-menggunakan-mysql/>> [Diakses 10 September 2018]
- Booth, D. 2004. *Web Services Architecture*, [online] Tersedia di: <<https://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#whatis>> [Diakses pada 8 Januari 2019]
- Firatmadi, A. 2017. *Pengaruh Kualitas Pelayanan dan Persepsi harga Terhadap Kepuasan Pelanggan Serta Dampaknya Terhadap Loyalitas Pelanggan*. Journal of Business Studies. 2(2). P 80-105. [e-journal] Tersedia melalui: <journal.uta45jakarta.ac.id> [Diakses 15 Januari 2019]
- Hohpe, G. & Woolf, B. 2004. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Boston: Addyson-Wesley.
- Izogo, E.E. & Ogba, Ike-Elechi. 2015. *Service Quality, Customer Satisfaction and Loyalty in Automobile Repair Service Sector*. International Journal of Quality & Reliability Management. 32(3). P 250-269. [e-journal] Tersedia melalui: <www.emeraldinsight.com> [Diakses 16 januari 2019]
- Mercer, D. 2002. *HTML Introduction to Web Page Design and Development*. Singapore: The Mc-Graw-Hill Companies, Inc.
- Naista, D. 2017. *CodeIgniter Vs Laravel: Kasus Membuat Website Pencari Kerja*. Yogyakarta: LOKOMEDIA
- Nixon, R. 2012. *Learning PHP, MySQL, JavaScript, & CSS*. USA: O'Reilly Media, Inc.
- Osman, M.N., dkk. 2017. *Online Car Rental System using Web-Based and SMS Technology*. Journam Computing Research & Innovation (CRINN). 2(27). P 277-286. [e-journal] Tersedia melalui: <<https://books.google.co.id>> [Diakses 15 Februari 2019]
- Parasuraman, A., Zeithaml, V.A., & Berry, L.L. 1988. *SERVQUAL: A Multiple-Item Scale for Measuring Consumer Perceptions of Service Quality*. Journal of Retailing. 64(1). P 12-37. [e-journal] Tersedia melalui: <www.researchgate.net> [Diakses 15 Januari 2019]

- Pressman, R.S., 2010, *Software Engineering: a practitioner's approach*, 7th Edition. New York: McGraw-Hill.
- Rachmadi, F.Z., Wijaya, I.D., & Ekojono. 2016. *Rancang Bangun Sistem Informasi Indeks Kepuasan Pelanggan pada PT Kartika Sari Mulya*. Seminar Informatika Aplikatif Polinema (SIAP). [e-journal] Tersedia melalui: <www.jurnalti.polinema.ac.id> [Diakses 14 Desember 2018]
- Rahadi, D.R. 2014. *Pengukuran Usability Sistem Menggunakan Use Questionnaire pada Aplikasi Android*. Jurnal Sistem Informasi (JSI). 6(1). P 661-671. [e-journal] Tersedia melalui: <<http://ejournal.unsri.ac.id>> [Diakses 29 Mei 2019]
- Sommerville, I. 2011. *Software Engineering Ninth Edition*. Boston: Addison-Wesley.
- Sunarfrihantono, B. 2002. *PHP dan MySql untuk Web*. Yogyakarta: Andi Offset.
- Winardi, A., Farida, I., & Hariyanto, D. 2017. *Rancang Bangun Sistem Informasi Manajemen Bengkel (Studi Kasus: CV. Anugrah Bogor)*. Indonesian Journal on Software Engineering (IJSE). 3(2). P 8-14. [e-journal] Tersedia melalui: <www.ejournal.bsi.ac.id/ejurnal> [Diakses 15 Februari 2019]



LAMPIRAN A HASIL WAWANCARA

A.1 Wawancara dengan Kepala UKM Bali Tekindo Jaya

1. Bagaimana alur layanan servis UKM terhadap pelanggan selama ini?

Jawaban:

Layanan dilakukan jika terdapat pesanan perbaikan alat elektronik oleh pelanggan yang masuk ke kantor. Pesanan tersebut kemudian disampaikan kepada pekerja lalu pekerja mengerjakan perbaikan sesuai keluhan ke tempat servis yang diutarakan pelanggan. Setelah memberikan layanan, pekerja menulis laporan harian sesuai yang dikerjakan.

2. Bagaimana proses pemesanan layanan jasa servis oleh pelanggan yang berjalan selama ini?

Jawaban:

Proses pemesanan layanan yang selama ini berjalan diterima melalui telepon oleh pelanggan atau pelanggan datang langsung ke kantor UKM. Penyampaian keluhan, waktu, dan tempat servis disampaikan dalam telepon maupun saat pelanggan datang ke kantor.

3. Bagaimana pembagian tugas pekerja untuk melayani servis alat elektronik pelanggan?

Jawaban:

Tugas pekerja disampaikan setiap pagi berupa detail keluhan atau alat yang harus diperbaiki, data pelanggan seperti alamat, nama dan nomor telepon. Jika ada pesanan yang baru dilakukan pada siang hari, biasanya saya menunggu sampai pekerja bebas (telah menyelesaikan seluruh pekerjaan) dahulu baru kemudian memberikan pekerjaan sesuai pesanan baru jika masih dalam jam kerja. Jika tidak, saya akan mengagendakan pekerjaan pada esok hari.

4. Apakah Bapak selalu mengetahui waktu terselesaikannya pekerjaan oleh pekerja? Adakah mekanisme penilaian kinerja pekerja di lapangan?

Jawaban:

Tidak selalu. Kalau saya sedang berada di kantor maka saya tahu pekerjaan yang telah diselesaikan pekerja dan yang belum. Namun, saat saya tidak di kantor, saya tidak tahu apakah pekerja telah kembali ke kantor atau masih bekerja. Selama ini belum ada penilaian pekerja, hanya saja saya mengetahui kinerja pekerja dari beberapa pelanggan yang memberitahu kepuasannya atau kekecewaannya.

5. Bagaimana pelaporan servis di UKM Bali Tekindo Jaya setiap harinya?

Jawaban:

Pelaporan servis dilakukan oleh pekerja setiap sore sebelum pulang dari kantor. Masing-masing pekerja memiliki buku laporan sendiri. Setiap malam saya melakukan rekap laporan pekerja dengan menambahkan biaya servis dan membuat daftar tagihan servis.

6. Adakah kendala selama proses pelayanan servis alat elektronik tersebut?

Jawaban:

Kendala selama ini yang paling terasa tentu saja berada di waktu. Kami belum bisa memaksimalkan waktu bekerja setiap harinya dikarenakan pekerja yang memang bekerja di lapangan. Kami tidak bisa memastikan waktu terselesaikannya pekerjaan atau memadatkan pekerjaan hanya saat jam kerja sehingga seringkali pekerja terpaksa lembur. Waktu juga cukup banyak terpakai untuk merekap laporan pekerja. Selain waktu, beberapa kali terdapat kesalahpahaman antara keluhan pelanggan yang saya sampaikan kepada pekerja setiap paginya sehingga menimbulkan kekecewaan pelanggan walaupun dapat diselesaikan dengan baik setelah melalui proses klarifikasi.

Tertanda,

Bapak Kadek Armita
Kepala UKM Bali Tekindo Jaya

LAMPIRAN B PERTANYAAN KUESIONER *USABILITY*

B.1 Pertanyaan Kuesioner *Usability*

Dalam pengujian *usability* untuk memastikan sistem yang dibuat sesuai dengan ekspektasi pengguna, berikut pertanyaan dalam kuesioner yang dapat dijawab oleh pengguna:

Berikan centang (✓) pada kolom yang sesuai dengan pendapat anda.

No.	Pernyataan	Jawaban				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1	Menurut saya, cara penggunaan web TeJa mudah dipelajari.					
2	Menurut saya, hanya dengan melihat desain antarmuka web TeJa sebagai sistem informasi manajemen layanan jasa servis, saya langsung memahami fiturnya.					
3	Menurut saya, cara menggunakan fitur web TeJa dalam layanan servis elektronik mudah diingat.					
4	Menurut saya, web TeJa memiliki desain antarmuka yang baik sehingga menghindari pengguna dari kesalahan maupun kekeliruan dalam menggunakan fitur.					
5	Saya merasa web TeJa sesuai dengan ekspektasi saya dalam pelayanan jasa servis alat elektronik rumah tangga.					
6	Saya merasa web TeJa sangat membantu saya dalam menjalankan pekerjaan mulai dari penjadwalan kerja hingga pelaporan. *)					
	Saya merasa web TeJa sangat membantu saya dalam melakukan pemesanan layanan servis elektronik hingga testimoni layanan. **)					

*) Jika anda pekerja UKM Bali Tekindo Jaya

**) Jika anda pelanggan UKM Bali Tekindo Jaya

LAMPIRAN C DOKUMEN PENDUKUNG

C.1 Nota Tagihan Kerja

[illegible]

C.2 Laporan Harian Pekerja UKM

LAPORAN KERJA HARIAN TEKNISI BALI TEKINDO JAYA				
TGL.	LOKASI	JENIS PEKERJAAN	MATERIAL	KETERANGAN
20/1/2019	5. Bangsalan	1. Cek AC room 1. Lantai 2nd. AC brand ES. Listrik gantung outdoor		
	8. Warung Bini	1. Cek rumah Pak Lempur + listrik. Kipas Exhaust gantung. 1. Kipas Sanyo Bini elum		
23/1/2019	1. Bu. Wawan Pongom	1. pancing pancing AC. nyal. Sisa		
	2. Vila Astina (Sela)	1. Service Lampu 5. Living room. ganti bola lampu Led 3 watt 220V	1. 1 Bola Lampu Led pancung 3 watt 220V = 39.000.	
		2. Service Lampu 5. teras depan ganti bola lampu Led E 27 3 watt	1. 1 Bola Lampu Led 3 watt E 27 = 19.000.	
	8. Vila Astina (Jawa)	1. Service Lampu 5. teras samping Kalam, ganti bola lampu Led pancung 3 watt 220 V.	1. 1 Bola Lampu Led pancung 3 watt 220V = 39.000.	
		2. Service Lampu 5. Living room ganti bola lampu Led pancung 3 watt 220 V	1. 1 Bola Lampu Led pancung 3 watt 220V = 39.000.	
		3. Service Lampu 5. teras belakang ganti bola lampu Led 3 watt E 27	1. 1 Bola Lampu Led 3 watt E 27 = 20.000.	
	8. Vila Satria	1. Service Lampu Kalam, Kalam teling teras. 1 buah.	1. 1 Teling Lampu Kalam = 7000.	
12/1/2019	1. Bangsalan	1. Service Listrik Konstat di Bangunan sebelah barat semua listrik mati. Rencanakan pd teling Lampu Kalam rumah 2nd room 9. (kerja malam).		

C.3 Rekap Laporan oleh Kepala UKM

23/11/2018	
Date:	
Villa Leyla	320 000
Bungalow	11 200 000
Villa Bizi	886 000
Villa Puteh	60 000
Villa Ruston	525 000
Villa Lina Belle	740 000
Villa Pehar	150 000
Villa Martini	640 000
Villa Aest	120 000
Villa Simula	540 000
Villa Visula	600 000
Villa Yeklatira	200 000
Villa Sahadun	315 000
Villa Simula	120 000
Villa Binua	455 000
Villa Nala	180 000
Villa Tira	480 000
Villa Juna	440 000
	18 931 000
Teyota Dps. 10/11/2018	
CCTV Bp Band.	1900 000
An. Dps.	1290 000
Lifabul, katal Rbby dll.	1400 000
Ac foruman	1200 000
gand. Lampu Cep	1025 000
murah Ac Bm	
	6 875 000