

**PEMBANGUNAN SISTEM INFORMASI MANAJEMEN TRAVEL
BERBASIS WEB (STUDI KASUS: TRAVEL LIA JAYA SENTOSA)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
Memperoleh gelar Sarjana Komputer

Disusun oleh:

Rizky Brama Sakti

145150201111013



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2019

PENGESAHAN

PEMBANGUNAN SISTEM INFORMASI MANAJEMEN TRAVEL BERBASIS WEB
(STUDI KASUS: TRAVEL LIA JAYA SENTOSA)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Rizky Brama Sakti
NIM: 145150201111013

Skripsi ini telah diuji dan dinyatakan lulus pada
02 Mei 2019

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II


Nurudin Santoso, S.T, M.T
NIP: 197409162000121001


Mahardeka Tri A, S.Kom, M.T., M.Sc
NIK: 2016078912041001

Mengetahui
Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 197105182003121001

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, 2 Mei 2019



Rizky Brama Sakti

NIM: 145150201111013

PRAKATA

Puji syukur selalu penulis panjatkan kehadirat Allah SWT karena atas limpahan rahmat dan kasih sayang-Nya, penulisan skripsi ini dapat penulis selesaikan. Salawat serta salam semoga tetap tercurahkan kepada Nabi Muhammad SAW hingga di akhir masa.

Penulisan skripsi ini tidak dapat penulis selesaikan tepat waktu tanpa dukungan moril dari orang-orang terdekat penulis. Pada kata pengantar ini, penulis ingin menyampaikan rasa terima kasih kepada:

1. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
2. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
3. Bapak Agus Wahyu Widodo, S.T, M.Cs selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
4. Bapak Nurudin Santoso, S.T, M.T dan Bapak Mahardeka Tri Ananta, S.Kom, M.T., M.Sc selaku Dosen Pembimbing skripsi yang telah banyak membantu dan mengarahkan dalam proses penulisan skripsi ini.
5. Bapak Taridjo dan Ibu Nesti Rahayu selaku orangtua serta keluarga yang selalu memberikan dukungan moril serta materiil bagi penulis.
6. Sofyan Syahri Huzaini dan Muhammad Bima Zehansyah yang selalu ikut memberikan masukan pada saat penulis merasa kesulitan dalam pengerjaan skripsi ini beserta Muhammad Rizqian Noor dan Andrian Bagus Mahendra selaku sahabat seperjuangan dalam pengerjaan skripsi ini.
7. Dan juga teman-teman penulis yang tidak dapat penulis sebutkan satu-persatu.

Penulis menyadari bahwa dalam proses penulisan skripsi ini masih terdapat kesalahan dan kekurangan di beberapa bagian. Karena itu kritikan dan saran yang membangun sangat penulis harapkan demi perbaikan dan pengembangan skripsi ini di masa mendatang.

Malang, 3 April 2019

Penulis

rizkybrama@gmail.com

ABSTRAK

Rizky Brama Sakti, Pembangunan Sistem Informasi Manajemen Travel Berbasis Web (Studi Kasus: Travel Lia Jaya Sentosa)

Pembimbing: Nurudin Santoso, S.T., M.T dan Mahardeka Tri Ananta, S.Kom., M.T., M.Sc

Lia Jaya Sentosa merupakan perusahaan travel yang beroperasi dalam bidang penyewaan mobil untuk mengantarkan penumpangnya ke kota tertentu. Berdasarkan wawancara yang dilakukan, terdapat ketidakefisienan dalam kegiatan pencarian jadwal keberangkatan, operator harus melihat ke dalam buku besar terlebih dahulu untuk mengecek jadwal keberangkatan yang masih tersedia sehingga menyebabkan calon penumpang menunggu cukup lama hanya untuk mengetahui jadwal yang kosong. Selain itu, dalam proses pencatatan pemesanan yang terjadi, pencatatan biasanya dilakukan ke dalam buku besar, kemudian pada setiap bulannya akan direkap ke dalam file Microsoft Excel. Hal ini tentu akan membuat operator untuk bekerja dua kali karena selain mencatatnya, kemudian operator harus merekapnya menjadi sebuah laporan ke dalam file Microsoft Excel sehingga akan membuatnya menjadi kurang efektif dan efisien. Oleh karena itu, untuk mempermudah pengolahan data mulai dari pencatatan, pemesanan hingga pembuatan laporan, diperlukan adanya suatu sistem yang dapat menangani hal-hal tersebut. Pengembangan sistem ini menggunakan metode waterfall, yang prosesnya dimulai dari analisis kebutuhan sistem yang dilakukan dengan melakukan wawancara dengan operator travel Lia Jaya Sentosa. Kemudian dilakukan tahap perancangan yang terbagi menjadi perancangan arsitektur, perancangan komponen, perancangan data dan perancangan antarmuka. Kemudian tahap implementasi merealisasikan sistem yang akan dibangun sesuai dengan perancangan yang telah dibuat, keduanya dilakukan dengan pendekatan object oriented. Hasil dari pengujian unit sistem didapatkan 100% valid, pengujian integrasi 100% valid, pengujian validasi 100% valid dan pada pengujian usability adalah sistem yang dibangun dapat memudahkan pekerjaan yang ada di perusahaan dengan berdasarkan skor SUS (System Usability Scale) yang didapatkan yaitu 83,5.

Kata kunci: sistem informasi, *travel*, manajemen

ABSTRACT

Rizky Brama Sakti, Construction of a Web-based Travel Management Information System (Case Study: Lia Jaya Sentosa Travel)

Supervisors: Nurudin Santoso, S.T., M.T and Mahardeka Tri Ananta, S.Kom., M.T., M.Sc

Lia Jaya Sentosa is a travel company that operates in the field of car rental to deliver passengers to certain cities. Based on interviews conducted, there is inefficiency in the departure schedule search activities, operators must look into the ledger first to check the departure schedule that is still available, causing prospective passengers to wait long enough just to find out the empty schedule. In addition, in the process of recording orders that occur, recording is usually carried out in a ledger, then every month it will be recapitulated into Microsoft Excel files. This will certainly make the operator work twice because in addition to recording it, then the operator must recap it into a report into Microsoft Excel files so that it will make it less effective and efficient. Therefore, to facilitate data processing starting from recording, ordering to making reports, it is necessary to have a system that can handle these things. The development of this system uses the waterfall method, which the process starts from the analysis of system requirements carried out by conducting interviews with Lia Jaya Sentosa travel operators. Then the design phase is carried out which is divided into architectural design, component design, data design and interface design. Then the implementation phase realizes the system to be built in accordance with the design that has been made, both of which are carried out with an object oriented approach. The results of unit system testing are 100% valid, 100% integration testing is valid, 100% validation testing is valid and in usability testing is a system built that can facilitate existing work in the company based on the SUS (System Usability Scale) score obtained 83,5.

Keywords: *information system, travel, management*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA.....	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN	xvi
BAB 1 PENDAHULUAN	1
1.1. Latar belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan.....	2
1.4. Manfaat	2
1.5. Ruang lingkup penelitian.....	3
1.6. Sistematika penulisan.....	3
BAB 1 PENDAHULUAN	3
BAB 2 DASAR TEORI	3
BAB 3 METODOLOGI	3
BAB 4 ANALISIS KEBUTUHAN.....	3
BAB 5 PERANCANGAN DAN IMPLEMENTASI.....	3
BAB 6 PENGUJIAN	3
BAB 7 PENUTUP	3
BAB 2 DASAR TEORI	4
2.1. Kajian Pustaka.....	4
2.2. <i>Travel</i> Lia Jaya Sentosa	4
2.3. Struktur Organisasi Lia Jaya Sentosa	5
2.4. Manajemen Layanan <i>Travel</i> Lia Jaya Sentosa	5

2.4.1.	Jadwal Travel.....	6
2.4.2.	Manajemen Layanan Travel.....	6
2.5.	Laporan Harian Pemesanan Travel	7
2.6.	<i>System Development Life Cycle (SDLC)</i>	7
2.6.1.	<i>Waterfall Model</i>	7
2.7.	Pengujian Perangkat Lunak	9
2.7.1.	White Box Testing	9
2.7.2.	Pengujian Blackbox	10
2.7.3.	Pengujian <i>Usability</i>	10
2.8.	<i>CodeIgniter</i>	13
2.9.	<i>MySQL</i>	13
2.10.	Proses Bisnis.....	13
BAB 3	METODOLOGI	14
3.1	Studi pustaka	15
3.2.	Analisis kebutuhan	15
3.3.	Perancangan	16
3.4.	Implementasi.....	16
3.5.	Pengujian	17
3.6.	Kesimpulan	17
BAB 4	ANALISIS KEBUTUHAN.....	18
4.1.	Elisitasi Kebutuhan	18
4.1.1.	Gambaran Umum Sistem.....	21
4.2.	Spesifikasi Kebutuhan.....	21
4.2.1.	Identifikasi Aktor	22
4.2.2.	Daftar Kebutuhan Fungsional	22
4.2.3.	Daftar Kebutuhan Non Fungsional.....	28
4.2.4.	Use Case Diagram.....	29
4.2.5.	Use Case Scenario	31
4.3.	Validasi	59
4.3.1.	Validasi Kebutuhan Fungsional	59

4.3.2.	Validasi Kebutuhan Non Fungsional	61
4.4.	Manajemen Kebutuhan.....	62
BAB 5	PERANCANGAN DAN IMPLEMENTASI	63
5.1	Perancangan	63
5.1.1	Perancangan Arsitektur.....	63
5.1.2	Perancangan Komponen	68
5.1.3	Perancangan Data	71
5.1.4	Perancangan Antarmuka.....	72
5.2	Implementasi.....	81
5.2.1	Spesifikasi Sistem	81
5.2.2	Batasan Implementasi.....	82
5.2.3	Implementasi Basis Data	83
5.2.4	Implementasi Komponen.....	83
5.2.5	Implementasi Antarmuka	87
BAB 6	PENGUJIAN	107
6.1	Pengujian <i>Unit</i>	107
6.1.1	Pengujian Unit Melihat Beranda.....	107
6.1.2	Pengujian Unit Menghapus Pemesanan Travel	112
6.1.3	Pengujian Unit Mencetak Laporan Keuangan.....	116
6.2	Pengujian Integrasi	118
6.3	Pengujian Validasi	121
6.3.1	Kebutuhan Fungsional	121
6.3.2	Kebutuhan Non Fungsional	144
BAB 7	PENUTUP	148
7.1	Kesimpulan	148
7.2	Saran.....	149
DAFTAR PUSTAKA.....		150
LAMPIRAN		151

DAFTAR TABEL

Tabel 2.1 Jadwal Travel.....	6
Tabel 2.2 System Usability Scale.....	11
Tabel 4.1 Identifikasi Stakeholder.....	21
Tabel 4.2 Identifikasi Aktor	22
Tabel 4.3 Spesifikasi Kebutuhan Fungsional.....	22
Tabel 4.4 Spesifikasi Kebutuhan Non Fungsional	28
Tabel 4.5 Use Case Scenario Login.....	31
Tabel 4.6 Use Case Scenario Logout	32
Tabel 4.7 Use Case Scenario Melihat Beranda	32
Tabel 4.8 Use Case Scenario Mencari Grafik Pemesanan.....	33
Tabel 4.9 Use Case Scenario Melihat Grafik Pemesanan Terbanyak	33
Tabel 4.10 Use Case Scenario Melihat Grafik Rute Pemesanan Travel.....	34
Tabel 4.11 Use Case Scenario Melihat Grafik Laporan Keuangan	34
Tabel 4.12 Use Case Scenario Menambah Jadwal Travel Harian	35
Tabel 4.13 Use Case Scenario Melihat Jadwal Travel Harian	35
Tabel 4.14 Use Case Scenario Mengubah Jadwal Travel Harian	36
Tabel 4.15 Use Case Scenario Menghapus Jadwal Travel Harian.....	36
Tabel 4.16 Use Case Scenario Menambah Jadwal Travel Tambahan.....	37
Tabel 4.17 Use Case Scenario Melihat Jadwal Travel Tambahan.....	38
Tabel 4.18 Use Case Scenario Mengubah Jadwal Travel Tambahan.....	38
Tabel 4.19 Use Case Scenario Menghapus Jadwal Travel Tambahan	39
Tabel 4.20 Use Case Scenario Menambah Pemesanan travel.....	40
Tabel 4.21 Use Case Scenario Mencari Jadwal Tersedia	41
Tabel 4.22 Use Case Scenario Melihat Pemesanan Travel	41
Tabel 4.23 Use Case Scenario Mengubah Pemesanan travel.....	42
Tabel 4.24 Use Case Scenario Menghapus Pemesanan Travel.....	42
Tabel 4.25 Use Case Scenario Melihat Manajemen Pemesanan.....	43
Tabel 4.26 Use Case Scenario Mengubah Driver	44
Tabel 4.27 Use Case Scenario Mencetak Surat Jalan.....	44
Tabel 4.28 Use Case Scenario Melihat Daftar Penumpang	45
Tabel 4.29 Use Case Scenario Mencetak Tiket Travel	45
Tabel 4.30 Use Case Scenario Mengubah Status Keberangkatan	46
Tabel 4.31 Use Case Scenario Menambah Data Mobil.....	46
Tabel 4.32 Use Case Scenario Melihat Data Mobil.....	47
Tabel 4.33 Use Case Scenario Mengubah Data Mobil.....	48
Tabel 4.34 Use Case Scenario Menghapus Data Mobil	48
Tabel 4.35 Use Case Scenario Menambah Data Driver	49

Tabel 4.36 Use Case Scenario Melihat Data Driver	49
Tabel 4.37 Use Case Scenario Mengubah Data Driver	50
Tabel 4.38 Use Case Scenario Menghapus Data Driver	50
Tabel 4.39 Use Case Scenario Menambah Data Kota	51
Tabel 4.40 Use Case Scenario Melihat Data Kota	51
Tabel 4.41 Use Case Scenario Mengubah Data Kota	52
Tabel 4.42 Use Case Scenario Menghapus Data Kota	52
Tabel 4.43 Use Case Scenario Melihat Gaji Driver	53
Tabel 4.44 Use Case Scenario Mencetak Gaji Supir	54
Tabel 4.45 Use Case Scenario Mencari Gaji Supir	54
Tabel 4.46 Use Case Scenario Melihat Laporan Keuangan Pemesanan Travel	55
Tabel 4.47 Use Case Scenario Menambah Laporan Pemesanan Keuangan Travel	55
Tabel 4.48 Use Case Scenario Mengubah Laporan Keuangan Pemesanan Travel	56
Tabel 4.49 Use Case Scenario Menghapus Laporan Keuangan Pemesanan Travel	57
Tabel 4.50 Use Case Scenario Mencetak Laporan Keuangan Pemesanan Travel	58
Tabel 4.51 Use Case Scenario Mencari Laporan Keuangan Pemesanan Travel	58
Tabel 4.52 Validasi Kebutuhan Fungsional	59
Tabel 4.53 Validasi kebutuhan Non Fungsional	62
Tabel 5.1 Algoritma Melihat Beranda	68
Tabel 5.2 Algoritma Menghapus Pemesanan Travel	70
Tabel 5.3 Penjelasan Gambar Perancangan Antarmuka untuk Melihat Tabel	73
Tabel 5.4 Penjelasan gambar perancangan antarmuka untuk menambah dan merubah tabel	74
Tabel 5.5 Penjelasan gambar perancangan antarmuka untuk menambah pemesanan travel	76
Tabel 5.6 Perancangan antarmuka melihat grafik pemesanan	78
Tabel 5.7 Penjelasan gambar perancangan antarmuka print surat jalan	79
Tabel 5.8 Perancangan antarmuka print tiket travel	81
Tabel 5.9 Spesifikasi Perangkat Keras	81
Tabel 5.10 Spesifikasi Perangkat Lunak	82
Tabel 5.11 Implementasi Melihat Beranda	84
Tabel 5.12 Implementasi Menghapus Pemesanan Travel	86
Tabel 5.13 Implementasi mencetak laporan keuangan	87
Tabel 6.1 Algoritme Melihat Grafik Beranda	107
Tabel 6.2 Test Case Melihat Grafik Pemesanan	111
Tabel 6.3 Algoritme Menghapus Pemesanan Travel	112
Tabel 6.4 Test Case Menghapus Pemesanan Travel	115
Tabel 6.5 Algoritme Mencetak Laporan Keuangan	116
Tabel 6.6 Test Case Mencetak Laporan	118

Tabel 6.7 Identifikasi dan Rencana Pengujian Integrasi	118
Tabel 6.8 Hasil Pengujian Integrasi Nomor 1.....	119
Tabel 6.9 Hasil Pengujian Integrasi Nomor 2.....	120
Tabel 6.10 Hasil Pengujian Integrasi Nomor 3.....	120
Tabel 6.11 Pengujian Validasi Login.....	121
Tabel 6.12 Pengujian Validasi Logout	122
Tabel 6.13 Pengujian Validasi Melihat Beranda	122
Tabel 6.14 Pengujian Validasi Mencari Grafik Pemesanan.....	123
Tabel 6.15 Pengujian Validasi Melihat Grafik Pemesanan Terbanyak	123
Tabel 6.16 Pengujian Validasi Melihat Grafik Rute Pemesanan Travel.....	124
Tabel 6.17 Pengujian Validasi Melihat Grafik Laporan Keuangan	124
Tabel 6.18 Pengujian Validasi Menambah Jadwal Travel Harian	124
Tabel 6.19 Pengujian Validasi Melihat Jadwal Travel Harian	125
Tabel 6.20 Pengujian Validasi Mengubah Jadwal Travel Harian	126
Tabel 6.21 Pengujian Validasi Menghapus Jadwal Travel Harian.....	126
Tabel 6.22 Pengujian Validasi Menambah Jadwal Travel Tambahan.....	127
Tabel 6.23 Pengujian Validasi Melihat Jadwal Travel Tambahan.....	127
Tabel 6.24 Pengujian Validasi Mengubah Jadwal Travel Tambahan.....	128
Tabel 6.25 Pengujian Validasi Menghapus Jadwal Travel Tambahan	128
Tabel 6.26 Pengujian Validasi Menambah Pemesanan Travel	129
Tabel 6.27 Pengujian Validasi Mencari Jadwal Tersedia	129
Tabel 6.28 Pengujian Validasi Melihat Pemesanan Travel	130
Tabel 6.29 Pengujian Validasi Mengubah Pemesanan Travel	130
Tabel 6.30 Pengujian Validasi Menghapus Pemesanan Travel.....	131
Tabel 6.31 Pengujian Validasi Melihat Manajemen Pemesanan	131
Tabel 6.32 Pengujian Validasi Mengubah Driver	132
Tabel 6.33 Pengujian Validasi Mencetak Surat Jalan	132
Tabel 6.34 Pengujian Validasi Melihat Daftar Penumpang	133
Tabel 6.35 Pengujian Validasi Mencetak Tiket	133
Tabel 6.36 Pengujian Validasi Menandai Transaksi Selesai.....	134
Tabel 6.37 Pengujian Validasi Menambah Data Mobil.....	134
Tabel 6.38 Pengujian Validasi Melihat Data Mobil.....	135
Tabel 6.39 Pengujian Validasi Mengubah Data Mobil.....	135
Tabel 6.40 Pengujian Validasi Menghapus Data Mobil	136
Tabel 6.41 Pengujian Validasi Menambah Data Driver	136
Tabel 6.42 Pengujian Validasi Melihat Data Driver	137
Tabel 6.43 Pengujian Validasi Mengubah Data Driver	137
Tabel 6.44 Pengujian Validasi Menghapus Data Driver	138
Tabel 6.45 Pengujian Validasi Menambah Data Kota.....	138

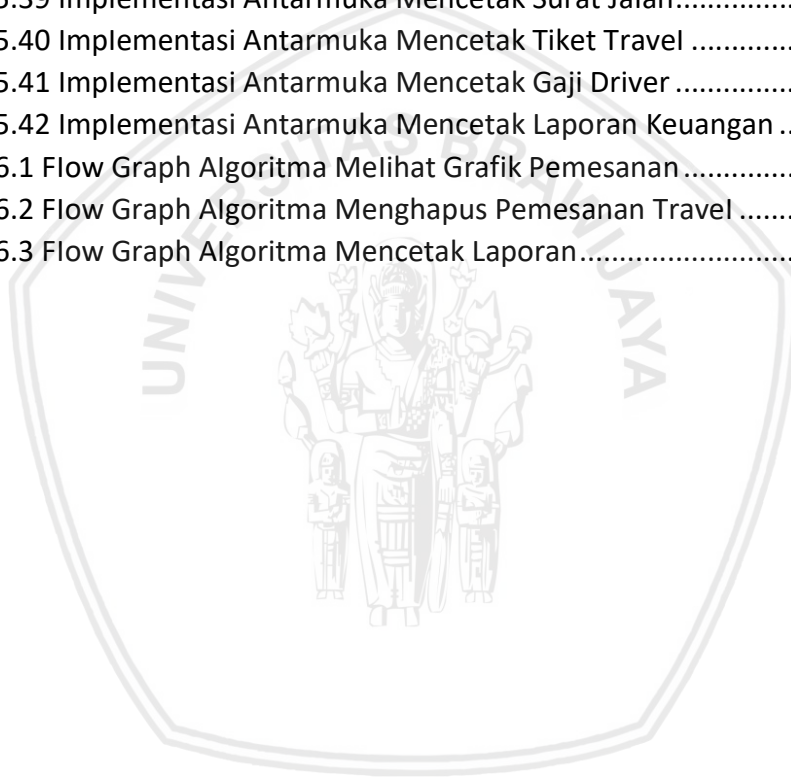
Tabel 6.46 Pengujian Validasi Melihat Data Kota.....	139
Tabel 6.47 Pengujian Validasi Mengubah Data Kota.....	139
Tabel 6.48 Pengujian Validasi Menghapus Data Kota	140
Tabel 6.49 Pengujian Validasi Melihat Gaji Driver.....	140
Tabel 6.50 Pengujian Validasi Mencari Gaji Driver.....	140
Tabel 6.51 Pengujian Validasi Mencetak Gaji Driver	141
Tabel 6.52 Pengujian Validasi Menambah Laporan Keuangan.....	142
Tabel 6.53 Pengujian Validasi Melihat Laporan Keuangan.....	142
Tabel 6.54 Pengujian Validasi Mengubah Laporan Keuangan.....	143
Tabel 6.55 Pengujian Validasi Menghapus Laporan Keuangan	143
Tabel 6.56 Pengujian Validasi Mencetak Laporan Keuangan	144
Tabel 6.57 Pengujian Validasi Mencari Laporan keuangan	144
Tabel 6.58 Data Responden	145
Tabel 6.59 Task.....	145
Tabel 6.60 Hasil Pengisian Kuesioner	146
Tabel 6.61 Pengolahan Skor Akhir SUS.....	146



DAFTAR GAMBAR

Gambar 2.1 Struktur Organisasi Lia Jaya Sentosa.....	5
Gambar 2.2 Siklus Waterfall Model (Sumber: Sommerville, 2016)	8
Gambar 2.3 Rating dan Skala Konversi Skor SUS (Ardiansyah & Ghazali, 2016)	12
Gambar 3.1 Diagram Alir Metode Penelitian	14
Gambar 4.1 Proses Bisnis As-Is	19
Gambar 4.2 Proses Bisnis To-Be.....	20
Gambar 4.3 Use Case Diagram.....	30
Gambar 5.1 Sequence Diagram Melihat Beranda	65
Gambar 5.2 Sequence Diagram Menghapus Pemesanan Travel.....	65
Gambar 5.3 Sequence Diagram Mencetak Laporan Keuangan	66
Gambar 5.4 Class Diagram	67
Gambar 5.5 Conceptual Data Model	72
Gambar 5.6 Perancangan Antarmuka untuk Melihat Tabel	73
Gambar 5.7 Perancangan antarmuka untuk menambah dan merubah tabel	74
Gambar 5.8 Perancangan antarmuka untuk menambah pemesanan travel	76
Gambar 5.9 Perancangan antarmuka melihat grafik pemesanan	78
Gambar 5.10 Perancangan antarmuka print surat jalan	79
Gambar 5.11 Perancangan antarmuka print tiket travel.....	80
Gambar 5.12 Physical Data Model Sistem informasi Manajemen Travel Lia Jaya Sentosa.....	83
Gambar 5.13 Implementasi Melihat Grafik Laporan	88
Gambar 5.14 Implementasi Menambah Jadwal Travel Harian	89
Gambar 5.15 Implementasi Melihat Jadwal Travel Tambahan.....	89
Gambar 5.16 Implementasi Antarmuka Melihat Pemesanan Travel	90
Gambar 5.17 Implementasi Antarmuka Melihat Data Mobil.....	90
Gambar 5.18 Implementasi Antarmuka Melihat Data Driver	91
Gambar 5.19 Implementasi Antarmuka Melihat Data Kota	91
Gambar 5.20 Implementasi Antarmuka Melihat Manajemen Pemesanan.....	92
Gambar 5.21 Implementasi Antarmuka Melihat Daftar Penumpang	92
Gambar 5.22 Implementasi Antarmuka Laporan Keuangan	93
Gambar 5.23 Implementasi Gaji Driver	93
Gambar 5.24 Implementasi Antarmuka Menambah Jadwal Travel Harian	94
Gambar 5.25 Implementasi Antarmuka Mengubah Jadwal Travel Harian	95
Gambar 5.26 Implementasi Antarmuka Menambah Jadwal Travel Tambahan Bagian 1.....	96
Gambar 5.27 Implementasi Antarmuka Menambah Jadwal Travel Tambahan Bagian 2.....	96

Gambar 5.28 Implementasi Antarmuka Mengubah Jadwal Travel Tambahan	97
Gambar 5.29 Implementasi Antarmuka Menambah Pemesanan Travel	98
Gambar 5.30 Implementasi Antarmuka Mengubah Pemesanan Travel	99
Gambar 5.31 Implementasi Antarmuka Menambah Data Mobil	100
Gambar 5.32 Implementasi Antarmuka Mengubah Data Mobil	100
Gambar 5.33 Implementasi Antarmuka Menambah Data Driver	101
Gambar 5.34 Implementasi Antarmuka Mengubah Data Driver	101
Gambar 5.35 Implementasi Antarmuka Menambah Data Kota	102
Gambar 5.36 Implementasi Antarmuka Mengubah Data Kota	102
Gambar 5.37 Implementasi Antarmuka Menambah Laporan Keuangan.....	103
Gambar 5.38 Implementasi Antarmuka Mengubah Laporan Keuangan.....	104
Gambar 5.39 Implementasi Antarmuka Mencetak Surat Jalan.....	105
Gambar 5.40 Implementasi Antarmuka Mencetak Tiket Travel	105
Gambar 5.41 Implementasi Antarmuka Mencetak Gaji Driver	106
Gambar 5.42 Implementasi Antarmuka Mencetak Laporan Keuangan	106
Gambar 6.1 Flow Graph Algoritma Melihat Grafik Pemesanan	110
Gambar 6.2 Flow Graph Algoritma Menghapus Pemesanan Travel	114
Gambar 6.3 Flow Graph Algoritma Mencetak Laporan.....	117



DAFTAR LAMPIRAN

Lampiran 1 Kuesioner Responden 1	151
Lampiran 2 Kuesioner Responden 2	152
Lampiran 3 Kuesioner Responden 3	153
Lampiran 4 Kuesioner Responden 4	154
Lampiran 5 Kuesioner Responden 5	155



BAB 1 PENDAHULUAN

1.1. Latar belakang

Beberapa tahun terakhir, teknologi terus berkembang dengan sangat cepat dan semakin canggih. Hal ini tentunya memberikan dampak kepada kehidupan masyarakat di dalam berbagai bidang. Salah satu contoh dampak dari perkembangan teknologi adalah informasi yang kini lebih mudah didapatkan. Informasi dapat tersebar dengan cepat melalui sebuah teknologi jaringan komputer yang bernama *web*. Penggunaan *web* sebagai sistem informasi dapat membantu sebuah perusahaan untuk lebih optimal dalam mengelola data perusahaan, maka dari itu sistem informasi berbasis *web* adalah salah satu sarana yang efektif untuk menjembatani kebutuhan perusahaan dalam menangani proses bisnis yang ada.

Berdasarkan wawancara yang telah dilakukan, Lia Jaya Sentosa merupakan perusahaan *travel* yang sudah berdiri sejak tahun 2005. Perusahaan ini beroperasi dalam bidang penyewaan transportasi, untuk mengantarkan pelanggannya ke kota-kota tertentu dengan menggunakan kendaraan mobil. Dalam pelayanannya, terdapat ketidakefisienan dalam kegiatan pencarian jadwal keberangkatan, operator harus melihat ke dalam buku besar pemesanan terlebih dahulu untuk mengecek jadwal keberangkatan yang masih tersedia sehingga menyebabkan calon penumpang menunggu cukup lama hanya untuk mengetahui jadwal yang kosong. Selain itu, dalam proses pencatatan pemesanan yang terjadi, pencatatan biasanya dilakukan ke dalam buku besar kemudian pada setiap bulannya akan direkap ke dalam file *Microsoft Excel*. Hal ini tentu akan membuat operator untuk bekerja dua kali karena selain mencatatnya, kemudian operator harus merekapnya menjadi sebuah laporan ke dalam file *Microsoft Excel* sehingga akan membuatnya menjadi kurang efektif dan efisien. Oleh karena itu, untuk mempermudah pengolahan data mulai dari pencatatan, pemesanan hingga pembuatan laporan, diperlukan adanya suatu sistem yang dapat menangani hal-hal tersebut.

Dalam perkembangan Teknologi Informasi, permasalahan ini bukan pertama kali dan sering ditemui pada beberapa perusahaan. Seperti penelitian yang sudah dilakukan oleh Zubaidi (2015) dengan judul Sistem Informasi Biro Perjalanan *Tour* Berbasis *Website* dengan Menggunakan *Framework Codeigniter* pada Anggada Indonesia *Tour & Travel*. Menyimpulkan bahwa sistem informasi ini dapat memudahkan operator *travel* dan memperlancar transaksi pemesanan *tour* dan *travel*. Dan penelitian yang telah dilakukan oleh Isnandi (2014) dengan judul Sistem Informasi Penjualan Tiket pada Al Fath *Tours dan Travel* Pacitan. Menyimpulkan bahwa dengan adanya sistem informasi penjualan tiket *travel* akan mempermudah dan mempercepat kinerja pada Al Fath *Tours & Travel* Pacitan seperti pembuatan laporan penjualan dalam setiap periode tertentu.

Berdasarkan latar belakang dan penelitian yang sudah dipaparkan, maka akan dilakukan penelitian dengan judul “Pembangunan Sistem Informasi Manajemen *Travel* Berbasis *Web* (Studi Kasus: *Travel* Lia Jaya Sentosa)”. Penelitian ini dikerjakan dengan menggunakan metode pengembangan *waterfall* karena metode ini cocok digunakan untuk sistem yang memiliki kebutuhan yang stabil dan sudah diketahui lengkap di awal dan kebutuhan tidak berubah secara radikal selama pengembangan sistem (Pressman & Maxim, 2015). Kemudian dengan adanya sistem informasi ini, diharapkan mampu memudahkan Lia Jaya Sentosa untuk mengelola proses bisnis yang terjadi di dalam perusahaan.

1.2. Rumusan Masalah

1. Bagaimana proses dan hasil analisis kebutuhan untuk sistem informasi berbasis web yang mampu memudahkan manajemen pada perusahaan Lia Jaya Sentosa?
2. Bagaimana proses dan hasil perancangan berdasarkan analisis yang telah dilakukan untuk pembangunan sistem informasi manajemen travel berbasis *web*?
3. Bagaimana proses dan hasil pembangunan sistem informasi manajemen travel berbasis web berdasarkan perancangan yang sudah dibuat?
4. Bagaimana proses dan hasil pengujian untuk implementasi sistem yang sudah dilakukan dengan pengujian *White Box* dan *Black Box*?

1.3. Tujuan

Tujuan umum yang dapat diperoleh dari penelitian ini adalah membangun sebuah aplikasi sistem informasi manajemen travel. Adapun tujuan khusus yang diperoleh dari penelitian ini adalah:

1. Mendapatkan kebutuhan sistem yang diperlukan dalam pembuatan sistem informasi manajemen travel berbasis web pada Perusahaan Lia Jaya Sentosa.
2. Merancang model sistem informasi manajemen travel sesuai dengan analisis kebutuhan.
3. Membangun sistem informasi manajemen travel berbasis web sesuai dengan perancangan yang telah dibuat.
4. Untuk mengetahui apakah sistem mudah digunakan dan berjalan dengan baik sesuai dengan perancangan yang telah dibuat.

1.4. Manfaat

1. Memberikan kemudahan kepada perusahaan dalam mengelola data perusahaan agar lebih efisien.
2. Memberikan kemudahan kepada perusahaan dalam menyusun laporan-laporan yang diperlukan perusahaan.

1.5. Ruang lingkup penelitian

1. Obyek penelitian pada *Travel Lia Jaya Sentosa*.
2. Aplikasi dibuat untuk bagian operator tiket saja.
3. Aplikasi ini berbasis *web*, dibangun dengan menggunakan framework *Codeigniter*, *Jquery* dan *Bootstrap* serta menggunakan *DBMS (Database Management System) mySQL*.

1.6. Sistematika penulisan

BAB 1 PENDAHULUAN

Menjelaskan tentang latar belakang, rumusan permasalahan, tujuan penelitian, manfaat penelitian, ruang lingkup penelitian, sistematika penulisan, dan jadwal pelaksanaan

BAB 2 DASAR TEORI

Menjelaskan tentang teori-teori pendukung yang berkaitan dengan penelitian yang sedang dikerjakan.

BAB 3 METODOLOGI

Bab ini berisi tentang metode dan langkah-langkah yang dilakukan dalam pembuatan sistem informasi pemesanan travel berbasis *web* pada *Travel Lia Jaya Sentosa*.

BAB 4 ANALISIS KEBUTUHAN

Bab ini berisi tentang analisis kebutuhan dalam pembuatan sistem informasi pemesanan travel berbasis *web* pada *Travel Lia Jaya Sentosa*.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Berisi perancangan sistem informasi pemesanan travel berbasis *web* pada *Travel Lia Jaya Sentosa* dan kemudian dari perancangan yang sudah dibuat akan diimplementasikan ke dalam sistem yang akan dibangun.

BAB 6 PENGUJIAN

Bab ini berisi pengujian pada sistem untuk menguji kelayakan sistem.

BAB 7 PENUTUP

Berisi kesimpulan dan saran-saran yang dapat digunakan sebagai acuan dalam pengembangan perangkat lunak lebih lanjut.

BAB 2 DASAR TEORI

2.1. Kajian Pustaka

Kajian pustaka yang digunakan dalam penelitian ini membahas tentang penelitian sebelumnya mengenai sistem informasi travel dalam memudahkan perusahaan untuk melakukan kegiatan bisnisnya.

Pada penelitian sebelumnya yang dilakukan oleh (Zubaidi, 2015) dengan judul “Sistem Informasi Biro Perjalanan *Tour* Berbasis *Website* dengan Menggunakan Framework Codeigniter pada Anggada Indonesia *Tour* dan *Travel*” terdapat fitur-fitur seperti pengelolaan data bus, pengelolaan data karyawan, pengelolaan data pelanggan, pengelolaan data jadwal, pengelolaan data paket tour, pengelolaan data travel, pemesanan travel, pemesanan tour dan pelaporan. Kesimpulan yang diperoleh dari penelitian ini adalah dengan menggunakan sistem ini seluruh data-data administrasi pada Anggada Indonesia Tour & Travel telah tersimpan secara terpusat dalam sebuah database, sistem informasi ini mempermudah petugas dalam mengelola administrasi travel dan memperlancar transaksi pemesanan tour dan travel.

Selain itu, pada penelitian sebelumnya yang dilakukan oleh (Isnandi, 2014) dengan judul “Sistem Informasi Penjualan Tiket Pada Al Fath Tours dan Travel Pacitan” terdapat fitur-fitur seperti pengelolaan data mobil, pengelolaan data supir, pengelolaan data penumpang dan data tiket, form data penumpang, form laporan penjualan perhari, perbulan dan pertahun. Kesimpulan yang diperoleh dari penelitian ini adalah dengan menggunakan sistem ini akan mempermudah dan mempercepat kinerja pada Al Fath Tours & Travel Pacitan.

Beberapa hal yang membedakan penelitian yang akan dibuat dengan kedua penelitian di atas adalah pada laporan travel, tidak ada laporan yang menunjukkan trend penjualan tiket yang berfungsi agar perusahaan dapat lebih bijak dalam pengalokasian mobil atau paket wisata yang ada nantinya, belum adanya juga fitur untuk menampilkan penggajian driver karena pada perusahaan Lia Jaya Sentosa driver mendapatkan gaji 15% pada setiap perjalanan yang dilakukan, fitur ini dapat diberikan sebagai rekapan pendapatan yang didapatkan oleh driver dan perusahaan pada setiap perjalanan travel.

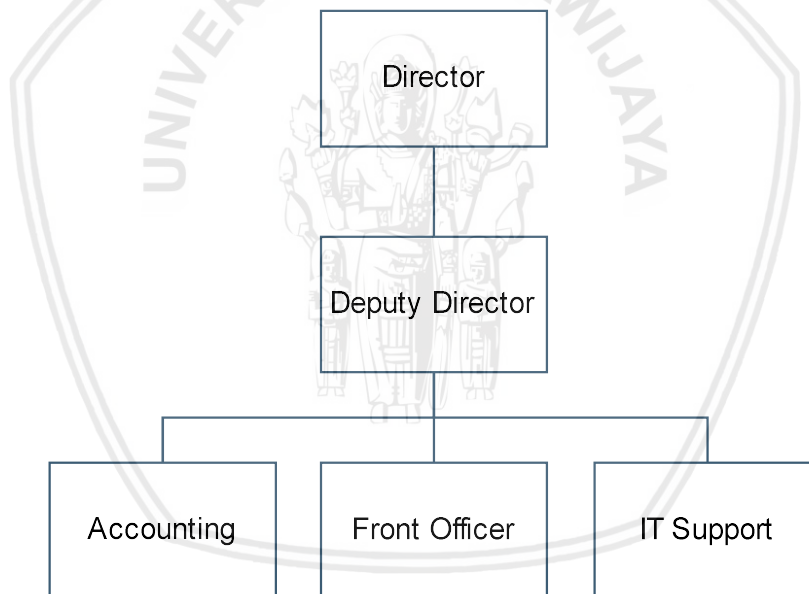
2.2. *Travel* Lia Jaya Sentosa

Travel Lia Jaya Sentosa adalah sebuah perusahaan travel yang bergerak dalam bidang transportasi. Travel ini berlokasi di Jalan Trunojoyo No 46 Malang. Travel Lia Jaya Sentosa sudah berdiri sejak tahun 2005 dan perusahaan ini masih tetap dapat menjalankan usahanya hingga saat ini. Usaha jasa ini berfokus pada layanan pengantaran penumpang dengan menggunakan kendaraan mobil. Motto yang

dibawa oleh perusahaan ini adalah sabar, sopan dan senyum yang memiliki makna setiap complain atau masalah yang ada akan dihadapi dengan sabar dan dalam pelayanannya operator yang bertugas akan mengutamakan kesopanan dan senyum pada setiap konsumen yang dilayani.

2.3. Struktur Organisasi Lia Jaya Sentosa

Pada Gambar 2.1, dapat dilihat struktur organisasi yang ada di perusahaan Lia Jaya Sentosa. Terdiri dari direktur dan wakil direktur, kemudian dibawahnya dapat dilihat bahwa wakil direktur membawahi 3 bagian yaitu bagian accounting, front officer dan IT Support. Pada bagian accounting, tugas dari seorang accounting pada perusahaan ini adalah menyusun laporan bulanan yang berasal dari laporan harian pada bagian front officer. Pada bagian front officer atau lebih sering disebut operator bertugas untuk melayani pemesanan layanan yang ada pada Lia Jaya Sentosa dan sebagai penerima pesan atau telepon yang masuk ke nomer kantor Lia Jaya Sentosa. Sedangkan untuk IT Support, adalah seorang front officer yang memiliki pengetahuan IT yang cukup untuk membantu jika terdapat masalah-masalah seputar IT yang terjadi di dalam perusahaan Lia Jaya Sentosa.



Gambar 2.1 Struktur Organisasi Lia Jaya Sentosa

2.4. Manajemen Layanan Travel Lia Jaya Sentosa

Dalam kesehariannya, untuk mendapatkan hasil yang optimal, Travel Lia Jaya tentu perlu mengatur segala sesuatu yang berhubungan dengan pelayanannya. Hal ini pun berlaku pada setiap layanan-layanan yang ada di Travel Lia Jaya Sentosa seperti layanan pengantaran penumpang.

2.4.1. Jadwal Travel

Jadwal pengantaran travel pada Travel Lia Jaya Sentosa dapat dilihat pada Tabel 2.1

Tabel 2.1 Jadwal Travel

Tujuan	Jam	Harga
Malang-Bunder Malang-Duduk Samean Malang-Lamongan Malang-Babat Malang-Pucuk	Jam 5.00 dan 15.00	110000
Malang-Baureno Malang-Sumberejo Malang-Bojonegoro Malang-Kapas	Jam 5.00 dan 15.00	110000
Malang-Gresik Malang-Tuban Malang-Paciran	Jam 5.00 dan 15.00	110000
Malang-Merakurak	Jam 5.00 dan 15.00	125000
Malang-Kerek Malang-Montong	Jam 5.00 dan 15.00	140000
Malang-Kalitidu Malang-Purwosari Malang-Padangan Malang-Cepu	Jam 5.00 dan 15.00	135000
Malang-Cabak Malang-Jiken Malang-Jepon Malang-Blora	Jam 5.00 dan 15.00	160000
Malang-Ngawen Malang-Kundur	Jam 5.00 dan 15.00	175000
Malang-Wirosari Malang-Purwodadi	Jam 5.00 dan 15.00	195000
Juanda	Jam Ganjil antara 5.00- 15.00	80000

2.4.2. Manajemen Layanan Travel

Untuk layanan pengantaran penumpang, agar tidak terjadi crash antara mobil satu dengan yang lainnya, tentunya perlu ada sebuah manajemen yang mengatur pemberangkatan mobil untuk pengantaran penumpang. Travel Lia Jaya memiliki

jumlah armada sebanyak 60 mobil. Dapat dilihat sesuai jadwal yang ada pada tabel 2.1, mobil berangkat untuk menjemput penumpang pada jam 5 pagi dan 3 sore. Maka dari itu, manajemen mobil yang dilakukan oleh pihak Travel Lia Jaya Sentosa adalah pada setiap tujuan tertentu dan jam tertentu terdapat 1 mobil yang dialokasikan ke dalamnya. Jadi misalnya, pada tujuan Malang-Bunder pada jam 5 pagi, terdapat 1 mobil yang dialokasikan, begitu juga dengan tujuan yang sama pada jam 3 sore, terdapat 1 mobil lagi yang dialokasikan ke dalamnya dan hal inipun juga berlaku pada setiap tujuan tertentu dan jam tertentu.

2.5. Laporan Harian Pemesanan Travel

Untuk setiap harinya, pihak Lia Jaya Sentosa merekap data pada setiap pemesanan yang terjadi, mencatatnya ke dalam buku besar dan memindahkannya ke dalam sebuah file excel setiap bulannya.

2.6. System Development Life Cycle (SDLC)

Dalam pengembangan perangkat lunak diperlukan sebuah *System Development Life Cycle (SDLC)* yang digunakan untuk menjadi acuan dalam pengembangan perangkat lunak. *SDLC* ini, nantinya akan mendefinisikan aliran tugas atau aksi apa saja yang harus dilakukan untuk membuat sebuah pengembangan perangkat lunak dalam sebuah organisasi agar berhasil dilakukan (Pressman & Maxim, 2015). Hal ini penting untuk dilakukan karena sebuah *SDLC* menyediakan stabilitas dan kontrol pada sebuah organisasi yang sedang melakukan pengembangan perangkat lunak agar sebuah pengembangan perangkat lunak ketika sudah selesai sampai ke tahap *deployment*, tidak ditinggalkan begitu saja, tetapi terus diperbaiki jika ada masalah dan lebih baik lagi jika ada pengembangan lebih lanjut.

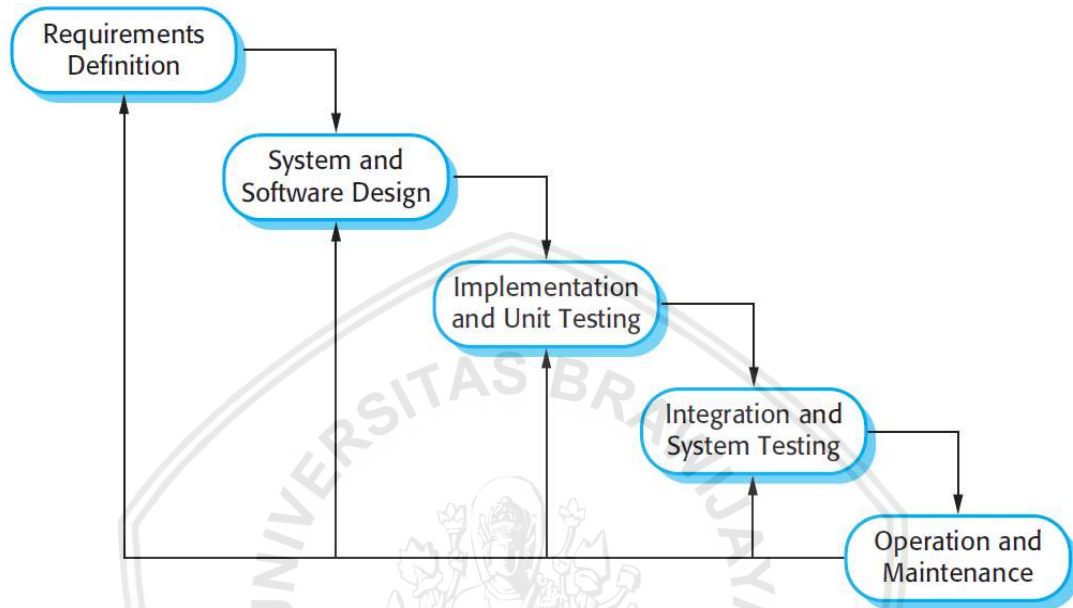
Ada beberapa *SDLC* yang dapat diadaptasi dan diikuti oleh pengembang perangkat lunak. Tentunya *SDLC* ini tidak asal-asalan dimasukkan dalam pengembangan perangkat lunak. Ada beberapa hal yang harus dipertimbangkan untuk memilih *SDLC* mana yang cocok untuk sebuah pengembangan perangkat lunak.

2.6.1. Waterfall Model

Waterfall Model, terkadang disebut sebagai *Classic Life Cycle*, yaitu sebuah *SDLC* yang mengusulkan sebuah pendekatan kepada pengembangan perangkat lunak yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan. Ada saatnya sebuah kebutuhan dari sebuah masalah telah diketahui lengkap di awal ketika pada saat fase awal yaitu fase *communication* (Pressman & Maxim, 2015).

Dalam prinsipnya, hasil dari setiap fase adalah satu atau lebih dokumen yang diberikan tanda *signed off*. Fase berikutnya tidak akan berjalan sebelum fase berikutnya selesai dikerjakan. Dalam praktiknya, setiap fase saling melengkapi dan

saling memberi informasi satu sama lain. Contohnya selama fase desain, masalah yang ada pada fase *requirement definition* akan digunakan untuk membuat desain model yang sesuai. Pada fase *coding*, desain yang telah dibuat pada fase desain akan diterapkan pada fase *coding* dan begitu seterusnya (Sommerville, 2011). Aktivitas-aktivitas yang dilakukan pada *Waterfall Model* adalah sebagai berikut tergambar di Gambar 2.2.



Gambar 2.2 Siklus *Waterfall Model* (Sumber: Sommerville, 2016)

Keterangan :

a. Requirement analysis and definition

Layanan sistem, batasan dan tujuan harus didapatkan melalui proses konsultasi dengan *user*. Setelah melalui proses tersebut, kemudian kebutuhan dianalisis dan harus dapat didefinisikan dengan lengkap agar dapat menghasilkan spesifikasi sistem yang jelas (Sommerville, 2011).

b. System and software design

Desain dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap dengan membuat diagram-diagram yang menggambarkan model dari sistem perangkat lunak (Sommerville, 2011).

c. Implementation and unit testing

Desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibangun langsung diuji secara unit (Sommerville, 2011).

d. *Integration and system testing*

Penyatuan unit-unit program kemudian diuji secara keseluruhan (*system testing*) (Sommerville, 2011).

e. *Maintenance*

Mengoperasikan program dilingkungannya dan melakukan pemeliharaan, seperti penyesuaian atau perubahan setelah beradaptasi dengan situasi sebenarnya (Sommerville, 2011).

Alasan menggunakan *Waterfall Model* adalah:

1. Kualitas dari sistem yang dihasilkan akan baik. Ini dikarenakan oleh pelaksanaannya secara bertahap. Sehingga tidak terfokus pada tahapan tertentu.
2. Dokumen pengembangan sistem sangat terorganisir, karena setiap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya.

2.7. Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah sebuah metode yang digunakan untuk menunjukkan bahwa suatu program sudah melakukan apa yang diinginkan atau dibutuhkan dan pengujian dilakukan untuk menemukan cacat program sebelum sebuah program digunakan oleh user (Sommerville, 2011). Pengujian perangkat lunak ditujukan untuk mengukur kualitas dari perangkat lunak yang diuji. Secara garis besar terdapat dua jenis pengujian yang umum dilakukan untuk menguji sebuah perangkat lunak. Pengujian yang paling umum digunakan tersebut adalah *Blackbox Testing* dan *Whitebox Testing*.

2.7.1. White Box Testing

White box testing merupakan sebuah langkah dalam pengujian untuk menguji bagian dalam koding program yang telah dibuat dengan mengikuti scenario atau test-case yang telah dibuat sebelumnya.

Dengan menggunakan white-box testing, penguji dapat membuat sebuah test-case yang menjamin bahwa semua independent path di dalam modul minimal sudah pernah dieksekusi 1 kali, menguji keputusan logika pada sisi true dan sisi false (if-else), mengeksekusi semua perulangan pada batasannya sehingga akan terdeteksi kapan sebuah perulangan akan berhenti (Pressman & Maxim, 2015)

1. Pengujian unit

Pengujian *unit* berfokus pada verifikasi *unit* terkecil dari perangkat lunak seperti modul atau komponen perangkat lunak (Pressman, 2010). Menggunakan dokumen perancangan sebagai dasar dalam melakukan pengujian, jalur penting diuji untuk

menemukan error dengan modul boundary. Pengujian unit berfokus pada proses logika dan struktur data.

Pengujian *unit* termasuk kedalam *WhiteBox Testing*. Pengujian unit diujikan untuk mengetahui aliran informasi masuk dan keluar dari program selama pengujian berlangsung. Setiap jalur diujikan untuk memastikan telah diuji setidaknya sekali.

2. Pengujian Integrasi

Pengujian integrasi merupakan teknik sistematis untuk membangun arsitektur perangkat lunak, sementara itu juga dilakukan pengujian untuk mengungkap kesalahan terkait dengan antarmuka (Pressman, 2010). Pengujian yang berlangsung akan menguji kelas-kelas yang saling terkait satu sama lain.

2.7.2. Pengujian Blackbox

Blackbox testing, focus kepada kebutuhan fungsional dari sebuah software. Blackbox testing digunakan untuk mendeteksi beberapa kesalahan seperti, kesalahan sebuah fungsional, kesalahan antarmuka, pada struktur data di dalam database (Pressman & Maxim, 2015).

1. Pengujian Validasi

Pengujian validasi berfokus pada tindakan pengguna dan keluaran yang dapat dikenali oleh sistem (Pressman, 2010). Pengujian validasi dapat dikatakan berhasil bila hasil keluaran dari sistem yang dibuat sama dengan yang diharapkan oleh pelanggan. Jika spesifikasi kebutuhan perangkat lunak telah dikembangkan, itu dapat menggambarkan atribut pengguna dan juga berisi bagian kriteria validasi yang membentuk dasar untuk pengujian validasi.

2.7.3. Pengujian Usability

Usability atau “ketergunaan” adalah sebuah atribut kualitas yang digunakan untuk menilai seberapa mudah antarmuka sebuah sistem digunakan oleh pengguna. Kata “kegunaan” juga mengacu pada metode yang digunakan untuk meningkatkan kemudahan penggunaan selama proses desain (Nielsen, 2012).

Menurut (Nielsen, 2012), *usability* didefinisikan menjadi 5 komponen kualitas, yaitu:

1. *Learnability*: Seberapa mudah aplikasi atau *website* digunakan oleh pengguna.
2. *Efficiency*: Kecepatan pengguna dalam menyelesaikan pengerjaan tugas dalam *website* atau aplikasi tertentu.
3. *Memorability*: Kemampuan pengguna mempertahankan ingatan pengetahuan terhadap desain *interface* yang ada setelah jangka waktu tertentu.

4. *Errors*: Kesalahan-kesalahan yang dilakukan oleh pengguna selama mengoperasikan *website* atau aplikasi yang tertentu.
5. *Satisfaction*: Kepuasan pengguna setelah menggunakan sebuah *website* atau aplikasi tertentu.

2.7.3.1 System Usability Scale

Kinerja usability diukur menggunakan metode System Usability Scale (SUS). SUS merupakan paket pengujian usability yang efektif dan handal untuk digunakan pada berbagai produk dan aplikasi (Bangor, Kortum, & Miller, 2009). SUS terdiri dari 10 pertanyaan dengan menggunakan skala likert 1 sampai 5. Pertanyaan bernomor ganjil (1, 3, 5, 7, 9) merupakan pertanyaan yang bernada positif. Sedangkan pertanyaan bernomor genap (2, 4, 6, 8, 10) merupakan pertanyaan yang bernada negative seperti yang ditunjukkan pada tabel 2.2.

Tabel 2.2 System Usability Scale

No	Pernyataan	STS	TS	N	S	SS
1	Saya sepertinya akan sering menggunakan aplikasi ini					
2	Saya melihat ada bagian fitur aplikasi ini yang cukup merepotkan, yang mestinya hal itu tidak perlu terjadi					
3	Saya rasa aplikasi ini mudah digunakan					
4	Saya sepertinya akan membutuhkan bantuan seorang teknisi agar bisa lancar menggunakan aplikasi ini					
5	Saya rasa fitur-fitur aplikasi ini sudah terhubung dengan baik satu sama lain					
6	Saya menemukan terlalu banyak ketidak konsistenan dalam aplikasi ini					
7	Saya pikir orang-orang akan dapat dengan cepat memahami cara kerja aplikasi ini					
8	Saya rasa aplikasi ini sangat sulit untuk digunakan					
9	Saya merasa aplikasi ini sesuai dengan apa yang dibutuhkan oleh perusahaan					
10	Saya harus mempelajari fitur-fitur yang terdapat di dalam aplikasi ini lebih lanjut agar merasa mudah menggunakan aplikasi ini					

Tabel 2.2 System Usability Scale

Saran :

Setiap pernyataan diberi bobot antara 0-4. Pada pernyataan bernomor ganjil (bernada positif), skor tiap pertanyaan dihitung dengan cara bobot tiap pertanyaan (xi) dikurangi 1 (Ardiansyah & Ghazali, 2016), sehingga ditulis:

$$x = xi - 1$$

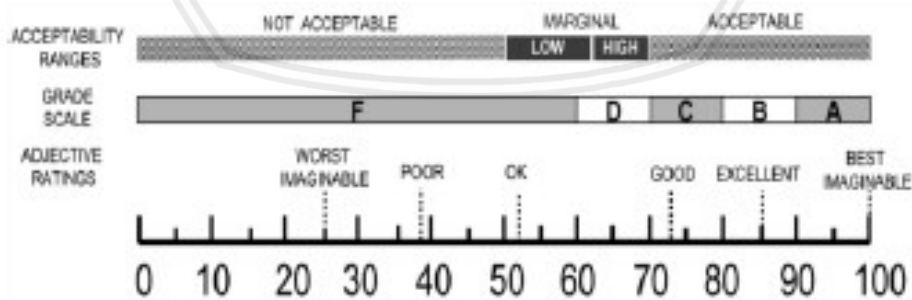
Begitu pula pertanyaan genap (bernada negatif), skor dihitung dengan cara 5 dikurangi bobot tiap pertanyaan (xi) (Ardiansyah & Ghazali, 2016) , sehingga ditulis :

$$y = 5 - xi$$

Total skor didapatkan dengan menjumlahkan seluruh skor tiap pertanyaan (genap maupun ganjil) (Ardiansyah & Ghazali, 2016). Sedangkan skor SUS didapat dengan rumus:

$$\text{Skor Akhir SUS} = (\sum x + \sum y) \times 2.5$$

Hasil perkalian jumlah skor x dan y dengan 2.5 merupakan skor akhir SUS. Kemudian dari skor akhir SUS tersebut, akan dapat ditentukan skala penerimaan pengguna terhadap sistem atau perangkat lunak yang dibangun berdasarkan 3 kategori yaitu, acceptability ranges, grade scale dan adjective ratings. Untuk lebih jelasna, dapat dilihat pada gambar 2.3.



Gambar 2.3 Rating dan Skala Konversi Skor SUS (Ardiansyah & Ghazali, 2016)

2.8. CodeIgniter

CodeIgniter adalah sebuah *web application framework* yang bersifat *open source* digunakan untuk membangun aplikasi *php* dinamis (Daqiqil, 2011). Tujuan utama pengembangan *CodeIgniter* adalah untuk membantu *developer* untuk mengerjakan aplikasi lebih cepat daripada menulis semua kode dari awal. *CodeIgniter* menyediakan berbagai macam *library* yang dapat mempermudah dalam pengembangan. *CodeIgniter* diperkenalkan kepada publik pada tahun 2006. *CodeIgniter* dibangun menggunakan konsep *Model-View-Controller pattern*. *CodeIgniter* sendiri merupakan salah satu *framework* tercepat dibandingkan *framework* lainnya.

2.9. MySQL

MySQL adalah sistem manajemen *database* yang cepat dan mudah digunakan, serta sudah banyak dipakai untuk berbagai kebutuhan (Enterprise, 2017). *MySQL* dikembangkan oleh *MySQL AB* Swedia. Hampir sebagian besar aplikasi *website* yang ada di internet dikembangkan menggunakan *MySQL*. Hal-hal yang membuat *MySQL* begitu populer adalah karena berlisensi *open-source*, sehingga dapat digunakan secara gratis, dapat bekerja dengan banyak sistem operasi dan bahasa-bahasa pemrograman, bekerja dengan cepat dan baik dan dapat dikustomisasi sesuai keinginan.

2.10. Proses Bisnis

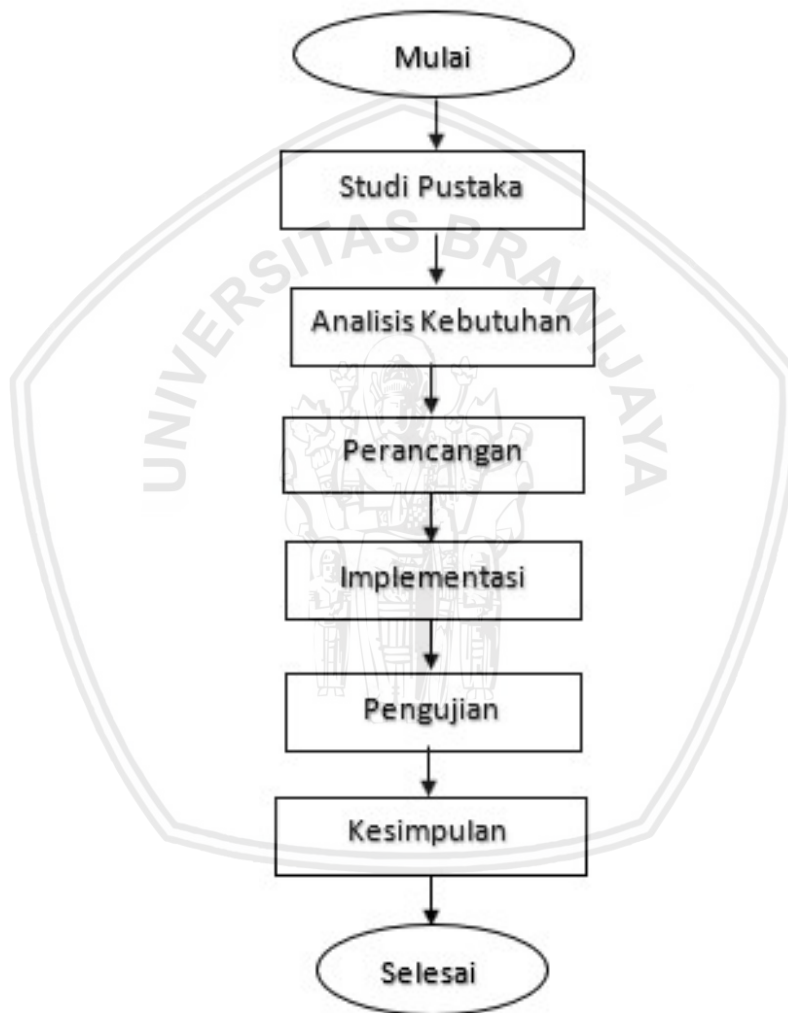
Bisnis proses adalah kumpulan aktivitas yang membutuhkan satu atau lebih inputan dan menghasilkan output yang bermanfaat/bernilai bagi pelanggan (Hammer & Champy, 1993).

Proses bisnis dapat dibagi menjadi beberapa subproses yang mempunyai atribut tersendiri tetapi juga dapat berkontribusi untuk mencapai tujuan dari superprosesnya. Analisis pada proses bisnis umumnya meliputi pemetaan pada proses dan subproses pada bagiannya hingga tingkatan aktivitas atau kegiatan. Berikut merupakan beberapa karakteristik umum yang dianggap harus ada pada proses bisnis yaitu:

1. Definitif: proses bisnis harus memiliki batasan, masukan dan keluaran yang jelas
2. Urutan: proses bisnis harus berurutan waktu dan ruang aktivitasnya
3. Pelanggan: proses bisnis harus memiliki penerima hasil proses
4. Nilai tambah: transformasi harus memiliki nilai tambah yang lebih baik untuk penerima
5. Keterkaitan: proses bisnis harus terkait dengan struktur organisasi sehingga tidak dapat berdiri sendiri.

BAB 3 METODOLOGI

Bab ini menjelaskan langkah-langkah apa saja yang akan dilakukan untuk membuat sistem informasi pemesanan *travel* berbasis web. Jenis penelitian pada penelitian ini adalah *Research and Development* yang berarti penelitian ini ditujukan untuk menghasilkan sebuah pengembangan produk terapan yang baru dari produk yang lama. Tahapan-tahapan yang akan dilakukan pada pengerjaan skripsi akan diilustrasikan pada Gambar 3.1 yang terdiri dari Studi Pustaka, Analisis Kebutuhan, Perancangan, Implementasi, Pengujian, Analisis, dan Kesimpulan.



Gambar 3.1 Diagram Alir Metode Penelitian

3.1 Studi pustaka

Studi pustaka adalah suatu penjelajahan teori-teori yang berasal dari penelitian sebelumnya yang dapat dijadikan sebagai acuan dan menjadi pendukung terhadap penelitian yang sedang dijalankan. Teori-teori yang dikaji berasal dari buku dan jurnal yang membahas tentang sistem informasi, *software development life cycle*, *codeigniter*, pengujian *blackbox*, *whitebox*, *usability testing* dan teori-teori yang memiliki keterkaitan dengan penelitian ini.

3.2. Analisis kebutuhan

Analisis kebutuhan yang akan dilakukan dalam penelitian ini adalah analisis kebutuhan pada Travel Lia Jaya Sentosa. Pada tahapan analisis kebutuhan ini, pendekatan yang digunakan untuk menganalisis kebutuhan adalah teknik yang berorientasi objek (*Object Oriented Analysis*), yang berarti setiap kebutuhan akan dipetakan menjadi sebuah objek-objek dan kelas-kelas. Proses analisis kebutuhan pada Travel Lia Jaya Sentosa adalah sebagai berikut:

1. Elisitasi Kebutuhan

Pada proses elisitasi kebutuhan akan dilakukan penggalan kebutuhan melalui wawancara pada Travel Lia Jaya Sentosa dan lebih lanjutnya akan dispesifikasikan pada proses spesifikasi kebutuhan.

2. Spesifikasi Kebutuhan

Pada proses spesifikasi kebutuhan akan dilakukan spesifikasi dari hasil elisitasi dan kemudian kebutuhan yang telah dianalisis akan dikelompokkan menjadi kebutuhan fungsional dan non-fungsional untuk mengetahui apa saja yang dapat dilakukan oleh sistem serta identifikasi aktor untuk mengetahui siapa saja yang dapat ikut terlibat di dalam sistem. Pada tahap ini pemodelan kebutuhan dalam bentuk *use case diagram* serta *use case scenario* akan dilakukan.

3. Validasi Kebutuhan

Pada proses ini akan dilakukan pengecekan untuk menjamin bahwa pernyataan kebutuhan yang telah didefinisikan dan dispesifikasikan adalah benar, akurat dan lengkap.

4. Manajemen Kebutuhan

Pada proses ini akan dilakukan control terhadap kebutuhan yang sedang maupun yang telah didefinisikan dan dispesifikasikan, setiap kebutuhan akan diberikan kode baru agar lebih mudah diidentifikasi.

3.3. Perancangan

Perancangan dilakukan dengan menggunakan pendekatan berbasis objek dengan membuat diagram-diagram *UML (Unified Modeling Language)* seperti *sequence diagram* dan *class diagram* dari sistem yang akan dibuat agar nantinya sistem yang sesuai dengan kebutuhan-kebutuhan yang sudah didefinisikan di awal. Pengerjaan pada perancangan ini akan dilakukan sebagai berikut:

1. Perancangan Arsitektur
Dari pemodelan *use case scenario* yang telah dilakukan sebelumnya kemudian akan dibuat pemodelan *sequence diagram*. *Sequence diagram* yang dijelaskan adalah tiga fitur utama. Kemudian akan dilakukan pengelompokan kelas dan atribut yang ada di dalamnya kemudian membuat *class diagram*.
2. Perancangan Komponen
Pada perancangan komponen nantinya dibuat 3 algoritma utama yang ada di dalam sistem. Algoritma tersebut kemudian akan diuraikan dalam bentuk *pseudocode*.
3. Perancangan Data
Pada perancangan data terdapat tabel basis data yang dibuat dalam sistem dalam *Conceptual Data Model (CDM)* yang selanjutnya akan menjadi petunjuk dalam pembuatan basis data di dalam sistem yang dibangun.
4. Perancangan Antarmuka
Selanjutnya, yang terakhir adalah pembuatan rancangan antarmuka yang nantinya akan dapat menjadi acuan desain antarmuka pada tahap implementasi antarmuka sistem.

3.4. Implementasi

Pada tahap implementasi akan dilakukan *coding* untuk merealisasikan analisis dan perancangan sistem informasi pemesanan *travel* Lia Jaya Sentosa yang telah dibuat agar nantinya hasil dari implementasi yang akan dibuat dapat ditelusuri dan sesuai dengan analisis dan perancangan yang telah dibuat sebelumnya. Implementasi itu sendiri terdiri dari beberapa tahapan yaitu:

1. Implementasi Komponen
Implementasi berbasis *web*, implementasi yang akan dilakukan nantinya adalah *coding* menggunakan bahasa pemrograman *PHP* dengan *framework CodeIgniter, JQuery, Bootstrap, CSS, Javascript, HTML*.
2. Implementasi Antar Muka
Perancangan antar muka yang telah dilakukan sebelumnya, selanjutnya diimplementasikan ke dalam bentuk antar muka perangkat lunak.
3. Implementasi Basis Data
Pada tahap ini, penerapan dilakukan dengan mengacu tahap perancangan basis data yang telah dibuat pada tahap perancangan sistem. Basis data yang diterapkan menggunakan basis data *Database Management System (DBMS)*

MySQL. Untuk pengelolaannya digunakan aplikasi *web phpMyAdmin* untuk memudahkan dalam manipulasi tabel atau basis data yang akan dibuat.

3.5. Pengujian

Pengujian dilakukan agar dapat meminimalisir *bug* dan kekurangan sistem. Untuk pengujiannya akan dilakukan dengan metode *black box* serta metode *white box*. Berikut adalah penjelasan dari pengujian yang akan dilakukan :

1. Pengujian *Unit*

Pengujian unit dilakukan dengan menggunakan metode *basic path testing* yang merupakan pengujian *white-box*. Pengujian ini dilakukan pada 3 sampel operasi.

2. Pengujian Integrasi

Pengujian yang melibatkan kelas-kelas di dalam sistem yang memiliki keterkaitan satu sama lain. Pengujian ini nantinya akan dilakukan pada 3 sampel operasi.

3. Pengujian Validasi

Pengujian sistem untuk memastikan sistem telah dibangun dan berjalan sesuai dengan kebutuhan apa saja yang telah didefinisikan di awal atau lebih tepatnya pada tahap analisis. Pengujian ini akan menguji kebutuhan fungsional dan non fungsional.

4. Pengujian *Usability*

Pengujian desain sistem dengan cara sebagai berikut:

- Menentukan siapa saja responden yang akan membantu untuk menguji sistem.
- Menentukan task apa saja yang akan dikerjakan responden.
- Responden mengerjakan task-task yang sudah ditentukan.
- Responden mengisi kuesioner.
- Menghitung hasil dari pengisian kuesioner berdasarkan rumus kemudian akan didapatkan skor akhir SUS dari sistem.
- Mengkonversi nilai skor akhir SUS berdasarkan rating dan skala skor SUS.

3.6. Kesimpulan

Kesimpulan akan diambil setelah melalui keseluruhan tahapan yang ada dimulai dari analisis kebutuhan, perancangan, implementasi hingga pengujian. Kesimpulan adalah hasil dari pengujian untuk mengetahui rumusan masalah dapat terselesaikan atau tidak. Setelah kesimpulan diambil, terdapat subbab saran yang digunakan untuk memperlihatkan apa saja kekurangan yang perlu ditambahkan ke dalam sistem bagi penelitian selanjutnya.

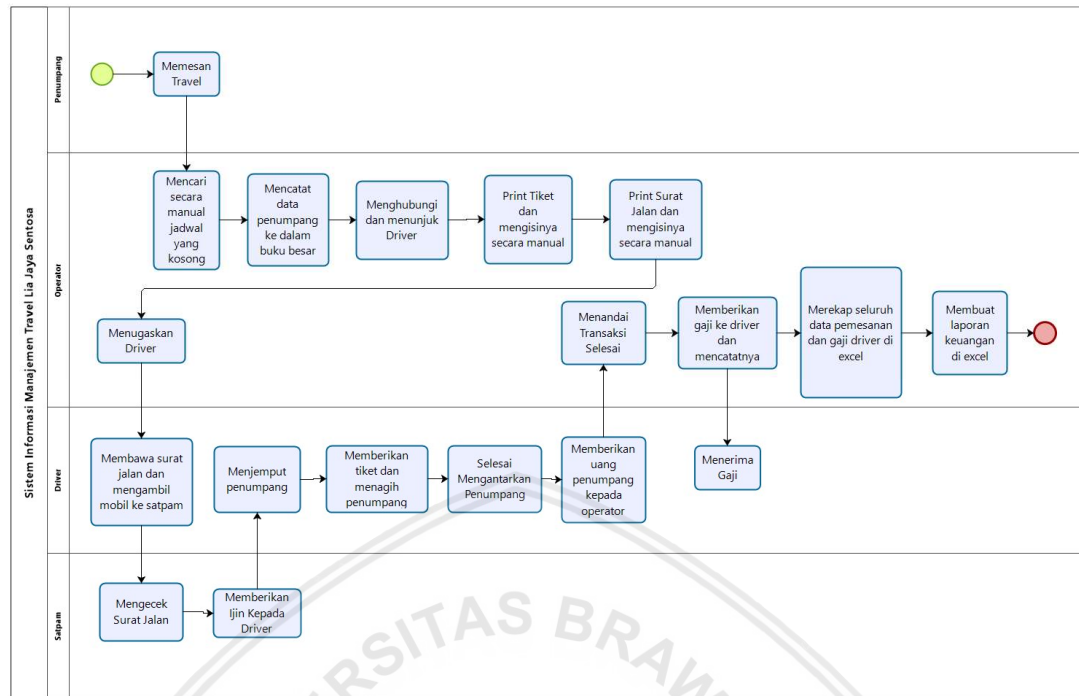
BAB 4 ANALISIS KEBUTUHAN

Bab analisis kebutuhan membahas proses analisis kebutuhan Sistem Informasi Manajemen *Travel* Berbasis *Web* pada Perusahaan Lia Jaya Sentosa. Analisa kebutuhan pada perangkat lunak dibuat untuk menggambarkan kebutuhan apa saja yang harus disediakan oleh sistem untuk memenuhi kebutuhan pengguna. Dalam rekayasa kebutuhan terdapat beberapa proses yang harus dilakukan yaitu proses elisitasi, spesifikasi, validasi, manajemen kebutuhan, pemodelan use case diagram dan use case scenario.

4.1. Elisitasi Kebutuhan

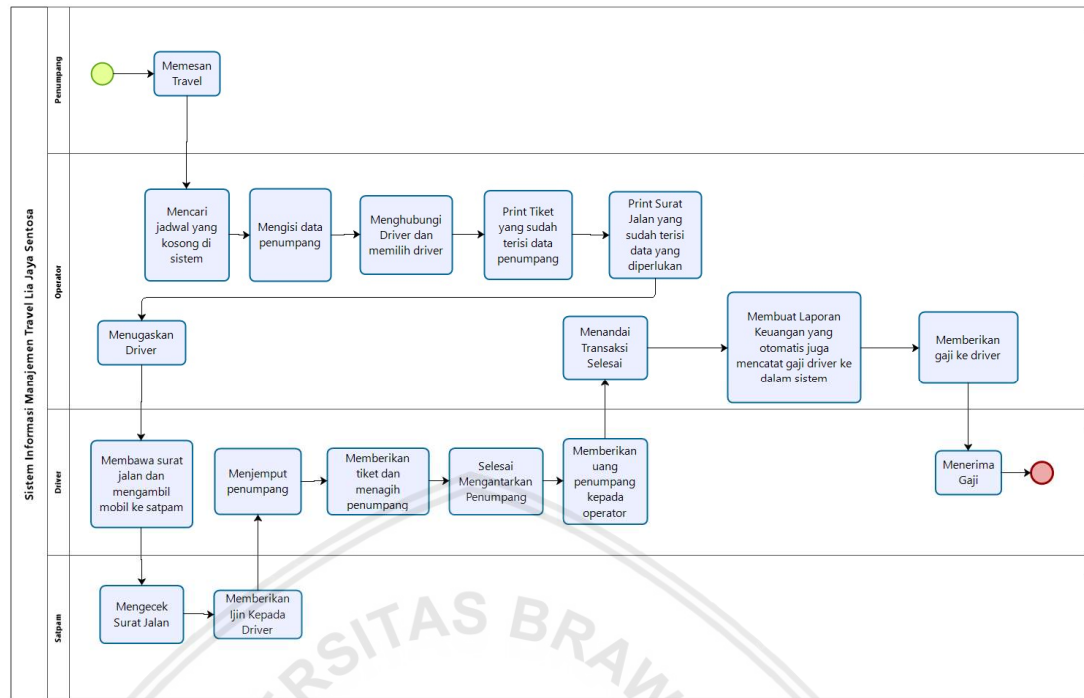
Elisitasi kebutuhan ialah proses yang dilakukan untuk menemukan atau menggali kebutuhan sistem yang didapatkan melalui pengamatan terhadap suatu sistem yang sudah ada atau dengan melakukan diskusi dengan calon user. Hal ini akan membantu kepada pengembang sistem untuk selanjutnya menspesifikasikan sistem yang dibutuhkan (Sommerville, 2011).

Pada pengembangan sistem informasi manajemen travel pada perusahaan Lia Jaya Sentosa, proses elisitasi dilakukan dengan wawancara dengan 3 orang operator. Dari hasil wawancara tersebut, diketahui bahwa untuk proses bisnis yang terjadi di perusahaan Lia Jaya Sentosa masih belum efektif dan mudah untuk dilakukan. Selain itu, grafik untuk menunjukkan rute yang paling banyak dipesan per bulan, grafik laporan keuangan dari pemesanan travel dan grafik pemesanan travel terbanyak setiap bulan diperlukan sebagai acuan perusahaan untuk menentukan strategi ke depannya. Misalnya, adalah strategi untuk mengalokasikan mobil, dengan adanya grafik yang menunjukkan rute yang paling banyak dipesan akan memberikan gambaran perusahaan rute mana yang lebih menguntungkan sehingga ke depannya untuk rute yang paling banyak menguntungkan dapat dialokasikan mobil lebih banyak pada jadwalnya. Kemudian, hasil dari proses bisnis pemesanan travel pada perusahaan Lia Jaya Sentosa dapat dilihat pada gambar 4.1 dan gambar 4.2.



Gambar 4.1 Proses Bisnis As-Is

Berdasarkan gambar 4.1, proses bisnis as-is merupakan proses bisnis pemesanan travel yang dijalani oleh perusahaan pada saat ini. Banyaknya proses yang masih menggunakan kertas (paperness) mulai dari pencarian jadwal keberangkatan yang kosong dengan cara harus melihat dan mencari jadwal yang kosong dari buku besar terlebih dahulu, pencatatan data penumpang ke dalam buku besar, pengisian tiket travel, pengisian surat jalan, pencatatan gaji driver, pembuatan laporan keuangan membuatnya kurang efektif, memakan biaya dan waktu yang lebih banyak.



Gambar 4.2 Proses Bisnis To-Be

Berdasarkan gambar 4.2, Proses Bisnis To-Be adalah proses bisnis yang merupakan usulan untuk memudahkan proses bisnis yang terjadi di dalam perusahaan dengan melibatkan sebuah sistem informasi di dalamnya. Tentunya, diharapkan dengan adanya sistem informasi ini, akan lebih memudahkan operator dalam mengelola proses bisnis yang ada di dalam perusahaan mulai dari mencari jadwal melalui fitur pencarian, menyimpan data pemesanan ke dalam database, pencetakan tiket dan surat jalan yang sudah terisi data yang dibutuhkan berdasarkan pemesanan yang telah dilakukan, menyimpan data gaji driver ke dalam database, pembuatan laporan keuangan hingga pembuatan grafik-grafik seperti grafik rute yang paling banyak dipesan, grafik laporan keuangan dan grafik jumlah pemesanan pada setiap bulan yang dapat digunakan perusahaan sebagai acuan dalam menentukan strategi pengalokasian mobil atau jadwal ke depannya.

Untuk mengenal siapa saja yang berperan di dalam perusahaan, berikut merupakan identifikasi stakeholder dapat dilihat dalam tabel 4.1 berikut.

Tabel 4.1 Identifikasi Stakeholder

No	Stakeholder	Deskripsi
1	Pimpinan	Merupakan orang yang menentukan jadwal keberangkatan travel pada layanan pemesanan travel.
2	Operator	Merupakan orang yang bertanggung jawab untuk menginputkan jadwal keberangkatan travel, mencatat setiap pemesanan travel yang terjadi dan pembuatan laporan keuangan pemesanan travel.
3	Penumpang	Merupakan orang yang memesan layanan pemesanan travel.
4	Driver	Merupakan orang yang membawa kendaraan untuk mengantarkan penumpang.
5	Satpam	Merupakan orang yang menjaga kendaraan travel Lia Jaya Sentosa.

4.1.1. Gambaran Umum Sistem

Berdasarkan permasalahan yang telah dirumuskan di dalam latar belakang, yang merupakan hasil dari wawancara terhadap calon pengguna, didapatkan sebuah kebutuhan sistem yang akan digunakan oleh operator Perusahaan Lia Jaya Sentosa untuk mempermudah proses pengolahan data pada layanan-layanan yang ada di Perusahaan Lia Jaya Sentosa mulai dari pencatatan, pemesanan, pembuatan laporan, pencetakan tiket, pencetakan surat jalan dan pembuatan grafik pemesanan. Data-data yang akan diolah pada sistem adalah data jadwal keberangkatan, data driver, data kota dan data mobil yang ada pada Perusahaan Lia Jaya Sentosa. Sedangkan laporan dan grafik yang akan diolah adalah laporan gaji driver, laporan keuangan pemesanan travel, grafik rute yang paling sering dipesan, grafik laporan keuangan dan grafik jumlah pemesanan per bulan.

4.2. Spesifikasi Kebutuhan

Spesifikasi kebutuhan dilakukan untuk mengidentifikasi aktor beserta batasannya serta menspesifikasi hasil dari proses elisitasi yang dikategorikan menjadi beberapa kebutuhan seperti kebutuhan fungsional dan kebutuhan *non-fungsional*. Kemudian dilakukan pemodelan kebutuhan dalam bentuk *use case* diagram serta *use case scenario*.

4.2.1. Identifikasi Aktor

Identifikasi aktor bertujuan untuk memberikan gambaran tentang pengguna yang akan terlibat dalam penggunaan sistem. Identifikasi aktor yang akan berperan dalam sistem dapat dilihat pada Tabel 4.2.

Tabel 4.2 Identifikasi Aktor

No	Aktor	Deskripsi
1	Operator	Operator memiliki peran untuk mengolah data pemesanan layanan seperti pengolahan data pemesanan travel. data mobil, data driver dan data jadwal keberangkatan.

4.2.2. Daftar Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan apa saja yang harus dimiliki oleh sistem, bagaimana reaksi sistem ketika menerima masukan tertentu dan situasi tertentu. Kebutuhan fungsional didapat dari hasil wawancara kepada operator perusahaan Lia Jaya Sentosa. Masing-masing kebutuhan akan diberikan kode MTF_YYZZ. Dimana MTF merupakan nama dari sistem yaitu Manajemen Travel Fungsional, YY menunjukkan nomor kebutuhan utama sedangkan ZZ menunjukkan nomor spesifikasi dari kebutuhan utama. Spesifikasi kebutuhan fungsional dapat dilihat pada tabel 4.3 berikut.

Tabel 4.3 Spesifikasi Kebutuhan Fungsional

No	Kode	Deskripsi Kebutuhan	Nama Use Case	Aktor
1	MTF_0100	Sistem harus dapat menyediakan fungsi login untuk mengautentikasi operator.	Login	Operator
2	MTF_0200	Sistem harus dapat menyediakan fungsi logout agar operator dapat keluar dari sistem.	Logout	Operator
3	MTF_0300	Sistem harus menyediakan fungsi agar operator dapat melihat grafik pemesanan.	Melihat Beranda	Operator
4	MTF_0400	Sistem harus menyediakan fungsi agar operator dapat menyeleksi grafik	Mencari Grafik Pemesanan	Operator

Tabel 4.3 Spesifikasi Kebutuhan Fungsional

No	Kode	Deskripsi Kebutuhan	Nama Use Case	Aktor
		pemesanan berdasarkan bulan dan tahun tertentu.		
5	MTF_0500	Sistem harus menyediakan fungsi untuk melihat grafik pemesanan terbanyak berdasarkan tahun yang telah ditentukan.	Melihat Grafik Pemesanan Terbanyak	Operator
6	MTF_0600	Sistem harus menyediakan fungsi untuk melihat grafik rute pemesanan travel yang paling sering dipesan berdasarkan bulan dan tahun yang telah ditentukan.	Melihat Grafik Rute Pemesanan Travel	Operator
7	MTF_0700	Sistem harus menyediakan fungsi untuk melihat grafik laporan keuangan pemesanan travel berdasarkan tahun yang telah ditentukan.	Melihat Grafik Laporan Keuangan	Operator
8	MTF_0800	Sistem harus menyediakan fungsi bagi operator untuk menambah data jadwal travel harian. Mobil yang dapat digunakan pada jadwal travel harian adalah mobil yang belum digunakan pada jadwal travel tambahan (MTF_0401).	Menambah Jadwal Travel Harian	Operator
9	MTF_0900	Sistem harus dapat menyediakan fungsi bagi operator untuk melihat jadwal keberangkatan travel harian.	Melihat Jadwal Travel Harian	Operator

Tabel 4.3 Spesifikasi Kebutuhan Fungsional

No	Kode	Deskripsi Kebutuhan	Nama Use Case	Aktor
10	MTF_1000	Sistem harus menyediakan fungsi bagi operator untuk mengubah data jadwal travel harian.	Mengubah Jadwal Travel Harian	Operator
11	MTF_1100	Sistem harus menyediakan fungsi bagi operator untuk menghapus data jadwal travel harian.	Menghapus Jadwal Travel Harian	Operator
12	MTF_1200	Sistem harus menyediakan fungsi bagi operator untuk menambah data jadwal travel tambahan. Mobil yang dapat digunakan pada jadwal travel tambahan adalah mobil yang belum digunakan pada jadwal travel harian (MTF_0801)	Menambah Jadwal Travel Tambahan	Operator
13	MTF_1300	Sistem harus dapat menyediakan fungsi bagi operator untuk melihat jadwal keberangkatan travel tambahan.	Melihat Jadwal Travel Tambahan	Operator
14	MTF_1400	Sistem harus menyediakan fungsi bagi operator untuk mengubah data jadwal travel harian.	Mengubah Jadwal Travel Tambahan	Operator
15	MTF_1500	Sistem harus menyediakan fungsi bagi operator untuk menghapus data jadwal travel tambahan.	Menghapus Jadwal Travel Tambahan	Operator
16	MTF_1600	Sistem harus dapat menyediakan fungsi bagi operator untuk menambah data pemesanan travel.	Menambah Pemesanan Travel	Operator

Tabel 4.3 Spesifikasi Kebutuhan Fungsional

No	Kode	Deskripsi Kebutuhan	Nama Use Case	Aktor
		Pemesanan travel dapat dilakukan oleh operator maksimal 2 jam sebelum keberangkatan (MTF_1201).		
17	MTF_1700	Sistem harus dapat menyediakan fungsi bagi operator untuk mencari jadwal yang tersedia berdasarkan asal, tujuan dan tanggal keberangkatan.	Mencari Jadwal Tersedia	Operator
18	MTF_1800	Sistem harus menyediakan fungsi bagi operator untuk melihat data pemesanan travel.	Melihat Pemesanan Travel	Operator
19	MTF_1900	Sistem harus menyediakan fungsi bagi operator untuk mengubah data pemesanan travel.	Mengubah Pemesanan Travel	Operator
20	MTF_2000	Sistem harus menyediakan fungsi bagi operator untuk menghapus data pemesanan travel.	Menghapus Pemesanan Travel	Operator
21	MTF_2100	Sistem harus menyediakan fungsi bagi operator untuk melihat manajemen pemesanan.	Melihat Manajemen Pemesanan	Operator
22	MTF_2200	Sistem harus menyediakan fungsi bagi operator untuk mengubah driver.	Mengubah Driver	Operator
23	MTF_2300	Sistem harus menyediakan fungsi bagi operator untuk mencetak surat jalan.	Mencetak Surat Jalan	Operator

Tabel 4.3 Spesifikasi Kebutuhan Fungsional

No	Kode	Deskripsi Kebutuhan	Nama Use Case	Aktor
24	MTF_2400	Sistem harus menyediakan fungsi bagi operator untuk melihat daftar penumpang berdasarkan mobil dan tanggal keberangkatan mobil.	Melihat Daftar Penumpang	Operator
25	MTF_2500	Sistem harus menyediakan fungsi bagi operator untuk mencetak tiket travel penumpang	Mencetak Tiket Travel	Operator
26	MTF_2600	Sistem harus menyediakan fungsi bagi operator untuk mengubah status transaksi.	Menandai Transaksi Selesai	Operator
27	MTF_2700	Sistem harus menyediakan fungsi bagi operator untuk menambah data mobil.	Menambah Data Mobil	Operator
28	MTF_2800	Sistem harus menyediakan fungsi bagi operator untuk melihat data mobil.	Melihat Data Mobil	Operator
29	MTF_2900	Sistem harus menyediakan fungsi bagi operator untuk mengubah data mobil.	Mengubah Data Mobil	Operator
30	MTF_3000	Sistem harus menyediakan fungsi bagi operator untuk menghapus data mobil.	Menghapus Data Mobil	Operator
31	MTF_3100	Sistem harus menyediakan fungsi bagi operator untuk menambah data driver.	Menambah Data Driver	Operator
32	MTF_3200	Sistem harus menyediakan fungsi bagi operator untuk melihat data driver.	Melihat Data Driver	Operator
33	MTF_3300	Sistem harus menyediakan fungsi bagi operator untuk mengubah data driver.	Mengubah Data Driver	Operator

Tabel 4.3 Spesifikasi Kebutuhan Fungsional

No	Kode	Deskripsi Kebutuhan	Nama Use Case	Aktor
34	MTF_3400	Sistem harus menyediakan fungsi bagi operator untuk menghapus data driver.	Menghapus Data Driver	Operator
35	MTF_3500	Sistem harus menyediakan fungsi bagi operator untuk menambah data kota.	Menambah Data Kota	Operator
36	MTF_3600	Sistem harus menyediakan fungsi bagi operator untuk melihat data kota.	Melihat Data Kota	Operator
37	MTF_3700	Sistem harus menyediakan fungsi bagi operator untuk mengubah data kota.	Mengubah Data Kota	Operator
38	MTF_3800	Sistem harus menyediakan fungsi bagi operator untuk menghapus data kota.	Menghapus Data Kota	Operator
39	MTF_3900	Sistem harus menyediakan fungsi bagi operator untuk menampilkan gaji driver.	Melihat Gaji Driver	Operator
40	MTF_4000	Sistem harus menyediakan fungsi untuk mencetak gaji driver.	Mencetak Gaji Driver	Operator
41	MTF_4100	Sistem harus menyediakan fungsi untuk menyeleksi gaji driver berdasarkan bulan.	Mencari Gaji Driver	Operator
42	MTF_4200	Sistem harus menyediakan fungsi untuk menampilkan laporan keuangan pemesanan travel.	Melihat Laporan Keuangan Pemesanan Travel	Operator
43	MTF_4300	Sistem harus menyediakan fungsi untuk menambah laporan keuangan pemesanan travel	Menambah Laporan Keuangan Pemesanan Travel	Operator

Tabel 4.3 Spesifikasi Kebutuhan Fungsional

No	Kode	Deskripsi Kebutuhan	Nama Use Case	Aktor
44	MTF_4400	Sistem harus menyediakan fungsi untuk mengubah laporan keuangan pemesanan travel	Mengubah Laporan Keuangan Pemesanan Travel	Operator
45	MTF_4500	Sistem harus menyediakan fungsi untuk menghapus laporan keuangan pemesanan travel	Menghapus Laporan Keuangan Pemesanan Travel	Operator
46	MTF_4600	Sistem harus menyediakan fungsi untuk mencetak laporan keuangan pemesanan travel.	Mencetak Laporan Keuangan Pemesanan Travel	Operator
47	MTF_4700	Sistem harus menyediakan fungsi untuk menyeleksi tampilan laporan keuangan berdasarkan bulan yang ditentukan.	Mencari Laporan Keuangan Pemesanan Travel	Operator

4.2.3. Daftar Kebutuhan Non Fungsional

Kebutuhan non fungsional merupakan kebutuhan yang berkaitan dengan batasan atau kualitas yang dimiliki oleh sistem. Dalam penelitian ini, kebutuhan non fungsional didapatkan melalui hasil wawancara dengan operator Lia Jaya Sentosa. Masing-masing kebutuhan akan diberikan kode MTNF-ZZ. Setiap kebutuhan telah divalidasi dan diverifikasi dengan cara melakukan diskusi dengan operator perusahaan Lia Jaya Sentosa. Spesifikasi kebutuhan non fungsional dapat dilihat pada Tabel 4.4 berikut.

Tabel 4.4 Spesifikasi Kebutuhan Non Fungsional

No	Nama Kebutuhan	Kode	Spesifikasi
1	Usability	MTNF-01	Sistem dapat memudahkan pekerjaan yang dilakukan oleh operator dengan mengujinya menggunakan SUS <i>questionnaire</i> dan kemudian

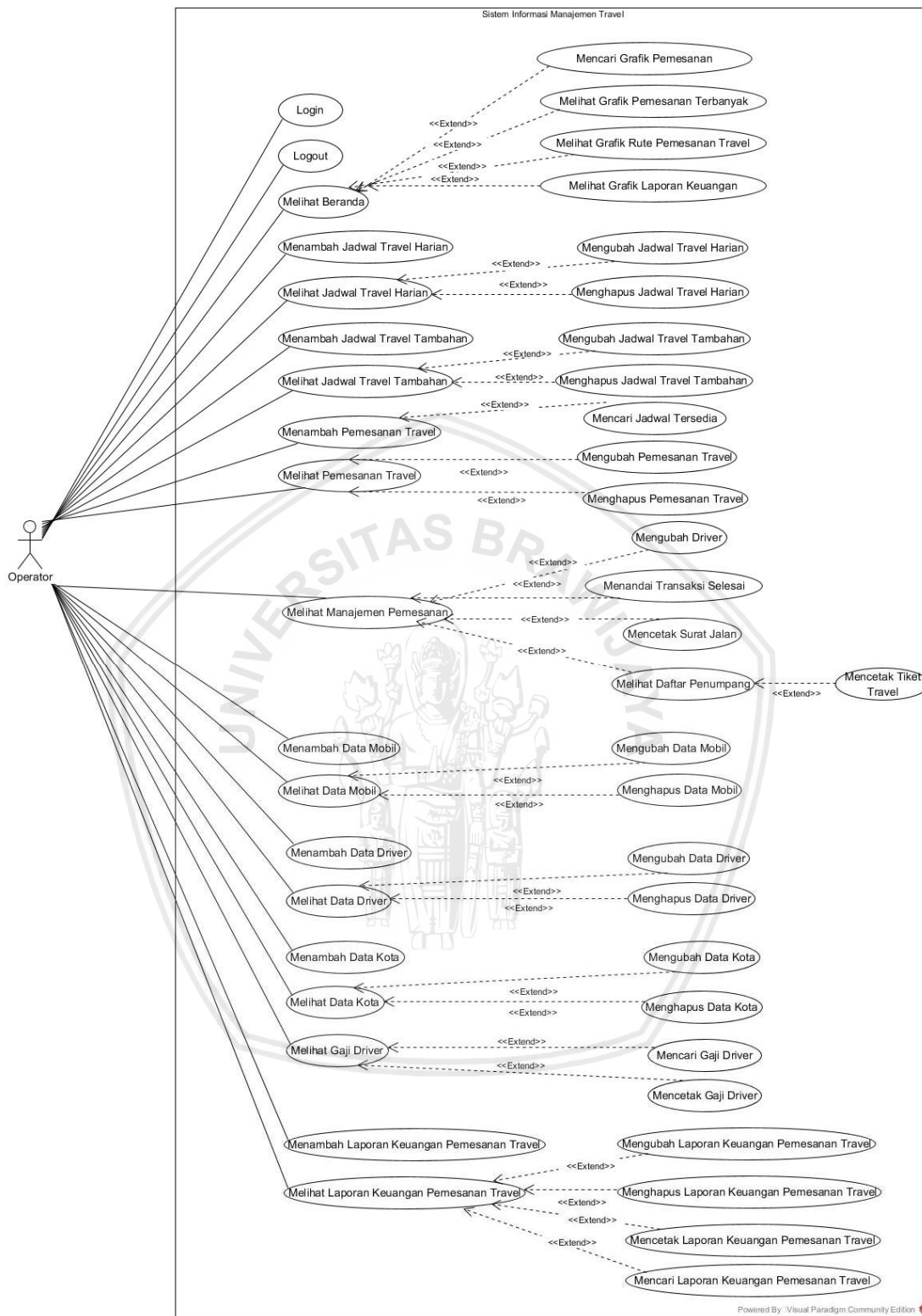
Tabel 4.4 Spesifikasi Kebutuhan Non Fungsional

No	Nama Kebutuhan	Kode	Spesifikasi
			didapatkan skor akhir SUS sebagai tolak ukur kemudahan sistem.

4.2.4. Use Case Diagram

Use case diagram digunakan untuk memodelkan apa saja perilaku aktor kepada sistem. Dalam pemodelannya, yang ditekankan adalah tentang “apa” yang dapat dilakukan oleh aktor terhadap sistem, bukan “bagaimana” perilaku aktor terhadap sistem. Use case diagram yang akan dimodelkan berdasarkan dari spesifikasi kebutuhan fungsional yang telah didefinisikan sebelumnya. Pemodelannya dapat dilihat pada gambar 4.3 berikut.





Gambar 4.3 Use Case Diagram

4.2.5. Use Case Scenario

Use case scenario menjelaskan alur dari usecase yang telah dibuat. Use case scenario dimodelkan menggunakan tabel yang terdiri dari actor, objective, pre-condition, main flow dan post condition. Di dalam penelitian ini, terdapat 44 use case scenario yang dibuat.

4.2.5.1. Use Case Scenario Login

Pada tabel 4.5 menjelaskan use case scenario login dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.5 Use Case Scenario Login

MTF_0100 : Login	
Objective	Melakukan proses autentikasi untuk masuk ke dalam sistem
Actor	Operator
Pre-Condition	Telah terhubung dengan internet dan membuka halaman <i>login website</i> Sistem Informasi Pemesanan Travel Lia Jaya Sentosa.
Main Flow	<ol style="list-style-type: none"> 1. Operator mengisi <i>username</i> dan <i>password</i>. 2. Menekan tombol Login. 3. Sistem mengautentikasi apakah <i>username</i> dan <i>password</i> yang dimasukkan telah terdaftar di dalam database. 4. Masuk ke halaman Beranda.
Alternative Flow	<ol style="list-style-type: none"> 3a. Jika username atau password belum terisi maka sistem akan menampilkan peringatan "Masukkan Username" dan "Masukkan Password" kemudian kembali pada main flow nomor 1. 4a. Jika username atau password gagal diautentikasi maka sistem akan menampilkan pesan "Username atau password yang dimasukkan salah" dan akan kembali pada main flow nomor 1.
Post-Condition	User berhasil terotentikasi dan telah masuk ke dalam halaman beranda.

4.2.5.2. Use Case Scenario Logout

Pada tabel 4.6 menjelaskan use case scenario *logout* dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.6 Use Case Scenario Logout

MTF_0200 : Logout	
Objective	Keluar dari sistem
Actor	Operator
Pre-Condition	Telah terautentikasi dan masuk ke dalam sistem.
Main Flow	<ol style="list-style-type: none"> 1. Operator menekan tombol logout. 2. Menghapus session dan menampilkan halaman <i>login</i>.
Alternative Flow	-
Post-Condition	Berhasil keluar dari sistem.

4.2.5.3. Use Case Scenario Melihat Beranda

Pada tabel 4.7 menjelaskan use case scenario melihat beranda dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.7 Use Case Scenario Melihat Beranda

MTF_0300 : Melihat Beranda	
Objective	Melihat Beranda
Actor	Operator
Pre-Condition	Telah terautentikasi dan masuk ke dalam sistem.
Main Flow	<ol style="list-style-type: none"> 1. Menekan tombol beranda. 2. Sistem menampilkan halaman beranda.
Alternative Flow	-
Post-Condition	Berhasil melihat beranda.

4.2.5.4. Use Case Scenario Mencari Grafik Pemesanan

Pada tabel 4.8 menjelaskan use case scenario mencari grafik pemesanan dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.8 Use Case Scenario Mencari Grafik Pemesanan

MTF_0400 : Mencari Grafik Pemesanan	
Objective	Menyeleksi grafik yang ditampilkan sesuai dengan bulan ataupun tahun yang dimasukkan
Actor	Operator
Pre-Condition	Membuka halaman beranda
Main Flow	<ol style="list-style-type: none"> 1. Memilih bulan dan tahun 2. Menekan tombol dengan icon kaca pembesar 3. Sistem menyeleksi penampilan grafik berdasarkan bulan dan tahun yang dipilih 4. Sistem menampilkan grafik-grafik
Alternative Flow	-
Post-Condition	Berhasil melihat grafik sesuai dengan bulan ataupun tahun yang dimasukkan

4.2.5.5. Use Case Scenario Melihat Grafik Pemesanan Terbanyak

Pada tabel 4.9 menjelaskan use case scenario melihat grafik pemesanan terbanyak dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.9 Use Case Scenario Melihat Grafik Pemesanan Terbanyak

MTF_0500 : Melihat Grafik Pemesanan Terbanyak	
Objective	Melihat Grafik Pemesanan Terbanyak
Actor	Operator
Pre-Condition	Telah terautentikasi dan masuk ke dalam sistem.
Main Flow	<ol style="list-style-type: none"> 1. Menekan tombol beranda. 2. Sistem menampilkan grafik pemesanan terbanyak.
Alternative Flow	-
Post-Condition	Berhasil melihat grafik grafik pemesanan terbanyak.

4.2.5.6. Use Case Scenario Melihat Grafik Rute Pemesanan Travel

Pada tabel 4.10 menjelaskan use case scenario melihat grafik rute pemesanan travel dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus

dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.10 Use Case Scenario Melihat Grafik Rute Pemesanan Travel

MTF_0600 : Melihat Grafik Rute Pemesanan Travel	
Objective	Melihat Grafik Rute Pemesanan Travel
Actor	Operator
Pre-Condition	Telah terautentikasi dan masuk ke dalam sistem.
Main Flow	1. Menekan tombol beranda. 2. Sistem menampilkan grafik rute pemesanan travel.
Alternative Flow	-
Post-Condition	Berhasil melihat grafik rute pemesanan travel.

4.2.5.7. Use Case Scenario Melihat Grafik Laporan Keuangan

Pada tabel 4.11 menjelaskan use case scenario melihat grafik laporan keuangan dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.11 Use Case Scenario Melihat Grafik Laporan Keuangan

MTF_0700 : Melihat Grafik Laporan Keuangan	
Objective	Melihat Grafik Laporan Keuangan
Actor	Operator
Pre-Condition	Telah terautentikasi dan masuk ke dalam sistem.
Main Flow	1. Menekan tombol beranda. 2. Sistem menampilkan grafik laporan keuangan.
Alternative Flow	-
Post-Condition	Berhasil melihat grafik laporan keuangan.

4.2.5.8. Use Case Scenario Menambah Jadwal Travel Harian

Pada tabel 4.12 menjelaskan use case scenario menambah jadwal travel harian dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.12 Use Case Scenario Menambah Jadwal Travel Harian

MTF_0800 : Menambah Jadwal Travel Harian	
Objective	Menambah jadwal travel harian ke dalam database.
Actor	Operator
Pre-Condition	Operator telah berada di form tambah jadwal travel harian
Main Flow	<ol style="list-style-type: none"> 1. Operator mengisi data pada form seperti asal, tujuan, mobil, jam keberangkatan, harga dan hari keberangkatan. 2. Menekan tombol Simpan. 3. Sistem mengambil data-data yang diisikan dan menyimpannya ke dalam database. 4. Sistem menampilkan halaman jadwal travel harian.
Alternative Flow	3a. Jika salah satu dari field tidak diisi maka sistem akan menampilkan peringatan "Harap isi bidang ini" dan kembali pada main flow nomor 1.
Post-Condition	Data jadwal travel harian telah berhasil ditambahkan ke dalam database.

4.2.5.9. Use Case Scenario Melihat Jadwal Travel Harian

Pada tabel 4.13 menjelaskan use case scenario melihat jadwal travel harian dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.13 Use Case Scenario Melihat Jadwal Travel Harian

MTF_0900 : Melihat Jadwal Travel Harian	
Objective	Melihat jadwal travel harian yang ada di database.
Actor	Operator
Pre-Condition	Telah terautentikasi dan masuk ke dalam sistem.
Main Flow	<ol style="list-style-type: none"> 1. Operator menekan tombol jadwal travel harian. 2. Sistem mengambil data dari database. 3. Sistem menampilkan data jadwal travel harian.
Alternative Flow	-
Post-Condition	Berhasil melihat jadwal travel harian yang ada di dalam database.

4.2.5.10. Use Case Scenario Mengubah Jadwal Travel Harian

Pada tabel 4.14 menjelaskan use case scenario mengubah jadwal travel harian dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.14 Use Case Scenario Mengubah Jadwal Travel Harian

MTF_1000: Mengubah Jadwal Travel Harian	
Objective	Mengubah jadwal travel harian yang ada di dalam database.
Actor	Operator
Pre-Condition	Operator telah berada di form ubah jadwal travel harian
Main Flow	<ol style="list-style-type: none"> 1. Operator mengubah data yang perlu diubah pada form seperti data asal, tujuan, mobil, jam keberangkatan, harga dan hari keberangkatan. 2. Menekan tombol Simpan. 3. Sistem mengambil data-data yang diisikan dan menyimpannya ke dalam database. 4. Sistem menampilkan halaman jadwal travel.
Alternative Flow	3a. Jika salah satu dari field tidak diisi maka sistem akan menampilkan peringatan "Harap isi bidang ini" dan kembali pada main flow nomor 1.
Post-Condition	Data jadwal travel harian telah berhasil diubah dan disimpan ke dalam database.

4.2.5.11. Use Case Scenario Menghapus Jadwal Travel Harian

Pada tabel 4.15 menjelaskan use case scenario menghapus jadwal travel harian dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.15 Use Case Scenario Menghapus Jadwal Travel Harian

MTF_1100: Menghapus Jadwal Travel Harian	
Objective	Menghapus jadwal travel harian yang ada di dalam database.
Actor	Operator
Pre-Condition	Operator telah berada di halaman jadwal travel harian
Main Flow	<ol style="list-style-type: none"> 1. Operator menekan tombol Delete. 2. Sistem menghapus jadwal travel.

Tabel 4.15 Use Case Scenario Menghapus Jadwal Travel Harian

MTF_1100: Menghapus Jadwal Travel Harian	
	3. Sistem menampilkan halaman jadwal travel harian.
Alternative Flow	-
Post-Condition	Data jadwal travel harian telah berhasil dihapus dari database.

4.2.5.12. Use Case Scenario Menambah Jadwal Travel Tambahan

Pada tabel 4.16 menjelaskan use case scenario menambah jadwal travel tambahan dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.16 Use Case Scenario Menambah Jadwal Travel Tambahan

MTF_1200 : Menambah Jadwal Travel Tambahan	
Objective	Menambah jadwal travel tambahan ke dalam database.
Actor	Operator
Pre-Condition	Operator telah berada di form tambah jadwal travel tambahan
Main Flow	<ol style="list-style-type: none"> 1. Operator mengisi tanggal keberangkatan. 2. Menekan tombol Pilih. 3. Sistem akan memfilter mobil yang belum dipilih pada jadwal harian berdasarkan tanggal yang telah dipilih lalu menampilkannya pada field mobil pada form jadwal travel. 4. Operator mengisi data pada form seperti asal, tujuan, mobil, jam keberangkatan dan harga. 5. Menekan tombol Simpan. 6. Sistem mengambil data-data yang diisikan dan menyimpannya ke dalam database. 7. Sistem menampilkan halaman jadwal travel tambahan.
Alternative Flow	6a. Jika salah satu dari field tidak diisi maka sistem akan menampilkan peringatan "Harap isi bidang ini" dan kembali pada main flow nomor 4.
Post-Condition	Data jadwal travel tambahan telah berhasil ditambahkan ke dalam database.

4.2.5.13. Use Case Scenario Melihat Jadwal Travel Tambahan

Pada tabel 4.17 menjelaskan use case scenario melihat jadwal travel tambahan dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.17 Use Case Scenario Melihat Jadwal Travel Tambahan

MTF_1300 : Melihat Jadwal Travel Tambahan	
Objective	Melihat jadwal travel tambahan yang ada di database.
Actor	Operator
Pre-Condition	Telah terautentikasi dan masuk ke dalam sistem.
Main Flow	<ol style="list-style-type: none"> 1. Operator menekan tombol jadwal travel tambahan. 2. Sistem mengambil data dari database. 3. Sistem menampilkan data jadwal travel tambahan.
Alternative Flow	-
Post-Condition	Berhasil melihat jadwal travel tambahan yang ada di dalam database.

4.2.5.14. Use Case Scenario Mengubah Jadwal Travel Tambahan

Pada tabel 4.18 menjelaskan use case scenario mengubah jadwal travel tambahan dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.18 Use Case Scenario Mengubah Jadwal Travel Tambahan

MTF_1400: Mengubah Jadwal Travel Harian	
Objective	Mengubah jadwal travel tambahan yang ada di dalam database.
Actor	Operator
Pre-Condition	Operator telah berada di form ubah jadwal travel tambahan

Tabel 4.18 Use Case Scenario Mengubah Jadwal Travel Tambahan

MTF_1400: Mengubah Jadwal Travel Harian	
Main Flow	<ol style="list-style-type: none"> 1. Operator mengubah data yang perlu diubah pada form seperti data asal, tujuan, mobil, jam keberangkatan dan harga. 2. Menekan tombol Simpan. 3. Sistem mengambil data-data yang diisikan dan menyimpannya ke dalam database. 4. Sistem menampilkan halaman jadwal travel tambahan.
Alternative Flow	3b. Jika salah satu dari field tidak diisi maka sistem akan menampilkan peringatan “Harap isi bidang ini” dan kembali pada main flow nomor 1.
Post-Condition	Data jadwal travel tambahan telah berhasil diubah dan disimpan ke dalam database.

4.2.5.15. Use Case Scenario Menghapus Jadwal Travel Tambahan

Pada tabel 4.19 menjelaskan use case scenario menghapus jadwal travel tambahan dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.19 Use Case Scenario Menghapus Jadwal Travel Tambahan

MTF_1500: Menghapus Jadwal Travel Tambahan	
Objective	Menghapus jadwal travel tambahan yang ada di dalam database.
Actor	Operator
Pre-Condition	Operator telah berada di halaman jadwal travel tambahan
Main Flow	<ol style="list-style-type: none"> 1. Operator menekan tombol Delete. 2. Sistem menghapus jadwal travel tambahan. 3. Sistem menampilkan halaman jadwal travel tambahan.
Alternative Flow	-
Post-Condition	Data jadwal travel tambahan telah berhasil dihapus dari database.

4.2.5.16. Use Case Scenario Menambah Pemesanan Travel

Pada tabel 4.20 menjelaskan use case scenario menambah pemesanan travel dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.20 Use Case Scenario Menambah Pemesanan travel

MTF_1600 : Menambah Pemesanan Travel	
Objective	Menambahkan pemesanan travel ke dalam database.
Actor	Operator
Pre-Condition	Telah berada di dalam form pemesanan travel
Main Flow	<ol style="list-style-type: none"> 1. Memilih kota asal, kota tujuan dan tanggal keberangkatan 2. Menekan tombol Cari Jadwal. 3. Kota asal, kota tujuan dan tanggal keberangkatan akan dicari jadwalnya ke dalam database kemudian ditampilkan ke halaman pemesanan travel. 4. Menekan tombol Pesan pada jadwal yang diinginkan. 5. Mengisi data nama lengkap, no telpon, alamat asal, alamat tujuan dan memilih seat yang diinginkan. 6. Menekan tombol Pesan 7. Sistem akan menyimpan form yang telah diisi ke dalam database. 8. Kembali ke halaman utama.
Alternative Flow	<ol style="list-style-type: none"> 3a. Jika terdapat salah satu field yang tidak diisi atau kosong maka sistem akan menampilkan peringatan "Pilih item pada daftar" dan kembali pada main flow nomor 1. 3b. Jika jadwal tidak ditemukan di dalam database maka akan muncul peringatan "Jadwal tidak ditemukan" dan kembali pada main flow nomor 1. 7a. Jika terdapat field yang masih kosong maka sistem akan menampilkan peringatan "Harap isi bidang ini" dan kembali pada main flow nomor 5.
Post-Condition	Pemesanan travel telah berhasil ditambahkan ke dalam database.

4.2.5.17. Use Case Scenario Mencari Jadwal Tersedia

Pada tabel 4.21 menjelaskan use case scenario mencari jadwal tersedia dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.21 Use Case Scenario Mencari Jadwal Tersedia

MTF_1700: Mencari Jadwal Tersedia	
Objective	Mencari jadwal yang tersedia
Actor	Operator
Pre-Condition	Berada di halaman pemesanan travel
Main Flow	<ol style="list-style-type: none"> 1. Memilih kota asal, kota tujuan dan tanggal keberangkatan 2. Menekan tombol Cari Jadwal. 3. Kota asal, kota tujuan dan tanggal keberangkatan akan dicari jadwalnya ke dalam database kemudian ditampilkan ke halaman pemesanan travel.
Alternative Flow	-
Post-Condition	Berhasil melihat jadwal travel sesuai dengan kota asal, kota tujuan dan tanggal keberangkatan yang dicari

4.2.5.18. Use Case Scenario Melihat Pemesanan Travel

Pada tabel 4.22 menjelaskan use case scenario melihat pemesanan travel dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.22 Use Case Scenario Melihat Pemesanan Travel

MTF_1800: Melihat Data Pemesanan Travel	
Objective	Melihat data pemesanan travel yang ada di database.
Actor	Operator
Pre-Condition	Operator telah terautentikasi dan masuk ke dalam sistem.
Main Flow	<ol style="list-style-type: none"> 1. Operator menekan tombol Pemesanan Travel. 2. Sistem mengambil data pemesanan travel dari database. 3. Sistem menampilkan data pemesanan travel.
Alternative Flow	-
Post-Condition	Berhasil melihat data pemesanan travel yang ada di dalam database.

4.2.5.19. Use Case Scenario Mengubah Pemesanan Travel

Pada tabel 4.23 menjelaskan use case scenario mengubah pemesanan travel dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.23 Use Case Scenario Mengubah Pemesanan travel

MTF_1900 : Mengubah Pemesanan Travel	
Objective	Mengubah data pemesanan travel dari database.
Actor	Operator
Pre-Condition	Telah berada di dalam form perubahan travel
Main Flow	<ol style="list-style-type: none"> 1. Mengubah data yang perlu diubah seperti nama lengkap, telepon, alamat asal, alamat tujuan dan seat yang diinginkan. 2. Menekan tombol Simpan. 3. Sistem akan menyimpan form yang telah diubah ke dalam database. 4. Kembali ke halaman utama.
Alternative Flow	3a. Jika terdapat field yang tidak diisi dan kosong maka sistem akan menampilkan peringatan “Mohon lengkapi data terlebih dahulu” dan kembali pada main flow nomor 1.
Post-Condition	Pengubahan data pemesanan travel telah berhasil disimpan ke dalam database.

4.2.5.20. Use Case Scenario Menghapus Pemesanan Travel

Pada tabel 4.24 menjelaskan use case scenario menghapus pemesanan travel dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.24 Use Case Scenario Menghapus Pemesanan Travel

MTF_2000 : Menghapus Pemesanan Travel	
Objective	Menghapus data pemesanan travel dari database.
Actor	Operator

Tabel 4.24 Use Case Scenario Menghapus Pemesanan Travel

MTF_2000 : Menghapus Pemesanan Travel	
Pre-Condition	Operator berada di halaman pemesanan travel.
Main Flow	<ol style="list-style-type: none"> 1. Operator menekan tombol Delete. 2. Sistem menghapus data pemesanan travel dari database. 3. Sistem menampilkan halaman pemesanan travel.
Alternative Flow	-
Post-Condition	Data pemesanan travel telah berhasil dihapus dari database.

4.2.5.21. Use Case Scenario Melihat Manajemen Pemesanan

Pada tabel 4.25 menjelaskan use case scenario melihat manajemen pemesanan dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.25 Use Case Scenario Melihat Manajemen Pemesanan

MTF_2100 : Melihat Manajemen Pemesanan	
Objective	Melihat manajemen pemesanan yang ada di dalam database.
Actor	Operator
Pre-Condition	Operator berada di dalam halaman pemesanan travel.
Main Flow	<ol style="list-style-type: none"> 1. Operator menekan tombol Manajemen Pemesanan. 2. Sistem mengambil data manajemen pemesanan dari database. 3. Sistem menampilkan data manajemen pemesanan.
Alternative Flow	-
Post-Condition	Data manajemen pemesanan telah berhasil ditampilkan.

4.2.5.22. Use Case Scenario Mengubah Driver

Pada tabel 4.26 menjelaskan use case scenario mengubah driver dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.26 Use Case Scenario Mengubah Driver

MTF_2200: Mengubah Driver	
Objective	Mengubah driver pada mobil yang ada di dalam database.
Actor	Operator
Pre-Condition	Operator telah berada di halaman manajemen pemesanan
Main Flow	<ol style="list-style-type: none"> 1. Operator menekan tombol Driver. 2. Operator memilih salah satu driver dari dropdown yang tersedia. 3. Operator menekan tombol Save Changes 4. Sistem mengambil data driver yang dipilih dan menyimpannya ke dalam database. 5. Sistem menampilkan halaman manajemen pemesanan.
Alternative Flow	4a. Jika field tidak diisi maka akan muncul peringatan “Pilih item pada daftar” dan kembali ke main flow 2.
Post-Condition	Data driver telah berhasil diubah dan disimpan ke dalam database.

4.2.5.23. Use Case Scenario Mencetak Surat Jalan

Pada tabel 4.27 menjelaskan use case scenario mencetak surat jalan dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.27 Use Case Scenario Mencetak Surat Jalan

MTF_2300 : Mencetak Surat Jalan	
Objective	Mencetak surat jalan.
Actor	Operator
Pre-Condition	Operator telah berada di halaman manajemen pemesanan.
Main Flow	<ol style="list-style-type: none"> 1. Operator menekan tombol Print Surat Jalan. 2. Sistem akan mengambil data yang dibutuhkan dari database untuk kemudian ditampilkan di dalam halaman cetak surat jalan. 3. Operator menekan tombol dengan icon print. 4. Operator menekan tombol Cetak.
Alternative Flow	-
Post-Condition	Operator berhasil mencetak surat jalan.

4.2.5.24. Use Case Scenario Melihat Daftar Penumpang

Pada tabel 4.28 menjelaskan use case scenario melihat daftar penumpang dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.28 Use Case Scenario Melihat Daftar Penumpang

MTF_2400 : Melihat Daftar Penumpang	
Objective	Melihat daftar penumpang pada mobil tertentu yang ada di dalam database.
Actor	Operator
Pre-Condition	Operator berada di dalam halaman manajemen pemesanan.
Main Flow	<ol style="list-style-type: none"> 1. Operator menekan tombol Daftar Penumpang. 2. Sistem mengambil data daftar penumpang terpilih dari database. 3. Sistem menampilkan data daftar penumpang.
Alternative Flow	-
Post-Condition	Data daftar penumpang telah berhasil ditampilkan.

4.2.5.25. Use Case Scenario Mencetak Tiket Travel

Pada tabel 4.29 menjelaskan use case scenario mencetak tiket travel dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.29 Use Case Scenario Mencetak Tiket Travel

MTF_2500 : Mencetak Status Keberangkatan	
Objective	Mencetak tiket travel penumpang.
Actor	Operator
Pre-Condition	Operator telah berada di halaman daftar penumpang.
Main Flow	<ol style="list-style-type: none"> 1. Operator menekan tombol Print Tiket. 2. Sistem akan mengambil data yang dibutuhkan dari database untuk kemudian ditampilkan pada halaman cetak tiket travel penumpang. 3. Operator menekan tombol dengan icon print. 4. Operator menekan tombol Cetak.
Alternative Flow	-

Tabel 4.29 Use Case Scenario Mencetak Tiket Travel

MTF_2500 : Mencetak Status Keberangkatan	
Post-Condition	Operator berhasil mencetak tiket travel penumpang.

4.2.5.26. Use Case Scenario Menandai Transaksi Selesai

Pada tabel 4.30 menjelaskan use case scenario mengubah status keberangkatan dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.30 Use Case Scenario Mengubah Status Keberangkatan

MTF_2600 : Mengubah Status Keberangkatan	
Objective	Mengubah status keberangkatan pada mobil yang ada di dalam database.
Actor	Operator
Pre-Condition	Operator telah berada di halaman manajemen pemesanan
Main Flow	<ol style="list-style-type: none"> 1. Operator menekan tombol Status Keberangkatan. 2. Operator memilih salah satu status keberangkatan dari dropdown yang tersedia. 3. Operator menekan tombol Save Changes 4. Sistem mengambil data status keberangkatan yang dipilih dan menyimpannya ke dalam database. 5. Sistem menampilkan halaman manajemen pemesanan.
Alternative Flow	4a. Jika field tidak diisi maka akan muncul peringatan "Pilih item pada daftar" dan kembali ke main flow 2.
Post-Condition	Data status keberangkatan telah berhasil diubah dan disimpan ke dalam database.

4.2.5.27. Use Case Scenario Menambah Data Mobil

Pada tabel 4.31 menjelaskan use case scenario menambah data mobil dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.31 Use Case Scenario Menambah Data Mobil

MTF_2700: Menambah Data Mobil	
Objective	Menambah data mobil ke dalam database.
Actor	Operator

Tabel 4.31 Use Case Scenario Menambah Data Mobil

MTF_2700: Menambah Data Mobil	
Pre-Condition	Operator telah berada di form tambah data mobil.
Main Flow	<ol style="list-style-type: none"> 1. Operator mengisi data pada form seperti data nama mobil, jumlah seat, warna dan nomor kendaraan. 2. Menekan tombol Simpan. 3. Sistem mengambil data-data yang diisikan dan menyimpannya ke dalam database. 4. Sistem menampilkan halaman data mobil.
Alternative Flow	3a. Jika terdapat field yang masih kosong maka sistem akan memberikan peringatan "Harap isi bidang ini".
Post-Condition	Data mobil telah berhasil ditambahkan ke dalam database.

4.2.5.28. Use Case Scenario Melihat Data Mobil

Pada tabel 4.32 menjelaskan use case scenario melihat data mobil dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.32 Use Case Scenario Melihat Data Mobil

MTF_2800: Melihat Data Mobil	
Objective	Melihat data mobil yang ada di dalam database.
Actor	Operator
Pre-Condition	Operator telah terautentikasi dan masuk ke dalam sistem.
Main Flow	<ol style="list-style-type: none"> 1. Operator menekan tombol Daftar Mobil. 2. Sistem mengambil data mobil dari database 3. Sistem menampilkan data mobil.
Alternative Flow	-
Post-Condition	Data mobil telah berhasil ditampilkan.

4.2.5.29. Use Case Scenario Mengubah Data Mobil

Pada tabel 4.33 menjelaskan use case scenario mengubah data mobil dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.33 Use Case Scenario Mengubah Data Mobil

MTF_2900: Mengubah Data Mobil	
Objective	Mengubah data mobil yang ada di dalam database.
Actor	Operator
Pre-Condition	Operator telah berada di form ubah data mobil
Main Flow	<ol style="list-style-type: none"> 1. Operator mengubah data yang perlu diubah seperti nama mobil, jumlah seat, warna dan nomor kendaraan. 2. Menekan tombol Simpan. 3. Sistem mengambil data-data yang diisikan dan menyimpannya ke dalam database. 4. Sistem menampilkan halaman data mobil.
Alternative Flow	3a. Jika terdapat field yang masih kosong maka sistem akan memberikan peringatan “Harap isi bidang ini”.
Post-Condition	Data mobil telah berhasil diubah dan disimpan ke dalam database.

4.2.5.30. Use Case Scenario Menghapus Data Mobil

Pada tabel 4.34 menjelaskan use case scenario menghapus data mobil dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.34 Use Case Scenario Menghapus Data Mobil

MTF_3000 : Menghapus Data Mobil	
Objective	Menghapus data mobil yang ada di dalam database.
Actor	Operator
Pre-Condition	Operator telah berada di halaman data mobil
Main Flow	<ol style="list-style-type: none"> 1. Operator menekan tombol Delete. 2. Sistem menghapus data mobil. 3. Sistem menampilkan halaman data mobil.
Alternative Flow	-
Post-Condition	Data mobil telah berhasil dihapus dari database.

4.2.5.31. Use Case Scenario Menambah Data Driver

Pada tabel 4.35 menjelaskan use case scenario menambah data driver dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.35 Use Case Scenario Menambah Data Driver

MTF_3100: Menambah Data Driver	
Objective	Menambah data driver ke dalam database.
Actor	Operator
Pre-Condition	Operator telah berada di form tambah data driver.
Main Flow	<ol style="list-style-type: none"> 1. Operator mengisi data pada form seperti nama driver, no telpon dan alamat. 2. Menekan tombol Simpan. 3. Sistem mengambil data-data yang diisikan dan menyimpannya ke dalam database. 4. Sistem menampilkan halaman data driver.
Alternative Flow	3a. Jika terdapat field yang masih kosong maka sistem akan memberikan peringatan "Harap isi bidang ini".
Post-Condition	Data driver telah berhasil ditambahkan ke dalam database.

4.2.5.32. Use Case Scenario Melihat Data Driver

Pada tabel 4.36 menjelaskan use case scenario melihat data driver dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.36 Use Case Scenario Melihat Data Driver

MTF_3200 : Melihat Data Driver	
Objective	Melihat data driver yang ada di dalam database.
Actor	Operator
Pre-Condition	Operator telah terautentikasi dan masuk ke dalam sistem.
Main Flow	<ol style="list-style-type: none"> 1. Operator menekan tombol Daftar Driver. 2. Sistem mengambil data driver dari database. 3. Sistem menampilkan data driver.
Alternative Flow	-
Post-Condition	Data driver telah berhasil ditampilkan.

4.2.5.33. Use Case Scenario Mengubah Data Driver

Pada tabel 4.37 menjelaskan use case scenario mengubah data driver dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.37 Use Case Scenario Mengubah Data Driver

MTF_3300: Mengubah Data Driver	
Objective	Mengubah data driver yang ada di dalam database.
Actor	Operator
Pre-Condition	Operator telah berada di form ubah data driver
Main Flow	<ol style="list-style-type: none"> 1. Operator mengubah data yang perlu diubah di dalam form seperti nama driver, no telpon dan alamat. 2. Menekan tombol Simpan. 3. Sistem mengambil data-data yang diisikan dan menyimpannya ke dalam database. 4. Sistem menampilkan halaman data driver.
Alternative Flow	3a. Jika terdapat field yang masih kosong maka sistem akan memberikan peringatan "Harap isi bidang ini".
Post-Condition	Data driver telah berhasil diubah dan disimpan ke dalam database.

4.2.5.34. Use Case Scenario Menghapus Data Driver

Pada tabel 4.38 menjelaskan use case scenario menghapus data driver dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.38 Use Case Scenario Menghapus Data Driver

MTF_3400: Menghapus Data Driver	
Objective	Menghapus data driver dari dalam database.
Actor	Operator
Pre-Condition	Operator telah berada di halaman data driver
Main Flow	<ol style="list-style-type: none"> 1. Operator menekan tombol Delete. 2. Sistem menghapus data driver. 3. Sistem menampilkan halaman data driver.
Alternative Flow	-

Tabel 4.38 Use Case Scenario Menghapus Data Driver

MTF_3400: Menghapus Data Driver	
Post-Condition	Data driver telah berhasil dihapus dari database.

4.2.5.35. Use Case Scenario Menambah Data Kota

Pada tabel 4.39 menjelaskan use case scenario menambah data kota dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.39 Use Case Scenario Menambah Data Kota

MTF_3500 : Menambah Data Kota	
Objective	Menambah data kota ke dalam database.
Actor	Operator
Pre-Condition	Operator telah berada di form tambah data kota.
Main Flow	<ol style="list-style-type: none"> 1. Operator mengisi data pada form seperti nama kota. 2. Menekan tombol Simpan. 3. Sistem mengambil data-data yang diisikan dan menyimpannya ke dalam database. 4. Sistem menampilkan halaman data kota.
Alternative Flow	3a. Jika terdapat field yang masih kosong maka sistem akan memberikan peringatan "Harap isi bidang ini".
Post-Condition	Data kota telah berhasil ditambahkan ke dalam database.

4.2.5.36. Use Case Scenario Melihat Data Kota

Pada tabel 4.40 menjelaskan use case scenario melihat data kota dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.40 Use Case Scenario Melihat Data Kota

MTF_3600 : Melihat Data Kota	
Objective	Melihat data kota yang ada di dalam database.
Actor	Operator
Pre-Condition	Operator telah terautentikasi dan masuk ke dalam sistem.
Main Flow	<ol style="list-style-type: none"> 1. Operator menekan tombol Daftar Kota. 2. Sistem mengambil data kota dari database.

Tabel 4.40 Use Case Scenario Melihat Data Kota

MTF_3600 : Melihat Data Kota	
	3. Sistem menampilkan data kota.
Alternative Flow	-
Post-Condition	Data kota telah berhasil ditampilkan.

4.2.5.37. Use Case Scenario Mengubah Data Kota

Pada tabel 4.41 menjelaskan use case scenario mengubah data kota dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.41 Use Case Scenario Mengubah Data Kota

MTF_3700: Mengubah Data Kota	
Objective	Mengubah data kota yang ada di dalam database.
Actor	Operator
Pre-Condition	Operator telah berada di form ubah data kota
Main Flow	<ol style="list-style-type: none"> 1. Operator mengubah data yang perlu diubah di dalam form seperti nama kota. 2. Menekan tombol Simpan. 3. Sistem mengambil data-data yang diisikan dan menyimpannya ke dalam database. 4. Sistem menampilkan halaman data kota.
Alternative Flow	3a. Jika terdapat field yang masih kosong maka sistem akan memberikan peringatan "Harap isi bidang ini".
Post-Condition	Data kota telah berhasil diubah dan disimpan ke dalam database.

4.2.5.38. Use Case Scenario Menghapus Data Kota

Pada tabel 4.42 menjelaskan use case scenario menghapus data kota dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.42 Use Case Scenario Menghapus Data Kota

MTF_3800: Menghapus Data Kota	
Objective	Menghapus data kota dari dalam database.
Actor	Operator

Tabel 4.42 Use Case Scenario Menghapus Data Kota

MTF_3800: Menghapus Data Kota	
Pre-Condition	Operator telah berada di halaman data kota
Main Flow	<ol style="list-style-type: none"> 1. Operator menekan tombol Delete. 2. Sistem menghapus data kota. 3. Sistem menampilkan halaman data kota.
Alternative Flow	-
Post-Condition	Data kota telah berhasil dihapus dari database.

4.2.5.39. Use Case Scenario Melihat Gaji Driver

Pada tabel 4.43 menjelaskan use case scenario melihat gaji driver dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.43 Use Case Scenario Melihat Gaji Driver

MTF_3900 : Melihat Gaji Driver	
Objective	Melihat gaji driver.
Actor	Operator
Pre-Condition	Operator telah terautentikasi dan masuk ke dalam sistem.
Main Flow	<ol style="list-style-type: none"> 1. Operator menekan tombol Gaji Driver. 2. Sistem mengambil data pemesanan yang telah terjadi dari database dan menghitung gaji driver pada setiap perjalanan yang sudah selesai. 3. Sistem menampilkan data gaji driver.
Alternative Flow	-
Post-Condition	Gaji driver telah berhasil ditampilkan.

4.2.5.40. Use Case Scenario Mencetak Gaji Driver

Pada tabel 4.44 menjelaskan use case scenario mencetak gaji driver dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.44 Use Case Scenario Mencetak Gaji Supir

MTF_4000: Mencetak Gaji Supir	
Objective	Mencetak gaji driver yang sedang ditampilkan berdasarkan bulan.
Actor	Operator
Pre-Condition	Operator berada di halaman gaji supir.
Main Flow	<ol style="list-style-type: none"> 1. Operator menekan tombol Cetak 2. Sistem menampilkan preview cetak gaji supir 3. Operator menekan tombol cetak dan gaji supir tercetak
Alternative Flow	-
Post-Condition	Gaji supir berhasil dicetak.

4.2.5.41. Use Case Scenario Mencari Gaji Driver

Pada tabel 4.45 menjelaskan use case scenario mencari gaji driver dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.45 Use Case Scenario Mencari Gaji Supir

MTF_4100: Mencari Gaji Supir	
Objective	Mencari gaji driver yang sedang ditampilkan berdasarkan bulan dan tahun.
Actor	Operator
Pre-Condition	Operator berada di halaman gaji supir.
Main Flow	<ol style="list-style-type: none"> 1. Operator memilih bulan dan tahun. 2. Operator menekan tombol dengan ikon kaca pembesar. 3. Sistem menampilkan gaji supir berdasarkan bulan dan tahun terpilih.
Alternative Flow	-
Post-Condition	Berhasil menampilkan gaji driver berdasarkan bulan dan tahun terpilih.

4.2.5.42. Use Case Scenario Melihat Laporan Keuangan Pemesanan Travel

Pada tabel 4.46 menjelaskan use case scenario melihat laporan keuangan pemesanan travel dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal

yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.46 Use Case Scenario Melihat Laporan Keuangan Pemesanan Travel

MTF_4200: Laporan Keuangan Melihat Pemesanan Travel	
Objective	Melihat laporan keuangan pemesanan travel.
Actor	Operator
Pre-Condition	Operator telah terautentikasi dan masuk ke dalam sistem.
Main Flow	<ol style="list-style-type: none"> 1. Operator menekan tombol Laporan Keuangan Pemesanan Travel. 2. Operator memilih bulan dan tahun dalam fungsi dropdown yang tersedia untuk memilih laporan pemesanan travel yang ingin ditampilkan. 3. Menekan tombol Cari 4. Sistem mencari ke dalam database mengenai pemesanan yang sudah terjadi dan gaji driver. 5. Sistem menampilkan data laporan pemesanan travel.
Alternative Flow	5a. Jika sistem tidak menemukan laporan dari database, maka sistem akan memberikan peringatan "Laporan tidak tersedia" dan kembali ke main flow 2.
Post-Condition	Laporan keuangan pemesanan travel telah berhasil ditampilkan.

4.2.5.43. Use Case Scenario Menambah Laporan Keuangan Pemesanan Travel

Pada tabel 4.47 menjelaskan use case scenario menambah laporan keuangan pemesanan travel dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.47 Use Case Scenario Menambah Laporan Pemesanan Keuangan Travel

MTF_4300 : Menambah Laporan Keuangan Pemesanan Travel	
Objective	Menambah laporan keuangan pemesanan travel ke dalam database.
Actor	Operator

Tabel 4.47 Use Case Scenario Menambah Laporan Pemesanan Keuangan Travel

MTF_4300 : Menambah Laporan Keuangan Pemesanan Travel	
Pre-Condition	Operator telah berada di form tambah laporan keuangan pemesanan travel.
Main Flow	<ol style="list-style-type: none"> 1. Operator mengisi data pada form seperti total uang, pengurangan total uang, penambahan total uang, keterangan total uang, gaji driver, penambahan gaji driver, pengurangan gaji driver, keterangan gaji driver. 2. Menekan tombol Simpan. 3. Sistem mengambil data-data yang diisikan dan menyimpannya ke dalam database. 4. Sistem menampilkan halaman laporan keuangan pemesanan travel.
Alternative Flow	3b. Jika terdapat field yang masih kosong maka sistem akan memberikan peringatan “Harap isi bidang ini”.
Post-Condition	Data laporan keuangan pemesanan travel telah berhasil ditambahkan ke dalam database.

4.2.5.44. Use Case Scenario Mengubah Laporan Keuangan Pemesanan Travel

Pada tabel 4.48 menjelaskan use case scenario mengubah laporan keuangan pemesanan travel dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.48 Use Case Scenario Mengubah Laporan Keuangan Pemesanan Travel

MTF_4400: Mengubah Laporan Keuangan Pemesanan Travel	
Objective	Mengubah laporan keuangan pemesanan travel yang ada di dalam database.
Actor	Operator
Pre-Condition	Operator telah berada di form ubah data kota
Main Flow	<ol style="list-style-type: none"> 1. Operator mengubah data yang perlu diubah di dalam form seperti total uang, pengurangan total uang, penambahan total uang, keterangan total uang, gaji driver, penambahan gaji driver, pengurangan gaji driver, keterangan gaji driver.

Tabel 4.48 Use Case Scenario Mengubah Laporan Keuangan Pemesanan Travel

MTF_4400: Mengubah Laporan Keuangan Pemesanan Travel	
	<ol style="list-style-type: none"> 2. Menekan tombol Simpan. 3. Sistem mengambil data-data yang diisikan dan menyimpannya ke dalam database. 4. Sistem menampilkan halaman laporan keuangan pemesanan travel.
Alternative Flow	3b. Jika terdapat field yang masih kosong maka sistem akan memberikan peringatan "Harap isi bidang ini".
Post-Condition	Data laporan keuangan pemesanan trave telah berhasil diubah dan disimpan ke dalam database.

4.2.5.45. Use Case Scenario Menghapus Laporan Keuangan Pemesanan Travel

Pada tabel 4.49 menjelaskan use case scenario menghapus laporan keuangan pemesanan travel dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.49 Use Case Scenario Menghapus Laporan Keuangan Pemesanan Travel

MTF_4500: Menghapus Laporan Keuangan Pemesanan Travel	
Objective	Menghapus laporan keuangan pemesanan travel dari dalam database.
Actor	Operator
Pre-Condition	Operator telah berada di halaman laporan keuangan pemesanan travle
Main Flow	<ol style="list-style-type: none"> 1. Operator menekan tombol Delete. 2. Sistem menghapus laporan keuangan pemesanan travel. 3. Sistem menampilkan halaman laporan keuangan pemesanan travel.
Alternative Flow	-
Post-Condition	Data laporan keuangan pemesanan travel telah berhasil dihapus dari database.

4.2.5.46. Use Case Scenario Mencetak Laporan Keuangan Pemesanan Travel

Pada tabel 4.50 menjelaskan use case scenario mencetak laporan keuangan pemesanan travel dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.50 Use Case Scenario Mencetak Laporan Keuangan Pemesanan Travel

MTF_4600: Mencetak Laporan Keuangan Pemesanan Travel	
Objective	Mencetak laporan keuangan pemesanan travel.
Actor	Operator
Pre-Condition	Operator telah berada di halaman laporan pemesanan travel.
Main Flow	<ol style="list-style-type: none"> 1. Operator menekan tombol Print Laporan Keuangan. 2. Sistem mengambil data yang diperlukan dari database. 3. Sistem menampilkan halaman cetak laporan keuangan 4. Operator menekan tombol dengan icon print. 5. Operator menekan tombol Cetak. 6. Laporan Keuangan tercetak.
Alternative Flow	-
Post-Condition	Laporan keuangan pemesanan travel telah berhasil dicetak.

4.2.5.47. Use Case Scenario Mencari Laporan Keuangan Pemesanan Travel

Pada tabel 4.51 menjelaskan use case scenario mencari laporan keuangan pemesanan travel dimulai dari tujuan, aktor yang berperan di dalamnya, kondisi awal yang harus dipenuhi, tugas utama yang dilakukan, tugas alternatif dan kondisi akhir yang harus dipenuhi.

Tabel 4.51 Use Case Scenario Mencari Laporan Keuangan Pemesanan Travel

MTF_4700 : Mencari Laporan Keuangan Pemesanan Travel	
Objective	Mencari laporan keuangan pemesanan travel.
Actor	Operator
Pre-Condition	Operator telah berada di dalam halaman laporan keuangan pemesanan travel.

Tabel 4.51 Use Case Scenario Mencari Laporan Keuangan Pemesanan Travel

MTF_4700 : Mencari Laporan Keuangan Pemesanan Travel	
Main Flow	<ol style="list-style-type: none"> 1. Operator memilih bulan dan tahun dalam fungsi dropdown yang tersedia untuk memilih laporan pemesanan travel yang ingin ditampilkan. 2. Menekan tombol dengan ikon kaca pembesar. 3. Sistem mengambil data yang diperlukan dari database. 4. Sistem menampilkan data laporan keuangan pemesanan travel
Alternative Flow	-
Post-Condition	Laporan keuangan pemesanan travel telah berhasil ditampilkan sesuai dengan bulan dan tahun yang dimasukkan.

4.3. Validasi

Proses validasi dilakukan untuk melihat apakah sistem telah memenuhi semua kebutuhan ataupun harapan dari calon pengguna. Dalam tahap ini akan dipastikan kebutuhan-kebutuhan yang telah didefinisikan sudah disepakati. Proses validasi pada penelitian ini dilakukan oleh operator pemesanan.

4.3.1. Validasi Kebutuhan Fungsional

Validasi kebutuhan fungsional dilakukan dengan cara melakukan wawancara langsung dengan operator yang merupakan calon pengguna sistem yang akan dibangun. Dari proses wawancara tersebut telah disepakati kebutuhan yang akan tersedia di dalam sistem yang akan dibangun dapat dilihat pada tabel 4.52 berikut.

Tabel 4.52 Validasi Kebutuhan Fungsional

Kode	Kebutuhan fungsional	Validasi
MTF_0100	Login	√
MTF_0200	Logout	√
MTF_0300	Melihat Beranda	√
MTF_0400	Mencari Grafik Pemesanan	√
MTF_0500	Melihat Grafik Pemesanan Terbanyak	√
MTF_0600	Melihat Grafik Rute Pemesanan Travel	√

Tabel 4.52 Validasi Kebutuhan Fungsional

Kode	Kebutuhan fungsional	Validasi
MTF_0700	Melihat Grafik Laporan Keuangan	√
MTF_0800	Menambah Jadwal Travel Harian	√
MTF_0900	Melihat Jadwal Travel Harian	√
MTF_1000	Mengubah Jadwal Travel Harian	√
MTF_1100	Menghapus Jadwal Travel Harian	√
MTF_1200	Menambah Jadwal Travel Tambahan	√
MTF_1300	Melihat Jadwal Travel Tambahan	√
MTF_1400	Mengubah Jadwal Travel Tambahan	√
MTF_1500	Menghapus Jadwal Travel Tambahan	√
MTF_1600	Menambah Pemesanan Travel	√
MTF_1700	Mencari Jadwal Tersedia	√
MTF_1800	Melihat Pemesanan Travel	√
MTF_1900	Mengubah Pemesanan Travel	√
MTF_2000	Menghapus Pemesanan Travel	√
MTF_2100	Melihat Manajemen Pemesanan	√
MTF_2200	Mengubah Driver	√
MTF_2300	Mencetak Surat Jalan	√
MTF_2400	Melihat Daftar Penumpang	√
MTF_2500	Mencetak Tiket Travel	√
MTF_2600	Menandai Transaksi Selesai	√
MTF_2700	Menambah Data Mobil	√
MTF_2800	Melihat Data Mobil	√
MTF_2900	Mengubah Data Mobil	√
MTF_3000	Menghapus Data Mobil	√
MTF_3100	Menambah Data Driver	√

Tabel 4.52 Validasi Kebutuhan Fungsional

Kode	Kebutuhan fungsional	Validasi
MTF_3200	Melihat Data Driver	√
MTF_3300	Mengubah Data Driver	√
MTF_3400	Menghapus Data Driver	√
MTF_3500	Menambah Data Kota	√
MTF_3600	Melihat Data Kota	√
MTF_3700	Mengubah Data Kota	√
MTF_3800	Menghapus Data Kota	√
MTF_3900	Melihat Gaji Driver	√
MTF_4000	Mencetak Gaji Driver	√
MTF_4100	Mencari Gaji Driver	√
MTF_4200	Melihat Laporan Keuangan Pemesanan Travel	√
MTF_4300	Menambah Laporan Keuangan Pemesanan Travel	√
MTF_4400	Mengubah Laporan Keuangan Pemesanan Travel	√
MTF_4500	Menghapus Laporan Keuangan Pemesanan Travel	√
MTF_4600	Mencetak Laporan Keuangan Pemesanan Travel	√
MTF_4700	Mencari Laporan Keuangan Pemesanan Travel	√

4.3.2. Validasi Kebutuhan Non Fungsional

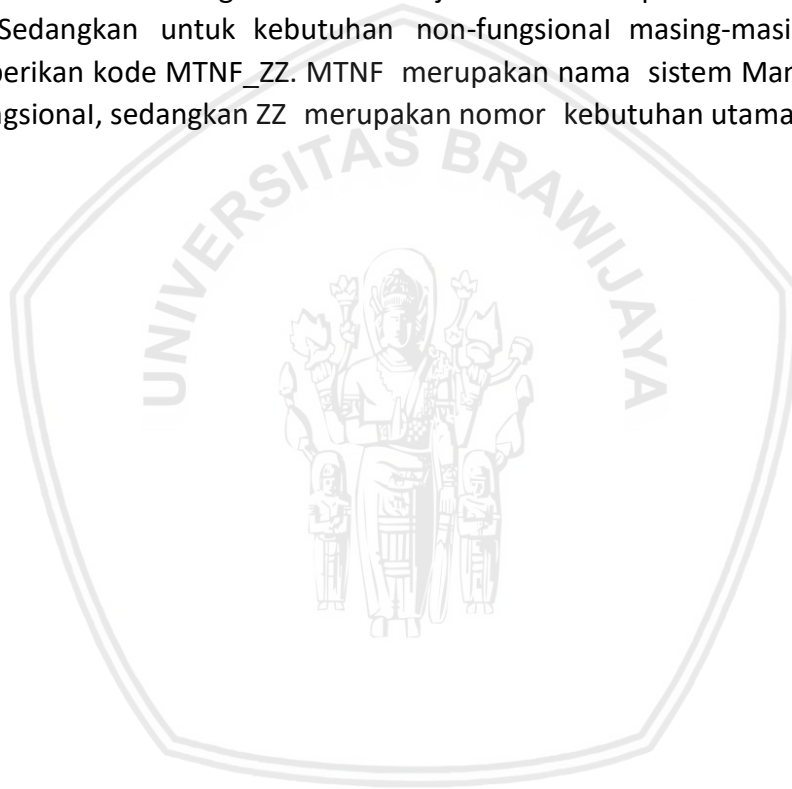
Validasi kebutuhan non fungsional dilakukan dengan melakukan wawancara langsung dengan operator pada Perusahaan Lia Jaya Sentosa. Dari proses wawancara tersebut, telah disepakai kebutuhan non fungsional yang akan tersedia di dalam sistem yang akan dibangun. Dapat dilihat pada tabel 4.53.

Tabel 4.53 Validasi kebutuhan Non Fungsional

Kode	Kebutuhan fungsional	Validasi
MTNF_01	<i>Usability</i>	√

4.4. Manajemen Kebutuhan

Dalam manajemen kebutuhan, dilakukan proses kontrol dengan memberikan nama kode pada masing-masing kebutuhan yang telah dispesifikasi sebelumnya. Untuk kebutuhan fungsional akan diberikan kode MTF_YYZZ. Dimana MTF merupakan nama sistem, Manajemen Travel Fungsional, YY menunjukkan nomor kebutuhan utama sedangkan ZZ menunjukkan nomor spesifikasi dari kebutuhan utama. Sedangkan untuk kebutuhan non-fungsional masing-masing kebutuhan akan diberikan kode MTNF_ZZ. MTNF merupakan nama sistem Manajemen Travel Non Fungsional, sedangkan ZZ merupakan nomor kebutuhan utama.



BAB 5 PERANCANGAN DAN IMPLEMENTASI

Perancangan dan implementasi berisi tentang pemodelan perangkat lunak yang terdiri dari perancangan arsitektur, perancangan komponen, perancangan data dan perancangan antarmuka. Pemodelan ini diperlukan sebagai acuan pengembang untuk mengembangkan sistem agar lebih terarah sehingga nantinya sistem *traceable* dan membantu dapat membantu dalam proses pengembangan lebih lanjut. Selain itu, pada bab implementasi berisi hasil realisasi dari perancangan atau pemodelan yang sudah dilakukan.

5.1 Perancangan

Perancangan atau pemodelan dilakukan untuk membangun model abstrak dari sebuah sistem. Proses perancangan digunakan untuk memodelkan sebuah sistem dalam bentuk *Unified Modelling Language* (UML). Pada tahap perancangan akan dilakukan empat tahap perancangan yang masing-masing dari tahap ini akan memberikan sudut pandang yang berbeda. Empat tahap tersebut adalah perancangan arsitektur, perancangan komponen, perancangan data dan perancangan antarmuka.

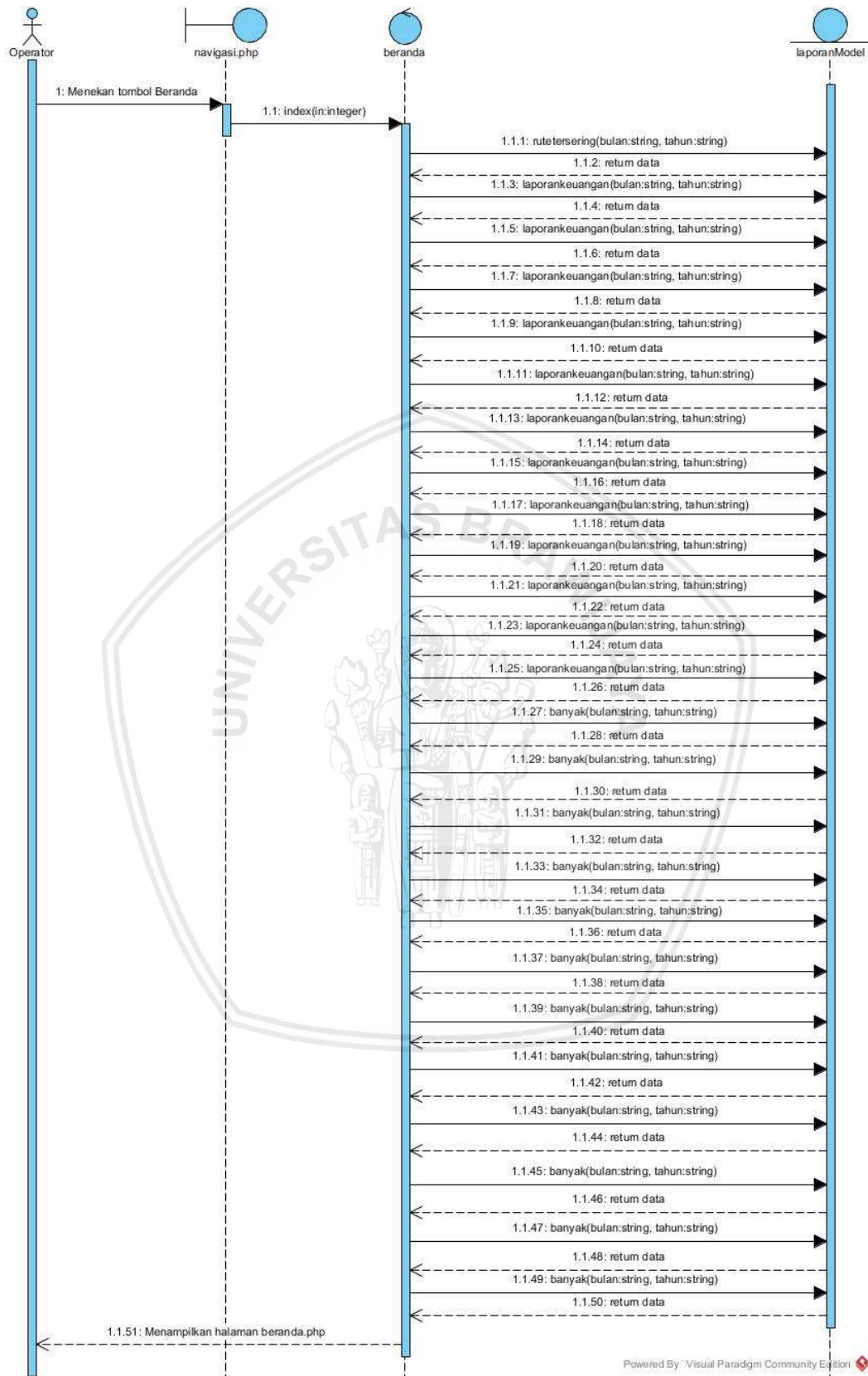
5.1.1 Perancangan Arsitektur

Perancangan arsitektur merupakan perancangan yang menjelaskan hubungan antar objek di dalam sistem. Perancangan arsitektur terdiri dari 2 diagram yaitu *sequence diagram* dan *class diagram*.

5.1.1.1 Sequence Diagram

Sequence Diagram merupakan gambaran dari interaksi antara aktor dengan sistem dan komponen-komponennya (Sommerville, 2011). Pada perancangan *sequence diagram* akan diambil 3 sampel dari sistem yang akan dibangun.

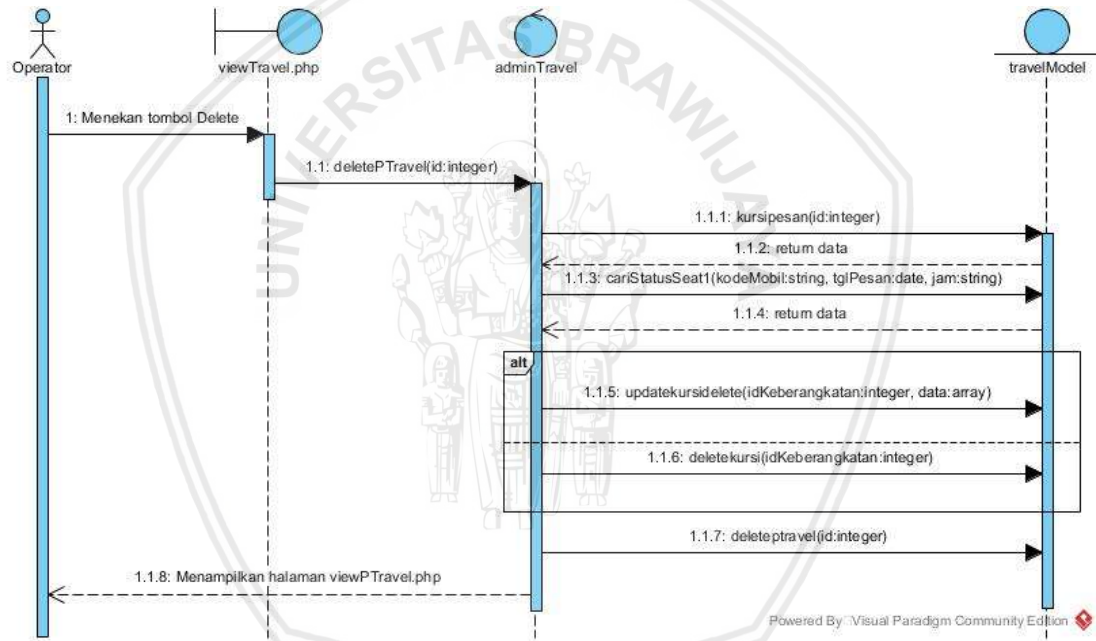
5.1.1.1.1 Sequence Diagram Melihat Beranda



Gambar 5.1 Sequence Diagram Melihat Beranda

Sequence diagram melihat beranda menjelaskan proses ketika operator ingin melihat grafik-grafik yang berhubungan dengan pemesanan travel seperti grafik pemesanan terbanyak, grafik rute pemesanan travel yang paling sering dipesan dan grafik laporan keuangan pemesanan travel pada halaman beranda. Prosesnya dimulai dari operator menekan tombol Beranda pada navigasi bar yaitu navigasi.php kemudian akan menjalankan *method* index() pada controller beranda. Setelah itu, pada controller beranda akan diambil seluruh data yang berkaitan dengan laporan yang akan ditampilkan dari database melalui model laporanModel dengan menggunakan *method* rutetersering(), laporankeuangan(), banyak(). Setelah semua data didapatkan kemudian akan ditampilkan ke halaman beranda.php kembali.

5.1.1.1.2 Sequence Diagram Menghapus Pemesanan Travel

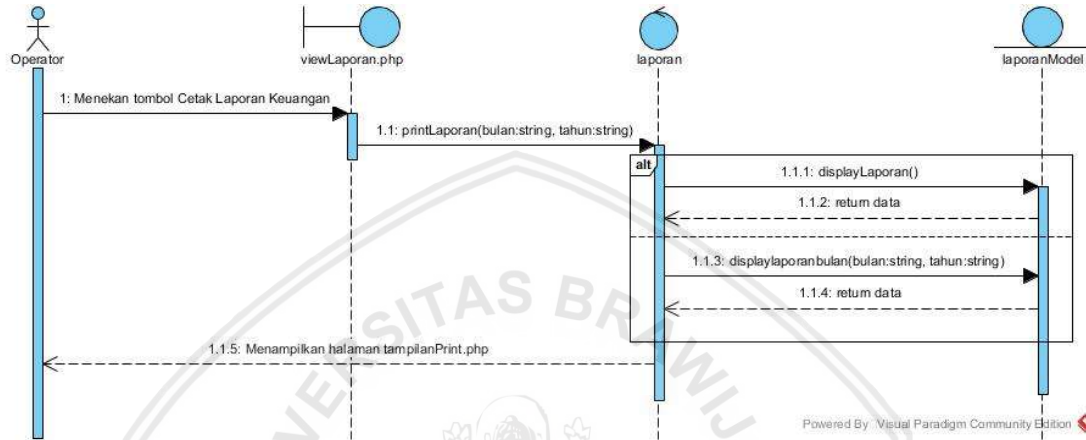


Gambar 5.2 Sequence Diagram Menghapus Pemesanan Travel

Sequence diagram menghapus pemesanan menggambarkan proses penghapusan data pemesanan travel dari database oleh operator. Proses ini dimulai dari operator menekan tombol Delete pada halaman viewTravel.php, kemudian akan dipanggil *method* deletePTravel pada controller adminTravel, kemudian controller adminTravel akan memanggil *method* yang berada pada model travelModel bernama kursipesan() untuk mengambil data pada tabel pemesanan travel. Kemudian, *method* cariStatusSeat1() digunakan untuk mengambil data pada tabel keberangkatan. Selanjutnya, terjadi proses percabangan *if else* yang jika masih terdapat kursi yang

dipakai pada tabel keberangkatan maka data hanya akan diupdate dengan *method* `updatekursidelete()`, sebaliknya jika sudah tidak tersisa kursi pada tabel keberangkatan maka data akan dihapus dari tabel keberangkatan. Setelah itu, data pemesanan travel terpilih dihapus dari database menggunakan *method* `deleteptravel()` dan akan diarahkan ke halaman `viewPTravel.php` kembali.

5.1.1.1.3 Sequence Diagram Mencetak Laporan Keuangan



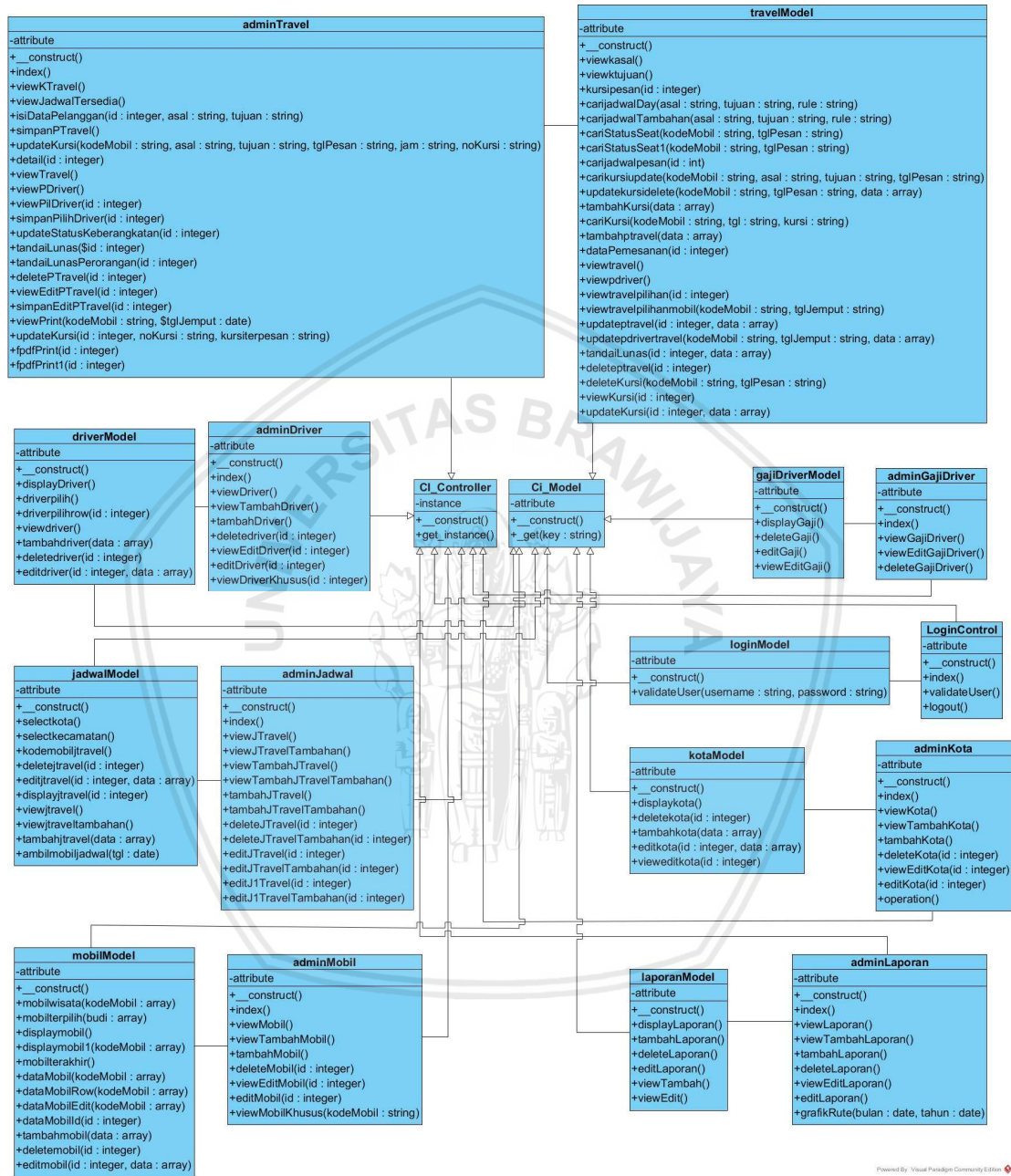
Gambar 5.3 Sequence Diagram Mencetak Laporan Keuangan

Sequence diagram mencetak laporan keuangan menggambarkan proses operator untuk mencetak laporan keuangan. Proses ini dimulai dari operator menekan tombol Cetak Laporan Keuangan pada halaman `viewLaporan.php`, kemudian akan dipanggil *method* yang terdapat pada *controller* laporan yaitu `printLaporan(bulan, tahun)`. Kemudian terdapat percabangan ketika akan memanggil *method* dari *model* laporanModel. Jika nilai parameter bulan pada `printLaporan()` berisi 0, maka akan dipanggil *method* `displayLaporan()` yang mana berarti seluruh laporan yang ada akan ditampilkan pada tampilan print untuk dicetak, sebaliknya nilai parameter bulan adalah selain 0, maka akan dijalankan *method* `displaylaporanbulan` yang mana hanya akan menampilkan dan mencetak laporan keuangan berdasarkan bulan dan tahun yang telah ditentukan sebelumnya. Kemudian operator akan diarahkan ke halaman `tampilanPrint.php` yang selanjutnya dapat dilakukan proses pencetakan laporan keuangan.

5.1.1.2 Class Diagram

Gambar 5.4 menjelaskan *class diagram* yang digunakan dalam pengembangan sistem. Dalam diagram ini terdapat sebuah *controller* yang berfungsi sebagai penghubung antara pengguna dengan sistem. Kemudian terdapat *entity* yang berfungsi sebagai penghubung antara sistem dengan database. Terdapat 9 kelas *controller* yang ada pada sistem ini, antara lain adalah `CI_Controller`, `adminTravel`,

adminLaporan, adminGajiDriver, adminDriver, adminJadwal, LoginControl, adminKota dan adminMobil. Sedangkan kelas entity yang ada di dalam sistem ini juga ada 9 kelas, yaitu CI_Model, travelModel, laporanModel, gajiDriverModel, driverModel, jadwalModel, loginModel, kotaModel dan mobilModel.



Gambar 5.4 Class Diagram

5.1.2 Perancangan Komponen

Perancangan komponen merupakan sebuah kegiatan mengambil setiap komponen yang terdapat di dalam sistem tentang bagaimana sebuah komponen tersebut berjalan (Sommerville, 2011). Pemodelan ini nantinya dapat digunakan untuk developer dalam proses implementasi. Perancangan komponen berisi tentang algoritma-algoritma yang akan diimplementasikan pada kelas-kelas yang akan didefinisikan. Algoritma ini dituliskan dalam bentuk *pseudocode* yang menjelaskan alur atau langkah-langkah dalam menyelesaikan permasalahan dalam kode-kode yang ditulis dengan lebih sederhana menggunakan bahasa yang mudah dipahami secara universal. Dalam perancangan komponen ini, terdapat 3 sampel yang diambil yaitu algoritma menambah pemesanan travel, menambah laporan keuangan dan menambah jadwal.

5.1.2.1 Algoritma Melihat Beranda

Algoritma melihat beranda merupakan algoritma untuk melihat halaman beranda yang didalamnya terdapat grafik. Tabel 5.1 akan menjelaskan algoritma menghapus data pemesanan.

Nama class : Beranda

Nama method : index()

Tabel 5.1 Algoritma Melihat Beranda

No	Pseudocode
1	Inisiasi variable pencarian = ""
2	Inisiasi variable bulan = ""
3	Inisiasi variable tahun = ""
4	Inisiasi array data dengan index status = 0
5	Jika parameter dengan nama in sama dengan 1, start branch
6	Mengambil nilai dari inputan operator dan memberikan nilainya ke variable pencarian
7	Mengubah nilai dari variable pencarian ke dalam array pecahpencarian berdasarkan karakter "-"
8	Memberikan nilai ke dalam variable bulan berdasarkan nilai dari array pecahpencarian index ke 1
9	Memberikan nilai ke dalam variable tahun berdasarkan nilai dari array pecahpencarian index ke 0
10	Else
11	Memberikan nilai ke dalam variable bulan dengan nilai bulan saat ini
12	Memberikan nilai ke dalam variable tahun dengan nilai tahun saat ini
13	End branch
14	array offset = 0
15	array ha = 1
16	array modules = view beranda
17	array title = Beranda
18	array tujuan = hasil query pengambilan data jumlah rute yang paling sering dipesan berdasarkan parameter bulan dan tahun

Tabel 5.1 Algoritma Melihat Beranda

No	Pseudocode
19	Jika jumlah elemen dari array tujuan lebih besar daripada 0, start branch
20	Array status = 0
21	Else
22	Array status sama dengan 1
23	End branch
24	Array bulan = Nilai date dengan format "F" berdasarkan nilai dari variable tahun dan bulan
25	Array tahun = tahun
26	Array bulen = bulan
27	Array laporankeuanganjan = mengambil data pendapatan, gaji driver dan keuntungan pada bulan januari kemudian menjumlahkannya
28	Array laporankeuanganfeb = mengambil data pendapatan, gaji driver dan keuntungan pada bulan february kemudian menjumlahkannya
29	Array laporankeuanganmar = mengambil data pendapatan, gaji driver dan keuntungan pada bulan maret kemudian menjumlahkannya
30	Array laporankeuanganapr = mengambil data pendapatan, gaji driver dan keuntungan pada bulan april kemudian menjumlahkannya
31	Array laporankeuanganmei = mengambil data pendapatan, gaji driver dan keuntungan pada bulan mei kemudian menjumlahkannya
32	Array laporankeuanganjun = mengambil data pendapatan, gaji driver dan keuntungan pada bulan juni kemudian menjumlahkannya
33	Array laporankeuanganjul = mengambil data pendapatan, gaji driver dan keuntungan pada bulan juli kemudian menjumlahkannya
34	Array laporankeuanganagu = mengambil data pendapatan, gaji driver dan keuntungan pada bulan agustus kemudian menjumlahkannya
35	Array laporankeuangansep = mengambil data pendapatan, gaji driver dan keuntungan pada bulan september kemudian menjumlahkannya
36	Array laporankeuanganokt = mengambil data pendapatan, gaji driver dan keuntungan pada bulan oktober kemudian menjumlahkannya
37	Array laporankeuangannov = mengambil data pendapatan, gaji driver dan keuntungan pada bulan november kemudian menjumlahkannya
38	Array laporankeuangandes = mengambil data pendapatan, gaji driver dan keuntungan pada bulan desember kemudian menjumlahkannya
39	Array banyakjan = mengambil jumlah pemesanan yang telah terjadi pada bulan januari
40	Array banyakfeb = mengambil jumlah pemesanan yang telah terjadi pada bulan february
41	Array banyakmar = mengambil jumlah pemesanan yang telah terjadi pada bulan maret
42	Array banyakapr = mengambil jumlah pemesanan yang telah terjadi pada bulan april
43	Array banyakmei = mengambil jumlah pemesanan yang telah terjadi pada bulan mei

Tabel 5.1 Algoritma Melihat Beranda

No	Pseudocode
44	Array banyakjun = mengambil jumlah pemesanan yang telah terjadi pada bulan juni
45	Array banyakjul = mengambil jumlah pemesanan yang telah terjadi pada bulan juli
46	Array banyakagu = mengambil jumlah pemesanan yang telah terjadi pada bulan agustus
47	Array banyaksep = mengambil jumlah pemesanan yang telah terjadi pada bulan september
48	Array banyakokt = mengambil jumlah pemesanan yang telah terjadi pada bulan oktober
49	Array banyaknov = mengambil jumlah pemesanan yang telah terjadi pada bulan november
50	Array banyakdes = mengambil jumlah pemesanan yang telah terjadi pada bulan desember
51	Load view beranda dengan membawa data array

5.1.2.2 Algoritma Menghapus Pemesanan Travel

Algoritma menghapus pemesanan travel merupakan algoritma untuk menghapus pemesanan travel dari database. Tabel 5.2 akan menjelaskan algoritma memfilter mobil jadwal travel tambahan.

Nama class : adminTravel

Nama method : deletePTravel()

Tabel 5.2 Algoritma Menghapus Pemesanan Travel

No	Pseudocode
1	pemesanan = hasil query mengambil data pemesanan berdasarkan id
2	kursi = hasil query data keberangkatan berdasarkan kode mobil, tgl keberangkatan dan jam keberangkatan
3	pisahPemesanan = memecah nilai noKursi pada object pemesanan ke dalam array berdasarkan karakter ','
4	pisahKursi = memecah nilai noKursi pada object kursi ke dalam array berdasarkan karakter ','
5	dataTersisa = ""
6	Untuk setiap elemen yang ada pada array pisahKursi sebagai variable value maka lakukan, start loop
7	Jika nilai variable value sama dengan salah satu elemen pada array pisahPemesanan maka lakukan, start if
8	Tulis "Data Sama"
9	Else
10	dataTersisa = nilai variable value
11	End if
12	End loop
13	pecahDataTersisa = memecah nilai variabel dataTersisa menjadi array berdasarkan karakter ','
14	Hapus elemen terakhir dari array

Tabel 5.2 Algoritma Menghapus Pemesanan Travel

No	Pseudocode
15	dataTersisa = menyatukan elemen array menjadi string berdasarkan karakter ','
16	Jika jumlah elemen array pecahDataTersisa lebih dari 0 maka lakukan, start if
17	Array noKursi = nilai variable dataTersisa
18	Update data pada tabel keberangkatan berdasarkan idKeberangkatan dan array data
19	Else
20	Delete data pada tabel keberangkatan berdasarkan idKeberangkatan
21	End if
22	Delete data pada tabel pemesanan berdasarkan id
23	Set flashdata bahwa Data telah dihapus
24	Mengarahkan ke view adminTravel/viewTravel

5.1.2.3 Algoritma Mencetak Laporan Keuangan

Algoritma mencetak laporan keuangan merupakan algoritma untuk mencetak laporan keuangan. Tabel 5.3 akan menjelaskan algoritma mencari jadwal tersedia.

Nama kelas : laporan

Nama method : printLaporan()

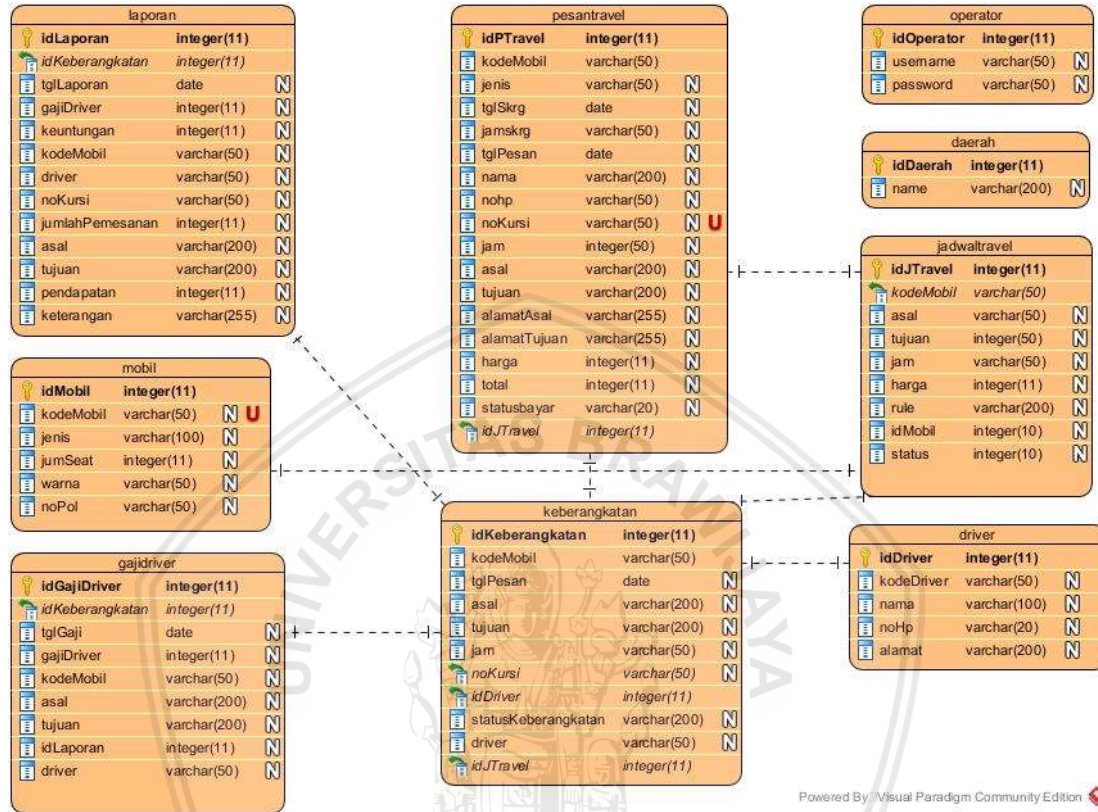
No	Pseudocode
1	Array offset = 0
2	Jika nilai parameter bulan = 0 maka lakukan, start if
3	bulan = ""
4	tahun = ""
5	array view = hasil dari query pengambilan data laporan
6	Else
7	Array bulan = nilai bulan dengan format "F" menurut parameter bulan dan tahun
8	Array tahun = nilai parameter tahun
9	Array view = hasil dari query pengambilan data laporan berdasarkan bulan dan tahun dari parameter
10	End if
11	Load view tampilanPrint

5.1.3 Perancangan Data

Perancangan data bertujuan untuk memberikan gambaran tentang basis data yang akan dibangun di dalam sistem. Perancangan ini nantinya juga akan menggambarkan objek-objek beserta field-field apa saja yang ada di dalam sistem.

5.1.3.1 Conceptual Data Model

Conceptual Data Model (CDM) merupakan gambaran secara detail basis data dalam bentuk konsep. Pada perancangan data *conceptual data model* akan dijelaskan oleh gambar 5.5.



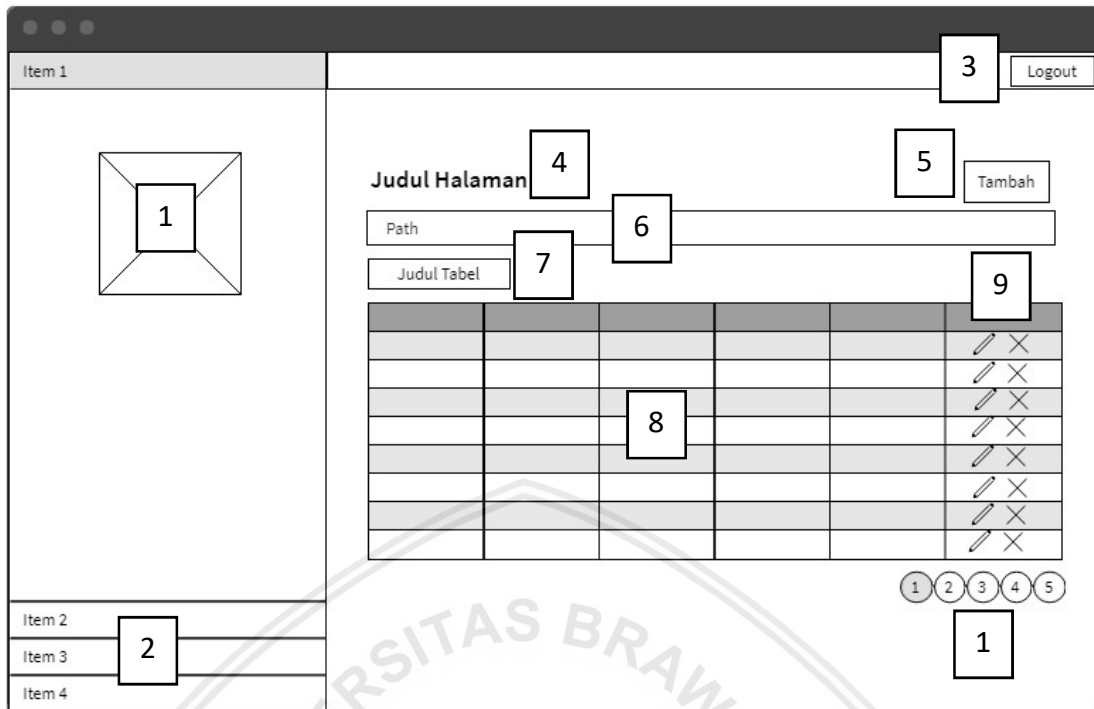
Gambar 5.5 Conceptual Data Model

5.1.4 Perancangan Antarmuka

Perancangan antarmuka dibuat untuk mengetahui gambaran antarmuka sistem yang akan dibangun sehingga dalam tahap implementasi perancangan ini akan membantu pengembang dalam membuat antarmuka sistem.

5.1.4.1 Perancangan Antarmuka untuk Melihat Tabel

Perancangan antarmuka untuk fungsional melihat tabel merupakan perancangan tampilan untuk fungsional-fungsional dalam sistem yang berkaitan dengan melihat tabel. Fungsional yang termasuk dalam kategori ini ada 10, yaitu jadwal travel harian, melihat jadwal travel tambahan, melihat pemesanan travel, melihat manajemen pemesanan, melihat daftar penumpang, melihat data mobil, melihat data driver, melihat data kota, melihat laporan keuangan dan melihat gaji driver. Rancangan antarmuka dapat dilihat pada gambar 5.6 dan penjelasan dari gambar ada pada tabel 5.3.



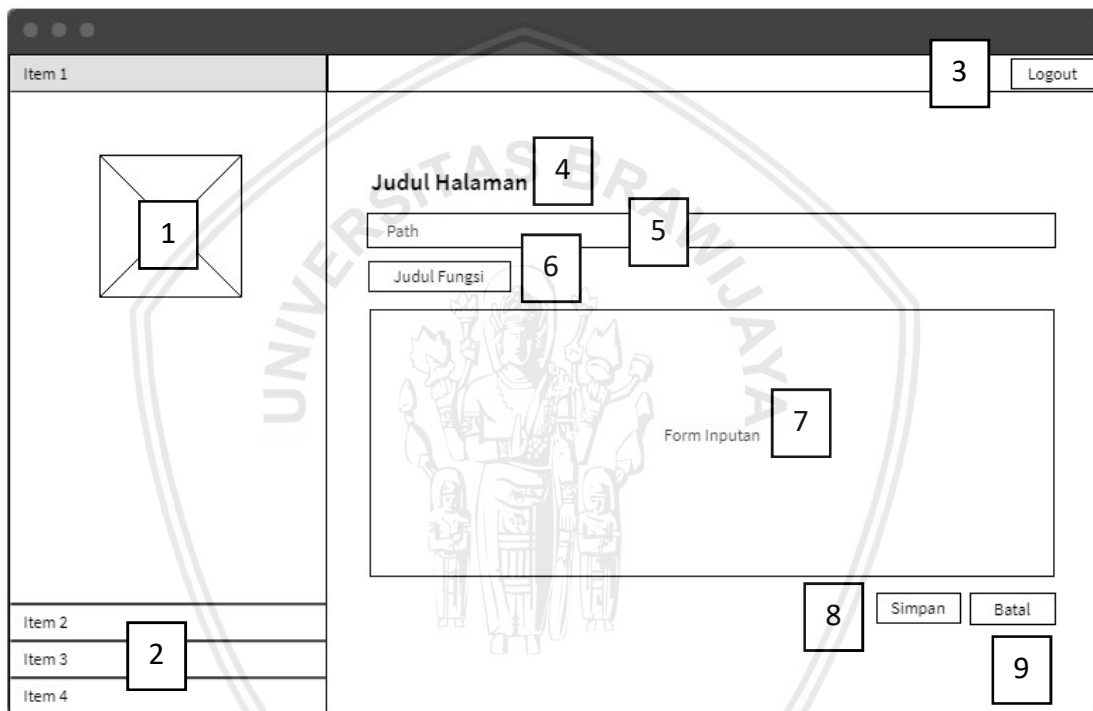
Gambar 5.6 Perancangan Antarmuka untuk Melihat Tabel

Tabel 5.3 Penjelasan Gambar Perancangan Antarmuka untuk Melihat Tabel

No	Nama Objek	Keterangan
1	Logo Perusahaan	Menampilkan Logo Perusahaan
2	Navbar	Menampilkan menu-menu navigasi dari sistem
3	Tombol Logout	Tombol yang digunakan untuk keluar dari sistem
4	Judul Halaman	Menampilkan judul halaman apa yang sedang dibuka
5	Button Tambah	Tombol yang digunakan untuk menambah data tabel tertentu
6	Tampilan Path	Tampilan dimana untuk mengetahui berada di halaman mana pengguna saat ini
7	Judul Tabel	Menampilkan judul dari tabel yang ditampilkan
8	Tabel	Menampilkan isi tabel yang datanya dari database
9	Tombol aksi	Tombol-tombol yang digunakan untuk memanipulasi sebuah baris data pada tabel

5.1.4.2 Perancangan Antarmuka untuk Menambah dan Merubah Tabel

Perancangan antarmuka untuk menambah dan merubah tabel merupakan perancangan tampilan halaman yang digunakan untuk menambah dan merubah data pada tabel yang ada di dalam database. Fungsional yang termasuk dalam kategori ini ada 14, yaitu menambah jadwal travel harian, mengubah jadwal travel harian, menambah jadwal travel tambahan, mengubah jadwal travel tambahan, mengubah pemesanan travel, menambah data mobil, mengubah data mobil, menambah data driver, mengubah data driver, menambah data kota, mengubah data kota, menambah laporan keuangan, mengubah laporan keuangan dan mengubah gaji driver. Rancangan antarmuka dapat dilihat pada gambar 5.7 dan penjelasan dari gambar ada pada tabel 5.4.



Gambar 5.7 Perancangan antarmuka untuk menambah dan merubah tabel

Tabel 5.4 Penjelasan gambar perancangan antarmuka untuk menambah dan merubah tabel

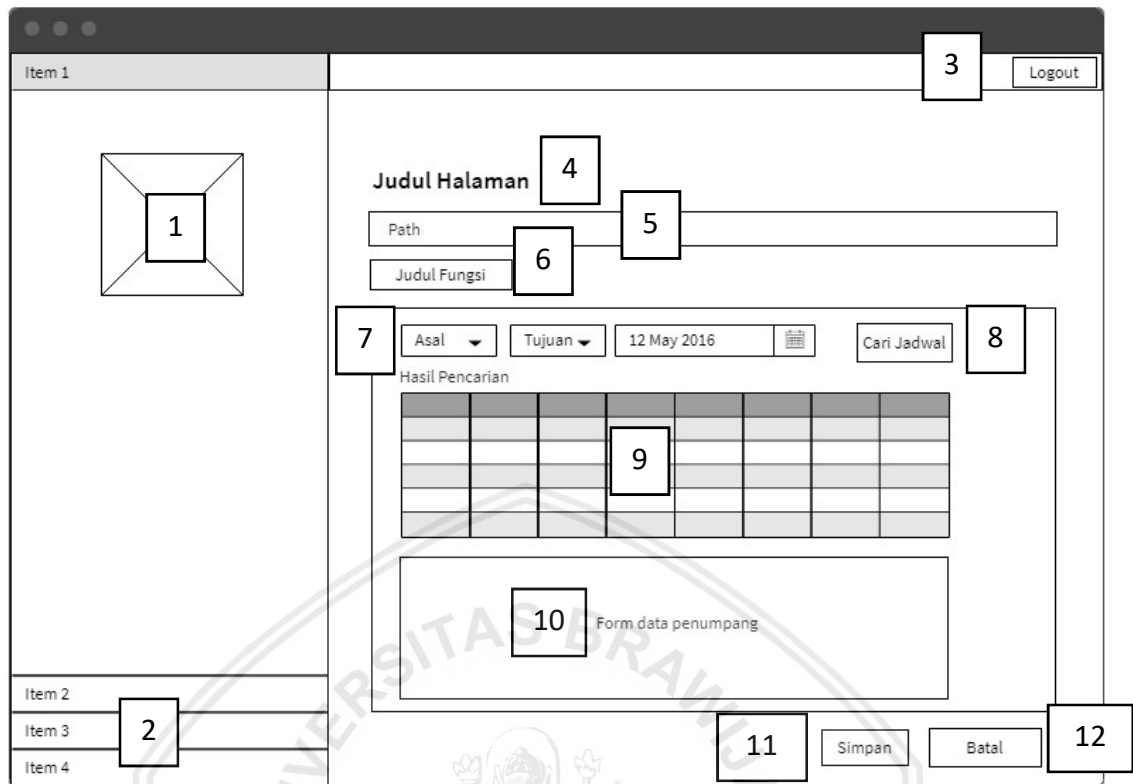
No	Nama Objek	Keterangan
1	Logo Perusahaan	Menampilkan Logo Perusahaan
2	Navbar	Menampilkan menu-menu navigasi dari sistem
3	Tombol Logout	Tombol yang digunakan untuk keluar dari sistem

Tabel 5.4 Penjelasan gambar perancangan antarmuka untuk menambah dan merubah tabel

No	Nama Objek	Keterangan
4	Judul Halaman	Menampilkan judul halaman apa yang sedang dibuka
5	Tampilan Path	Tampilan dimana untuk mengetahui berada di halaman mana pengguna saat ini
6	Judul Fungsi	Menampilkan penambahan atau pengubahan data apa yang akan dilakukan
7	Form Inputan	Menampilkan inputan-inputan yang diperlukan sesuai dengan data yang ada di dalam database
8	Tombol Simpan	Tombol yang digunakan untuk menyimpan atau mengubah data yang ada di dalam database berdasarkan dari form inputan
9	Tombol Batal	Tombol yang digunakan untuk membatalkan proses penambahan atau pengubahan data dan kembali ke halaman melihat data yang ada di dalam tabel

5.1.4.3 Perancangan Antarmuka untuk Menambah Pemesanan Travel

Perancangan antarmuka untuk menambah pemesanan travel merupakan perancangan tampilan halaman yang digunakan untuk menambah pemesanan travel dan menyimpannya ke dalam database. Rancangan antarmukanya dapat dilihat pada gambar 5.8 dan penjelasan dari gambar ada pada tabel 5.5.



Gambar 5.8 Perancangan antarmuka untuk menambah pemesanan travel

Tabel 5.5 Penjelasan gambar perancangan antarmuka untuk menambah pemesanan travel

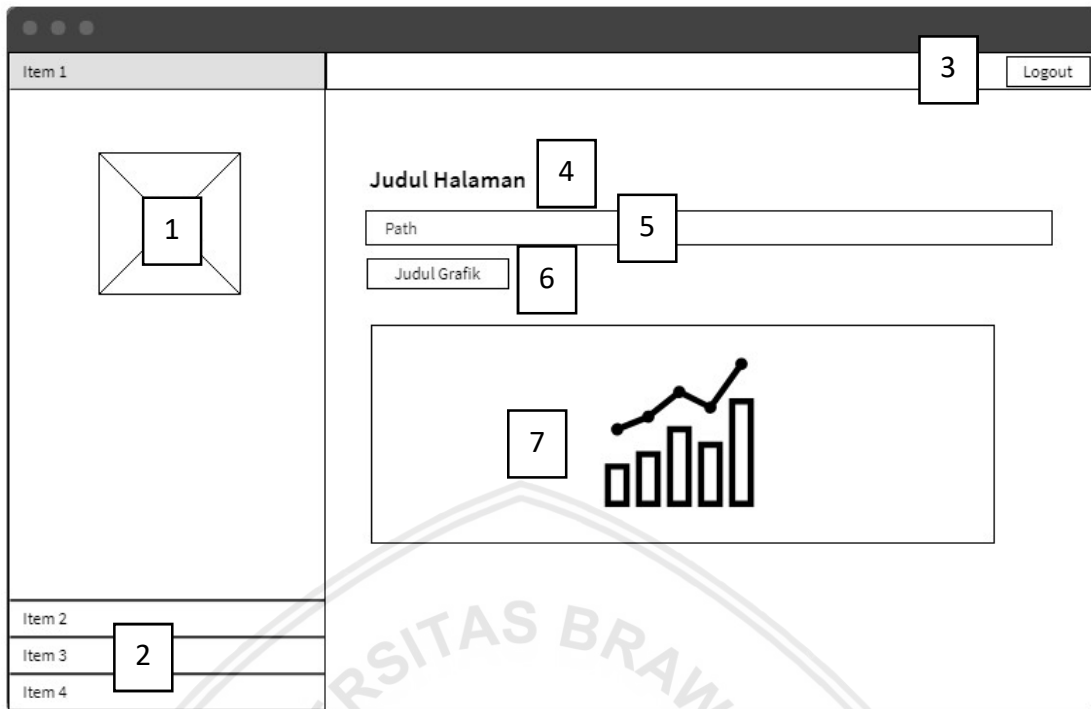
No	Nama Objek	Keterangan
1	Logo Perusahaan	Menampilkan Logo Perusahaan
2	Navbar	Menampilkan menu-menu navigasi dari sistem
3	Tombol Logout	Tombol yang digunakan untuk keluar dari sistem
4	Judul Halaman	Menampilkan judul halaman apa yang sedang dibuka
5	Tampilan Path	Tampilan dimana untuk mengetahui berada di halaman mana pengguna saat ini
6	Judul Fungsi	Menampilkan judul fungsi yang akan dilakukan
7	Filter Inputan	Inputan-inputan seperti asal, tujuan dan tanggal pemesanan yang digunakan untuk memfilter data jadwal travel

Tabel 5.5 Penjelasan gambar perancangan antarmuka untuk menambah pemesanan travel

No	Nama Objek	Keterangan
8	Tombol Cari Jadwal	Tombol yang digunakan memfilter data jadwal berdasarkan inputan yang telah dipilih
9	Tabel Hasil Pencarian	Tabel yang menampilkan jadwal-jadwal yang tersedia
10	Form Data Penumpang	Menampilkan form berisi inputan-inputan yang digunakan untuk melengkapi data penumpang yang memesan layanan travel
11	Tombol Simpan	Tombol yang digunakan untuk menyimpan data pemesanan travel berdasarkan form inputan
12	Tombol Batal	Tombol yang digunakan untuk membatalkan proses pemesanan dan kembali ke halaman melihat data yang ada di dalam tabel

5.1.4.4 Perancangan Antarmuka untuk Melihat Grafik Pemesanan

Perancangan antarmuka melihat grafik pemesanan merupakan rancangan tampilan halaman yang digunakan untuk melihat grafik pemesanan seperti grafik laporan keuangan pemesanan travel dan grafik rute yang paling banyak dipesan. Rancangan antarmuka untuk melihat grafik pemesanan dapat dilihat pada gambar 5.9 dan penjelasan dari gambar ada pada tabel 5.6.



Gambar 5.9 Perancangan antarmuka melihat grafik pemesanan

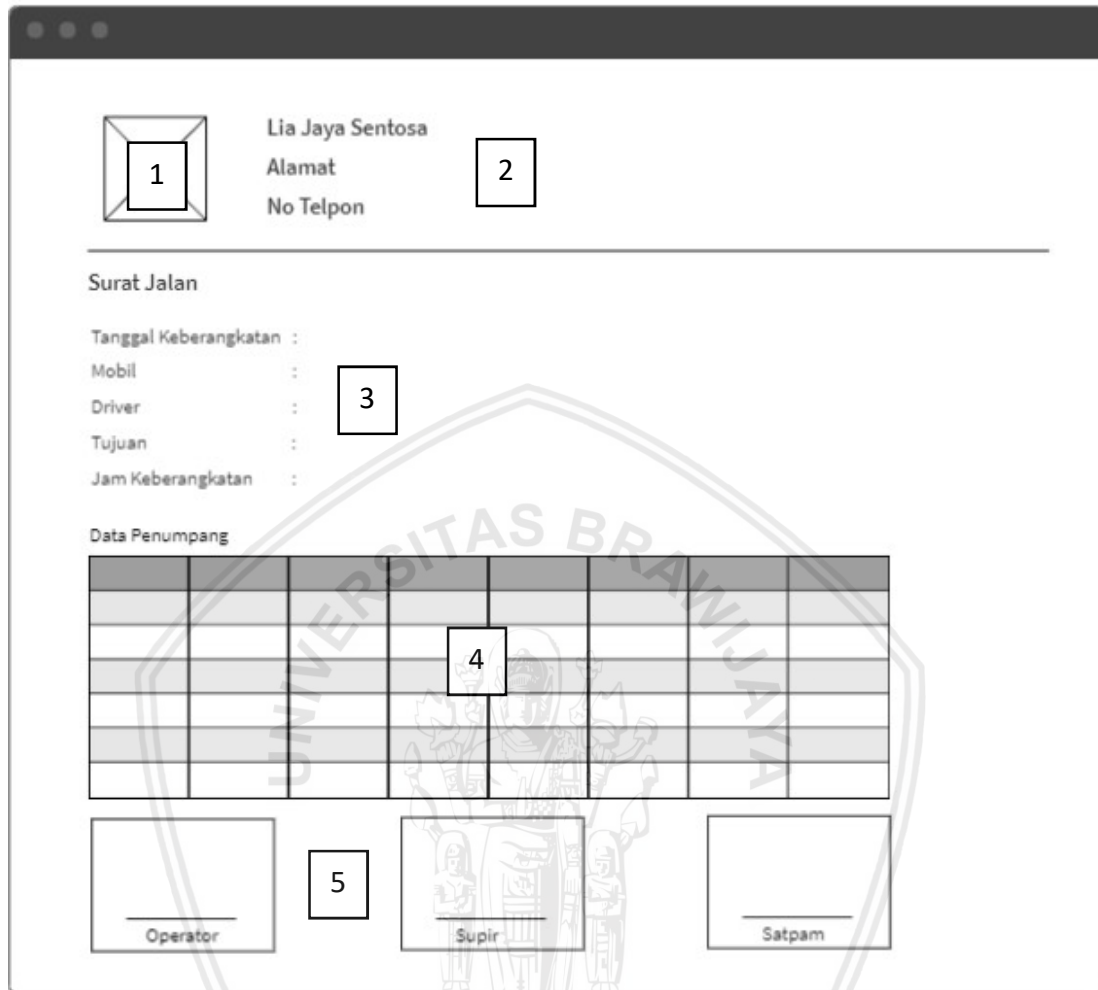
Tabel 5.6 Perancangan antarmuka melihat grafik pemesanan

No	Nama Objek	Keterangan
1	Logo Perusahaan	Menampilkan Logo Perusahaan
2	Navbar	Menampilkan menu-menu navigasi dari sistem
3	Tombol Logout	Tombol yang digunakan untuk keluar dari sistem
4	Judul Halaman	Menampilkan judul halaman apa yang sedang dibuka
5	Tampilan Path	Tampilan dimana untuk mengetahui berada di halaman mana pengguna saat ini
6	Judul Grafik	Menampilkan judul grafik yang ditampilkan
7	Grafik	Menampilkan grafik untuk laporan keuangan dan laporan rute pemesanan berdasarkan bulan dan tahun

5.1.4.5 Perancangan Antarmuka Print Surat Jalan

Perancangan antarmuka print surat jalan merupakan rancangan tampilan halaman print surat jalan yang biasanya digunakan sebagai perizinan driver

membawa melakukan keberangkatan. Rancangan antarmuka print surat jalan dapat dilihat pada gambar 5.10 dan penjelasan dari gambar ada pada tabel 5.7.



Gambar 5.10 Perancangan antarmuka print surat jalan

Tabel 5.7 Penjelasan gambar perancangan antarmuka print surat jalan

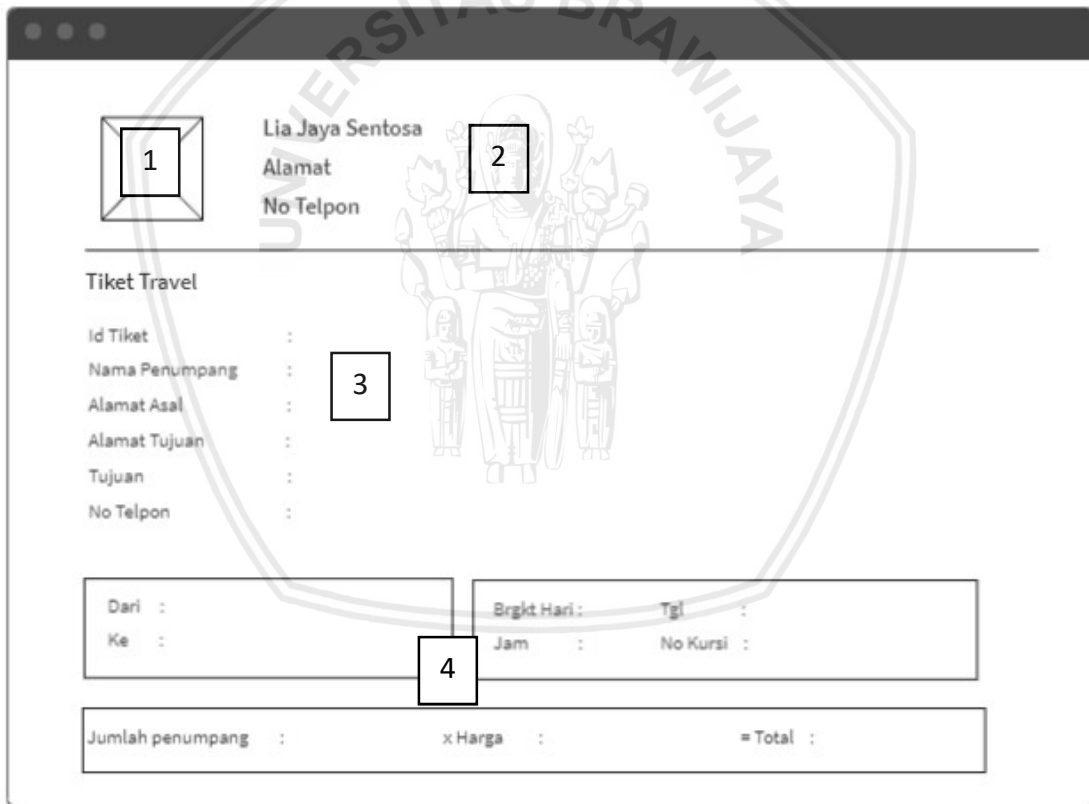
No	Nama Objek	Keterangan
1	Logo Perusahaan	Menampilkan Logo Perusahaan
2	Identitas Perusahaan	Menampilkan identitas dari perusahaan seperti nama perusahaan, nomer telepon dan alamat
3	Data Keberangkatan	Berisi data yang menerangkan tentang keberangkatan yang akan dilakukan

Tabel 5.7 Penjelasan gambar perancangan antarmuka print surat jalan

No	Nama Objek	Keterangan
4	Data Penumpang	Berisi data penumpang yang dapat digunakan untuk memberikan informasi kepada driver mengenai data penumpang yang akan diantarkan
5	Perizinan	Perizinan berupa tanda tangan yang dilakukan oleh operator, supir dan satpam yang menjaga mobil

5.1.4.6 Perancangan Antarmuka Print Tiket Travel

Perancangan antarmuka print tiket travel merupakan rancangan tampilan halaman print tiket travel yang digunakan dalam transaksi pembayaran travel. Rancangan antarmuka dapat dilihat pada gambar 5.11 dan penjelasan dari gambar ada pada tabel 5.8.



Gambar 5.11 Perancangan antarmuka print tiket travel

Tabel 5.8 Perancangan antarmuka print tiket travel

No	Nama Objek	Keterangan
1	Logo Perusahaan	Menampilkan Logo Perusahaan
2	Identitas Perusahaan	Menampilkan identitas dari perusahaan seperti nama perusahaan, nomer telepon dan alamat
3	Identitas Penumpang	Berisi data yang menerangkan identitas penumpang
4	Data pemesanan	Berisi data pemesanan yang diambil dari tabel pemesanan

5.2 Implementasi

Pada tahap ini berisi tentang implementasi sistem informasi manajemen travel Lia Jaya Sentosa mengacu dari hasil analisis kebutuhan dan perancangan yang telah dimodelkan sebelumnya. Pembahasan sub bab ini terdiri dari spesifikasi sistem, batasan implementasi, implementasi basis data, implementasi komponen dan implementasi antarmuka.

5.2.1 Spesifikasi Sistem

Pada tahap ini berisi tentang spesifikasi perangkat keras dan spesifikasi perangkat lunak yang digunakan. Berikut adalah spesifikasi perangkat keras dan spesifikasi perangkat lunak.

5.2.1.1 Spesifikasi Perangkat Keras

Implementasi Sistem Informasi Manajemen Travel Lia Jaya Sentosa menggunakan spesifikasi perangkat keras yang dapat dilihat pada tabel berikut.

Tabel 5.9 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
Proessor	Intel Core i3
Memori (RAM)	4 GB
Kartu Grafis	Nvidia Geforce GT 720M
Harddisk	500 GB
Monitor	14.0" Resolusi 1366 x 768

5.2.1.2 Spesifikasi Perangkat Lunak

Implementasi Sistem Informasi Manajemen Travel Lia Jaya Sentosa menggunakan spesifikasi perangkat lunak yang dapat dilihat pada tabel berikut.

Tabel 5.10 Spesifikasi Perangkat Lunak

Nama Komponen	Spesifikasi
Sistem Operasi	Windows 10 Enterprise
<i>Browser</i>	Google Chrome versi 73.0.3683.86
<i>Code Editor</i>	Sublime Text 3
Basis Data	MySQL
Pemodelan, UML	Visual Paradigm 15.2
<i>Framework</i>	CodeIgniter, Bootstrap, DataTables
Dokumentasi	Microsoft Word 2016

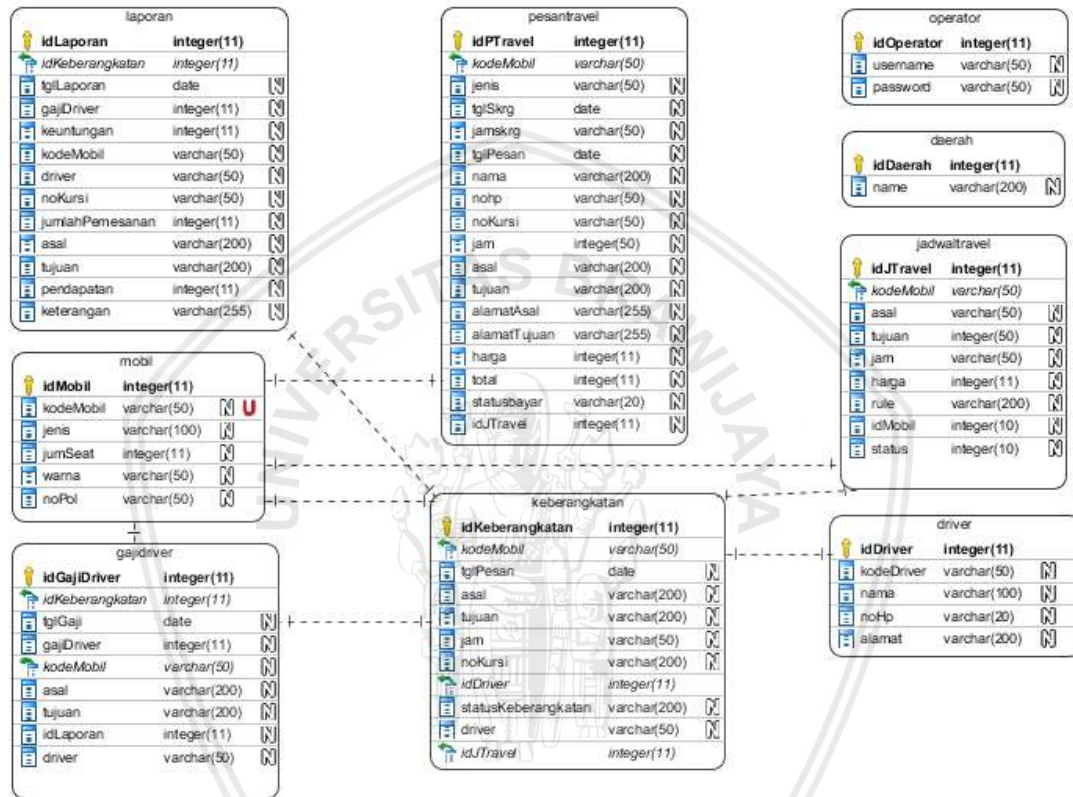
5.2.2 Batasan Implementasi

Beberapa batasan dalam implementasi Sistem Informasi Manajemen Travel Lia Jaya Sentosa diantaranya:

1. Implementasi Sistem Informasi Manajemen Travel Lia Jaya Sentosa ini dibangun menggunakan bahasa pemrograman PHP, framework CodeIgniter dan Bootstrap dalam pembangunan websitenya, serta menggunakan basis data MySQL.
2. Implementasi Sistem Informasi Manajemen Travel tidak menggunakan sistem GPS Tracking.
3. Gaji yang dipresentasikan di dalam sistem hanya gaji driver saja.
4. Laporan keuangan pemesanan travel berasal dari pemasukan travel dan pengeluarannya berupa gaji driver.
5. Data yang digunakan dalam implementasi ini diambil dari Perusahaan Travel Lia Jaya Sentosa.
6. Implementasi Sistem Informasi Manajemen Travel Lia Jaya Sentosa memiliki fitur utama berupa pengelolaan jadwal travel, pemesanan travel, laporan keuangan pemesanan travel dan laporan grafik pemesanan travel.

5.2.3 Implementasi Basis Data

Implementasi basis data pada Sistem Informasi Manajemen Travel Lia Jaya Sentosa didasarkan pada hasil perancangan basis data sebelumnya yang telah dimodelkan. Pada tahapan ini akan dibuat *Physical Data Model* (PDM) yang merupakan tahapan untuk mentransformasikan hasil dari identifikasi entitas yang sebelumnya telah dilakukan pada *Conceptual Data Model* (CDM) ke dalam tabel database fisik yang terperinci. Hasil implementasi basis data Sistem Informasi Manajemen Travel Lia Jaya Sentosa dapat dilihat pada gambar berikut.



Gambar 5.12 Physical Data Model Sistem informasi Manajemen Travel Lia Jaya Sentosa

5.2.4 Implementasi Komponen

Dalam implementasi komponen dilakukan implementasi logika program yang akan dibangun dengan mengacu pada perancangan komponen yang telah dimodelkan sebelumnya dengan memperhatikan algoritma-algoritma utama yang berada didalam sistem yaitu menghapus pemesanan travel, memfilter mobil jadwal travel tambahan, mencari jadwal tersedia.

5.2.4.1 Implementasi Melihat Beranda

Melihat beranda merupakan kegiatan untuk melihat halaman beranda yang didalamnya terdapat grafik. Implementasi melihat beranda dapat dilihat pada tabel 5.12 berikut.

Tabel 5.11 Implementasi Melihat Beranda

No	Source Code
1	<code>public function index(\$in)</code>
2	<code>{</code>
3	<code> \$ Pencarian="";</code>
4	<code> \$bulan ="";</code>
5	<code> \$tahun = "";</code>
6	<code> \$data['status'] = 0;</code>
7	<code> if(\$in==1){</code>
8	<code> \$ Pencarian=\$_POST['1'];</code>
9	<code> \$ pecah Pencarian = explode("-", \$ Pencarian);</code>
10	<code> \$bulan=\$ pecah Pencarian[1];</code>
11	<code> \$tahun=\$ pecah Pencarian[0];</code>
12	<code> } else{</code>
13	<code> date_default_timezone_set('Asia/Jakarta');</code>
14	<code> \$bulan = date("m");</code>
15	<code> \$tahun = date("Y");</code>
16	<code> }</code>
17	<code> // echo \$bulan.'-'. \$tahun;</code>
18	<code> \$data['offset'] = 0;</code>
19	<code> \$data['ha'] = 1;</code>
20	<code> \$data['modules'] = 'admin/beranda';</code>
21	<code> \$data['title'] = 'Beranda';</code>
22	<code> \$data['tujuan'] = \$this->laporanModel-</code>
23	<code>>rutetersering(\$bulan,\$tahun);</code>
24	<code> // echo count(\$data['tujuan']);</code>
25	<code> if(count(\$data['tujuan'])>0){</code>
26	<code> \$data['status']=0;</code>
27	<code> } else{</code>
28	<code> \$data['status']=1;</code>
29	<code> }</code>
30	<code> \$data['bulan'] = date("F",strtotime(\$tahun.'-</code>
31	<code>.'. \$bulan));</code>
32	<code> \$data['tahun'] = \$tahun;</code>
33	<code> \$data['bulen'] = \$bulan;</code>
34	<code> \$data['laporankeuanganjan'] = \$this->laporanModel-</code>
35	<code>>laporankeuangan("01", \$tahun);</code>
36	<code> \$data['laporankeuanganfeb'] = \$this->laporanModel-</code>
37	<code>>laporankeuangan("02", \$tahun);</code>
38	<code> \$data['laporankeuanganmar'] = \$this->laporanModel-</code>
39	<code>>laporankeuangan("03", \$tahun);</code>
40	<code> \$data['laporankeuanganapr'] = \$this->laporanModel-</code>
41	<code>>laporankeuangan("04", \$tahun);</code>
42	<code> \$data['laporankeuanganmei'] = \$this->laporanModel-</code>
43	<code>>laporankeuangan("05", \$tahun);</code>
44	<code>}</code>

Tabel 5.11 Implementasi Melihat Beranda

No	Source Code
46	<code>\$data['laporankeuanganjun'] = \$this->laporanModel-</code>
47	<code>>laporankeuangan("06", \$tahun);</code>
48	<code>\$data['laporankeuanganjul'] = \$this->laporanModel-</code>
49	<code>>laporankeuangan("07", \$tahun);</code>
50	<code>\$data['laporankeuanganagu'] = \$this->laporanModel-</code>
51	<code>>laporankeuangan("08", \$tahun);</code>
52	<code>\$data['laporankeuangansep'] = \$this->laporanModel-</code>
53	<code>>laporankeuangan("09", \$tahun);</code>
54	<code>\$data['laporankeuanganokt'] = \$this->laporanModel-</code>
55	<code>>laporankeuangan("10", \$tahun);</code>
56	<code>\$data['laporankeuangannov'] = \$this->laporanModel-</code>
57	<code>>laporankeuangan("11", \$tahun);</code>
58	<code>\$data['laporankeuangandes'] = \$this->laporanModel-</code>
59	<code>>laporankeuangan("12", \$tahun);</code>
60	
61	<code>\$data['banyakjan'] = \$this->laporanModel-</code>
62	<code>>banyak("01", \$tahun);</code>
63	<code>\$data['banyakfeb'] = \$this->laporanModel-</code>
64	<code>>banyak("02", \$tahun);</code>
65	<code>\$data['banyakmar'] = \$this->laporanModel-</code>
66	<code>>banyak("03", \$tahun);</code>
67	<code>\$data['banyakapr'] = \$this->laporanModel-</code>
68	<code>>banyak("04", \$tahun);</code>
69	<code>\$data['banyakmei'] = \$this->laporanModel-</code>
70	<code>>banyak("05", \$tahun);</code>
71	<code>\$data['banyakjun'] = \$this->laporanModel-</code>
72	<code>>banyak("06", \$tahun);</code>
73	<code>\$data['banyakjul'] = \$this->laporanModel-</code>
74	<code>>banyak("07", \$tahun);</code>
75	<code>\$data['banyakagu'] = \$this->laporanModel-</code>
76	<code>>banyak("08", \$tahun);</code>
77	<code>\$data['banyaksep'] = \$this->laporanModel-</code>
78	<code>>banyak("09", \$tahun);</code>
79	<code>\$data['banyakokt'] = \$this->laporanModel-</code>
80	<code>>banyak("10", \$tahun);</code>
81	<code>\$data['banyaknov'] = \$this->laporanModel-</code>
82	<code>>banyak("11", \$tahun);</code>
83	<code>\$data['banyakdes'] = \$this->laporanModel-</code>
84	<code>>banyak("12", \$tahun);</code>
85	<code>// var_dump(\$data['laporankeuanganjan']);</code>
86	<code>// var_dump(\$data['tujuan']);</code>
87	<code>// echo_count(\$data['tujuan']);</code>
88	<code>// echo \$data['status'];</code>
89	
90	<code>\$this->load->view('app', \$data);</code>
91	<code>}</code>

5.2.4.2 Implementasi Menghapus Pemesanan Travel

Menghapus pemesanan travel adalah kegiatan menghapus data pemesanan travel dari dalam database. Untuk proses penghapusannya, selain menghapus data yang ada di dalam tabel pemesanan travel, penghapusan data pada tabel keberangkatan juga dilakukan. Implementasi menghapus pemesanan travel dapat dilihat pada tabel 5.13 berikut.

Tabel 5.12 Implementasi Menghapus Pemesanan Travel

No	Source Code
1	<code>public function deletePTravel(\$id)</code>
2	<code>{</code>
3	<code> \$pesanan = \$this->travelModel->kursipesan(\$id);</code>
4	<code> \$kursi = \$this->travelModel->cariStatusSeat1(\$pesanan-></code>
5	<code>kodeMobil,\$pesanan->tglPesan,\$pesanan->jam);</code>
6	<code> \$pisahPesanan = explode(',', \$pesanan->noKursi);</code>
7	<code> \$pisahKursi = explode(',', \$kursi->noKursi);</code>
8	<code></code>
9	<code> \$dataTersisa = "";</code>
10	<code> foreach (\$pisahKursi as \$value) {</code>
11	<code> if (in_array(\$value, \$pisahPesanan)) {</code>
12	<code> echo "Data Sama
";</code>
13	<code> }else{</code>
14	<code> \$dataTersisa .= \$value . ',';</code>
15	<code> // echo \$dataTersisa;</code>
16	<code> }</code>
17	<code> }</code>
18	<code></code>
19	<code> \$pecahDataTersisa = explode(',', \$dataTersisa);</code>
20	<code> array_pop(\$pecahDataTersisa);</code>
21	<code> \$dataTersisa = implode(',', \$pecahDataTersisa);</code>
22	<code> echo count(\$pecahDataTersisa) . 'datatersisa
';</code>
23	<code> if(count(\$pecahDataTersisa)>0){</code>
24	<code> \$data['noKursi'] = \$dataTersisa;</code>
25	<code> \$this->travelModel->updatekursidelete(\$kursi-></code>
26	<code>idKeberangkatan, \$data);</code>
27	<code> } else{</code>
28	<code> \$this->travelModel->deleteKursi(\$kursi-></code>
29	<code>idKeberangkatan);</code>
30	<code> }</code>
31	<code> \$this->travelModel->deleteptravel(\$id);</code>
32	<code> \$this->session->set_flashdata('category_success', 'Data</code>
33	<code>Pemesanan Telah Dihapus');</code>
34	<code> redirect(site_url('adminTravel/viewTravel'));</code>
35	<code>}</code>

5.2.4.3 Implementasi Mencetak Laporan Keuangan

Mencetak laporan keuangan merupakan kegiatan untuk mencetak laporan keuangan. Implementasi mencetak laporan keuangan dapat dilihat pada tabel 5.14 berikut.

Tabel 5.13 Implementasi mencetak laporan keuangan

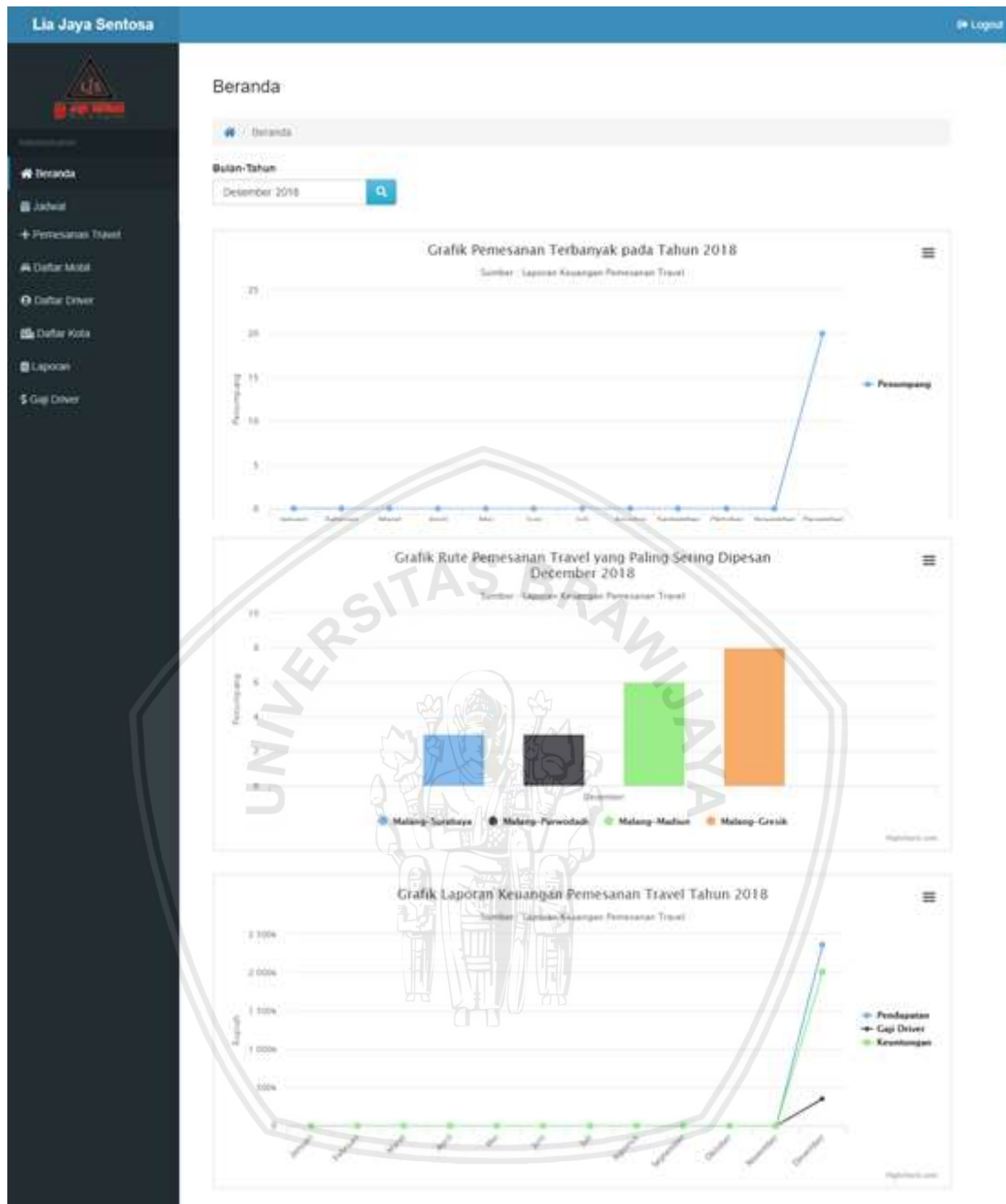
No	Source Code
1	<code>public function printLaporan(\$bulan, \$tahun){</code>
2	<code> \$data['offset'] = 0;</code>
3	<code> // echo \$bulan;</code>
4	<code> if(\$bulan==0){</code>
5	<code> \$bulan = "";</code>
6	<code> \$tahun = "";</code>
7	<code> \$data['view'] = \$this->laporanModel->displaylaporan();</code>
8	<code> } else{</code>
9	<code> \$data['bulan'] = date("F", strtotime(\$tahun.'-</code>
10	<code>.\$bulan));</code>
11	<code> // echo \$data['bulan'];</code>
12	<code> \$data['tahun'] = \$tahun;</code>
13	<code> \$data['view'] = \$this->laporanModel-</code>
14	<code>>displaylaporanbulan(\$bulan, \$tahun);</code>
15	<code> }</code>
16	<code> \$this->load->view('admin/laporan/tampilanPrint', \$data);</code>
17	<code>}</code>

5.2.5 Implementasi Antarmuka

Implementasi antarmuka merupakan proses pembuatan antarmuka sistem berdasarkan rancangan antarmuka yang telah dimodelkan sebelumnya. Berikut adalah implementasi antarmuka.

5.2.5.1 Implementasi Antarmuka Melihat Grafik Laporan

Antarmuka ini merupakan halaman untuk melihat grafik laporan. Di dalam grafik ini terdapat 3 grafik yaitu grafik pemesanan yang paling sering dipesan dalam satu bulan dalam satu tahun, grafik laporan rute pemesanan yang paling sering dipesan setiap bulan dan grafik laporan keuangan setiap bulan dalam satu tahun. Untuk implementasi antarmuka melihat jadwal travel harian dapat dilihat pada gambar berikut.

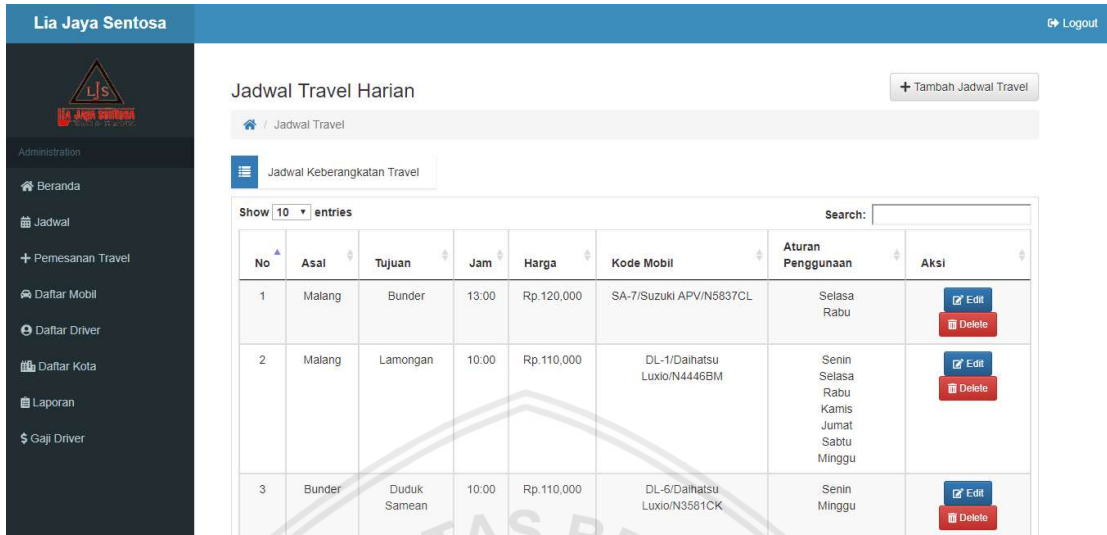


Gambar 5.13 Implementasi Melihat Grafik Laporan

5.2.5.2 Implementasi Antarmuka Melihat Jadwal Travel Harian

Antarmuka ini merupakan halaman untuk melihat jadwal travel harian. Travel harian merupakan jadwal travel yang dapat digunakan sehari-hari, jadi jadwal ini

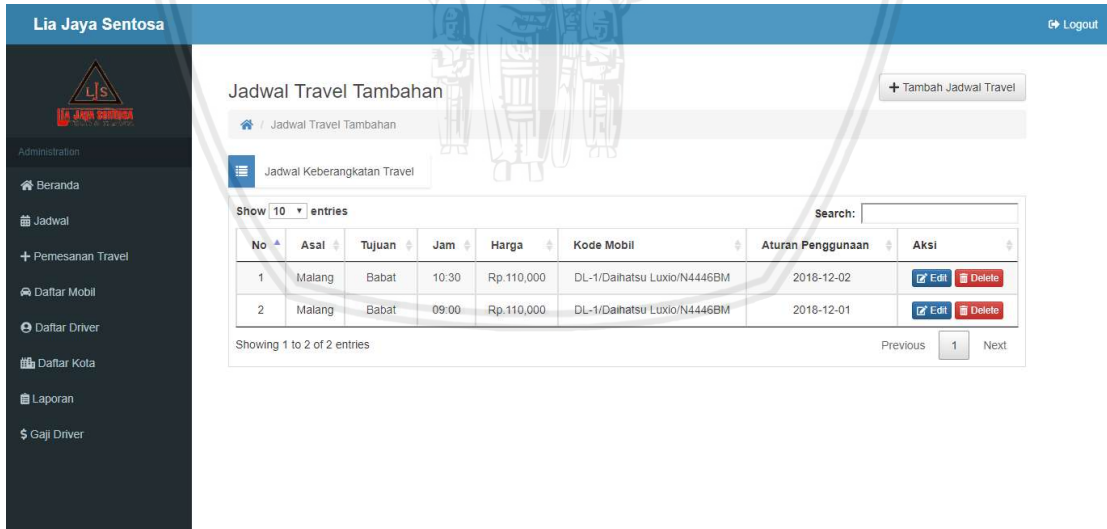
diseleksi berdasarkan hari. Untuk implementasi antarmuka melihat jadwal travel harian dapat dilihat pada gambar berikut.



Gambar 5.14 Implementasi Menambah Jadwal Travel Harian

5.2.5.3 Implementasi Antarmuka Melihat Jadwal Travel Tambahan

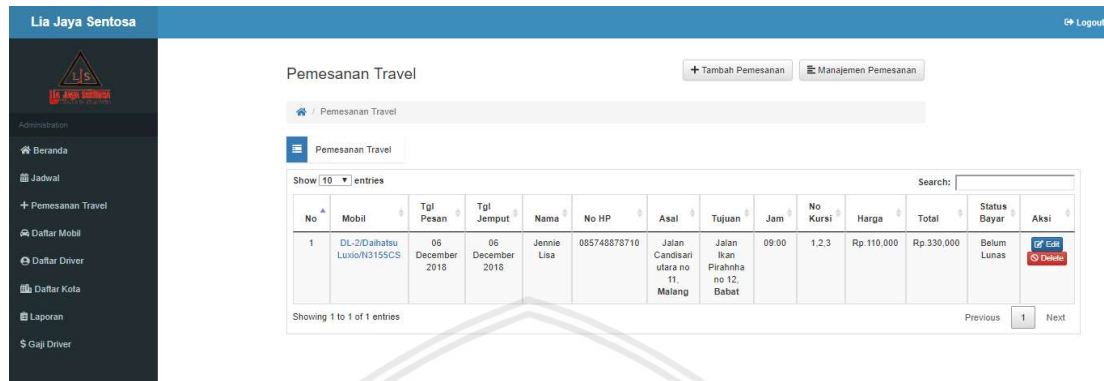
Antarmuka ini merupakan halaman untuk melihat jadwal travel tambahan. Travel tambahan merupakan jadwal travel yang dapat digunakan hanya sekali, pada tanggal tertentu saja, jadi jadwal ini diseleksi berdasarkan tanggal. Untuk implementasi antarmuka melihat jadwal travel tambahan dapat dilihat pada gambar berikut.



Gambar 5.15 Implementasi Melihat Jadwal Travel Tambahan

5.2.5.4 Implementasi Antarmuka Melihat Pemesanan Travel

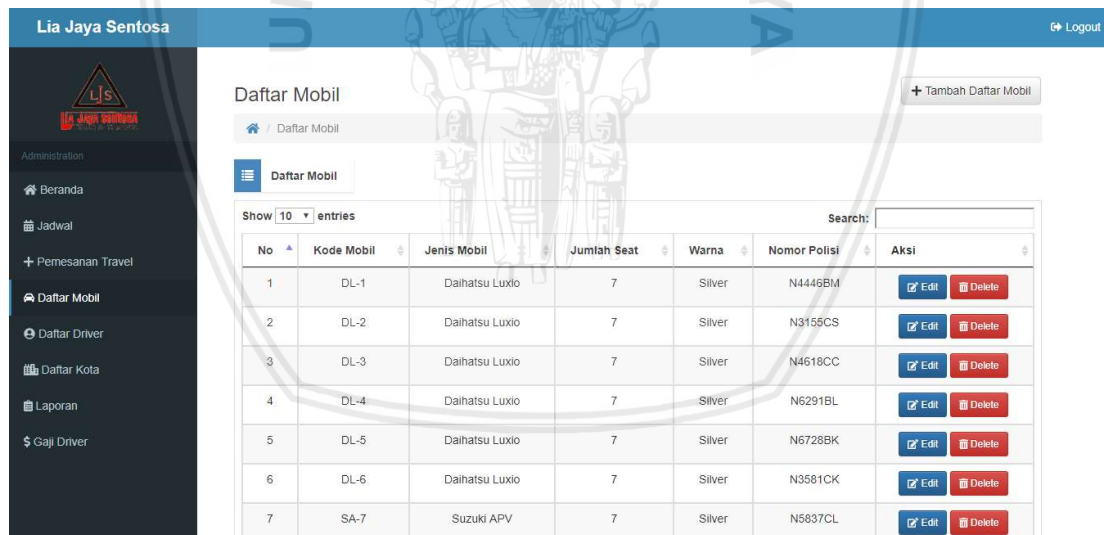
Antarmuka ini merupakan halaman untuk melihat pemesanan travel yang telah dilakukan. Untuk implementasi antarmuka melihat pemesanan travel dapat dilihat pada gambar berikut.



Gambar 5.16 Implementasi Antarmuka Melihat Pemesanan Travel

5.2.5.5 Implementasi Antarmuka Melihat Data Mobil

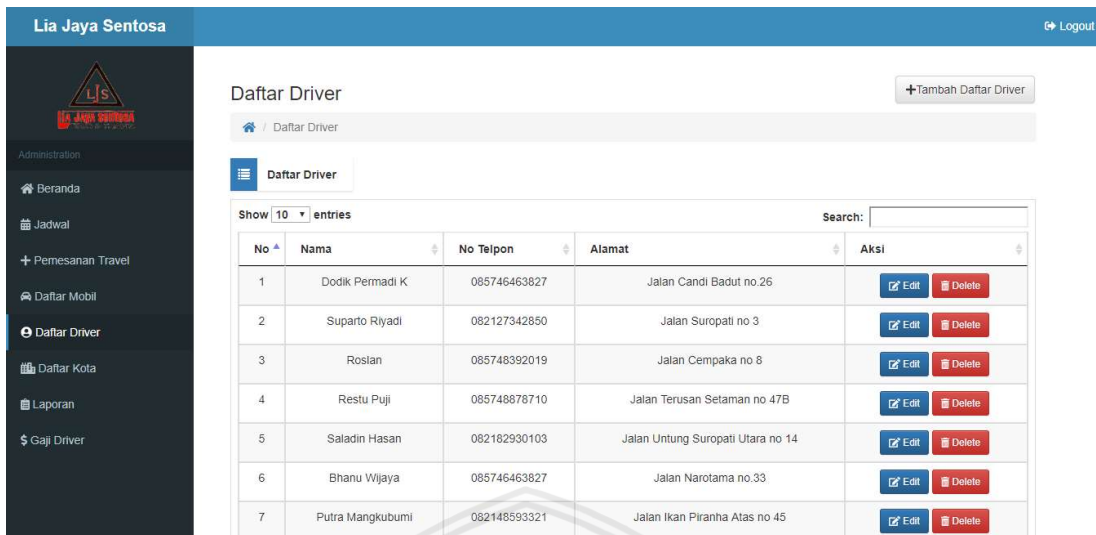
Antarmuka ini merupakan halaman untuk melihat data mobil yang ada di dalam perusahaan lia jaya sentosa. Untuk implementasi antarmuka melihat data mobil dapat dilihat pada gambar berikut.



Gambar 5.17 Implementasi Antarmuka Melihat Data Mobil

5.2.5.6 Implementasi Antarmuka Melihat Data Driver

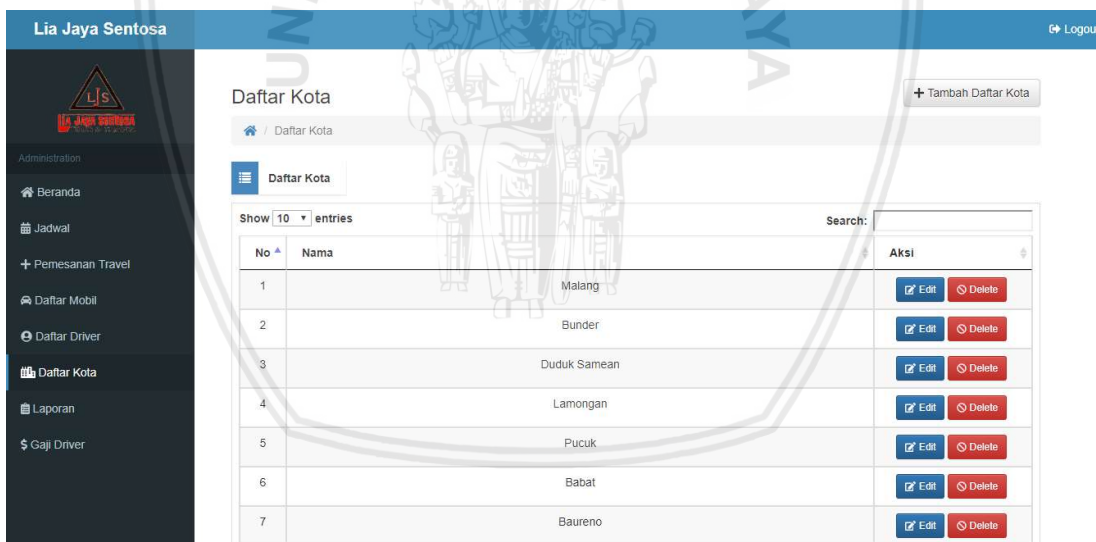
Antarmuka ini merupakan halaman untuk melihat data driver yang ada di dalam perusahaan lia jaya sentosa. Untuk implementasi antarmuka melihat data driver dapat dilihat pada gambar berikut.



Gambar 5.18 Implementasi Antarmuka Melihat Data Driver

5.2.5.7 Implementasi Antarmuka Melihat Data Kota

Antarmuka ini merupakan halaman untuk melihat data kota yang dilayani dalam pemesanan travel. Untuk implementasi antarmuka melihat data kota dapat dilihat pada gambar berikut.

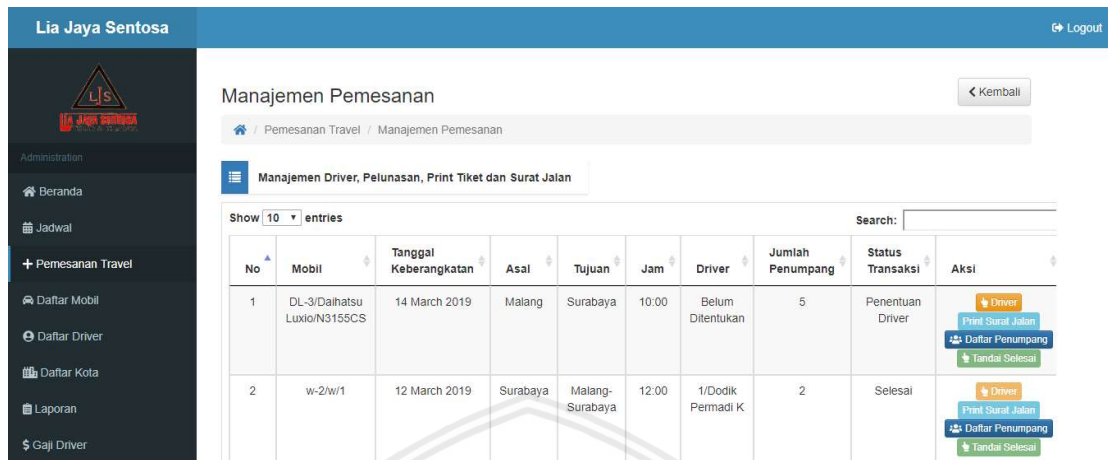


Gambar 5.19 Implementasi Antarmuka Melihat Data Kota

5.2.5.8 Implementasi Antarmuka Melihat Manajemen Pemesanan

Antarmuka ini merupakan halaman untuk melihat manajemen pemesanan yang ada di dalam perusahaan lia jaya sentosa. Dalam halaman manajemen pemesanan, operator dapat mengubah driver, mengubah status keberangkatan, melihat daftar

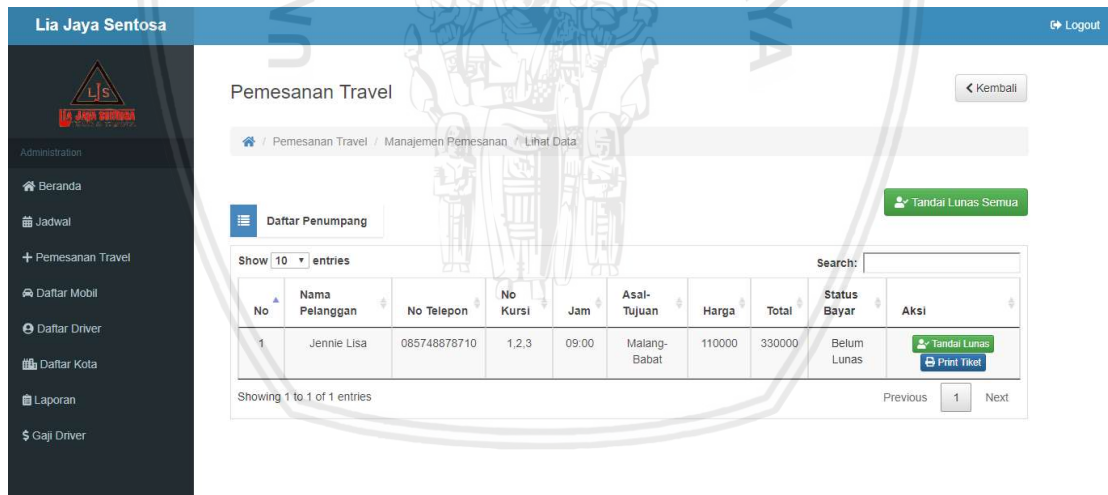
penumpang dan mencetak surat jalan. Untuk implementasi antarmuka melihat manajemen pemesanan dapat dilihat pada gambar berikut.



Gambar 5.20 Implementasi Antarmuka Melihat Manajemen Pemesanan

5.2.5.9 Implementasi Antarmuka Melihat Daftar Penumpang

Antarmuka ini merupakan halaman untuk melihat daftar penumpang yang ada di dalam sebuah mobil. Untuk implementasi antarmuka melihat daftar penumpang dapat dilihat pada gambar berikut.



Gambar 5.21 Implementasi Antarmuka Melihat Daftar Penumpang

5.2.5.10 Implementasi Antarmuka Melihat Laporan Keuangan

Antarmuka ini merupakan halaman untuk melihat laporan keuangan setelah sebuah keberangkatan ditandai selesai. Untuk implementasi antarmuka melihat laporan keuangan dapat dilihat pada gambar berikut.

Lia Jaya Sentosa Logout

Laporan Keuangan Pemesanan Travel Cetak Laporan Keuangan

Laporan Keuangan

Filter Berdasarkan Bulan-Tahun

Show 10 entries

No	Tanggal Laporan	Mobil	Asal	Tujuan	Driver	Pendapatan	Gaji Driver	Keuntungan	No Best	Jumlah Pemes
1	21 December 2018	DL-4Daihatsu LuxioN6291BL	Malang	Gresik	1/Doek Permadi K	300000	49500	250500	2,3,4	
2	14 December 2018	DL-6Daihatsu LuxioN3581CK	Malang	Madun	8/Putra Mangubumi	660000	99000	561000	1,2,4,5,6,7	
3	15 January 2019	DL-1Daihatsu LuxioN4446BM	Malang	Lamongan	9/Doek Kusuma	440000	66000	374000	1,2,4,5	
4	14 December 2018	DL-4Daihatsu LuxioN6291BL	Malang	Gresik	2/Suparto Riyadi	500000	82500	417500	2,3,4,1,5	
5	19 December 2018	SA-9Suzuki APV/N6526CJ	Malang	Purwodadi	2/Suparto Riyadi	585000	87750	497250	3,4,7	
6	14 December 2018	DL-5Daihatsu LuxioN6728BK	Malang	Surabaya	1/Doek Permadi K	340000	38000	302000	1,4,7	

Showing 1 to 6 of 6 entries

Gambar 5.22 Implementasi Antarmuka Laporan Keuangan

5.2.5.11 Implementasi Antarmuka Melihat Gaji Driver

Antarmuka ini merupakan halaman untuk melihat gaji driver. Untuk implementasi antarmuka melihat gaji driver dapat dilihat pada gambar berikut.

Lia Jaya Sentosa Logout

Gaji Driver Cetak Gaji Driver

Gaji Driver

Filter Berdasarkan Bulan-Tahun

Show 10 entries

No	Tanggal Gaji	Driver	Asal	Tujuan	Mobil	Gaji Driver
1	14 December 2018	1/Doek Permadi K	Malang	Surabaya	DL-5Daihatsu LuxioN6728BK	30000
2	15 December 2018	2/Suparto Riyadi	Malang	Purwodadi	SA-9Suzuki APV/N6526CJ	87750
3	14 December 2018	2/Suparto Riyadi	Malang	Gresik	DL-4Daihatsu LuxioN6291BL	82500
4	15 January 2019	9/Doek Kusuma	Malang	Lamongan	DL-1Daihatsu LuxioN4446BM	66000
5	14 December 2018	8/Putra Mangubumi	Malang	Madun	DL-6Daihatsu LuxioN3581CK	99000
6	21 December 2018	1/Doek Permadi K	Malang	Gresik	DL-4Daihatsu LuxioN6291BL	49500

Showing 1 to 6 of 6 entries

Gambar 5.23 Implementasi Gaji Driver

5.2.5.12 Implementasi Antarmuka Menambah Jadwal Travel Harian

Antarmuka ini merupakan halaman untuk menambah jadwal travel harian. Di dalam proses penambahan jadwal travel harian, setiap mobil dapat dialokasikan. Untuk implementasi antarmuka menambah jadwal travel harian dapat dilihat pada gambar berikut.

The screenshot shows a web interface for adding a daily travel schedule. The page title is 'Tambah Jadwal Travel'. Below the title, there is a breadcrumb trail: 'Jadwal Travel > Tambah'. A blue button labeled 'Masukkan Data Jadwal Keberangkatan' is at the top left of the form area. The form contains several input fields: 'Asal' with a dropdown menu showing 'Pilih Asal', 'Tujuan' with a dropdown menu showing 'Pilih Tujuan', 'Mobil' with a dropdown menu showing 'Pilih Mobil', 'Jam Keberangkatan' with a time selection field, 'Harga' with a currency symbol '\$' and a text input field, and 'Hari Keberangkatan' with radio button options for 'Senin', 'Selasa', 'Rabu', 'Kamis', 'Jumat', 'Sabtu', 'Minggu', and 'Centang Semua'. At the bottom right, there are two buttons: 'Simpan' (Save) and 'Batal' (Cancel). A large watermark of the Universitas Brawijaya logo is overlaid on the form.

Gambar 5.24 Implementasi Antarmuka Menambah Jadwal Travel Harian

5.2.5.13 Implementasi Antarmuka Mengubah Jadwal Travel Harian

Antarmuka ini merupakan halaman untuk mengubah jadwal travel harian. Di dalam proses pengubahan jadwal travel harian, setiap mobil dapat dialokasikan. Untuk implementasi antarmuka mengubah jadwal travel harian dapat dilihat pada gambar berikut.

Lia Jaya Sentosa Logout

Ubah Jadwal Travel Kembali

Ubah Jadwal Travel / Ubah

Ubah Data Jadwal Keberangkatan

Asal
A Malang

Tujuan
A Bunder

Mobil
R SA-Tribuati APV/ND837CL

Jam Keberangkatan
19.00

Harga
\$ 120000

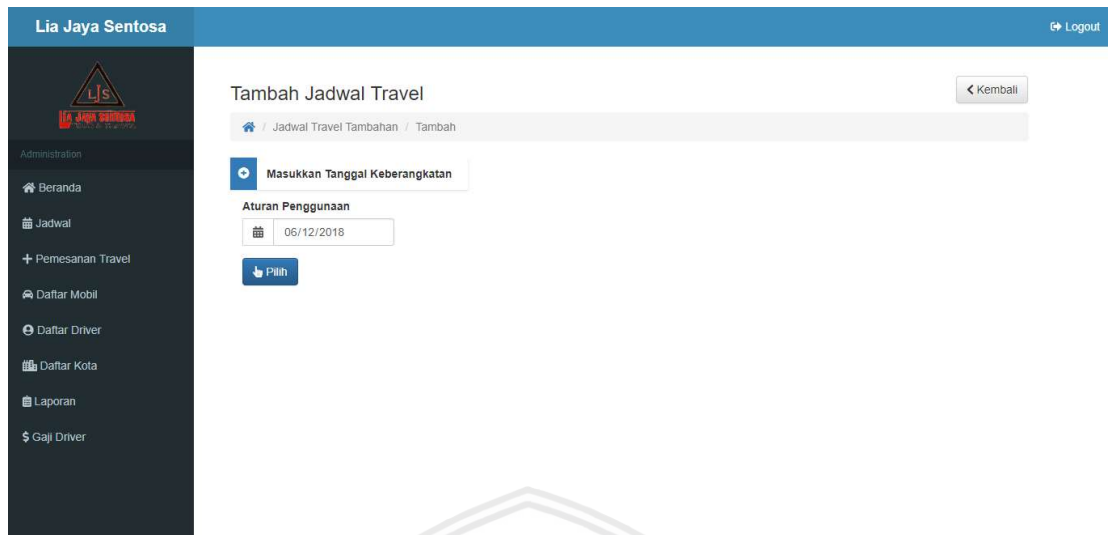
Hari Keberangkatan
 Senin
 Selasa
 Rabu
 Kamis
 Jumat
 Sabtu
 Minggu
 Centang Semua

Simpan Hatal

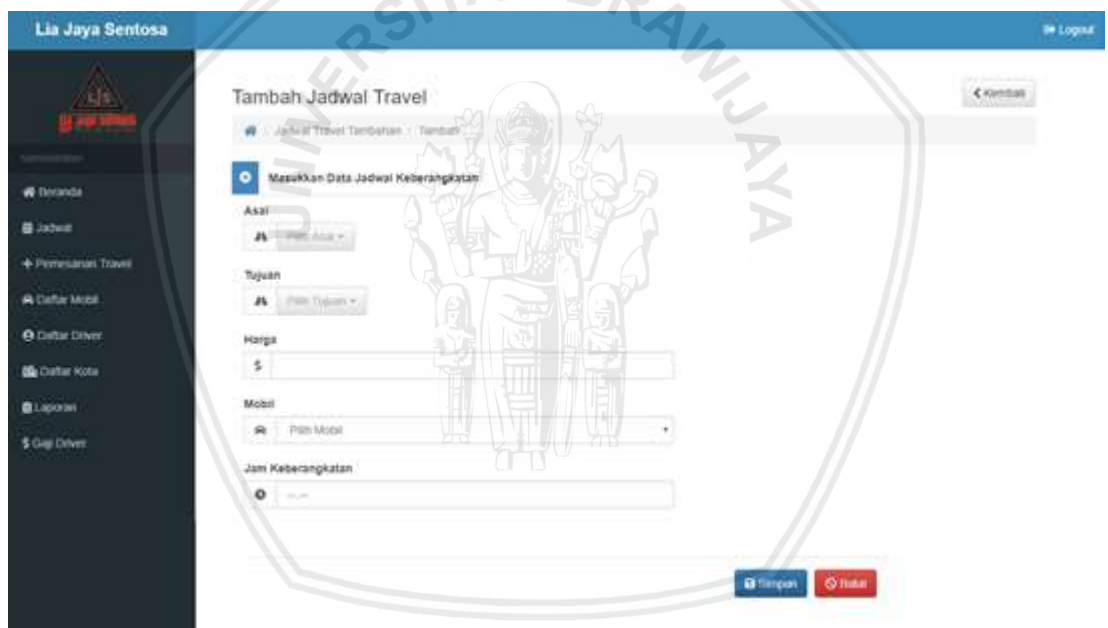
Gambar 5.25 Implementasi Antarmuka Mengubah Jadwal Travel Harian

5.2.5.14 Implementasi Antarmuka Tambah Jadwal Travel Tambahan

Antarmuka ini merupakan halaman untuk menambah jadwal travel tambahan. Di dalam proses penambahan jadwal travel tambahan, setiap mobil yang sudah dialokasikan di dalam jadwal travel harian tidak dapat dialokasikan lagi di dalam jadwal travel tambahan ini. Untuk implementasi antarmuka menambah jadwal travel tambahan dapat dilihat pada gambar berikut.



Gambar 5.26 Implementasi Antarmuka Menambah Jadwal Travel Tambahan Bagian 1



Gambar 5.27 Implementasi Antarmuka Menambah Jadwal Travel Tambahan Bagian 2

5.2.5.15 Implementasi Antarmuka Mengubah Jadwal Travel Tambahan

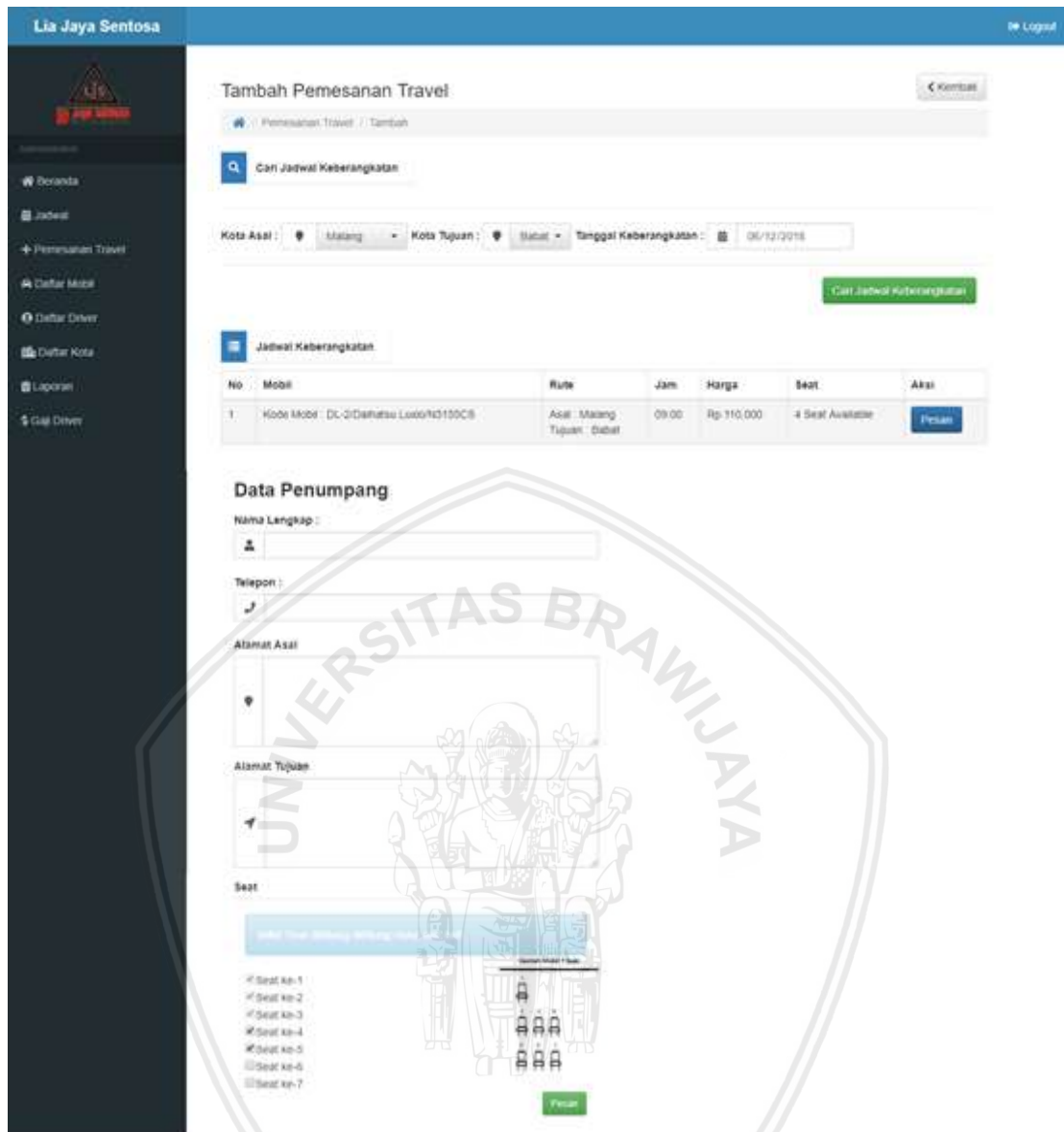
Antarmuka ini merupakan halaman untuk mengubah jadwal travel tambahan. Di dalam proses pengubahan jadwal travel tambahan, setiap mobil yang sudah dialokasikan di dalam jadwal travel harian tidak dapat dialokasikan lagi di dalam jadwal travel tambahan ini. Untuk implementasi antarmuka mengubah jadwal travel tambahan dapat dilihat pada gambar berikut.

The screenshot shows a web interface for editing travel schedules. The header includes the company name 'Lia Jaya Sentosa' and a 'Logout' link. A sidebar on the left lists navigation options: Beranda, Jadwal, Pemesanan Travel, Daftar Mobil, Daftar Driver, Daftar Kota, Laporan, and Gaji Driver. The main content area is titled 'Edit Jadwal Travel Tambahan' and contains a form with the following fields: 'Asal' (Mtatog), 'Tujuan' (Batal), 'Mobi' (DL-1/Dishatsu Loko/64448JM), 'Jam Keberangkatan' (10:30), 'Harga' (110000), and 'Tgl Keberangkatan' (02/12/2018). A 'Kembali' button is located in the top right corner of the form area.

Gambar 5.28 Implementasi Antarmuka Mengubah Jadwal Travel Tambahan

5.2.5.16 Implementasi Antarmuka Menambah Pemesanan Travel

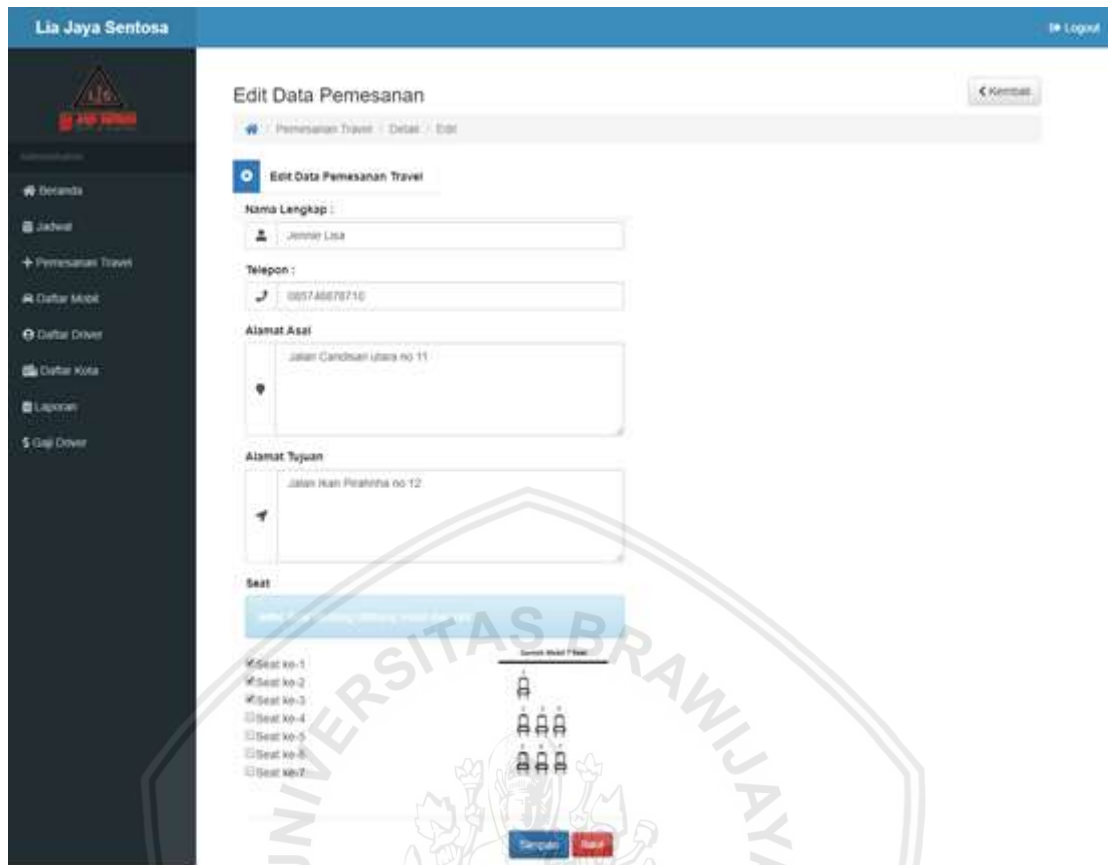
Antarmuka ini merupakan halaman untuk menambah pemesanan travel. Proses yang dilakukan dalam antarmuka ini adalah mencari jadwal keberangkatan yang dapat dipesan, mengisi data penumpang kemudian menyimpannya ke dalam database. Untuk implementasi antarmuka menambah pemesanan travel dapat dilihat pada gambar berikut.



Gambar 5.29 Implementasi Antarmuka Menambah Pemesanan Travel

5.2.5.17 Implementasi Antarmuka Mengubah Pemesanan Travel

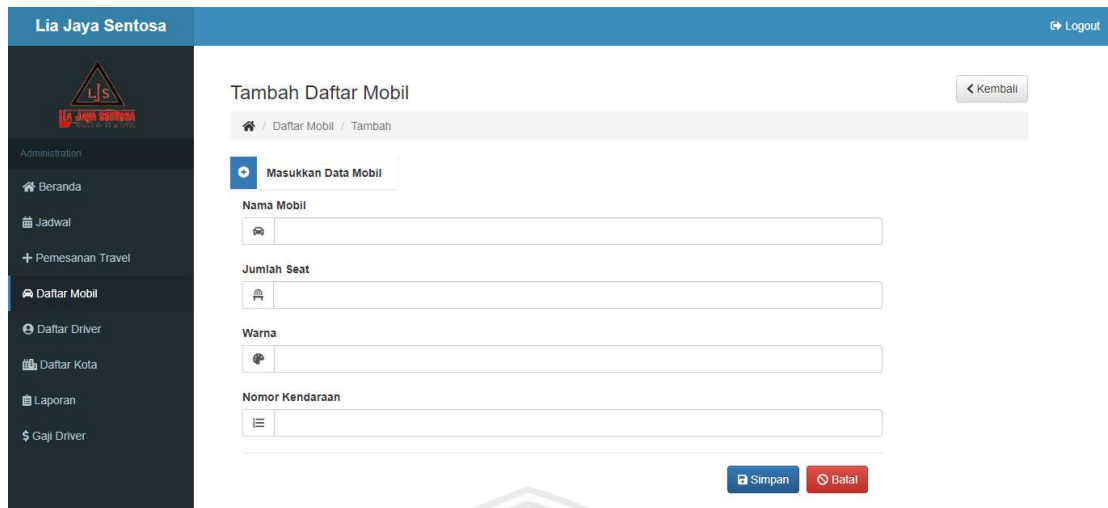
Antarmuka ini merupakan halaman untuk mengubah pemesanan travel. Field yang dapat dirubah dalam halaman ini hanyalah field data penumpang. Untuk implementasi antarmuka mengubah pemesanan travel dapat dilihat pada gambar berikut.



Gambar 5.30 Implementasi Antarmuka Mengubah Pemesanan Travel

5.2.5.18 Implementasi Antarmuka Menambah Data Mobil

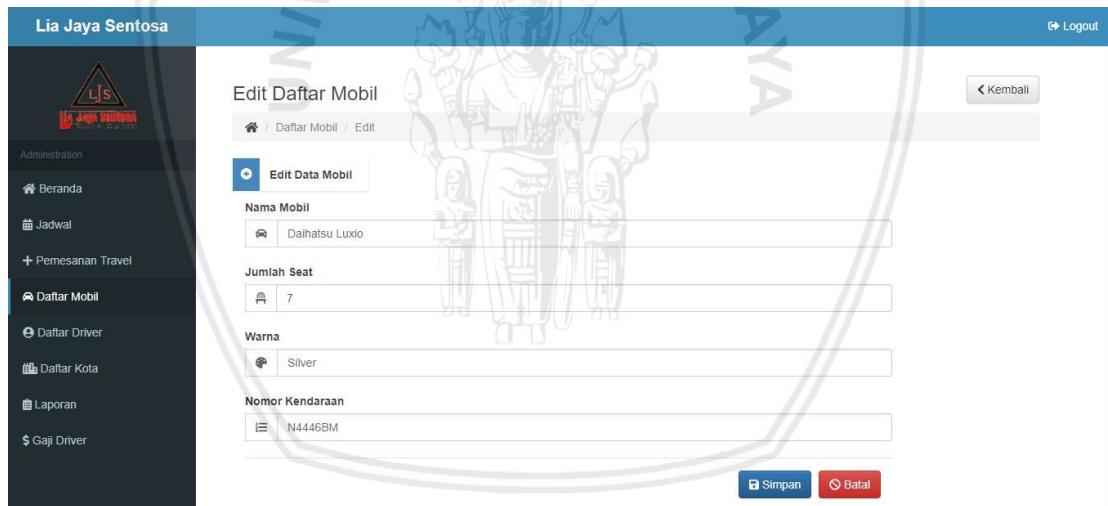
Antarmuka ini merupakan halaman untuk menambah data mobil yang dimiliki oleh perusahaan ke dalam database. Untuk implementasi antarmuka menambah data mobil dapat dilihat pada gambar berikut.



Gambar 5.31 Implementasi Antarmuka Menambah Data Mobil

5.2.5.19 Implementasi Antarmuka Mengubah Data Mobil

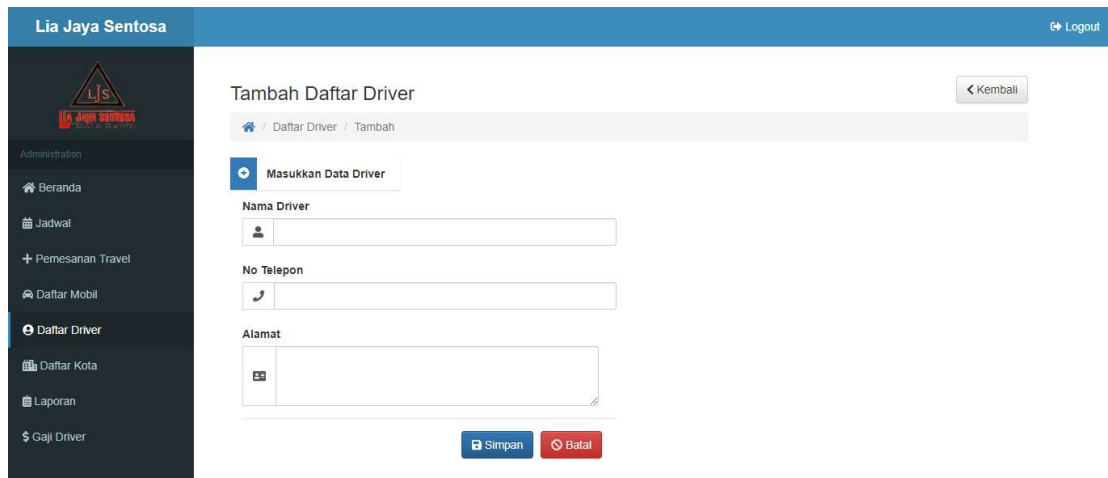
Antarmuka ini merupakan halaman untuk mengubah data mobil yang dimiliki oleh perusahaan ke dalam database. Untuk implementasi antarmuka mengubah data mobil dapat dilihat pada gambar berikut.



Gambar 5.32 Implementasi Antarmuka Mengubah Data Mobil

5.2.5.20 Implementasi Antarmuka Menambah Data Driver

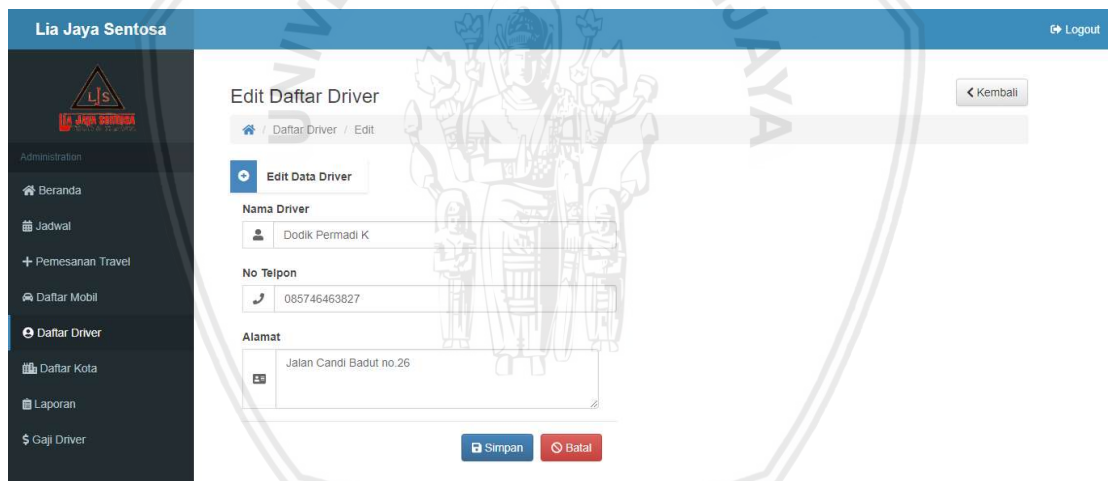
Antarmuka ini merupakan halaman untuk menambah data driver yang dimiliki oleh perusahaan ke dalam database. Untuk implementasi antarmuka menambah data driver dapat dilihat pada gambar berikut.



Gambar 5.33 Implementasi Antarmuka Menambah Data Driver

5.2.5.21 Implementasi Antarmuka Mengubah Data Driver

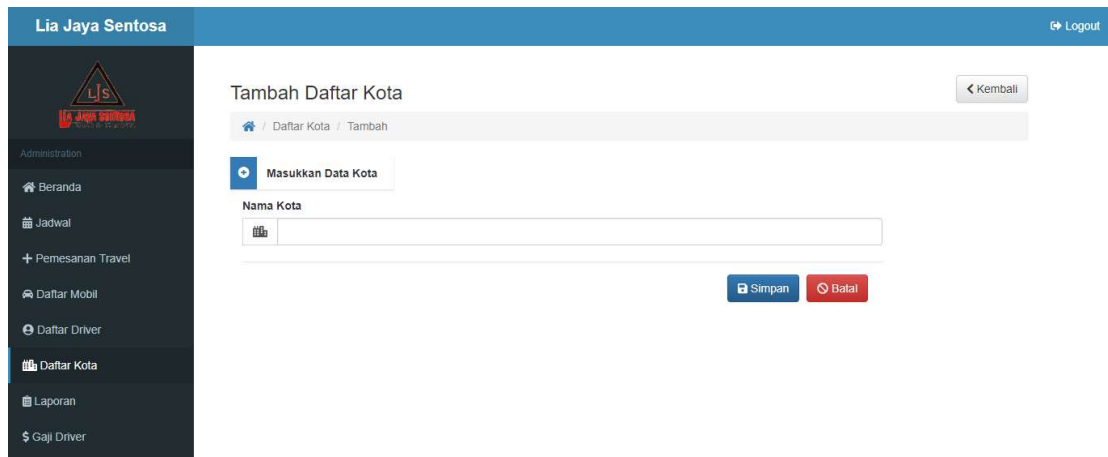
Antarmuka ini merupakan halaman untuk mengubah data driver yang dimiliki oleh perusahaan ke dalam database. Untuk implementasi antarmuka mengubah data driver dapat dilihat pada gambar berikut.



Gambar 5.34 Implementasi Antarmuka Mengubah Data Driver

5.2.5.22 Implementasi Antarmuka Menambah Data Kota

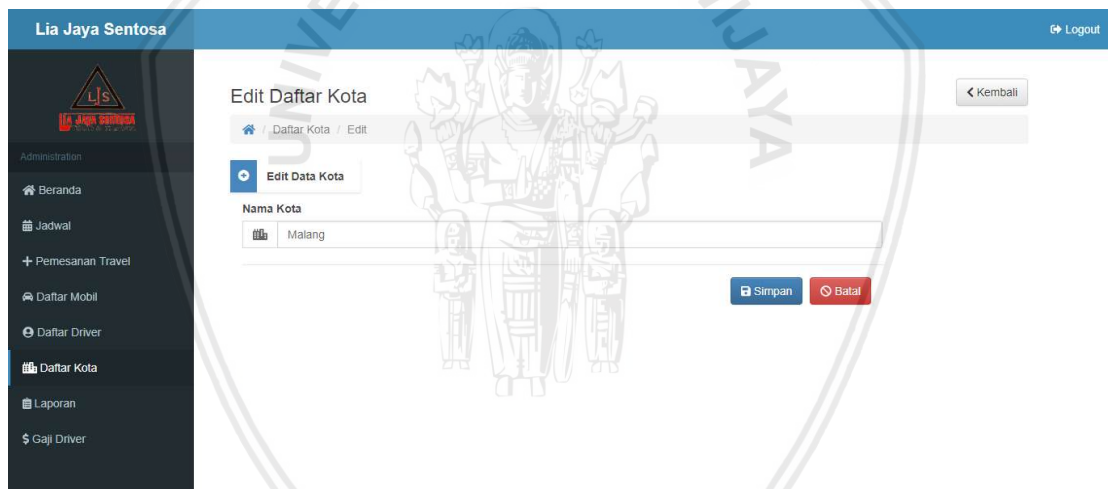
Antarmuka ini merupakan halaman untuk menambah data kota yang dilayani oleh perusahaan ke dalam database. Untuk implementasi antarmuka menambah data kota dapat dilihat pada gambar berikut.



Gambar 5.35 Implementasi Antarmuka Menambah Data Kota

5.2.5.23 Implementasi Antarmuka Mengubah Data Kota

Antarmuka ini merupakan halaman untuk mengubah data kota yang dilayani oleh perusahaan ke dalam database. Untuk implementasi antarmuka mengubah data kota dapat dilihat pada gambar berikut.



Gambar 5.36 Implementasi Antarmuka Mengubah Data Kota

5.2.5.24 Implementasi Antarmuka Menambah Laporan Keuangan

Antarmuka ini merupakan halaman untuk menambah data kota yang dilayani oleh perusahaan ke dalam database. Untuk implementasi antarmuka menambah laporan keuangan dapat dilihat pada gambar berikut.

Tambah Laporan Logout

[← Laporan](#) [Tambah](#)

Masukkan Laporan Keuangan

Tanggal Laporan

Mobi

Driver

Nama Penumpang	Rute	No Seat	Jumlah Pemesanan	Harga	Total	Check
Riky Brana	Klateng-Surabaya	6.7	2	80000	160000	✓

No Seat

Jumlah Pemesanan

Total Harga

Persentase Gaji Driver

Gaji Driver

Keterangan

Gambar 5.37 Implementasi Antarmuka Menambah Laporan Keuangan

5.2.5.25 Implementasi Antarmuka Mengubah Laporan Keuangan

Antarmuka ini merupakan halaman untuk mengubah data kota yang dilayani oleh perusahaan ke dalam database. Untuk implementasi antarmuka mengubah laporan keuangan dapat dilihat pada gambar berikut.

Edit Laporan Kembali

Laporan - Edit

Masukkan Laporan Keuangan

Tanggal Laporan: 21 December 2018

Mobil: DU-5/Dametu Lixo/N67288K

Driver: F30Stameli Aboq.

Nama Penumpang	Rute	No Seat	Jumlah Pemesanan	Harga	Total	Check
Ilby Brana	Malang-Surabaya	6.7	2	80000	160000	✓

No Seat: 6.7

Jumlah Pemesanan: 2

Total Harga: Rp. 160000

Persentase Gaji Driver: 15%

Gaji Driver: Rp. 24000


Keterangan:

Simpan Batal

Gambar 5.38 Implementasi Antarmuka Mengubah Laporan Keuangan

5.2.5.26 Implementasi Antarmuka Mencetak Surat Jalan

Antarmuka ini merupakan halaman untuk mencetak surat jalan. Untuk implementasi antarmuka mencetak surat jalan dapat dilihat pada gambar berikut.


Lia Jaya Sentosa
 Jl. Trunojoyo 46
 No Telpn : (0341) 331402, 333062, 356356 WA : 081 233 14 900

Surat Jalan

Tanggal : 2018-12-06
 Kode Mobil : DL-2/Daihatsu Luxio/N3155CS
 Nama Driver : Belum Ditentukan
 Tujuan : Babat
 Jam Keberangkatan : 09:00

Data Penumpang

No	Nama Pelanggan	No Telepon	Alamat Asal	Alamat Tujuan	No Kursi
1	Jennie Lisa	085748878710	Jalan Candisari utara no 11, Malang	Jalan Ikan Pirahnha no 12, Babat	1,2,3
Operator		Supir		Satpam	

Gambar 5.39 Implementasi Antarmuka Mencetak Surat Jalan

5.2.5.27 Implementasi Antarmuka Mencetak Tiket Travel

Antarmuka ini merupakan halaman untuk mencetak tiket travel. Untuk implementasi antarmuka mencetak tiket travel dapat dilihat pada gambar berikut.


Lia Jaya Sentosa
 Jl. Trunojoyo 46
 No Telpn : (0341) 331402, 333062, 356356 WA : 081 233 14 900

Tiket Travel Penumpang

Id. Tiket : 300
 Mobil : DL-2/Daihatsu Luxio/N3155CS
 Nama Penumpang : Jennie Lisa
 Alamat Asal : Jalan Candisari utara no 11
 Alamat Tujuan : Jalan Ikan Pirahnha no 12
 Telp. : 085748878710

Dari : Malang	Brgkt Hari : Kamis	Tgl : 2018-12-06
Tujuan : Babat	Jam : 09:00	No Kursi : 1,2,3
Jumlah Penumpang : 3 orang x Harga	: Rp. 110000	= Total : Rp. 330000

1/1

Gambar 5.40 Implementasi Antarmuka Mencetak Tiket Travel

5.2.5.28 Implementasi Antarmuka Mencetak Gaji Driver

Antarmuka ini merupakan halaman untuk mencetak gaji driver. Untuk implementasi antarmuka mencetak gaji driver dapat dilihat pada gambar berikut.



Lia Jaya Sentosa
 Jl.Trunojoyo 46
 No Telpn : (0341) 331402, 333062, 356356 WA : 081 233 14 900

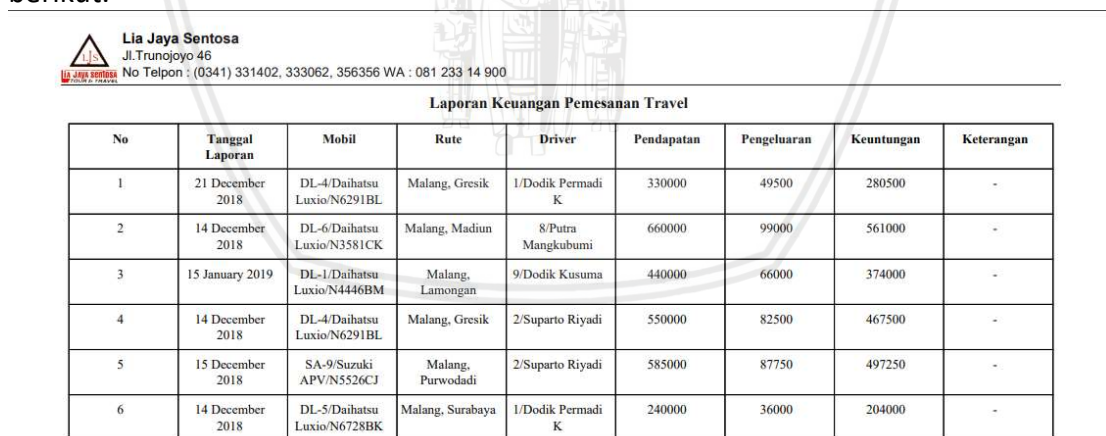
Laporan Gaji Driver

No	Tanggal Gaji	Mobil	Rute	Driver	Gaji Driver
1	14 December 2018	DL-5/Daihatsu Luxio/N6728BK	Malang, Surabaya	1/Dodik Permadi K	36000
2	15 December 2018	SA-9/Suzuki APV/N5526CJ	Malang, Purwodadi	2/Suparto Riyadi	87750
3	14 December 2018	DL-4/Daihatsu Luxio/N6291BL	Malang, Gresik	2/Suparto Riyadi	82500
4	15 January 2019	DL-1/Daihatsu Luxio/N4446BM	Malang, Lamongan	9/Dodik Kusuma	66000
5	14 December 2018	DL-6/Daihatsu Luxio/N3581CK	Malang, Madiun	8/Putra Mangkubumi	99000
6	21 December 2018	DL-4/Daihatsu Luxio/N6291BL	Malang, Gresik	1/Dodik Permadi K	49500

Gambar 5.41 Implementasi Antarmuka Mencetak Gaji Driver

5.2.5.29 Implementasi Antarmuka Mencetak Laporan Keuangan

Antarmuka ini merupakan halaman untuk mencetak laporan keuangan. Untuk implementasi antarmuka mencetak laporan keuangan dapat dilihat pada gambar berikut.



Lia Jaya Sentosa
 Jl.Trunojoyo 46
 No Telpn : (0341) 331402, 333062, 356356 WA : 081 233 14 900

Laporan Keuangan Pemesanan Travel

No	Tanggal Laporan	Mobil	Rute	Driver	Pendapatan	Pengeluaran	Keuntungan	Keterangan
1	21 December 2018	DL-4/Daihatsu Luxio/N6291BL	Malang, Gresik	1/Dodik Permadi K	330000	49500	280500	-
2	14 December 2018	DL-6/Daihatsu Luxio/N3581CK	Malang, Madiun	8/Putra Mangkubumi	660000	99000	561000	-
3	15 January 2019	DL-1/Daihatsu Luxio/N4446BM	Malang, Lamongan	9/Dodik Kusuma	440000	66000	374000	-
4	14 December 2018	DL-4/Daihatsu Luxio/N6291BL	Malang, Gresik	2/Suparto Riyadi	550000	82500	467500	-
5	15 December 2018	SA-9/Suzuki APV/N5526CJ	Malang, Purwodadi	2/Suparto Riyadi	585000	87750	497250	-
6	14 December 2018	DL-5/Daihatsu Luxio/N6728BK	Malang, Surabaya	1/Dodik Permadi K	240000	36000	204000	-

Gambar 5.42 Implementasi Antarmuka Mencetak Laporan Keuangan

BAB 6 PENGUJIAN

Pengujian dilakukan setelah proses implementasi selesai dilakukan. Pada tahap pengujian dilakukan pengecekan kembali apakah sistem yang telah diimplementasikan sudah sesuai dengan analisis kebutuhan dan perancangan yang sudah didefinisikan sebelumnya. Dalam tahap ini akan dilakukan pengujian unit, pengujian integrasi, pengujian validasi.

6.1 Pengujian *Unit*

Pengujian *unit* merupakan sebuah pengujian yang dilakukan untuk mengetahui apakah *unit* atau komponen dari suatu perangkat yang diimplementasikan sudah sesuai dengan tujuan dari fungsionalitas itu sendiri. Pengujian ini menerapkan salah satu metode dari *whitebox testing* yaitu dengan *basis path testing*. Pengujian unit akan dilakukan pada 3 sampel uji yaitu Melihat Grafik Pemesanan, Menghapus Pemesanan Travel, Mencetak Laporan.

6.1.1 Pengujian Unit Melihat Beranda

Berikut adalah algoritma dari operasi melihat grafik pemesanan.

Nama class : Beranda

Nama method : index()

Tabel 6.1 Algoritme Melihat Grafik Beranda

No	Pseudocode
1.	<p><i>Inisiasi variable pencarian = ""</i> <i>Inisiasi variable bulan = ""</i> <i>Inisiasi variable tahun = ""</i> <i>Inisiasi array data dengan index status = 0</i></p>
2.	<p><i>Jika parameter dengan nama in sama dengan 1, start branch</i></p>
3.	<p><i>Mengambil nilai dari inputan operator dan memberikan nilainya ke variable pencarian</i></p> <p><i>Mengubah nilai dari variable pencarian ke dalam array pecahpencarian berdasarkan karakter "-"</i></p> <p><i>Memberikan nilai ke dalam variable bulan berdasarkan nilai dari array pecahpencarian index ke 1</i></p> <p><i>Memberikan nilai ke dalam variable tahun berdasarkan nilai dari array pecahpencarian index ke 0</i></p>
4.	<p><i>Else</i></p>

Tabel 6.1 Algoritme Melihat Grafik Beranda

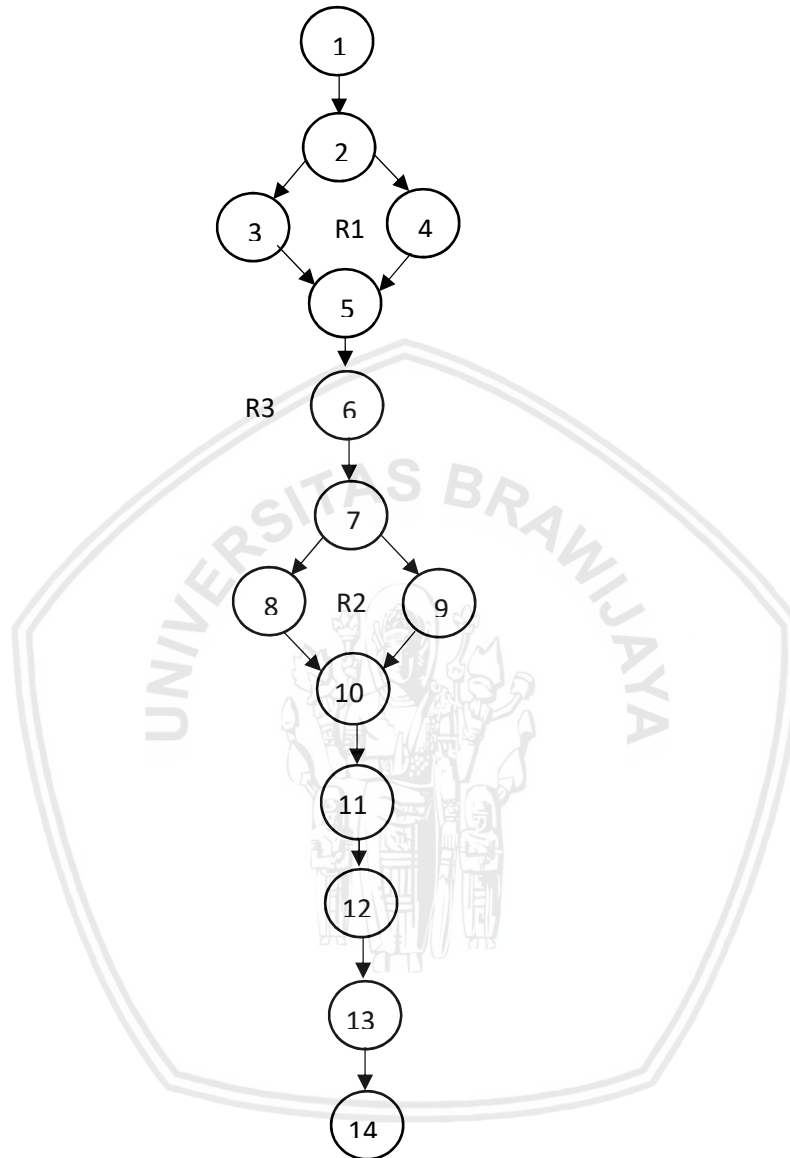
No	Pseudocode
	<p>Memberikan nilai ke dalam variable bulan dengan nilai bulan saat ini</p> <p>Memberikan nilai ke dalam variable tahun dengan nilai tahun saat ini</p>
5.	End branch
6.	<p>array offset = 0</p> <p>array ha = 1</p> <p>array modules = view beranda</p> <p>array title = Beranda</p> <p>array tujuan = hasil query pengambilan data jumlah rute yang paling sering dipesan berdasarkan parameter bulan dan tahun</p>
7.	Jika jumlah elemen dari array tujuan lebih besar daripada 0, start branch
8.	Array status = 0
9.	Else Array status sama dengan 1
10.	End branch
11.	<p>Array bulan = Nilai date dengan format "F" berdasarkan nilai dari variable tahun dan bulan</p> <p>Array tahun = tahun</p> <p>Array bulen = bulan</p>
12.	<p>Array laporankeuanganjan = mengambil data pendapatan, gaji driver dan keuntungan pada bulan januari kemudian menjumlahkannya</p> <p>Array laporankeuanganfeb = mengambil data pendapatan, gaji driver dan keuntungan pada bulan february kemudian menjumlahkannya</p> <p>Array laporankeuanganmar = mengambil data pendapatan, gaji driver dan keuntungan pada bulan maret kemudian menjumlahkannya</p> <p>Array laporankeuanganapr = mengambil data pendapatan, gaji driver dan keuntungan pada bulan april kemudian menjumlahkannya</p> <p>Array laporankeuanganmei = mengambil data pendapatan, gaji driver dan keuntungan pada bulan mei kemudian menjumlahkannya</p> <p>Array laporankeuanganjun = mengambil data pendapatan, gaji driver dan keuntungan pada bulan juni kemudian menjumlahkannya</p> <p>Array laporankeuanganjul = mengambil data pendapatan, gaji driver dan keuntungan pada bulan juli kemudian menjumlahkannya</p> <p>Array laporankeuanganagu = mengambil data pendapatan, gaji driver dan keuntungan pada bulan agustus kemudian menjumlahkannya</p>

Tabel 6.1 Algoritme Melihat Grafik Beranda

No	Pseudocode
	<p><i>Array laporankeuangansep = mengambil data pendapatan, gaji driver dan keuntungan pada bulan september kemudian menjumlahkannya</i></p> <p><i>Array laporankeuanganokt = mengambil data pendapatan, gaji driver dan keuntungan pada bulan oktober kemudian menjumlahkannya</i></p> <p><i>Array laporankeuangannov = mengambil data pendapatan, gaji driver dan keuntungan pada bulan november kemudian menjumlahkannya</i></p> <p><i>Array laporankeuangandes = mengambil data pendapatan, gaji driver dan keuntungan pada bulan desember kemudian menjumlahkannya</i></p>
13.	<p><i>Array banyakjan = mengambil jumlah pemesanan yang telah terjadi pada bulan januari</i></p> <p><i>Array banyakfeb = mengambil jumlah pemesanan yang telah terjadi pada bulan februari</i></p> <p><i>Array banyakmar = mengambil jumlah pemesanan yang telah terjadi pada bulan maret</i></p> <p><i>Array banyakapr = mengambil jumlah pemesanan yang telah terjadi pada bulan april</i></p> <p><i>Array banyakmei = mengambil jumlah pemesanan yang telah terjadi pada bulan mei</i></p> <p><i>Array banyakjun = mengambil jumlah pemesanan yang telah terjadi pada bulan juni</i></p> <p><i>Array banyakjul = mengambil jumlah pemesanan yang telah terjadi pada bulan juli</i></p> <p><i>Array banyakagu = mengambil jumlah pemesanan yang telah terjadi pada bulan agustus</i></p> <p><i>Array banyaksep = mengambil jumlah pemesanan yang telah terjadi pada bulan september</i></p> <p><i>Array banyakokt = mengambil jumlah pemesanan yang telah terjadi pada bulan oktober</i></p> <p><i>Array banyaknov = mengambil jumlah pemesanan yang telah terjadi pada bulan november</i></p> <p><i>Array banyakdes = mengambil jumlah pemesanan yang telah terjadi pada bulan desember</i></p>
14.	<i>Load view beranda dengan membawa data array</i>

a. *Flow graph*

Berikut merupakan *flow graph* dari algoritma melihat grafik pemesanan



Gambar 6.1 *Flow Graph* Algoritma Melihat Grafik Pemesanan

b. *Cyclomatic Complexity*

Berikut merupakan perhitungan *Cyclomatic Complexity* dari algoritma melihat grafik pemesanan:

1. $V(G) = \text{Total Ruang (R)}$. Total R = 3

2. $V(G) = \text{Jumlah edge} - \text{Jumlah Node} + 2 = 15 - 14 + 2 = 3$

3. $V(G) = \text{Total Percabangan (P)} + 1 = 2 + 1 = 3$

c. *Independent Path*

Berikut merupakan jalur-jalur *independent* dari algoritma melihat grafik pemesanan travel :

1. Jalur 1 = 1 – 2 – 3 – 5 – 6 – 7 – 8 – 10 – 11 – 12 – 13 – 14

2. Jalur 2 = 1 – 2 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 12 – 13 – 14

3. Jalur 3 = 1 – 2 . . 5 – 6 – 7 – 9 – 10 – 11 – 12 – 13 – 14

d. Kasus Uji dan Hasil Pengujian

Berikut merupakan kasus uji dan hasil uji dari algoritma melihat grafik pemesanan berdasarkan jalur *independent* yang dihasilkan :

Tabel 6.2 Test Case Melihat Grafik Pemesanan

No.	No Jalur	Prosedur Uji	Expected Result	Result	Status
1.	1	Nilai parameter 1, variable pencarian bernilai "Januari 2019", jumlah elemen array tujuan bernilai 1	Berhasil menampilkan halaman Beranda dengan memuat grafik pemesanan yang sesuai dengan bulan dan tahun yang diinputkan.	Berhasil menampilkan halaman Beranda dengan memuat grafik pemesanan yang sesuai dengan bulan dan tahun yang diinputkan.	Valid
2.	2	Nilai parameter 2, variable pencarian bernilai bulan dan tahun saat ini yaitu "Desember 2018", jumlah elemen array tujuan bernilai 3	Berhasil menampilkan halaman Beranda dengan memuat grafik	Berhasil menampilkan halaman Beranda dengan memuat grafik	Valid

Tabel 6.2 Test Case Melihat Grafik Pemesanan

No.	No Jalur	Prosedur Uji	Expected Result	Result	Status
			pemesanan yang sesuai dengan bulan dan tahun saat ini.	pemesanan yang sesuai dengan bulan dan tahun saat ini.	
3.	3	Nilai parameter 2, variable pencarian bernilai bulan dan tahun saat ini yaitu "Desember 2018", jumlah elemen array tujuan bernilai 0	Berhasil menampilkan view Beranda dengan memuat grafik pemesanan dengan data yang kosong.	Berhasil menampilkan view Beranda dengan memuat grafik pemesanan dengan data yang kosong.	Valid

6.1.2 Pengujian Unit Menghapus Pemesanan Travel

Berikut adalah algoritma dari operasi menghapus pemesanan travel.

Nama class : adminTravel

Nama method : deletePTravel()

Tabel 6.3 Algoritme Menghapus Pemesanan Travel

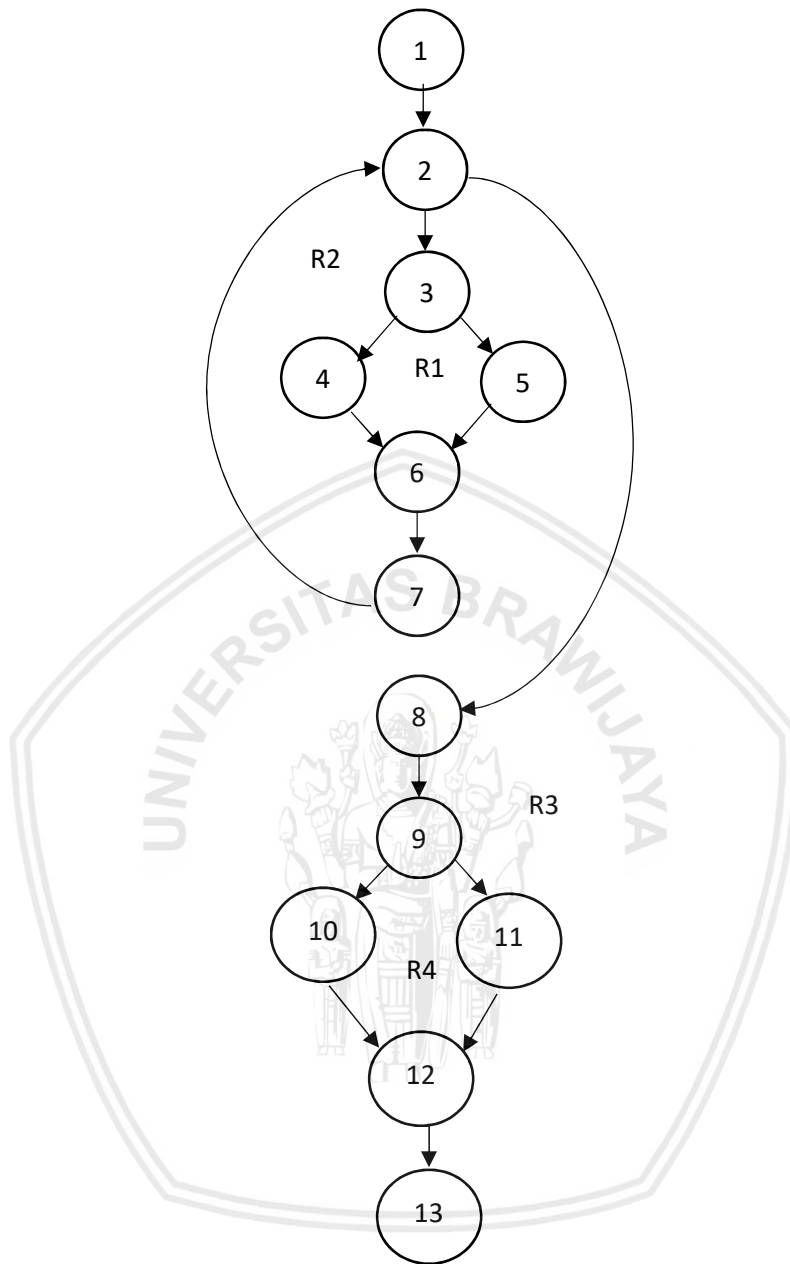
No	Pseudocode
1.	<i>pemesanan = hasil query mengambil data pemesanan berdasarkan id kursi = hasil query data keberangkatan berdasarkan kode mobil, tgl keberangkatan dan jam keberangkatan pisahPemesanan = memecah nilai noKursi pada object pemesanan ke dalam array berdasarkan karakter ',' pisahKursi = memecah nilai noKursi pada object kursi ke dalam array berdasarkan karakter ',' dataTersisa = ""</i>
2.	<i>Untuk setiap elemen yang ada pada array pisahKursi sebagai variable value maka lakukan, start loop</i>

Tabel 6.3 Algoritme Menghapus Pemesanan Travel

No	Pseudocode
3.	<i>Jika nilai variable value sama dengan salah satu elemen pada array pisahPemesanan maka lakukan, start if</i>
4.	<i>Tulis "Data Sama"</i>
5.	<i>Else</i> <i>dataTersisa = nilai variable value</i>
6.	<i>End if</i>
7.	<i>End loop</i>
8.	<i>pecahDataTersisa = memecah nilai variabel dataTersisa menjadi array berdasarkan karakter ','</i> <i>Hapus elemen terakhir dari array</i> <i>dataTersisa = menyatukan elemen array menjadi string berdasarkan karakter ','</i>
9.	<i>Jika jumlah elemen array pecahDataTersisa lebih dari 0 maka lakukan, start if</i>
10.	<i>Array noKursi = nilai variable dataTersisa</i> <i>Update data pada tabel keberangkatan berdasarkan idKeberangkatan dan array data</i>
11.	<i>Else</i> <i>Delete data pada tabel keberangkatan berdasarkan idKeberangkatan</i>
12.	<i>End if</i>
13.	<i>Delete data pada tabel pemesanan berdasarkan id</i> <i>Setflashdata bahwa Data telah dihapus</i> <i>Mengarahkan ke view adminTravel/viewTravel</i>

a. *Flow graph*

Berikut merupakan *flow graph* dari algoritma menghapus pemesanan travel



Gambar 6.2 Flow Graph Algoritma Menghapus Pemesanan Travel

b. Cyclomatic Complexity

Berikut merupakan perhitungan *Cyclomatic Complexity* dari algoritma menghapus pemesanan travel:

1. $V(G) = \text{Total Ruang (R)}$. Total R = 4

2. $V(G) = \text{Jumlah edge} - \text{Jumlah Node} + 2 = 15 - 13 + 2 = 4$

3. $V(G) = \text{Total Percabangan (P)} + 1 = 3 + 1 = 4$

c. *Independent Path*

Berikut merupakan jalur-jalur *independent* dari algoritma menghapus pemesanan travel :

1. Jalur 1 = 1 - 2 - 8 - 9 - 10 - 12 - 13

2. Jalur 2 = 1 - 2 - 8 - 9 - 11 - 12 - 13

3. Jalur 3 = 1 - 2 - 3 - 4 - 6 - 7 - 2 - 8 - 9 - . . . - 12 - 13

4. Jalur 4 = 1 - 2 - 3 - 5 - 6 - 7 - 2 - 8 - 9 - . . . - 12 - 13

d. Kasus Uji dan Hasil Pengujian

Berikut merupakan kasus uji dan hasil uji dari algoritma menghapus pemesanan travel berdasarkan jalur *independent* yang dihasilkan :

Tabel 6.4 Test Case Menghapus Pemesanan Travel

No.	No Jalur	Prosedur Uji	Expected Result	Result	Status
1.	1	Nilai parameter id bernilai 2000, jumlah elemen array pecahDataTersisa = 1	Tidak ada data yang terhapus	Tidak ada data yang terupdate	Valid
2.	2	Nilai parameter id bernilai 2000, jumlah elemen array pecahDataTersisa = 0	Tidak ada data yang terhapus	Tidak ada data yang terhapus	Valid
3.	3	Nilai parameter id bernilai 423, value=2,3,4 jumlah elemen array pecahDataTersisa = 1	Berhasil mengupdate data keberangkatan	Berhasil mengupdate data keberangkatan dan menghapus data pemesanan	Valid
4.	4	Nilai parameter id bernilai 423, value=1,5 jumlah	Berhasil menghapus data	Berhasil menghapus data	Valid

Tabel 6.4 Test Case Menghapus Pemesanan Travel

No.	No Jalur	Prosedur Uji	Expected Result	Result	Status
		elemen array pecahDataTersisa = 1	pemesanan dan mengupdate data keberangkatan	keberangkatan dan menghapus data pemesanan	

6.1.3 Pengujian Unit Mencetak Laporan Keuangan

Berikut adalah algoritma dari operasi mencetak laporan.

Nama class : laporan

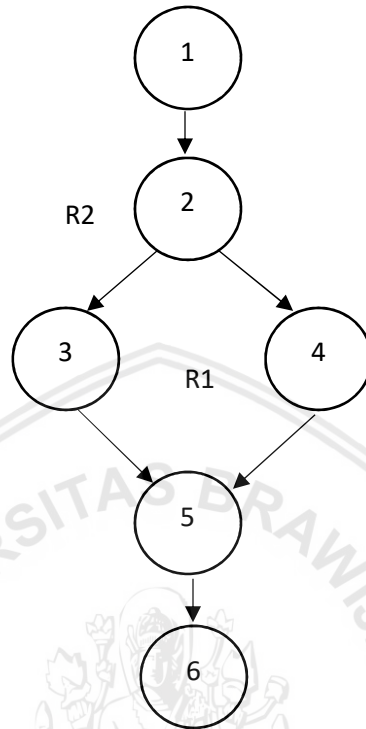
Nama method : printLaporan()

Tabel 6.5 Algoritme Mencetak Laporan Keuangan

No	Pseudocode
1.	<i>Array offset = 0</i>
2.	<i>Jika nilai parameter bulan = 0 maka lakukan, start if</i>
3.	<i>bulan = "" tahun = "" array view = hasil dari query pengambilan data laporan</i>
4.	<i>Else Array bulan = nilai bulan dengan format "F" menurut parameter bulan dan tahun Array tahun = nilai parameter tahun Array view = hasil dari query pengambilan data laporan berdasarkan bulan dan tahun dari parameter</i>
5.	<i>End if</i>
6.	<i>Load view tampilanPrint</i>

a. Flow graph

Berikut merupakan *flow graph* dari algoritma menghapus pemesanan travel



Gambar 6.3 Flow Graph Algoritma Mencetak Laporan

b. Cyclomatic Complexity

Berikut merupakan perhitungan *Cyclomatic Complexity* dari algoritma mencetak laporan:

1. $V(G) = \text{Total Ruang (R)}. \text{ Total R} = 2$
2. $V(G) = \text{Jumlah edge} - \text{Jumlah Node} + 2 = 6 - 6 + 2 = 2$
3. $V(G) = \text{Total Percabangan (P)} + 1 = 1 + 1 = 2$

c. Independent Path

Berikut merupakan jalur-jalur *independent* dari algoritma mencetak laporan :

1. Jalur 1 = 1 - 2 - 3 - 5 - 6
 2. Jalur 2 = 1 - 2 - 4 - 5 - 6
- d. Kasus Uji dan Hasil Pengujian*

Berikut merupakan kasus uji dan hasil uji dari algoritma mencetak laporan berdasarkan jalur *independent* yang dihasilkan :

Tabel 6.6 Test Case Mencetak Laporan

No.	No Jalur	Prosedur Uji	Expected Result	Result	Status
1.	1	Nilai parameter bulan = 0	Berhasil menampilkan view tampilanPrint dengan semua data laporan	Berhasil menampilkan view tampilanPrint dengan semua data laporan	Valid
2.	2	Nilai parameter bulan = 12	Berhasil menampilkan view tampilanPrint dengan data laporan pada bulan 12	Berhasil menampilkan view tampilanPrint dengan data laporan pada bulan 12	Valid

6.2 Pengujian Integrasi

Pengujian integrasi merupakan jenis pengujian yang berfokus pada hubungan dan interaksi antar kelas yang telah dibangun berdasarkan arsitektur kelas yang sebelumnya telah dirancang. Untuk melihat identifikasi dan perencanaan pengujian integrasi dapat dilihat pada tabel 6.7.

Tabel 6.7 Identifikasi dan Rencana Pengujian Integrasi

No	Nama Kelas	Nama <i>Method</i>	Tujuan
1	laporan	tambahLaporan()	Menambahkan data laporan ke dalam database
	laporanModel	tambahlaporan()	
2	laporan	viewBulanLaporan()	Menampilkan laporan berdasarkan bulan dan tahun yang dimasukkan
	laporanModel	displaylaporanbulan()	
3	beranda	index()	Menampilkan grafik rute yang paling sering dipesan
	laporanModel	rutetersering()	

Tabel 6.8 Hasil Pengujian Integrasi Nomor 1

Nomor Uji	1
Input pertama	<p>tglLaporan = "2019-01-15"</p> <p>kodeMobil = "DL-1/Daihatsu Luxio/N4446BM"</p> <p>asal = "Malang"</p> <p>tujuan = "Lamongan"</p> <p>driver = "9/Dodik Kusuma"</p> <p>pendapatan = 440000</p> <p>gajiDriver = 66000</p> <p>keuntungan = 374000</p> <p>noKursi = 1,2,4,6</p> <p>jumlahPemesanan = 4</p> <p>keterangan = -</p>
Method dari kelas laporan	tambahLaporan()
Output pertama / input kedua	<p>tglLaporan = "2019-01-15"</p> <p>kodeMobil = "DL-1/Daihatsu Luxio/N4446BM"</p> <p>asal = "Malang"</p> <p>tujuan = "Lamongan"</p> <p>driver = "9/Dodik Kusuma"</p> <p>pendapatan = 440000</p> <p>gajiDriver = 66000</p> <p>keuntungan = 374000</p> <p>noKursi = 1,2,4,6</p> <p>jumlahPemesanan = 4</p> <p>keterangan = -</p>
Method dari kelas Dblaporan	tambahlaporan()
<i>Expected result</i>	Data laporan tercatat ke dalam database sesuai dengan data yang telah dimasukkan.

Tabel 6.8 Hasil Pengujian Integrasi Nomor 1

<i>Result</i>	Data laporan tercatat ke dalam database sesuai dengan data yang telah dimasukkan.
Status	Valid

Tabel 6.9 Hasil Pengujian Integrasi Nomor 2

Nomor Uji	2
Input pertama	"tgl" = "Desember-2018"
Method dari kelas Core	viewBulanLaporan ()
Output pertama / input kedua	"bulan" = "tgl[1]" "tahun" = "tgl[0]"
Method dari kelas Dblaporan	displaylaporanbulan ()
<i>Expected result</i>	Berhasil menampilkan data laporan pemesanan pada bulan Desember 2018
<i>Result</i>	Berhasil menampilkan data laporan pemesanan pada bulan Desember 2018
Status	Valid

Tabel 6.10 Hasil Pengujian Integrasi Nomor 3

Nomor Uji	2
Input pertama	"tgl" = "Desember-2018"
Method dari kelas Core	index()
Output pertama / input kedua	"bulan" = "12" "tahun" = "2018"
Method dari kelas Dblaporan	rutetersering()
<i>Expected result</i>	Berhasil menampilkan grafik rute yang paling sering dipesan pada bulan Desember 2018
<i>Result</i>	Berhasil menampilkan grafik rute yang paling sering dipesan pada bulan Desember 2018

Tabel 6.10 Hasil Pengujian Integrasi Nomor 3

Status	Valid
--------	-------

6.3 Pengujian Validasi

Pengujian validasi bertujuan untuk memeriksa apakah kebutuhan yang telah di implementasi telah sesuai dengan yang telah dirancang sebelumnya didalam analisis kebutuhan. Pengujian validasi dilakukan untuk memastikan agar tidak ada fungsi yang menyimpang, serta mamastikan semua fungsi dapat berjalan dengan benar. Pengujian validasi menerapkan metode *blackbox testing* yaitu pengujian yang dilakukan tanpa perlu memperhatikan kode program, hanya menjalankan setiap fungsi dengan menyesuaikan terhadap kebutuhan yang telah didefinisikan sebelumnya.

6.3.1 Kebutuhan Fungsional

6.3.1.1 Pengujian Validasi Login

Pada tabel 6.11 menjelaskan pengujian validasi *login* mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.11 Pengujian Validasi *Login*

MTF_0100 : Login	
Kode Kebutuhan	MTF_0100
Nama Kasus Uji	<i>Login</i>
Prosedur	<ol style="list-style-type: none"> 1. Mengisi <i>username</i> berupa <i>admin</i>. 2. Mengisi <i>password</i> berupa <i>password</i>. 3. Menekan tombol <i>Login</i>.
<i>Expected Result</i>	Berhasil terautentikasi dan masuk kedalam halaman beranda.
<i>Result</i>	Berhasil terautentikasi dan masuk kedalam halaman beranda.
Status	Valid

6.3.1.2 Pengujian Validasi *Logout*

Pada tabel 6.12 menjelaskan pengujian validasi *logout* mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.12 Pengujian Validasi *Logout*

MTF_0200 : <i>Logout</i>	
Kode Kebutuhan	MTF_0200
Nama Kasus Uji	<i>Logout</i>
Prosedur	1. Menekan tombol <i>Logout</i> .
<i>Expected Result</i>	Berhasil keluar dari sistem
<i>Result</i>	Berhasil keluar dari sistem
Status	Valid

6.3.1.3 Pengujian Validasi Melihat Beranda

Pada tabel 6.13 menjelaskan pengujian validasi melihat beranda mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.13 Pengujian Validasi Melihat Beranda

MTF_0300 : Melihat Beranda	
Kode Kebutuhan	MTF_0300
Nama Kasus Uji	Melihat Beranda
Prosedur	1. Menekan tombol Beranda
<i>Expected Result</i>	Halaman beranda berhasil ditampilkan
<i>Result</i>	Halaman beranda berhasil ditampilkan
Status	Valid

6.3.1.4 Pengujian Validasi Mencari Grafik Pemesanan

Pada tabel 6.14 menjelaskan pengujian validasi mencari grafik pemesanan mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.14 Pengujian Validasi Mencari Grafik Pemesanan

MTF_0400 : Melihat Mencari Grafik Pemesanan	
Kode Kebutuhan	MTF_0400
Nama Kasus Uji	Mencari Grafik Pemesanan
Prosedur	1. Mengisi bulan dan tahun 2. Menekan tombol Beranda
<i>Expected Result</i>	Menampilkan grafik pemesanan berdasarkan bulan dan tahun yang diinputkan
<i>Result</i>	Menampilkan grafik pemesanan berdasarkan bulan dan tahun yang diinputkan
Status	Valid

6.3.1.5 Pengujian Validasi Melihat Grafik Pemesanan Terbanyak

Pada tabel 6.15 menjelaskan pengujian validasi melihat grafik pemesanan terbanyak mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.15 Pengujian Validasi Melihat Grafik Pemesanan Terbanyak

MTF_0500 : Melihat Grafik Pemesanan Terbanyak	
Kode Kebutuhan	MTF_0500
Nama Kasus Uji	Melihat Grafik Pemesanan Terbanyak
Prosedur	1. Menekan tombol Beranda
<i>Expected Result</i>	Menampilkan grafik pemesanan terbanyak
<i>Result</i>	Menampilkan grafik pemesanan terbanyak
Status	Valid

6.3.1.6 Pengujian Validasi Melihat Grafik Rute Pemesanan Travel

Pada tabel 6.16 menjelaskan pengujian validasi melihat grafik rute pemesanan travel mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.16 Pengujian Validasi Melihat Grafik Rute Pemesanan Travel

MTF_0600 : Melihat Grafik Rute Pemesanan Travel	
Kode Kebutuhan	MTF_0600
Nama Kasus Uji	Melihat Grafik Rute Pemesanan Travel
Prosedur	1. Menekan tombol Beranda
<i>Expected Result</i>	Menampilkan grafik rute pemesanan travel
<i>Result</i>	Menampilkan grafik rute pemesanan travel
Status	Valid

6.3.1.7 Pengujian Validasi Melihat Grafik Laporan Keuangan

Pada tabel 6.17 menjelaskan pengujian validasi melihat grafik laporan keuangan mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.17 Pengujian Validasi Melihat Grafik Laporan Keuangan

MTF_0700 : Melihat Grafik Laporan Keuangan	
Kode Kebutuhan	MTF_0700
Nama Kasus Uji	Mencari Grafik Laporan Keuangan
Prosedur	1. Menekan tombol Beranda
<i>Expected Result</i>	Menampilkan grafik laporan keuangan
<i>Result</i>	Menampilkan grafik laporan keuangan
Status	Valid

6.3.1.8 Pengujian Validasi Menambah Jadwal Travel Harian

Pada tabel 6.18 menjelaskan pengujian validasi menambah jadwal travel harian mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.18 Pengujian Validasi Menambah Jadwal Travel Harian

MTF_0800 : Menambah Jadwal Travel Harian	
Kode Kebutuhan	MTF_0800
Nama Kasus Uji	Menambah Jadwal Travel Harian

Tabel 6.18 Pengujian Validasi Menambah Jadwal Travel Harian

MTF_0800 : Menambah Jadwal Travel Harian	
Prosedur	<ol style="list-style-type: none"> 1. Memilih hari keberangkatan 2. Menekan tombol Pilih 3. Mengisi field pada form seperti asal, tujuan, mobil, jam keberangkatan dan harga 4. Menekan tombol Simpan
<i>Expected Result</i>	Jadwal travel harian berhasil ditambahkan ke dalam database
<i>Result</i>	Jadwal travel harian berhasil ditambahkan ke dalam database
Status	Valid

6.3.1.9 Pengujian Validasi Melihat Jadwal Travel Harian

Pada tabel 6.19 menjelaskan pengujian validasi melihat jadwal travel harian mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.19 Pengujian Validasi Melihat Jadwal Travel Harian

MTF_0900 : Melihat Jadwal Travel Harian	
Kode Kebutuhan	MTF_0900
Nama Kasus Uji	Melihat Jadwal Travel Harian
Prosedur	<ol style="list-style-type: none"> 1. Menekan tombol Jadwal Travel Harian
<i>Expected Result</i>	Jadwal travel harian berhasil ditampilkan
<i>Result</i>	Jadwal travel harian berhasil ditampilkan
Status	Valid

6.3.1.10 Pengujian Validasi Mengubah Jadwal Travel Harian

Pada tabel 6.20 menjelaskan pengujian validasi mengubah jadwal travel harian mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.20 Pengujian Validasi Mengubah Jadwal Travel Harian

MTF_1000 : Mengubah Jadwal Travel Harian	
Kode Kebutuhan	MTF_1000
Nama Kasus Uji	Mengubah Jadwal Travel Harian
Prosedur	1. Mengubah data jadwal yang diperlukan seperti asal, tujuan, mobil, jam dan harga 2. Menekan tombol Simpan
<i>Expected Result</i>	Jadwal travel harian berhasil diubah dari database
<i>Result</i>	Jadwal travel harian berhasil diubah dari database
Status	Valid

6.3.1.11 Pengujian Validasi Menghapus Jadwal Travel Harian

Pada tabel 6.21 menjelaskan pengujian validasi menghapus jadwal travel harian mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.21 Pengujian Validasi Menghapus Jadwal Travel Harian

MTF_1100 : Menghapus Jadwal Travel Harian	
Kode Kebutuhan	MTF_1100
Nama Kasus Uji	Menghapus Jadwal Travel Harian
Prosedur	1. Menekan tombol Delete
<i>Expected Result</i>	Jadwal travel harian berhasil dihapus dari database
<i>Result</i>	Jadwal travel harian berhasil dihapus dari database
Status	Valid

6.3.1.12 Pengujian Validasi Menambah Jadwal Travel Tambahan

Pada tabel 6.22 menjelaskan pengujian validasi menambah jadwal travel tambahan mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.22 Pengujian Validasi Menambah Jadwal Travel Tambahan

MTF_1200 : Menambah Jadwal Travel Tambahan	
Kode Kebutuhan	MTF_1200
Nama Kasus Uji	Menambah Jadwal Travel Tambahan
Prosedur	<ol style="list-style-type: none"> 1. Memilih tanggal keberangkatan 2. Menekan tombol Pilih 3. Mengisi field yang dibutuhkan seperti asal, tujuan, mobil, jam keberangkatan dan harga 4. Menekan tombol Simpan
<i>Expected Result</i>	Jadwal travel tambahan berhasil ditambahkan ke database
<i>Result</i>	Jadwal travel tambahan berhasil ditambahkan ke database
Status	Valid

6.3.1.13 Pengujian Validasi Melihat Jadwal Travel Tambahan

Pada tabel 6.23 menjelaskan pengujian validasi melihat jadwal travel tambahan mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.23 Pengujian Validasi Melihat Jadwal Travel Tambahan

MTF_1300 : Melihat Jadwal Travel Tambahan	
Kode Kebutuhan	MTF_1300
Nama Kasus Uji	Melihat Jadwal Travel Tambahan
Prosedur	<ol style="list-style-type: none"> 1. Menekan tombol Jadwal Travel Tambahan
<i>Expected Result</i>	Jadwal travel tambahan berhasil ditampilkan
<i>Result</i>	Jadwal travel tambahan berhasil ditampilkan
Status	Valid

6.3.1.14 Pengujian Validasi Mengubah Jadwal Travel Tambahan

Pada tabel 6.24 menjelaskan pengujian validasi mengubah jadwal travel tambahan mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.24 Pengujian Validasi Mengubah Jadwal Travel Tambahan

MTF_1400 : Mengubah Jadwal Travel Tambahan	
Kode Kebutuhan	MTF_1400
Nama Kasus Uji	Mengubah Jadwal Travel Tambahan
Prosedur	1. Mengubah data jadwal yang diperlukan seperti asal, tujuan, mobil, jam dan harga 2. Menekan tombol Simpan
<i>Expected Result</i>	Jadwal travel tambahan berhasil diubah dari database
<i>Result</i>	Jadwal travel tambahan berhasil diubah dari database
Status	Valid

6.3.1.15 Pengujian Validasi Menghapus Jadwal Travel Tambahan

Pada tabel 6.25 menjelaskan pengujian validasi menghapus jadwal travel tambahan mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.25 Pengujian Validasi Menghapus Jadwal Travel Tambahan

MTF_1500 : Menghapus Jadwal Travel Tambahan	
Kode Kebutuhan	MTF_1500
Nama Kasus Uji	Menghapus Jadwal Travel Tambahan
Prosedur	1. Menekan tombol Delete
<i>Expected Result</i>	Jadwal travel tambahan berhasil dihapus dari database
<i>Result</i>	Jadwal travel tambahan berhasil dihapus dari database
Status	Valid

6.3.1.16 Pengujian Validasi Menambah Pemesanan Travel

Pada tabel 6.26 menjelaskan pengujian validasi menambah pemesanan travel mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.26 Pengujian Validasi Menambah Pemesanan Travel

MTF_1600 : Menambah Pemesanan Travel	
Kode Kebutuhan	MTF_1600
Nama Kasus Uji	Menambah Pemesanan Travel
Prosedur	<ol style="list-style-type: none"> 1. Memilih kota asal, kota tujuan dan tanggal keberangkatan 2. Menekan tombol Cari Jadwal. 3. Menekan tombol Pesan pada jadwal yang diinginkan. 4. Mengisi data nama lengkap, no telpon, alamat asal, alamat tujuan dan memilih seat yang diinginkan. 5. Menekan tombol Pesan
<i>Expected Result</i>	Data pemesanan travel berhasil ditambahkan ke database
<i>Result</i>	Data pemesanan travel berhasil ditambahkan ke database
Status	Valid

6.3.1.17 Pengujian Validasi Mencari Jadwal Tersedia

Pada tabel 6.27 menjelaskan pengujian validasi mencari jadwal tersedia mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.27 Pengujian Validasi Mencari Jadwal Tersedia

MTF_1700 : Mencari Jadwal Tersedia	
Kode Kebutuhan	MTF_1700
Nama Kasus Uji	Mencari Jadwal Tersedia
Prosedur	<ol style="list-style-type: none"> 1. Memilih kota asal, kota tujuan dan tanggal keberangkatan 2. Menekan tombol Cari Jadwal.
<i>Expected Result</i>	Jadwal travel yang masih tersedia berhasil ditampilkan
<i>Result</i>	Jadwal travel yang masih tersedia berhasil ditampilkan
Status	Valid

6.3.1.18 Pengujian Validasi Melihat Pemesanan Travel

Pada tabel 6.28 menjelaskan pengujian validasi melihat pemesanan travel mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.28 Pengujian Validasi Melihat Pemesanan Travel

MTF_1800 : Melihat Pemesanan Travel	
Kode Kebutuhan	MTF_1800
Nama Kasus Uji	Melihat Pemesanan Travel
Prosedur	1. Menekan tombol Pemesanan Travel
<i>Expected Result</i>	Pemesanan travel berhasil ditampilkan
<i>Result</i>	Pemesanan travel berhasil ditampilkan
Status	Valid

6.3.1.19 Pengujian Validasi Mengubah Pemesanan Travel

Pada tabel 6.29 menjelaskan pengujian validasi mengubah pemesanan travel mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.29 Pengujian Validasi Mengubah Pemesanan Travel

MTF_1900 : Mengubah Pemesanan Travel	
Kode Kebutuhan	MTF_1900
Nama Kasus Uji	Mengubah Pemesanan Travel
Prosedur	1. Mengubah data yang perlu diubah seperti nama lengkap, telepon, alamat asal, alamat tujuan dan seat yang diinginkan. 2. Menekan tombol Simpan.
<i>Expected Result</i>	Pemesanan travel berhasil diubah dari database
<i>Result</i>	Pemesanan travel berhasil diubah dari database
Status	Valid

6.3.1.20 Pengujian Validasi Menghapus Pemesanan Travel

Pada tabel 6.30 menjelaskan pengujian validasi menghapus pemesanan travel mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.30 Pengujian Validasi Menghapus Pemesanan Travel

MTF_2000 : Menghapus Pemesanan Travel	
Kode Kebutuhan	MTF_2000
Nama Kasus Uji	Menghapus Pemesanan Travel Tambahan
Prosedur	1. Menekan tombol Delete
<i>Expected Result</i>	Pemesanan travel berhasil dihapus dari database
<i>Result</i>	Pemesanan travel berhasil dihapus dari database
Status	Valid

6.3.1.21 Pengujian Validasi Melihat Manajemen Pemesanan

Pada tabel 6.31 menjelaskan pengujian validasi melihat manajemen pemesanan mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.31 Pengujian Validasi Melihat Manajemen Pemesanan

MTF_2100 : Melihat Manajemen Pemesanan	
Kode Kebutuhan	MTF_2100
Nama Kasus Uji	Melihat Manajemen Pemesanan
Prosedur	1. Menekan tombol Manajemen Pemesanan
<i>Expected Result</i>	Data manajemen pemesanan berhasil ditampilkan
<i>Result</i>	Data manajemen pemesanan berhasil ditampilkan
Status	Valid

6.3.1.22 Pengujian Validasi Mengubah Driver

Pada tabel 6.32 menjelaskan pengujian validasi mengubah driver mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.32 Pengujian Validasi Mengubah Driver

MTF_2200 : Mengubah Driver	
Kode Kebutuhan	MTF_2200
Nama Kasus Uji	Mengubah Driver
Prosedur	1. Memilih driver 2. Menekan tombol Simpan.
<i>Expected Result</i>	Driver yang dipilih ditempatkan ke dalam data keberangkatan
<i>Result</i>	Driver yang dipilih ditempatkan ke dalam data keberangkatan
Status	Valid

6.3.1.23 Pengujian Validasi Mencetak Surat Jalan

Pada tabel 6.33 menjelaskan pengujian validasi mencetak surat jalan mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.33 Pengujian Validasi Mencetak Surat Jalan

MTF_2300 : Mencetak Surat Jalan	
Kode Kebutuhan	MTF_2300
Nama Kasus Uji	Mencetak Surat Jalan
Prosedur	1. Menekan tombol Print Surat Jalan 2. Menekan tombol dengan ikon <i>print</i> 3. Menekan tombol Cetak
<i>Expected Result</i>	Surat jalan berhasil dicetak
<i>Result</i>	Surat jalan berhasil dicetak
Status	Valid

6.3.1.24 Pengujian Validasi Melihat Daftar Penumpang

Pada tabel 6.34 menjelaskan pengujian validasi melihat daftar penumpang mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.34 Pengujian Validasi Melihat Daftar Penumpang

MTF_2400 : Melihat Daftar Penumpang	
Kode Kebutuhan	MTF_2400
Nama Kasus Uji	Melihat Daftar Penumpang
Prosedur	1. Menekan tombol Daftar Penumpang
<i>Expected Result</i>	Daftar penumpang berhasil ditampilkan
<i>Result</i>	Daftar penumpang berhasil ditampilkan
Status	Valid

6.3.1.25 Pengujian Validasi Mencetak Tiket

Pada tabel 6.35 menjelaskan pengujian validasi mencetak tiket mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.35 Pengujian Validasi Mencetak Tiket

MTF_2500 : Mencetak Tiket	
Kode Kebutuhan	MTF_2500
Nama Kasus Uji	Mencetak Tiket
Prosedur	1. Menekan tombol Print Tiket 2. Menekan tombol dengan ikon <i>print</i> 3. Menekan tombol Cetak
<i>Expected Result</i>	Surat jalan berhasil dicetak
<i>Result</i>	Surat jalan berhasil dicetak
Status	Valid

6.3.1.26 Pengujian Validasi Menandai Transaksi Selesai

Pada tabel 6.36 menjelaskan pengujian validasi menandai transaksi selesai dimulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.36 Pengujian Validasi Menandai Transaksi Selesai

MTF_2600 : Menandai Transaksi Selesai	
Kode Kebutuhan	MTF_2600
Nama Kasus Uji	Menandai Transaksi Selesai
Prosedur	<ol style="list-style-type: none"> 1. Menekan tombol Tandai Selesai 2. Mengisi data yang diperlukan 3. Menekan Simpan
<i>Expected Result</i>	Status pada manajemen pemesanan menjadi selesai
<i>Result</i>	Status pada manajemen pemesanan menjadi selesai
Status	Valid

6.3.1.27 Pengujian Validasi Menambah Data Mobil

Pada tabel 6.37 menjelaskan pengujian validasi menambah data mobil mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.37 Pengujian Validasi Menambah Data Mobil

MTF_2700 : Menambah Data Mobil	
Kode Kebutuhan	MTF_2700
Nama Kasus Uji	Menambah Data Mobil
Prosedur	<ol style="list-style-type: none"> 1. Mengisi field nama mobil, jumlah seat, warna dan nomor kendaraan 2. Menekan tombol Simpan
<i>Expected Result</i>	Data mobil berhasil ditambahkan ke dalam database
<i>Result</i>	Data mobil berhasil ditambahkan ke dalam database
Status	Valid

6.3.1.28 Pengujian Validasi Melihat Data Mobil

Pada tabel 6.38 menjelaskan pengujian validasi melihat data mobil mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.38 Pengujian Validasi Melihat Data Mobil

MTF_2800 : Melihat Data Mobil	
Kode Kebutuhan	MTF_2800
Nama Kasus Uji	Melihat Data Mobil
Prosedur	1. Menekan tombol Daftar Mobil
<i>Expected Result</i>	Data mobil berhasil ditampilkan
<i>Result</i>	Data mobil berhasil ditampilkan
Status	Valid

6.3.1.29 Pengujian Validasi Mengubah Data Mobil

Pada tabel 6.39 menjelaskan pengujian validasi mengubah data mobil mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.39 Pengujian Validasi Mengubah Data Mobil

MTF_2900 : Mengubah Data Mobil	
Kode Kebutuhan	MTF_2900
Nama Kasus Uji	Mengubah Jadwal Travel Harian
Prosedur	1. Mengubah data yang diperlukan seperti nama mobil, jumlah seat, warna dan nomor kendaraan 2. Menekan tombol Simpan
<i>Expected Result</i>	Data mobil berhasil diubah dari database
<i>Result</i>	Data mobil berhasil diubah dari database
Status	Valid

6.3.1.30 Pengujian Validasi Menghapus Data Mobil

Pada tabel 6.40 menjelaskan pengujian validasi menghapus data mobil mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.40 Pengujian Validasi Menghapus Data Mobil

MTF_3000 : Menghapus Data Mobil	
Kode Kebutuhan	MTF_3000
Nama Kasus Uji	Menghapus Data Mobil
Prosedur	1. Menekan tombol Delete
<i>Expected Result</i>	Data mobil berhasil dihapus dari database
<i>Result</i>	Data mobil berhasil dihapus dari database
Status	Valid

6.3.1.31 Pengujian Validasi Menambah Data Driver

Pada tabel 6.41 menjelaskan pengujian validasi menambah data driver mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.41 Pengujian Validasi Menambah Data Driver

MTF_3100 : Menambah Data Driver	
Kode Kebutuhan	MTF_3100
Nama Kasus Uji	Menambah Data Driver
Prosedur	1. Mengisi field nama driver, no telepon, alamat 2. Menekan tombol Simpan
<i>Expected Result</i>	Data driver berhasil ditambahkan ke dalam database
<i>Result</i>	Data driver berhasil ditambahkan ke dalam database
Status	Valid

6.3.1.32 Pengujian Validasi Melihat Data Driver

Pada tabel 6.42 menjelaskan pengujian validasi melihat data driver mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.42 Pengujian Validasi Melihat Data Driver

MTF_3200 : Melihat Data Driver	
Kode Kebutuhan	MTF_3200
Nama Kasus Uji	Melihat Data Driver
Prosedur	1. Menekan tombol Daftar Driver
<i>Expected Result</i>	Data driver berhasil ditampilkan
<i>Result</i>	Data driver berhasil ditampilkan
Status	Valid

6.3.1.33 Pengujian Validasi Mengubah Data Driver

Pada tabel 6.43 menjelaskan pengujian validasi mengubah data driver mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.43 Pengujian Validasi Mengubah Data Driver

MTF_3300 : Mengubah Data Driver	
Kode Kebutuhan	MTF_3300
Nama Kasus Uji	Mengubah Data Driver
Prosedur	1. Mengubah data yang diperlukan seperti nama driver, no telepon, alamat 2. Menekan tombol Simpan
<i>Expected Result</i>	Data driver berhasil diubah dari database
<i>Result</i>	Data driver berhasil diubah dari database
Status	Valid

6.3.1.34 Pengujian Validasi Menghapus Data Driver

Pada tabel 6.44 menjelaskan pengujian validasi menghapus data driver mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.44 Pengujian Validasi Menghapus Data Driver

MTF_3400 : Menghapus Data Driver	
Kode Kebutuhan	MTF_3400
Nama Kasus Uji	Menghapus Data Driver
Prosedur	1. Menekan tombol Delete
<i>Expected Result</i>	Data driver berhasil dihapus dari database
<i>Result</i>	Data driver berhasil dihapus dari database
Status	Valid

6.3.1.35 Pengujian Validasi Menambah Data Kota

Pada tabel 6.45 menjelaskan pengujian validasi menambah data kota mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.45 Pengujian Validasi Menambah Data Kota

MTF_3500 : Menambah Data Kota	
Kode Kebutuhan	MTF_3500
Nama Kasus Uji	Menambah Data Kota
Prosedur	1. Mengisi field nama kota 2. Menekan tombol Simpan
<i>Expected Result</i>	Data kota berhasil ditambahkan ke dalam database
<i>Result</i>	Data kota berhasil ditambahkan ke dalam database
Status	Valid

6.3.1.36 Pengujian Validasi Melihat Data Kota

Pada tabel 6.46 menjelaskan pengujian validasi melihat data kota mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.46 Pengujian Validasi Melihat Data Kota

MTF_3600 : Melihat Data Driver	
Kode Kebutuhan	MTF_3600
Nama Kasus Uji	Melihat Data Kota
Prosedur	1. Menekan tombol Daftar Kota
<i>Expected Result</i>	Data kota berhasil ditampilkan
<i>Result</i>	Data kota berhasil ditampilkan
Status	Valid

6.3.1.37 Pengujian Validasi Mengubah Data Kota

Pada tabel 6.47 menjelaskan pengujian validasi mengubah data kota mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.47 Pengujian Validasi Mengubah Data Kota

MTF_3700 : Mengubah Data Kota	
Kode Kebutuhan	MTF_3700
Nama Kasus Uji	Mengubah Data Kota
Prosedur	1. Mengubah data yang diperlukan seperti nama kota 2. Menekan tombol Simpan
<i>Expected Result</i>	Data kota berhasil diubah dari database
<i>Result</i>	Data kota berhasil diubah dari database
Status	Valid

6.3.1.38 Pengujian Validasi Menghapus Data Kota

Pada tabel 6.48 menjelaskan pengujian validasi menghapus data kota mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.48 Pengujian Validasi Menghapus Data Kota

MTF_3800 : Menghapus Data Kota	
Kode Kebutuhan	MTF_3800
Nama Kasus Uji	Menghapus Data Kota
Prosedur	1. Menekan tombol Delete
<i>Expected Result</i>	Data kota berhasil dihapus dari database
<i>Result</i>	Data kota berhasil dihapus dari database
Status	Valid

6.3.1.39 Pengujian Validasi Melihat Gaji Driver

Pada tabel 6.49 menjelaskan pengujian validasi melihat gaji driver mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.49 Pengujian Validasi Melihat Gaji Driver

MTF_3900 : Melihat Gaji Driver	
Kode Kebutuhan	MTF_3900
Nama Kasus Uji	Melihat Gaji Driver
Prosedur	1. Menekan tombol Gaji Driver
<i>Expected Result</i>	Gaji Driver berhasil ditampilkan
<i>Result</i>	Gaji Driver berhasil ditampilkan
Status	Valid

6.3.1.40 Pengujian Validasi Mencari Gaji Driver

Pada tabel 6.50 menjelaskan pengujian validasi mencari gaji driver mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.50 Pengujian Validasi Mencari Gaji Driver

MTF_4000 : Mencari Gaji Driver	
Kode Kebutuhan	MTF_4000

Tabel 6.50 Pengujian Validasi Mencari Gaji Driver

MTF_4000 : Mencari Gaji Driver	
Nama Kasus Uji	Mencari Gaji Driver
Prosedur	1. Mengisi bulan dan tahun 2. Menekan tombol dengan ikon pencarian
<i>Expected Result</i>	Menampilkan gaji driver sesuai bulan dan tahun yang diinputkan
<i>Result</i>	Menampilkan gaji driver sesuai bulan dan tahun yang diinputkan
Status	Valid

6.3.1.41 Pengujian Validasi Mencetak Gaji Driver

Pada tabel 6.51 menjelaskan pengujian validasi mencetak gaji driver mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.51 Pengujian Validasi Mencetak Gaji Driver

MTF_4100 : Mencetak Gaji Driver	
Kode Kebutuhan	MTF_4100
Nama Kasus Uji	Mencetak Gaji Driver
Prosedur	1. Menekan tombol Cetak Gaji Driver 2. Menekan tombol dengan ikon <i>print</i> 3. Menekan tombol Cetak
<i>Expected Result</i>	Gaji driver berhasil dicetak
<i>Result</i>	Gaji driver berhasil dicetak
Status	Valid

6.3.1.42 Pengujian Validasi Menambah Laporan Keuangan

Pada tabel 6.52 menjelaskan pengujian validasi menambah laporan keuangan mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.52 Pengujian Validasi Menambah Laporan Keuangan

MTF_4200 : Menambah Laporan Keuangan	
Kode Kebutuhan	MTF_4200
Nama Kasus Uji	Menambah Laporan Keuangan
Prosedur	1. Mengisi field yang diperlukan 2. Menekan tombol Simpan
<i>Expected Result</i>	Data laporan keuangan berhasil ditambahkan ke database
<i>Result</i>	Data laporan keuangan berhasil ditambahkan ke database
Status	Valid

6.3.1.43 Pengujian Validasi Melihat Laporan Keuangan

Pada tabel 6.53 menjelaskan pengujian validasi melihat laporan keuangan mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.53 Pengujian Validasi Melihat Laporan Keuangan

MTF_4300 : Melihat Laporan Keuangan	
Kode Kebutuhan	MTF_4300
Nama Kasus Uji	Melihat Laporan Keuangan
Prosedur	1. Menekan tombol Laporan
<i>Expected Result</i>	Laporan Keuangan berhasil ditampilkan
<i>Result</i>	Laporan keuangan berhasil ditampilkan
Status	Valid

6.3.1.44 Pengujian Validasi Mengubah Laporan Keuangan

Pada tabel 6.54 menjelaskan pengujian validasi mengubah laporan keuangan mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.54 Pengujian Validasi Mengubah Laporan Keuangan

MTF_4400 : Mengubah Laporan Keuangan	
Kode Kebutuhan	MTF_4400
Nama Kasus Uji	Mengubah Laporan Keuangan
Prosedur	1. Mengubah data yang perlu diubah 2. Menekan tombol Simpan.
<i>Expected Result</i>	Laporan keuangan berhasil diubah dari database
<i>Result</i>	Laporan keuangan berhasil diubah dari database
Status	Valid

6.3.1.45 Pengujian Validasi Menghapus Laporan Keuangan

Pada tabel 6.55 menjelaskan pengujian validasi menghapus laporan keuangan mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.55 Pengujian Validasi Menghapus Laporan Keuangan

MTF_4500 : Menghapus Laporan Keuangan	
Kode Kebutuhan	MTF_4500
Nama Kasus Uji	Menghapus Laporan Keuangan
Prosedur	1. Menekan tombol Delete
<i>Expected Result</i>	Laporan keuangan berhasil dihapus dari database
<i>Result</i>	Laporan keuangan berhasil dihapus dari database
Status	Valid

6.3.1.46 Pengujian Validasi Mencetak Laporan Keuangan

Pada tabel 6.56 menjelaskan pengujian validasi mencetak laporan keuangan mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.56 Pengujian Validasi Mencetak Laporan Keuangan

MTF_4600 : Mencetak Laporan Keuangan	
Kode Kebutuhan	MTF_4600
Nama Kasus Uji	Mencetak Laporan Keuangan
Prosedur	<ol style="list-style-type: none"> 1. Menekan tombol Cetak Laporan Keuangan 2. Menekan tombol dengan ikon <i>print</i> 3. Menekan tombol Cetak
<i>Expected Result</i>	Laporan Keuangan berhasil dicetak
<i>Result</i>	Laporan Keuangan berhasil dicetak
Status	Valid

6.3.1.47 Pengujian Validasi Mencari Laporan Keuangan

Pada tabel 6.57 menjelaskan pengujian validasi mencari laporan keuangan mulai dari nama kasus uji, prosedur, hasil yang diharapkan, hasil yang tercapai, serta status apakah valid atau tidak.

Tabel 6.57 Pengujian Validasi Mencari Laporan keuangan

MTF_4700 : Mencari Laporan Keuangan	
Kode Kebutuhan	MTF_4700
Nama Kasus Uji	Mencari Laporan Keuangan
Prosedur	<ol style="list-style-type: none"> 1. Mengisi bulan dan tahun 2. Menekan tombol dengan ikon pencarian
<i>Expected Result</i>	Menampilkan laporan keuangan sesuai bulan dan tahun yang diinputkan
<i>Result</i>	Menampilkan laporan keuangan sesuai bulan dan tahun yang diinputkan
Status	Valid

6.3.2 Kebutuhan Non Fungsional

6.3.2.1 Pengujian Validasi *Usability*

Pengujian usability diperlukan untuk menguji antarmuka perangkat lunak agar dapat digunakan dengan mudah dan memiliki nilai kebergunaan yang tinggi bagi

pengguna akhir perangkat lunak dalam penelitian ini yaitu operator pada perusahaan *travel* Lia Jaya Sentosa.

Untuk pengujian *usability*, terdapat dua metode yang dapat digunakan yaitu *heuristic evaluation* (HE) dan *system usability scale* (SUS). Kedua metode ini memiliki fokus yang sama-sama menilai interaksi antarmuka perangkat lunak, namun memiliki perbedaan pada orang yang mengujinya. HE merupakan pengujian antarmuka yang dilakukan oleh ahli (Kurniawan, Rokhmawati, & Rachmadi, 2018), sedangkan SUS merupakan pengujian antarmuka untuk mengukur aspek-aspek *usability* menurut penilaian subyektif pengguna (Aprilia, Santoso, & Ferdiana, 2015). Penggunaan SUS lebih menekankan perspektif pengguna akhir sehingga hasil evaluasi akan lebih sesuai dengan keadaan nyata yang terjadi di dalam perusahaan. Maka dari itu, penggunaan SUS lebih cocok digunakan dalam pengujian *usability* sistem informasi manajemen Travel Lia Jaya Sentosa.

1. Responden

Terdapat 5 responden yang akan membantu dalam pengisian kuesioner SUS. Data dari responden tersebut dapat dilihat pada tabel 6.58.

Tabel 6.58 Data Responden

Responden	Nama	Umur	Pekerjaan
Resp 1	Ana Sukmawati	25 tahun	Operator
Resp 2	Indah Indrianti	26 tahun	Operator
Resp 3	Kistiyah	27 tahun	Akuntan
Resp 4	Irawan	35 tahun	Pimpinan
Resp 5	Herman	25 tahun	Operator

2. Task

Terdapat 10 task yang harus dikerjakan oleh responden untuk menilai dan mengisi kuesioner SUS. Dapat dilihat pada Tabel 6.59.

Tabel 6.59 Task

No	Task
1	Login
2	Menambah Jadwal Travel Harian
3	Menambah Jadwal Travel Tambahan

4	Menambah Pemesanan Travel
5	Menambah Laporan Keuangan
6	Print Surat Jalan
7	Print Tiket
8	Print Laporan Keuangan
9	Mencari Laporan Keuangan
10	Mencari Gaji Driver

3. Hasil Pengisian Kuesioner

Berikut merupakan hasil pengisian kuesioner SUS dapat dilihat pada tabel 6.60.

Tabel 6.60 Hasil Pengisian Kuesioner

Responden	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
Responden 1	5	1	4	2	4	1	5	1	5	2
Responden 2	4	2	4	2	4	2	5	1	4	2
Responden 3	3	1	4	1	4	1	4	1	4	2
Responden 4	4	2	4	2	4	2	3	1	3	1
Responden 5	4	1	5	1	4	1	3	1	5	2

4. Skor Akhir SUS

Berikut merupakan nilai skor akhir SUS setelah dihitung berdasarkan rumus, dapat dilihat pada tabel 6.61.

Tabel 6.61 Pengolahan Skor Akhir SUS

Responden	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Jumlah	Skor Akhir
Resp 1	4	4	3	3	3	4	4	4	4	3	36	90
Resp 2	3	3	3	3	3	3	4	4	3	3	32	80
Resp 3	2	4	3	4	3	4	3	4	3	3	33	82,5
Resp 4	3	4	3	3	3	3	2	4	2	4	31	77,5
Resp 5	3	4	4	4	3	4	2	4	4	3	35	87,5
Rata-rata											33,4	83,5

Skor akhir SUS akan berada pada kisaran 0-100. Berdasarkan skor akhir SUS tersebut, dapat diketahui seberapa tinggi tingkat *usability* dan akseptabilitas

(*acceptable*) desain sistem aplikasi yang sedang dikembangkan. Penilaiannya berdasarkan tiga kategori *acceptability ranges*, *grade scale* dan *adjective ratings*. Pada tabel diatas, didapatkan rata-rata skor SUS sebanyak 83.5, dari skor tersebut dapat disimpulkan bahwa sistem informasi manajemen travel yang telah dibangun berada pada kategori *acceptable* menurut skala *acceptability ranges*, bernilai B pada skala *grade scale* dan bernilai *excellent* pada skala *adjective ratings* yang berarti sistem ini dapat diterima dengan baik dan memudahkan pekerjaan yang ada di Lia Jaya Sentosa.



BAB 7 PENUTUP

Pada bab ini, dilakukan penarikan kesimpulan mulai dari tahap analisis kebutuhan hingga tahap pengujian sistem. Selain penarikan kesimpulan, penjabaran saran yang juga akan diberikan. Penjabaran saran dilakukan untuk mengetahui kekurangan yang ada di dalam penelitian ini dan kemudian diharapkan dapat disempurnakan di dalam penelitian dan pengembangan yang lebih lanjut.

7.1 Kesimpulan

Berdasarkan hasil penelitian yang telah peroleh sesuai dengan metodologi yang telah dideskripsikan, dapat ditarik kesimpulan sebagai berikut:

1. Untuk proses analisis kebutuhan sistem informasi manajemen Travel Lia Jaya Sentosa, langkah-langkah yang dilakukan adalah mendeskripsikan gambaran umum sistem yang dibangun dan kemudian didapatkan 1 orang aktor dan 47 kebutuhan fungsional serta 1 kebutuhan non fungsional. Kemudian untuk setiap kebutuhan fungsional yang telah ditemukan, akan dimodelkan ke dalam *use case diagram* dan *use case scenario*.
2. Proses perancangan sistem informasi manajemen Travel Lia Jaya Sentosa dilakukan berdasarkan analisis kebutuhan yang telah didefinisikan sebelumnya. Pada perancangan dilakukan dengan menggunakan *sequence diagram* untuk menggambarkan interaksi antar komponen sistem serta antara aktor dan sistem. Sequence diagram yang dihasilkan berjumlah sama dengan kebutuhan yang didefinisikan. Pemodelan atau perancangan pada pengembangan sistem ini juga menggunakan *class diagram*, perancangan komponen, perancangan basis data serta perancangann antarmuka. Class diagram digunakan untuk menggambarkan kelas-kelas yang ada di dalam sistem beserta hubungannya. Pada perancangan komponen dihasilkan 3 sampel algoritma. Pada perancangan basis data menghasilkan 9 tabel yang diperoleh pada tahap *Concept Data Model*. Pada perancangan antarmuka dihasilkan gambaran umum halaman-halaman yang akan diimplementasikan ke dalam sistem yang sesungguhnya.
3. Pembangunan atau implementasi di dalam sistem informasi manajemen Travel Lia Jaya Sentosa dilakukan dengan menggunakan bahasa pemrograman *php*, *html*, *javascript* sesuai dengan bahasa pemrograman yang mendukung pengembangan sistem berbasis *web*. Pada proses implementasi juga menghasilkan implementasi basis data yang ditunjukkan dalam bentuk *physical data model* serta implementasi antarmuka berdasarkan hasil perancangan antarmuka yang digambarkan sebelumnya.

4. Proses pengujian dilakukan dengan 3 pengujian, yaitu pengujian *unit*, pengujian integrasi dan pengujian validasi. Pengujian *unit* dalam penelitian ini dilakukan terhadap 3 fungsi di dalam sistem dan memberikan status valid pada setiap kasus ujinya. Pengujian integrasi dilakukan pada 2 fungsi dengan jumlah 3 kasus uji yang memiliki keterkaitan pada kelas yang berbeda dan memberikan status *valid* pada setiap kasus ujinya. Pengujian validasi dilakukan terhadap seluruh kebutuhan fungsional, hasil status yang diperoleh dari pengujian validasi fungsional adalah valid pada setiap fungsional yang diuji, kemudian pada pengujian validasi non fungsional *usability*, hasil yang diperoleh adalah sistem yang dibangun dapat memudahkan pekerjaan yang ada di perusahaan dengan berdasarkan skor SUS yang didapatkan yaitu 83,5 yang berarti jika dikonversi menurut rating dan skala konversi SUS berada pada kategori *acceptable* menurut skala *acceptability ranges*, bernilai B pada skala *grade scale* dan bernilai *excellent* pada skala *adjective ratings*.

7.2 Saran

Dalam pembangunan sistem informasi manajemen Travel Lia Jaya Sentosa yang telah dilakukan dalam penelitian ini, tentunya masih terdapat beberapa kekurangan yang perlu diperbaiki, sehingga saran untuk pengembangan sistem yang lebih lanjut yaitu:

1. Adanya penyempurnaan dan penambahan diagram atau chart yang dapat memberikan lebih informatif dan lebih menarik, sehingga nantinya akan memberikan perusahaan pilihan strategi lebih banyak dalam pengalokasian mobil dan penentuan jadwal keberangkatan travel.
2. Adanya sistem pemesanan tiket secara online, agar memudahkan calon penumpang untuk pemesanan tiket travel.
3. Adanya fitur khusus bagian keuangan perusahaan travel.
4. Adanya fitur GPS untuk melacak lokasi mobil.

DAFTAR PUSTAKA

- Aprilia, I., Santoso, P. I., & Ferdiana, R. (2015). Website Usability Testing using System Usability Scale. *IPTEK-KOM, Vol.17 No.1*, 31-38.
- Ardiansyah, & Ghazali, M. I. (2016). Pengujian Usability User Interface dan User Experience Aplikasi E-Reader Skripsi Berbasis Hypertext. *Jurnal Ilmiah Teknologi Informasi Terapan Volume II, No.3*, 213-220.
- Bangor, A., Kortum, P., & Miller, J. (2009). Determining What Individual SUS Scores Mean : Adding an Adjective Rating Scale. *Journal of Usability Studies*, 114-123.
- Daqiqil, I. (2011). Framework Codeigniter Sebuah Panduan dan Best Practice. Pekanbaru.
- Enterprise, J. (2017). *Otodidak MySQL untuk Pemula*. Jakarta: PT Elex Media Komputindo.
- Hammer, M., & Champy, J. (1993). *Reengineering the Corporation: A Manifesto for Business Revolution*. Harper Business.
- Isnandi. (2014). Sistem Informasi Penjualan Tiket pada Al Fath Tours dan Travel Pacitan. *Speed Journal*, Volume 11 No 2.
- Kurniawan, A., Rokhmawati, R. I., & Rachmadi, A. (2018). Evaluasi User Experience dengan Metode Heuristic Evaluation dan Persona (Studi pada : Situs Web Dalang Ki Purbo Asmoro). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol.2, No.8*, 2918-2926.
- Nielsen, J. (2012, January 4). *Usability 101: Introduction to Usability*. Retrieved from Nielsen Norman Group: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>
- Pressman, R. S., & Maxim, R. B. (2015). *Software Engineering A Practitioners Approach Eighth Edition*. New York: McGraw-Hill Education.
- Sommerville, I. (2011). *Software Engineering Ninth Edition*. Pearson Education Inc.
- Zubaidi, A. (2015). Sistem Informasi Biro Perjalanan Tour Berbasis Website dengan Menggunakan Framework Codeigniter pada Anggada Indonesia Tour dan Travel. *J-Intech*, Volume 03 Nomor 01.