

**APLIKASI ONLINE PEMBELIAN TIKET BUS DENGAN
LAYANAN LOCATION BASED SERVICE**

SKRIPSI

**HERAWATI HASIBUAN
72153002**



**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2019**

**APLIKASI ONLINE PEMBELIAN TIKET BUS DENGAN
LAYANAN LOCATION BASED SERVICE**

SKRIPSI

Diajukan untuk Memenuhi Syarat Mencapai Gelar Sarjana Komputer

HERAWATI HASIBUAN

72153002



**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2019**

PERSETUJUAN SKRIPSI

Hal : Surat Persetujuan Skripsi
Lamp : -

Kepada Yth.,
Dekan Fakultas Sains dan Teknologi
Universitas Islam Negeri Sumatera Utara Medan

Assalamu 'alaikum Wr. Wb.

Setelah membaca, meneliti, memberikan petunjuk, dan mengoreksi serta mengadakan perbaikan, maka kami selaku pembimbing berpendapat bahwa skripsi saudara,

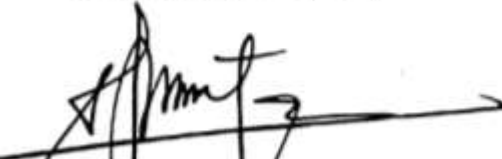
| | |
|-----------------------|--|
| Nama | : Herawati Hasibuan |
| Nomor Induk Mahasiswa | : 72153003 |
| Program Studi | : Sistem Informasi |
| Judul | : Aplikasi Online Pembelian Tiket Bus Dengan Layanan Location Based Service |

dapat disetujui untuk segera *dimunajasyahkan*. Atas perhatiannya kami ucapkan terimakasih.

Medan, 11 September 2019 M
11 Muharram 1441 H


Komisi Pembimbing,

Pembimbing Skripsi I,



Sanjudin, S.T., M.Kom
NIP. 197612172011011002

Pembimbing Skripsi II,



Triase, M.Kom
NIB. 1100000122

SURAT PERNYATAAN KEASLIAN SKRIPSI


Saya yang bertanda tangan di bawah ini,

Nama : Herawati Hasibuan
Nomor Induk Mahasiswa : 72153002
Program Studi : Sistem Informasi
Judul : Aplikasi Online Pembelian Tiket Bus
Dengan Layanan Location Based Service

menyatakan bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya. Apabila dikemudian hari ditemukan plagiat dalam skripsi ini maka saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi lainnya sesuai dengan peraturan yang berlaku.

Medan, 25 September 2019




Herawati Hasibuan
NIM. 72153002



**KEMENTERIAN AGAMA REPUBLIK INDONESIA
UNIVERSITAS ISLAM NEGERISUMATERA UTARA MEDAN
FAKULTAS SAINS DAN TEKNOLOGI**

Jl. IAIN No. 1 Medan 20235

Telp. (061) 6615683-6622925, Fax. (061) 6615683

Url: <http://saintek.uinsu.ac.id>, E-mail: saintek@uinsu.ac.id

PENGESAHAN SKRIPSI

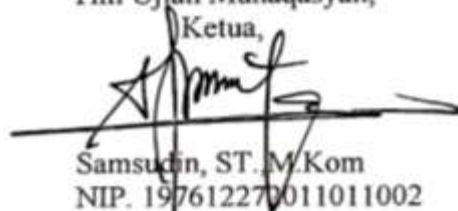
Nomor:027/ST/ST.V/PP.01.1/02/2020

Judul : Aplikasi Online Pembelian Tiket Bus Dengan Layanan
Location Based Service
Nama : Herawati Hasibuan
NomorIndukMahasiswa : 72153002
Program Studi : Sistem Informasi
Fakultas : Sains danTeknologi

Telah dipertahankan dihadapan Dewan Penguji Skripsi Program Studi Sistem Informasi
Fakultas Sains dan Teknologi UIN Sumatera Utara Medan dan dinyatakan **LULUS**.

Pada hari/tanggal : Senin, 11 November 2019
Tempat : Ruang Sidang Fakultas Sains dan Teknologi

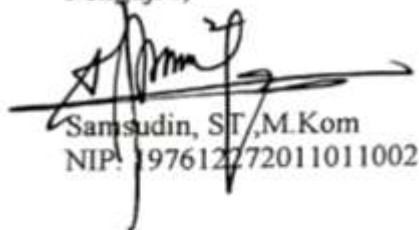
Tim Ujian Munaqasyah,
Ketua,



Samsudin, ST., M.Kom
NIP. 19761227011011002

Dewan Penguji,

Penguji I,



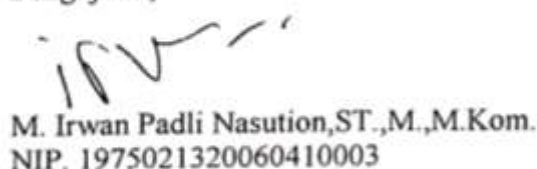
Samsudin, ST., M.Kom
NIP. 197612272011011002

Penguji II,



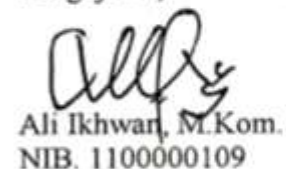
Triase, ST., M.Kom
NIB. 1100000122

Penguji III,



M. Irwan Padli Nasution, ST., M., M.Kom.
NIP. 1975021320060410003

Penguji IV,



Ali Ikhwan, M.Kom.
NIB. 1100000109

Mengesahkan,
Dekan Fakultas Sains dan Teknologi
Universitas Islam Sumatera Utara Medan,



Amil, M.A.
NIP. 19101999031002

ABSTRAK

Dalam dunia bisnis teknologi digunakan sebagai alat penghubung yang sangat praktis untuk komunikasi antara perusahaan dengan pelanggan tanpa harus memikirkan jarak dan waktu. Salah satu permasalahan yang terjadi adalah pelayanan dalam hal pemesanan tiket yaitu kurangnya penyampaian informasi mengenai jadwal keberangkatan, pemilihan rute dan harga tiket secara *online*. Dengan adanya permasalahan ini maka perlu dibuat aplikasi pemesanan tiket bus secara *online*. Program aplikasi ini dirancang berbasis android dengan memanfaatkan bahasa pemrograman *java* dan PHP serta layanan *Location Based Service* (LBS) yang dalam pengaplikasiannya memanfaatkan *Global Positioning System* (GPS) guna mempermudah dalam pengambilan lokasi calon penumpang yang akan mengakses informasi dan pemesanan tiket bus tanpa harus datang ke loket dan naik dari titik awal bus berangkat. Jika seorang *client* dapat memaksimalkan teknologi ini maka dia tidak perlu takut akan ketinggalan bus saat bus sudah berangkat ketika posisi *client* masih jauh dari titik awal bus berangkat. Hasil dari aplikasi yang dibangun adalah sebuah aplikasi *smartphone* berbasis android yang dapat memesan tiket bus dengan layanan lokasi naik yang diinginkan *client* dengan menggunakan layanan *Location Based Service* (LBS).

Kata kunci : Tiket, *Location Based Service*, *Global Positioning System* (GPS), Android.

ABSTRACT

In the business world, technology is used as a very practical interface for communication between companies and customers without having to think about distance and time. One of the problems that occurs is the service in terms of booking tickets, namely the lack of delivery of information about the scheduled departure, route selection and ticket prices online. With this problem, it is necessary to make an application for ordering bus tickets online. This application program is designed based on Android by utilizing Java and PHP programming languages and Location Based Service (LBS) services which in its application utilize the Global Positioning System (GPS) to facilitate the taking of the location of prospective passengers who will access information and booking bus tickets without having to come to window and ride from the starting point of the departing bus. If a client can maximize this technology, he does not need to be afraid to miss the bus when the bus has departed when the client's position is far from the starting point of the departing bus. The results of the application built is an android-based smartphone application that can order bus tickets with the desired ride location service of the client by using the Location Based Service (LBS) service.

Keywords: Ticket, Location Based Service, Global Positioning System (GPS), Android.

KATA PENGANTAR

Syukur Alhamdulillah kepada Allah SWT atas rahmatNya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Aplikasi Online Pembelian Tiket Bus Dengan Layanan Location Based Service”.

Penulisan skripsi ini dapat diselesaikan dengan bantuan baik moril maupun materil serta dorongan dan arahan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Saidurrahman, M.Ag. selaku Rektor UIN Sumatera Utara Medan.
2. Bapak Dr.H. M. Jamil, M.A. selaku Dekan Fakultas Sains dan Teknologi UIN Sumatera Utara Medan.
3. Bapak Samsudin S.T., M.Kom. Selaku Ketua Program Studi Sistem Informasi Fakultas Sains dan Teknologi UIN Sumatera Utara Medan.
4. Bapak Samsudin, S.T., M.Kom dan Ibu Triase, S.Kom selaku Pembimbing Skripsi yang telah memberikan motivasi dan bimbingan selama proses penyelesaian skripsi.
5. Bapak Muhammad Irwan Padli Nasution, S.T., M.M., M.Kom selaku dosen Penasehat Akademik yang telah memberikan bimbingan selama menempuh pendidikan di Fakultas Sain dan Teknologi UIN Sumatera Utara Medan.
6. Bapak Rahmad Kurniawan, M.Kom selaku Kepala Laboratorium Fakultas Sains dan Teknologi UIN Sumatera Utara Medan yang telah memfasilitasi penelitian dalam rangka penyelesaian skripsi.
7. Seluruh Bapak/Ibu dosen dan staff administrasi yang telah membantu selama proses perkuliahan.
8. Bapak Legimin Hasibuan dan Ibu Misni selaku orang tua saya yang telah membimbing dan mengarahkan dengan penuh kasih sayang serta memberikan arti sebuah kesabaran dalam menjalani kehidupan.
9. Teman dan sahabat sistem informasi stambuk 2015 yang senantiasa memberikan tawa, duka, semangat, dan motivasi.

Penulis berharap semoga skripsi ini dapat bermanfaat dan menambah wawasan keilmuan. Kritik dan saran yang sifatnya membangun sangat penulis harapkan untuk perbaikan dimasa yang akan datang.

Medan, 25 November 2019

Penulis

Herawati Hasibuan

DAFTAR ISI

| | Halaman |
|--|-------------|
| ABSTRAK | i |
| ABSTRACT | ii |
| KATA PENGANTAR..... | iii |
| DAFTAR ISI..... | v |
| DAFTAR GAMBAR..... | ix |
| DAFTAR TABEL | xii |
| DAFTAR LAMPIRAN..... | xiii |
| BAB I PENDAHULUAN | |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Rumusan Masalah..... | 2 |
| 1.3 Batasan Masalah | 3 |
| 1.4 Tujuan Penelitian | 3 |
| 1.5 Manfaat Penelitian | 3 |
| BAB II TINJAUAN PUSTAKA | |
| 2.1 Aplikasi..... | 5 |
| 2.2 Pengertian Tiket..... | 6 |
| 2.3 <i>Location Based Service</i> | 6 |
| 2.3.1 Unsur Utama <i>Location Based Service</i> | 7 |
| 2.3.2 Komponen <i>Location Based Service</i> | 7 |
| 2.3.3 Tantangan Pengguna <i>Location Based Service</i> | 9 |
| 2.4 Google Maps..... | 10 |
| 2.5 GPS (<i>Global Positioning System</i>)..... | 11 |
| 2.5.1 Sistem Koordinat Geografis | 11 |
| 2.6 Android..... | 12 |
| 2.6.1 Versi Android..... | 14 |
| 2.6.2 Arsitektur Android | 15 |
| 2.6.3 Komponen aplikasi Android | 16 |
| 2.6.4 Struktur aplikasi Android..... | 18 |
| 2.6.5 Software Pendukung Android | 19 |

| | |
|---|----|
| 2.7 UML (<i>Unified Modelling Language</i>) | 20 |
| 2.7.1 Desain Proses | 22 |
| 2.7.1.1 <i>Use Case Diagram</i> | 22 |
| 2.7.1.2 <i>Activity Diagram</i> | 25 |
| 2.7.1.3 <i>Sequence Diagram</i> | 26 |
| 2.7.2 Desain <i>Database</i> | 28 |
| 2.7.2.1 <i>Class Diagram</i> | 29 |
| 2.8 Metode Pengembangan Sistem..... | 32 |
| 2.8.1 Metode <i>Waterfall</i> | 32 |
| 2.9 Basis Data | 33 |
| 2.9.1 PHPMyadmin..... | 34 |
| 2.10 Perangkat Lunak Pendukung..... | 34 |
| 2.10.1 Android Studio | 35 |
| 2.10.2 MySql | 37 |
| 2.10.3 XAMPP | 37 |
| 2.10.4 Microsoft Visio..... | 37 |
| 2.10.4.1 Jenis File Visio | 38 |
| 2.11 Pengujian | 39 |
| 2.12 Studi Sejenis..... | 39 |

BAB III METODOLOGI PENELITIAN

| | |
|---|----|
| 3.1 Metodologi <i>CaseStudy</i> | 41 |
| 3.1.1 Tempat dan Waktu Penelitian Alat Penelitian | 41 |
| 3.1.2 Kebutuhan Sistem | 42 |
| 3.1.3 Teknik Pengolahan Data dan Analisis Data..... | 43 |
| 3.2 Cara Kerja..... | 44 |
| 3.2.1 Metode Pengumpulan Data..... | 44 |
| 3.2.1.1 Observasi | 44 |
| 3.2.1.2 Wawancara | 44 |
| 3.2.1.3 Analisis Dokumen | 45 |
| 3.2.2 Metode Pengembangan Sistem | 45 |
| 3.2.3 Kerangka Berfikir..... | 48 |

BAB IV HASIL DAN PEMBAHASAN

| | |
|---|----|
| 4.1 Analisis Kebutuhan..... | 49 |
| 4.1.1 Profil Perusahaan | 49 |
| 4.1.1.1 Visi dan Misi | 53 |
| 4.1.2 Analisis Sistem Berjalan | 54 |
| 4.1.2.1 Gambaran Analisis Sistem Berjalan | 55 |
| 4.1.2.2 Narasi Sistem Berjalan | 56 |
| 4.1.2.3 Identifikasi Masalah Sistem Berjalan | 56 |
| 4.1.3 Analisis Sistem Usulan | 56 |
| 4.1.3.1 Gambaran Analisis Sistem Berjalan | 57 |
| 4.1.3.2 Narasi Sistem Usulan..... | 58 |
| 4.1.4 Analisis Keputusan | 58 |
| 4.2 Desain Sistem | 59 |
| 4.2.1 Desain Proses | 59 |
| 4.2.1.1 <i>Use case diagram</i> Aplikasi Online Pembelian Tiket Bus Dengan Layanan <i>Location Based Service</i> | 59 |
| 4.2.1.2 <i>Activity diagram</i> Aplikasi Online Pembelian Tiket Bus Dengan Layanan <i>Location Based Service</i> | 62 |
| 4.2.1.3 <i>Sequence diagram</i> Aplikasi Online Pembelian Tiket Bus Dengan Layanan <i>Location Based Service</i> | 66 |
| 4.2.2 Desain Database | 71 |
| 4.2.2.1 Class Diagram Pembelian Tiket Bus | 71 |
| 4.2.2.2 Entity Relationship Diagram (ERD) Pembelian Tiket Bus..... | 72 |
| 4.2.2.3 Spesifikasi Database | 73 |
| 4.2.3 Desain Interface | 77 |
| 4.2.3.1 Rancangan Halaman <i>Admin</i> dan Agen (<i>Website</i>)..... | 78 |

| | |
|--|------------|
| 4.2.3.2 Rancangan Halaman <i>Client</i> dan Supir (Android) | 85 |
| 4.3 Implementasi | 89 |
| 4.3.1 Pengkodean | 89 |
| 4.3.2 Pengujian Sistem..... | 90 |
| BAB V KESIMPULAN DAN SARAN | |
| 5.1 Kesimpulan | 112 |
| 5.2 Saran | 112 |
| DAFTAR PUSTAKA | 113 |
| LAMPIRAN-LAMPIRAN | |

DAFTAR GAMBAR

| Gambar | Judul Gambar | Halaman |
|---------------|---|----------------|
| 2.1 | Kategori <i>Location Based Service</i> | 8 |
| 2.2 | Garis <i>Latitude</i> Dan <i>Longitude</i> | 12 |
| 2.3 | Logo Android | 13 |
| 2.4 | Arsitektur Android | 15 |
| 2.5 | Diagram UML | 21 |
| 2.6 | Contoh Diagram <i>Use Case</i> Perpustakaan | 24 |
| 2.7 | Contoh <i>Activity Diagram</i> Kaprodi..... | 26 |
| 2.8 | Contoh <i>Sequence Diagram</i> Perpustakaan | 28 |
| 2.9 | Contoh <i>Class Diagram</i> Perpustakaan..... | 31 |
| 2.10 | Model <i>Waterfaal</i> | 32 |
| 2.11 | Tampilan Android Studio | 36 |
| 3.1 | Kerangka Berfikir | 48 |
| 4.1 | Sistem Berjalan Pembelian Tiket Bus | 55 |
| 4.2 | Sistem Usulan Pembelian Tiket Bus | 57 |
| 4.3 | <i>Use Case Diagram</i> | 61 |
| 4.4 | <i>Activity Diagram</i> Master Data..... | 63 |
| 4.5 | <i>Activity Diagram</i> Input Data..... | 64 |
| 4.6 | <i>Activity Diagram</i> Pemesanan Tiket | 65 |
| 4.7 | <i>Activity Diagram</i> Laporan Pemesanan | 66 |
| 4.8 | <i>Sequence Diagram</i> Kelola Master Data..... | 67 |
| 4.9 | <i>Sequence Diagram</i> Input Data..... | 68 |
| 4.10 | <i>Sequence Diagram</i> Pemesanan..... | 69 |
| 4.11 | <i>Sequence Diagram</i> Lihat Laporan Pemesanan | 70 |
| 4.12 | <i>Class Diagram</i> | 71 |
| 4.13 | <i>Entity Relationship Diagram</i> | 72 |
| 4.14 | Rancangan Halaman Login Admin dan Agen..... | 79 |
| 4.15 | Rancangan Halaman Dashboard Admin..... | 79 |
| 4.16 | Rancangan Halaman Dashboard Agen | 79 |

| | | |
|------|---|----|
| 4.17 | Rancangan Halaman Menu Data Loker Admin..... | 81 |
| 4.18 | Rancangan Halaman Menu Data Agen | 81 |
| 4.19 | Rancangan Halaman Menu Data Bus | 82 |
| 4.20 | Rancangan Halaman Menu Data Supir | 83 |
| 4.21 | Rancangan Halaman Jadwal Keberangkatan..... | 83 |
| 4.22 | Rancangan Halaman Menu Data Titik Jemput..... | 84 |
| 4.23 | Rancangan Halaman Daftar Data Pemesanan | 84 |
| 4.24 | Rancangan Halaman Laporan Pemesanan Tiket | 85 |
| 4.25 | Rancangan Halaman <i>Login</i> dan Verifikasi Client | 86 |
| 4.26 | Rancangan Tampilan <i>Dashboard Client</i> | 86 |
| 4.27 | Rancangan Halaman Bus yang Tersedia | 87 |
| 4.28 | Rancangan Halaman <i>Input Data Client</i> | 87 |
| 4.28 | Rancangan Halaman Pilih Tipe Pembayaran | 88 |
| 4.30 | Rancangan Halaman Pilihan Lokasi Penjemputan | 88 |
| 4.31 | Rancangan Halaman Detail Pemesanan Tiket..... | 89 |
| 4.32 | <i>Interface</i> Login Admin dan Agen..... | 91 |
| 4.33 | Menampilkan Halaman Utama | 91 |
| 4.34 | Tampilan Master Data Admin | 92 |
| 4.35 | Tampilan Data Loker | 92 |
| 4.36 | Tampilan Tambah Loker | 93 |
| 4.37 | Tampilan Dapat <i>Username</i> dan <i>Password</i> Agen..... | 93 |
| 4.38 | Tampilan Hasil <i>Input Data Loker</i> | 94 |
| 4.39 | <i>Interface</i> Data Agen..... | 94 |
| 4.40 | Tampilan Tambah Data Agen..... | 95 |
| 4.41 | Tampilan Data Agen Setelah ditambah | 95 |
| 4.42 | <i>Interface</i> Data Bus | 96 |
| 4.43 | Tampilan Tambah Data Bus | 96 |
| 4.44 | <i>Interface</i> Data Supir..... | 97 |
| 4.45 | Tampilan Form Tambah Data Supir | 97 |
| 4.46 | Tampilan Dapat <i>Password</i> dan <i>Username</i> Supir..... | 98 |
| 4.47 | Tampilan Data Supir Yang Telah Ditambah | 98 |
| 4.48 | <i>Interface</i> Data Jadwal Keberangkatan..... | 99 |

| | | |
|------|--|-----|
| 4.49 | Form Tambah Data Jadwal Keberangkatan..... | 99 |
| 4.50 | Tampilan Menambahkan Data Jadwal Berangkat | 100 |
| 4.51 | <i>Interface</i> Data Titik Jemput | 100 |
| 4.52 | Tampilan Form Tambah Data Titik Jemput | 101 |
| 4.53 | Tampilan Setelah Tambah Data Titik Jemput | 101 |
| 4.54 | Tampilan Lihat Lokasi Titik Jemput | 102 |
| 4.55 | Tampilan Daftar Data Pemesanan | 102 |
| 4.56 | Tampilan Laporan Data Pemesanan Tiket Bus | 103 |
| 4.57 | Tampilan Print Laporan Pemesanan | 103 |
| 4.58 | Tampilan Login Aplikasi Android | 104 |
| 4.59 | Tampilan Menu Verifikasi dan Pengguna Baru | 104 |
| 4.60 | Tampilan Dashboard..... | 105 |
| 4.61 | Tampilan Bus yang Tersedia | 105 |
| 4.62 | Tampilan Input Data Calon Penumpang | 106 |
| 4.63 | Tampilan Pilihan Tipe Pembayaran | 106 |
| 4.64 | Tampilan Pilih Titik Penjemputan..... | 107 |
| 4.65 | Tampilan Detail Pemesanan Tiket..... | 107 |
| 4.66 | Tampilan Berhasil Melakukan Pemesanan Tiket Bus..... | 108 |
| 4.67 | Tampilan Menu Dashboard | 108 |
| 4.68 | Tampilan Pesananku..... | 109 |
| 4.69 | Tampilan Upload Bukti Transfer..... | 109 |
| 4.70 | Tampilan Konfirmasi Pembayaran Oleh Admin | 110 |
| 4.71 | Tampilan Posisi Bus dan Client..... | 110 |
| 4.72 | Halaman Login Supir..... | 111 |
| 4.73 | Tampilan Halaman Daftar Keberangkatan Supir | 111 |

DAFTAR TABEL

| Tabel | Judul Tabel | Halaman |
|--------------|---|----------------|
| 2.1 | Daftar Tingkatan Versi Android | 14 |
| 2.2 | Direktori Struktur Aplikasi Android | 18 |
| 2.3 | Daftar Simbol <i>Use Case Diagram</i> | 23 |
| 2.4 | Daftar Simbol <i>Activity Diagram</i> | 25 |
| 2.5 | Daftar Simbol <i>Sequence Diagram</i> | 27 |
| 2.6 | Daftar Simbol Dalam <i>Class Diagram</i> | 29 |
| 2.7 | Indikator <i>Class Diagram</i> | 29 |
| 2.8 | Studi Sejenis..... | 39 |
| 4.1 | Jadwal Keberangkatan PT. Chandra | 50 |
| 4.2 | Jadwal Keberangkatan Bus KPB | 51 |
| 4.3 | Jadwal Keberangkatan PT Sentosa | 52 |
| 4.4 | Identifikasi <i>Actor</i> | 59 |
| 4.5 | Spesifikasi Data Agen | 73 |
| 4.6 | Spesifikasi Data Bus | 73 |
| 4.7 | Spesifikasi Data Pesan Bus | 74 |
| 4.8 | Spesifikasi Data Loket | 75 |
| 4.9 | Spesifikasi Data Pengguna..... | 75 |
| 4.10 | Spesifikasi Data Supir | 76 |
| 4.11 | Spesifikasi Data Titik Jemput | 76 |
| 4.12 | Spesifikasi Data Tujuan | 77 |
| 4.13 | Spesifikasi Data User | 78 |
| 4.14 | Tabel Pengujian Metode <i>Black Box</i> Admin dan Agen..... | 91 |
| 4.15 | Tabel Pengujian Metode <i>Black Box Client</i> | 104 |
| 4.16 | Tabel Pengujian Metode <i>Black Box</i> Supir | 111 |

DAFTAR LAMPIRAN

| Lampiran | Judul Lampiran |
|----------|--------------------------------|
| 1. | Kuisisioner Pengujian Aplikasi |
| 2. | <i>Source Code</i> |

BAB I PENDAHULUAN

1.1 Latar Belakang

Dalam dunia bisnis, internet digunakan sebagai alat penghubung yang sangat praktis untuk komunikasi antara perusahaan dengan pelanggan tanpa harus memikirkan jarak dan waktu. Dalam dunia usaha, komputer memiliki peranan yang cukup besar dalam hal penyediaan informasi baik bagi pihak perusahaan ataupun pihak luar perusahaan. Sebagai contoh dalam melakukan transaksi pemesanan dan pembelian melalui internet sangat cepat dan praktis meski harus melewati langkah-langkah pembayaran yang berbeda-beda.

PT Chandra, KPB (Kota Pinang Baru) dan Sentosa yang beralamat di Jln SM. Raja Km 7 No. 11 merupakan perusahaan yang bergerak dibidang transportasi yaitu pengangkutan bus umum yang melalui jalur trayek AKAP (Antar Kota Antar Propinsi). Sebagai perusahaan yang profesional dibidangnya, ketiga dari perusahaan tersebut yaitu PT Chandra, KPB dan Sentosa senantiasa berusaha memberikan pelayanan cepat, mudah dan memuaskan pelanggannya. Namun dalam pembelian tiket masih sangat manual yang dilakukan dengan harus mendatangi loket, sehingga sangat direkomendasikan untuk mengadakan perbaikan dalam kinerja yang akan mengarahkan perusahaan menuju ke sistem yang lebih akurat.

Berdasarkan survei diawal pada tanggal 22 Juni 2019 kepada calon penumpang telah ditemukan permasalahan dilapangan bahwasanya pembelian tiket bus sangat diresahkan dengan alasan pembelian tiket bus yang sulit dikarenakan pembelian tiket masih sangat manual seperti antri dalam loket hanya untuk membeli tiket dan mengetahui jadwal keberangkatan melalui *outlet*. Adapula pemesanan melalui telepon yang sudah *booking* namun belum melakukan pembayaran sering mengalami kasus tertinggal bus saat sudah memesan tiket dikarenakan pihak bus tidak memberi kabar apabila bus telah berangkat, sehingga para calon penumpang merasa kecewa.

Dengan kondisi pembelian tiket yang diresahkan calon penumpang , maka perlu dibuat aplikasi pembelian tiket bus dengan layanan *Location Based Service* (LBS) agar dapat membantu menyelesaikan masalah dalam akses pembelian tiket bus dan mempermudah *client* untuk menentukan lokasi naik yang diinginkan saat melakukan pemesanan tiket bus tanpa khawatir tertinggal bus ketika sudah berangkat dari titik awal bus berangkat. *Location Based Service* adalah layanan berbasis lokasi yang dapat mengetahui posisi pengguna (calon penumpang) dan dapat menentukan posisi tempat-tempat tertentu. Penggunaan aplikasi LBS dapat menghubungkan posisi *user* dengan pihak loket untuk dapat memberikan informasi pembelian tiket dan jadwal keberangkatan tanpa harus naik dari titik awal bus berangkat dengan jadwal yang telah ditentukan. Melalui teknologi tersebut, *client* dapat melakukan transaksi pembelian tiket online kapan saja dan dimana saja tanpa memikirkan jarak dan waktu.

Berdasarkan masalah diatas maka peneliti mengajukan “**Aplikasi Online Pembelian Tiket Bus Dengan Layanan *Location Based Service***” untuk mengatasi masalah sulitnya pemesanan tiket keberangkatan. Dengan pembelian *online*, para konsumen bisa mengakses pemesanan tiket yang diharapkan dapat membantu mempersingkat dan memudahkan calon penumpang untuk memesan tiket bus.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka dirumuskan permasalahannya adalah

1. Bagaimana membangun aplikasi tiket *online* dengan menggunakan Android dan *Location Based Service* ?
2. Bagaimana mendapatkan informasi keberangkatan pemesanan tiket bus di PT Chandra, Kota Pinang Baru dan PT Sentosa ?

1.3 Batasan Masalah

Adapun batasan masalah dari penelitian ini adalah sebagai berikut :

1. Aplikasi yang dikembangkan dapat memberikan informasi cara pembelian tiket bus, jadwal keberangkatan bus, harga tiket bus, dan menentukan lokasi naik.
2. Aplikasi pembelian tiket *online* bus berbasis Android.
3. Pembayaran dilakukan melalui transaksi bank dan COD (*Cash Order Delivery*).
4. Aplikasi yang dibangun hanya memiliki tiga armada bus yaitu : PT Chandra, KPBB (Kota Pinang Baru) dan Sentosa.
5. Terdapat empat tingkatan *class* Bus yang terdiri dari *class Economy*, *class Ac toilet*, *class Ac non toilet*, dan *Super Executive*.
6. Pengambilan LBS dalam aplikasi ini menggunakan Google Maps API dan GPS.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian pembelian tiket bus *online* dengan layanan *location based service* adalah:

1. Untuk pembuatan fitur informasi berupa aplikasi *online* pembelian tiket bus dengan layanan *Location Based Service* (LBS).
2. Memudahkan penyampaian informasi dan pemesanan tiket.
3. Dapat menghemat waktu dan biaya.
4. Menambah pengetahuan tentang program aplikasi berbasis android.

1.5 Manfaat Penelitian

Manfaat dari melakukan penelitian atas pengembangan Aplikasi *Online* Pembelian Tiket Bus Dengan Layanan *Location Based Service* ini adalah sebagai berikut :

1. Sebagai bahan masukan bagi perusahaan untuk mengadakan perbaikan kinerja serta mengarahkan perusahaan menuju ke sistem

yang lebih akurat dengan bantuan teknologi khususnya dalam pembelian tiket.

2. Memudahkan konsumen dalam hal pembelian tiket bus.
3. Memudahkan konsumen dalam hal menentukan lokasi loket keberangkatan untuk naik bus sesuai dengan rute Medan – Pekan baru.
4. Diharapkan menjadi tambahan referensi bahan penelitian mahasiswa lain yang lebih mendalam di masa akan datang.
5. Sebagai penambah pengetahuan dalam pengembangan aplikasi online pembelian tiket bus dengan layanan *Location Based Service*.
7. Memperoleh kesempatan untuk melatih keterampilan yang ada dalam melakukan pekerjaan nantinya khususnya dalam pengembangan sistem pembelian tiket bus.

BAB II

LANDASAN TEORI

2.1 Aplikasi

Aplikasi adalah sebuah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi salah satu tehnik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya aplikasi merupakan suatu perangkat komputer yang siap pakai bagi *user*.

Pengertian aplikasi menurut para ahli adalah sebagai berikut :

1. Menurut Jugianto (1999:12) adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output*.
2. Menurut Kamus Besar Bahasa Indonesia (1998:52) adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna.
3. Menurut Rachmad Hakim S, Aplikasi adalah perangkat lunak yang digunakan untuk tujuan tertentu, seperti mengelolah dokumen, mengatur *windows* & permainan (*game*) dan sebagainya.

Menurut Harip Santoso, Aplikasi adalah suatu kelompok aktivitas tertentu yang saling terkait, misalnya aplikasi *payroll*, aplikasi *fixed assets*, dan lain-lain (Alfeno & Devi, 2017).

2.2 Pengertian Tiket

Tiket adalah salah satu dokumen perjalanan yang dikeluarkan oleh maskapai penerangan dan merupakan kontrak tertulis satu pihak yang berisikan ketentuan yang harus dipenuhi oleh penumpang selama memakai jasa penerbangan, dan data penerbangan penumpang yang mempunyai masa periode waktu tertentu. Tiket merupakan dokumen pasasi yang berisi perjanjian kontrak antara perusahaan dengan penumpang yang memuat ketentuan yang telah disepakati bersama dan ditaati oleh kedua belah pihak. Menurut wilayah operasi, tiket terbagi menjadi 2 yaitu Tiket domestik dan Tiket International. Tiket domestik adalah tiket yang dikeluarkan untuk berpergian didalam negeri dan untuk tiket domestik ada yang berbentuk elektronik tiket yang seperti dikeluarkan untuk digunakan didalam negeri. Tiket Internasional adalah tiket yang dipergunakan untuk ke luar negri dan hamir semua tiket yang dikeluarkan oleh tiket internasional sudah berupa elektronik tiket (Rahmat & Octaviano, 2016).

2.3 *Location Based Service (LBS)*

LBS adalah layanan berbasis lokasi yang berfungsi untuk mencari dengan teknologi *Global Positioning System (GPS)* dan *Google's cell based location*. Map dan layanan berbasis lokasi menggunakan lintang dan bujur untuk menentukan lokasi geografis. Android menyediakan *geocoder* yang mendukung *forward* dan *reverse geocoding*. Menggunakan *geocoder*, dapat mengkovarsi nilai lintang bujur menjadi alamat dunia nyata atau sebaliknya. *Location based service* atau layanan berbasis lokasi adalah istilah umum yang digunakan untuk menggambarkan teknologi yang digunakan untuk menemukan lokasi perangkat yang kita gunakan (Alfeno & Devi, 2017).

LBS termasuk dalam kategori teknologi yang sama dengan *geographic information system (GIS)*, dan aplikasi *global positioning system (GPS)*, yaitu dikenal dengan teknologi geospital. Teknologi ini terdiri atas perangkat untuk mengumpulkan, menyimpan, menganalisa dan mendistribusikan data yang sesuai dengan kebutuhan pengguna terhadap sistem koordinat bumi. Layanan ini menjadi sangat penting bagi

penggunanya karena mampu menghubungkan antara lokasi *geographic* informasi terhadap lokasi penggunanya, hal ini sangat mendukung *era mobilitas* seperti pada masa kini. Identifikasi koordinat pengguna memungkinkan aplikasi LBS untuk menyediakan layanan bagi pengguna perangkat *mobile* (Anwar, Jaya, Kusuma, Studi, & Komputer, 2014) .

2.3.1 Unsur Utama *Location Based Service*

Dua unsur utama dari *Location Based Service* adalah:

1. *Location Manager (API Maps)*

Menyediakan perangkat bagi sumber atau *source* untuk LBS, *Application Programming Interface (API) maps* menyediakan fasilitas untuk menampilkan atau memanipulasi data.

2. *Location Providers (API Location)*

Menyediakan teknologi pencarian lokasi yang digunakan oleh perangkat. *API location* berhubungan dengan data GPS dan data lokasi *real-time*. *API Location* berada pada paket Android yaitu dalam paket “android.location”. Lokasi, perpindahan, serta kedekatan dengan lokasi tertentu dapat ditemukan melalui *Location manager* (Alfeno & Devi, 2017).

2.3.2 Komponen *Location Based Service*

Pengguna layanan berbasis lokasi ini juga memerlukan beberapa komponen. Beberapa elemen yang digunakan antara lain:

1. *Mobile Device*

Sebuah alat yang digunakan oleh pengguna untuk meminta informasi yang dibutuhkan. Perangkat memungkinkan yaitu perangkat yang memiliki fasilitas navigasi seperti PDA, *mobile phone*, laptop dan lainnya.

2. *Communication Network*

Jaringan selular yang mengirimkan data pengguna dan permintaan layanan.

3. *Position Component*

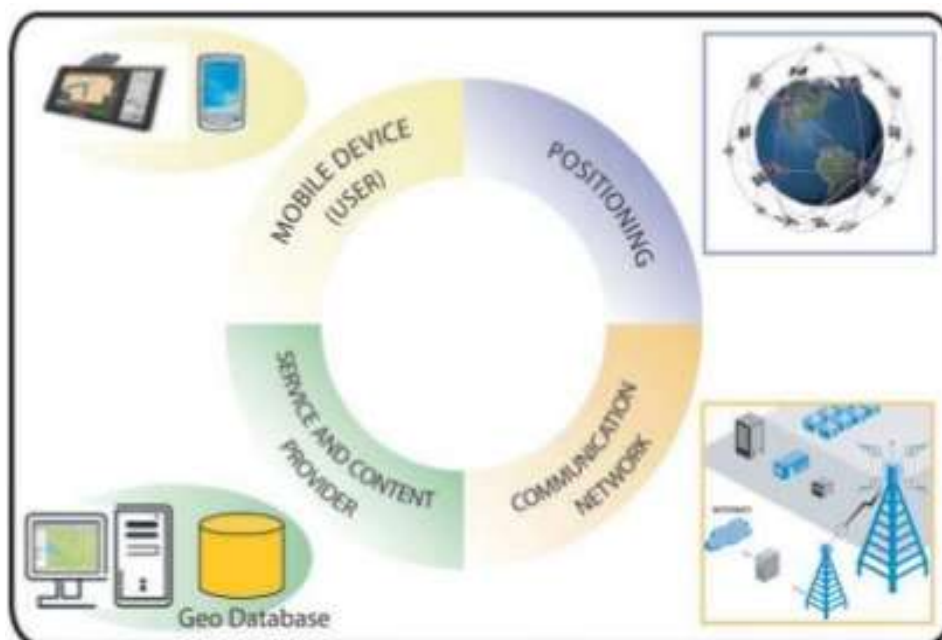
Biasanya posisi pengguna harus ditentukan untuk pengolahan layanan. Posisi pengguna dapat diperoleh menggunakan jaringan komunikasi atau dengan menggunakan *Global Positioning System (GPS)*.

4. *Service and Content Provider*

Penyedia layanan pengguna selular yang bertanggung jawab untuk proses layanan.

5. *Data and content Provider*

Penyedia layanan informasi data yang dapat diminta oleh pengguna (Nugroho, Nurhayati, & Widiyanto, 2015).



Gambar 2.1 Kategori *Location Based Service* (Sidiq Permana, 2013)

2.3.3 Tantangan Penggunaan *Location Based Service (LBS)*

Selain memberikan keuntungan, tentu ada beberapa tantangan yang harus dihadapi ketika menggunakan layanan berbasis lokasi. Berikut disajikan beberapa tantangan yang harus dihadapi.

1. Konsumsi Energi

Tingkat konsumsi energy penyedia layanan berbeda-beda. Ketika aktif, piranti GPS paling banyak menguras energi dibanding Wi-fi. Oleh

karena itu, penting untuk memastikan piranti ini bila tidak diperlakukan atau aplikasi dalam kondisi dihentikan sementara (*pause*).

2. Akurasi

Tingkat akurasi penyedia layanan lokasi juga berbeda-beda. *LocationManager.NETWORK_PROVIDER* memiliki akurasi terendah. *LocationManager.GPS_PROVIDER* memiliki akurasi lebih baik. Kesalahan pada GPS hanya sekitar beberapa meter sedangkan Wi-fi dapat meleset beberapa kilometer.

3. Ketersediaan Layanan

Layanan berbasis lokasi dengan GPS, Wi-fi hanya dapat dinikmati apabila piranti pengguna mendukungnya. Namun, meskipun sudah mendukung, tetap tidak ada jaminan layanan akses selalu tersedia. Sinyal GPS akan bagus apabila pengguna berada di ruang terbuka dan posisi pancaran sinyal dari satelit tidak terhalang. Bila berada di bawah tanah atau ruangan berdinding tebal, penerimaan sinyal GPS mungkin buruk karena tidak cukup kuat untuk menembusnya. Hal yang serupa mungkin pengguna alami bila menggunakan Wi-fi.

4. Waktu Untuk Memperoleh Data Lokasi

Penentuan lokasi piranti penerima GPS membutuhkan waktu yang lama dibandingkan Wi-fi. Lamanya waktu beda-beda tergantung piranti penerima GPS. Sebagian besar ponsel yang dilengkapi dengan GPS menemukan lokasi dalam waktu 20-60 detik. Hal terbaik yang dapat dilakukan selama menunggu piranti GPS menemukan lokasi adalah menggunakan data lokasi yang sudah diketahui sebelumnya (Juhara, 2016).

2.4 Google Maps

Google Maps adalah sebuah jasa peta *globe virtual* gratis dan *online* disediakan oleh *google*. Fasilitas *GoogleMaps* dihadirkan oleh *google* sejak tahun 2005 dan terus berkembang hingga sekarang ini. Di dalam *GoogleMaps*, tidak hanya mendapatkan tampilan peta dunia, namun juga informasi pendukung berupa informasi jalan, lokasi layanan publik, bisnis

dan sebagainya. *GoogleMap* menawarkan layanan API (*Application Programming Interface*) yang memungkinkan peta untuk dimasukkan pada situs web pihak ketiga (Alfeno & Devi, 2017).

Dengan kata lain, *GoogleMaps* merupakan suatu peta yang dapat kita lihat dengan menggunakan suatu *browser*. Kita dapat menambahkan fitur *GoogleMaps* dalam *web* yang telah kita buat yang berbayar maupun gratis sekalipun dengan *GoogleMaps API*. *GoogleMaps API* adalah suatu *library* yang berbentuk *JavaScript*. Dengan menggunakan *GoogleMaps API*, kita dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal, sehingga kita dapat focus hanya pada data-data yang akan ditampilkan. Dengan kata lain, kita hanya membuat suatu data sedangkan peta yang akan ditampilkan adalah milik *Google* sehingga kita tidak dipusingkan dengan membuat peta suatu lokasi, bahkan dunia.

Pada *Google Maps API* terdapat 4 jenis pilihan model peta yang disediakan oleh *Google*, diantaranya adalah :

1. *ROADMAP*, untuk menampilkan peta biasa 2 dimensi.
2. *SATELITE*, untuk menampilkan foto satelit.
3. *TERRAIN*, untuk menunjukkan relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi, contohnya akan menunjukkan gunung dan sungai.
4. *HYBRID*, akan menunjukkan foto satelit yang di atasnya tergambar pula apa yang tampil pada *ROADMAP* (jalan dan nama kota) (Rompas, 2015).

2.5 GPS (*Global Positioning System*)

GPS adalah singkatan dari *Global Positioning System*, yang merupakan sistem navigasi dengan menggunakan teknologi satelit yang dapat menerima sinyal dari satelit yang awalnya dikembangkan untuk menunjang kegiatan militer, namun kemudian dibuka untuk umum. Teknologi GPS terdiri dari konstelasi 27 satelit yang mengelilingi bumi (24 satelit aktif ditambahkan 3 satelit cadangan). Tiap satelit mengelilingi bumi dua kali dalam satu hari. Orbit tiap satelit sudah diatur sehingga pada tiap titik di

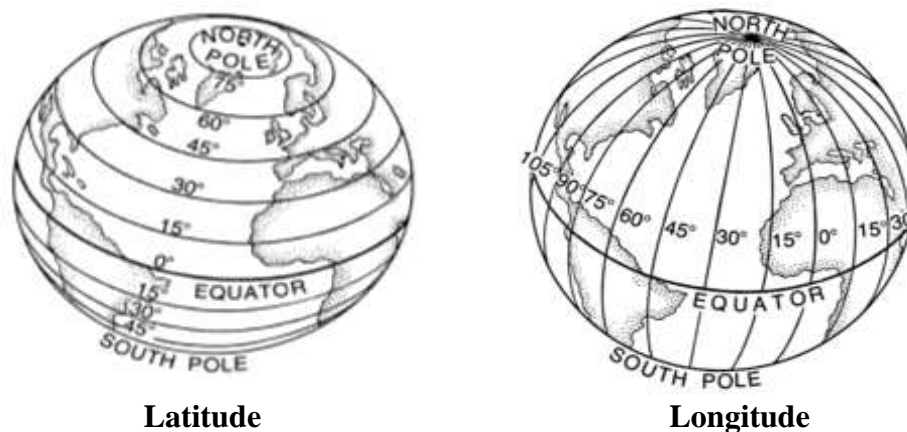
seluruh permukaan bumi, orang akan menemukan paling tidak 4 satelit di langit tiap saat.

Mengapa 4 satelit, bicara piranti GPS artinya mengacu pada piranti penerima yang tugas utamanya adalah menemukan lokasi empat satelit atau lebih, mendapatkan jarak piranti penerima ke masing-masing satelit, dan menentukan lokasi piranti penerima berdasarkan jarak ke masing-masing satelit (Juhara, 2016).

2.5.1 Sistem Koordinat Geografis

Untuk keperluan pemetaan, permukaan bumi dipetakan menggunakan sistem koordinat 2D. Tiap titik di permukaan bumi diacu oleh dua besaran, yakni *latitude* dan *longitude*. *Latitude* (Lintang) adalah jarak anguler arah Utara atau Selatan terhadap khatulistiwa dan dinyatakan dalam satuan derajat ($^{\circ}$). Pengguna dapat menambahkan menit dan detik pada *latitude* untuk pengukuran yang lebih rinci. Contoh, daerah Kutub Utara memiliki *latitude* $+90^{\circ}$ dan Kutub Selatan memiliki nilai *latitude* -90° . Kota Pontianak memiliki *latitude* 0° .

Longitude (bujur) adalah jarak anguler arah Timur atau Barat terhadap *Prime Meridian* yang melintasi *Greenwich*, Inggris. *Longitude* juga dinyatakan dalam satuan sudut derajat ($^{\circ}$) dan dapat ditambahkan menit dan detik untuk pengukuran yang lebih rinci. *Longitude* berkisar -180° (ke arah Barat) hingga $+180^{\circ}$ (ke arah Timur). *Greenwich* ditentukan berada pada *longitude* 0° (Juhara, 2016).



Gambar 2.2 Garis *Latitude* dan *Longitude* (Juhara, 2016)

2.6 Android

Menurut Irwan Padli Nasution & Samsudin (2018) Android adalah sistem operasi untuk perangkat seluler berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan *platform* terbuka bagi pengembang untuk membuat aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat perangkat lunak untuk ponsel cerdas dan kemudian mengembangkan Android membentuk *Open Handset Alliance*, sebuah konsorsium 34 perangkat keras perusahaan, perangkat lunak dan telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia*. Pertama ponsel yang menggunakan sistem operasi Android adalah HTC mimpi, yang dirilis pada 22 Oktober 2008. Di akhir 2010, diperkirakan hampir semua vendor ponsel di dunia gunakan android sebagai sistem operasinya. Popularitas Android terus meningkat, tidak hanya dari jumlah perangkat yang mengadopsi ini *platform*, tetapi juga operator telekomunikasi yang mulai menyediakan lebih fokus pada *platform opensource* yang dikembangkan Google. Perangkat lunak android sebagai *platform* yang lengkap, terbuka, gratis, dan informasi lainnya dapat diunduh dengan bebas dan lengkap dengan mengunjungi situs web:<http://developer.android.com> (Irwan Padli Nasution & Samsudin, 2018).

Menurut Tim (2016) Android adalah nama *software* yang dipakai pada perangkat mobile yang mencakup berbagai komponen, yaitu sistem operasi, *middle-ware* dan aplikasi kunci yang diliris oleh *google*. Jadi, *android* ini mencakup keseluruhan aplikasi, mulai dari sistem operasi hingga pengembangan aplikasi itu sendiri. Dan pengembangan aplikasi pada *platform android* ini menggunakan dasar bahasa pemograman *Java*. *Platform* pengembangan aplikasi *android* yang merupakan bagian dari *android* memiliki lisensi *open-source* atau terbuka, sehingga anda dapat membangun aplikasi yang kaya dan inovatif (Tim, 2013).

Menurut Huda (2014) Android merupakan sebuah sistem operasi berbasis Linux yang didesain khusus untuk perangkat bergerak seperti *smarphone* atau tablet. Sistem operasi Android bersifat *open source*

sehingga sangat membantu kita dalam mengembangkan aplikasi Android karena banyak kita jumpai programmer lain yang berbagi tutorial maupun tips dan trik mengembangkan aplikasi Android. Selain itu, Android bisa juga dikembangkan menggunakan *framework* lain seperti:

1. *Basic4android* : bahasa pemrograman Basic.
2. *Adobe AIR* : bahasa pemrograman *ActionScript*.
3. *PhoneGap* : bahasa pemrograman HTML5 dan JavaScript.
4. *PHP for Android* : bahasa pemrograman PHP.
5. *AndEngine* : *Framework* untuk mengembangkan game Android (Huda, 2014).



Gambar 2.3 Logo Android (Tim, 2013)

2.6.1 Versi Android

Android telah mengalami sejumlah pembaruan sejak pertama kali diliris. Rata-rata, versi terbaru dari Android diliris tiap 6 bulan. Tabel 2.1 menunjukkan beberapa jenis android dan nama kodenya. Penamaan kode menggunakan nama makanan dan huruf depannyaurut sesuai abjad (Juhara, 2016). Adapun daftar versi android bisa dilihat pada tabel 2.1 dibawah ini.

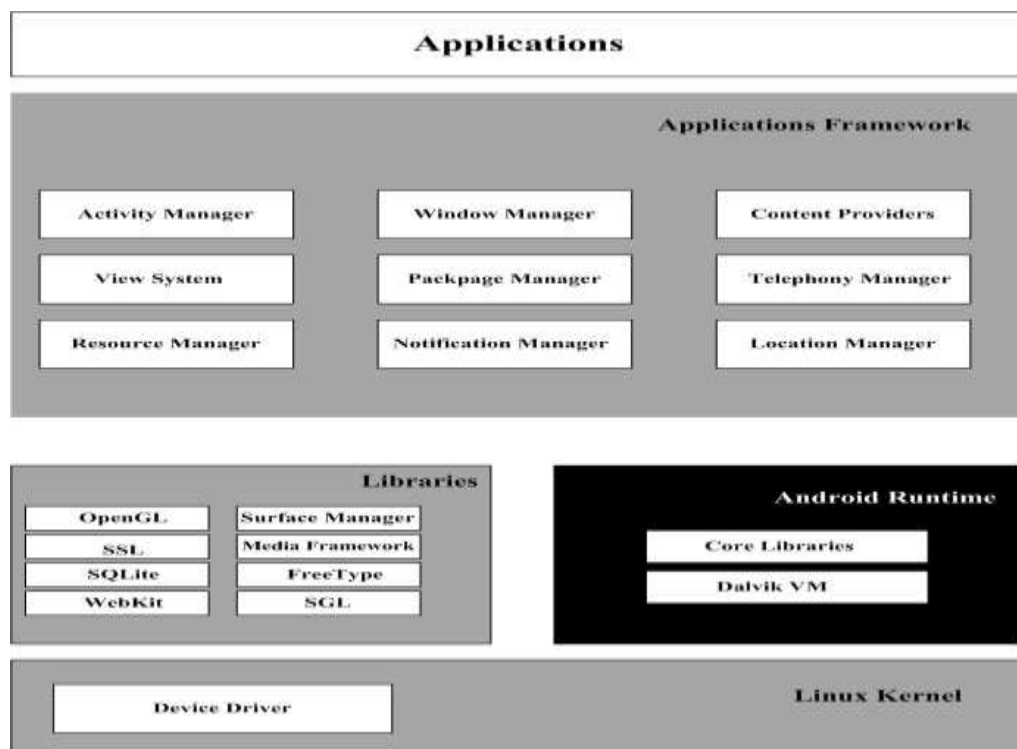
Tabel 2.1 Daftar Tingkatan Versi Android

| No | Nama | Versi | Tahun Peluncuran |
|----|--------------------|--------------|--------------------------|
| 1 | Cupcake | 1.5 | 27 April 2009 |
| 2 | Donut | 1.6 | 15 September 2009 |
| 3 | Eclair | 2.0 – 2.1 | 26 Oktober 2009 |
| 4 | Froyo | 2.2 – 2.2.3 | 20 Mei 2010 |
| 5 | Gingerbread | 2.3 – 2.3.7 | 6 Desember 2010 |
| 6 | Honeycomb | 3.0–3.2.6 | 22 Pebruari 2011 |
| 7 | Ice Cream Sandwich | 4.0 – 4.0.4 | 18 Oktober 2011 |
| 8 | Jelly Bean | 4.1 – 4.3.1 | 9 Juli 2012 |
| 9 | KitKat | 4.4 – 4.4.4 | 31 Oktobe 2013 |
| 10 | Lollipop | 55.0 – 5.1.1 | 12 November 2014 |
| 11 | Marshmallow | 6.0 – 6.0.1 | 5 Oktober 2015 |
| 12 | Nougat | 7.0 | Agustus / September 2016 |
| 13 | Oreo | 8.0 | Agustus 2017 |
| 14 | Pie | 9.0 | Agustus 2018 |

Berdasarkan penjelasan tersebut, maka versi android merupakan sistem operasi android yang dari masa kemasa memiliki perubahan serta penambahan *fiture* yang membuatnya lebih baik dari versi sebelumnya, seperti halnya sistem operasi terakhir android yaitu pie yang sudah ada penggunaan *artificial intelligence* (AI).

2.6.2 Arsitektur Android

Secara umum, arsitektur android terdiri dari lapisan *software*, yaitu lapisan aplikasi, lapisan *frame work*, Android *run-time* sebagai lapisan perantara (*middleware*), serta kernel Linux yang membungkus *hardware* di bawahnya.



Gambar 2.4 Arsitektur Android (Juhara, 2016)

Keterangan gambar 2.4 adalah sebagai berikut :

1. Aplikasi

Aplikasi yang anda buat berada pada level tertinggi dalam *hierarki* struktur *software* di Android. Termasuk di dalamnya aplikasi utama bawaan Android seperti telepon, pembaca email, program SMS, kontak, browser, dan kalender. Aplikasi umumnya ditulis dalam bahasa pemrograman Java.

2. Framework Aplikasi

Di bawah aplikasi terdapat sejumlah *software* pendukung, meliputi layanan pengelola *Activity*, *View*, Sumber daya, pemebritahuan (*notifikasi*) dan lain-lain.

3. Pustaka

Android menyediakan sejumlah pustaka dasar seperti pustaka grafis 2D dan 3D, *multimedia playback*, *browser engine*, pencetakan *font*, *database* dan lain-lain. Aplikasi yang memanfaatkan pustaka fungsi ini melalui lapisan *framework* aplikasi.

4. Android *Runtime*

Android menyediakan pustaka inti bagi pemrograman java. Di bawahnya terdapat Dalvik Virtual Machine (Dalvik VM) yang akan menjalankan aplikasi. Tiap aplikasi akan dijalankan pada proses terpisah dengan VM yang berbeda sehingga terisolasi satu sama lain. Aplikasi masih dapat berkomunikasi dengan aplikasi pada lapisan di bawahnya (kernel linux) untuk *multi-threading* dan pengelolaan memori tingkat rendah.

5. Kernal Linux

Kernel Linux bertanggung jawab menyediakan layanan dasar seperti keamanan, pengelolaan proses, pengelolaan file, pengelolaan sumber daya memori, dan *hardware* (Juhara, 2016).

2.6.3 Komponen Aplikasi Android

Aplikasi Android ditulis dengan bahasa pemrograman java. Semua file kode *intermediate* dan asset disatukan dalam satu paket berupa file bereksensi. APK, sebuah file yang dapat didistribusi. Tiap file APK adalah sebuah aplikasi tunggal. Komponen aplikasi android terdiri dari beberapa jenis, antara lain :

1. *Activity*

Activity adalah istilah yang digunakan dalam pemrograman Android untuk mengacu pada satuan interaksi dengan pengguna melalui antarmuka grafis (*graphical user-interface, GUI*). Sebagai salah satuan inteaksi, *Activity* adalah tampilan yang dilihat di layar seperti *windows* atau kotak dialog pada pemrograman aplikasi desktop. Tiap aplikasi dapat terdiri dari nol atau lebih *activity*. Selain sebagai satuan interaksi dengan pengguna, *Activity* juga satuan eksekusi. Sebagai satuan eksekusi, *Activity* selalu memiliki paling tidak satu buah *Thread*, yakni *Thread* utama yang digunakan untuk memperbarui tampilan *user-interface* (UI Thread).

2. *Intent*

Intent adalah istilah yang digunakan dalam pemrograman Android untuk mengacu pada mekanisme berbagai pesan pemberitahuan atau bertukar data *activity* atau untuk menjalankan aplikasi lain.

3. *Service*

Service adalah komponen aplikasi yang berjalan di belakang layar tanpa *user-interface* untuk menyediakan layanan tertentu seperti mengecek RSS *feed* secara kontinu atau memainkan music. *Service* tetap berjalan meski *Activity* yang mengendalikannya telah berhenti. Media player adalah sebuah contoh aplikasi yang menggunakan *Service*.

4. *Content Provider*

Content Provider membuat suatu aplikasi dapat berbagi sejumlah data tertentu kepada aplikasi lain. Jika membutuhkan data nama-nama kontak, aplikasi tinggal meminta data tersebut.

5. *Broadcast Receiver*

Broadcast Receiver adalah komponen yang memantau, menerima, dan beraksi terhadap pesan yang disebarkan, baik oleh sistem maupun aplikasi lain. Misalnya, ketika baterai lemah, Android akan mengirim pesan “baterai lemah” kepada semua *broadcast receiver* yang ingin diberihatu pesan ini. Untuk menggunakan *broadcast receiver*, pada dasarnya, Anda hanya perlu membuat turunan tipe *broadcast receiver*, melengkapi metode *onReceive()*, dan mendaftarkannya di *AndroidManifest.xml* atau dengan metode *Context.registerReceiver()*. *Instance broadcast receiver* hanya valid selama pemanggilan metode *OnReceive()* sehingga tidak boleh menyimpan referensi ke *instance* ini. Satu hal yang dapat dilakukan ketika *onReceive()* dipanggil adalah mewakilkannya ke komponen lain, misalnya dengan memanggil metode *startActivity()* atau *startService()* milik *context* (Juhara, 2016).

2.6.4 Struktur Aplikasi Android

Proyek aplikasi Android yang diciptakan menggunakan Android SDK tersusun dari sejumlah direktori dan beberapa file penting.

Tabel 2.2 Direktori Struktur Aplikasi Android (Juhara,2016)

| Item | Keterangan |
|---------|--|
| /src | Direktori ini berisi <i>source code</i> aplikasi. Karena aplikasi Android ditulis dalam bahasa pemrograman Java, struktur direktori didalamnya mengikuti konvensi penamaan pemrograman Java. Contoh, kelas bernama <i>com.juhara.demo.App</i> akan disimpan dalam file bernama <i>/src/comjuhara/App.java</i> . |
| /res | Direktori ini berisi semua <i>resource</i> yang digunakan aplikasi seperti file <i>layout</i> , gambar, <i>icon</i> , <i>string</i> , <i>theme</i> . |
| /assets | Direktori ini berisi file-file yang dibutuhkan oleh aplikasi saat <i>run-time</i> . Untuk mengakses isi direktori ini, aplikasi menggunakan <i>Java File Application Programming Interface</i> (API). |
| /gen | Direktori ini berisi file-file yang dihasilkan oleh compiler secara otomatis. Tidak boleh mengubah file dalam direktori ini. Ketika melakukan kompilasi, file layout berformat XML dalam <i>/res</i> akan diubah menjadi file <i>R.java</i> yang dapat ditemukan dalam direktori ini. File ini berisi nilai konstan untuk mengacu pada <i>resource</i> dalam <i>/res</i> . |
| /bin | Direktori ini berisi file biner yang dihasilkan compiler. Di dalamnya akan ditemukan file <i>.DEX</i> (<i>Dalvik executable</i>) dan file <i>.APK</i> yang nantinya dapat digunakan untuk mendistribusikan aplikasi. |

| | |
|---------------------|---|
| /libs | Direktori ini berisi pustaka yang digunakan aplikasi. |
| AndroidManifest.xml | File Manifest yang berisi informasi penting terkait aplikasi seperti nama aplikasi, deklarasi <i>Activity</i> , atau <i>BroadcastReceiver</i> yang digunakan atau izin yang diminta aplikasi. |
| Project.properties | File berisi daftar target build. |

2.6.5 Software Pendukung Android

Untuk mengembangkan sistem operasi Android, dapat menggunakan *Windows*. Semua *tool* yang dibutuhkan adalah gratis dan dapat di-*download* dari *web*. Adapun kebutuhan *software* lainnya untuk pendukung Android adalah sebagai berikut :

1. Java Development Kit (JDK)

Program Android umumnya dibuat menggunakan bahasa java. Oleh karena itu, anda akan membutuhkan *java development kit (JDK)* untuk mengembangkan aplikasi android. JDK adalah piranti lunak yang anda perlukan untuk mengembangkan aplikasi dengan bahasa java yang terdiri dari beberapa *utility* seperti *compiler*, *interpreter*, *java virtual machine*, dan lain-lain. Ingat, *Java Runtime Environment (JRE)* saja tidak cukup untuk mengembangkan aplikasi Android (Juhara, 2016).

Android SDK berjalan dengan menggunakan *resource* dan *Java SE Development Kit (JDK)*. Oleh karena itu, jika komputer belum terinstall JDK, maka harus memulai untuk mendownload JDK dari alamat www.oracle.com/technetwork/java/javase/downloads/index.html. (Komputer, 2013).

2. Android SDK (*Software Development Kit*)

Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada *platform android* menggunakan bahasa pemrograman *java*. Android SDK dapat di

download dari <http://developer.android.com/sdk/index.html>. Android SDK berisi *utility* untuk kompilasi, debugging, dan rilis (*deployment*). Ada beberapa versi yang tersedia sesuai operasi yang digunakan untuk menulis program. Jika menggunakan Windows, sebaiknya gunakan versi *installer* daripada versi dokumen terkomputerisasi zip. Jika menggunakan zip, maka harus menambah *path* Android SDK ke Windows secara manual. Instalasi dengan file zip atau tgz cukup dengan mengekstrak isi dokumen ke suatu direktori dan menambahkan nama direktori tersebut ke path sistem operasi (Juhara, 2016).

Menurut Kadir (2014) *Software Development Kit* (SDK) adalah kumpulan *software* yang berisi mengenai pustaka, *debugger* (alat pencari kesalahan program), emulator (peniru perangkat bergerak), dokumentai, kode contoh, dan panduan. Android SDK dapat diunduh secara gratis di <http://developer.android.com/sdk/>. Keberadaan emulator dapat membuat dan menguji aplikasi Android, tanpa harus mempunyai perangkat keras berbasis Android. Ya, Anda dapat membuat dan menguji aplikasi Android di komputer Anda yang tidak berbasis Android. Bahkan, Anda tidak hanya dapat menguji di Windows, tetapi juga di *platform* lain seperti Mac dan Linux (Kadir, 2014).

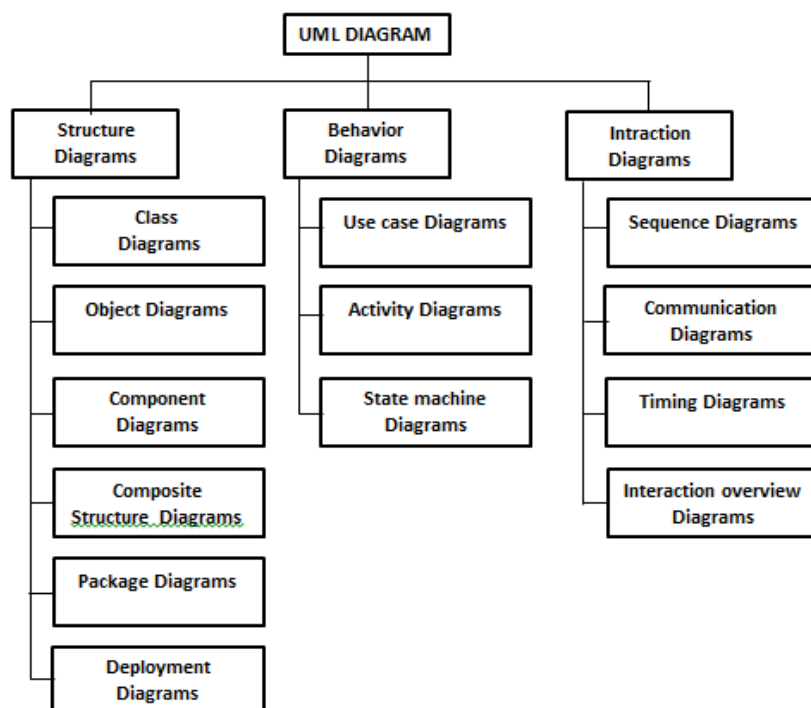
2.7 Unified Modelling Language (UML)

Menurut Marakas (2006) *The Unified Modelling Language* (UML) adalah bahasa standar industri untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan artefak sistem perangkat lunak berbasis objek. Dapat menyederhanakan proses desain perangkat lunak yang rumit, membuat cetak biru untuk konstruksi (Marakas, 2006).

UML merupakan kumpulan diagram-diagram yang sudah memiliki standar untuk pembangunan perangkat lunak berbasis objek. Perancangan dan pembangunan aplikasi perangkat lunak berbasis objek *Objek Oriented Analysis and Design* (OOAD) sedang marak digunakan saat ini. Perancangan berbasis objek dimodelkan menggunakan *Unified Modelling Language* (UML). Dengan menggunakan UML kita dapat membuat model

untuk semua jenis aplikasi piranti lunak, di mana aplikasi tersebut dapat berjalan pada piranti keras, serta ditulis dalam bentuk pemrograman (Feri, 2017).

Menurut Rosa dan Shalahuddin (2019) UML merupakan bahasa visual untuk pemodelan dalam mengenai sebuah sistem dengan menggunakan diagram dan terdiri dari teks-teks pendukung. UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori dan macam-macam diagram tersebut dapat dilihat pada gambar 2.3 dibawah ini.



Gambar 2.5 Diagram UML (Rosa dan Shalahuddin, 2019)

Keterangan gambar 2.5 adalah sebagai berikut :

1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem (AS M.Shalahudin, 2014).

2.7.1 Desain Proses

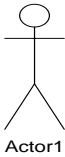
Berdasarkan uraian yang telah di jelaskan pada model UML diatas maka penulis membuat sebuah alur sistem yang akan di tampilkan dalam bentuk *Use Case Diagram*, *Activity Diagram*, *Sequence diagram* sebagai desain proses.


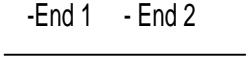
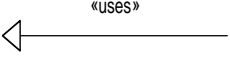
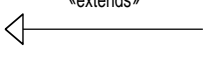
2.7.1.1 Use Case Diagram

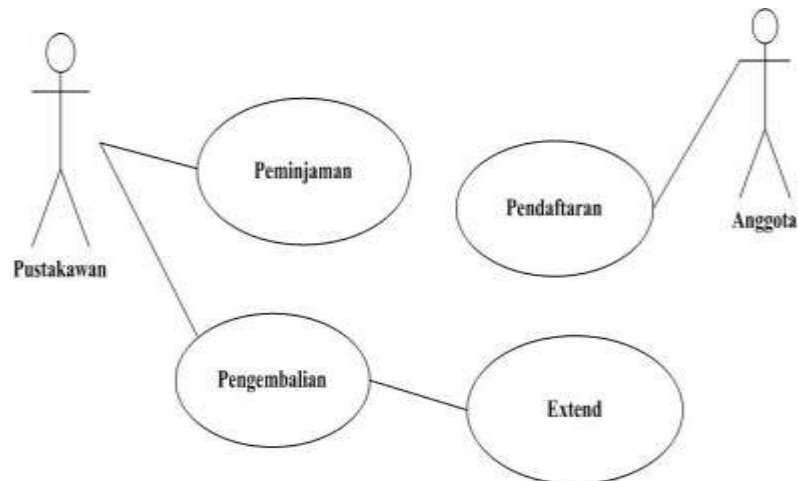
Menurut Marakas (2006) *Use case* adalah antarmuka yang membuat sistem terlihat dan tersedia untuk dunia luar di mana aktor dapat berinteraksi. Salah satu cara berpikir yang umum tentang *use case* adalah sebagai skenario. *Use case* mewakili urutan langkah-langkah, baik manual atau otomatis, yang menentukan penyelesaian tugas bisnis tunggal (Marakas, 2006).

Menurut Feri (2017) *Use case diagram* merupakan diagram yang harus dibuat pertama kali saat pemodelan perangkat lunak berorientasi objek dilakukan. Diagram *Use case* akan menggambarkan apa yang dikerjakan oleh *actor* yang disebut *actor* disini adalah pengguna aplikasi, sama seperti pembangunan perangkat lunak terstruktur saat membuat DFD, untuk menggambar *Use Case* mengacu pada proses sebelumnya, yaitu analisis kebutuhan Rekayasa Perangkat Lunak (Feri, 2017). Simbol- simbol yang digunakan dalam *Use Case diagram* adalah sebagai berikut :

Tabel 2.3 Daftar Simbol Dalam *Use Case Diagram* (Feri, 2017)

| Gambar | Nama | Keterangan |
|---|-------|---|
|  | Actor | Merupakan Pengguna dari sistem. Penamaan actor menggunakan kata benda |

| | | |
|---|----------|---|
|  | Use case | Merupakan pekerjaan yang dilakukan oleh actor. Penamaan use case dengan kata kerja |
|  | Asosiasi | Hubungan antar actor dengan use case |
|  | Include | Hubungan antara use case dengan use case, include menyatakan bahwa sebelum pekerjaan dilakukan harus mengerjakan pekerjaan lain terlebih dahulu. |
|  | Extends | Hubungan antara use case dengan use case, extends menyatakan bahwa jika pekerjaan yang dilakukan tidak sesuai atau terdapat kondisi khusus, maka lakukan pekerjaan itu. |



Gambar 2.6 Contoh Diagram *Use Case* Perpustakaan (Feri, 2017)






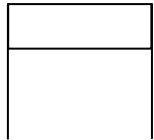
Keterangan gambar 2.6 adalah sebagai berikut :

Diagram diatas mengacu pada siapa pengguna sistem. Sebagai contoh, mengapa tidak ada hubungan antara anggota dengan peminjaman, jawabannya adalah karena anggota tidak berhubungan dengan peminjaman itu sendiri. Memasukkan data tentang peminjaman yang berupa nomor anggota, tanggal pinjam, tanggal kembali, buku apa saja yang dipinjam adalah pustakawan, maka dari itu hanya pustakawanlah yang memiliki hubungan dengan peminjaman (Feri, 2017).

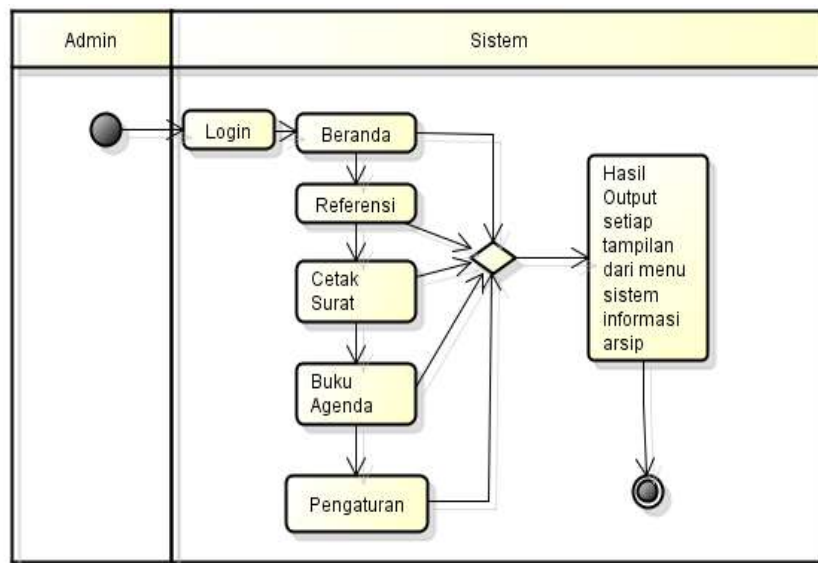
2.7.1.2 Activity Diagram

Menurut Rosa dan Shalahuddin (2019) *activity diagram* adalah menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat luna (AS M.Shalahudin, 2014). Adapun simbol-simbol yang digunakan dalam *activity diagram* adalah sebagai berikut:

Tabel 2.4 Daftar Simbol *Activity Diagram* (Paramanindo, 2014)

| Simbol | Nama | Deskripsi |
|---|-----------------|---|
|  | Status awal | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
|  | Aktivitas | Aktivitas yang dilakukan sistem biasanya diawali dengan kata kerja. |
|  | <i>Decision</i> | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
|  | <i>Join</i> | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
|  | Status akhir | Status akhir yang dilakukan sebuah sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
|  | <i>Swimlane</i> | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

Berikut ini merupakan contoh *activity diagram* :




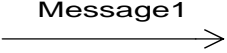
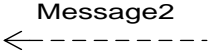
Gambar 2.7 Contoh *Activity Diagram* Kaprodi
(Irawan dan Simargolang, 2018)

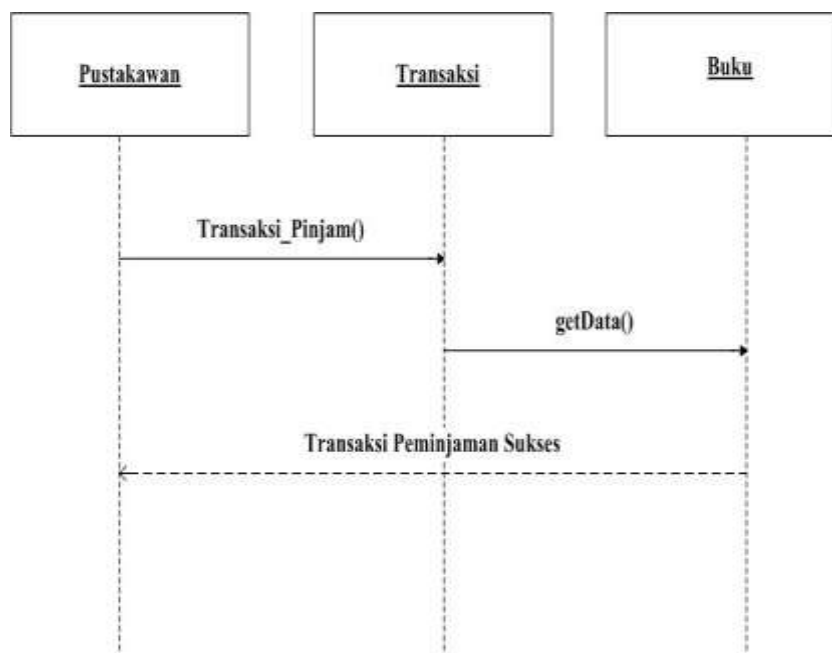
2.7.1.3 Sequence Diagram

Sequence Diagram dapat menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan atau menceritakan waktu hidup objek dan *message* yang akan dikirimkan pengguna dan diterima pengguna antar objek. Oleh karena itu agar dapat menggambar *Sequence Diagram* maka pengguna (*user*) harus ketahui objek-objekapa saja yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek tersebut. Membuat *Sequence Diagram* juga dibutuhkan untuk melihat skenario yang terdapat pada *use case* (AS M.Shalahudin, 2014).

Menurut Feri (2017) *sequence diagram* adalah diagram yang dibuat untuk mengetahui dari interaksi antarobjek. Isi dari diagram *sequence* harus sama dengan *use case* dan digram kelas. Satu *use case* tunggal akan digambarkan satu diagram *sequence*-nya.

Tabel 2.5 Daftar Simbol Dalam *Sequence Diagram* (Feri, 2017)

| Gambar | Nama | Keterangan |
|---|---------------|---|
|  | Objek / Aktor | Sebuah objek yang berasal dari kelas atau dapat dinamai dengan kelasnya saja. Aktor termasuk objek. Garis putus-putus menunjukkan garis hidup suatu objek. |
|  | Aktivasi | Menunjukkan masa hidup dari objek |
|  | Pesan | Interaksi antara satu objek dengan objek lainnya. Objek dapat mengirimkan pesan ke objek lain. Interaksi antarobjek ditunjukkan pada bagian operasi pada diagram kelas. |
|  | <i>Return</i> | Pesan kembalian dari komunikasi antarobjek. |



Gambar 2.8 Contoh *Sequence Diagram* Pustakawan Dengan Peminjam
(Feri, 2017)

Keterangan gambar 2.8 adalah sebagai berikut :

Dari diagram diatas bahwa pustakawan memanggil operasi `Transaksi_Pinjam()` dari kelas transaksi dan kelas transaksi memanggil operasi `getData()` pada kelas buku, dimana operasi `Transaksi_Pinjam()` dimiliki oleh kelas buku pada diagram kelas buku (Feri, 2017).

2.7.2 Desain Database

Berdasarkan uraian yang telah di jelaskan pada model UML maka penulis membuat sebuah desain *database* yang akan di tampilkan dalam bentuk *class diagram*.

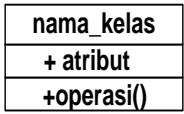
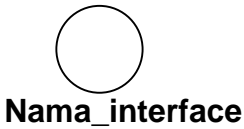


2.7.2.1 Class Diagram


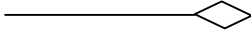
Menurut Feri (2017) *Class Diagram* adalah sebuah diagram yang menggambarkan struktur objek statis didalam sistem, *class diagram* dibuat setelah diagram *use case* dibuat terlebih dahulu. Pada pembuatan diagram ini harus menjelaskan hubungan apa saja yang terjadi antara suatu objek dengan objek lainnya sehingga terbentuklah suatu sistem aplikasi.

Pembuatan *class diagram* dibagi menjadi dua bagian, yaitu kelas itu sendiri dan relasi antarkelas. Kelas dibagi menjadi tiga bagian, yakni nama kelas, atribut kelas, serta operasi kelas (*methods*). Nama kelas adalah nama dari kelas itu sendiri, misalnya kelas mobil, kelas dosen, kelas mahasiswa. Atribut adalah data yang dimiliki oleh kelas tersebut misalnya kelas mahasiswa memiliki atribut NIM, nama, tanggal lahir, alamat jenis kelamin. Lalu operasi kelas adalah menunjukkan apa yang kelas tersebut bisa lakukan, misalnya kelas dosen dapat melakukan operasi mengajar, absensi, penelitian, dll (Feri, 2017).

Tabel 2.6 Daftar Simbol Dalam *Class Diagram*

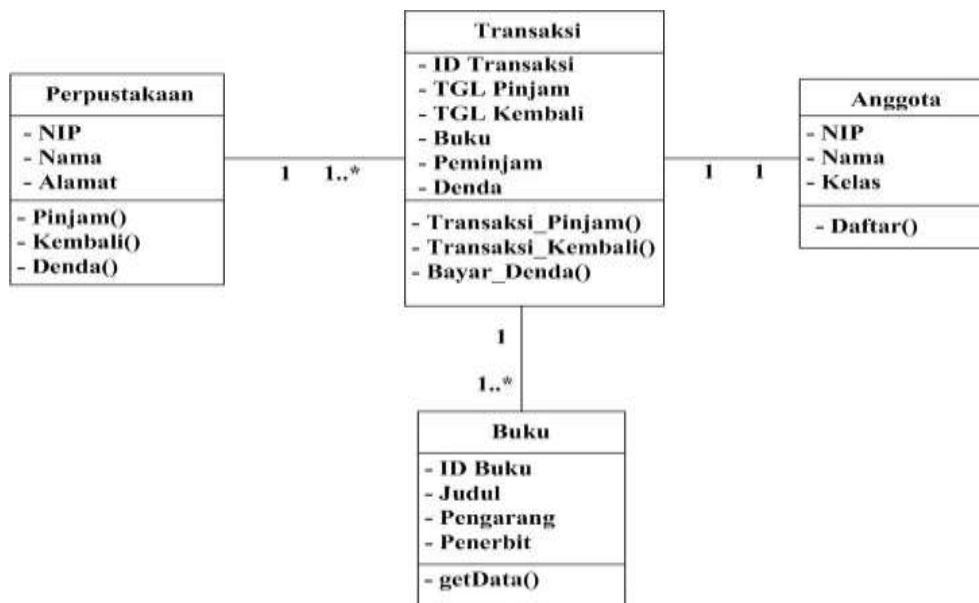
(Rosa dan Shalahuddin, 2019)

| <i>Gambar</i> | <i>Nama</i> | <i>Keterangan</i> |
|---|------------------|---|
|  | Kelas | Kelas pada struktur sistem |
|  | <i>Interface</i> | Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek |
|  | Asosiasi | Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> |
|  | Generalisasi | Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus) |

| | | |
|---|--------------------------------------|--|
|  | Kebergantungan/ <i>Dependency</i> | Relasi antarkelas dengan makna kebergantungan antarkelas |
|  | Agregasi | Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>) |

Tabel 2.7 Indikator *Class Diagram* (Feri, 2017)

| Indikator | Arti | Penjelasan |
|-----------|--|---|
| 0..1 | Kosong ke satu | Dalam hal ini objek yang berinteraksi dengan objek dari kelas lainnya hanya 1 atau tidak ada sama sekali. |
| 1 | Hanya satu | Objek yang berinteraksi dengan objek dari kelas lainnya hanya 1. |
| 0..* | Kosong ke banyak | Objek yang berinteraksi dengan objek dari kelas lainnya minimal 1 atau tidak sama sekali. |
| 1..* | Satu ke banyak | Objek yang berinteraksi dengan objek dari kelas lainnya minimal 1 atau lebih. |
| N | Banyak (n) di mana n lebih besar dari nol. | Objek yang berinteraksi dengan objek dari kelas lainnya sejumlah n. |



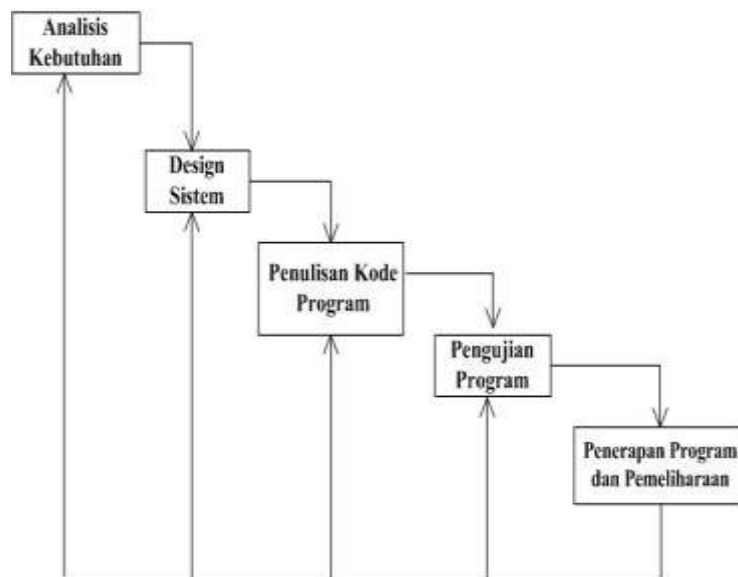
Gambar 2.9 Contoh *Class Diagram* Perpustakaan (Feri, 2017)

2.8 Metode Pengembangan Sistem

Dalam rekayasa perangkat lunak, metode pengembangan sistem merupakan suatu kerangka kerja yang digunakan untuk menstrukturkan, merencanakan, dan mengendalikan proses pengembangan suatu sistem.

2.8.1 Metode *Waterfall*

Metode *waterfall* merupakan metode yang sering digunakan oleh penganalisa sistem pada umumnya. Metode *waterfall* adalah proses pekerjaan dari suatu sistem yang dilakukan secara berurutan. Jadi jika langkah pertama belum selesai dalam pengerjaannya maka langkah kedua tidak bisa dilakukan, begitu pula langkah seterusnya.



Gambar 2.10 Model *waterfall*

Keterangan gambar 2.10 menurut Kadir (2003) adalah sebagai berikut :

1. Analisis Kebutuhan

Tahap ini merupakan analisa pada sistem. Pengumpulan data dalam tahap ini bisa dilakukan dengan cara *interview*, studi literatur. Sistem analisis ini akan menggali dan mendapatkan informasi sebanyak-banyaknya dari user sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas yang diinginkan *user*.

2. Desain Sistem

Tahap ini merupakan tahapan dimana akan dilakukannya perancangan sistem terhadap solusi permasalahan yang ada dengan menggunakan perancangann pemodelan berupa UML, DFD, maupun ERD.

3. Penulisan Kode Program

Tahap ini merupakan tahapan pembuatan *source code (coding)* yang dilakukanoleh *programmer* yang akan menerjemahkan *design* dalam bahasa yang dikenal oleh komputer.

4. Pengujian Program

Tahap ini merupakan tahap sistem yang harus diuji kemampuannya agar dapat ditemukan kekurangan pada sistem yang nantinya akan diadakan

perbaikan dan pengkajian ulang terhadap aplikasi agar menjadi lebih baik dan sempurna.

5. Penerapan Program dan Pemeliharaan

Tahap ini merupakan tahapan penggunaan sistem yang telah tersampaikan oleh pengguna dimana pengguna akan mengalami perubahan dengan adanya perangkat lunak yang baru (Paramanindo, 2014).

2.9 Basis Data

Database (basis data) merupakan kumpulan data yang saling berhubungan satu dengan lainnya yang tersimpan di perangkat keras komputer dan diperlukan suatu perangkat lunak untuk memanipulasi basis data tersebut (Kostaman & Sumaryana, 2018).

Meskipun desain *database* memiliki beberapa hal, ada keputusan tertentu yang harus dibuat mengenai *database* itu sendiri. Penting untuk diingat bahwa basis data lebih dari sekadar kumpulan *file*. Ini dimaksudkan sebagai sumber data yang koheren dan ringkas untuk berbagai pengguna dan berbagai aplikasi. Sistem saraf pusat dari suatu basis data adalah Sistem Manajemen Basis Data (Marakas, 2006).

2.9.1 PHPMysqladmin

PHPMysqladmin adalah sebuah *software* berbasis pemrograman PHP yang dipergunakan sebagai administrator MySQL melalui *browser (web)* yang digunakan untuk manajemen *database* (Kostaman & Sumaryana, 2018).

Agar aplikasi Android (*client*) dapat berkomunikasi dengan *database (server)*, maka harus dibuatkan sebuah penghubung. Penghubung ini sering disebut dengan *web service*. Karena *web service* menggunakan protokol HTTP, tentu PHP sebagai bahasa pemrograman web menjadi salah satu kekuatan dalam bahasa pemrograman yang mengelola *web service*.

Web service adalah aplikasi yang dibuat agar dapat dipanggil atau diakses oleh aplikasi lain melalui internet dengan menggunakan format

pertukaran data sebagai format pengiriman pesan, pertukaran data dengan JSON (Kasman Dharma, 2015).

1. JSON (*Java Script Object Notation*)

JSON adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari bahasa pemrograman JavaScript, Standar ECMA-262 edisi ke-3 Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data (Kasman Dharma, 2015).

2.10 Perangkat Lunak Pendukung

Dalam penelitian ini tentu saja pembuatan aplikasi harus ada perangkat lunak yang mendukung yang menjadi alat utama untuk pembuatan aplikasi. Adapun perangkat lunak yang di pakai dalam penelitian ini adalah sebagai berikut :

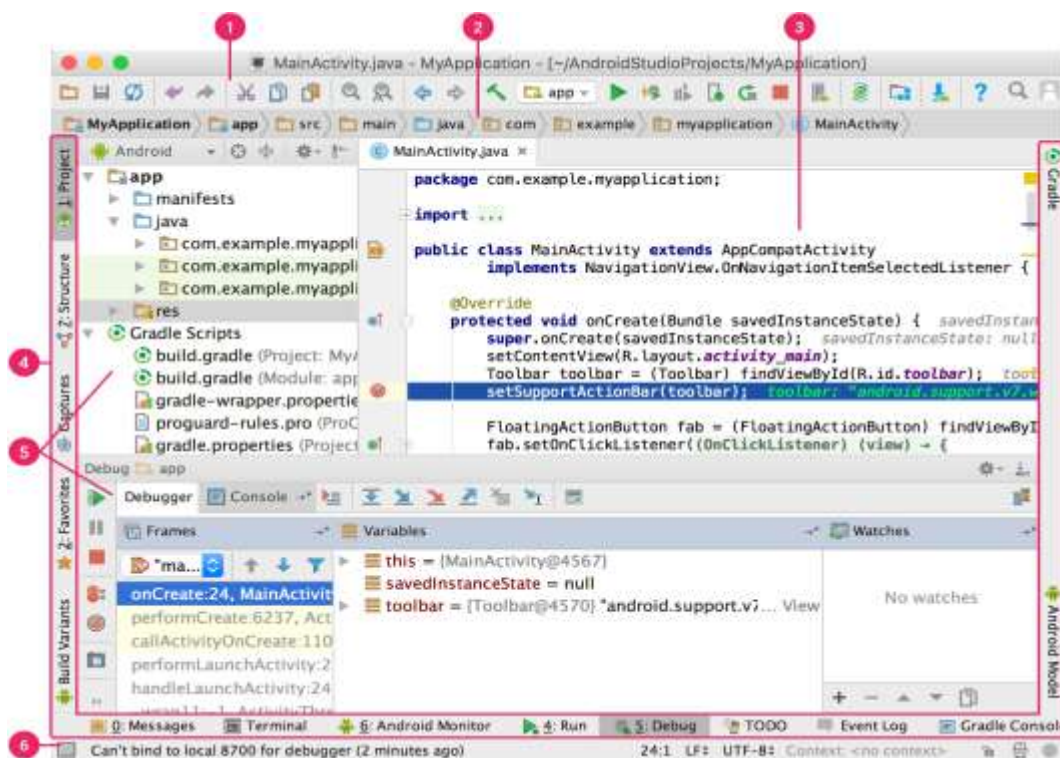
2.10.1 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu *Development* (IDE) untuk *Integrated Environment* pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Selain merupakan editor kode *IntelliJ* dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas saat membuat aplikasi Android, misalnya:

1. Sistem versi berbasis *Gradle* yang *fleksibel*
2. *Emulator* yang cepat dan kaya fitur
3. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android
4. Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru

5. Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh
6. Alat pengujian dan kerangka kerja yang ekstensif
7. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain.
8. Dukungan C+ dan NDK

Dukungan bawaan untuk Google *Cloud Platform*, mempermudah pengintegrasian Google *Cloud Messaging* dan *App Engine* (Suryana, 2018).



Gambar 2.11 Tampilan Android Studio

Berdasarkan gambar diatas, ada beberapa element yang penting dalam pembuatan aplikasi android. Berikut ini penjelasannya:

1. **Toolbar** memungkinkan Anda melakukan berbagai tindakan, termasuk menjalankan aplikasi dan meluncurkan fitur Android.
2. **Menu navigasi** membantu Anda menjelajah project dan membuka file untuk diedit. Menu ini memberikan tampilan struktur yang lebih ringkas yang terlihat di jendela Project.

3. **Jendela editor** adalah tempat Anda membuat dan memodifikasi kode. Tergantung jenis file yang ada, editor ini dapat berubah. Misalnya, saat menampilkan file tata letak, editor akan menampilkan Layout Editor.
4. **Panel jendela fitur** berada di sisi luar jendela IDE dan berisi tombol-tombol yang memungkinkan Anda memperluas atau menciutkan setiap jendela fitur.
5. **Jendela fitur** memberi Anda akses ke tugas tertentu seperti pengelolaan project, penelusuran, kontrol versi, dan banyak lagi. Anda dapat memperluas dan menciutkan jendela ini.
6. **Status bar** menampilkan status project Anda dan IDE itu sendiri, serta semua peringatan atau pesan (Suryana, 2018).

2.10.2 MySql

SQL merupakan singkatan dari *Structure Query Language*, didefinisikan sebagai suatu sintaks perintah-perintah tertentu atau bahasa program yang digunakan untuk mengelolah suatu *database*. Jadi MySql adalah *software*nya dan SQL adalah bahasa perintahnya. MySql bersifat RDBMS (*Relational Database Management System*) yang memungkinkan seorang *admin* dapat menyimpan banyak informasi kedalam *table*, dimana *table* tersebut saling berkaitan satu sama lain (Rahmat & Octaviano, 2016).

2.10.3 XAMPP

Xampp merupakan sebuah aplikasi *web server*. *Web Server* sendiri adalah sebuah aplikasi tempat menyimpan file-file maupun data-data untuk membuat *website*. Juga sering diartikan sebagai layanan data pada *web browser*. Fungsi dari *web server* sebagai penerima permintaan berupa halaman *client* dan mengirimkan kembali hasil yang diminta dalam bentuk halaman web (Nugroho et al., 2015)

2.10.4 Microsoft Visio

Microsoft Visio merupakan aplikasi yang digunakan untuk membuat diagram dan *flowchart*. Aplikasi ini menyediakan banyak fasilitas pembuatan diagram dan *flowchart* untuk menggambarkan informasi dan sistem, dari penjelasan dalam bentuk teks menjadi bentuk diagram disertai penjelasan singkat (Komputer, n.d.).

2.10.4.1 Jenis File Visio

Secara umum, sketsa diagram yang dibuat menggunakan visio dapat disimpan ke dalam 4 (empat) penggolongan *file* berdasarkan jenis dan penggunaannya, yaitu :

1. *File Drawing*

File Drawing merupakan *file visio* yang berisikan shape tertentu, yang dapat dibuat, termasuk *shape-shape* yang ada di dalamnya. *File drawing* ini menggunakan ekstensi file *.vsd.

2. *File Stencil*

File Stencil merupakan file visio yang berisikan bentuk atau *shape* tertentu, yang dapat digunakan dengan cara men-drag shape ke halaman gambar. File stencil ini secara langsung akan terbuka pada saat membuka halaman gambar baru berdasarkan template tertentu. Selain itu bisa juga dapat membuka *file stencil* secara tersendiri tanpa membuka *template* nya. *File stencil* ini menggunakan ekstensi *.vss.

3. *File Template*

File template merupakan file visio yang satu atau lebih *stencil* dengan bentuk, *style* dan *setting* yang dapat digunakan untuk merancang gambar tertentu. *File template* ini menggunakan ekstensi *.vst.

4. Format Lain

Selain ketiga penggolongan di atas, diagram yang dibuat menggunakan visio dapat disimpan ke dalam format lain, seperti file AutoCAD (*.dwg atau *.dwi). File web (*.htm atau *.html) dan file gambar (*.jpg, *.bmp, *.png, dan sebagainya) (Komputer, n.d.).

2.11 Pengujian

Pengujian adalah bagian dari proses rekayasa perangkat lunak secara terintegrasi demi memastikan kualitas dari perangkat lunak serta memenuhi kebutuhan teknis yang telah disepakati dari awal. Berikut ini jenis pengujian sistem :

1. White-box testing

White box testing merupakan jenis pengujian yang berkonsentrasi kepada *source code (coding)* dari perangkat lunak yang telah dibuat sehingga membutuhkan proses pengujian yang jauh lebih lama karena membutuhkan ketelitian dari para penguji serta kemampuan teknis pemrograman bagi parapengujinya.

2. Black-box testing

Black box testing merupakan jenis pengujian yang hanya memandang perangkat lunak dari sisi kebutuhan yang telah didefinisikan pada saat awal perancangan. Beberapa yang diujikan dalam *black box testing* adalah fungsi yang gagal, tampilan antar muka yang salah, tidak dapat mengakses basis data, dan tindakan yang tidak tepat (Wicaksono Rizky, 2017).

2.12 Studi Sejenis

Berikut ini penelitian sejenis yang digunakan peneliti sebagai bahan referensi dalam melakukan penelitian ini :

Tabel 2.8 Studi Sejenis

| No | Nama | Judul | Kelebihan | Kelemahan |
|----|---|---|--|---|
| 1. | Abdul Rohman Amat Rahmat (Jurnal Informatika Universitas Pamulang Tangerang Selatan 2016) | Aplikasi Pemesanan Tiket Bus Berbasis Web (Studi Kasus Pada PO. Harapan Jaya) | Jurnal ini hanya membahas mengenai sistem informasi pembelian tiket bus. | Perancangannya hanya membahas tentang bagaimana cara melakukan pemesanan tiket bus. |
| 2. | Haris (Jurnal Cerita | Rancang Bangun | Jurnal ini | Tidak menambah |

| | | | | |
|----|---|--|---|---|
| | Dosen dan Mahasiswa STMIK Raharja Tangerang 2016) | Aplikasi Reservasi Tiket Pesawat <i>Online</i> berbasis Android | membahas reservasi tiket menggunakan metode DAD (<i>Disciplined Agile Delivery</i>) | fitur lokasi penjualan tiket dan rute. |
| 3. | Robby Rachmatullah (Indonesian Journal on Networking and Security STMIK AUB Surakarta 2015) | Perancangan Sistem Pememsanan Tiket bus Online Berbasis Web. | Jurnal ini hanya membahas proses pembelian tiket saja | Sistem informasi penjualan tiket masih menggunakan VB 2008 |
| 4. | Nandang Kostaman, Yusuf Sumaryana (Jumantaka STMIK DCI 2018) | Aplikasi Pemesanan Tiket OTOBUS Budiman Berbasis Online | Jurnal ini membahas pembelian tiket bus dan menampilkan rute perjalanan. | Diperlukan pengembangan terhadap aplikasi karena belum dapat melakukan Back-Up. |
| 5. | Dolly Aulia, dkk (Program Studi Sistem Informasi Sekolah Tinggi Teknik Harapan Medan | Sistem Informasi Pelayanan Reservasi Tiket Pada Cv Chandra Berbasis Mobile Android | Membahas tentang pembelian tiket bus dengan cara <i>booking</i> tiket | Hanya membahas pembelian tiket bus namun sudah berbasis Android. |

BAB III

METODOLOGI PENELITIAN

Metodologi penelitian merupakan suatu proses, yaitu suatu langkah-langkah yang dilakukan secara terencana dan sistematis untuk mendapatkan jawaban terhadap pertanyaan tertentu. Metodologi penelitian ini sangat penting karena baik atau buruknya hasil penelitian tergantung dari metodologi yang digunakan.

3.1 Metodologi Penelitian *Case Study*

Penelitian ini menggunakan metode penelitian studi kasus (*case study*). Objek dari penelitian ini adalah perusahaan bus PT Chandra, Kota Pinang Baru (KPB) dan PT Sentosa. Menurut Hasibuan (2007) penelitian studi *case study* adalah penelitian yang memfokuskan perhatian pada suatu kasus tertentu dengan menggunakan individu atau kelompok sebagai bahan studinya. Studi kasus juga menggali fenomena tertentu (kasus) dalam suatu waktu kegiatan serta mengumpulkan informasi yang rinci dengan menggunakan berbagai prosedur pengumpulan data selama periode tertentu.

3.1.1 Tempat dan Waktu Penelitian

Dalam pelaksanaan skripsi ini penulis mengambil tempat penelitian pada Perusahaan PT Chandra, KPB (Kota Pinang Baru), Sentosa yang bergerak dibidang transportasi yaitu pengangkutan bus umum beralamat di Jln Sisingamangaraja KM 7,5 Kec. Medan Amplas. Pada penelitian ini dibutuhkan jadwal serta tahapan penelitian untuk mencapai batas waktu dalam pembuatan sistem. Penelitian ini dilaksanakan pada bulan Mei 2019 hingga September 2019.

3.1.2 Kebutuhan Sistem

Kebutuhan sistem adalah syarat yang digunakan untuk merancang dan mengoperasikan sebuah aplikasi maupun sistem agar dapat digunakan secara baik dan benar. Kebutuhan sistem dapat berupa perangkat yang digunakan oleh penulis dalam mengerjakan skripsi ini mulai dari mendesain sampai tahap pemrograman penulis menggunakan perlengkapan komputer sebagai *server*. Secara lebih spesifik perlengkapan komputer beserta perangkat pendukung yang digunakan yaitu :

1. Perangkat Keras (*Hardware*)

Perangkat keras (*hardware*) adalah komponen fisik yang digunakan untuk membuat aplikasi. Adapun kebutuhan perangkat keras (*hardware*) yang digunakan dalam proses perancangan pembuatan aplikasi pada penelitian ini yaitu :

a. Laptop

Deskripsi laptop yang digunakan oleh penulis untuk perancangan aplikasi pada penelitian ini yaitu :

- *Procesor* : *Intel ® Core™ i3 Duo 2.00 GHz.*
- *Memory* : *DDR3 10 GB*
- *Harddisk* : *500 GB*
- *VGA* : *intel ® HD Graphics 3000*
- *Layar* : *14 Inchi LED HD*
- *Operating Sytem* : *Windows 7 ultimate 64-bit*

b. *Handphone*

Deskripsi *handphone* yang digunakan oleh penulis untuk mengoperasikan aplikasi pada penelitian ini adalah:

- *Layar* : *5 inci (960 x 540)*
- *Jenis Layar* : *Thin-film Transistor (TFT)*
- *Sistem Operasi* : *Android 6.1 Marshmallow*
- *Memory* : *Internal 8 GB dan RAM (Random Access Memory) 1.5 GB.*
- *Bobot* : *160 Gram*

2. Perangkat Lunak (*software*)

Perangkat lunak (*software*) adalah komponen non fisik yang digunakan untuk membuat aplikasi. Adapun kebutuhan perangkat lunak (*software*) untuk pembuatan aplikasi ini adalah sebagai berikut :

- Sistem Operasi *Windows 7 Ultimate*
- Editor Android Studio
- Java Development Kit (JDK)
- Notepad ++
- Chrome
- M.S Visio 2007
- XAMPP

3.1.3 Teknik Pengolahan Data dan Analisis Data

Permasalahan merupakan latar belakang untuk membangun sebuah sistem yang baru, sering kali rekayasa perangkat lunak kesulitan dalam menentukan apa saja yang harus dilakukan oleh sistem. Deskripsi layanan merupakan hal yang paling penting bagi sistem untuk kebutuhan proses menemukan, menganalisis, mengolah, mendokumentasikan dan memeriksa layanan yang disebut dengan analisis kebutuhan data. Analisis data diperlukan agar sistem dapat mencapai tujuan yang diharapkan.

Analisis data yang telah dibuat diharapkan mampu berjalan sesuai dengan keinginan sebagai berikut :

1. Aplikasi mampu menampilkan lokasi pengguna berada dan lokasi loket yang berada dekat dengan pengguna.
2. Aplikasi dapat menampilkan detail informasi tentang jadwal keberangkatan, harga tiket, gambar, dan fasilitas apa saja yang didapat dibus.
3. Aplikasi mampu menampilkan rute yang akan dituju berdasarkan lokasi pengguna.
4. Pengguna dapat melihat bus apakah sudah berangkat atau belum dengan memanfaatkan GPS.

3.2 Cara Kerja

3.2.1 Metode Pengumpulan Data

Didalam penulisan skripsi ini dilakukan metode pengumpulan data yang dibutuhkan terkait dengan penelitian ini. Berikut ini metode pengumpulan data yang digunakan dalam penelitian:

3.2.1.1 Observasi

Metode ini dilakukan dengan pengamatan secara langsung pada objek penelitian untuk mengetahui bagaimana proses pembelian tiket bus yang dilakukan oleh *client* setiap harinya, dimana proses pembelian tiket bus masih secara manual yaitu dengan mendatangi loket untuk mendapatkan sebuah tiket perjalanan yang diinginkan oleh calon penumpang atau biasa disebut dengan belum terkomputerisasi.

3.2.1.2 Wawancara

Metode ini dilakukan dengan cara bertanya langsung kepada salah satu pegawai dibagian pelayanan dan pemesanan mengenai proses pembelian tiket yang ada pada PT Chandra, Kota Pinang Baru dan Sentosa. Dengan melakukan wawancara penulis dapat mengetahui permasalahan yang ada sehingga dapat merumuskan kebutuhan untuk membangun sistem pembelian tiket bus secara *online*.

Wawancara ini dilakukan pada :

Hari : Sabtu
Tanggal : 13 Juni 2019
User : Bapak Koko, Bapak Izzat dan bapak Wiyanto
Jabatan : Bos dari PT Chandra dan para petugas pelayanan tiket dari KPB dan Sentosa
Hasil : Mengetahui alur proses pembelian tiket bus yang dilakukan setiap harinya oleh *client*.

3.2.1.3 Analisis Dokumen

Penelitian ini menggunakan analisis dokumen untuk meninjau dokumentasi yang disediakan, dokumen yang dimiliki akan digunakan pada Bab IV pada tahap perencanaan. Dokumen yang didapatkan untuk dianalisis adalah dokumen profil jadwal keberangkatan visi dan misi.

3.2.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah Model *Waterfall*. Alasan penulis menggunakan cara *waterfall* dikarenakan metode ini mempunyai tahapan-tahapan yang jelas, nyata dan praktis. Setiap tahap harus diselesaikan terlebih dahulu untuk menghindari terjadinya pengulangan dalam tahapan sehingga pengembangan sistem yang dilakukan dapat memperoleh hasil yang diinginkan. Adapun tahapannya metode *waterfall* sebagai berikut :

1. Analisis Kebutuhan

Pada tahap ini penulis melakukan pencarian dan pengumpulan data-data dan informasi secara lengkap dan akurat diantaranya seperti:

Profil dari perusahaan PT. Chandra, KPB dan Sentosa yaitu data yang mengenai tentang profil, Visi dan Misi, Jadwal keberangkatan, harga tiket bus dan data mengenai sistem yang sedang berjalan seperti informasi yang digunakan oleh perusahaan tersebut.

2. Design Sistem

Pada tahap ini penulis akan melakukan perancangan sistem usulan yang dikembangkan berdasarkan tahap analisis kebutuhan agar sistem yang diusulkan dapat berjalan dengan baik dan dapat menjadi solusi permasalahan yang ada. Adapun tahap perancangan yang dilakukan adalah sebagai berikut:

a. Desain Proses

Pada tahap ini penulis mengidentifikasi aktor-aktor yang terlibat dalam sistem pembelian tiket bus dengan tugas antar aktor berdasarkan informasi yang telah didapatkan penulis. *Tools* yang digunakan adalah *Unified Modelling Language* (UML) dengan

software Microsoft Visio 2007. Berikut ini adalah diagram yang digunakan :

- *Membuat Use Case Diagram*
 Pada tahap ini penulis menggambarkan keterkaitan antara sistem usulan dengan aktor (*admin, agent, client*) yaitu apa saja yang akan dilakukan oleh *client* terhadap sistem melalui *use case*.
 - *Membuat Activity Diagram*
 Pada tahap ini penulis menggambarkan aliran kerja dari satu aktivitas ke aktivitas sistem lainnya.
 - *Membuat Sequence diagram*
 Pada tahap ini penulis akan menggambarkan interaksi objek yang disusun dalam suatu waktu dan hubungan timbal balik terhadap aplikasi pembelian tiket bus.
- b. *Desain Database*
- *Membuat class diagram*
 Pada tahap ini penulis menggambarkan truktur sistem dan pendefenisian kelas-kelas yang akan dibuat untuk membangun sistem
 - *Membuat Entity Relationship Diagram (ERD)*
 Tahap ini penulis merancang data yang akan digunakan dalam sistem dengan mengelompokkan elemen data menjadi tabel yang menunjukkan entitas sekaligus relasinya.
 - *Relasi Antar Tabel*
 Penulis membuat relasi antar tabel berdasarkan *class diagram* yang telah dipetakan kedalam himpunan relasi.
- c. *Desain Interface*
- *Rancangan Struktur Menu*
 Penulis merancang struktur menu dari admin, agen, supir, dan *client*.
 - *Rancangan Interface*
 Penulis merancang *interface* sistem yang akan dibuat untuk menggambarkan tampilan sistem.

3. Penulisan Kode Program

Pada tahap ini dilakukan perancangan tampilan sistem berdasarkan proses, objek, serta tampilan yang telah penulis rancang pada tahap *design* dan sesuai dengan data yang dibutuhkan sistem yang telah penulis dapat pada tahap analisis kebutuhan. Pengkodean aplikasi menggunakan bahasa pemrograman *java* dan basis data MySQL.

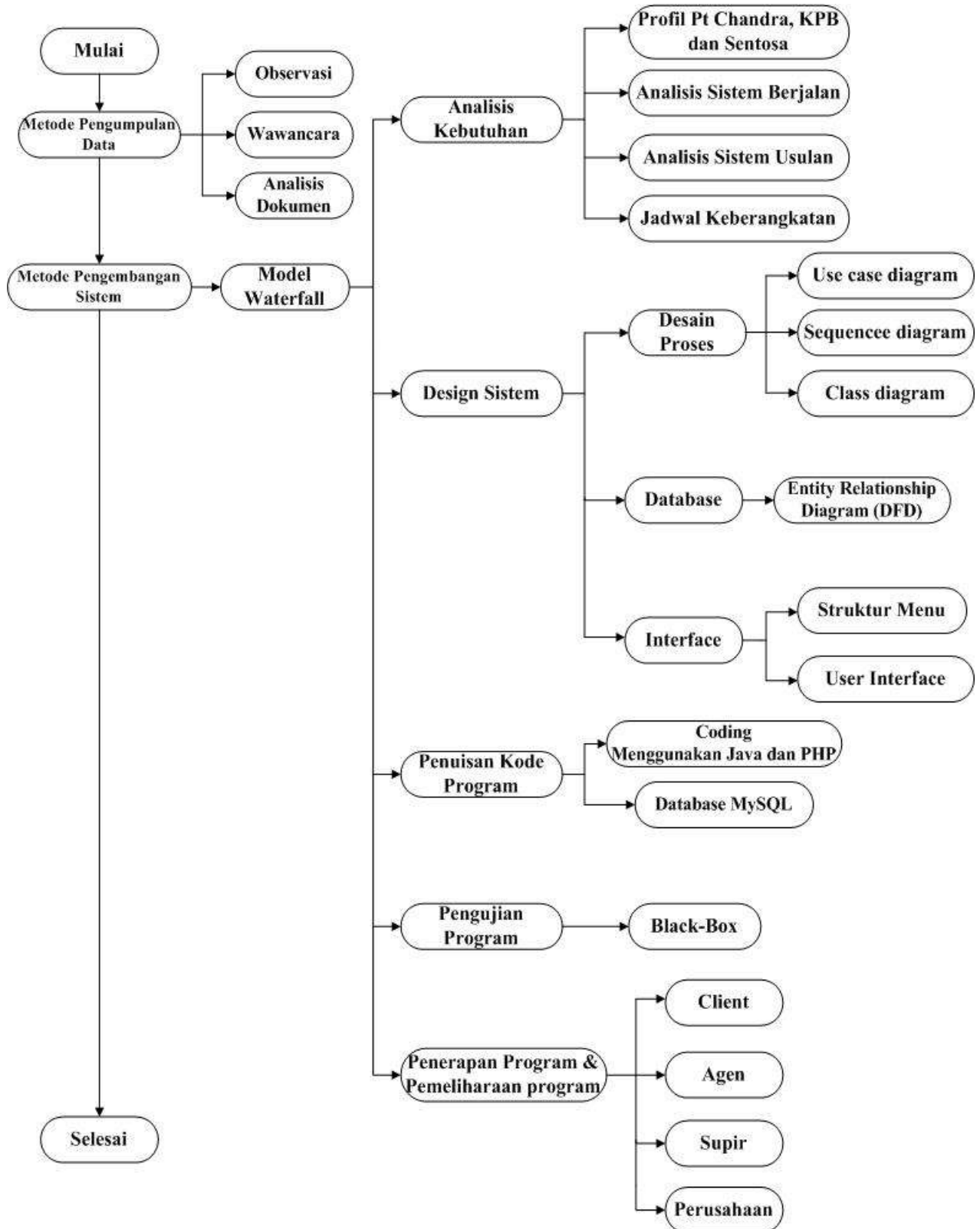
4. Pengujian Program

Pada tahap ini penulis melakukan pengujian sistem yang telah dibangun pada tahap sebelumnya yaitu penulisan kode program dengan menggunakan metode *black-box*, dengan melakukan login pada sistem dan dapat melihat apakah masing-masing dari fungsi dalam sistem beroperasi dengan baik sesuai analisis kebutuhan dan desain atau tidak.

5. Penerapan program dan pemeliharaan sistem

Setelah tahap testing dan dilakukan revisi sesuai kebutuhan, aplikasi sudah dapat disebarkan kepada *client* untuk digunakan sesuai hak akses masing-masing. Aplikasi yang sudah dioperasikan sesuai dengan kebutuhan perusahaan secara efektif dan efisien perlu dilakukan perawatan sesuai dengan yang diharapkan.

3.2.3 Kerangka Berpikir



Gambar 3.1 Kerangka Berpikir

BAB IV

HASIL DAN PEMBAHASAN

4.1 Analisis Kebutuhan

Analisis kebutuhan adalah sebuah proses untuk mendapatkan informasi tentang data yang diinginkan mengenai profil perusahaan, jadwal keberangkatan bus, Visi dan Misi, analisis sistem berjalan, analisis sistem usulan.

4.1.1 Profil Perusahaan

Profil adalah keadaan atau gambaran yang memberikan laporan untuk tujuan tertentu. Dalam penulisan ini peneliti mengambil 3 (tiga) profil perusahaan yang akan dijabarkan diantaranya adalah sebagai berikut :

1. PT Chandra

PT Chandra yang beramat di Jl Sisingamangaraja K.M 7.5 Medan – Amplas adalah sebuah Perusahaan Otobus (PO) yang memakai nama dari Chandra Adi Darma Raya yang berkantor pusat di Kota Pinang, Labuhanbatu Selatan, Sumatera Utara. Saat ini PT Chandra mempunyai sekitar 20 unit bus yang mencakup *class* armada Ekonomi, AC, AC Toilet, dan *Royal Class*. Adapun bus CV Chandra melayani trayek Medan – Kota Pinang – Bagan Batu – Ajamu – Cikampak – Teluk Panji – Riau dan Trip yang dilalui adalah sebagai berikut :

- a. Medan – Kota Pinang AC
- b. Medan – Bagan Batu AC Toilet / AC / Ekonomi
- c. Medan – Kubu Ekonomi
- d. Medan – Ajamu Ekonomi / AC
- e. Medan – Cikampak AC
- f. Medan – Pekan Baru *Royal Class*
- g. Medan – SP Damar Ekonomi
- h. Medan – Huta Godang AC

Adapun jadwal keberangkatan bus pada PT Chandra Medan yaitu sebagai berikut:

Tabel 4.1 Jadwal Keberangkatan PT. Chandra

| TRIP | JAM | TUJUAN | TIPE BUS | NO/NAMA BUS |
|-------------|--------------|---------------|-----------------|--------------------|
| 1 | 08.00 WIB | Kota Pinang | AC | 09 |
| 2 | 08.45 WIB | Bagan Batu | AC TOILET | 4009 |
| 3 | 16.30 WIB | Kubu | EKONOMI | 125 |
| 4 | 19.15 WIB | Ajamu | EKONOMI | Aquarius |
| 5 | 19.45 WIB | Ajamu | AC | 109 |
| 6 | 20.30 WIB | Bagan Batu | EKONOMI | Pisces |
| 7 | 20.45 WIB | Kota Pinang | AC | 909 |
| 8 | 21.00 WIB | Teluk Panji | AC | 1009 |
| 9 | 21.15 WIB | Bagan Batu | AC | 3209 |
| 10 | 21.30 WIB | Cikampak | AC | 009 |
| 11 | 21.45 WIB | Bagan Batu | ROYAL CLASS | 3909 |
| 12 | 22.00 WIB | Bagan Batu | AC TOILET | 2709 |
| 13 | 22.15 WIB | Kota Pinang | AC | 3609 |

| | | | | |
|----|--------------|------------------------|---------|-------|
| 14 | 20.00 WIB | Huta Godang | AC | 2809 |
| 15 | 22.15 WIB | Sp. Damar Pr. Jatim | EKONOMI | Aries |

2. Kota Pinang Baru (KPB)

Kota Pinang Baru yang beralamat di jalan Sisingamangaraja K.M 6,5 Medan - Amplas adalah perusahaan yang bergerak di bidang transportasi yang mengkhususkan pada jasa angkut penumpang darat (Bus). Trayek yang dilalui adalah jalur yang termasuk dalam Angkutan Kota Dalam Propinsi (AKDP) khususnya Propinsi Sumatera Utara antara lain Kab. Batubara, Asahan dan Kab. Labuhan Batu Selatan. Kota Pinang Baru berkantor pusat di Kota Pinang, saat ini bus di KPB ada sekitar 15 unit yang mencakup beberapa armada diantaranya Ekonomi, AC Ekonomi, AC Toilet. Adapun trip-trip dari KPB adalah :

- a. Medan – Simpang Kanan Ekonomi
- b. Medan – Teluk Panji AC Ekonomi / Toilet
- c. Medan – Huta Godang Ekonomi
- d. Medan – Kota Pinang AC Ekonomi / Toilet
- e. Medan – Kota Pinang Ekonomi
- f. Medan – Bagan Batu – Dusun Bakti AC Toilet
- g. Medan – Bagan Batu – Harapan Jaya AC Ekonomi / Toilet.

Adapun jadwal keberangkatan bus KPB Medan yaitu sebagai berikut :

Tabel 4.2 Jadwal Keberangkatan Bus KPB

| TRIP | WAKTU | TIPE BUS | TUJUAN |
|------|-----------|-----------|----------------|
| 1 | 08.00 WIB | AC | Kota Pinang |
| 2 | 19.30 WIB | Ekonomi | Teluk Panji |
| 3 | 20.00 WIB | AC Toilet | Teluk Panji |
| 4 | 20.30 WIB | Ekonomi | SP. 4 SIDODADI |

| | | | |
|---|-----------|------------|---------------------------|
| 5 | 20.45 WIB | AC Toilet | Huta Godang |
| 6 | 21.00 WIB | AC Toilet | Bagan Batu / Harapan Jaya |
| 7 | 21.30 WIB | AC Ekonomi | Kota Pinang |
| 8 | 22.00 WIB | AC Toilet | Kota Pinang |

3. Sentosa

Cv Indah Transport atau yang disingkat menjadi CV intra yang beralamat di Jl Sisingamangaraja K.M 7,5 Medan – Amplas adalah perusahaan otobus yang berasal dari kota Pematangsiantar dan bergerak dibidang jasa layanan angkutan transportasi darat untuk penumpang. CV intra didirikan pada tanggal 1 juli 1995 oleh bapak Kasianus Butarbutar. Pada awal berdirinya CV Intra hanya memiliki 7 armada saja untuk melayani penumpang untuk Siantar - Medan, namun dengan berjalannya waktu CV Intra menambah armada untuk antar provinsi yaitu Riau - Jambi. Pada tanggal 25 Juli 2013 CV Intra berubah menjadi PT Sentosa Transport. Perubahan ini terjadi karena didasari oleh perusahaan yang sudah semakin besar. Trip yang dilalui jalur PT Sentosa sebagai berikut :

- a. Medan - Pekan baru Ekonomi / Toilet.
- b. Medan - Duri Ekonomi / Toilet
- c. Medan - Rantauprapat Ekonomi / Toilet
- d. Medan – Jambi Ekonomi / Toilet.

Adapun jadwal keberangkatan PT Sentosa Medan yaitu sebagai berikut :

Tabel 4.3 Jadwal Keberangkatan PT Sentosa

| TRIP | TUJUAN | TIPE BUS |
|------|--------------|------------------|
| 1 | Aek Kenopan | Ekonomi / Toilet |
| 2 | Rantauprapat | Ekonomi / Toilet |
| 3 | Kota Pinang | Ekonomi / Toilet |
| 4 | Bagan Batu | Ekonomi / Toilet |

| | | |
|----|----------------|------------------|
| 5 | Balam | Ekonomi / Toilet |
| 6 | Duri | Ekonomi / Toilet |
| 7 | Kandis | Ekonomi / Toilet |
| 8 | Perawang | Ekonomi / Toilet |
| 9 | L. Dalam | Ekonomi / Toilet |
| 10 | Pekan Baru | Ekonomi / Toilet |
| 11 | SIMP. Rambutan | Ekonomi / Toilet |
| 12 | Jambi | Ekonomi / Toilet |

4.1.1.1 Visi dan Misi

Visi adalah gambaran dan tujuan suatu perusahaan di masa depan sedangkan Misi adalah cara untuk mencapai tujuan itu. Adapun Visi dan Misi dari Perusahaan yang terkait dari penelitian ini adalah :

1. PT Chandra

Visi

“Menjadi yang terbaik dan terpercaya, didalam memberikan pelayanan kebutuhan transportasi bus dengan mengutamakan keselamatan dan kenyamanan dalam perjalanan secara optimal”.

Misi

“Menyediakan layanan transportasi dan menjadi sahabat penumpang dalam perjalanan dengan memberikan layanan prima”.

2. Kota Pinang Baru (KPB)

Visi

“Menjadi perusahaan penyedia layanan transportasi yang terdepan”.

Misi

“Memberikan pengalaman menikmati perjalanan yang nyaman, mewah dan aman kepada para penumpang dengan harga yang kompetitif”.

3. Sentosa

Visi

“Menjadi perusahaan otobus terkemuka di kawasan Sumatera Utara dengan pelayanan terbaik demi kenyamanan dan keselamatan penumpang”.

Misi

1. Memberikan pelayanan terbaik demi kenyamanan penumpang.
2. Mendukung pemberdayaan transportasi wisata Danau Toba.
3. Memberikan pengalaman menikmati pejalan yang nyaman dan aman kepada penumpang.
4. Mematuhi seluruh peraturan per-Undang-undangan lalu lintas yang berlaku.

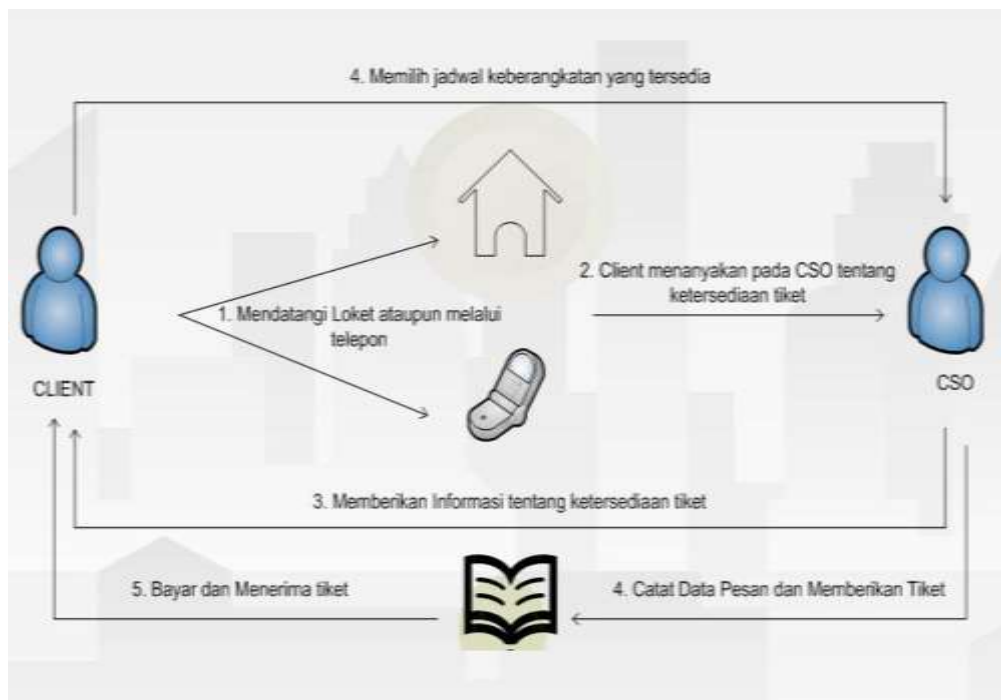
4.1.2 Analisis Sistem Berjalan

Untuk dapat mengetahui adanya permasalahan dalam suatu sistem, maka perlu dilakukan analisa pada sistem yang berjalan dengan tujuan untuk mengetahui secara lebih detail proses dari sistem tersebut. Analisis sistem adalah tahap melakukan identifikasi terhadap kebutuhan sistem secara keseluruhan, sehingga sistem yang dibangun sesuai dengan keinginan dari *end user* yang akan menggunakan sistem dalam menunjang aktivitasnya. Berdasarkan pengamatan dan observasi yang telah dilakukan penulis dari ke 3 (tiga) perusahaan OTObus telah didapati alur kerja dari pembelian tiket bus yang dilakukan oleh *client* :

1. *Client*/Pelanggan/konsumen dapat langsung datang ke loket PT Chandra, KPB, Sentosa atau bisa juga melalui telepon untuk melakukan pemesanan tiket.
2. *Client*/Pelanggan menanyakan informasi tentang jadwal keberangkatan dan ketersediaan tempat duduk.
3. Bagian CSO (*customer system officer*) memberikan informasi yang ditanyakan oleh *Client*.

4. Jika menemukan jadwal dan ketersediaan tempat yang diinginkan, *Client* akan menyatakan pada bagian CSO untuk melakukan reservasi.
5. Bagian CSO meminta data diri pelanggan dan data lainnya yaitu nama pemesan dan nomor *handphone* yang dapat dihubungi, tanggal keberangkatan, jam keberangkatan.
6. Bagian CSO melakukan pencatatan pada dokumen data pemesan.
7. Pembayaran dapat dilakukan secara langsung ketika melakukan pemesanan dengan cara datang secara langsung ke loket, bisa juga dilakukan paling lama 30 menit sebelum keberangkatan, jika 30 menit sebelum keberangkatan konsumen belum melakukan pembayaran maka reservasi tiket akan dibatalkan.
8. Apabila pelanggan telah melakukan pembayaran maka bagian CSO membuat bukti pembayaran.
9. Pelanggan menerima bukti pembayaran yang juga berguna sebagai tiket perjalanan.

4.1.2.1 Gambaran Analisis Sistem Berjalan



Gambar 4.1 Sistem Berjalan Pembelian Tiket Bus

4.1.2.2 Narasi Sistem Berjalan

1. Calon pembeli harus mendatangi loket atau menelpon untuk mendapatkan informasi pemesanan tiket.
2. Lalu menanyakan ketersediaan tiket kepada CS apakah ada jadwal dan tiket yang tersedia untuk keberangkatan yang diinginkan.
3. CS memberikan informasi tentang tiket yang tersedia seperti jam keberangkatan bus, harga tiket bus, dan tipe bus.
4. *Client* memilih jadwal yang telah tersedia.
5. Lalu CS mencatat dan memberikan sebuah tiket yang berisikan nama jam keberangkatan, harga tiket bus dan tipe bus serta nomor bus.
6. *Client* lalu membayar dan menerima tiket.

4.1.2.3 Identifikasi Masalah Sistem Berjalan

Dari sistem berjalan tersebut, penulis mengidentifikasi beberapa masalah yang terjadi pada sistem tersebut diantaranya :

1. Sistem yang berjalan masih sangat manual, dalam hal pembelian tiket bus *client* harus mendatangi loket hanya untuk mendapatkan sebuah informasi tiket perjalanan dan mengetahui jadwal keberangkatan bus, pencatatan data *client* yang akan membeli tiket bus masih menggunakan buku besar / kertas formulir.
2. Proses penyimpanan data pelanggan yang telah memesan tiket masih menggunakan arsip. Sehingga selalu membutuhkan tambahan tempat untuk penyimpanan yang di catat di buku besar.
3. Pengumpulan data *client* dari pemeriksaan dalam bentuk kertas rentan akan kerusakan dan kehilangan.

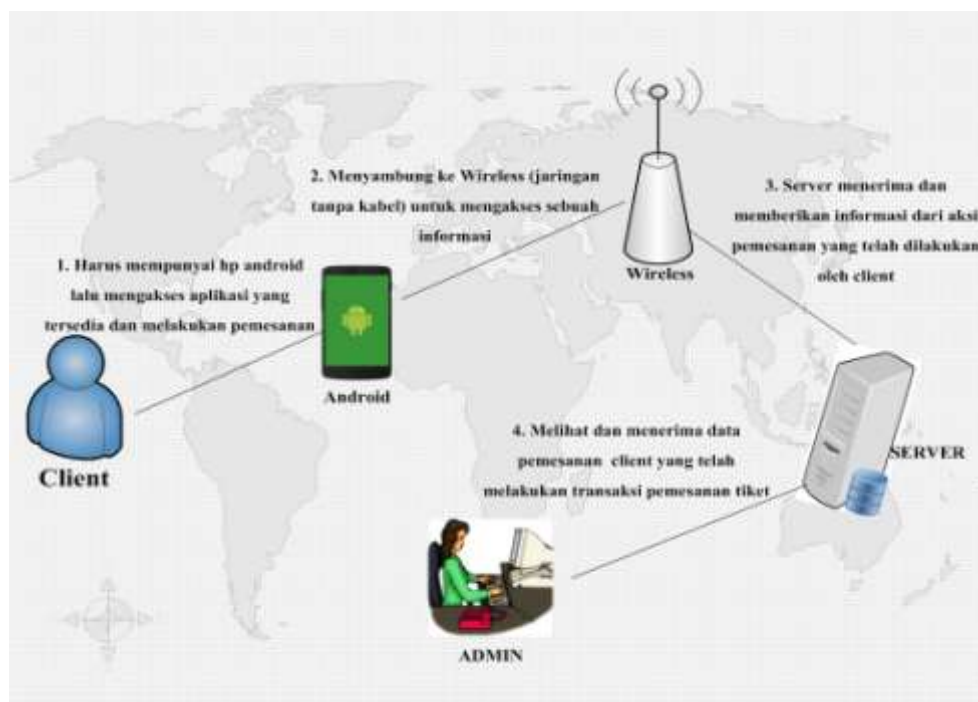
4.1.3 Analisis Sistem Usulan

Penulis mengusulkan untuk membangun sebuah sistem penunjang keputusan. Sistem yang akan dibangun diharapkan dapat membantu dalam pembelian tiket bus dengan layanan lokasi yang diinginkan oleh *client* tanpa harus

memikirkan jarak dan waktu. Adapun gambaran sistem yang akan dibangun adalah sebagai berikut :

1. Calon penumpang yang jarak tempuhnya jauh maupun dekat dari lokasi loket tidak harus mendatangi loket untuk memesan tiket perjalanan karena sudah *online* dalam hal pembelian tiket bus.
2. *Client* bisa menentukan lokasi naik sesuai keinginan apabila lokasi loket bus sangat jauh dari lokasi *Client* untuk naik dari titik awal bus berangkat.
3. *Client* bisa melihat jadwal yang tersedia melalui aplikasi yang akan dibangun.
4. Bagi *Client* yang memesan tiket dengan lokasi naik yang diinginkan maka *Client* tersebut bisa melihat posisi bus apabila bus sudah berangkat tanpa khawatir ketinggalan bus.
5. Pembayaran bisa dilakukan melalui transfer bank dan cod.

4.1.3.1 Gambaran Analisis Sistem Usulan



Gambar 4.2 Sistem Usulan Pembelian Tiket Bus

4.1.3.2 Narasi Sistem Usulan

1. *Client*

- a. *Client* membuka aplikasi pemesanan tiket bus melalui *Handphone* Android.
- b. Melakukan verifikasi *username* dan *password (login)* terlebih dahulu.
- c. Masuk ke menu pemesanan tiket
- d. Kemudian melakukan pemesanan dengan memilih tujuan keberangkatan yang tersedia, lalu menentukan tanggal keberangkatan, dan lokasi naik yang diinginkan.
- e. Lalu mengisi data pribadi yang terdiri dari nama, alamat, nomor telepon dan tempat keberangkatan
- f. Melakukan konfirmasi pembayaran tiket.

2. Admin

- a. Menambah dan meng-*update* data mengenai jadwal keberangkatan, harga tiket bus.
- b. Melihat data yang sudah melakukan transaksi baik itu yang belum bayar atau sudah bayar dalam hal pemesanan tiket bus.

4.1.4 Analisis Keputusan

Dari tahapan analisis yang telah dilakukan, maka selanjutnya kita memasuki tahap analisis keputusan untuk menentukan komponen dari sistem usulan yang dibutuhkan. Berikut ini sistem usulan yang dirancang:

1. Data Master

Proses pemasukan data yang berhubungan dengan pembelian tiket bus yaitu data loket, data agen, data bus, data supir, data tujuan, dan daftar perjalanan.

2. Menentukan Hak Akses (*user level*)

Setiap pengguna diberikan hak akses sesuai dengan bagian dan tugasnya masing-masing. Setelah mengetahui komponen-komponen sistem yang diusulkan, kemudian menentukan jenis perangkat sistem (*tools*) yang ada dalam

merancang sistem usulan sehingga dapat menghasilkan suatu arsitektur yang sesuai dengan kebutuhan *client*. Sistem yang akan dirancang ini menggunakan diagram *Unified Modelling Language* (UML) untuk mendeskripsikan sistem pembelian tiket bus.

4.2 Desain Sistem

Setelah melakukan analisis, tahap selanjutnya desain sistem. Desain sistem dibuat untuk mengetahui gambaran sistem. Dalam tahap ini, dilakukan desain proses, desain *database* dan *interface* sistem.

4.2.1 Desain Proses

Pada tahap ini dijelaskan proses perancangan sistem yang diusulkan menggunakan *Unified Modelling Language* (UML) yakni *use case* untuk mendeskripsikan hubungan antara *actor* dengan sistem yang akan dibangun, *activity diagram* untuk menjelaskan aliran aktivitas dari sistem, *sequence diagram* untuk menggambarkan antara objek melalui pesan.

4.2.1.1 Use Case Diagram Aplikasi Online Pembelian Tiket Bus Dengan Layanan *Location Based Service*

Use Case diagram merupakan diagram yang dibuat untuk mendeskripsikan hubungan aktor yang terlibat dengan sistem yang akan dibuat.

1. Identifikasi *Actor*

Tabel 4.4 Identifikasi *Actor*

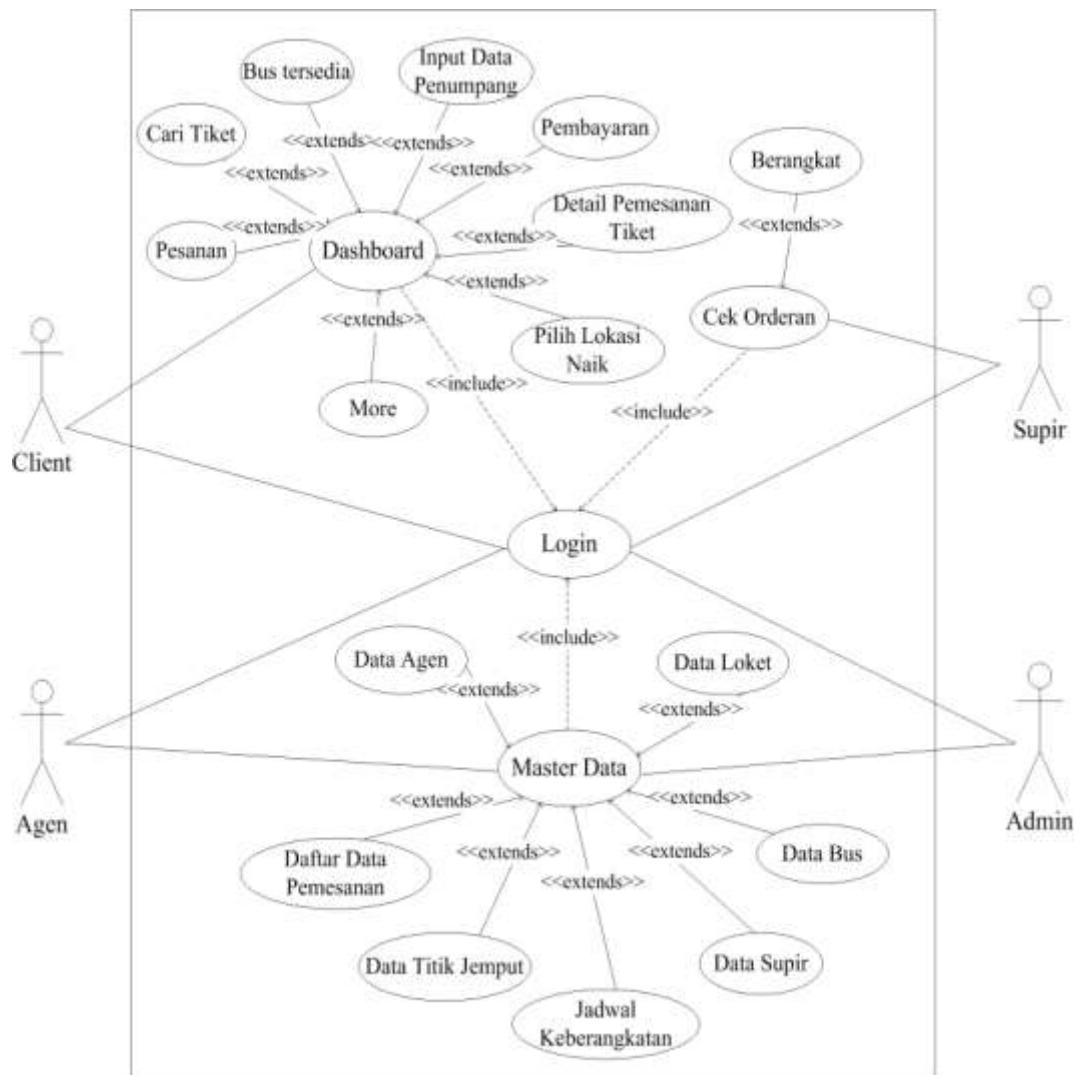
| No | <i>Actor</i> | <i>Description</i> |
|----|---------------|---|
| 1. | Admin | Orang yang dapat mengelola hak akses <i>user</i> . Pada sistem ini admin hanya <i>login</i> , <i>logout</i> , ubah <i>password</i> dan kelola <i>account</i> seperti tambah data hapus data edit data seperti data agen, loket, jadwal, supir dan bus . |
| 2 | <i>Client</i> | Orang yang melakukan pengaksesan dalam |

| | | |
|---|-------|---|
| | | transaksi pembelian tiket. Pada sistem ini, <i>client</i> dapat <i>login</i> , <i>logout</i> , ubah password, mencari jadwal keberangkatan bus, pesan tiket bus dan melihat bukti transaksi pembayaran tiket. |
| 3 | Agen | Orang yang menambah data keberangkatan sesuai dengan daftar loket masing-masing. |
| 4 | Supir | Orang yang menghandel posisi bus yang dapat diakses <i>client</i> bahwa bus sudah berangkat atau belum. |

2. Perancangan *use case* diagram Aplikasi *Online* Pembelian Tiket Bus Dengan Layanan *Location Based Service*.

Setelah mengidentifikasi aktor dan *use case*, maka tahap selanjutnya adalah perancangan *use case* diagram. Berikut ini gambar dan keterangan *use case* pembelian tiket bus.

Berdasarkan analisa keputusan, orang yang berhak mengakses sistem berjalan berjumlah 4 (empat) *actor* diantaranya admin,client, agen dan supir.



Gambar 4.3 Use Case Model Diagram Aplikasi *Online* Pembelian Tiket Bus Dengan Layanan *Location Based Service*.

Pada *use case* diagram diatas, terdapat 4 (empat) aktor yang memiliki hak akses untuk *login* terlebih dahulu untuk masuk ke sistem yang diantaranya adalah admin, *client*, agen dan supir. Admin mendapatkan hak akses melakukan kelola master data seperti menambah mengedit dan menghapus data seperti data loket, data agen, data bus, data supir, data jadwal berangkat, dan daftar data pemesanan. Agen juga mendapatkan hak akses sama seperti admin yang melakukan kelola master data seperti menambah mengedit dan menghapus *update* data seperti data agen, data bus, data supir, data jadwal berangkat, dan daftar data pemesanan namun hanya saja agen harus mendaftar terlebih dahulu kepada admin untuk

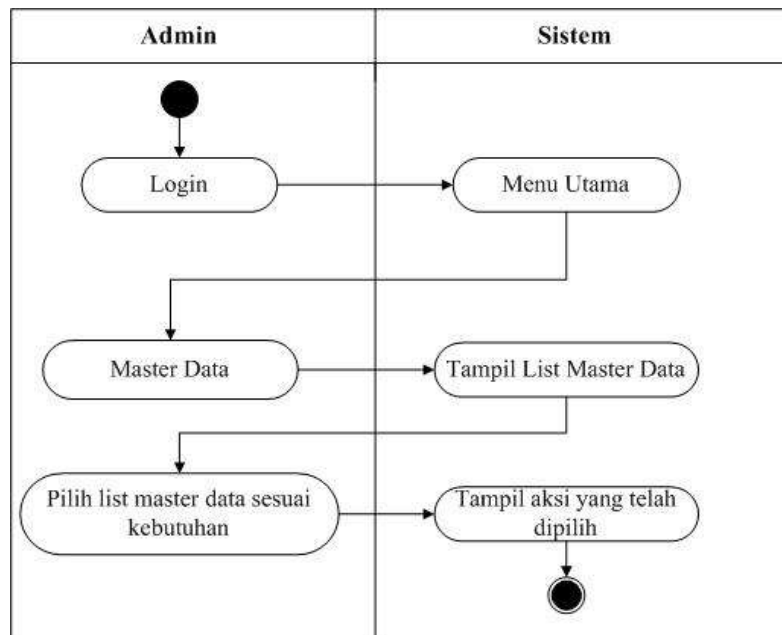
mendapatkan hak akses. *Client* mendapatkan hak akses untuk mencari tiket, lihat daftar bus yang tersedia, input data penumpang, pembayaran, lokasi naik, detail pemesanan, *more*, pesanan. Supir mendapatkan hak akses untuk cek orderan dan meng-*handle* bus apakah sudah berjalan (mulai) atau belum dan bus sudah tiba (selesai).

4.2.1.2 Activity Diagram Aplikasi Online Pembelian Tiket Bus Dengan Layanan Location Based Service

Setelah membuat *use case* diagram, tahap selanjutnya adalah pembuatan *activity diagram* yang berguna untuk menggambarkan aktifitas yang dilakukan oleh sistem yang sudah dibuat. Berikut ini adalah gambar dan keterangan *activity diagram* pada pembelian tiket bus.

1. *Activity diagram* kelola Master Data

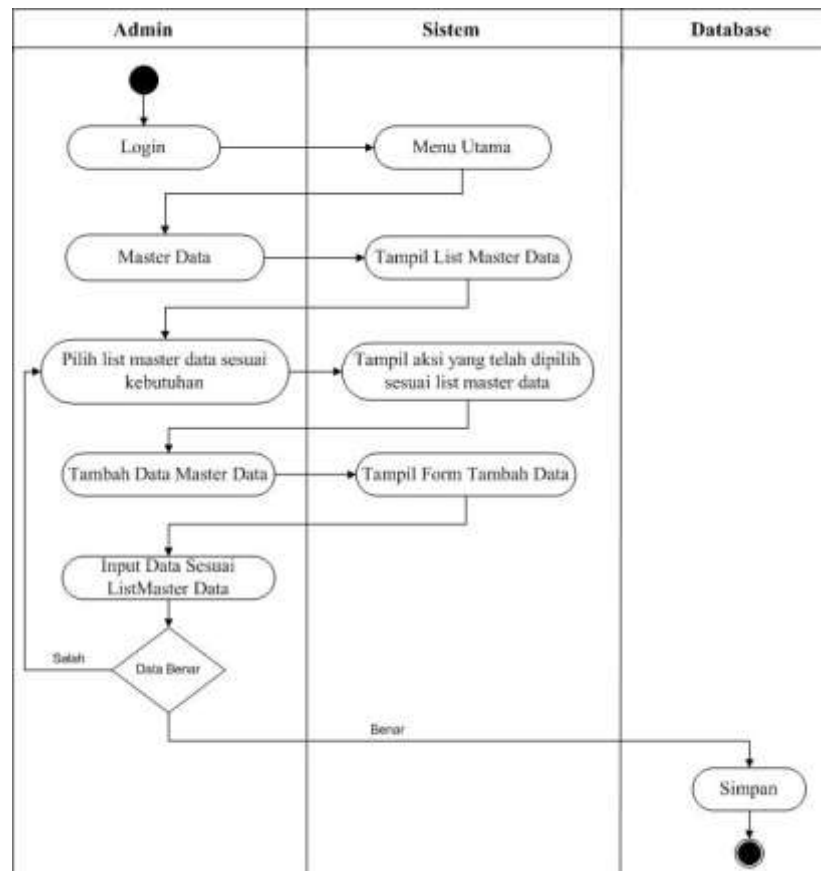
Activity ini menggambarkan kelola master data yang hanya bisa diakses oleh admin, dimana admin terlebih dahulu melakukan *login* untuk masuk kedalam sistem. Pertama sistem akan menampilkan menu utama admin yang akan melakukan kelolah master data. Jika admin memilih data loket maka sistem otomatis akan menampilkan *form* data loket, lalu admin mengisi data loket tersebut, jika data benar maka otomatis akan tersimpan ke dalam *database* dan begitu pula aksi-aksi yang akan dipilih admin selain data loket maka sistem akan memberikan *form* sesuai dengan yang dipilih admin.



Gambar 4.4 Activity Diagram Master Data

2. Activity diagram dari input data

Activity diagram ini menggambarkan *input* data yang berkaitan dengan pemesanan tiket bus. Admin atau pun agen terlebih dahulu melakukan *login* untuk masuk ke sistem, kemudian sistem akan menampilkan halaman utama. Lalu memilih Master Data, sistem akan menampilkan Data loket, data agen, data bus, data supir, jadwal keberangkatan, data titik jemput yang terdapat tiga pilihan aksi diantaranya *input*, *delete*, *edit*. jika pilih *edit* maka sistem akan menampilkan form *edit* jika pilih *input* maka akan menampilkan *input* data, lalu admin/agen mengisi form, jika memilih simpan maka perubahan data akan tersimpan, jika memilih batal maka perubahan jadwal bus tidak akan tersimpan. namun jika memilih *delete* maka sistem akan otomatis menghapus data di dalam *database*. Berikut ini gambar dari *activity diagram input* data.

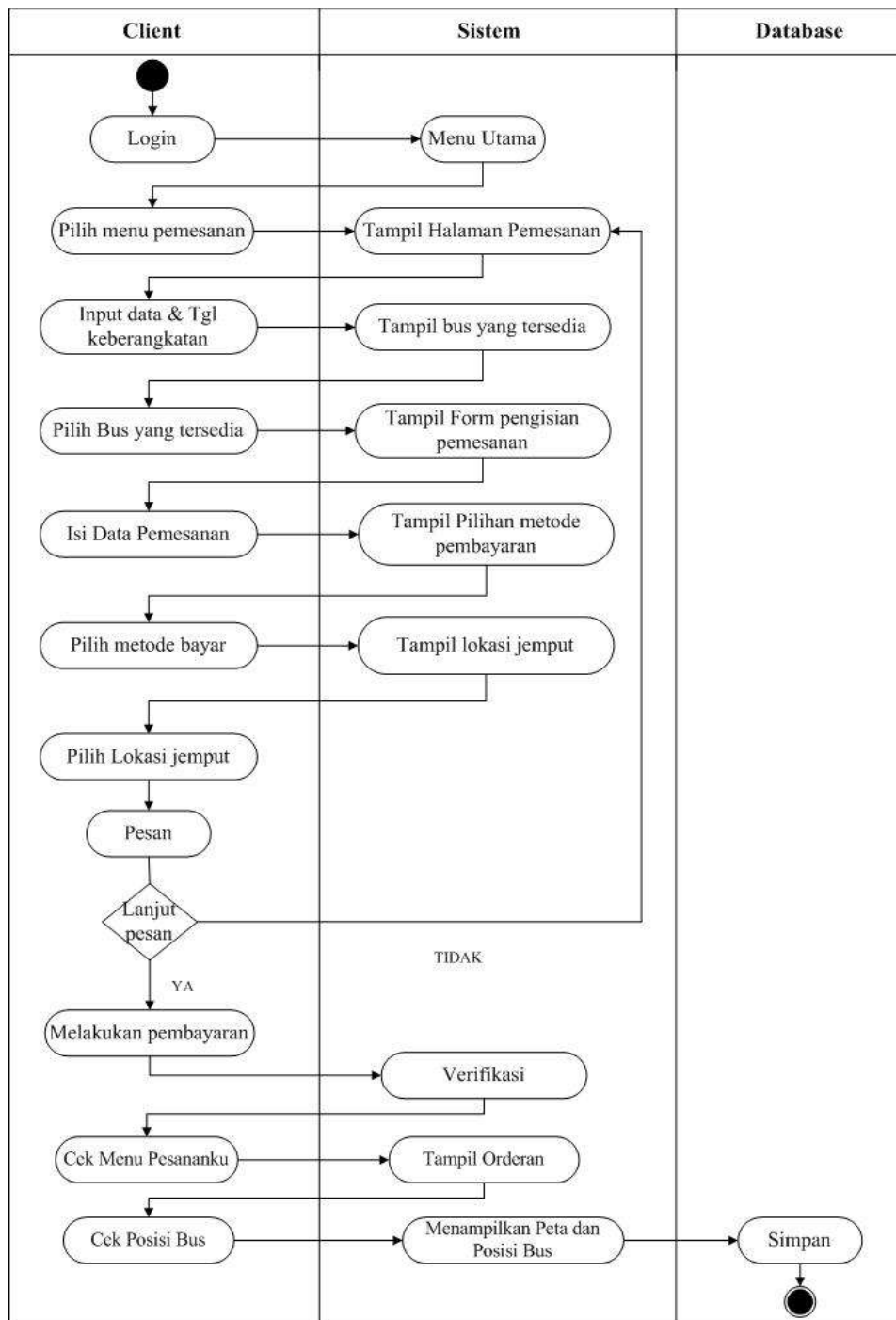


Gambar 4.5 Activity Diagram Input Data

3. Activity diagram dari pemesanan tiket

Activity diagram ini menggambarkan proses pemesanan yang dilakukan oleh *client* untuk pemesanan tiket bus. Terlebih dahulu melakukan *login* untuk masuk ke dalam sistem, sistem menampilkan halaman utama. *Client* memilih menu pemesanan, sistem menampilkan halaman pemesanan yang meliputi data keberangkatan seperti tujuan awal, tujuan akhir dan tgl berangkat, *client* mengisi data keberangkatan, sistem menampilkan daftar bus yang tersedia, lalu *client* memilih bus yang diinginkan sesuai jam berangkat lalu sistem menampilkan form untuk pengisian data pemesanan, *client* mengisi data dan sistem menampilkan halaman pilih metode pembayaran, lalu *client* memilih metode bayar dan lokasi naik dan otomatis sistem menampilkan detail pemesanan yang telah dilakukan. Jika *client* melanjutkan untuk pemesanan maka *client* melakukan transaksi pembayaran, lalu sistem akan melakukan

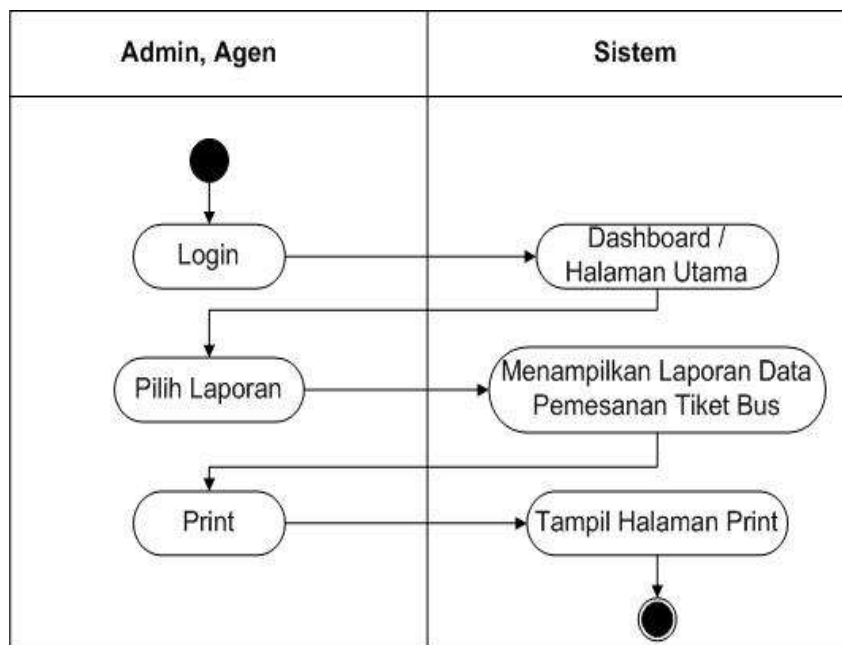
verifikasi pembayaran yang telah dilakukan oleh *client* dan *database* pun otomatis melakukan penyimpanan. Namun jika *client* tidak ingin melakukan lanjut pemesanan maka sistem akan kembali ke menu pemesanan.



Gambar 4.6 Activity Diagram Pemesanan Tiket

4. *Activity diagram* dari laporan pemesanan

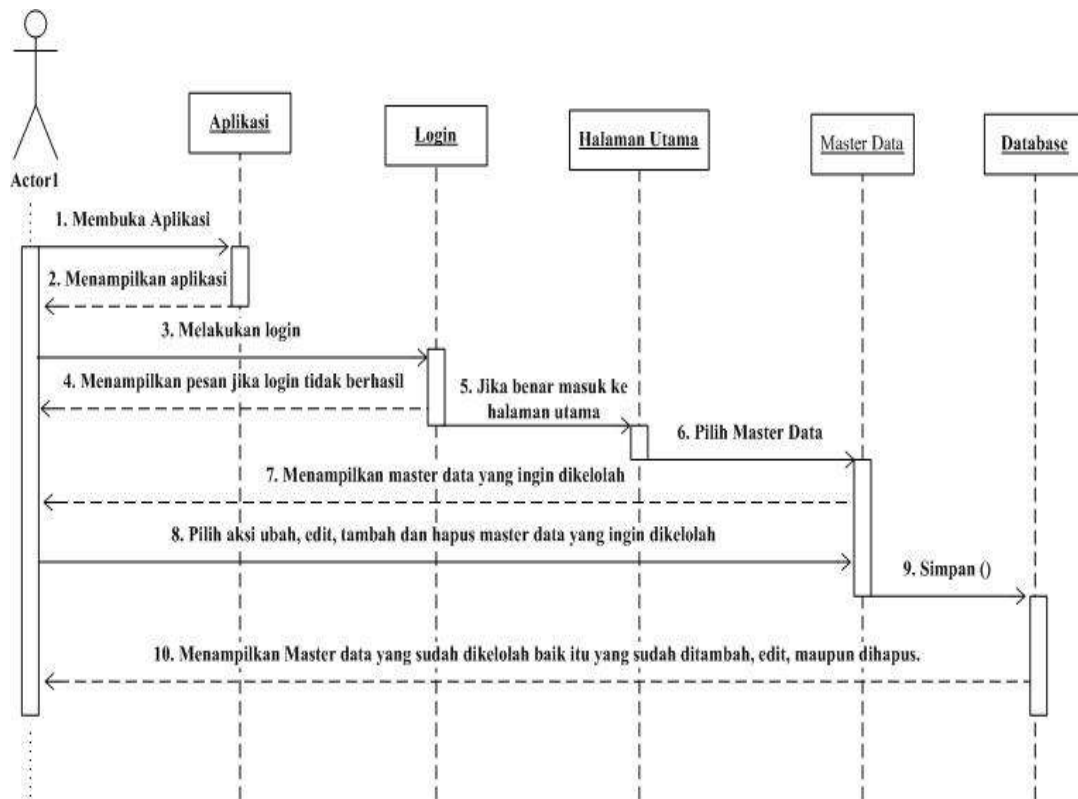
Activity diagram ini menggambarkan aktor (admin, agen) yang terlibat harus melakukan *login* terlebih dahulu untuk mengakses sebuah sistem, lalu sistem akan menampilkan halaman utama yaitu dashboard admin maupun agen tergantung aktor yang akan *login*. Disini Aktor akan laporan dan otomatis sistem menampilkan tabel daftar pesan, jika admin yang melakukan login maka admin bisa melihat seluruh laporan tetapi jika agen yang melakukan login maka hanya bisa melihat laporan satu agen saja tidak bisa melihat laporan dari agen yang lain. Berikut *activity diagram* laporan pemesanan.



4.2.1.3 *Sequence Diagram* Aplikasi Online Pembelian Tiket Bus Dengan Layanan Location Based Service

Setelah membuat *activity diagram*, kemudian tahap selanjutnya adalah membuat *sequence diagram* untuk menggambarkan bagaimana antara objek berinteraksi satu dengan yang lainnya melalui pesan terkirim dan pesan diterima diantara objek dan *sequence* nya apa. Berikut ini gambar dan keterangan *sequence diagram* untuk pembelian tiket bus yang telah dirancang.

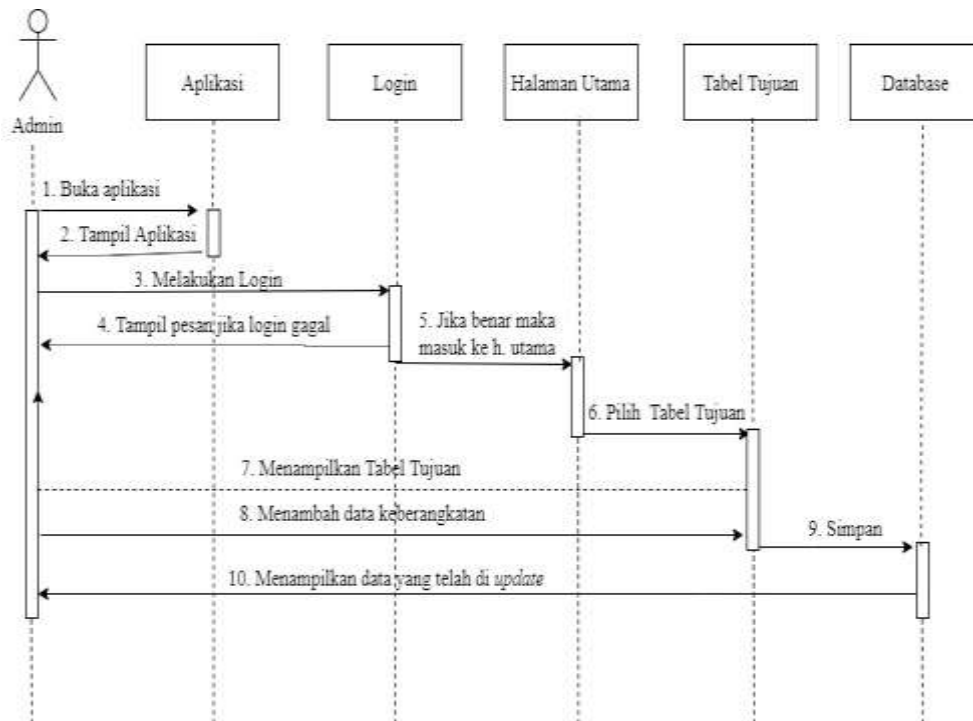
1. Sequence Diagram Kelola Master Data



Gambar 4.8 Sequence Diagram Kelola Master Data

Gambar 4.8 Sequence diagram kelola master data menerangkan bahwa message 1-3 Admin membuka aplikasi dan melakukan login. Message 4-5 jika username dan password salah akan menampilkan pesan login gagal dan harus melakukan login ulang, namun jika login berhasil maka akan masuk ke halaman utama. Message 6-10 proses dimana admin sudah memilih master data yang ingin diubah, ditambah dan dihapus yang akan disimpan ke dalam database.

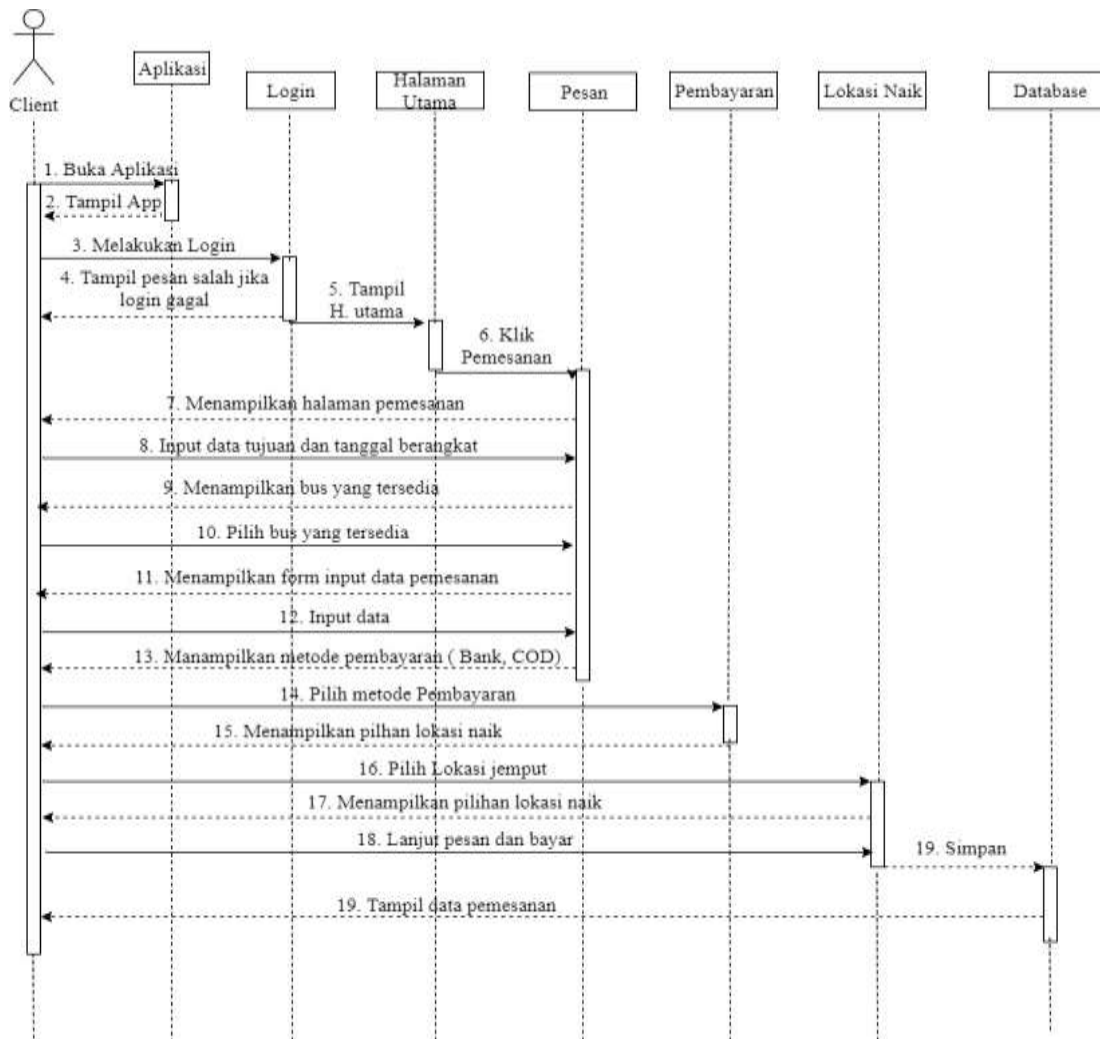
2. Sequence Diagram Input Data



Gambar 4.9 Sequence Diagram Input Data

Gambar 4.9 Sequence diagram input data menerangkan bahwa message 1-3 aktor membuka aplikasi dan melakukan login. Message 4-5 jika username dan password salah sistem akan menampilkan pesan login gagal dan harus melakukan login ulang, namun jika login berhasil maka akan masuk ke halaman utama. Message 5-10 proses dimana login sudah berhasil dan masuk ke halaman utama lalu meng-update data tujuan yang ingin ditambah dan diubah maupun dihapus, maka otomatis database akan menyimpan data perubahan yang telah di update.

3. Sequence Diagram Pemesanan

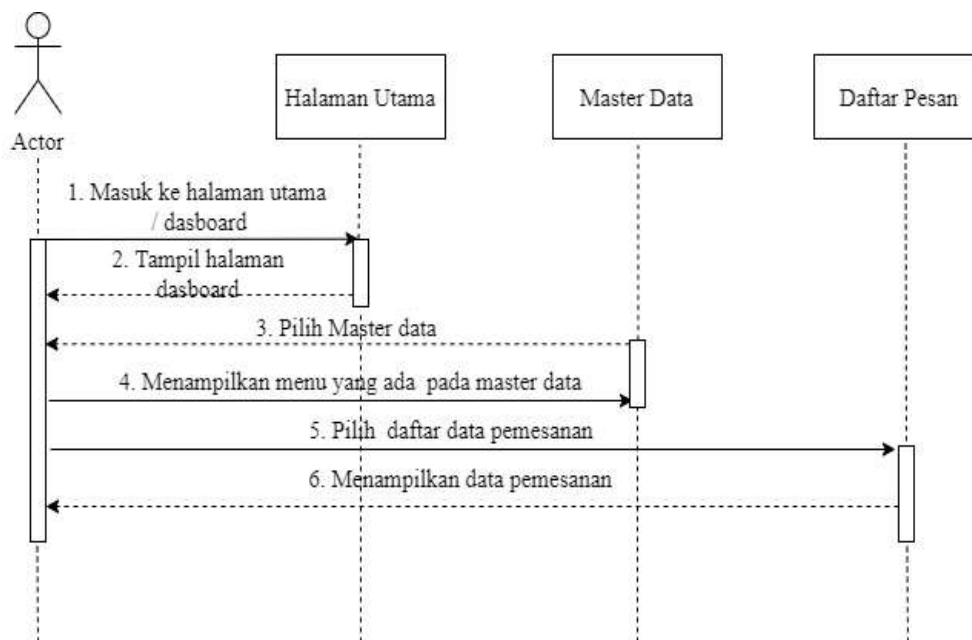


Gambar 4.10 Sequence Diagram Pemesanan

Gambar 4.10 Sequence diagram pemesanan menerangkan bahwa *message* 1-3 *client* membuka aplikasi dan melakukan *login*. *Message* 4-5 jika salah dalam memasukkan *username* dan *password* maka sistem akan menampilkan pesan *login* gagal dan harus melakukan *login* kembali, namun jika *login* berhasil maka akan masuk ke halaman utama. *Message* 6-13 menjelaskan proses *client* melakukan pemesanan mulai dari cari tgl keberangkatan, memilih bus yang tersedia, mengisi data keberangkatan, input data pemesanan, menentukan metode pembayaran. *Message* 14-19 proses dimana *client* akan memilih lokasi naik dan melanjutkan pembelian tiket dengan melakukan pembayaran melalui metode pembayaran yang

sudah dipilih *client* dan sudah ditetapkan oleh sistem, lalu setelah *client* sudah melakukan pembayaran maka sistem akan otomatis melakukan verifikasi mengenai status bayar atau belum dan sistem akan menyimpan data ke dalam *database* lalu kemudian *client* bisa melihat keterangan tiket yang sudah dipesan.

4. Sequence Diagram Lihat Laporan Pemesanan



Gambar 4.11 Sequence Diagram Lihat Laporan Pemesanan

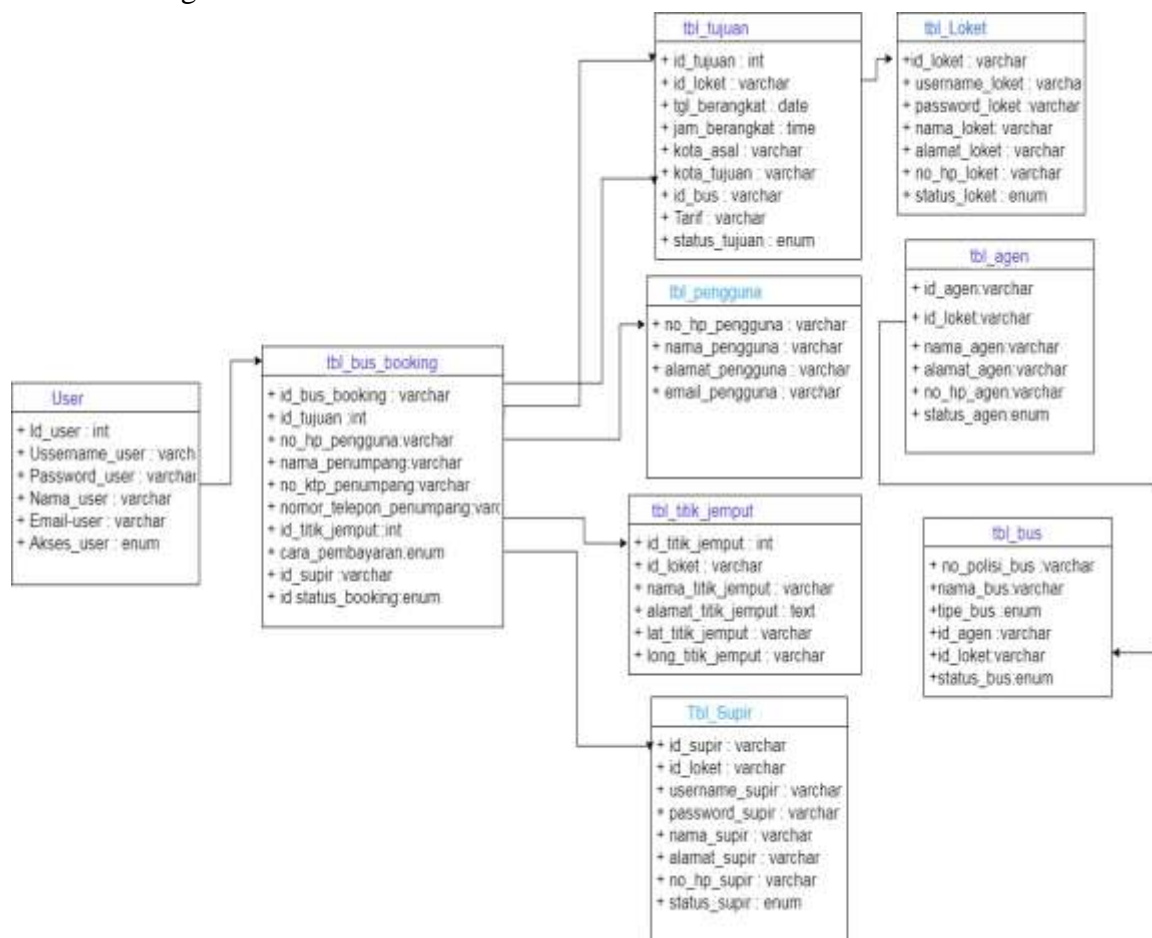
Gambar 4.11 Sequence diagram lihat laporan pemesanan menerangkan bahwa *message* 1-2 aktor sudah masuk ke sistem dan masuk ke halaman utama, *Message* 3-6 aktor memilih data master dan melihat daftar laporan pemesanan tiket yang berjalan.

4.2.2 Desain Database

Pada tahap ini dijelaskan tentang bagaimana mendesain *database* sistem dimulai dari *class diagram* lalu membuat *Entity Relationship Diagram (ERD)* setelah itu merancang spesifikasi *database* lalu membuat relasi antar tabel dengan menentukan *foreign key* dan *primary key* pada *class*. Berikut ini adalah tahapan yang dilakukan untuk mendesain *database*:

4.2.2.1 Class Diagram Pembelian Tiket Bus

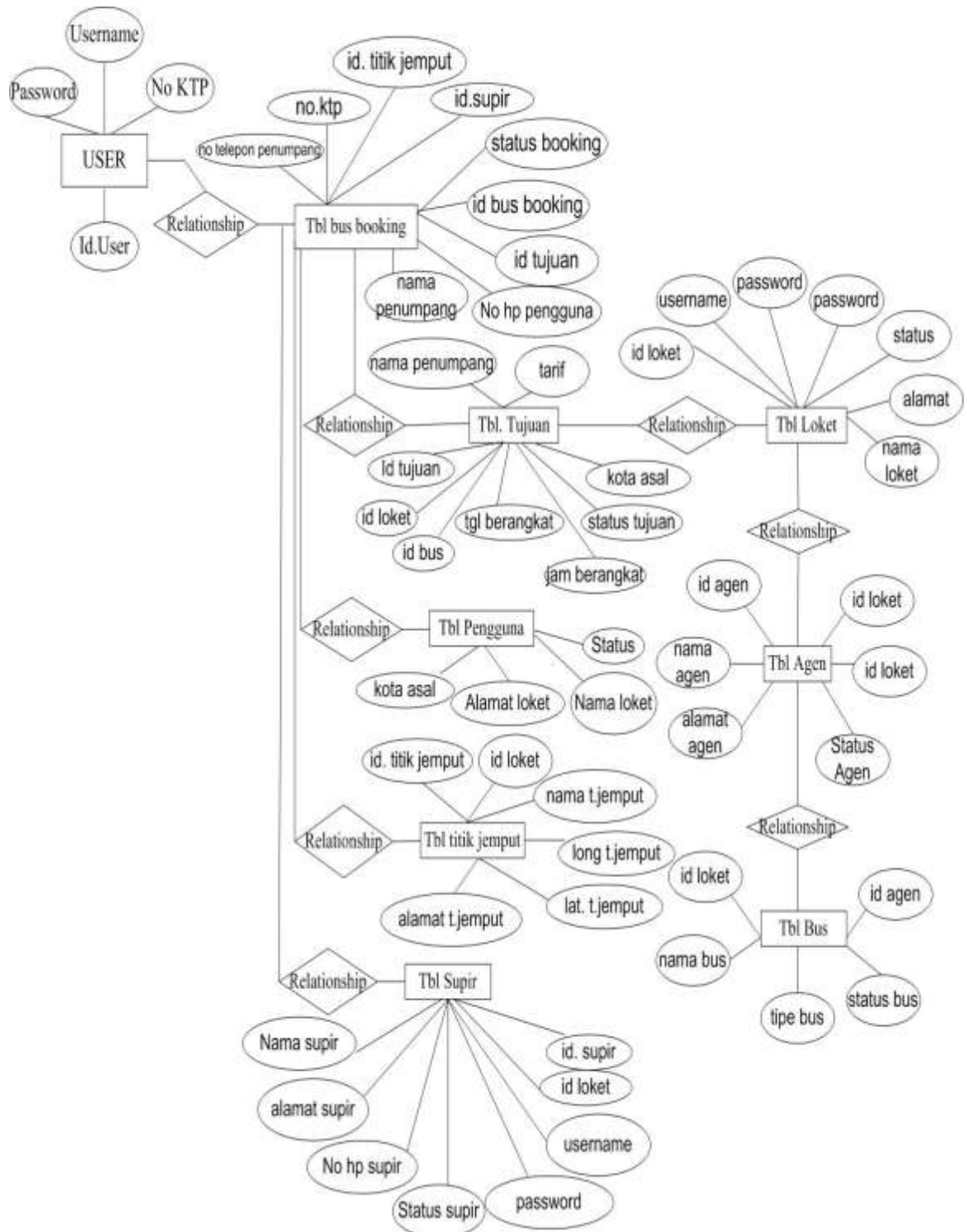
Class diagram berfungsi untuk menggambarkan kumpulan dari *class* dan hubungannya. *Class diagram* ini merupakan diagram yang paling umum ditemukan dalam proses pemodelan sistem berorientasi objek. *Class* menggambarkan keadaan suatu sistem dan *class* juga bisa menjadi sebuah *form* saat pembuatan program. *Class diagram* yang diusulkan pada penelitian ini adalah sebagai berikut :



Gambar 4.12 Class Diagram

4.2.2.2 Entity Relationship Diagram (ERD) Pembelian Tiket Bus

Adapun ERD yang dirancang pada penelitian ini adalah sebagai berikut :



Gambar 4.13 Entity Relationship Diagram (ERD)

4.2.2.3 Spesifikasi Database

Pada penelitian ini peneliti mempunyai spesifikasi database "aplika29_bus" yang dapat diakses secara *online* di *localhost/phpmyadmin/* ada pada gambar dibawah ini:

1. Agen

Nama Tabel : tbl_agen

PrimaryKey : id_agen

Tabel 4.5 Spesifikasi Data Agen

| Nama <i>field</i> | Tipe data | Ukuran |
|-------------------|-----------|------------------------|
| id_agen | Varchar | 10 |
| id_loket | Varchar | 10 |
| nama_agen | Varchar | 50 |
| alamat_agen | Varchar | 100 |
| no_hp_agen | Varchar | 13 |
| status_agen | Enum | 'Aktif', 'Tidak Aktif' |

2. Bus

Nama Tabel : tbl_bus

PrimaryKey : no_polisi_bus

ForeignKey:-

Tabel 4.6 Spesifikasi Data Bus

| Nama <i>Field</i> | Tipe Data | Ukuran |
|-------------------|-----------|--|
| no_polisi_bus | Varchar | 15 |
| nama_bus | Varchar | 100 |
| tipe_bus | Enum | 'AC', 'AC TOILET', 'EKONOMI', 'ROYAL CLASS' |

| | | |
|------------|---------|---|
| id_agen | Varchar | 10 |
| id_loket | Varchar | 50 |
| status_bus | Enum | 'Tersedia', 'Berangkat', 'Tidak Tersedia' |

3. Pesan

Nama Tabel:tbl_bus_booking

PrimaryKey:id_bus_booking

ForeignKey:-

Tabel 4.7 Spesifikasi Data Pesan Bus

| Nama <i>fiel</i> | Tipe data | Ukuran |
|-------------------------|-----------|---|
| id_bus_booking | Varchar | 50 |
| id_tujuan | Int | 11 |
| no_hp_pengguna | Varchar | 15 |
| nama_penumpang | Varchar | 100 |
| no_ktp_penumpang | Varchar | 20 |
| nomor_telepon_penumpang | Varchar | 13 |
| id_titik_jemput | Int | 11 |
| cara_pembayaran | Enum | 'TFBANK', 'COD' |
| id_supir | Varchar | 50 |
| status_book | Enum | 'Menunggu Pembayaran', 'Sudah Dibayar' |

4. Loker

Nama Tabel : tbl_loket

PrimaryKey : id_loket

ForeignKey : -

Tabel 4.8 Spesifikasi Data Locket

| Nama <i>field</i> | Tipe data | Ukuran |
|-------------------|-----------|------------------------|
| id_loket | Varchar | 10 |
| username_loket | Varchar | 50 |
| password_loket | Varchar | 100 |
| nama_loket | Varchar | 50 |
| alamat_loket | Varchar | 100 |
| no_hp_loket | Varchar | 13 |
| status_loket | Enum | 'Aktif', 'Tidak Aktif' |

5. Pengguna

Nama Tabel : tbl_pengguna

PrimaryKey : no_hp_pengguna

ForeignKey : -

Tabel 4.9 Spesifikasi Data Pengguna

| Nama <i>field</i> | Tipe data | Ukuran |
|-------------------|-----------|--------|
| no_hp_pengguna | Varchar | 15 |
| nama_pengguna | Varchar | 50 |
| alamat_pengguna | Varchar | 100 |
| email_pengguna | Varchar | 50 |

6. Supir

Nama Tabel : tbl_supir

PrimaryKey : *id_supir*

ForeignKey : -

Tabel 4.10 Spesifikasi Data Supir

| Nama <i>field</i> | Tipe data | Ukuran |
|-------------------|-----------|----------------------|
| id_supir | Varchar | 50 |
| id_loket | Varchar | 50 |
| username_supir | Varchar | 50 |
| password_supir | Varchar | 100 |
| nama_supir | Varchar | 50 |
| alamat_supir | Varchar | 100 |
| no_hp_supir | Varchar | 13 |
| status_supir | Enum | Aktif, 'Tidak Aktif' |

7. Titik Jemput

Nama Tabel : *tbl_titik_jemput*

PrimaryKey : *id_titik_jemput*

ForeignKey : -

Tabel 4.11 Spesifikasi Data Titik Jemput

| Nama <i>field</i> | Tipe data | Ukuran |
|---------------------|-----------|--------|
| id_titik_jemput | Int | 11 |
| id_loket | Varchar | 50 |
| nama_titik_jemput | Varchar | 100 |
| alamat_titik_jemput | Text | |

| | | |
|-------------------|---------|----|
| lat_titik_jemput | Varchar | 50 |
| long_titik_jemput | Varchar | 50 |

8. Tujuan

Nama Tabel : tbl_tujuan

PrimaryKey : id_tujuan

ForeignKey : -

Tabel 4.12 Spesifikasi Data Tujuan

| Nama <i>field</i> | Tipe data | Ukuran |
|-------------------|-----------|---|
| id_tujuan | Int | 11 |
| id_loket | Varchar | 50 |
| tgl_berangkat | Date | |
| jam_berangkat | Time | |
| kota_asal | Varchar | 100 |
| kota_tujuan | Varchar | 50 |
| id_bus | Varchar | 50 |
| Tarif | Varchar | 100 |
| status_tujuan | Enum | 'Tersedia', 'Tidak Tersedia', 'Dalam Perjalanan' |

9. User

Nama Tabel : tbl_user

PrimaryKey : id-user

ForeignKey : -

Tabel 4.13 Spesifikasi Data user

| Nama <i>field</i> | Tipe data | Ukuran |
|-------------------|-----------|-----------------|
| Id_user | Int | 11 |
| Usurname_user | Varchar | 25 |
| Password_user | Varchar | 50 |
| Nama_user | Varchar | 50 |
| Email-user | Varchar | 25 |
| Akses_user | Enum | 'admin', 'agen' |

4.2.3 Desain *Interface*

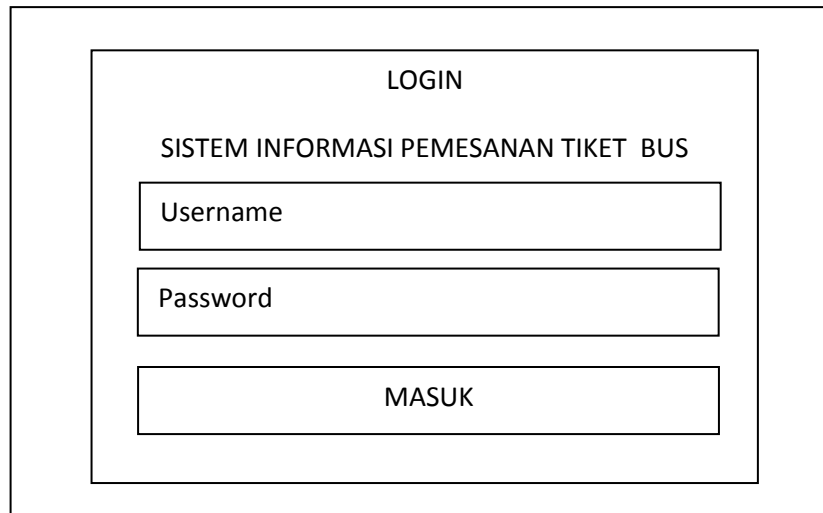
Pada tahap ini akan dilakukan perancangan pada *layout* yang akan menggambarkan *interface* antarmuka sistem. Adapun Perancangan *layout* pada penelitian ini dibedakan menjadi 2 (dua) yaitu proses penginputan data yang akan dilakukan oleh admin dan bisa dilakukan juga oleh agen berada pada tampilan *website* namun pada bagian proses pemesanan tiket bus yang akan dilakukan oleh *client* serta mengontrol keberadaan bus yang akan dilakukan oleh supir berada pada aplikasi Android, berikut adalah perancangan *layout* pada penelitian ini antara lain :

4.2.3.1 Rancangan Halaman *Admin* dan *Agen* (*Website*)

Halaman ini hanya bisa dapat diakses oleh admin dan agen dengan cara harus melakukan *login* untuk dapat masuk kedalam sistem yang akan dikelola.

1. Halaman *Login Admin* dan *Agen*

Form Login merupakan halaman untuk autentikasi pengguna. Setiap user yang akan menggunakan wajib untuk memasukkan *username* dan *password* agar dapat masuk ke dalam Sistem. Pengguna yang tidak memiliki *username* dan *password* tidak berhak untuk menggunakan atau mengelola sistem.



LOGIN

SISTEM INFORMASI PEMESANAN TIKET BUS

Username

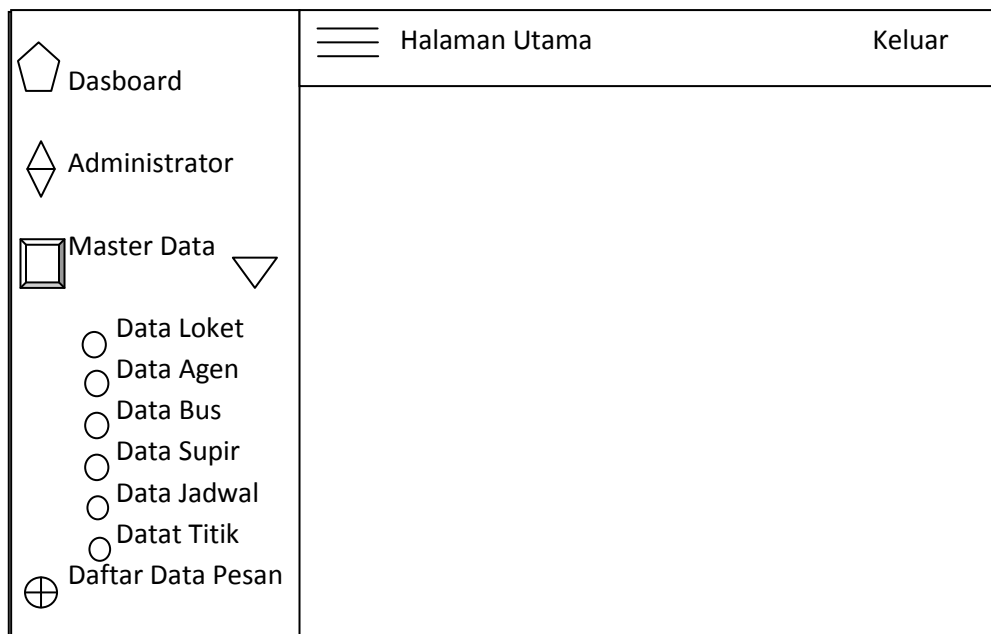
Password







MASUK

Gambar 4.14 Rancangan Halaman Login Admin dan Agen

2. Halaman Dashboard Admin

Halaman ini hanya bisa diakses oleh admin dengan cara harus *login* terlebih dahulu karena hanya admin yang mempunyai wewenang atas sistem yang akan dikelola yang ada pada halaman dashboard.

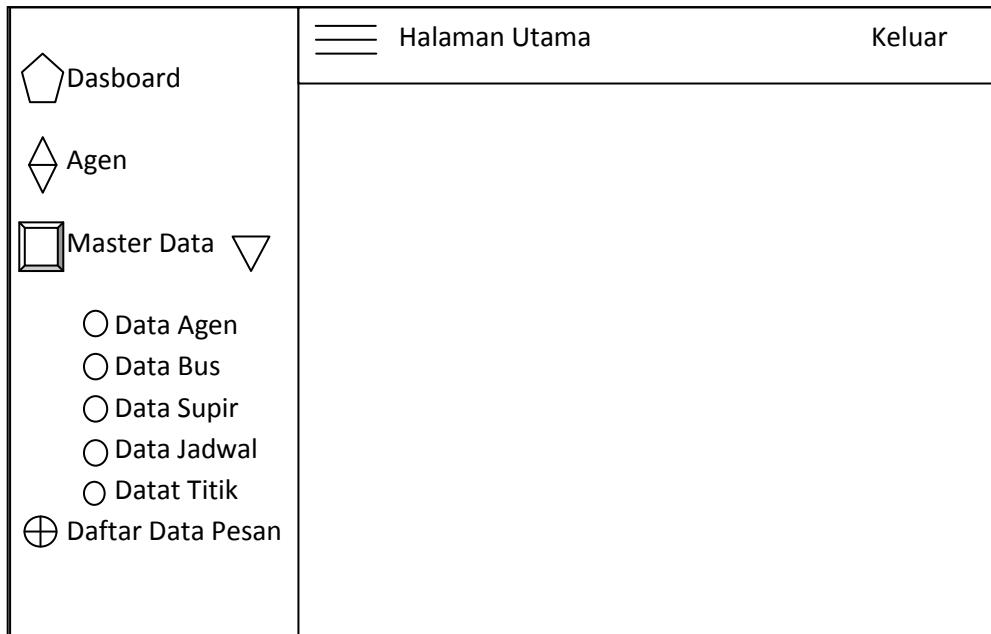


| | |
|---|---|
| <ul style="list-style-type: none">  Dashboard  Administrator  Master Data  <li style="margin-left: 20px;"><input type="radio"/> Data Locket <li style="margin-left: 20px;"><input type="radio"/> Data Agen <li style="margin-left: 20px;"><input type="radio"/> Data Bus <li style="margin-left: 20px;"><input type="radio"/> Data Supir <li style="margin-left: 20px;"><input type="radio"/> Data Jadwal <li style="margin-left: 20px;"><input type="radio"/> Data Titik  Daftar Data Pesan | <div style="display: flex; justify-content: space-between; align-items: center;">  Halaman Utama Keluar </div> <div style="height: 200px; border: 1px solid black; margin-top: 10px;"></div> |
|---|---|

Gambar 4.15 Rancangan Halaman Dashboard Admin

3. Halaman Dashboard Agen

Halaman ini hanya bisa diakses oleh agen dengan cara harus *login* terlebih dahulu karena hanya agen yang mempunyai wewenang atas sistem yang akan dikelola untuk menginput data pada halaman dashboard.



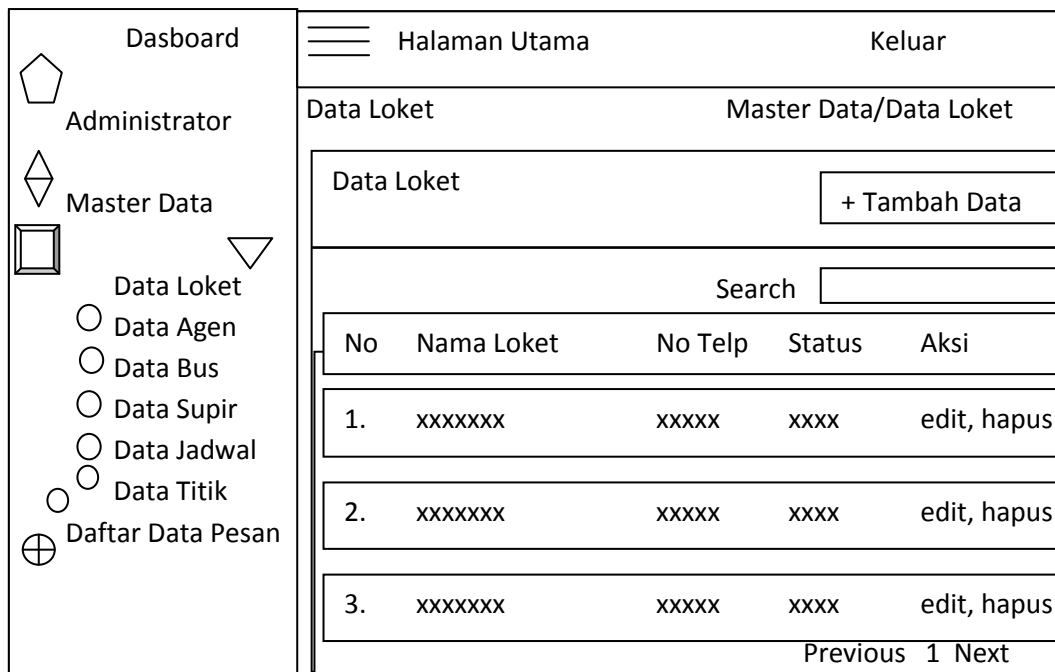
Gambar 4.16 Rancangan Halaman Dashboard Agen

4. Halaman Menu Master Data Admin dan Agen

Pada master data akan dilakukan proses penginputan data mengenai pemesanan tiket bus yang terdiri dari 5 (lima) diantaranya adalah sebagai berikut :

a. Data Loket

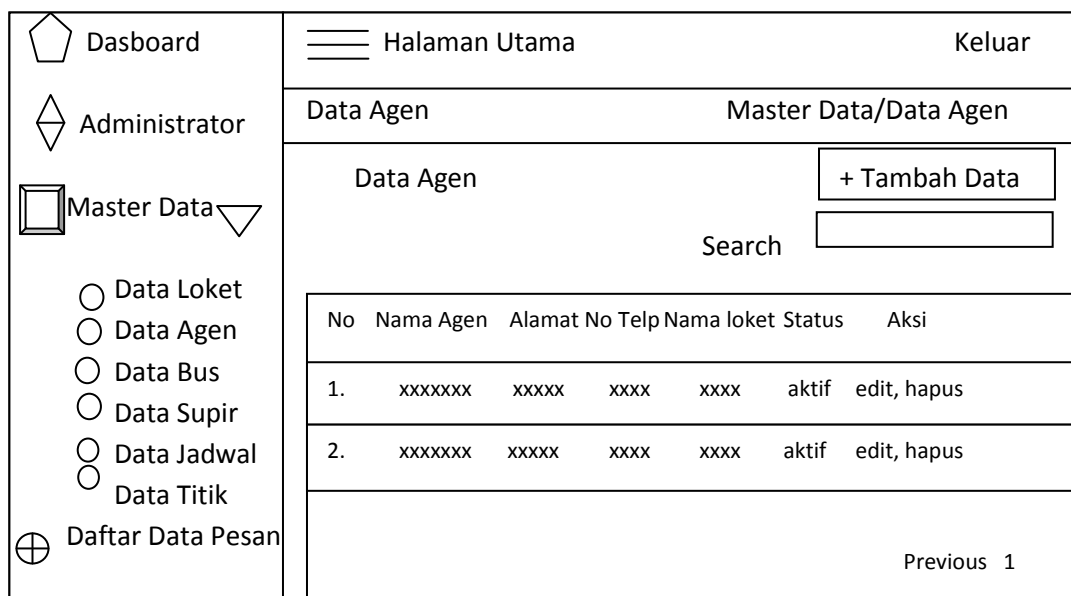
Halaman ini menampilkan proses penambahan data loket yang akan diinput dan didaftarkan oleh admin guna mengetahui status dari loket tersebut apakah masih beroperasi atau tidak.



Gambar 4.17 Rancangan Halaman Menu Data Locket Admin

b. Data Agen

Halaman ini berfungsi untuk menambah data agen dari setiap perusahaan guna mengetahui alamat agen setiap perusahaan yang terdaftar.



Gambar 4.18 Rancangan Halaman Menu Data Agen

c. Data Bus

Halaman ini berfungsi untuk menambah data bus guna mengetahui tipe dan no bus dari setiap perusahaan yang terdaftar.

| <ul style="list-style-type: none"> Dashboard Administrator Master Data ▾ <ul style="list-style-type: none"> <input type="radio"/> Data Locket <input type="radio"/> Data Agen <input type="radio"/> Data Bus <input type="radio"/> Data Supir <input type="radio"/> Data Jadwal <input type="radio"/> Data Titik Daftar Data Pesan | Halaman Utama Keluar | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---------------|---------------|----------|----------|-------------|-------------|-------------|------|----|---------|-------|------|---|------|-------|-------------|----|---------|-------|------|---|------|-------|-------------|----|---------|-------|------|---|------|-------|-------------|
| | Data Bus Master Data/Data Bus | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Data Bus <input type="button" value="+ Tambah Data"/> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Search <input type="text"/> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>No</th> <th>No Polisi Bus</th> <th>No Bus</th> <th>Tipe Bus</th> <th>Photo</th> <th>Nama locket</th> <th>Status</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>xxxxxxx</td> <td>xxxxx</td> <td>xxxx</td> <td>-</td> <td>xxxx</td> <td>aktif</td> <td>edit, hapus</td> </tr> <tr> <td>2.</td> <td>xxxxxxx</td> <td>xxxxx</td> <td>xxxx</td> <td>-</td> <td>xxxx</td> <td>aktif</td> <td>edit, hapus</td> </tr> <tr> <td>3.</td> <td>xxxxxxx</td> <td>xxxxx</td> <td>xxxx</td> <td>-</td> <td>xxxx</td> <td>aktif</td> <td>edit, hapus</td> </tr> </tbody> </table> | No | No Polisi Bus | No Bus | Tipe Bus | Photo | Nama locket | Status | Aksi | 1. | xxxxxxx | xxxxx | xxxx | - | xxxx | aktif | edit, hapus | 2. | xxxxxxx | xxxxx | xxxx | - | xxxx | aktif | edit, hapus | 3. | xxxxxxx | xxxxx | xxxx | - | xxxx | aktif | edit, hapus |
| | No | No Polisi Bus | No Bus | Tipe Bus | Photo | Nama locket | Status | Aksi | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 1. | xxxxxxx | xxxxx | xxxx | - | xxxx | aktif | edit, hapus | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2. | xxxxxxx | xxxxx | xxxx | - | xxxx | aktif | edit, hapus | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3. | xxxxxxx | xxxxx | xxxx | - | xxxx | aktif | edit, hapus | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Previous 1 Next | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Gambar 4.19 Rancangan Halaman Menu Data Bus

d. Data Supir

Halaman ini berfungsi untuk menambah data supir dari setiap perusahaan guna mengetahui identitas supir.

| <ul style="list-style-type: none"> Dashboard Administrator Master Data ▾ <ul style="list-style-type: none"> <input type="radio"/> Data Locket <input type="radio"/> Data Agen <input type="radio"/> Data Bus <input type="radio"/> Data Supir <input type="radio"/> Data Jadwal <input type="radio"/> Data Titik Daftar Data Pesan | Halaman Utama Keluar | | | | | | | | | | | | | | | | | | | | | |
|---|--|-------------|-------------|------------|--------|-------------|-------------|------|----|---------|-------|------|------|-------|-------------|----|---------|-------|------|------|-------|-------------|
| | Data Supir Master Data/Data Supir | | | | | | | | | | | | | | | | | | | | | |
| | Data Bus <input type="button" value="+ Tambah"/> | | | | | | | | | | | | | | | | | | | | | |
| | Search <input type="text"/> | | | | | | | | | | | | | | | | | | | | | |
| | <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>No</th> <th>Nama Locket</th> <th>Nama Supir</th> <th>Alamat</th> <th>No Hp</th> <th>Status</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>xxxxxxx</td> <td>xxxxx</td> <td>xxxx</td> <td>xxxx</td> <td>aktif</td> <td>edit, hapus</td> </tr> <tr> <td>3.</td> <td>xxxxxxx</td> <td>xxxxx</td> <td>xxxx</td> <td>xxxx</td> <td>aktif</td> <td>edit, hapus</td> </tr> </tbody> </table> | No | Nama Locket | Nama Supir | Alamat | No Hp | Status | Aksi | 1. | xxxxxxx | xxxxx | xxxx | xxxx | aktif | edit, hapus | 3. | xxxxxxx | xxxxx | xxxx | xxxx | aktif | edit, hapus |
| | No | Nama Locket | Nama Supir | Alamat | No Hp | Status | Aksi | | | | | | | | | | | | | | | |
| | 1. | xxxxxxx | xxxxx | xxxx | xxxx | aktif | edit, hapus | | | | | | | | | | | | | | | |
| 3. | xxxxxxx | xxxxx | xxxx | xxxx | aktif | edit, hapus | | | | | | | | | | | | | | | | |

Gambar 4.20 Rancangan Halaman Menu Data Supir

e. Data Jadwal Keberangkatan

Halaman ini berfungsi untuk menginput data jadwal keberangkatan dari setiap bus yang terdaftar berdasarkan jam yang telah ditentukan oleh perusahaan.

| <ul style="list-style-type: none"> Dashboard Administrator Master Data ▾ <ul style="list-style-type: none"> Data Locket Data Agen Data Bus Data Supir Data Jadwal Data Titik Daftar pesan | <div style="display: flex; justify-content: space-between;"> ☰ Halaman Utama Keluar </div> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|-------------|-------------|-----|------|--------|----------|----------|---------|-------------|-------|------|----|---------|---|----|----|----|----|----|----|-------|-------------|----|---------|---|----|----|----|----|----|----|-------|-------------|
| | <div style="display: flex; justify-content: space-between;"> Data Jadwal Keberangkatan Master Data/Data Jadwal Keberangkata </div> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <div style="display: flex; justify-content: flex-end; align-items: center;"> <input type="text" value="Search"/> <input type="button" value="+ Tambah Data"/> </div> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>No</th> <th>Nama Locket</th> <th>tgl</th> <th>Jam</th> <th>Asal</th> <th>Tujuan</th> <th>Nama Bus</th> <th>Tipe</th> <th>Tarif S</th> <th>tatus</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>xxxxxxx</td> <td>x</td> <td>xx</td> <td>xx</td> <td>xx</td> <td>xx</td> <td>xx</td> <td>xx</td> <td>aktif</td> <td>edit, hapus</td> </tr> <tr> <td>3.</td> <td>xxxxxxx</td> <td>x</td> <td>xx</td> <td>xx</td> <td>xx</td> <td>xx</td> <td>xx</td> <td>xx</td> <td>aktif</td> <td>edit, hapus</td> </tr> </tbody> </table> | No | Nama Locket | tgl | Jam | Asal | Tujuan | Nama Bus | Tipe | Tarif S | tatus | Aksi | 1. | xxxxxxx | x | xx | xx | xx | xx | xx | xx | aktif | edit, hapus | 3. | xxxxxxx | x | xx | xx | xx | xx | xx | xx | aktif | edit, hapus |
| | No | Nama Locket | tgl | Jam | Asal | Tujuan | Nama Bus | Tipe | Tarif S | tatus | Aksi | | | | | | | | | | | | | | | | | | | | | | | |
| 1. | xxxxxxx | x | xx | xx | xx | xx | xx | xx | aktif | edit, hapus | | | | | | | | | | | | | | | | | | | | | | | | |
| 3. | xxxxxxx | x | xx | xx | xx | xx | xx | xx | aktif | edit, hapus | | | | | | | | | | | | | | | | | | | | | | | | |
| <div style="display: flex; justify-content: flex-end; align-items: center;"> Previous 1 Next </div> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Gambar 4.21 Rancangan Halaman Menu Data Jadwal Keberangkatan

f. Data Titik Jemput

Halaman ini berfungsi untuk menampilkan data titik jemput berdasarkan alamat loket dari setiap perusahaan yang terdaftar.

| <ul style="list-style-type: none"> Dashboard Administrator Master Data ▾ <ul style="list-style-type: none"> Data Locket Data Agen Data Bus Data Supir DataJadwal Data Titik Daftar Pesan | <div style="display: flex; justify-content: space-between;"> Data Titik Jemput Master Data/Data Titik Jemput </div> | | | | | | | | | | | | | | | | | | | | | |
|---|--|-------------|--------------|--------------|-----------|-------------------|------|------|----|---------|-------|------|------|--|-------------------|----|---------|-------|------|------|--|-------------------|
| | <div style="display: flex; justify-content: flex-end; align-items: center;"> <input type="button" value="+ Tambah Data"/> </div> | | | | | | | | | | | | | | | | | | | | | |
| | <div style="display: flex; justify-content: flex-end; align-items: center;"> <input type="text" value="Search"/> </div> | | | | | | | | | | | | | | | | | | | | | |
| | <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>No</th> <th>Nama Locket</th> <th>Titik Jemput</th> <th>Alamat</th> <th>Lattitude</th> <th>Long</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>xxxxxxx</td> <td>xxxxx</td> <td>xxxx</td> <td>xxxx</td> <td></td> <td>aktif edit, hapus</td> </tr> <tr> <td>3.</td> <td>xxxxxxx</td> <td>xxxxx</td> <td>xxxx</td> <td>xxxx</td> <td></td> <td>aktif edit, hapus</td> </tr> </tbody> </table> | No | Nama Locket | Titik Jemput | Alamat | Lattitude | Long | Aksi | 1. | xxxxxxx | xxxxx | xxxx | xxxx | | aktif edit, hapus | 3. | xxxxxxx | xxxxx | xxxx | xxxx | | aktif edit, hapus |
| | No | Nama Locket | Titik Jemput | Alamat | Lattitude | Long | Aksi | | | | | | | | | | | | | | | |
| 1. | xxxxxxx | xxxxx | xxxx | xxxx | | aktif edit, hapus | | | | | | | | | | | | | | | | |
| 3. | xxxxxxx | xxxxx | xxxx | xxxx | | aktif edit, hapus | | | | | | | | | | | | | | | | |
| <div style="display: flex; justify-content: flex-end; align-items: center;"> Previous 1 Next </div> | | | | | | | | | | | | | | | | | | | | | | |

Gambar 4.22 Rancangan Halaman Menu Data Titik Jemput

g. Daftar Data Pesan

Halaman ini berfungsi untuk menampilkan data yang masuk mengenai pemesanan tiket bus yang dilakukan oleh *client*.

| <ul style="list-style-type: none"> 🏠 Dashboard 🔑 Administrator 📁 Master Data <ul style="list-style-type: none"> 🔍 Data Locket 🔍 Data Agen 🔍 Data Bus 🔍 Data Supir 🔍 Data Jadwal 🔍 Data Titik ⊕ Daftar Data Pesan | ☰ Halaman Utama Keluar | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---------|------|--------|--------|-------|-----------|-----------|-------|-------|----|---------|---|----|----|----|----|----|----|----|---------|---|----|----|----|----|----|----|----|---------|---|----|----|----|----|----|----|
| | Master Data/Daftar Data Pemesanan Tiket + Tambah Data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Daftar Data Pemesanan Tiket Search | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>No</th> <th>Nama</th> <th>No</th> <th>Tujuan</th> <th>Bus</th> <th>Supir</th> <th>Penumpang</th> <th>Hp</th> <th>Bayar</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>xxxxxxx</td> <td>x</td> <td>xx</td> <td>xx</td> <td>xx</td> <td>xx</td> <td>xx</td> <td>xx</td> </tr> <tr> <td>2.</td> <td>xxxxxxx</td> <td>x</td> <td>xx</td> <td>xx</td> <td>xx</td> <td>xx</td> <td>xx</td> <td>xx</td> </tr> <tr> <td>3.</td> <td>xxxxxxx</td> <td>x</td> <td>xx</td> <td>xx</td> <td>xx</td> <td>xx</td> <td>xx</td> <td>xx</td> </tr> </tbody> </table> | No | Nama | No | Tujuan | Bus | Supir | Penumpang | Hp | Bayar | 1. | xxxxxxx | x | xx | xx | xx | xx | xx | xx | 2. | xxxxxxx | x | xx | xx | xx | xx | xx | xx | 3. | xxxxxxx | x | xx | xx | xx | xx | xx | xx |
| | No | Nama | No | Tujuan | Bus | Supir | Penumpang | Hp | Bayar | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 1. | xxxxxxx | x | xx | xx | xx | xx | xx | xx | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2. | xxxxxxx | x | xx | xx | xx | xx | xx | xx | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3. | xxxxxxx | x | xx | xx | xx | xx | xx | xx | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Gambar 4.23 Rancangan Halaman Menu Daftar Data Pemesanan

h. Laporan

Pada halaman ini akan menampilkan seluruh data pemesanan tiket bus yang dilakukan oleh *client* yang akan dilihat oleh admin maupun agen dari setiap perusahaan yang terdaftar.

| <ul style="list-style-type: none"> 🏠 Dashboard 🔑 Administrator 📁 Master Data ▾ <ul style="list-style-type: none"> 🔍 Data Locket 🔍 Data Agen 🔍 Data Bus 🔍 Data Supir 🔍 Data Jadwal 🔍 Data Titik ⊕ Daftar Pesan | --- Halaman Utama Keluar | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|-----|---------|---------|---------|---------|-------|-----------|-----------|----|----|----|---|----|----|----|----|----|----|----|----|---|----|----|----|----|----|----|
| | Laporan Data Pemesanan Tiket + Tambah Data | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <div style="display: flex; justify-content: space-between; align-items: center;"> <div> EXCEL PDF Print </div> <div style="text-align: center;"> Search </div> </div> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>No</th> <th>Tgl</th> <th>Nama L.</th> <th>No T.</th> <th>Nama P.</th> <th>Bus</th> <th>Supir</th> <th>Penumpang</th> <th>Hp</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>xx</td> <td>x</td> <td>xx</td> <td>xx</td> <td>xx</td> <td>xx</td> <td>xx</td> <td>xx</td> </tr> <tr> <td>2.</td> <td>xx</td> <td>x</td> <td>xx</td> <td>xx</td> <td>xx</td> <td>xx</td> <td>xx</td> <td>xx</td> </tr> </tbody> </table> | No | Tgl | Nama L. | No T. | Nama P. | Bus | Supir | Penumpang | Hp | 1. | xx | x | xx | xx | xx | xx | xx | xx | 2. | xx | x | xx | xx | xx | xx | xx | xx |
| | No | Tgl | Nama L. | No T. | Nama P. | Bus | Supir | Penumpang | Hp | | | | | | | | | | | | | | | | | | | |
| | 1. | xx | x | xx | xx | xx | xx | xx | xx | | | | | | | | | | | | | | | | | | | |
| 2. | xx | x | xx | xx | xx | xx | xx | xx | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Gambar 4.24 Rancangan Halaman Laporan Pemesanan Tiket Bus

4.2.3.2 Rancangan Halaman *Client* dan Supir (Android)

Halaman ini hanya bisa dapat diakses oleh *client* yang akan melakukan transaksi pemesanan tiket bus dan supir yang *handle* bus sudah berangkat atau belum.

1. Halaman Login dan Verifikasi

Form Login merupakan halaman untuk autentikasi pengguna. Setiap *client* yang akan menggunakan aplikasi diharuskan untuk memasukkan Nomor Telepon dan memasukkan kode verifikasi masing-masing pengguna agar dapat masuk ke dalam Sistem.

Gambar 4.25 Rancangan Halaman *Login* dan *Verifikasi Client*

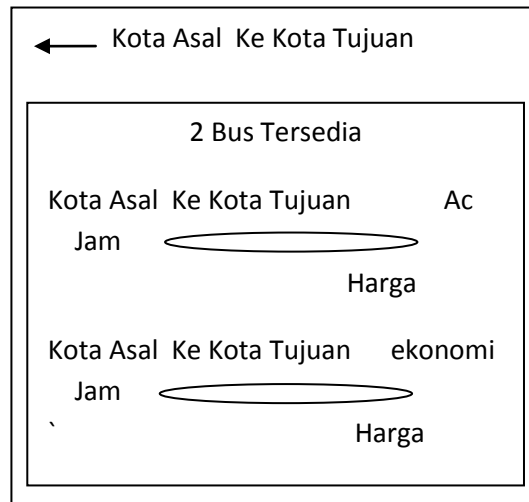
2. Halaman Dashboard

Halaman ini akan menampilkan informasi ketersediaan tiket bus dengan cara meng-*input* data tujuan sesuai keberangkatan.

Gambar 4.26 Rancangan Tampilan *Dashboard Client*

3. Halaman Bus Tersedia

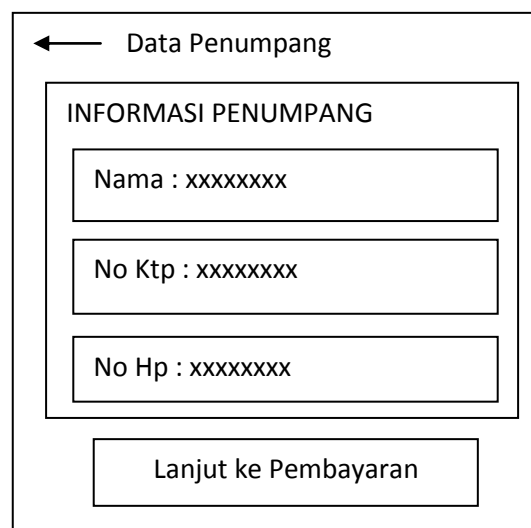
Pada halaman ini akan ditampilkan keseluruhan bus yang tersedia dari berbagai tipe bus sesuai dengan inputan tanggal yang di cari oleh *client*.



Gambar 4.27 Rancangan Halaman Bus yang Tersedia

4. Halaman Data Penumpang

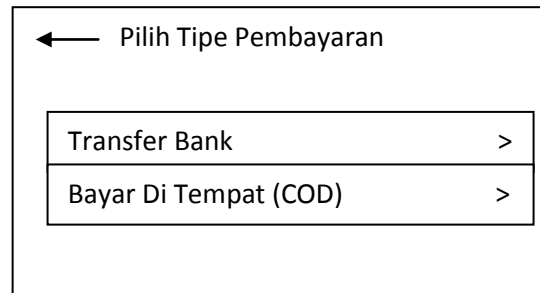
Setelah memilih bus yang tersedia maka *client* akan diarahkan untuk mengisi data calon penumpang untuk mengetahui identitas pengguna.



Gambar 4.28 Rancangan Halaman *Input Data Client*

5. Halaman Menu Pembayaran

Pada halaman ini akan ditampilkan 2 (dua) pilihan untuk proses pembayaran yang akan dilakukan oleh *client* untuk proses pemesanan tiket bus.



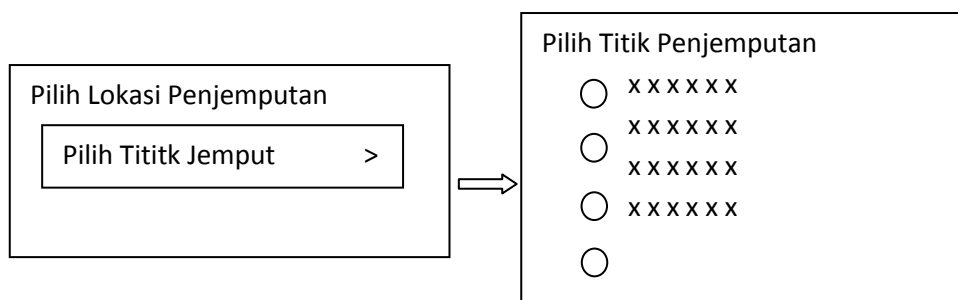
← Pilih Tipe Pembayaran

| | |
|-----------------------|---|
| Transfer Bank | > |
| Bayar Di Tempat (COD) | > |

Gambar 4.29 Rancangan Halaman Pilih Tipe Pembayaran

6. Halaman Pilih Lokasi Naik

Setelah memilih tipe pembayaran pengguna akan diarahkan ketampilan lokasi jemput yang di sesuaikan berdasarkan loket setiap bus yang akan berhenti.



Pilih Lokasi Penjemputan

| | |
|--------------------|---|
| Pilih Titik Jemput | > |
|--------------------|---|

⇒

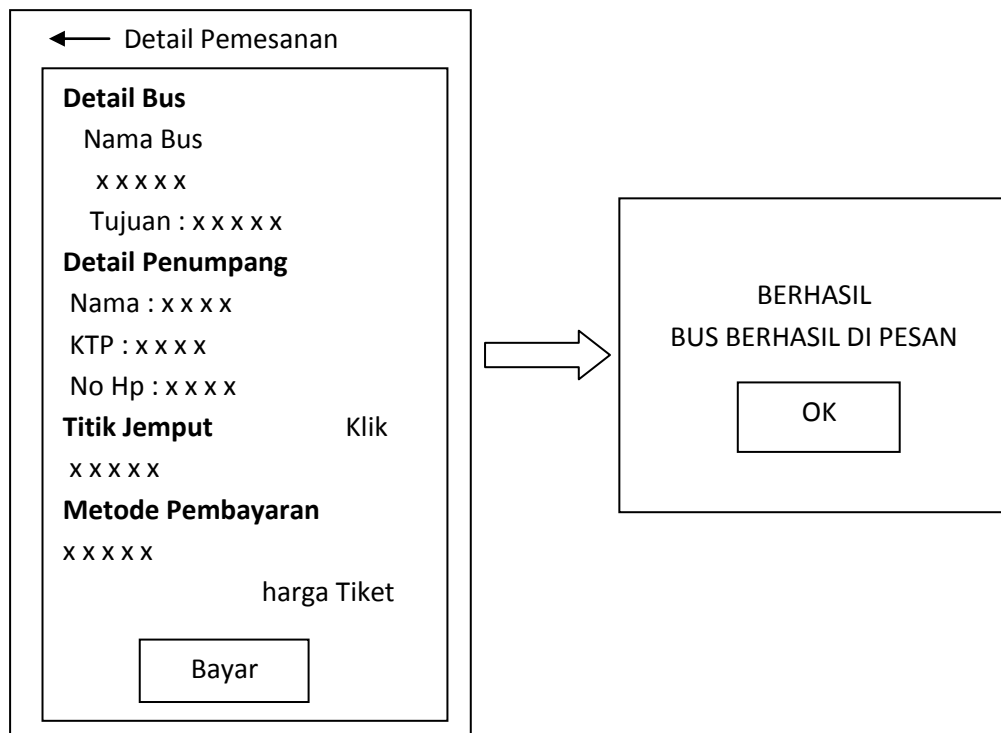
Pilih Titik Penjemputan

- x x x x x x
- x x x x x x
- x x x x x x
- x x x x x x
-

Gambar 4.30 Rancangan Halaman Pilihan Lokasi Penjemputan

7. Halaman Detail Pemesanan

Pada halaman ini akan ditampilkan seluruh data yang telah di *input* untuk melakukan pemesanan tiket bus.



Gambar 4.31 Rancangan Halaman Detail Pemesanan Tiket

4.3 Implementasi

Dalam membangun sebuah sistem yang sudah didesain, maka perlu dibuat implementasi, karena implementasi digunakan untuk pengujian terhadap suatu sistem apakah sistem sudah berjalan sesuai dengan analisa program yang telah dibuat.

4.3.1 Pengkodean

Dalam pengkodean untuk membuat dan membangun sistem ini dilakukan dengan menggunakan:

1. Bahasa pemrograman *Java*.
2. Perangkat keras:
 - *Processor* : *Intel ® Core™ i3 Duo 2.00 GHz.*
 - *Memory* : *DDR3 10 GB*
 - *Harddisk* : *500 GB*

- VGA : intel ® HD Graphics 3000
- Layar : 14 Inchi LED HD
- *Operating Sytem* : *Windows 7 ultimate* 64-bit

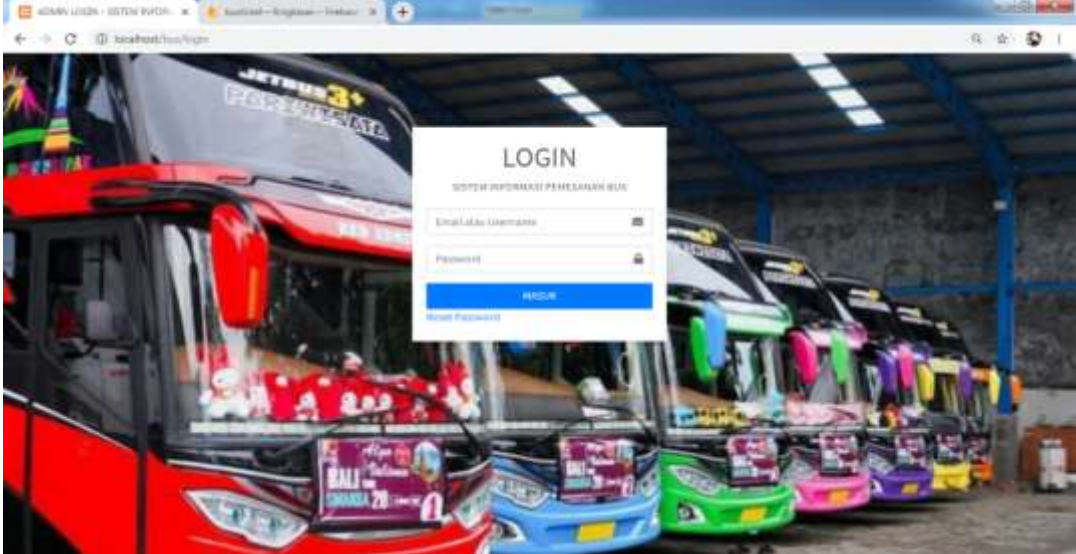
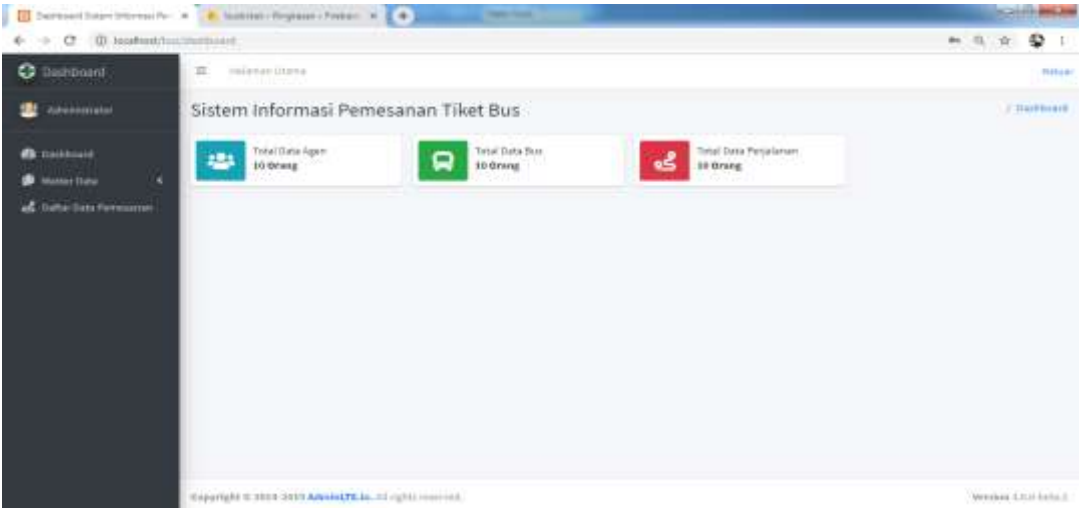
3. Perangkat lunak

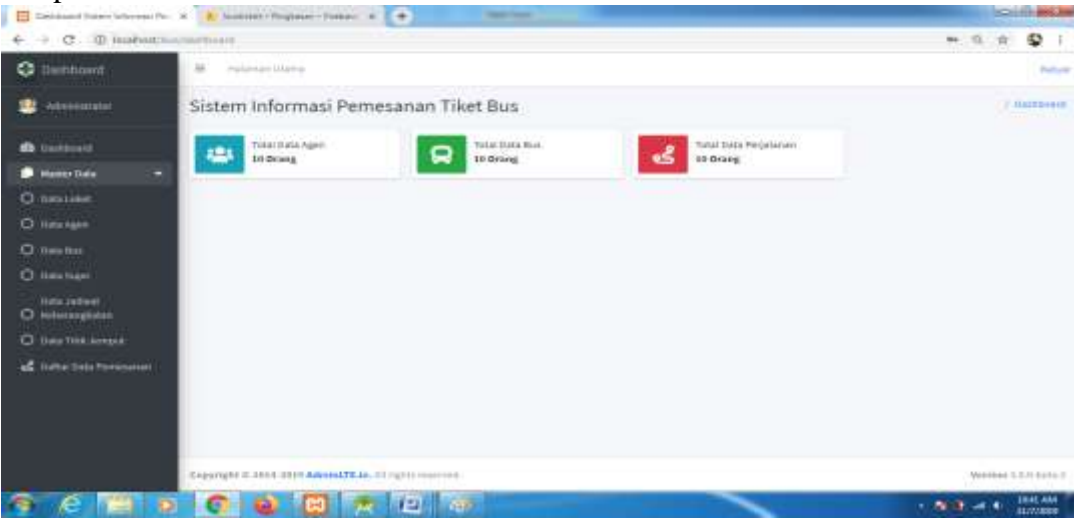
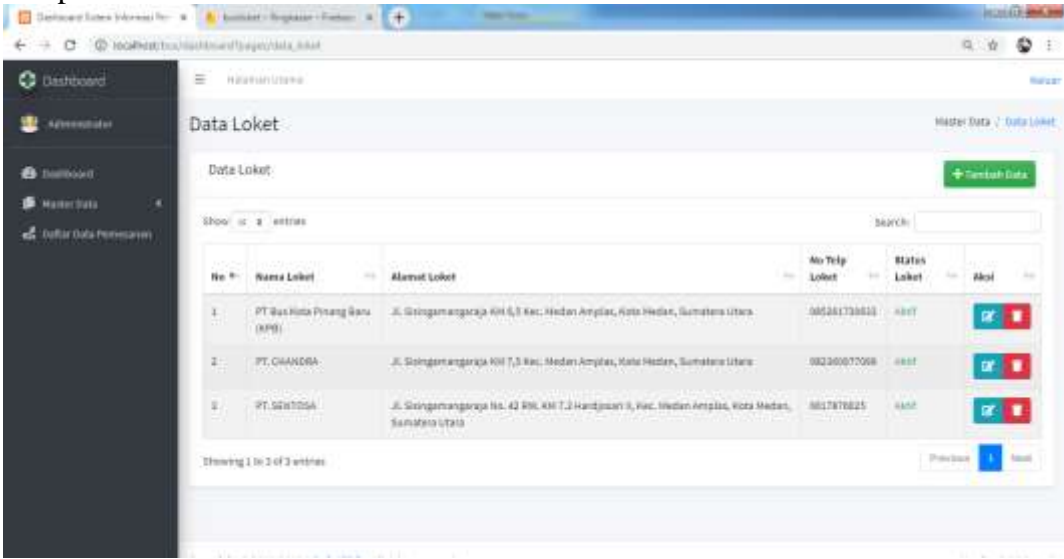
- a. Sistem Operasi *Windows 7 Ultimate* sebagai sistem operasi komputer.
- b. Microsoft Visio sebagai alat bantu untuk mendesain tampilan aplikasi dan sebagai alat desain perancangan diagram.
- c. Editor Android Studio sebagai alat bantu untuk mengkodekan aplikasi berbasis android.
- d. *Java Development Kit* (JDK) untuk mendukung penggunaan Android Studio.
- e. Notepad ++ sebagai alat bantu untuk mengkodekan *web service*.
- f. XAMPP (MySQL sebagai *database server*, *PhpMyAdmin* sebagai *webserver*).
- g. *Browser internet chrome* sebagai *web browser* untuk menjalankan *webservice*.
- h. *Handphone* digunakan untuk menjalankan aplikasi android.

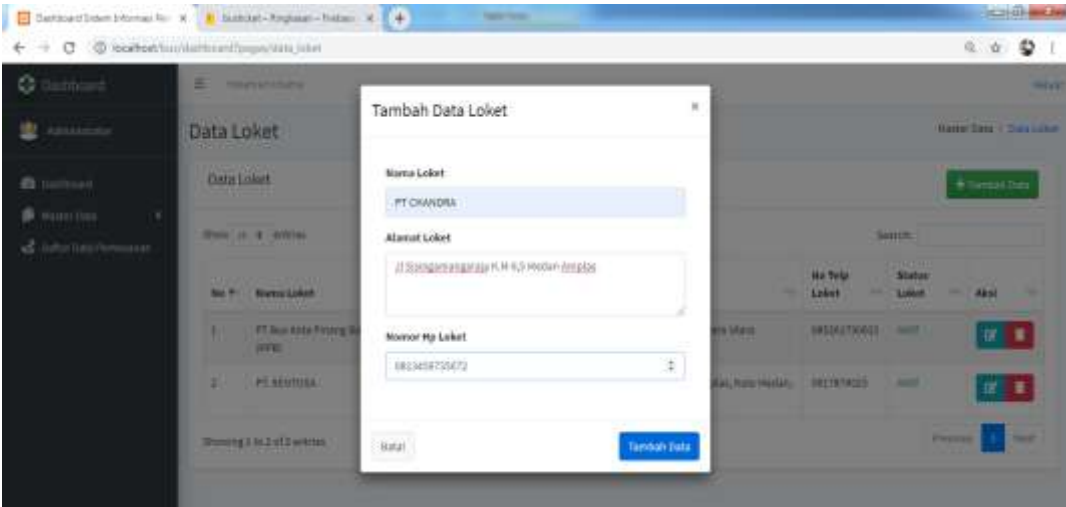
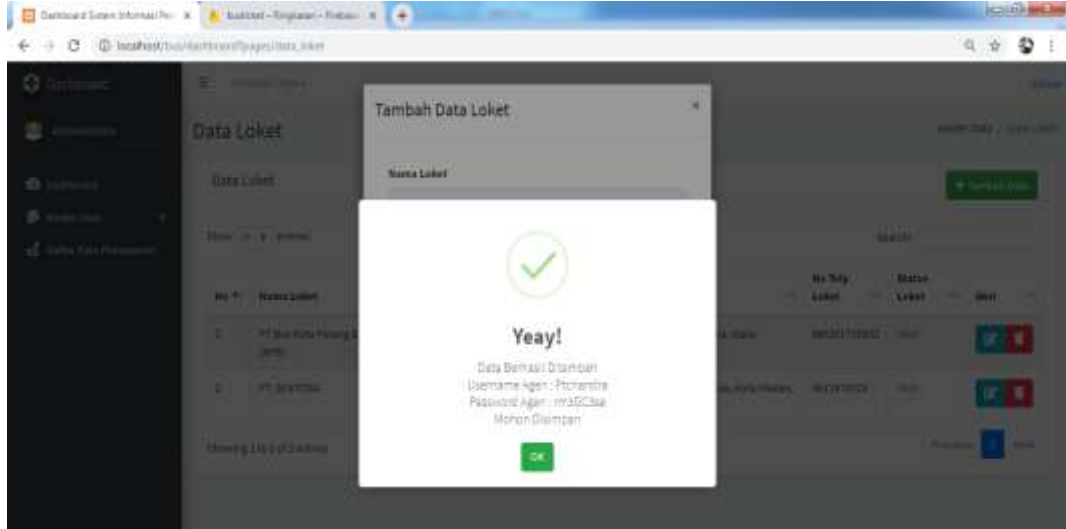
4.3.2 Pengujian Sistem

Pengujian sistem yang digunakan dalam penelitian ini menggunakan metode *black-box testing*. Metode ini merupakan metode pengujian dengan melakukan login pada sistem dan dapat melihat apakah masing-masing dari fungsi yang ada pada sistem sudah beroperasi dengan baik sesuai analisis kebutuhan dan desain atau tidak. Berikut adalah tabel pegujian yang akan dilakukan pada admin, agen, client dan supir yaitu sebagai berikut.

Tabel 4.14 Tabel Pengujian Metode Black-Box Admin dan Agen

| No. | Rancangan Input/Ouput | Hasil yang diharapkan | Hasil |
|-----|---|------------------------------|-------|
| 1. | Buka aplikasi | Masuk kehalaman login | Ok |
| 2. | Tampilan aplikasi:  <p style="text-align: center;">Gambar 4.32 Interface Login Admin dan Agen</p> | | |
| 3 | Klik Masuk | Masuk kehalaman utama | Ok |
| 4. | Tampilan aplikasi :  <p style="text-align: center;">Tabel 4.33 Tampilan Halaman Utama</p> | | |
| 5. | Klik menu master data | Tampil kehalaman master data | OK |

| | | | |
|-----------|---|---|-----------|
| <p>6.</p> | <p>Tampilan master data :</p>  <p>Tabel 4.34 Tampilan Master Data Admin</p> | | |
| <p>7.</p> | <p>Klik → Data loket</p> | <p>Menampilkan <i>form</i> data loket</p> | <p>OK</p> |
| <p>8.</p> | <p>Tampilan data loket :</p>  <p>Gambar 4.35 Tampilan Data Loket</p> | | |
| <p>9.</p> | <p>Klik → Tambah loket</p> | <p>Menampilkan halaman tambah loket</p> | <p>OK</p> |

| | | | |
|------------|--|---|-----------|
| <p>10.</p> | <p>Tampilan tambah loket :</p>  <p>Gambar 4.36 Tampilan Tambah Loket</p> | | |
| <p>11.</p> | <p>Klik → Tambah data</p> | <p>Menampilkan data <i>username</i> dan <i>password</i> untuk agen mengakses sistem</p> | <p>OK</p> |
| <p>12.</p> | <p>Tampilan tambah data :</p>  <p>Gambar 4.37 Tampilan Mendapat <i>Username</i> dan <i>Password</i> Agen</p> | | |
| <p>13.</p> | <p>Klik Oke</p> | <p>Menampilkan hasil dari rekaman input</p> | <p>OK</p> |

14. Tampilan ok :

Gambar4.38 Tampilan Hasil *Input*Data Loket

| | | | |
|-----|----------------|-----------------------|----|
| 15. | Klik data agen | Menampilkan data agen | OK |
|-----|----------------|-----------------------|----|

16. Tampilan Data agen :

Gambar 4.39 *Interface* Data Agen

| | | | |
|-----|--------------------|------------------------------|----|
| 17. | Klik → Tambah Data | Menampilkan tambah data agen | OK |
|-----|--------------------|------------------------------|----|

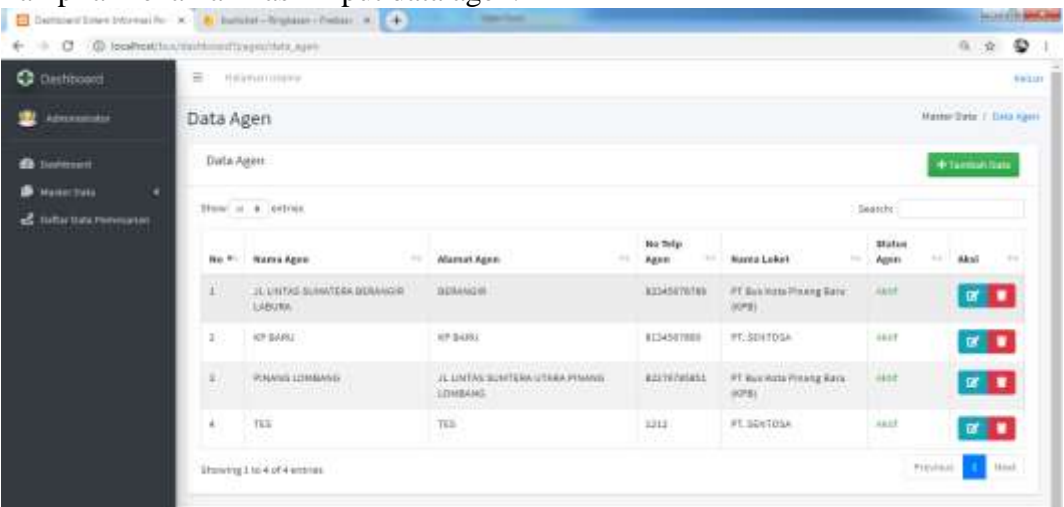
18. **Tampilan Tambah Data Agen**



Gambar 4.40 Tampilan Tambah Data Agen

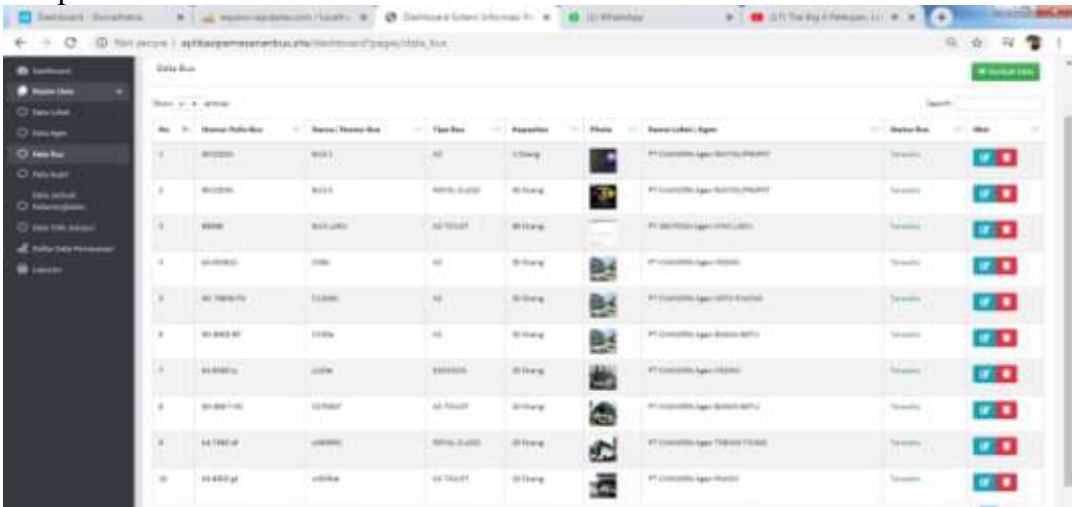
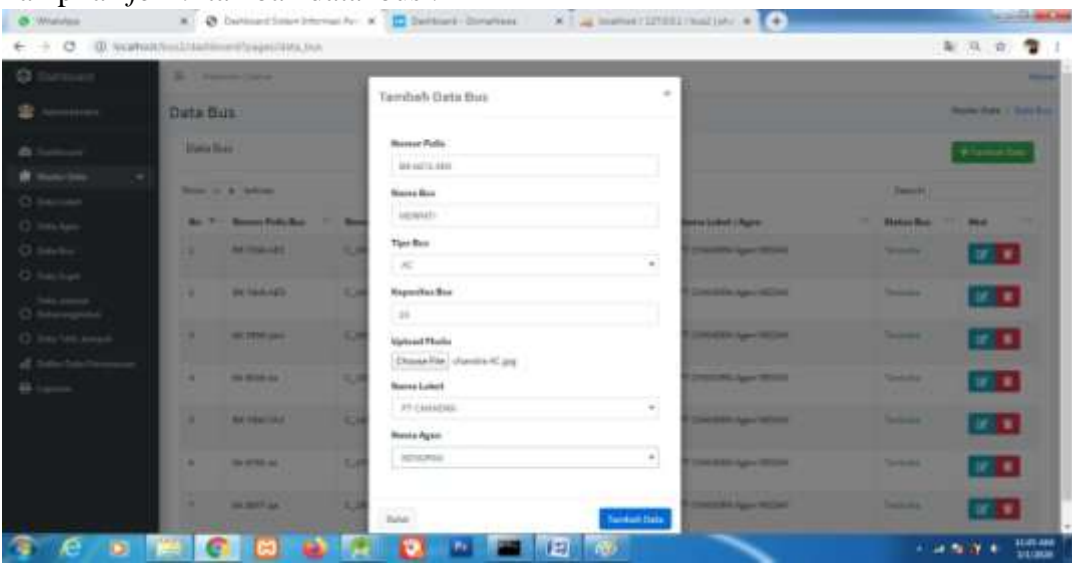
| | | | |
|-----|--------------------|---|----|
| 19. | Klik → Tambah Agen | Menampilkan hasil rekaman <i>input</i> tambah data agen | OK |
|-----|--------------------|---|----|

20. **Tampilan rekaman hasil input data agen:**

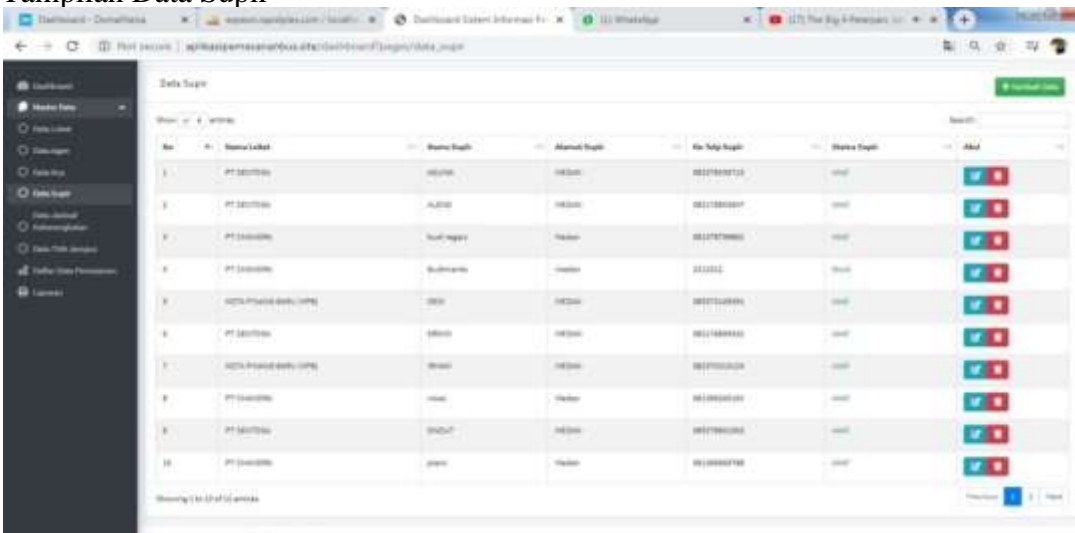


Gambar 4.41 Tampilan Data Agen Setelah ditambah

| | | | |
|-----|-----------------|----------------------|----|
| 21. | Klik → Data Bus | Menampilkan Data Bus | OK |
|-----|-----------------|----------------------|----|

| | | | |
|------------|---|------------------------------------|-----------|
| <p>22.</p> | <p>Tampilan Data Bus :</p>  <p style="text-align: center;">Gambar 4.42 Interface Data Bus</p> | | |
| <p>23.</p> | <p>Klik tambah data</p> | <p>Tampil form tambah data bus</p> | <p>OK</p> |
| <p>24.</p> | <p>Tampilan form tambah data bus :</p>  <p style="text-align: center;">Gambar 4.43 Tampilan Tambah Data Bus</p> | | |
| <p>25.</p> | <p>Klik → Data Supir</p> | <p>Menampilkan Data Supir</p> | <p>OK</p> |

26. **Tampilan Data Supir**



Gambar 4.44 Interface Data Supir

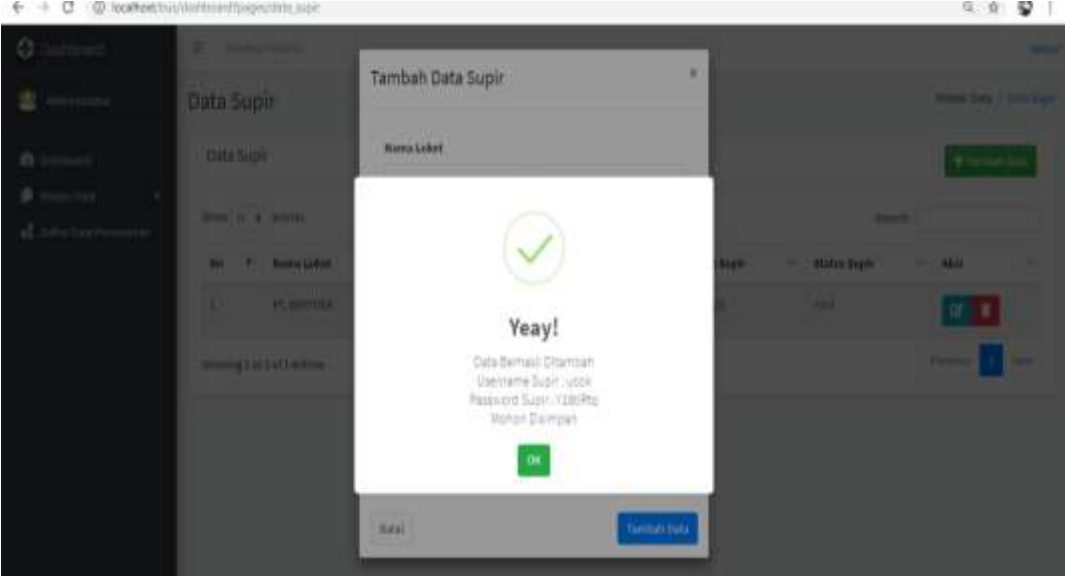
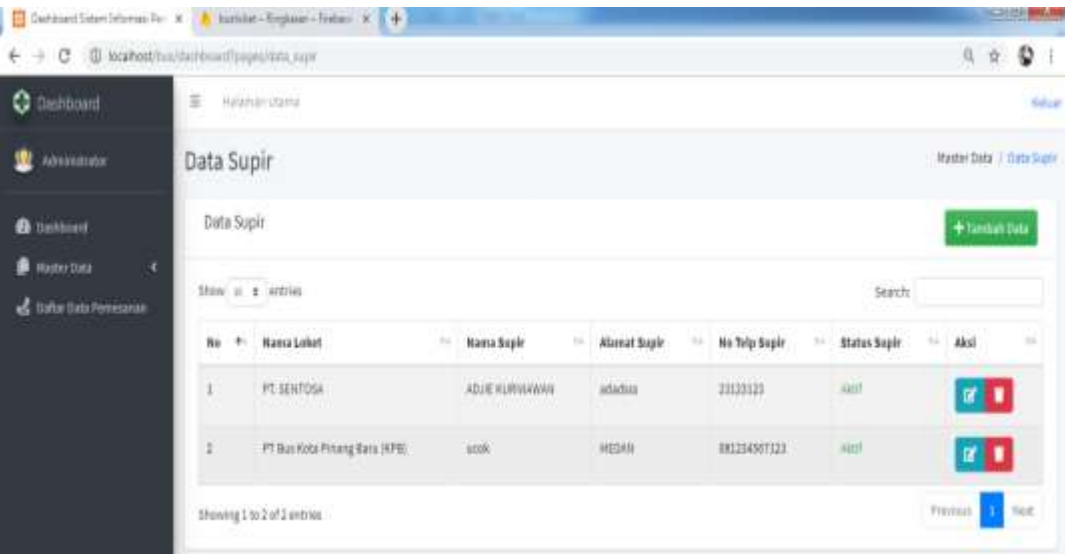
| | | | |
|-----|--------------------|--------------------------------------|----|
| 27. | Klik → Tambah Data | Tampil <i>form</i> tambah data supir | OK |
|-----|--------------------|--------------------------------------|----|

28. **Tampilan *Form* tambah supir :**

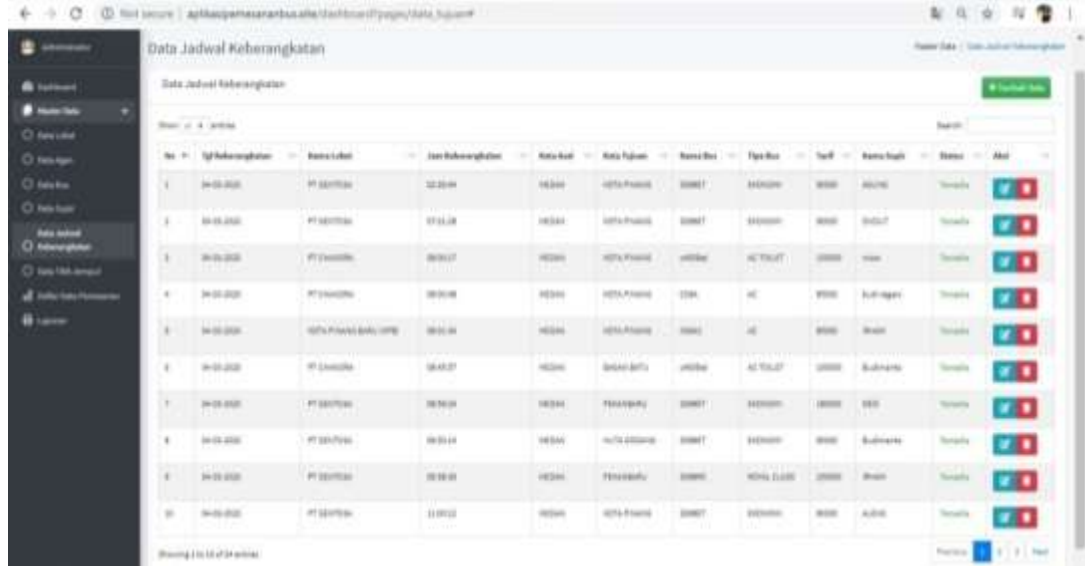


Gambar 4.45 Tampilan *Form* Tambah Data Supir

| | | | |
|-----|------------------|---|----|
| 29. | Klik Tambah data | Menampilkan <i>username</i> dan <i>password</i> supir | OK |
|-----|------------------|---|----|

| | | | |
|------------|--|---|-----------|
| <p>30.</p> | <p>Tampilan OK :</p>  <p>Gambar 4.46 Tampilan Mendapat <i>Password</i> dan <i>Username</i> Supir</p> | | |
| <p>31.</p> | <p>Klik OK</p> | <p>Menampilkan hasil rekaman data supir yang telah ditambah</p> | <p>OK</p> |
| <p>32.</p> | <p>Tampilan data supir :</p>  <p>Gambar 4.47 Tampilan Data Supir Yang Telah Ditambah</p> | | |
| <p>33.</p> | <p>Klik Data jadwal Keberangkatan</p> | <p>Menampilkan jadwal keberangkatan</p> | <p>OK</p> |

34. Tampilan data jadwal keberangkatan :

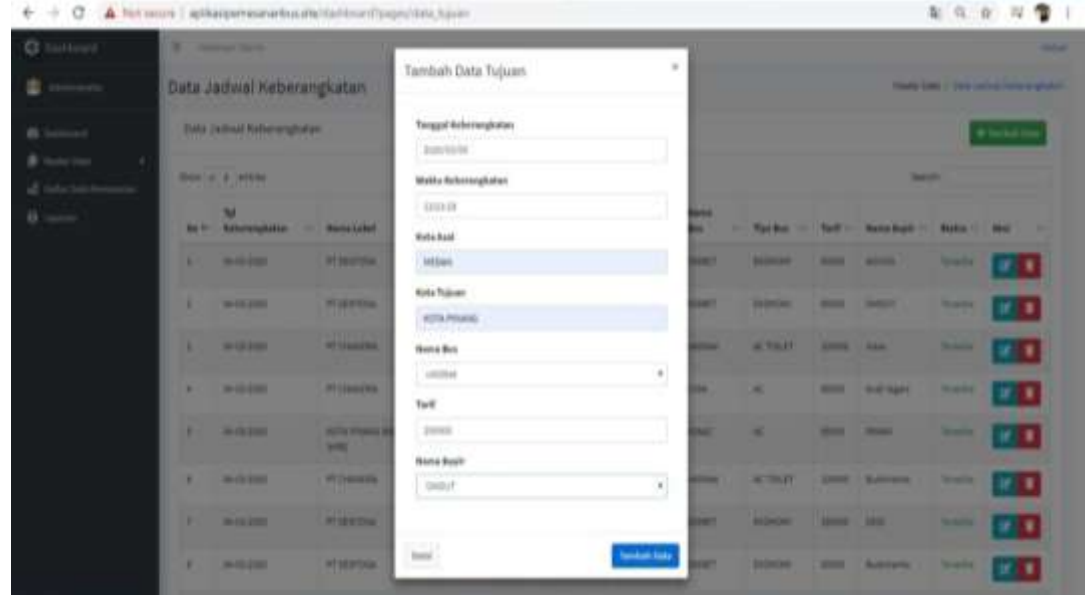


| No | Id Keberangkatan | Nama Label | Jam Keberangkatan | Kota Asal | Kota Tujuan | Nama Bus | Tipe Bus | Tarif | Nama Buspi | Status | Aksi |
|----|------------------|-------------------------|-------------------|-----------|--------------|----------|-----------|-------|------------|-----------|----------------|
| 1 | 34-03-2023 | PT 123456789 | 22:20:04 | 48244 | 4874 Pinaris | 20007 | 4820004 | 4000 | 48200 | Terselesa | [Edit] [Hapus] |
| 2 | 34-03-2023 | PT 123456789 | 17:11:28 | 48244 | 4874 Pinaris | 20007 | 4820004 | 4000 | 48200 | Terselesa | [Edit] [Hapus] |
| 3 | 34-03-2023 | PT 123456789 | 09:00:17 | 48244 | 4874 Pinaris | 48004 | 4874007 | 20000 | 48000 | Terselesa | [Edit] [Hapus] |
| 4 | 34-03-2023 | PT 123456789 | 09:00:08 | 48244 | 4874 Pinaris | 2204 | 48 | 4000 | 48000 | Terselesa | [Edit] [Hapus] |
| 5 | 34-03-2023 | 4874 Pinaris Baku (SPE) | 09:00:04 | 48244 | 4874 Pinaris | 20007 | 48 | 4000 | 48000 | Terselesa | [Edit] [Hapus] |
| 6 | 34-03-2023 | PT 123456789 | 08:45:17 | 48244 | 4874 Baku | 48004 | 4874007 | 20000 | 48000 | Terselesa | [Edit] [Hapus] |
| 7 | 34-03-2023 | PT 123456789 | 08:30:04 | 48244 | Pinaris Baku | 20007 | 4820004 | 4000 | 48000 | Terselesa | [Edit] [Hapus] |
| 8 | 34-03-2023 | PT 123456789 | 08:30:14 | 48244 | 4874 Baku | 20007 | 4820004 | 4000 | 48000 | Terselesa | [Edit] [Hapus] |
| 9 | 34-03-2023 | PT 123456789 | 08:30:03 | 48244 | Pinaris Baku | 20007 | 4874 Baku | 20000 | 48000 | Terselesa | [Edit] [Hapus] |
| 10 | 34-03-2023 | PT 123456789 | 11:00:12 | 48244 | 4874 Pinaris | 20007 | 4820004 | 4000 | 48000 | Terselesa | [Edit] [Hapus] |

Gambar 4.48 Interface Data Jadwal Keberangkatan

| | | | |
|-----|-------------|--|----|
| 35. | Klik Tambah | Menampilkan Form tambah data keberangkatan | OK |
|-----|-------------|--|----|

36. Tampilan aplikasi :



Gambar 4.49 Form Tambah Data Jadwal Keberangkatan

| | | | |
|-----|------------------|--|----|
| 37. | Klik Tambah Data | Menampilkan hasil rekaman tambah data jadwal keberangkatan | OK |
|-----|------------------|--|----|

38. Tampilan setelah ditambah:

| No | Tgl Keberangkatan | Nama Loker | Jam Keberangkatan | Kota Asal | Kota Tujuan | Nama Bus | Tipe Bus | Tarif | Nama Supir | Status | Aksi |
|----|-------------------|---------------------|-------------------|-----------|-------------|----------|----------|-------|------------|----------|----------------|
| 14 | 25.05.2022 | PT DAMARA | 07:00:00 | MDM | PTAKA | 1401A | AC | 20000 | 10000 | Tersedia | [Edit] [Hapus] |
| 15 | 25.05.2022 | PT DAMARA | 22:00:00 | MDM | PTAKA | 1401 | AC | 20000 | Bulkarya | Tersedia | [Edit] [Hapus] |
| 16 | 24.05.2022 | PT DAMARA | 20:30:00 | MDM | MDM | 1401A | AC | 20000 | 10000 | Tersedia | [Edit] [Hapus] |
| 17 | 25.05.2022 | PT DAMARA | 20:25:00 | MDM | MDM | 1401 | AC | 20000 | Bulkarya | Tersedia | [Edit] [Hapus] |
| 18 | 25.05.2022 | PT DAMARA | 20:25:00 | MDM | MDM | 1401 | AC | 20000 | Bulkarya | Tersedia | [Edit] [Hapus] |
| 19 | 24.05.2022 | MDM PANG BARU (PFB) | 20:00:00 | MDM | MDM | 1401 | AC | 20000 | 10000 | Tersedia | [Edit] [Hapus] |
| 20 | 24.05.2022 | MDM PANG BARU (PFB) | 20:00:00 | MDM | MDM | 1401 | AC | 20000 | Bulkarya | Tersedia | [Edit] [Hapus] |
| 21 | 24.05.2022 | MDM PANG BARU (PFB) | 20:00:00 | MDM | MDM | 1401 | AC | 20000 | 10000 | Tersedia | [Edit] [Hapus] |
| 22 | 24.05.2022 | MDM PANG BARU (PFB) | 20:00:00 | MDM | MDM | 1401 | AC | 20000 | Bulkarya | Tersedia | [Edit] [Hapus] |
| 23 | 24.05.2022 | PT DAMARA | 20:00:00 | MDM | MDM | 1401 | AC | 20000 | Bulkarya | Tersedia | [Edit] [Hapus] |
| 24 | 24.05.2022 | MDM PANG BARU (PFB) | 21:00:00 | MDM | SADA BATU | 1401 | AC | 20000 | 10000 | Tersedia | [Edit] [Hapus] |

Gambar 4.50 Tampilan Setelah Sudah Menambahkan Data Jadwal Berangkat

| | | | |
|-----|------------------------|-------------------------------|----|
| 39. | Klik data titik jemput | Menampilkan data titik jemput | OK |
|-----|------------------------|-------------------------------|----|

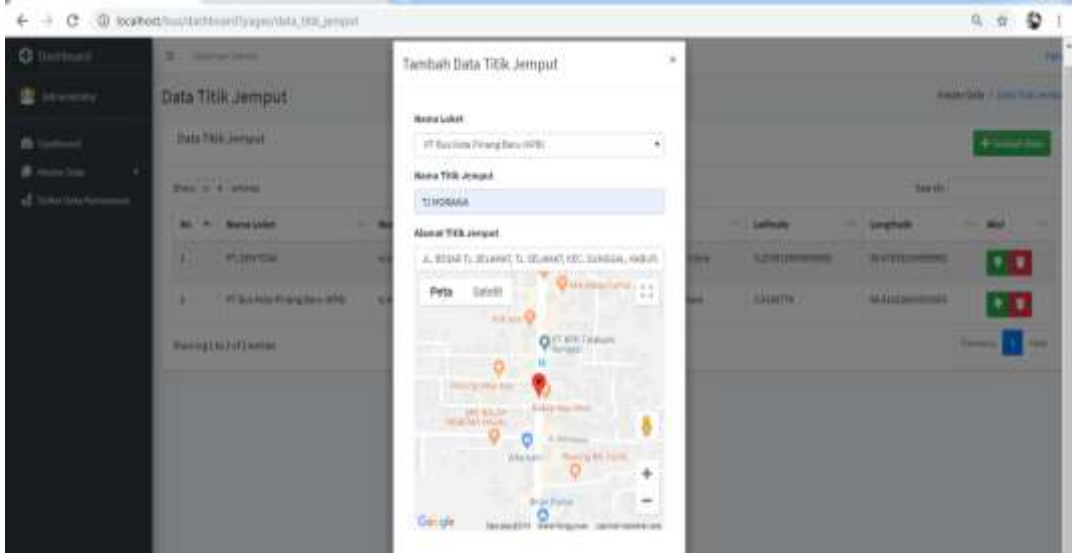
40. Tampilan data titik jemput

| No | Nama Loker | Nama Titik Jemput | Alamat Titik Jemput | Latitude | Longitude | Aksi |
|----|-------------------------------|-------------------|--|-------------------|--------------------|----------------|
| 1 | PT. SENTOSA | sumber pad | Kec. Limaukekoh, Kabupaten Batu Bara, Sumatera Utara | 3.233819599999999 | 101.47638159999999 | [Edit] [Hapus] |
| 2 | PT Bus Kota Pinang Baru (KPB) | ij marawa | Tj. Marawa, Kabupaten Deli Serdang, Sumatera Utara | 3.5330778 | 101.81422046000005 | [Edit] [Hapus] |

Gambar 4.51 Interface Data Titik Jemput

| | | | |
|-----|------------------|------------------------------------|----|
| 41. | Klik tambah data | Menampilkan form data titik jemput | OK |
|-----|------------------|------------------------------------|----|

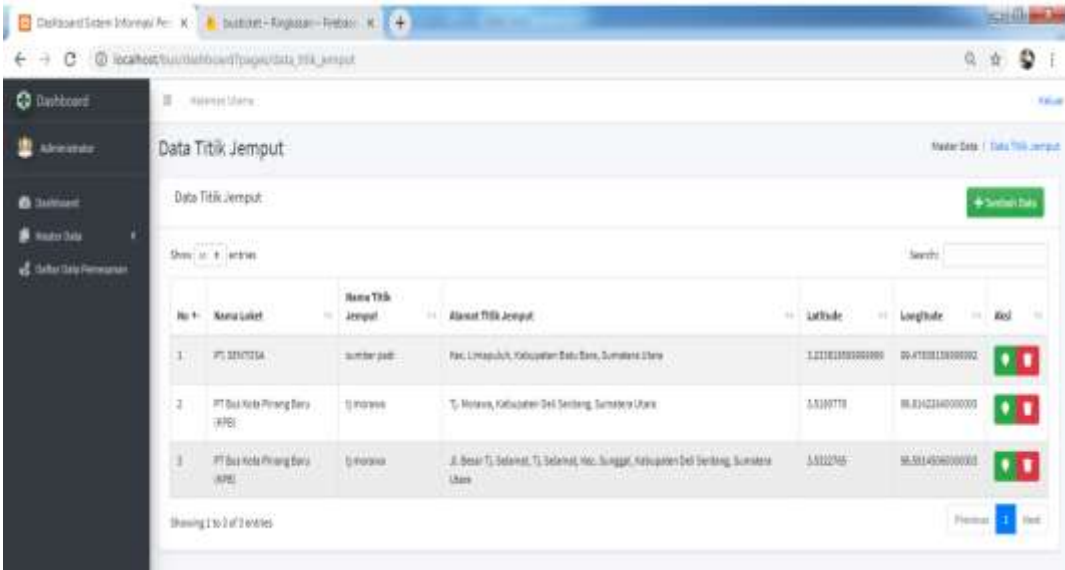
42. Menampilkan form data titik jemput:



Gambar 4.52 Tampilan Form Tambah Data Titik Jemput

| | | | |
|-----|------------------|---------------------------|----|
| 43. | Klik tambah data | Tampil hasil rekaman data | OK |
|-----|------------------|---------------------------|----|

44. Tampilan tambah data:



Gambar 4.53 Tampilan Setelah Tambah Data Titik Jemput

| | | | |
|-----|-------------------|----------------------------------|----|
| 45. | Klik lihat lokasi | Menampilkan lihat lokasi di peta | OK |
|-----|-------------------|----------------------------------|----|

46. Tampilan lokasi :

Gambar 4.54 Tampilan Lihat Lokasi Titik Jemput

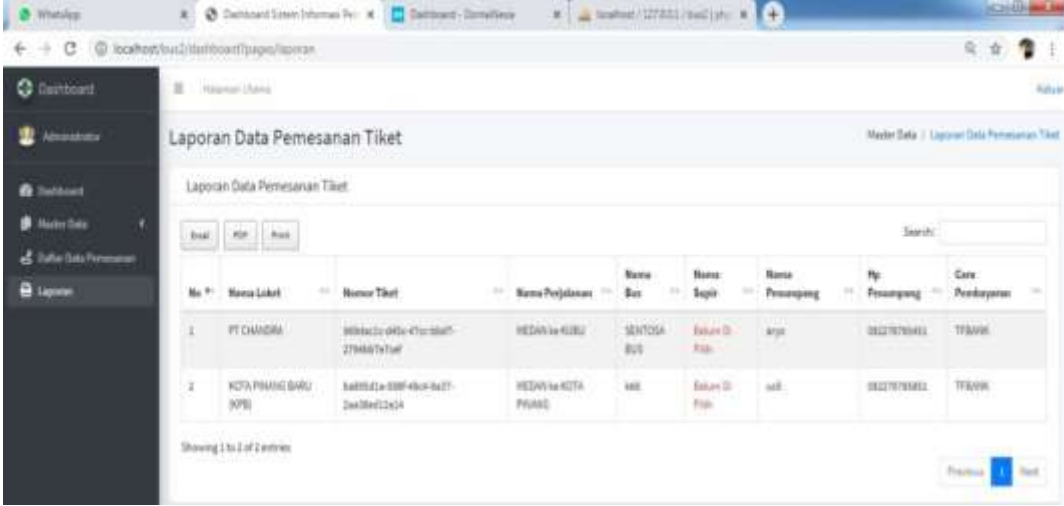

| | | | |
|-----|----------------------------|----------------------------|----|
| 47. | Klik daftar data pemesanan | Menampilkan data pemesanan | OK |
|-----|----------------------------|----------------------------|----|

48. Tampilan data pemesanan:


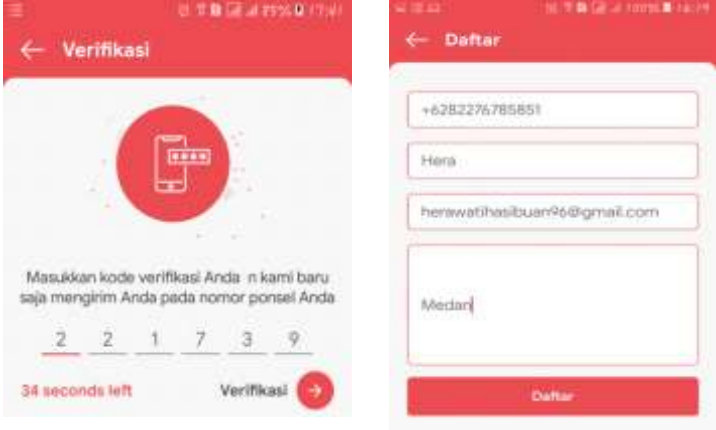
| No. Tiket | Nama Perjalanan | Nama Bus | Nama Penumpang | Nomor Hp Penumpang | Cara Pembayaran | Status | Aksi |
|--------------------------------------|-------------------------|--------------|----------------|--------------------|-----------------|---------------------|---|
| 093556e7-3605-4c50-a026-0e40313d3be0 | MEDAN ke BANTAL/PINAKIT | Sintang | hera | 082276188851 | TFBANK | Menunggu Pembayaran | ✔ ✖ + |
| fd028a90-ae08-4200-a327-81f542309fc3 | MEDAN ke PEKAN BARU | SIMPATI STAR | hawan | 082168724001 | TFBANK | Menunggu Pembayaran | ✔ ✖ + |



Gambar 4.55 Tampilan Daftar Data Pemesanan



| | | | |
|-----|--------------|--------------------------|----|
| 49. | Klik Laporan | Menampilkan data laporan | OK |
|-----|--------------|--------------------------|----|

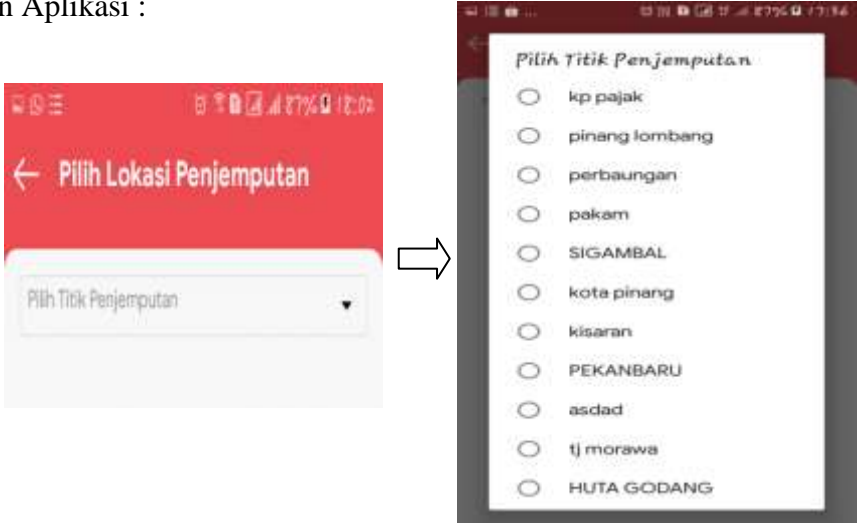
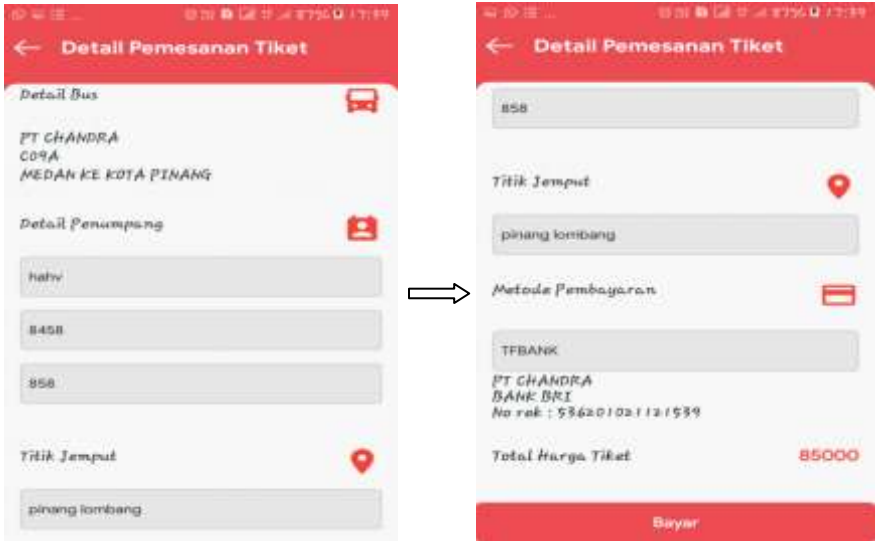
| | | | |
|--|-------------------|------------------------|----|
| 50. | Tampilan laporan: | | |
|  | | | |
| Gambar4.56 Tampilan Laporan Data Pemesanan Tiket Bus | | | |
| 51. | Klik Print | Menampilkan Form Print | OK |
| 52. | Tampilan Print: | | |
|  | | | |
| Gambar 4.57 Tampilan Print Laporan Pemesanan | | | |



Tabel 4.15 Tabel Pengujian Metode Black Box *Client*

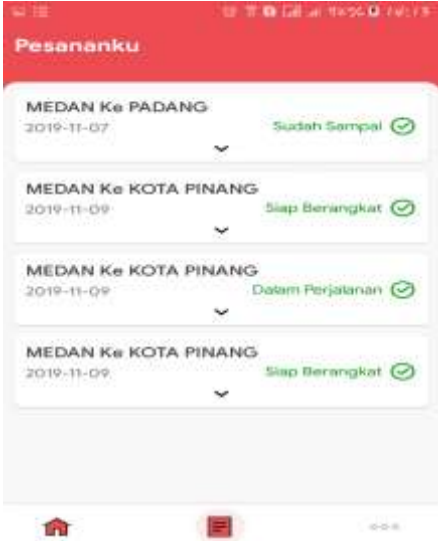
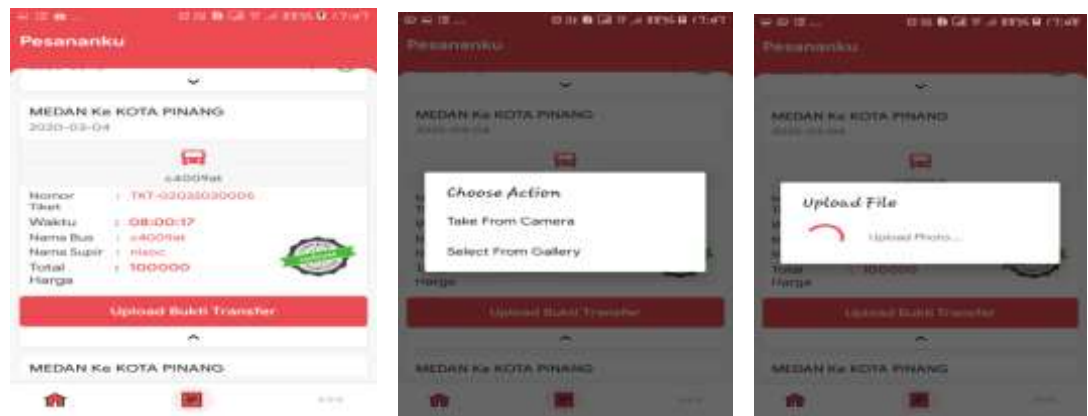
| No. | Rancangan Input/Output | Hasil yang diharapkan | Hasil |
|-----|---|------------------------------------|-------|
| 1. | Membuka aplikasi | Masuk kehalaman login | Ok |
| 2. | Tampilan aplikasi :  <p style="text-align: center;">\</p> Gambar 4.58 Tampilan Login Untuk Menggunakan Aplikasi Android | | |
| 3. | Klik lanjut | Menampilkan <i>form verifikasi</i> | OK |
| 4. | Tampilan aplikasi :  <p style="text-align: center;">\</p> Gambar 4.59 Tampilan Menu Verifikasi dan Daftar Pengguna Baru | | |
| 5. | Klik Verifikasi | Masuk ke halaman dashboard | OK |


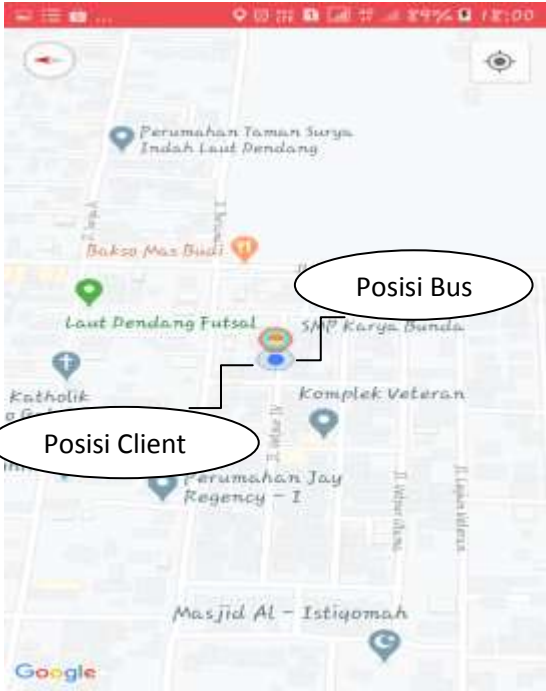
| | | | |
|----|-----------------------------------|--|----|
| 6. | Tampilan aplikasi : |  <p style="text-align: center;">Gambar 4.60 Tampilan Dashboard</p> | |
| 7. | Klik logo search | Masuk ke halaman bus tersedia | OK |
| 8. | Tampilan Aplikasi : |  <p style="text-align: center;">Gambar 4.61 Tampilan Bus yang Tersedia</p> | |
| 9. | Klik salah satu bus yang tersedia | Masuk ke input data penumpang | OK |

| | | | |
|-----|---------------------------------|---|----|
| 10. | Tampilan Aplikasi: |  <p style="text-align: center;">Gambar 4.62 Tampilan Input Data Calon Penumpang</p> | |
| 11. | Klik lanjut pembayaran | Masuk ke halaman tipe pembayaran | OK |
| 12. | Tampilan Aplikasi : |  <p style="text-align: center;">Gambar 4.63 Tampilan Pilihan Tipe Pembayaran</p> | |
| 13. | Klik salah satu tipe pembayaran | Masuk ke halaman lokasi naik | OK |




| | | | |
|-----|--|---|----|
| 14. | <p>Tampilan Aplikasi :</p>  <p>Gambar 4.64 Tampilan Pilih Titik Penjemputan</p> | | |
| 15. | Klik salah satu titik penjemputan | Masuk ke halaman detail pemesanan tiket | OK |
| 16. | <p>Tampilan Aplikasi :</p>  <p>Gambar 4.65 Tampilan Detail Pemesanan Tiket</p> | | |
| 17. | Klik Bayar | Menampilkan peringatan berhasil | OK |

| | | | |
|-----|---------------------|---|----|
| 18. | Tampilan Aplikasi : |  <p style="text-align: center;">Gambar 4.66 Tampilan Berhasil Melakukan Pemesanan Tiket Bus</p> | |
| 19. | Klik OK | Kembali ke halaman dashboard | OK |
| 20. | Tampilan Aplikasi: |  <p style="text-align: center;">Gambar 4.67 Tampilan Menu Dashboard</p> | |
| 21. | Klik Menu Pesananku | Menampilkan bus yang dipesan | OK |

| | | | |
|------------|---|--|-----------|
| <p>22.</p> | <p>Tampilan Aplikasi :</p> |  <p>Gambar 4.68 Tampilan Pesananku</p> | |
| <p>23.</p> | <p>Klik siap berangkat / sesuaikan tanggal keberangkatan yang dipesan</p> | <p>Upload bukti transfer</p> | <p>OK</p> |
| <p>24.</p> | <p>Tampilan Aplikasi :</p>  <p>Gambar 4.69 Tampilan Upload Bukti Transfer</p> | | |
| <p>25.</p> | <p>Setelah Sudah Terkonfirmasi Pembayaran oleh admin klik pesananku</p> | <p>Tampil Pesananku</p> | <p>OK</p> |

| | | | |
|-----|---------------------|--|----|
| 26. | Tampilan Aplikasi : |  <p data-bbox="379 770 1460 808">Gambar 4.70 Tampilan Pesananaku Setelah Terkonfirmasi Pembayaran Oleh Admin</p> | |
| 27. | Klik lihat maps bus | Tampil peta dan posisi bus | OK |
| 28. | Tampilan Aplikasi: |  <p data-bbox="627 1715 1219 1753">Gambar 4.71 Tampilan Posisi Bus dan Client</p> | |

Tabel 4.16 Tabel Pengujian Metode Black Box Supir

| No. | Rancangan Input/Ouput | Hasil yang diharapkan | Hasil |
|-----|--|----------------------------------|-------|
| 1. | Buka aplikasi(supir) | Masuk kehalaman login | Ok |
| 2. | Tampilan Aplikasi : <div style="text-align: center;">  <p>Gambar 4.72 Halaman Login Supir</p> </div> | | |
| 3. | Klik list keberangkatan | Tampil detail list keberangkatan | OK |
| 4. | Tampilan aplikasi: <div style="display: flex; justify-content: space-around; align-items: center;">   </div> <p>Gambar 4.73 Tampilan Halaman Daftar Keberangkatan Supir</p> | | |

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil akhir dari pemecahan masalah pada penelitian ini maka penulis menguraikan beberapa kesimpulan-kesimpulan. Adapun kesimpulan tersebut antara lain adalah sebagai berikut :

1. Aplikasi ini bisa membantu calon penumpang untuk dapat mengetahui informasi lebih cepat mengenai jadwal keberangkatan dan harga tiket dari posisi *user* dengan memanfaatkan GPS.
2. Melalui aplikasi ini calon penumpang tidak perlu khawatir akan tertinggalnya bus ketika posisi naik penumpang tidak dari titik awal bus berangkat.
3. Aplikasi yang dibangun menggunakan Android Studio dan memanfaatkan GPS untuk mengakses Location Based Service.
4. Keakuratan posisi *user* dapat meleset dari posisi sebenarnya dikarenakan penentuan posisi tergantung oleh GPS di hp pengguna.

5.2 Saran

Berdasarkan hasil penelitian ini penulis menguraikan beberapa saran bagi mahasiswa, pembaca dan pengguna. Adapun saran-saran yang dapat penulis berikan antara lain :

1. Perlu adanya pengembangan aplikasi sehingga aplikasi yang dibangun untuk perjalanan bus AKDP (Antar Kota Dalam Propinsi) lebih lengkap seperti penambahan fitur dari segi pembayaran.
2. Karena aplikasi ini *online* dan bergantung pada koneksi internet maka perlu adanya penelitian lanjut mengenai pembuatan aplikasi secara *offline* dengan menggunakan *database* di Android.

DAFTAR PUSTAKA

- Alfeno, S., & Devi, R. E. C. (2017). Implementasi Global Positioning System (GPS) dan Location Based Service (LSB) pada Sistem Informasi Kereta Api untuk Wilayah Jabodetabek. *Sisfotek Global*, 7(2), 27–33.
- Anwar, B., Jaya, H., Kusuma, P. I., Studi, P., & Komputer, S. (2014). *Issn : 1978-6603 implementasi location based service berbasis android untuk mengetahui posisi user*. 121–133.
- AS M. Shalahudin, R. (2014). *Rekayasa Perangkat Lunak*. BANDUNG: INFORMATIKA.
- Feri, S. (2017). *Teknik Perancangan Arsitektur Sistem Informasi*. Yogyakarta: Andi Offset.
- Huda, A. A. (2014). *Live Coding 9 Aplikasi Android Buatan Sendiri*. Yogyakarta: Andi Publisher.
- Irawan, M. D., & Simargolang, S. A. (2018). Implementasi E-Arsip pada Program Studi Teknik Informatika. *Jurnal Teknologi Informasi*, 2, 67–68.
- Irwan Padli Nasution, M., & Samsudin, S. (2018). Using google location APIs to find an accurate criminal accident location. *International Journal of Engineering & Technology*, 7(3), 1818. <https://doi.org/10.14419/ijet.v7i3.14144>
- Juhara, Z. P. (2016). *Panduan Lengkap Pemrograman Android*. Yogyakarta: Andi Offset.
- Kadir, A. (2014). *Pemrograman Aplikasi Android*. Yogyakarta: Andi Offset.
- Kasman Dharma, A. (2015). *Bikin Aplikasi Pemesanan Tiket Online Berbasis Web & Android (PERTAMA)*. JAWA BARAT: ASFA SOLUTIONS.
- Komputer, W. (n.d.). *Microsoft Visio Untuk Desain Diagram dan Flowchart*. Semarang.
- Komputer, W. (2013). *Step By Step Menjadi Programmer Android*. Yogyakarta: Andi Offset.
- Kostaman, N., & Sumaryana, Y. (2018). Aplikasi pemesanan tiket oto bus budiman berbasis online. *Jumantaka*, 1(1), 121–130.

- Marakas, G. (2006). *System Analysis & Design an active approach*. NEWYORK: McGraw-Hill.
- Nugroho, S. C., Nurhayati, O. D., & Widiyanto, E. D. (2015). Aplikasi Pencarian Rute Perguruan Tinggi Berbasis Android Menggunakan Location Based Service (LBS) di Kota Semarang. *Jurnal Teknologi Dan Sistem Komputer*, 3(2), 311. <https://doi.org/10.14710/jtsiskom.3.2.2015.311-319>
- Paramanindo, F. (2014). *APLIKASI SIMULASI UJIAN NASIONAL PADA SMA NEGERI 1 RAMBANG DANGKU BERBASIS DESKTOP*. (061130801319).
- Rahmat, A. R. A., & Octaviano, A. (2016). Aplikasi Pemesanan Tiket Bus Berbasis Web Pada Po . Harapan Jaya. *Jurnal Informatika Universitas Pamulang*, 1(1), 1–11.
- Rompas, B. . (2015). APLIKASI LOCATION BASED SERVICE PENCARIAN TEMPAT DI KOTA MANADO BERBASIS ANDROID. *JAWRA Journal of the American Water Resources Association*, 18(5), 901–903. <https://doi.org/10.1111/j.1752-1688.1982.tb00091.x>
- Suryana, D. (2018). *Android Studio: Belajar Android Studio*. BANDUNG.
- Tim, E. (2013). *Android All In One*. Jakarta: Elex Media Komputindo.
- Wicaksono Rizky, S. (2017). *Konsep Dasar Rekayasa Perangkat Lunak*. Jakarta: Prestasi Pustakarya.

LAMPIRAN 1

Kuisisioner Pengujian Aplikasi

Pertanyaan:

1. Sejauh mana aplikasi ini membantu anda dalam proses pemesanan tiket bus?
2. Apakah hasil yang ditampilkan sesuai dengan kebutuhan?
3. Sejauh mana anda memahami tampilan keseluruhan dari aplikasi ini?
4. Apakah aplikasi ini cukup mudah untuk digunakan?
5. Bagaimana pendapat anda mengenai waktu yang dibutuhkan aplikasi saat dijalankan?

Pedoman Dari Pertanyaan

| No | Petunjuk |
|----|---|
| 1. | A. Sangat membantu saya dalam hal pemesanan tiket bus. B. Cara yang mudah untuk mencari informasi tiket keberangkatan. C. Saya akan memikirkan kembali untuk memakainya. D. Tidak membantu sama sekali. |
| 2. | A. Saya sangat setuju dengan hasil yang diberikan sistem. B. Hasil yang saya harapkan muncul setelah mencoba aplikasi. C. Pemesanan tiket bus yang diberikan aplikasi cukup baik, namun belum tentu menjadi pilihan. D. Saya kurang menyukai proses pemesanan tiket bus yang diberikan oleh sistem |
| 3. | A. Tampilan aplikasi secara keseluruhan sangat baik. B. Desain dari sistem sangat baik, tetapi warnanya kurang tepat. C. Desain sistem cukup baik. |

| | |
|----|--|
| | D. Tampilan dan warna sistem kurang. |
| 4. | A. Saya dapat menggunakan aplikasi tanpa dibantu. B. Saya dapat menggunakan aplikasi dengan sedikit bantuan. C. Ada beberapa bagian kurang dimengerti. D. Ada bagian yang kurang saya mengerti. |
| 5. | A. Sangat cepat. B. Memuaskan. C. Cukup memenuhi kebutuhan. D. Memakan banyak waktu. |

Jawaban kuisioner

Ket.

A : Amat Baik

C : Cukup

B : Baik

D : Kurang

| No | Nama | Jawaban | | | | |
|----|---------------------------|---------|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| 1. | Mauliza | B | B | C | A | B |
| 2. | Ninda Andriani | B | B | A | B | B |
| 3. | Randi Pratama | A | C | A | B | A |
| 4. | Sella Yunita | B | C | A | B | B |
| 5. | Irna Safitri Panjaitan | A | A | A | B | A |

Catatan: Uji coba dilakukan pada mahasiswa UIN-SU

| No | Pertanyaan | Amat Baik | Baik | Cukup | Kurang |
|-------|----------------------|-----------|------|-------|--------|
| 1. | Proses pemesanan | 1 | 3 | 0 | 0 |
| 2. | Hasil tampilan | 1 | 2 | 2 | 0 |
| 3. | Tampilan keseluruhan | 4 | 0 | 1 | 0 |
| 4. | Kemudahan aplikasi | 1 | 4 | 0 | 0 |
| 5. | Proses waktu | 2 | 3 | 0 | 0 |
| TOTAL | | 9 | 13 | 3 | 0 |

Pkurang = $(0/100) \times 25\%$ = 0%

Pcukup = $(3/100) \times 50\%$ = 1.5%

Pbaik = $(13/100) \times 75\%$ = 9.75%

Pamat baik = $(9/100) \times 100\%$ = 9%

LAMPIRAN 2

Login.php

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-
Compatible" content="IE=edge">
  <title>ADMIN LOGIN - SISTEM
INFORMASI PEMESANAN
BUS</title>
<?php
  include ('../include/header.php');
  include ('../include/footer.php');
  include ('../function/koneksi.php');
  include ('../function/function.php');
  ?>
</head>
<body class="hold-transition login-
page" style="background-
image:url('admin/assets/img/Gambar.j
peg');height: 100%; background-
position: center;
background-repeat: no-repeat;
background-size: cover;">
<div class="login-box">
<div class="">
</div>
<!-- /.login-logo -->
<div class="card">
```

```
<div class="card-body login-card-
body">
<h1><center> LOGIN </center></h1>
<p class="login-box-msg">SISTEM
INFORMASI PEMESANAN TIKET
BUS</p>
<form method="post"
id="formlogin">
<div class="input-group mb-3">
<input type="text" id='emaillogin'
name='emaillogin' class="form-
control" placeholder="Email atau
Username">
  <div class="input-group-
append">
    <div class="input-group-text">
      <span class="fas fa-
envelope"></span>
    </div>
  </div>
</div>
<div class="input-group mb-3">
  <input type="password"
id='passwordlogin'
name='passwordlogin' class="form-
control" placeholder="Password">
  <div class="input-group-
append">
    <div class="input-group-text">
```

```

<span class="fas fa-lock"></span>
  </div>
</div>
</div>
<div class="row">
  <!-- /.col -->
  <div class="col-12">
    <button      type="submit"
id='btnlogin' class="btn btn-primary
btn-block btn-flat">MASUK</button>
  </div>
  <!-- /.col -->
</div>
</form>
</div>
<!-- /.login-card-body -->
</div>
</div>
<!-- /.login-box -->

<?php      include
('./include/footer.php');?>

</body>
</html>

Dashboard.php
<?php
<div class="content-header">
<div class="container-fluid">
  <div class="row mb-2">
    <div class="col-sm-6">
      <h1 class="m-0 text-dark">Sistem
Informasi Pemesanan Tiket Bus</h1>
    </div><!-- /.col -->
    <div class="col-sm-6">
      <ol class="breadcrumb float-sm-
right">
        <li class="breadcrumb-item"></li>
        <li class="breadcrumb-item
active"><a href='dashboard'>Dashboar
d</a></li>
      </ol>
    </div><!-- /.col -->
  </div><!-- /.row -->
</div><!-- /.container-fluid -->
</div>
<!-- /.content-header -->

<!-- Main content -->
<div class="content">
  <div class="container-fluid">
    <div class="row">
      <div class="col-md-3 col-sm-6
col-12">
        <div class="info-box">
          <span class="info-box-icon bg-
info"><i class="fas fa-
users"></i></span>
        <div class="info-box-content">

```

```

<span class="info-box-text">Total
Data Agen</span>
<span class="info-box-number">10
Orang</span>
</div>
</div>
</div>
<div class="col-md-3 col-sm-6 col-
12">
    <div class="info-box">
        <span class="info-box-
icon bg-success"><i class="fas fa-
bus"></i></span>
        <div class="info-box-
content">
            <span class="info-box-
text">Total Data Bus</span>
            <span class="info-box-
number">10 Orang</span>
        </div>
    </div>
</div>
<div class="col-md-3 col-sm-6
col-12">
    <div class="info-box">
        <span class="info-box-
icon bg-danger"><i class="fas fa-
route"></i></span>
        <div class="info-box-
content">

```

```

<span class="info-box-
text">Total Data Perjalanan</span>
<span class="info-box-
number">10 Orang</span>
</div>
</div>
</div>
</div>
<!-- /.row -->
</div>
<!-- /.container-fluid -->
</div>

```

SignInActivity.java

```

package com.hera.bus.activities;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Log;
import android.view.KeyEvent;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.view.inputmethod.EditorInfo;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

```



```

import
com.google.firebase.FirebaseExceptio
n;
import
com.google.firebase.auth.FirebaseAuth
;
import
com.google.firebase.auth.PhoneAuthC
redential;
import
com.google.firebase.auth.PhoneAuthPr
ovider;
import    com.hera.bus.BaseActivity;
import    com.hera.bus.R;

import  java.util.concurrent.TimeUnit;

public class SignInActivity extends
BaseActivity    implements
View.OnClickListener{

    /*variable        declaration*/
    private  Button    mBtnContinue;
    private          EditText
mEdMobileNumber;
    private  ImageView  mIvFacebook,
mIvGoogle;
    private  String  TAG = "Signin";
    private
PhoneAuthProvider.ForceResendingT
oken          mResendToken;
    private  FirebaseAuth  auth;
    private  String  verificationCode;

    @Override
    protected void  onCreate(Bundle
savedInstanceState)    {

super.onCreate(savedInstanceState);

requestWindowFeature(Window.FEA
TURE_NO_TITLE);

getWindow().setFlags(WindowManag
er.LayoutParams.FLAG_FULLSCREE
N,
WindowManager.LayoutParams.FLA
G_FULLSCREEN);

setContentView(R.layout.activity_sign
_in);

        initLayouts();
        initializeListeners();
    }

    /*    init    layout    */
    private void  initLayouts()  {
        mEdMobileNumber    =
findViewById(R.id.edMobileNumber);
        mBtnContinue    =
findViewById(R.id.btnContinue);
        mIvFacebook    =
findViewById(R.id.ivFacebook);
        mIvGoogle    =
findViewById(R.id.ivGoogle);
    }

    private void  initializeListeners()  {

mBtnContinue.setOnClickListener(this
);

mIvFacebook.setOnClickListener(this)
;

mIvGoogle.setOnClickListener(this);

mBtnContinue.setStateListAnimator(n
ull);

mEdMobileNumber.setOnEditorActio
nListener(new
EditText.OnEditorActionListener()  {
        @Override
        public          boolean
onEditorAction(Textview  v,  int
actionId,  KeyEvent  event)  {

```

```

        if (actionId == EditorInfo.IME_ACTION_DONE) {
            if (validate()) {

                //
                startActivity(VerificationActivity.class);
            }
            return true;
        }
        return false;
    }

});

private void sendOtp(String phoneNumber){

    PhoneAuthProvider.getInstance().verifyPhoneNumber(
        phoneNumber,
        60,
        TimeUnit.SECONDS,
        this,
        mCallbacks
    );

    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {

        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode==2){

            sendOtp(data.getStringExtra("phoneNumber"));
        }

    }

    private
    PhoneAuthProvider.OnVerificationSta
teChangedCallbacks mCallbacks =
new
PhoneAuthProvider.OnVerificationSta
teChangedCallbacks() {
    @Override
    public void
onVerificationCompleted(@NonNull
PhoneAuthCredential
phoneAuthCredential) {
        hideLoading();
    }

    @Override
    public void
onVerificationFailed(@NonNull
FirebaseException e) {
        hideLoading();

        Toast.makeText(SignInActivity.this, e.g
etMessage(), Toast.LENGTH_SHORT).
show();
    }

    @Override
    public void onCodeSent(String
verificationId,
PhoneAuthProvider.ForceResendingT
oken token) {
        hideLoading();
        String mVerificationId =
verificationId;
        Log.e("MainActivity"
, "Verification id : " + verificationId);
        Intent intent = new
Intent(SignInActivity.this
, VerificationActivity.class);
        intent.putExtra("verificationId"
, mVerificationId);
        intent.putExtra("phoneNumber"
, mEdMobileNumber.getText().toString
());
        startActivity(intent);
        finish();
    }
}

```

```

};

/*      validation      */
private boolean validate() {
    boolean flag = true;
    showLoading("Mengirim Kode
Verifikasi");
    if
(TextUtils.isEmpty(mEdMobileNumber.getText()))
        flag = false;

    Toast.makeText(this,getString(R.string
.msg_mobile_number),Toast.LENGTH
_SHORT).show();
    }else{
        sendOtp("+62" +
mEdMobileNumber.getText());
    }
    return flag;
}

/*      onClick      listener      */
@Override
public void onClick(View v) {
    if (v == mBtnContinue) {
        if (validate()) {
            //
startActivity(VerificationActivity.class
);
        }
        } else if (v == mIvFacebook) {

            Intent i = new
Intent(Intent.ACTION_VIEW);

            i.setData(Uri.parse(getString(R.string.t
ext_facebooklink)));
            startActivity(i);

        } else if (v == mIvGoogle) {
            Intent i = new
Intent(Intent.ACTION_VIEW);

            i.setData(Uri.parse(getString(R.string.t

```

```

ext_googlelink)));
            startActivity(i);
        }
    }
}

```

SignUpActivity.java

```

package com.hera.bus.activities;

import android.content.Intent;
import android.os.Bundle;
import android.os.Trace;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.androidnetworking.AndroidNetworking;
import com.androidnetworking.common.Priority;
import com.androidnetworking.error.ANError;
import com.androidnetworking.interfaces.JSONObjectRequestListener;
import com.hera.bus.BaseActivity;
import com.hera.bus.R;
import com.hera.bus.utils.Servers;

import org.json.JSONException;
import org.json.JSONObject;

public class SignUpActivity extends
BaseActivity implements
View.OnClickListener {

    private ImageView ivBack;

```

```

private TextView tvTitle;
private EditText edMobileNumber;
private EditText namaPegguna;
private EditText emailPegguna;
private EditText alamatPegguna;
private Button btnRegister;
private String phoneNumber;
private Intent intent;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_sign_up);
    initView();
    initializeListeners();
}

private void initializeListeners() {
    btnRegister.setOnClickListener(this);
}

@Override
public void onClick(View v) {
    if (v == ivBack) {
        onBackPressed();
    }
    if (v == btnRegister){
        showLoading("Mendaftarkan Akun");
        validation();
    }
}

private boolean validation(){
    boolean flag = true;

    if (TextUtils.isEmpty(edMobileNumber.getText()) ||
    TextUtils.isEmpty(namaPegguna.getText()) ||
    TextUtils.isEmpty(emailPegguna.getText()) ||
    TextUtils.isEmpty(alamatPegguna.getText())){
        flag = false;
        Toast.makeText(this,getString(R.string.msg_register),Toast.LENGTH_SHORT).show();
        hideLoading();
    }else{
        flag = true;
        registerAccount();
    }

    return flag;
}

private void registerAccount(){
    AndroidNetworking.post(Servers.REGISTER_ACCOUNT)
        .addBodyParameter("no_hp_pegguna",phoneNumber)
        .addBodyParameter("nama_pegguna",namaPegguna.getText().toString())
        .addBodyParameter("alamat_pegguna",alamatPegguna.getText().toString())
        .addBodyParameter("email_pegguna",emailPegguna.getText().toString())
        .setPriority(Priority.HIGH)
        .build()
        .getAsJSONObject(new JSONObjectRequestListener() {
            @Override
            public void onResponse(JSONObject response) {
                try {
                    switch

```

```

(response.getString("sukses")){
    case "1":
        hideLoading();
        Toast.makeText(SignUpActivity.this,
            response.getString("pesan"),
            Toast.LENGTH_SHORT).show();
        CreateLoginSession(phoneNumber);
        Intent Main =
        new Intent(SignUpActivity.this,
            MainActivity.class);
        Main.putExtra("phonenumber",
            phoneNumber);
        startActivity(Main);
        break;
    case "0":
        hideLoading();
        Toast.makeText(SignUpActivity.this,
            response.getString("pesan"),
            Toast.LENGTH_SHORT).show();
        break;
    }
} catch
(JSONException e) {
    e.printStackTrace();
}
}

@Override
public void
onError(ANError anError) {
}
}

private void initView() {
    ivBack =
    findViewById(R.id.ivBack);
    tvTitle =
    findViewById(R.id.tvTitle);
    edMobileNumber =
    findViewById(R.id.edMobileNumber);
    namaPengguna =
    findViewById(R.id.namaPengguna);
    emailPengguna =
    findViewById(R.id.emailPengguna);
    alamatPengguna =
    findViewById(R.id.alamatPengguna);
    btnRegister =
    findViewById(R.id.btnRegister);

    intent = getIntent();
    phoneNumber =
    intent.getStringExtra("phonenumber");

    edMobileNumber.setText(phoneNumber);

    edMobileNumber.setEnabled(false);

    namaPengguna.setFocusable(true);
}
}

```