

**SISTEM INFORMASI *E*-KONDEKTUR UNTUK
TRANSAKSI TIKET BUS**

TUGAS AKHIR



**UNIVERSITAS
MA CHUNG**

**RENDY JHON KAILOLA
311510022**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MA CHUNG
MALANG
2021**

LEMBAR PENGESAHAN

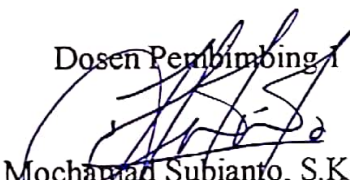
SISTEM INFORMASI E-KONDEKTUR UNTUK TRANSAKSI TIKET BUS

Oleh :
RENDY JHON KAILOLA
311510022

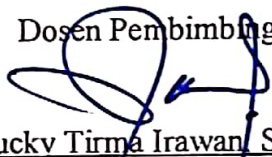
dari
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MA CHUNG

Telah dinyatakan lulus dalam melaksanakan Tugas Akhir (TA)
sebagai syarat kelulusan dan berhak mendapatkan gelar
Sarjana Komputer (S.Kom)

Dosen Pembimbing 1


Mochamad Subianto, S.Kom., M.Cs.
NIP. 20100002

Dosen Pembimbing 2


Paulus Lucky Tirna Irawan, S.Kom., MT
NIP. 20100005


Dekan Fakultas Sains dan
Teknologi
Dr. Kestriha Rega Brilianti, M.Si
NIP. 20120035

PERNYATAAN KEASLIAN
TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “SISTEM INFORMASI E-KONDEKTUR UNTUK TRANSAKSI TIKET BUS” adalah benar benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri. Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Malang, Februari 2021

Rendy Jhon Kailola
311510022

Sistem Informasi E-Kondektur Untuk Transaksi Tiket Bus

Rendy Jhon Kailola
311510022

Abstrak

Saat ini bus di Indonesia menggunakan 2 tipe pembayaran tiket yaitu pembayaran konvensional dan *online*. Kedua pembayaran tersebut memiliki kelemahannya masing-masing. Sistem konvensional yang masih belum terlalu efisien, dan juga masih terdapat beberapa tiket bus yang tidak mencetak harganya sehingga memungkinkan terjadinya kecurangan tentang harga bus. Pembayaran *online* juga memiliki kelemahan yaitu masih belum terlalu cocok dengan karakteristik orang-orang di Indonesia. Selain itu, sistem *online* juga membutuhkan infrastruktur tambahan yang pengaplikasiannya. Pada sisi penyedia dan pengguna jasa bus sendiri memiliki beberapa hal yang menjadi masalah. Pada sisi supir, tidak dapat mengetahui ada atau tidaknya penumpang di pinggir jalan. Kondektur diharuskan membawa sebandel tiket yang membuat pekerjaan kondektur tidak efisien. Pada penumpang, mereka tidak mengetahui informasi terkait bus yang akan lewat pada jalur yang mereka pilih.

Maka dari itu, akan dirancang sebuah sistem pembayaran tiket bus yang akan membantu pihak penyedia dan pengguna jasa bus. sistem akan dibuat menggunakan *framework* laravel dan juga angular serta memanfaatkan fitur Progressive Web Apps (PWA). Fitur ini dapat membantu aplikasi untuk tetap dapat tampil walaupun dalam mode *offline* (tidak ada koneksi internet). Selain itu, fitur ini membuat pengguna bisa merasakan pengalaman seperti menggunakan aplikasi *native* walaupun aplikasi yang dibuat adalah dalam bentuk *web*.

Hasil dari pengujian yang dilakukan terhadap aplikasi didapatkan hasil yaitu aplikasi dapat membantu pekerjaan pegawai dan penumpang menjadi lebih efisien serta mempermudah dalam hal penjualan tiket sesuai dengan hasil ujicoba yang menunjukkan hasil 60% sangat setuju dan 40% setuju (supir dan kondektur) dan 40% sangat setuju dan 40% setuju, sisanya 20% menjawab netral pada penumpang. Kemudian, Tampilan aplikasi sudah cukup menarik dan terstruktur sehingga mudah digunakan oleh penumpang dan pegawai. Hal ini sesuai dengan hasil ujicoba yang menunjukkan hasil pada penumpang 35% sangat setuju, 50% setuju, dan 15% netral, kondektur menunjukkan hasil 20% sangat setuju, 60% setuju, dan 20% netral, supir menunjukkan hasil 50% sangat setuju, 25% setuju, dan 25% netral. Terakhir, proses pembelian tiket menjadi lebih terstruktur dan teratur sehingga data penjualan dapat diolah lebih mudah.

Kata kunci : bus, tiket bus, kondektur, supir, *progressive web apps*, laravel, angular

E-Conductor Information System For Bus Ticket Transactions

Rendy Jhon Kailola
311510022

Abstract

Currently buses in Indonesia use 2 types of ticket payment, namely conventional and online payments. Both of these payments have their weaknesses. The conventional system is still not very efficient, and there are still some bus tickets that do not print the price, allowing fraud about bus prices. Online payments also have a drawback, namely that they don't really match the characteristics of the people in Indonesia. In addition, the online system also requires additional infrastructure for its application. On the side of the bus service providers and users themselves have several problems. On the driver's side, he cannot know whether there are passengers on the roadside. The conductor is required to carry a ticket which makes the conductor's job inefficient. For passengers, they do not know information related to buses that will pass on the route they choose.

Therefore, a bus ticket payment system will be designed that will assist bus service providers and users. The system will be built using laravel and angular frameworks and take advantage of Progressive Web Apps (PWA) features. This feature can help the application to appear even though it is in offline mode (no internet connection). In addition, this feature allows users to experience such as using a native application even though the application created is in the form of a web.

The results of the tests carried out on the application obtained results, namely the application can help the work of employees and passengers to be more efficient and make it easier in terms of ticket sales according to the test results which show the results of 60% strongly agree and 40% agree (drivers and conductors) and 40% strongly agreed and 40% agreed, the remaining 20% answered neutral to passengers. Then, the appearance of the application is quite attractive and structured so that it is easy to use by passengers and employees. This is in accordance with the test results which show the results of the passengers, 35% strongly agree, 50% agree, and 15% are neutral, the conductor shows the results of 20% strongly agree, 60% agree, and 20% neutral, the driver shows the results of 50% strongly agree, 25% agree, and 25% are neutral. Finally, the ticket purchase process becomes more structured and orderly so that sales data can be processed more easily.

Keywords : bus, bus ticket, conductor, driver, progressive web apps, laravel, angular

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa karena atas rahmat dan penyertaan-Nya sehingga tugas akhir dapat berjalan dengan baik. Laporan ini berisi hasil dari tugas akhir yang telah dilaksanakan di Universitas Ma Chung. Atas dukungan moral dan materil yang diberikan dalam pembuatan laporan ini, penulis ingin mengucapkan terima kasih kepada:

1. Orang tua dan keluarga yang telah mendukung penulis dalam menyelesaikan Tugas Akhir,
2. Bapak Hendry Setiawan, ST, M.Kom selaku Kepala Program Studi Teknik Informatika,
3. Bapak Mochamad Subianto, S.Kom., M.Cs selaku dosen pembimbing utama Tugas Akhir Universitas Ma Chung,
4. Paulus Lucky Tirma Irawan, S.Kom., MT. selaku dosen pembimbing kedua Tugas Akhir Universitas Ma Chung,
5. Marcell, Devina, Ko Andra. Daniel, Vanno, Sovi, Ekky, Natha, Yuan, Bima, dan teman-teman Teknik Informatika angkatan 2015 lainnya yang telah mendukung selama pengerjaan Tugas Akhir.

Laporan Tugas Akhir ini disusun berdasarkan materi yang diberikan yaitu “Sistem Informasi E-Kondektur Untuk Transaksi Tiket Bus”. Tugas akhir ini adalah sebagai mata kuliah yang wajib ditempuh mahasiswa Teknik Informatika Universitas Ma Chung Malang. Diharapkan dengan penulisan laporan tugas akhir ini, dapat bermanfaat bagi seluruh pembaca.

Malang, Januari 2021

Rendy Jhon Kailola

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	v
DAFTAR TABEL	vii
Bab I Pendahuluan	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	3
1.3 Batasan Masalah	3
1.4 Rumusan Masalah	3
1.5 Tujuan	4
1.6 Manfaat	4
1.7 Sistematika Penulisan	4
Bab II Tinjauan Pustaka	6
2.1 Transportasi	6
2.2 Bus	9
2.3 Tarif Angkutan Penumpang	10
2.4 PHP	13
2.5 MYSQL	14
2.6 PWA	16
2.7 Google <i>Maps</i> API	23
2.8 <i>GPS</i>	25
2.9 <i>Framework</i> Angular	26
2.10 <i>Framework</i> Laravel	26
2.11 <i>Flowchart</i>	27
2.12 Penelitian Terdahulu	29
Bab III Analisis dan Perancangan Sistem	31
3.1 Alur Penelitian	31
3.2 Analisis Masalah	31
3.3 Perancangan Sistem	32
3.3.1 <i>Flowchart</i> Sistem	32

3.3.2 <i>Data Flow Diagram (DFD)</i>	34
3.3.3 Basis Data Sistem	36
3.3.4 Rancangan Antarmuka Sistem	42
3.3.5 Garis Besar Aplikasi	48
3.3.6 Diagram Alir Aplikasi	48
3.4 Implementasi Sistem	50
3.5 Pengujian & Evaluasi Sistem	50
3.5.1 Hasil Pengujian	51
3.6 Pemeliharaan	53
Bab IV Hasil dan Pembahasan	54
4.1 Hasil Rancangan Aplikasi (<i>Back-End</i>)	54
4.1.1 Antarmuka Halaman <i>Login</i> Admin	54
4.1.2 Antarmuka Halaman <i>Dashboard</i>	55
4.1.3 Antarmuka Halaman Data	55
4.1.4 Antarmuka Halaman Tambah Data Baru	56
4.2 Hasil Rancangan Aplikasi (<i>Front-End</i>)	56
4.2.1 Antarmuka Halaman <i>Login</i> (Pegawai)	57
4.2.2 Antarmuka Halaman Jadwal (Pegawai)	57
4.2.3 Antarmuka Halaman <i>Map</i> (Supir)	58
4.2.4 Antarmuka Halaman Tiket (Kondektur)	59
4.2.5 Antarmuka Halaman <i>Login</i> (Penumpang)	62
4.2.6 Antarmuka Halaman Registrasi (Penumpang)	63
4.2.7 Antarmuka Halaman Cari Bus (Penumpang)	64
4.3 Sitemap Aplikasi	64
4.3.1 Sitemap Aplikasi Pada Sisi Admin	65
4.3.2 Sitemap Aplikasi Pada Sisi Kondektur	66
4.3.3 Sitemap Aplikasi Pada Sisi Supir	67
4.3.4 Sitemap Aplikasi Pada Sisi Penumpang	67
4.4 Penerapan Sistem PWA (<i>Progressive Web App</i>) Dalam Aplikasi	68
4.4.1 Antarmuka Aplikasi Pada Bagian Penumpang Dengan PWA	69
4.4.2 Antarmuka Aplikasi Pada Bagian Pegawai Dengan PWA	71
4.5 Hasil Uji Coba Penggunaan Aplikasi	72

4.5.1 Hasil Pengujian <i>Black Box</i>	73
4.5.2 Hasil Survei	73
Bab V Penutup	79
5.1 Kesimpulan	79
5.2 Saran	79
DAFTAR PUSTAKA	80

DAFTAR GAMBAR

Gambar 2.1 Tarif Seragam	11
Gambar 2.2 Tarif Berdasarkan Jarak	11
Gambar 2.3 Tarif Bertahap	12
Gambar 2.4 Tarif Zona	13
Gambar 2.5 Service Worker	20
Gambar 2.6 Google Maps API Key	23
Gambar 2.7 Contoh google maps Roadmap	24
Gambar 2.8 Contoh google maps Satellite	24
Gambar 2.9 Contoh google maps Terrain	25
Gambar 2.10 Contoh google maps Hybrid	25
Gambar 3.1 Alur Penelitian Waterfall	29
Gambar 3.2 Flowchart transaksi tiket bus secara manual	31
Gambar 3.3 Contex Diagram Aplikasi	33
Gambar 3.4 Antarmuka Halaman Login	40
Gambar 3.5 Antarmuka Halaman Dashboard Admin	41
Gambar 3.6 Antarmuka Halaman Data	41
Gambar 3.7 Antarmuka Halaman Rekapitulasi Penjualan Tiket	42
Gambar 3.8 Antarmuka Halaman Jadwal Pegawai	43
Gamabr 3.9 Antarmuka Halaman Input Tiket	43
Gambar 3.10 Antarmuka Halaman Supir	44
Gambar 3.11 Antarmuka Halaman Login Penumpang	45
Gambar 3.12 Antarmuka Halaman Penumpang	45
Gambar 3.13 Proses Pencetakan Tiket	47
Gambar 3.14 Proses Pembuatan Jadwal	47
Gambar 3.15 Proses Penginputan Data	48
Gambar 3.16 Contoh Pengujian User Interface	50
Gambar 3.17 Contoh Pengujian User Experience	50
Gambar 4.1 Tampilan Antarmuka Halaman Login Admin	54
Gambar 4.2 Tampilan Antarmuka Halaman Dashboard	55
Gambar 4.3 Tampilan Antarmuka Halaman Data	56

Gambar 4.4 Tampilan Antarmuka Halaman Tambah Data Baru	56
Gambar 4.5 Tampilan Antarmuka Halaman Login Pegawai	57
Gambar 4.6 Tampilan Antarmuka Halaman Jadwal	58
Gambar 4.7 Tampilan Antarmuka Halaman Map	59
Gambar 4.8 Tampilan Antarmuka Halaman Tiket	60
Gambar 4.9 Antarmuka Halaman Report	61
Gambar 4.10 Tampilan Antarmuka Halaman Confirm Request	62
Gambar 4.11 Tampilan Antarmuka Halaman Login Penumpang	62
Gambar 4.12 Tampilan Antarmuka Halaman Registrasi	63
Gambar 4.13 Tampilan Antarmuka Halaman Cari Bus	64
Gambar 4.14 Sitemap Aplikasi Pada Sisi Admin	65
Gambar 4.15 Sitemap Aplikasi Pada Sisi Kondektur	66
Gambar 4.16 Sitemap Aplikasi Pada Sisi Supir	67
Gambar 4.17 Sitemap Aplikasi Pada Sisi Penumpang	67
Gambar 4.18 Service Worker Pada Aplikasi	69
Gambar 4.19 Antarmuka Halaman Login Penumpang Dengan PWA	69
Gambar 4.20 Antarmuka Halaman Register Penumpang Dengan PWA	70
Gambar 4.21 Antarmuka Halaman Utama Penumpang Dengan PWA	70
Gambar 4.22 Antarmuka Halaman Profile Penumpang Dengan PWA	71
Gambar 4.23 Antarmuka Halaman Login Pegawai Dengan PWA	71
Gambar 4.24 Antarmuka Halaman Utama Kondektur Dengan PWA	72
Gambar 4.25 Antarmuka Halaman Report Pegawai Dengan PWA	72
Gambar 4.26 Pilihan Jawaban Responden Terhadap Pertanyaan 1	74
Gambar 4.27 Pilihan Jawaban Responden Terhadap Pertanyaan 2	75
Gambar 4.28 Pilihan Jawaban Responden Terhadap Pertanyaan 3	75
Gambar 4.29 Pilihan Jawaban Responden Terhadap Pertanyaan 4	76
Gambar 4.30 Pilihan Jawaban Responden Terhadap Pertanyaan 5	76
Gambar 4.31 Pilihan Jawaban Responden Terhadap Pertanyaan 6	77
Gambar 4.32 Pilihan Jawaban Responden Terhadap Pertanyaan 7	78

DAFTAR TABEL

Tabel 2.1 Jenis Moda Transportasi	9
Tabel 2.2 Simbol Flowchart	27
Tabel 2.2 Simbol Flowchart (Lanjutan)	28
Tabel 3.1 Tabel User	37
Tabel 3.2 Tabel Employee	37
Tabel 3.2 Tabel Employee (Lanjutan)	37
Tabel 3.3 Tabel Schedule	38
Tabel 3.4 Tabel Request	39
Tabel 3.5 Tabel Bus	39
Tabel 3.6 Tabel Track	40
Tabel 3.7 Tabel Trip	40
Tabel 3.8 Tabel Ticket	40
Tabel 3.9 Tabel City	41
Tabel 3.10 Tabel Type	41
Tabel 3.11 Tabel Occupation	41
Tabel 4.1 Tabel Hasil Pengujian Black Box Aplikasi	7

Bab I

Pendahuluan

1.1 Latar Belakang

Sistem pembayaran tiket bus yang ada saat ini masih menggunakan sistem konvensional atau yang disebut *paper ticket*. Saat ini, pembayaran tiket secara *online* sudah bisa digunakan, namun orang-orang masih banyak yang memilih menggunakan sistem tiket konvensional karena sistem tiket yang masih sesuai dengan karakteristik orang-orang di Indonesia. Selain itu, banyak kasus-kasus khusus di Indonesia yang belum bisa ditangani oleh sistem *online* seperti penumpang yang naik tidak dari halte, kemudian orang yang naik ke bus namun bukan penumpang (pengamen atau pedagang asongan). Pembangunan sistem *online* juga memerlukan perangkat dan infrastruktur tambahan, dan masih banyak pengguna bus yang butuh di sosialisasi tentang bagaimana cara kerja sistem *online* sebelum sistem ini diterapkan.

Tiket konvensional juga memiliki kelemahan yakni kurang efisien karena banyak mengeluarkan biaya pada saat mencetak tiket, kemudian pada tiket yang diberikan tidak ada informasi tarif yang telah ditetapkan oleh perusahaan sehingga rentan terjadi kecurangan dalam penentuan tarif perjalanan. Meskipun beberapa tiket bus sudah mencantumkan harga seperti bus patas jurusan surabaya-jakarta, namun masih banyak ditemui tiket konvensional yang belum mencantumkan tarif pada lembaran tiket yang diberikan ke penumpang. Penumpang hanya mendapatkan informasi tarif dari kondektur yang memberikan tiket. Selain itu, pelaporan hasil pendapatan tiap bus membutuhkan waktu yang lama karena harus menghitung jumlah tiket yang terjual secara manual.

Selain masalah tiket konvensional tersebut, ada masalah lain yang dihadapi. Bus yang telah melakukan perjalanan akan mengangkut penumpang di halte-halte bus pada jalur yang dilalui oleh bus jika tempat duduk bus belum terisi semua. Terkadang, bus juga akan mengangkut penumpang yang tidak berada di halte melainkan berada di tepi jalan. Calon penumpang yang berada pada halte bus juga mengalami masalah. Kurangnya informasi tentang bus yang akan melewati halte tersebut. Informasi tersebut meliputi waktu sampai bus tiba ke

halte, kondisi tempat duduk di bus (penuh atau tidak), dan juga jurusan dari bus tersebut.

Pada setiap PO juga terdapat beberapa orang yang disebut dengan tower. Ini adalah sebutan untuk orang yang ditempatkan pada beberapa titik halte untuk memantau ada atau tidaknya penumpang yang menuju ke kota tertentu. Tower akan melaporkan kepada awak pada bus yang sedang menuju halte yang di tempati apakah ada penumpang yang ingin melakukan perjalanan yang sama dengan jurusan dari bus tersebut. Fungsi tower lainnya adalah untuk memberitahukan kepada kondektur/supir jarak bus yang berada di depan mereka sehingga supir bisa mengatur agar jarak antara mereka tidak terlalu jauh atau dekat.

Pada penelitian terdahulu yang dilakukan oleh Saputra (2017), yang membahas tentang pengembangan aplikasi pemesanan tiket bus pada PO Puspa Jaya berbasis android yang bisa diakses secara *online*. Penelitian ini menghasilkan aplikasi pemesanan tiket bus Puspa Jaya berbasis android. Pemesanan tiket dilakukan sebelum penumpang menaiki bus yang ingin ditumpangi melalui aplikasi pemesanan tiket yang telah dibuat. Penumpang harus memiliki *smartphone* yang tersambung dengan koneksi internet agar bisa menggunakan aplikasi. Pada penelitian ini dilakukan pembuatan aplikasi transaksi tiket bus berbasis aplikasi web (PWA), sehingga penumpang bus yang belum memiliki *smartphone* pun dapat mengakses aplikasi tersebut. Selain itu, dengan menggunakan konsep PWA, masalah keterbatasan koneksi masih bisa diatasi.

Berdasarkan latar belakang di atas, peneliti akan membuat sistem pengganti tiket konvensional (*paper ticket*) dengan menggunakan media *smartphone*. Dengan sistem ini maka kondektur tidak perlu lagi membawa sebungkus tiket bus. Sistem ini diharapkan dapat membuat sistem pembayaran tiket pada bus lebih efisien dari sebelumnya. Diharapkan juga sistem ini membantu supir atau kondektur untuk mengetahui apakah di tepi jalan ada penumpang yang ingin naik atau tidak. Selain itu sistem ini membantu penumpang mendapatkan informasi tentang bus yang akan lewat pada halte tersebut. Sistem juga diharapkan bisa menggantikan tugas *tower* pada beberapa halte sehingga mengurangi pengeluaran biaya bus.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah dijabarkan sebelumnya, didapatkan masalah yaitu sistem pembayaran tiket bus yang masih dilakukan dengan cara konvensional sehingga didapati beberapa kelemahan, tidak dicantumkannya tarif pada tiket yang diberikan sehingga memunculkan kecurangan kondektur. Kemudian, kurangnya informasi kepada supir tentang ada atau tidaknya penumpang di tepi jalan mengakibatkan fokus supir teralihkan dari jalan raya untuk mencari penumpang. Selain itu, penumpang yang tidak mendapat informasi bus apa saja yang akan melewati halte yang mereka tempati mengakibatkan penumpang datang ke halte terlalu cepat atau lama.

1.3 Batasan Masalah

Adapun batasan masalah pada penelitian ini yaitu:

- a. Sistem tiket bus akan dibuat berbasis web (*Progressive Web Apps*),
- b. Sistem akan dibuat untuk 4 pihak yaitu admin, kondektur, supir, dan penumpang,
- c. Sistem akan diuji cobakan pada bus ekonomi antar kota dalam provinsi dimana tipe bus ini bisa berhenti di tepi jalan untuk mengangkut penumpang,
- d. Kondektur akan mencetak tiket dengan *smartphone* dibantu *pocket printer*,
- e. Admin bisa mengakses sistem yang dibuat melalui *website*,
- f. Kondektur, supir, dan penumpang dapat mengakses sistem melalui *smartphone*.

1.4 Rumusan Masalah

Berdasarkan identifikasi masalah tersebut, maka dapat disimpulkan rumusan masalah yaitu bagaimana merancang sebuah sistem pembayaran tiket bus *realtime* yang akan membantu pihak penyedia jasa bus yaitu PO dan awak bus yang bertugas serta pihak pengguna jasa bus yaitu pengguna.

1.5 Tujuan

Tujuan penelitian ini adalah merancang sebuah sistem pembayaran tiket bus berbasis web yang akan membantu pihak penyedia dan pengguna jasa bus.

1.6 Manfaat

Manfaat yang didapatkan dari tugas akhir ini adalah sebagai berikut:

- a. Bagi Universitas Ma Chung, khususnya program studi teknik informatika dapat menghasilkan sebuah aplikasi yang dapat digunakan untuk media pembelajaran dan mempersiapkan lulusan yang berkompeten dan siap kerja.
- b. Bagi peneliti, mampu menerapkan ilmu-ilmu yang telah didapatkan di Universitas Ma Chung.
- c. Bagi pembaca, dapat menambah informasi dan menjadi inspirasi untuk melakukan pengembangan terkait sistem yang telah dibuat.
- d. Bagi pengguna, diharapkan dapat membantu dalam hal operasional bus dan juga menggunakan jasa transportasi bus.

1.7 Sistematika Penulisan

Sistematika penulisan laporan Tugas Akhir terbagi dalam beberapa bagian yaitu:

Bab I : Pendahuluan

Bab pendahuluan terdiri dari latar belakang, identifikasi masalah, batasan masalah, rumusan masalah, Tujuan, manfaat, dan sistematika penulisan.

Bab II : Tinjauan Pustaka

Bab tinjauan pustaka terdiri dari penguraian teori tentang transportasi, bus, *website*, *Hyper Text Markup Language*, *Google Maps API*, *GPS*, *MYSQL*, *printer*, *PHP Hypertext Processor*, *PWA*.

Bab III : Analisis dan Perancangan Sistem

Bab analisis dan perancangan sistem terdiri dari penguraian detail mengenai analisis kebutuhan, perancangan penelitian dan metode yang akan digunakan dalam penelitian.

Bab IV : Hasil dan Pembahasan

Bab hasil dan pembahasan terdiri dari hasil perancangan sistem yang telah dibuat dan hasil pengujian sistem yang telah dibuat.

Bab V : Penutup

Bab penutup terdiri dari kesimpulan akan hasil sistem yang telah dibuat dan juga diuji pada bab sebelumnya dan juga saran bagi pengembangan sistem selanjutnya.

Bab II

Tinjauan Pustaka

2.1 Transportasi

Transportasi berasal dari kata *transportare* (bahasa latin), yang terbagi atas dua kata, yaitu *trans* dan *portare*. Kata *trans* memiliki arti menyeberangi dan *portare* yang berarti membawa. Definisi transportasi menurut beberapa ahli adalah sebagai berikut:

1. Menurut Bowersox (1981), transportasi adalah perpindahan barang atau penumpang dari suatu tempat ke tempat lain, dimana produk dipindahkan ke tempat tujuan dibutuhkan. Secara umum transportasi adalah suatu kegiatan memindahkan sesuatu (barang atau barang) dari suatu tempat ke tempat lain, baik dengan atau tanpa sarana.
2. Menurut Papacostas (1987), transportasi didefinisikan sebagai suatu sistem yang terdiri dari fasilitas tertentu beserta arus dan sistem *control* yang memungkinkan orang atau barang dapat berpindah dari suatu tempat ke tempat lain secara efisien dalam setiap waktu untuk mendukung aktivitas manusia.
3. Menurut Salim (2000), transportasi adalah kegiatan pemindahan barang (muatan) dan penumpang dari suatu tempat ke tempat lain.
4. Menurut Miro (2004), transportasi dapat diartikan usaha memindahkan, mengerakkan, mengangkut, atau mengalihkan suatu objek dari suatu tempat ke tempat lain, di mana di tempat lain ini objek tersebut lebih bermanfaat atau dapat berguna untuk tujuan-tujuan tertentu.

Dengan demikian, transportasi dapat diartikan sebagai proses perpindahan suatu benda/objek dari suatu titik menuju titik lainnya dengan menggunakan bantuan perantara.

Pengertian dari moda yaitu merupakan sarana yang digunakan untuk memindahkan orang dan/atau barang dari suatu tempat ke tempat yang lain. Menurut Munawar (2005), moda transportasi dapat berupa moda transportasi

darat, moda transportasi laut, dan moda transportasi udara, di mana masing-masing moda tersebut memiliki ciri dan karakteristik sendiri. Dalam penelitian ini yang akan dibahas adalah moda transportasi darat, khususnya angkutan umum (bus). Menurut Miro (2008), secara umum, ada dua kelompok besar moda transportasi yaitu :

1. Kendaraan Pribadi (*Private Transportation*), yaitu moda transportasi yang dikhususkan buat pribadi seseorang dan seseorang itu bebas memakainya ke mana saja, di mana saja dan kapan saja dia mau, bahkan mungkin juga dia tidak memakainya sama sekali (mobilnya disimpan di garasi). Contoh kendaraan pribadi seperti:
 - a. Sepeda untuk pribadi,
 - b. Sepeda motor untuk pribadi,
 - c. Mobil pribadi.
2. Kendaraan Umum (*Public Transportation*), yaitu moda transportasi yang diperuntukkan buat bersama (orang banyak), kepentingan bersama, menerima pelayanan bersama, mempunyai arah dan titik tujuan yang sama, serta terikat dengan peraturan trayek yang sudah ditentukan dan jadwal yang sudah ditetapkan dan para pelaku perjalanan harus wajib menyesuaikan diri dengan ketentuan tersebut apabila angkutan umum ini sudah mereka pilih. Contoh kendaraan umum seperti:
 - a. sepeda motor (ojek),
 - b. Becak, bajaj, bemo,
 - c. Mikrolet, taksi,
 - d. Bus umum (kota dan antar kota),
 - e. Kereta api (kota dan antar kota),
 - f. Kapal penyeberangan (feri),
 - g. Pesawat komersial.

Secara garis besar dengan melihat mediumnya transportasi ini dapat dibedakan menjadi moda darat air dan udara. Menurut Aziz dan Asrul (2014), lebih jauh moda darat dapat dibedakan menjadi moda jalan dan kereta api. Moda

transportasi darat terdiri dari berbagai variasi dan jenis alat transportasinya. Menurut Miro (2012), Transportasi darat dapat di klasifikasikan menjadi :

1. Geografis fisik, terdiri dari moda transportasi jalan rel, moda transportasi perairan daratan, moda transportasi khusus dari pipa dan kabel serta moda transportasi jalan raya,
2. Geografis administratif, terbagi atas transportasi dalam kota, transportasi desa, transportasi antar-kota dalam provinsi (AKDP), transportasi perkotaan antar-kota antar provinsi (AKAP) dan transportasi lintas batas antar-negara (internasional).

Menurut Undang-Undang Republik Indonesia Nomor 22 tahun 2009 tentang Lalu Lintas dan Angkutan Jalan, yang dimaksud dengan jalan adalah prasarana transportasi darat yang meliputi segala bagian jalan, termasuk bangunan pelengkap dan perlengkapannya yang diperuntukkan bagi lalu lintas, yang berada pada permukaan tanah, di atas permukaan tanah, di bawah permukaan tanah dan/atau air, serta di atas permukaan air, kecuali jalan kereta api, jalan lori, dan jalan kabel. Menurut Undang-Undang Nomor 38 Tahun 2004 Pasal 5, peran jalan dibagi menjadi 3 yaitu:

1. Sebagai prasarana transportasi : mempunyai peran penting dalam bidang ekonomi, sosial, budaya, lingkungan hidup, politik, hankam, serta dipergunakan untuk sebesar-besar kemakmuran rakyat,
2. Sebagai prasarana distribusi barang dan jasa : merupakan urat nadi kehidupan masyarakat, bangsa, dan negara,
3. Merupakan satu kesatuan sistem jaringan jalan : menghubungkan dan mengikat seluruh Wilayah Republik Indonesia.

Menurut Undang-Undang Nomor 38 Tahun 2004 Pasal 5 komponen prasarana transportasi terbagi menjadi dua kelompok yaitu :

1. Jalan yang berupa jalur gerak seperti jalan raya, jalan baja, jalan air, jalan udara dan jalan khusus,
2. Terminal yang berupa suatu tempat pemberhentian alat transportasi guna menurunkan atau menaikkan penumpang dan barang seperti:

- a. Terminal jalan raya (stasiun bus, halte bus dan lain-lain),
- b. Terminal jalan rel yaitu kereta api,
- c. Terminal jalan khusus seperti gudang dan lain-lain.

Penyediaan armada dan fasilitas atau prasarana untuk mendukung pergerakan dapat disesuaikan dengan jenis moda yang digunakan. Tabel 2.1 merujuk pada Jenis moda dan angkutan umum penumpang yang ada dalam transportasi darat yaitu :

Tabel 2.1 Jenis Moda Transportasi (Sumber : Hadihardaja, Sistem Transportasi, 1997)

Jenis angkutan penumpang	Badan/ <i>Body</i>	Tenaga Penggerak	Cara Bergerak	Sistem Kontrol
sedan	kabin untuk pengemudi (4-5 orang)	mesin diesel	menggunakan roda karet	pengemudi
mini bus	kabin untuk pengemudi (4-5 orang)	mesin diesel	menggunakan roda karet	pengemudi
bus	kabin untuk pengemudi (30 orang)	mesin diesel	menggunakan roda karet	pengemudi
	kabin untuk pengemudi (50 orang)	mesin diesel	menggunakan roda karet	pengemudi
kereta	gerbong tertutup	diesel	menggunakan roda karet	signal
		listrik	menggunakan roda karet	signal
		listrik induksi linear	tolak menolak gaya magnet	signal

2.2 Bus

Menurut Kamus Besar Bahasa Indonesia (KBBI), bus adalah setiap kendaraan bermotor yang dilengkapi dengan tempat duduk untuk lebih dari 8 orang, tidak termasuk tempat duduk untuk pengemudi baik dilengkapi atau tidak

dilengkapi dengan bagasi. Bus menurut Vuchic (1981) didefinisikan sebagai moda perjalanan darat dengan kapasitas medium. Bus diklasifikasikan dalam 2 bagian yaitu :

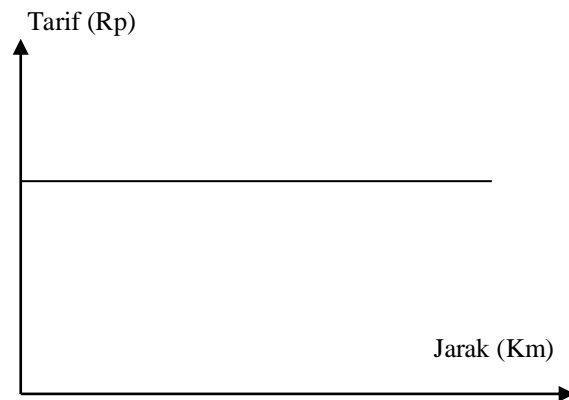
1. Bus *regular*/umum dengan karakteristik :
 - a. Beroperasi dengan rute tetap dan memiliki jadwal/durasi yang tetap,
 - b. Jenis mulai dari bus sedang (kapasitas 20-35 penumpang), sampai dengan bus artikulasi (kapasitas >150 penumpang),
 - c. Pelayanan bervariasi meliputi tingkat pelayanan, ongkos, kinerja dan dampaknya.
2. Bus *express*/cepat dengan karakteristik :
 - a. Cepat, nyaman dengan rute panjang dan tetap,
 - b. Pemberhentian sedikit dan terbatas,
 - c. Biaya perjalanan lebih mahal,
 - d. Lebih cepat namun dipengaruhi juga oleh kondisi lalu-lintas.

2.3 Tarif Angkutan Penumpang

Menurut Frids (2002), jenis tarif yang berlaku dapat dikelompokkan sebagai berikut:

1. Tarif Seragam (*Flat Fare*)

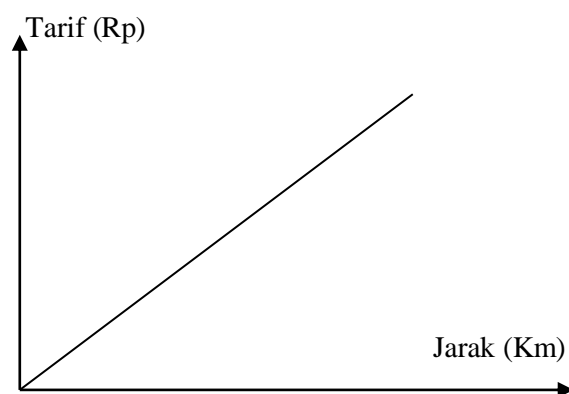
Pada sistem ini, tarif dikenakan tanpa memperhatikan jarak yang ditempuh, baik perjalanan jarak pendek maupun jauh dikenakan tarif yang sama. Secara umum, tarif seragam biasanya diterapkan untuk penumpang yang mempunyai panjang perjalanan rata-rata hampir sama. Kerugian tarif ini adalah pada penumpang yang melakukan perjalanan jarak pendek karena harus membayar dengan tarif yang sama dengan penumpang yang melakukan perjalanan jarak jauh. Sebaliknya penumpang yang melakukan perjalanan jarak jauh akan diuntungkan dengan kondisi ini. Gambar 2.1 adalah gambaran grafik dari jenis tarif seragam.



Gambar 2.1 Tarif Seragam

2. Tarif Berdasarkan Jarak (*Distance-Based Fare*)

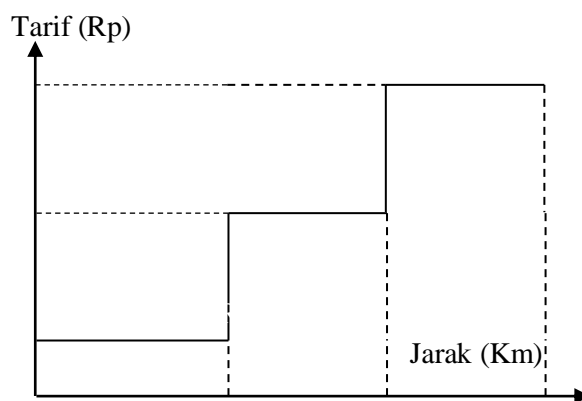
Sistem tarif ini ditentukan berdasarkan jarak yang ditempuh, yaitu besarnya tarif yang ditetapkan adalah perkalian besar tarif per kilometer dengan panjang perjalanan, dimana jarak minimum dan tarif minimum ditetapkan terlebih dahulu nilainya. Sistem tarif ini memiliki kelemahan, yaitu kesulitan dalam pengumpulan ongkos karena sebagian penumpang melakukan perjalanan yang relatif pendek menggunakan angkutan lokal. Gambar 2.2 adalah gambaran grafik dari jenis tarif berdasarkan jarak.



Gambar 2.2 Tarif Berdasarkan Jarak

3. Tarif Bertahap

Sistem tarif ini didasarkan pada jarak yang ditempuh oleh penumpang yang di bagi persatuan tahapan. Tahapan adalah suatu penggalan dari rute yang jaraknya antar satu atau lebih tempat pemberhentian sebagai dasar perhitungan tarif. Tarif bertahap mencerminkan usaha penggabungan secara wajar keinginan penumpang dan pertimbangan biaya yang dikeluarkan perusahaan dengan waktu untuk mengeluarkan ongkos. Struktur seperti ini tidak hanya digunakan dengan memperhitungkan bermacam-macam permintaan pelayanan perangkutan untuk jarak pendek dan panjang tapi juga akan menguntungkan jika memperhatikan metode pengumpulan tarif. Gambar 2.3 adalah gambaran grafik dari jenis tarif bertahap.

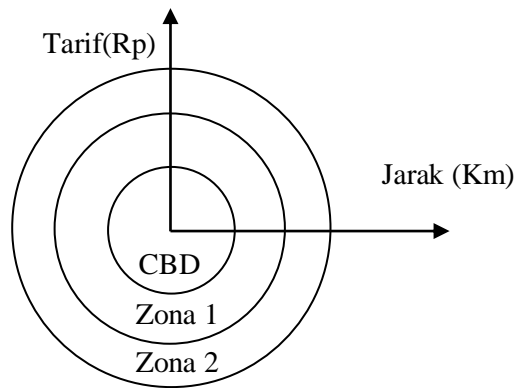


Gambar 2.3 Tarif Bertahap

4. Tarif Zona

Sistem tarif ini adalah penyederhanaan dari tarif bertahap dimana daerah pelayanan perangkutan tersebut dibagi kedalam zona-zona. Pusat kota biasanya sebagai zona terdalam (CBD) dan dikelilingi oleh zona terluar yang tersusun seperti sebuah sabuk. Daerah pelayanan angkutan juga dapat dibagi kedalam zona-zona yang berdekatan. Jika terdapat jalan yang melintang dan melingkar, panjang jalan ini harus dibatasi dengan membagi zona kedalam sektor-sektor. Skala jarak dan tarif dibentuk dengan cara yang sama dengan struktur tarif bertahap yang berdasarkan suatu jarak dan suatu tingkatan tarif. Kerugian akan

terjadi bagi penumpang yang hanya melakukan perjalanan jarak pendek didalam dua zona yang berdekatan, mereka harus membayar ongkos untuk dua zona. Sebaliknya suatu perjalanan yang panjang dapat menjadi lebih murah jika dilakukan didalam sebuah zona dibandingkan dengan perjalanan pendek yang melintasi batas zona. Gambar 2.4 adalah gambaran grafik dari jenis tarif zona.



Gambar 2.4 Tarif Zona

2.4 PHP

PHP *Hypertext Processor* atau yang disingkat PHP merupakan bahasa pemrograman web yang bekerja pada *server side*. Menurut Suteja, Prijono dan Agustaf (2007) menyatakan bahwa *Server side* merupakan *interpreter* pada *web scripting* dilakukan oleh *web server* seperti Apache, Tomcat, IIS dan lain-lain.

Menurut Peraninangin (2006) menyatakan bahwa PHP disisipkan dalam dokumen HTML. Penggunaan PHP memungkinkan web menjadi dinamis sehingga *maintenance* situs menjadi lebih mudah dan efisien. Dinamis disini juga berarti halaman yang ditampilkan dibuat sesuai dengan permintaan *client*. PHP merupakan *software Open Source* yang dilisensikan secara gratis dan dapat diunduh secara bebas melalui situs resmi.

PHP memiliki variabel (bentuk data dari nilai atau varian). Variabel terbagi atas dua sesuai pembentukannya, yaitu variabel biasa atau nilai yang didalamnya dapat diubah sepanjang variabel tersebut berada, dan variabel konstanta (variabel data yang nilainya tidak dapat diubah). Variabel juga terbagi atas dua jenis, yaitu variabel *local* yang berada pada satu *file* atau aktif pada lokasi dibuatnya variabel, dan variabel *global* yang aktif dan dapat digunakan pada

setiap *file* PHP yang menggunakan *file* pembuatan variabel. Variabel global pada PHP dapat menggunakan dua jenis, yaitu variabel global dengan menggunakan *Session* dan variabel global dengan menggunakan *cookies*. Terdapat perbedaan pada variabel global tersebut. Perbedaan tersebut dapat dilihat pada *session* apabila halaman tersebut tertutup atau *browser* tertutup maka variabel ini juga akan hilang. Sedangkan *cookies* apabila halaman atau *browser* ditutup maka variabel ini tetap ada sesuai jangka waktu yang ditentukan.

Pada PHP setiap variabel harus diawali dengan tanda ('\$'). Setiap variabel pada PHP memungkinkan untuk tidak dapat didefinisikan terlebih dahulu dan dapat secara otomatis didefinisikan ketika variabel tersebut digunakan. Pada PHP terdapat *statement* berupa pengambilan keputusan *if* dan *switch* atau perulangan dengan *for*, *while*, dan *do while*, serta memiliki operator-operator seperti logika, aritmetika atau *String*.

2.5 MYSQL

Menurut Nugroho (2005) menyatakan bahwa MySQL merupakan sebuah program *database server* yang mampu menerima dan mengirimkan data dengan sangat cepat, *multi-user* serta menggunakan perintah standar SQL (*Structured Query Language*). MySQL dikembangkan sekitar tahun 1994 oleh sebuah perusahaan pengembang perangkat lunak dan konsultan *database* bernama MySQL AB yang bertempat di Swedia. MySQL merupakan *multi-user* database yang menggunakan SQL. SQL adalah bahasa standar yang digunakan untuk mengakses *server database*.

MySQL memiliki banyak sekali peralatan yang dapat memudahkan untuk administrasi. Sebagai *database server* yang memiliki konsep *database* modern, MySQL memiliki banyak sekali keistimewaan. Keistimewaan yang dimiliki MySQL merupakan fungsi-fungsi yang sangat membantu dalam permasalahan pada konsep.

Berikut ini adalah keistimewaan MySQL :

1. Portability

MySQL dapat berjalan stabil pada berbagai sistem operasi, seperti Windows, Linux, Solaris, dan lain-lain.

2. *Open Source*

MySQL didistribusikan secara *open source* dibawah lisensi GPL, sehingga dapat digunakan tanpa dipungut biaya.

3. *Multi-User*

MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik. Hal ini memungkinkan sebuah *database server* MySQL dapat diakses *client* secara bersamaan.

4. *Security*

MySQL memiliki beberapa lapisan sekuritas seperti level *subnetmask*, nama *host* dan izin akses pengguna dengan sistem perizinan yang mendetail serta *password* yang terenkripsi.

5. *Scalability*

MySQL mampu menangani *database* dalam skala besar, dengan jumlah lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris.

Ada beberapa tipe-tipe data dalam MySQL yang sering digunakan. Tipe-tipe data itu antara lain:

1. *Numeric Values* yaitu angka atau bilangan seperti 10; 123; 100.50; -10; 1.2E+17; 2.7e-11; dan sebagainya. Tipe data ini dibagi menjadi dua yaitu Bilangan Bulat (*Integer*) dan Bilangan Pecahan (*Floating-point*). Bilangan bulat adalah bilangan tanpa tanda desimal sedangkan bilangan pecahan adalah bilangan dengan tanda desimal. Kedua jenis bilangan ini dapat bernilai positif (+) dan juga negatif (-). Jika bilangan tersebut menggunakan tanda positif (+) atau (-), maka disebut *SIGNED*. Sebaliknya bila tanpa tanda apapun disebut *UNSIGNED*. Karena tanda positif (+) dapat diabaikan penulisannya maka pada bilangan yang bernilai positif disebut *UNSIGNED*.
2. *String/Character Values* adalah semua karakter (atau teks) yang penulisannya selalu diapit oleh tanda kutip baik kutip tunggal (') maupun kutip ganda ("). Hal ini tidak hanya berlaku pada huruf alfabet

saja tetapi angka yang ditulis dengan tanda kutip pun akan menjadi karakter atau string.

3. *Date and Time Values* yaitu tanggal dan waktu. Untuk jenis data tanggal dan waktu format standar (*default*) penulisan tanggalnya adalah “tahun-bulan-tanggal”, Misalnya untuk 22 Januari 2001 dituliskan “2002-01-22”. Untuk penulisan waktu, formatnya adalah “jam-menit-detik”. Contoh, “13:55:07”. Data tanggal dan waktu bisa digabung penulisannya menjadi “2002-01-22 13:55:07”.
4. NULL. NULL sebenarnya bukan data, tapi dia mewakili sesuatu yang “tidak pasti”, “tidak diketahui” atau “belum ada nilainya”. Sebagai contoh dalam kehidupan sehari-hari Anda melakukan suatu survei berapa jumlah pengguna sistem operasi LINUX dan Windows di Indonesia. Selama survei belum tuntas maka data pastinya belum dapat diketahui. Oleh sebab itu, data tersebut bisa diwakili dengan NULL, alias belum diketahui.

2.6 PWA

Menurut Putu Alfred Crosby (2017) *Progressive Web Apps* (PWA) adalah sebuah aplikasi web yang menggunakan beberapa teknologi modern yang dapat menciptakan *user experience* seperti menggunakan *Native App* (Android/iOS). Berbeda dengan aplikasi *hybrid*, aplikasi *hybrid* lebih menggabungkan antara aplikasi *native* dan web. PWA sepenuhnya merupakan aplikasi web namun pengguna bisa merasakan pengalaman seperti menggunakan aplikasi *native*. Tujuan utama PWA ialah untuk memungkinkan *web developer* mengubah web yang sudah ada agar bisa berperilaku layaknya aplikasi *mobile native* tanpa melakukan banyak perubahan atau menambah *programmer* khusus. Tujuan ini bisa dicapai berkat kumpulan teknologi web. Ia bisa memanfaatkan kelebihan-kelebihan web dan aplikasi *mobile native* sekaligus. Aplikasi PWA bisa dijalankan secara *offline* atau dengan koneksi internet yang buruk sekalipun. Pengguna tidak lagi perlu memasang aplikasi PWA, namun ikon dari aplikasi tersebut bisa di pasang di *homescreen* pada *smartphone* untuk memudahkan akses

aplikasi. *Progressive Web Apps* sepenuhnya mengandalkan *browser*. Berikut ini merupakan karakteristik sebuah PWA:

1. *Progressive*

Bekerja untuk setiap pengguna. Tak peduli apapun *browser* yang digunakan, tidak masalah. Pasalnya, PWA dibangun dengan peningkatan progresif sebagai konsep intinya.

2. *Responsive*

Cocok dengan setiap faktor bentuk: perangkat *desktop*, seluler, tablet, atau apa saja yang muncul berikutnya.

3. *Connectivity independent*

Disempurnakan dengan *service worker* agar bisa bekerja *offline* atau pada jaringan berkualitas rendah.

4. *App-like*

Terasa seperti sebuah aplikasi untuk pengguna dengan interaksi dan navigasi bergaya-aplikasi karena mereka dibangun di atas model *shell application*.

5. *Fresh*

Selalu terkini berkat proses pembaruan *service worker*.

6. *Safe*

Disediakan melalui HTTPS untuk mencegah *snooping* dan memastikan materi belum dirusak.

7. *Discoverable*

Dapat diidentifikasi sebagai "aplikasi" berkat web manifes dan cakupan registrasi *service worker*. Ini memungkinkan mesin telusur untuk menemukannya.

8. *Re-engageable*

Dengan fitur pemberitahuan seperti *push notification*, dapat mengajak (*engaged*) kembali pengguna untuk menggunakan aplikasi.

9. *Installable*

Memungkinkan pengguna untuk "menyimpan" aplikasi yang mereka anggap paling berguna di layar beranda tanpa kerumitan *application store*.

10. *Easy to share*

Dapat dengan mudah dibagikan melalui URL. Pemasangannya pun mudah.

Seperti teknologi lain PWA juga memiliki kelebihan serta kekurangan. Berikut ini beberapa kelebihan yang dimiliki oleh PWA:

1. PWA adalah web sehingga tidak memerlukan pemasangan

Berbeda dengan aplikasi *mobile native* yang mana pengguna harus mengunduh dan memasang berkas .apk atau .ipa sebelum bisa digunakan, *progressive web apps* bisa dipakai tanpa mengunduh atau memasang berkas apapun. Meskipun begitu, jika ingin kita juga tetap bisa memasang PWA agar tampil di *homescreen*.

2. Tidak memerlukan *install update*

Saat *programmer* memperbaiki sebuah fitur atau menambah fitur baru di aplikasi *mobile native*, pengguna harus mengunduh berkas .apk atau .ipa yang baru agar bisa mendapatkan *update* yang dikirim. Dengan PWA, pengguna bisa mendapatkan *update* terbaru tanpa perlu mengunduh berkas apapun. Karena PWA adalah web, maka untuk mendapatkan *update* baru pengguna cukup me-*refresh* aplikasi tersebut.

3. Tidak perlu didistribusikan melalui *Store*

Google *Play Store* pada bulan Maret 2018 sudah memiliki lebih dari 3.3 juta aplikasi. Membuat aplikasi baru yang bersaing dengan 3.3 juta aplikasi bukan pekerjaan yang mudah. Beruntung aplikasi PWA tidak perlu didistribusikan lewat *Store*. Pengguna cukup mengunjungi *website* utama untuk kemudian bisa memanfaatkan penerapan teknologi yang sudah dilakukan oleh *developer*.

4. Bisa memanfaatkan fitur *native*.

Meski integrasinya tidak sempurna aplikasi *mobile native*, PWA bisa berkomunikasi dengan perangkat keras yang dimiliki oleh *smartphone* (tapi belum semua fitur bisa digunakan).

5. Bisa membuka halaman manapun lewat tautan.

Sekali lagi, karena PWA merupakan web, maka setiap halaman bisa dibuka dari mana saja, lewat sebuah tautan. seperti layaknya *website* biasa. Di aplikasi *mobile native*, kita harus melalui alur navigasi yang telah didesain dan tidak bisa memuat halaman manapun secara sembarang.

6. Bersifat universal.

Mendistribusikan aplikasi melalui *Store* harus mengikuti aturan-aturan yang diterapkan oleh pengelolanya. Dengan PWA, kita sebagai *developer* yang mengatur sendiri distribusi aplikasi.

7. Mengurangi penggunaan data.

Website e-commerce Konga, berhasil mengurangi penggunaan data sebesar 92% untuk *initial load* dibandingkan dengan aplikasi *mobile native*. Dengan pengurangan beban data yang besar artinya pengguna bisa memuat halaman yang diinginkan lebih cepat dan lebih hemat. Kelebihan ini akan sangat berguna di daerah dengan koneksi internet yang lambat.

Selain kelebihan-kelebihan tersebut, PWA juga memiliki kelemahan antara lain:

1. Masih ada fitur *native* yang belum bisa dimanfaatkan PWA

Pada pembahasan kelebihan PWA, kita melihat bahwa banyak fitur *native* yang sudah bisa dimanfaatkan melalui PWA. Akan tetapi, masih ada beberapa fitur penting yang belum bisa dipakai. Mayoritasnya berkaitan dengan masalah keamanan diantaranya aplikasi kontak, sms, telepon, kalender, *bookmark*, *alarm*, pengunci layar, Bluetooth, dan lain-lain.

2. Belum mendukung semua *browser*

Untuk saat ini baru Google Chrome, Firefox, Opera, dan *browser* Android Samsung yang bisa memanfaatkan seluruh teknologi PWA, sedangkan *browser* lain seperti Internet Explorer, Safari, Edge dan

browser-browser lainnya belum mengimplementasi keseluruhan fitur PWA.

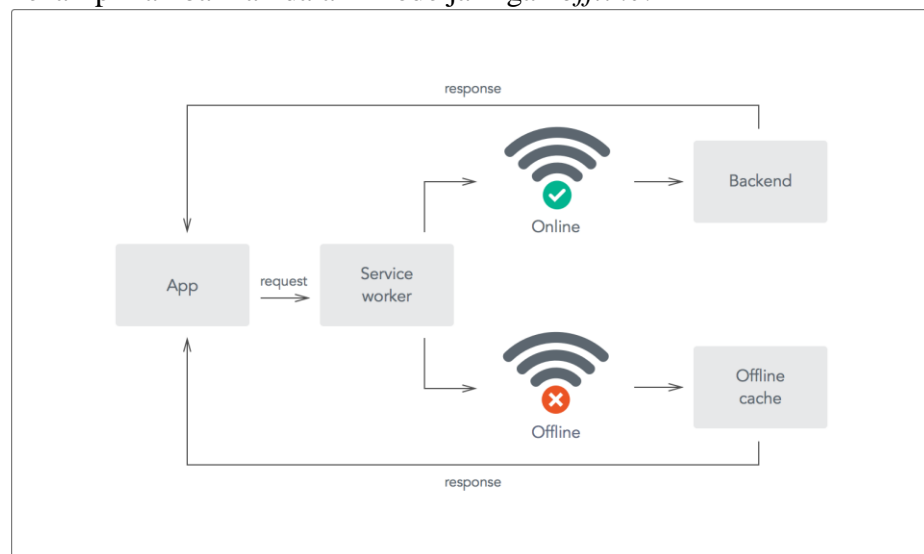
3. Tidak tersedia lewat aplikasi *Store*

Kelebihannya sudah kita bahas di atas yaitu pengguna tidak perlu mengunduh lewat aplikasi *Store* dan tidak perlu bersaing dengan jutaan aplikasi lain. Pengguna tidak mengunduh aplikasi lewat *Store*. Maka yang kita harus bisa adalah membawa pengguna tersebut ke beranda *website*. Kita juga tidak bisa memanfaatkan fitur promosi yang ditawarkan oleh *Store* tersebut.

Dalam membangun PWA, ada beberapa komponen yang harus digunakan. Berikut ini adalah penjelasan ringkas dari beberapa komponen PWA yang akan kita pelajari.

1. *Service Worker*

Service Worker adalah skrip yang dijalankan oleh *browser* di latar belakang, yang terpisah dengan skrip lain di halaman web *browser*. *Service worker* ditulis menggunakan bahasa pemrograman JavaScript, namun dipanggil dengan cara yang berbeda dari kode JavaScript pada umumnya. Dengan menggunakan *service worker*, kita dapat memanfaatkan *resource* yang telah disimpan di dalam cache untuk menampilkan bahkan dalam mode jaringan *offline*.



Gambar 2.6 *Service Worker*

2. *App shell*

Application shell atau biasa disingkat dengan *app shell* adalah kerangka antarmuka aplikasi yang dibangun oleh beberapa komponen halaman dan aset lainnya. Mereka disimpan lebih dahulu di dalam *cache* sehingga dapat tampil secara instan saat aplikasi dibuka. Umumnya komponen halaman yang menjadi bagian dari *app shell* adalah komponen yang selalu ada untuk semua halaman seperti *header*, *sidebar* dan *footer*. Adapun komponen dari konten halaman akan dimuat secara dinamis dan asinkron.

3. *Cache*

Cache adalah tempat penyimpanan lokal yang digunakan *browser* untuk menyimpan *resource* untuk dapat digunakan kembali pada pemanggilan selanjutnya tanpa perlu mengirim permintaan (*request*) lagi ke jaringan. Ada 2 jenis *cache* di *browser*, yaitu *Browser-managed cache* dan *Application-managed cache*. *Browser-managed cache* adalah *cache* sementara untuk menyimpan berkas unduhan *browser* secara otomatis. *Cache* jenis ini diatur oleh *browser* dan tidak tersedia dalam mode jaringan *offline*. Adapun *application-managed cache* adalah *cache* yang dibuat oleh aplikasi menggunakan *Cache API* dan terpisah dari *cache* yang dikelola oleh *browser*. *Cache* jenis inilah yang dapat kita gunakan untuk menyimpan *resource* dan dapat ditampilkan dalam mode jaringan *offline* melalui *service worker*.

4. *Web App Manifest*

Web app manifest adalah *file* JSON sederhana yang mengontrol bagaimana aplikasi ditampilkan dan dijalankan di sisi pengguna. Umumnya *file* ini diberi nama *manifest.json*. Saat aplikasi pertama kali dibuka di *browser*, *browser* akan membaca *file manifest*, mengunduh *resource* dan menampilkan konten.

5. *Fetch API*

Fetch API adalah salah satu diantara web API yang dapat digunakan untuk mengambil *resource* dari *server* di dalam dan lintas jaringan

secara asinkron. Fetch API sama seperti penggunaan API XMLHttpRequest namun memiliki beberapa fitur yang lebih baik.

6. *Progressive Enhancement*

Progressive enhancement atau peningkatan progresif adalah pendekatan dalam pengembangan web dimana kita memulai pengembangan dari fitur-fitur yang umum berjalan di semua *browser*. Kemudian secara bertahap kita meningkatkan fungsionalitas sesuai dengan kemampuan dari jenis *browser* yang digunakan oleh setiap pengguna. Setiap kali kita menggunakan suatu web API kita perlu memperhatikan apakah API tersebut sudah didukung oleh *browser* bawaan pengguna aplikasi kita. Kita juga perlu melakukan penanganan bilamana *browser* yang digunakan pengguna belum mendukung fitur web API yang kita gunakan. Misalnya melakukan *fallback* ke fitur sejenis yang telah didukung atau menampilkan pesan untuk memperbaharui versi *browser* atau menghimbau agar menggunakan selalu *browser* modern.

7. *PWA Architecture Styles*

PWA Architecture Styles maksudnya pendekatan yang digunakan dalam membangun PWA ditinjau dari teknologi *back-end* yang digunakan dan kebutuhan performa aplikasi. Beberapa istilah terkait hal ini diantaranya *app shell*, *server-side rendering*, *client-side rendering*, dan sebagainya.

8. IndexedDB

IndexedDB adalah sistem penyimpanan lokal berbasis NoSQL di *browser*. Kita dapat menyimpan data apapun di *browser* pengguna untuk keperluan aplikasi. Kamu dapat melakukan aksi pencarian, pembaharuan dan penghapusan data.

9. *Push Notification*

Notification adalah pesan *popup* yang muncul di perangkat pengguna. Aplikasi PWA kita mendukung pemunculan *notification* untuk memberitahu pengguna terkait pembaharuan tertentu di dalam aplikasi. Aplikasi PWA juga dapat menerima *event push* yang dikirim oleh *server* dan menampilkan pesan *popup* meskipun pengguna sedang tidak membuka aplikasi tersebut.

10. Workbox

Workbox adalah koleksi *library* dan *tool* yang dapat kita gunakan untuk meng-generate *file service worker*, *precaching*, *routing* dan *runtime-caching*. *Workbox* memudahkan kita dalam menulis kode PWA dengan lebih sederhana dan mudah dikelola.

2.7 Google Maps API

Menurut petanikode (2016), *Google Maps Application Programming Interface* (API) merupakan sebuah layanan peta dunia yang disediakan oleh Google yang memungkinkan kita membangun aplikasi dengan memanfaatkan Google API. Pada dasarnya *Google Maps API* adalah *library javascript*. Dengan menggunakan *Google Maps API* dapat menghemat waktu untuk membangun aplikasi peta digital yang handal. Untuk menggunakan *Google Maps API* tersebut maka pengguna harus mendaftarkan diri pada layanan Google API untuk mendapatkan *key* yang digunakan menampilkan peta pada *website*. Bentuk *key* adalah sebuah kalimat yang sudah di enkripsi seperti pada gambar 2.7 *Key* yang dimaksudkan adalah yang digaris bawahi dan di tebakkan.

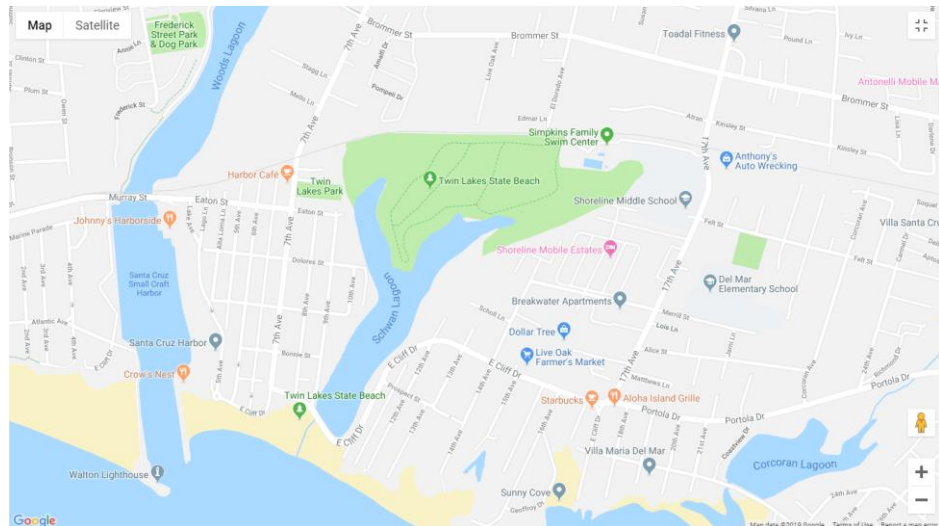
```
<script  
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyAE  
GGw0nGGqnYP2LR53UaoHHor HCSdVeQ&libraries=places&callback  
=initialize"></script>
```

Gambar 2.7 Google Maps API Key

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

1. Roadmap

Jenis map ini menampilkan peta biasa 2 dimensi biasa digunakan untuk menunjukan navigasi suatu lokasi. Pada gambar 2.8 adalah contoh tampilan peta *Roadmap*.



Gambar 2.8 Contoh google maps Roadmap

(Sumber: <https://developers.google.com/maps/documentation/javascript/maptypes>)

2. Satellite

Jenis map ini berfungsi untuk menampilkan foto satelit. Pada gambar 2.9 adalah contoh tampilan peta *Satellite*.



Gambar 2.9 Contoh google maps Satellite

(Sumber: <https://developers.google.com/maps/documentation/javascript/maptypes>)

3. Terrain

Jenis map ini menunjukkan relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi. Pada gambar 2.10 adalah contoh tampilan peta *Terrain*.



Gambar 2.10 Contoh google maps *Terrain*

(Sumber: <https://developers.google.com/maps/documentation/javascript/maptypes>)

4. *Hybrid*

Jenis map *hybrid* akan menunjukkan foto satelit yang diatasnya tergambar pula apa yang tampil pada *Roadmap* (jalan dan nama kota). Pada gambar 2.11 adalah contoh tampilan peta *Hybrid*.



Gambar 2.11 Contoh google maps *Hybrid*

(Sumber: <https://developers.google.com/maps/documentation/javascript/maptypes>)

2.8 GPS

Menurut mandalamaya (2015), GPS adalah sistem untuk menentukan letak dipermukaan bumi dengan bantuan penyelarasan sinyal satelit. Selain itu GPS juga dapat diartikan dengan sistem navigasi yang menggunakan satelit yang didesain agar dapat menyediakan posisi secara instan, kecepatan dan informasi

waktu di hampir semua tempat di muka bumi, setiap saat dan dalam kondisi cuaca apapun. Sedangkan alat untuk menerima sinyal satelit yang dapat digunakan oleh pengguna secara umum dinamakan *GPS Tracker* atau *GPS Tracking*, dengan menggunakan alat ini maka dimungkinkan user dapat melacak posisi kendaraan, armada ataupun mobil dalam keadaan *Real-Time*.

2.9 *Framework Angular*

AngularJS adalah sebuah *framework* yang di kembangkan atau di bangun oleh tim Google. Angularjs menggunakan metode MVC (*Model View Controller*) yang membuat *source code* aplikasi kita menjadi bersih dan mudah di kembangkan. Sampai saat ini angularjs sudah menjadi salah satu *framework javascript* yang paling populer dan sangat banyak di gunakan oleh para *developer* di seluruh dunia untuk membangun aplikasi. Kehandalan dari angularjs sendiri tidak bisa di ragukan lagi, salah satu faktornya karena di kembangkan oleh para tim ahli dari google.

2.10 *Framework Laravel*

Laravel merupakan salah satu *framework* PHP terbaik yang dikembangkan oleh Taylor Otwell pada tahun 2011. Otwell melihat bahwa belum menemukan adanya *framework* yang *up-to-date* dengan PHP. Oleh karena itu, laravel mensyaratkan PHP versi 5.3 ke atas. Di tahun 2015, laravel adalah *framework* yang paling banyak mendapatkan bintang di Github. Sekarang *framework* ini menjadi salah satu yang populer di dunia, tidak terkecuali di Indonesia. Laravel mengubah pengembangan *website* menjadi lebih elegan, ekspresif, dan menyenangkan, sesuai dengan jargonnya “*The PHP Framework For Web Artisans*”. Selain itu, laravel juga mempermudah proses pengembangan *website* dengan bantuan beberapa fitur unggulan, seperti *template engine*, *routing*, dan *modularity*.

Laravel menawarkan beberapa keuntungan ketika anda mengembangkan *website* menggunakan dasar *framework* ini. Pertama, *website* menjadi lebih *scalable* (mudah dikembangkan). Kedua, terdapat *namespace* dan tampilan yang membantu Anda untuk mengorganisir dan mengatur sumber daya *website*. Ketiga,

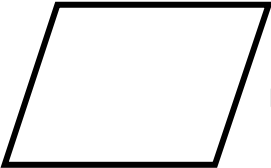

proses pengembangan menjadi lebih cepat sehingga menghemat waktu karena laravel dapat dikombinasikan dengan beberapa komponen dari *framework* lain untuk mengembangkan *website*.

2.11 Flowchart

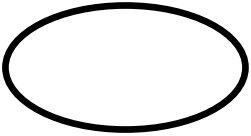
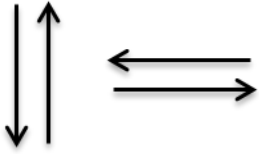

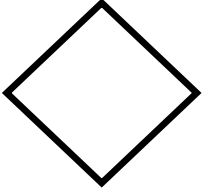
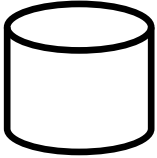

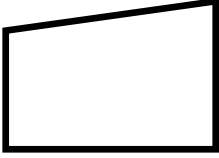
Menurut Dosenpinter (2020), *Flowchart* (diagram alir) adalah suatu jenis diagram yang merepresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam suatu sistem. *Flowchart* biasanya digunakan sebagai bukti dokumentasi untuk menjelaskan gambaran logis sebuah sistem yang akan dibangun kemudian diberikan kepada *programmer*, dengan begitu, *flowchart* dapat membantu untuk memberikan solusi terhadap masalah yang bisa saja terjadi dalam membangun sistem. Pada dasarnya, *flowchart* digambarkan dengan menggunakan simbol-simbol. Setiap simbol mewakili suatu proses tertentu, adapun untuk menghubungkan satu proses ke proses selanjutnya selanjutnya digambarkan dengan menggunakan garis penghubung.

Menurut ANSI (*American National Standards Institute*) dan ISO (*International Standards Organization*) (1970), terdapat beberapa simbol yang digunakan dalam *flowchart*. Simbol-simbol tersebut terdiri dari:

Tabel 2.2 Simbol *Flowchart*

Simbol	Keterangan
 <p>Input/Output</p>	Suatu operasi masukan atau luaran data.
 <p>Document</p>	mewakili data dalam bentuk <i>input hardcopy</i> atau <i>output</i> dari jenis apa pun.

Tabel 2.2 Simbol *Flowchart* (Lanjutan)

Simbol	Keterangan
 Terminal	berfungsi untuk menunjukkan awal, akhir, atau pemutusan pada garis aliran yang biasa.
 Flowlines	Panah dengan panjang berapa pun yang menghubungkan garis besar lainnya secara berurutan untuk menunjukkan urutan operasi atau data ("arah aliran").
 Process	Transformasi data, pergerakan data, dan operasi logika.
 Decision	Menunjukkan pertanyaan atau cabang dalam aliran proses. Biasanya, bentuk bagan alur Keputusan digunakan ketika ada 2 pilihan (Ya / Tidak)
 Disk Storage	Penyimpanan disk mewakili data disimpan pada perangkat disk jenis apa pun, terutama disk magnetik.
 Manual Offline Operation	Setiap <i>input</i> atau <i>output</i> yang dilakukan secara manual (tanpa bantuan komputer)
 Manual Input	Data yang diperoleh oleh kontrol manusia atas peralatan <i>online</i> yang dioperasikan secara manual.

2.12 Penelitian Terdahulu

Pada penelitian terdahulu yang dilakukan oleh Saputra (2017), yang membahas tentang pengembangan aplikasi pemesanan tiket bus pada PO. Puspa Jaya berbasis android yang bisa diakses secara *online*. Penelitian ini menghasilkan aplikasi pemesanan tiket bus Puspa Jaya berbasis android. Aplikasi ini mempermudah penumpang bus dalam melakukan transaksi pembelian tiket bus. Penumpang bus juga bisa menentukan tempat duduk dan waktu keberangkatan bus. Berdasarkan hasil pengujian sistem, didapati bahwa aplikasi dapat berjalan sesuai dengan kriteria pengujian. Aplikasi yang dibuat terbagi menjadi 2, yaitu dalam bentuk *mobile* android untuk pengguna dan dalam bentuk web untuk admin. Namun, terdapat kekurangan pada aplikasi yang telah dibuat yaitu kurangnya kompatibilitas terdapat *platform mobile* lain seperti iOS, *Windows Phone*, dan BlackBerry OS. Selain itu, tidak adanya teknologi untuk menyimpan data ke *database* lokal terlebih dahulu sehingga ketika koneksi internet tidak terhubung data tetap dapat ditampilkan dari *database*.

Menurut Pradana (2015), yang melakukan penelitian tentang Perancangan Aplikasi Alat Pemesanan Tiket Bus Dengan Perangkat Telepon Genggam Berbasis J2ME. Tujuan dari penelitian ini adalah membangun sistem pemesanan tiket bus berbasis J2ME yang dapat diterapkan di PO. Bejeu serta menguji kelayakan dan keandalan sistem informasi laboratorium ini. Pengguna dari sistem informasi laboratorium ini terdiri dari administrator, agen, dan pelanggan PO. Bejeu. Dari hasil penelitian, Telah berhasil dibuat sistem pemesanan tiket bus Bejeu sehingga pemesanan tiket menjadi lebih efektif, efisien dan aman, dan sistem ini layak digunakan di PO. Bejeu. Meninjau hasil pengujian dengan *blackbox testing* didapatkan persentasi keberhasilan sebesar 87,76%.

Menurut Nganga dan Tedy (2015), yang melakukan penelitian tentang Rancang Bangun Sistem Pemesanan Tiket Bus Kupang-Atambua Berbasis SMS Gateway. Penelitian ini bertujuan merancang bangun sistem pemesanan tiket bus berbasis SMS Gateway yang memberikan kemudahan layanan pemesanan tiket bus bagi masyarakat secara jarak jauh serta kemudahan mengetahui informasi penjadwalan bus dengan cepat. Aplikasi dibangun menggunakan metode *Unified Process*, dengan bahasa pemrograman Java NetBeansIDE 7.3.1, MySQL sebagai

database dan Gammu sebagai software untuk membantu pembangunan aplikasi *SMS Gateway*. Hasil dari penelitian ini adalah sebuah sistem pemesanan tiket bus yang memberikan alternatif baru proses pemesanan tiket bus secara baik tanpa ada permasalahan jarak dan waktu. Selain itu masyarakat juga dapat mengetahui informasi jadwal bus, harga tiket dan juga informasi bila terjadi perubahan jadwal keberangkatan bus.

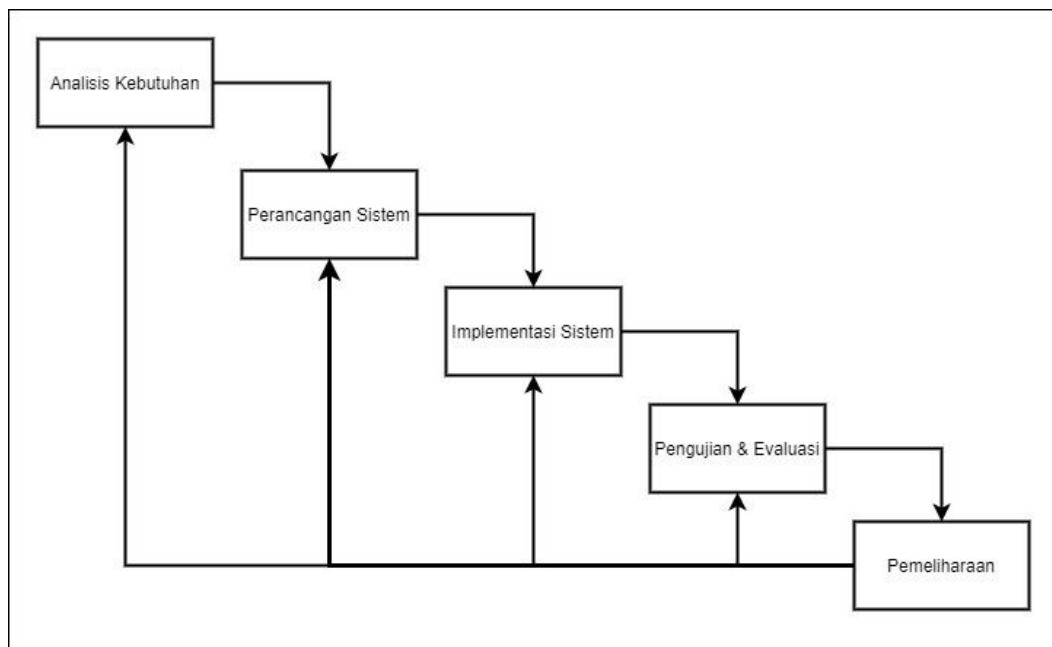
Berdasarkan penelitian sebelumnya yang dilakukan oleh Suranta (2012), Idris (2017), Guntoro (2007), Rahmat (2016), Rachmatullah (2015), Patrisia (2019), dan Wiraswandana (2016), yang membahas tentang pemesanan tiket bus. Pada penelitian-penelitian yang dilakukan tersebut, digunakan *website* sebagai media penelitian. Sementara, pada penelitian ini akan diterapkan konsep baru yang dinamakan PWA. Penelitian ini memiliki kelebihan dibandingkan dengan penelitian-penelitian terdahulu. Dengan menggunakan media *website*, maka koneksi internet sangat dibutuhkan untuk menjalankan media tersebut. Jika menggunakan PWA, masalah keterbatasan koneksi atau tidak adanya koneksi masih bisa diatasi.

Bab III

Analisis dan Perancangan Sistem

3.1 Alur Penelitian

Proses penelitian ini dirancang dengan menggunakan alur penelitian *waterfall*. Menurut Sommerville (2011), alur penelitian *waterfall* ini dibagi menjadi 5 tahapan yakni tahap analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian & evaluasi, dan dokumentasi. Gambar 3.1 menggambarkan tahapan dari alur penelitian *waterfall*.



Gambar 3.1 Alur Penelitian *Waterfall*

3.2 Analisis Masalah

Analisis kebutuhan dilakukan dengan tujuan untuk mengidentifikasi permasalahan-permasalahan yang ada pada sistem serta menentukan kebutuhan dari sistem yang akan dibangun. Dengan melihat permasalahan yang ada serta menyelesaikan dan memberikan solusi atas permasalahan yang ada merupakan tujuan dari sebuah penelitian. Oleh karena itu, tahapan ini bertujuan untuk mengetahui permasalahan utama serta mengetahui kebutuhan-kebutuhan yang diperlukan guna mendukung pembuatan aplikasi yang dapat menyelesaikan dan menjadi solusi dari permasalahan tersebut.

Berikut ini merupakan hasil analisis masalah yang diperoleh oleh peneliti :

- Sistem pembayaran *online* yang membutuhkan perangkat dan infrastruktur tambahan, dan masih banyak pengguna bus yang harus di sosialisasi tentang sistem *online* sebelum siap untuk diterapkan. Selain itu, ada beberapa kasus-kasus khusus di Indonesia yang belum bisa ditangani oleh sistem *online*.
- Tiket konvensional yang masih kurang efisien karena membutuhkan biaya banyak untuk percetakan tiket. Kemudian, belum adanya informasi tarif yang tertera pada tiket bisa memicu terjadinya kecurangan dalam penentuan tarif tiket. Pelaporan hasil pendapatan tiap bus juga membutuhkan waktu yang lama karena harus menghitung jumlah tiket yang terjual secara manual.
- Kurangnya informasi kepada penumpang tentang bus yang akan melewati halte tersebut. Informasi tersebut meliputi waktu sampai bus tiba ke halte, kondisi tempat duduk di bus (penuh atau tidak), dan juga jurusan dari bus tersebut.

Berdasarkan analisis masalah yang sudah diuraikan tersebut, maka akan dibuat sistem informasi *e-kondektur* untuk transaksi tiket bus. Pembuatan sistem ini diharapkan dapat menyelesaikan atau menjadi solusi akan permasalahan yang ada.

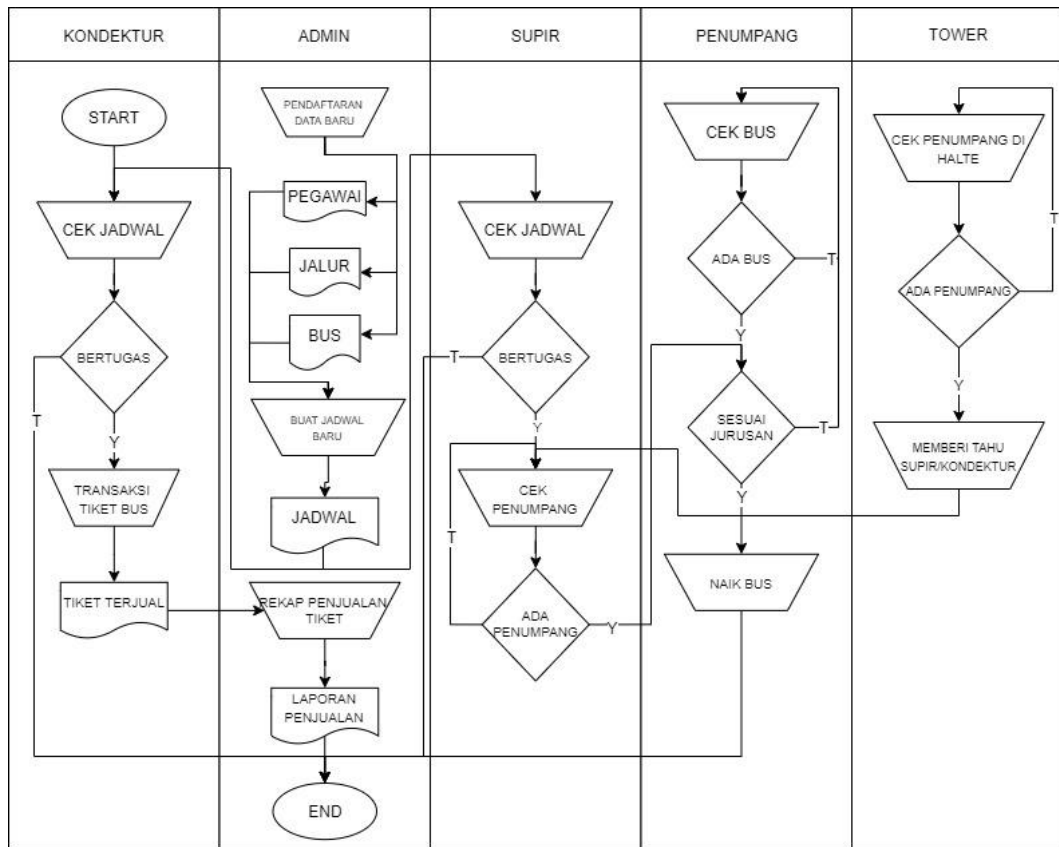
3.3 Perancangan Sistem

Bagian ini akan menjelaskan tentang perancangan perangkat lunak (*software*) pada sistem yang akan dibuat. Bagian ini akan memuat beberapa hal yang terdiri dari *flowchart* sistem, *data flow diagram* (DFD), basis data sistem, rancangan antarmuka sistem, garis besar aplikasi, dan diagram alir aplikasi.

3.3.1 Flowchart Sistem

Pada sistem yang akan dibuat terdapat 4 pihak yang mendapat akses aplikasi. *Flowchart* dibawah ini akan menjelaskan tentang bagaimana proses transaksi tiket bus secara manual (belum terkomputerisasi) yang melibatkan 5 pihak. Pada proses ini diawali dengan kondektur yang memeriksa jadwal bertugas, jadwal ini diperoleh dari admin. Setelah proses tersebut, kondektur yang sudah berada di dalam bus akan langsung mendatangi bangku penumpang untuk melakukan transaksi tiket bus dengan penumpang. Setelah tiket diberikan kepada

penumpang, kondektur akan mengambil sebagian dari tiket untuk nantinya akan dijadikan bukti bahwa telah terjadi transaksi penjualan tiket bus. bukti transaksi tersebut kemudian akan direkap oleh admin untuk dijadikan laporan penjualan tiket. Proses lain yang terjadi adalah proses supir yang mencari penumpang yang ingin naik bus dan juga penumpang yang mencari bus yang sesuai dengan tujuan dari penumpang. Pada proses ini, supir akan dibantu juga dengan pihak tower, pihak tower disini berfungsi sebagai pihak yang akan mengabari supir jika terdapat penumpang pada halte yang akan dilewati oleh bus.



Gambar 3.2 *Flowchart* transaksi tiket bus secara manual

Flowchart transaksi tiket secara manual adalah proses yang dilakukan tanpa adanya proses komputerisasi. Adapun proses yang akan dibuat pada sistem ini dengan cara menambahkan proses komputerisasi pada beberapa proses yang sebelumnya manual. *Flowchart* sistem yang sudah ditambahkan proses komputerisasi dapat dilihat pada lampiran A.

Terdapat 4 pihak yang terlibat pada *flowchart* sistem yang dibuat. 4 pihak tersebut terdiri dari admin, kondektur, supir, dan penumpang. Berikut penjelasan proses-proses yang terjadi pada *flowchart* di atas:

1. Admin

Awalnya admin akan melakukan *login* pada aplikasi menggunakan *username* dan *password* yang telah dibuat sebelumnya. setelah itu, admin akan memeriksa apakah data baru yang harus dimasukkan ke dalam sistem, seperti data pegawai (supir atau kondektur), data bus, data jalur, atau data jadwal. Data jadwal tugas dapat dibuat admin dengan menggabungkan data dari pegawai, jalur, dan bus.

2. Pegawai (Supir dan Kondektur)

Supir dan kondektur memiliki proses yang sama pada awalnya yakni pemeriksaan jadwal (untuk memeriksa jadwal tugas masing-masing) dan juga proses *login*. Proses supir dilanjutkan dengan melihat *marker* penumpang pada *map* untuk mengecek ada atau tidaknya penumpang di tepi jalan. Proses pada kondektur akan dilanjutkan dengan memasukkan data tiket bus yang akan dicetak. Kondektur juga dapat menerima/menolak *request* dari penumpang untuk menumpang bus. Kemudian, rekapan penjualan tiket bus juga dapat dilihat oleh kondektur pada aplikasinya.

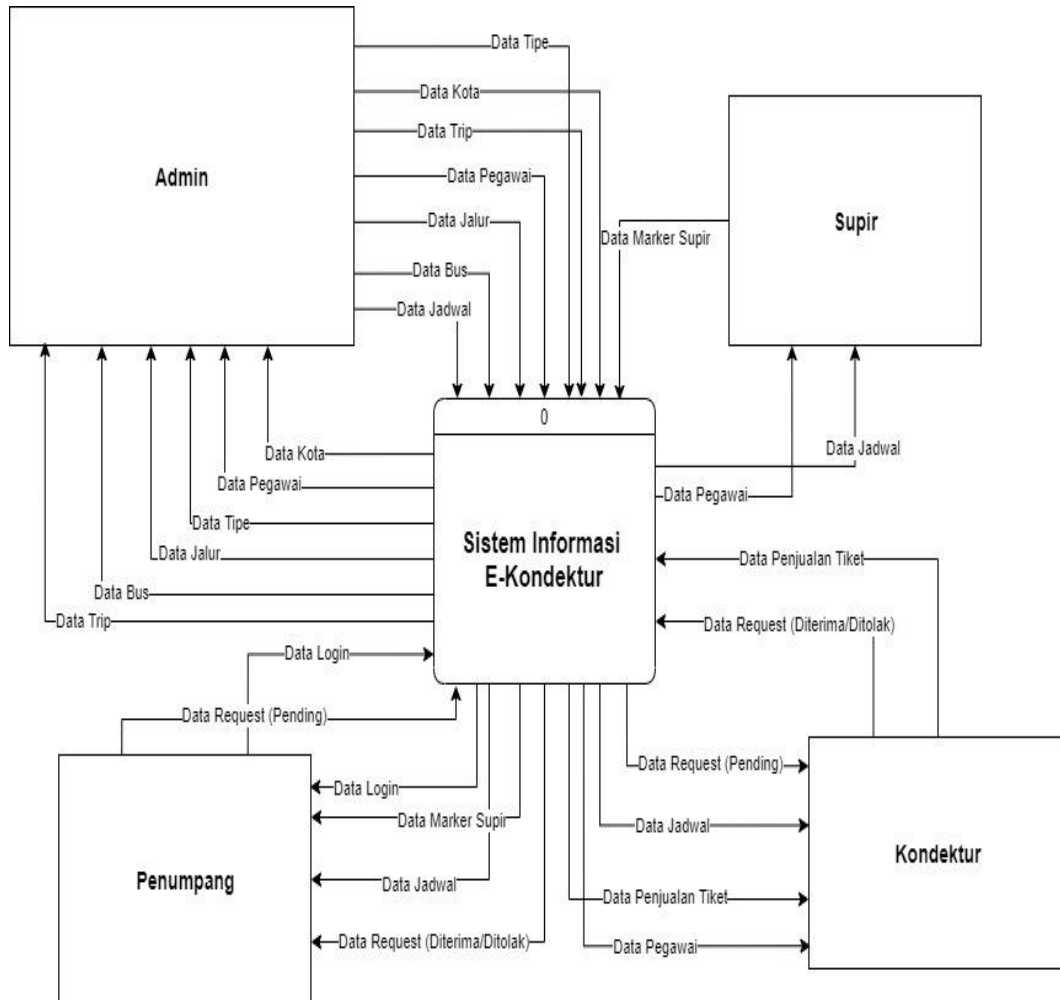
3. Penumpang

Penumpang yang belum memiliki akses pada aplikasi hari mendaftarkan diri dahulu. Setelah itu penumpang dapat melakukan proses *login*. Setelah masuk, penumpang dapat memilih data bus yang sesuai agar *marker* bus dapat muncul pada *map* di aplikasi (jika terdapat jadwal yang aktif saat itu). Penumpang dapat menekan *marker* bus untuk melakukan *request* kepada bus tersebut.

3.3.2 Data Flow Diagram (DFD)

Data Flow Diagram menjelaskan tentang aliran data yang berasal dari entitas-entitas pada aplikasi yang dibangun. Di dalam gambaran *Data Flow Diagram* tersebut akan menggambarkan siapa saja entitas yang terlibat dalam proses aplikasi. Selain itu, asal mula suatu aliran data hingga kemana aliran data

tersebut pergi akan dijelaskan disini. Gambar 3.3 menjelaskan tentang *context diagram*/DFD level 0 pada aplikasi. Terdapat 4 entitas yang terlibat pada DFD level 0 ini yakni admin, kondektur, supir, dan penumpang.



Gambar 3.3 *Context Diagram* Aplikasi

Data Flow Diagram level 1 dapat dilihat pada lampiran B. Pada gambar tersebut, dijelaskan proses-proses yang terjadi di dalam sistem. Pada DFD level 1 ini terdapat beberapa proses utama yang terjadi pada sistem. Proses-proses tersebut meliputi pengolahan input data sistem, pengolahan data jadwal bus, pengolahan data tiket bus, pengolahan data *marker bus*, dan pengolahan data *request*.

Pengolahan input data sistem akan melibatkan entitas admin dan penumpang. Proses ini berfungsi untuk melakukan pengolahan terhadap data-data yang diinputkan ke dalam sistem (data kondektur, data supir, data penumpang,

dan lainnya). Adapun proses pengolahan data jadwal bus, proses ini melibatkan semua entitas yang ada. Pada proses admin akan mengolah jadwal bus menggunakan data yang sebelumnya sudah ada (data kondektur, supir, bus, dan lainnya). Kemudian, jadwal bus tersebut akan dibagikan kepada kondektur, supir, dan penumpang. Proses lain yang ada pada sistem ialah pengolahan data tiket bus. Entitas kondektur terlibat pada proses ini. Proses ini berfungsi untuk mengolah data tiket yang telah diinputkan oleh kondektur ke dalam sistem. Proses pengolahan data *marker bus* melibatkan entitas penumpang dan supir. Supir akan memasukkan data *marker bus* pada sistem, kemudian penumpang dapat melihat *marker* tersebut pada sistem (untuk sisi penumpang). Proses yang terakhir adalah pengolahan data *request* yang melibatkan entitas penumpang dan kondektur. Proses ini diawali dengan penumpang yang melakukan *request* terhadap *bus* yang telah dipilih, kondektur lalu akan menanggapi *request* tersebut dengan menerima atau menolaknya. *Request* tersebut akan dikembalikan kepada penumpang dengan status sudah diterima atau ditolak.

Data Flow Diagram level 2 dapat dilihat pada lampiran B. Pada gambar tersebut, setiap proses pada DFD level 1 dijelaskan lebih mendalam. Proses tersebut meliputi Pengolahan Data Tiket Bus, Pengolahan Data Jadwal Bus, dan Pengolahan Data *Request*.

3.3.3 Basis Data Sistem

Pada sistem yang dibuat, akan dirancang sebuah *database* yang berisikan 9 tabel. Tabel tersebut terdiri dari penumpang, trip, penjualan, tipe, jalur, jadwal, bus, admin, dan pegawai. Tiap tabel memiliki relasi juga dengan tabel yang lainnya. Rancangan *Entity Relationship Diagram* (ERD) yang akan digunakan pada sistem pada lampiran D.

Penjelasan masing-masing atribut yang ada dalam tabel tiap basis data tersebut adalah sebagai berikut :

1. Tabel User

Tabel user merupakan tabel yang bertujuan untuk menyimpan data setiap penumpang. Pada tabel user terdapat struktur yang terbagi atas id sebagai *primary key*, name sebagai *field* nama penumpang, username sebagai field

username, password sebagai *field* kata sandi, email sebagai *field* alamat email, address sebagai *field* alamat rumah, phone_number sebagai *field* nomor telepon, Latitude sebagai *field* latitude penumpang, longitude sebagai *field* longitude penumpang, status sebagai *field* status data penumpang.

Tabel 3.1 Tabel User

No.	Nama Field	Tipe Data	Keterangan
1	id	int(11)	id penumpang
2	name	varchar(200)	nama penumpang
3	username	varchar(200)	username untuk aplikasi
4	password	varchar(200)	password untuk aplikasi
5	email	varchar(200)	alamat email
6	address	varchar(200)	alamat rumah
7	phone_number	varchar(200)	nomor telepon
8	latitude	float	latitude penumpang
9	longitude	float	longitude penumpang
10	status	varchar(10)	status data penumpang

2. Tabel Employee

Tabel employee merupakan tabel yang bertujuan untuk menyimpan data *login* setiap pegawai. Pada tabel employee terdapat struktur yang terbagi atas employee_id sebagai *primary key*, name sebagai *field* nama, username sebagai *field* username, password sebagai *field* kata sandi, email sebagai *field* alamat email, address sebagai *field* alamat rumah, phone_number sebagai *field* nomor telepon, occupation_id sebagai *field* jabatan pegawai, latitude sebagai *field* latitude pegawai, longitude sebagai *field* longitude pegawai, status sebagai *field* status data pegawai.

Tabel 3.2 Tabel Employee

No.	Nama Field	Tipe Data	Keterangan
1	employee_id	int(11)	id pegawai
2	name	varchar(200)	nama pegawai
3	username	varchar(200)	username untuk aplikasi
4	password	varchar(200)	password untuk aplikasi
5	email	varchar(200)	alamat email
6	address	varchar(200)	alamat rumah
7	phone_number	varchar(200)	nomor telepon
8	occupation_id	int(11)	id jabatan
9	latitude	float	latitude pegawai

Tabel 3.2 Tabel Employee (Lanjutan)

No.	Nama Field	Tipe Data	Keterangan
10	longitude	float	longitude pegawai
11	status	varchar(10)	status data pegawai

3. Tabel Schedule.

Tabel schedule merupakan tabel yang bertujuan untuk menyimpan data jadwal tugas setiap pegawai. Tabel schedule terbagi atas *schedule_id* sebagai *primary key*, *driver_id* sebagai *field* supir yang bertugas, *conductor_id* sebagai *field* kondektur yang bertugas, *bus_id* sebagai *field* bus yang digunakan saat bertugas, *schedule_start* sebagai *field* tanggal dan jam mulai bertugas, *schedule_end* sebagai *field* tanggal dan jam tugas selesai, *status* sebagai *field* status data jadwal.

Tabel 3.3 Tabel Schedule

No.	Nama Field	Tipe Data	Keterangan
1	<i>schedule_id</i>	int(11)	id jadwal
2	<i>driver_id</i>	int(11)	id supir
3	<i>conductor_id</i>	int(11)	id kondektur
4	<i>bus_id</i>	int(11)	id bus yang digunakan
5	<i>track_id</i>	int(11)	id Jalur yang dilalui
6	<i>schedule_start</i>	datetime	tanggal tugas dimulai
7	<i>schedule_end</i>	datetime	tanggal tugas selesai
8	<i>status</i>	varchar(10)	status data jadwal

4. Tabel Request

Tabel request merupakan tabel yang bertujuan untuk menyimpan data permintaan bus dari penumpang. Pada tabel request terdapat struktur yang terbagi atas *request_id* sebagai *primary key*, *user_id* sebagai *field* id penumpang, *schedule_id* sebagai *field* id jadwal, *trip_id* sebagai tujuan penumpang, *quantity* sebagai *field* jumlah kursi yang diminta penumpang, *date_time* sebagai *field* waktu saat penumpang melakukan request, *status* sebagai *field* status data request.

Tabel 3.4 Tabel Request

No.	Nama Field	Tipe Data	Keterangan
1	request_id	int(11)	id request penumpang
2	user_id	int(11)	id penumpang
3	schedule_id	int(11)	id jadwal
4	trip_id	int(11)	id kota tujuan
5	quantity	int(11)	jumlah kursi yang diminta
6	date_time	datetime	waktu dari request
7	status	varchar(100)	status data request

5. Tabel Bus

Tabel bus merupakan tabel yang bertujuan untuk menyimpan data bus. Pada tabel bus terdapat struktur yang terbagi atas bus_id sebagai *primary key*, police_number sebagai *field* plat nomor, bus_name sebagai nama bus, capacity sebagai *field* kapasitas, brand sebagai *field* merek, type_id sebagai *field* id tipe bus, status sebagai *field* status data bus.

Tabel 3.5 Tabel Bus

No.	Nama Field	Tipe Data	Keterangan
1	bus_id	int(11)	id bus
2	police_number	varchar(50)	plat nomor bus
3	bus_name	varchar(200)	nama bus
4	capacity	int(11)	kapasitas
5	brand	varchar(200)	merek
6	type_id	int(11)	id tipe bus
7	status	varchar(10)	status data bus

6. Tabel Track

Tabel track merupakan tabel yang bertujuan untuk menyimpan data jalur bus. Pada tabel track terdapat struktur yang terbagi atas track_id sebagai *primary key*, track_name sebagai *field* nama jalur, city_origin_id sebagai *field* kota asal keberangkatan bus, city_destination_id sebagai *field* kota tujuan keberangkatan bus, status sebagai *field* status data jalur.

Tabel 3.6 Tabel Track

No.	Nama Field	Tipe Data	Keterangan
1	track_id	int(11)	id jalur
2	track_name	varchar(200)	nama jalur
3	city_origin_id	float	id kota asal
4	city_destination_id	float	id kota tujuan akhir
5	status	varchar(10)	status data jalur

7. Tabel Trip

Tabel trip merupakan tabel yang bertujuan untuk menyimpan data kota yang dilalui pada setiap jalur. Pada tabel trip terdapat struktur yang terbagi atas trip_id sebagai *primary key*, fee sebagai *field* biaya perjalanan, track_id sebagai *field* id jalur, city_id sebagai *field* id kota, status sebagai *field* status data trip.

Tabel 3.7 Tabel Trip

No.	Nama Field	Tipe Data	Keterangan
1	trip_id	int(11)	id perjalanan
2	fee	int(11)	biaya perjalanan
3	track_id	int(11)	id jalur
4	city_id	int(11)	id kota
5	status	varchar(10)	status data trip

8. Tabel Ticket

Tabel ticket merupakan tabel yang bertujuan untuk menyimpan data tiket yang telah terjual. Pada tabel ticket terdapat struktur yang terbagi atas ticket_id sebagai *primary key*, schedule_id sebagai *field* id jadwal, trip_id sebagai *field* id trip, quantity sebagai *field* jumlah tiket yang dicetak, income sebagai *field* jumlah hasil penjualan tiket, datetime sebagai *field* waktu tiket dicetak.

Tabel 3.8 Tabel Ticket

No.	Nama Field	Tipe Data	Keterangan
1	ticket_id	int(11)	id tiket
2	schedule_id	int(11)	id jadwal
3	trip_id	int(11)	id trip
4	quantity	int(11)	jumlah tiket tercetak
5	income	Int(11)	hasil penjualan tiket
6	date_time	datetime	waktu dari tiket

9. Tabel City

Tabel city merupakan tabel yang bertujuan untuk menyimpan data kota. Pada tabel city terdapat struktur yang terbagi atas *city_id* sebagai *primary key*, *city_name* sebagai *field* nama kota, *latitude* sebagai *field* latitude kota, *longitude* sebagai *field* longitude kota, *status* sebagai *field* status data kota.

Tabel 3.9 Tabel City

No.	Nama Field	Tipe Data	Keterangan
1	city_id	int(11)	kota
2	city_name	varchar(200)	nama kota
3	latitude	float	latitude kota
4	longitude	float	longitude
5	status	varchar(10)	status data kota

10. Tabel Type

Tabel type merupakan tabel yang bertujuan untuk menyimpan data tipe bus. Pada tabel type terdapat struktur yang terbagi atas *type_id* sebagai *primary key*, *type_name* sebagai *field* nama tipe bus, *status* sebagai *field* status data tipe.

Tabel 3.10 Tabel Type

No.	Nama Field	Tipe Data	Keterangan
1	type_id	int(11)	id tipe bus
2	type_name	varchar(200)	nama tipe
3	status	varchar(10)	status data tipe

11. Tabel Occupation

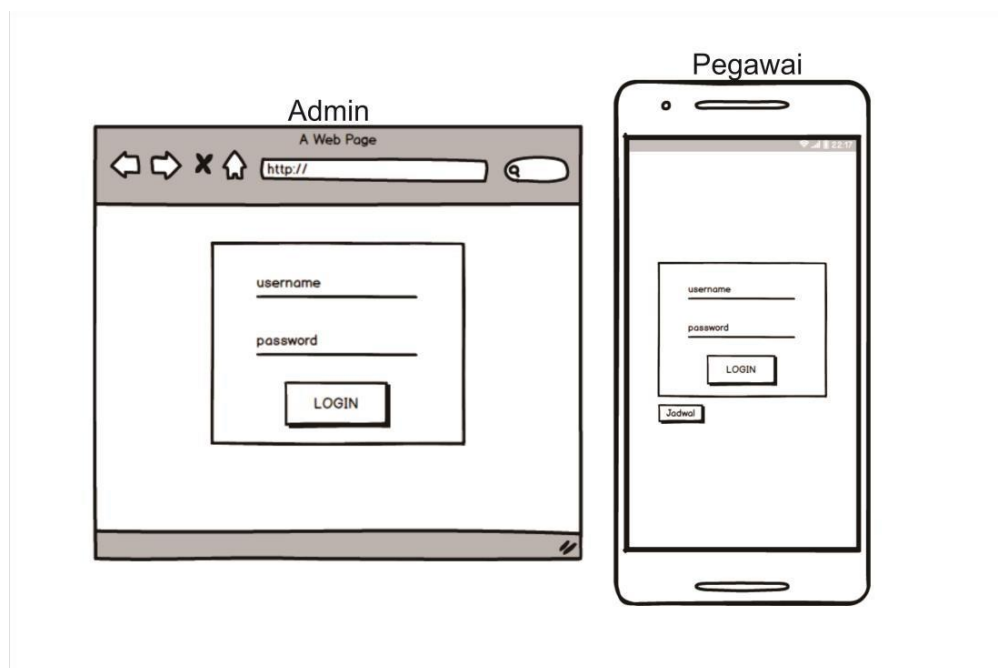
Tabel occupation merupakan tabel yang bertujuan untuk menyimpan data jabatan. Pada tabel occupation terdapat struktur yang terbagi atas *occupation_id* sebagai *primary key*, *occupation_name* sebagai *field* nama jabatan, *status* sebagai *field* status data jabatan.

Tabel 3.11 Tabel Occupation

No.	Nama Field	Tipe Data	Keterangan
1	occupation_id	int(11)	id jabatan
2	occupation_name	varchar(200)	nama jabatan
3	status	varchar(10)	status data jabatan

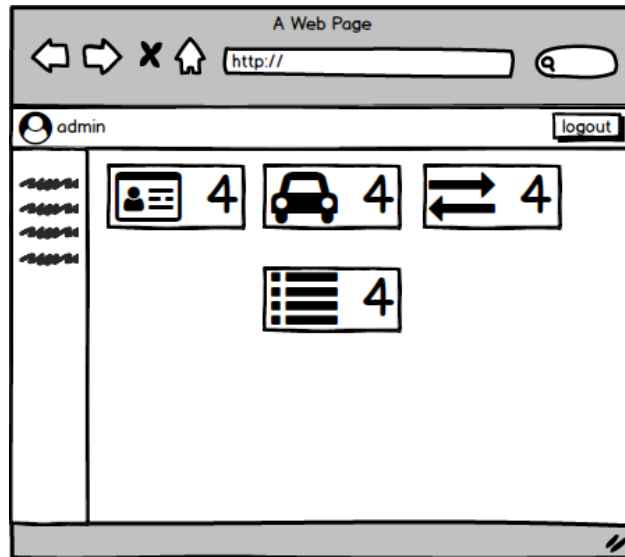
3.3.4 Rancangan Antarmuka Sistem

Pada tahap perancangan antarmuka sistem, terdapat bagian-bagian yang mewakili fitur- sistem informasi *e-kondektur* untuk transaksi tiket bus. Rancangan antarmuka yang pertama adalah halaman *login*, yang berfungsi sebagai jalan masuk admin/pegawai ke dalam sistem. Halaman ini dibuat untuk memproteksi agar tidak sembarang orang dapat masuk ke dalam web. Gambar 3.4 merupakan rancangan antarmuka untuk halaman *login* bagi admin dan juga pegawai bus. pada halaman *login* pegawai bus, akan ada tombol yang berfungsi untuk menunjukkan jadwal dari para pegawai.



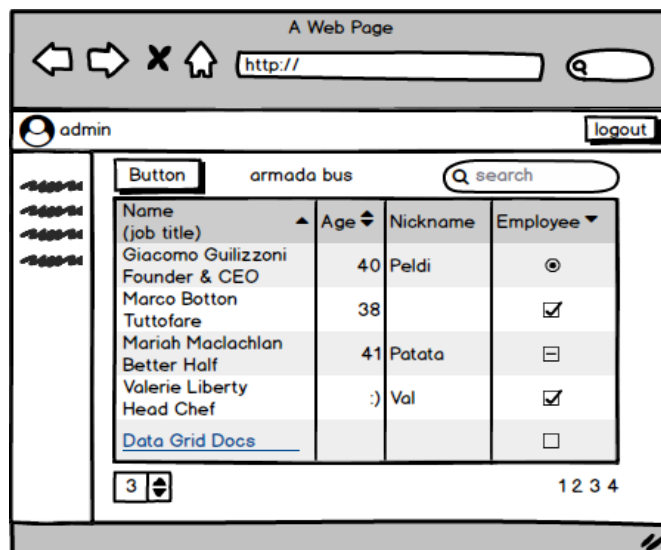
Gambar 3.4 Antarmuka Halaman *Login*

Rancangan antarmuka selanjutnya adalah halaman *dashboard* admin. Pada halaman ini admin bisa melihat jumlah pegawai bus, armada bus, jurusan bus, dan jadwal dari pegawai bus. Admin juga dapat mengakses rekapitulasi penjualan tiket bus dari halaman ini. Gambar 3.5 merupakan rancangan antarmuka untuk *dashboard* admin.



Gambar 3.5 Antarmuka Halaman *Dashboard* Admin

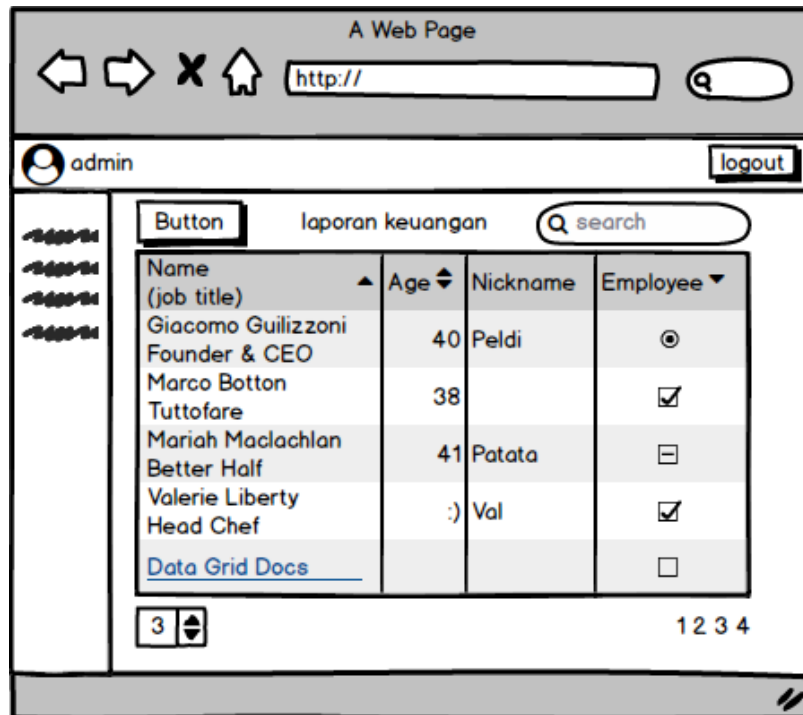
Rancangan antarmuka selanjutnya adalah halaman yang menampilkan data dari pegawai, bus, jurusan, atau jadwal. Masing-masing data akan ditampilkan pada menu yang berbeda namun dengan format antarmuka yang sama. Hal ini dilakukan untuk mempermudah admin dalam melihat data, dibandingkan jika menggunakan format antarmuka yang cenderung membingungkan. Gambar 3.6 merupakan rancangan antarmuka untuk halaman data.



Gambar 3.6 Antarmuka Halaman Data

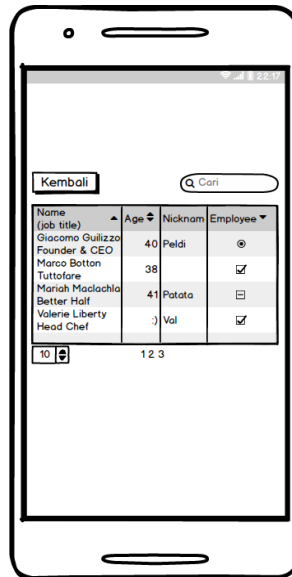
Rancangan antarmuka selanjutnya adalah halaman rekapitulasi penjualan tiket bus. Pada halaman ini admin bisa melihat daftar penjualan tiket bus yang

telah dilakukan oleh kondektur bus. Admin juga dapat memilih untuk mencetak rekapitulasi tersebut sehingga menjadi laporan penjualan tiket bus dengan tombol yang sudah disediakan. Gambar 3.7 merupakan rancangan antarmuka untuk halaman rekapitulasi penjualan tiket.



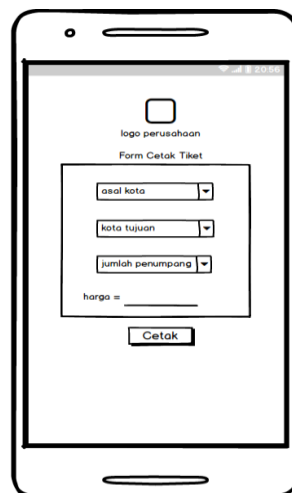
Gambar 3.7 Antarmuka Halaman Rekapitulasi Penjualan Tiket

Rancangan antarmuka selanjutnya adalah halaman jadwal untuk pegawai bus. Pada halaman ini para pegawai (supir dan kondektur) bisa melihat informasi tentang jadwal bekerja mereka. Pada halaman ini akan ditampilkan informasi tentang waktu (tanggal dan jam) pegawai akan bekerja dan juga bus yang akan ditumpangi. Informasi lain yang tercantum pada halaman ini adalah nama para pegawai yang bertugas. Jadwal ini sebelumnya telah ditentukan oleh admin. Gambar 3.8 merupakan rancangan antarmuka untuk halaman jadwal pegawai.



Gambar 3.8 Antarmuka Halaman Jadwal Pegawai

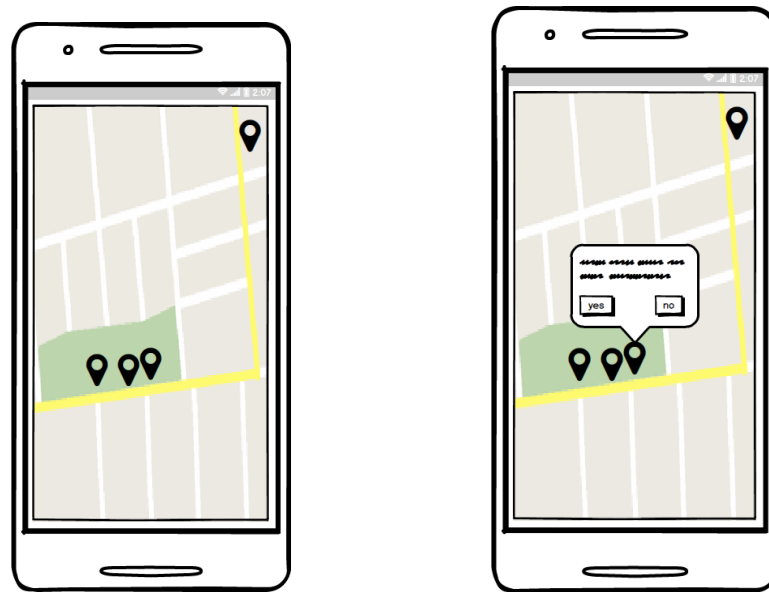
Rancangan antarmuka selanjutnya adalah halaman untuk kondektur saat *input* tiket bus. Pada halaman ini kondektur bus akan memasukkan data-data yang dibutuhkan sebelum mencetak tiket. Data tersebut meliputi kota asal penumpang ketika naik bus, kota tujuan penumpang, dan jumlah penumpang. Setelah memasukkan data-data tersebut, kondektur dapat menekan tombol cetak untuk mencetak tiket. Gambar 3.9 merupakan rancangan antarmuka untuk halaman *input* tiket.



Gambar 3.9 Antarmuka Halaman *Input* Tiket

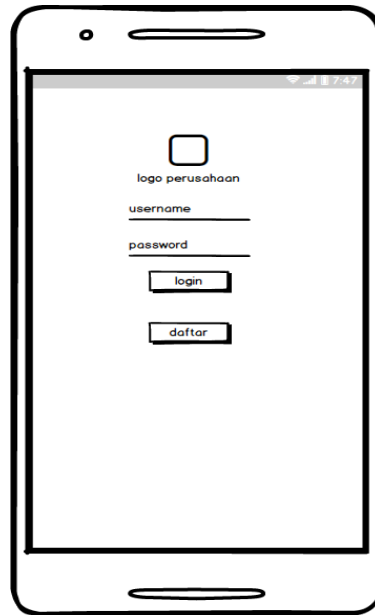
Rancangan antarmuka selanjutnya adalah halaman untuk supir. Pada halaman ini supir dapat melihat *marker* penumpang yang ingin menaiki bus yang

dikendarainya. Supir juga bisa memilih menghapus *marker* penumpang jika penumpang telah naik ke bus dengan menekan *marker* penumpang tersebut. Pada *marker* setiap penumpang yang muncul akan ada informasi tentang penumpang tersebut, contohnya jurusan bus yang ingin ditumpangi dan juga jumlah penumpang yang ingin naik bus tersebut (1 *marker* dapat mewakili lebih dari 1 penumpang). Gambar 3.10 merupakan rancangan antarmuka untuk halaman supir dan juga antarmuka saat akan menghapus *marker* penumpang.



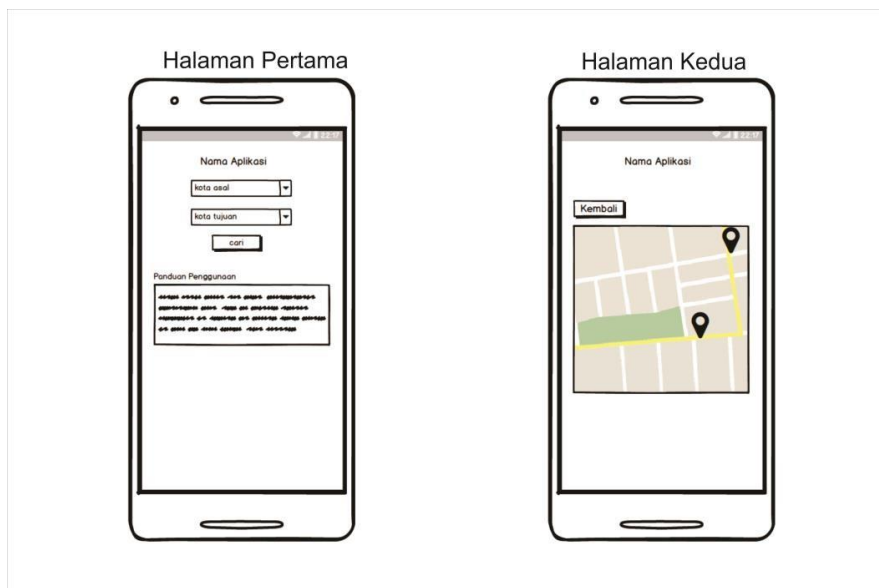
Gambar 3.10 Antarmuka Halaman Supir

Rancangan antarmuka selanjutnya adalah halaman *login* untuk penumpang. Pada halaman ini penumpang dapat masuk ke dalam aplikasi menggunakan *username* dan *password* masing-masing. Gambar 3.11 merupakan rancangan antarmuka untuk halaman *login* penumpang.



Gambar 3.11 Antarmuka Halaman *Login* Penumpang

Rancangan antarmuka yang terakhir adalah halaman untuk penumpang. Pada halaman pertama penumpang harus memilih kota asal keberangkatan dan juga kota tujuan untuk memunculkan *marker* bus yang sesuai dengan pilihan mereka. halaman kedua menampilkan posisi *marker* bus yang sesuai dengan jurusan pilihan penumpang. Pada *marker* akan muncul informasi dari bus, seperti tipe bus, jurusan bus, dan jumlah tempat duduk. Gambar 3.12 merupakan rancangan antarmuka untuk halaman penumpang.



Gambar 3.12 Antarmuka Halaman Penumpang

3.3.5 Garis Besar Aplikasi

Berikut ini merupakan garis besar dari aplikasi yang akan dibuat:

1. Admin selaku pengelola aplikasi yang dibuat memiliki hak untuk menambahkan data supir, Kondektur, bus, jalur, atau jadwal baru. Admin juga akan melakukan konfirmasi pada data yang diberikan oleh penumpang ketika mendaftarkan diri pada aplikasi.
2. Kondektur akan menerima data jadwal bertugas terbaru pada aplikasi. Data jadwal tersebut telah dibuat sebelumnya oleh admin. Pada saat kondektur telah melakukan input data tiket bus menggunakan aplikasi. Data ini bisa dilihat oleh admin. Kondektur bisa juga melihat rekapitulasi penjualan tiket yang telah dilakukan.
3. Supir juga akan menerima data jadwal baru seperti kondektur. Pada aplikasi ini supir bisa melihat ada atau tidaknya *marker* penumpang yang ingin naik bus tempat dia berada.
4. Penumpang akan melakukan pendaftaran pada aplikasi sebelum dapat menggunakan aplikasi. Data pendaftaran ini akan dikonfirmasi terlebih dahulu oleh admin. Pada aplikasi ini penumpang harus memilih kota asal (tempat mereka berada) dan juga kota tujuan mereka jika ingin melihat *marker* bus yang sesuai dengan pilihan kota mereka.

3.3.6 Diagram Alir Aplikasi

Berikut ini merupakan diagram alir dari proses-proses pada aplikasi yang dibuat:

1. Proses Pencetakan Tiket

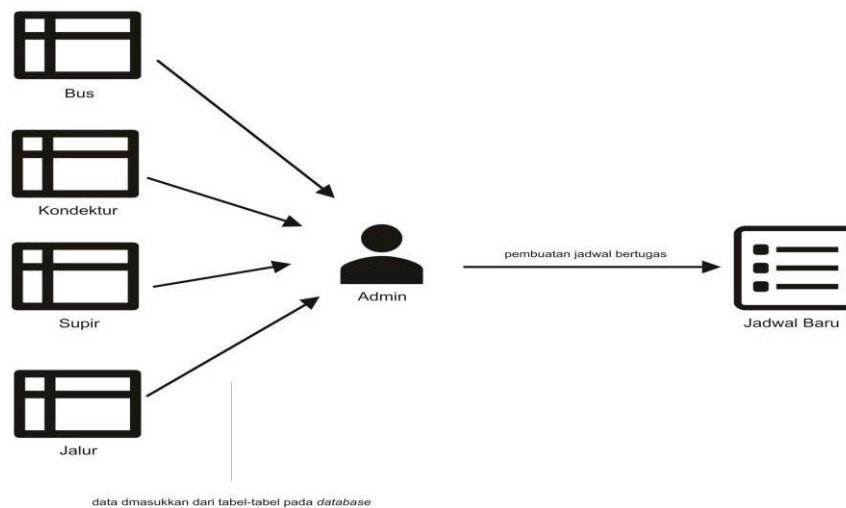
Proses ini diawali dengan kondektur yang mendatangi penumpang pada tempat duduk. Kondektur akan meminta beberapa data seperti asal kota, tujuan, dan juga jumlah tiket yang ingin dibeli kepada penumpang sebelum tiket akan dicetak. Setelah itu kondektur bisa mencetak tiket tersebut untuk nantinya akan diberikan kepada penumpang.



Gambar 3.13 Proses Pencetakan Tiket

2. Proses Pembuatan Jadwal

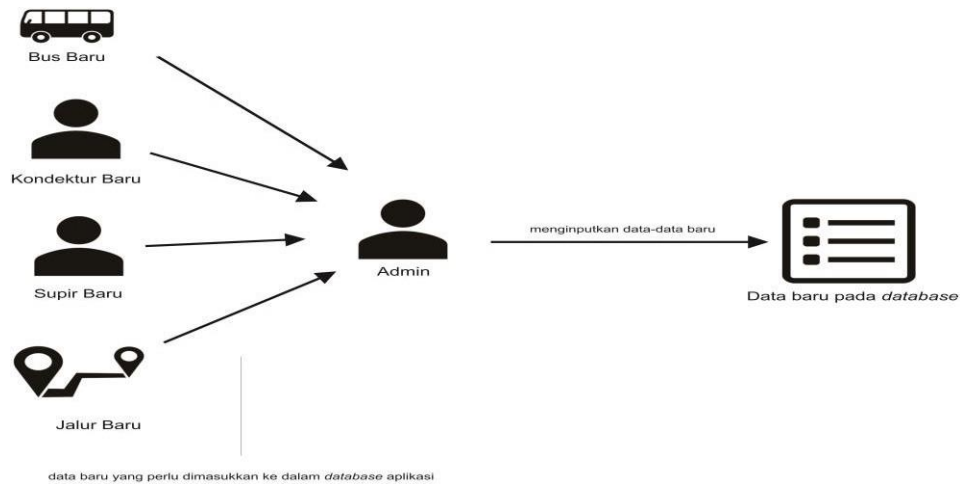
Proses ini diawali saat admin telah mendapat data pegawai (supir dan kondektur), data bus, dan data jalur. Data-data tersebut bisa didapatkan admin melalui proses input data yang sudah dilakukan sebelumnya. Dari data-data tersebut admin akan membuat jadwal bertugas bagi setiap pegawai.



Gambar 3.14 Proses Pembuatan Jadwal

3. Proses Penginputan Data

Proses ini dimulai saat ada data kondektur, supir, bus, dan jalur baru yang perlu dimasukkan ke dalam aplikasi. Proses ini dilakukan oleh pihak admin. Hasil dari proses ini adalah tabel-tabel *database* yang berisi daftar kondektur, supir, bus, dan jalur bus.



Gambar 3.15 Proses Penginputan Data

3.4 Implementasi Sistem

Pada tahap implementasi sistem merupakan tahap yang berlanjut setelah perancangan sistem selesai. Implementasi yang dilakukan adalah membuat *website* dengan menerapkan PWA. PWA merupakan teknologi yang menjembatani antara *native app* dan *mobile app*. Penggunaan PWA disini untuk membuat pengguna bisa merasakan pengalaman yang lebih baik ketika aplikasi memasuki mode *offline* (tidak ada koneksi internet), dibandingkan dengan menggunakan *native app* dan *web app*. Fitur dalam PWA dapat membuat aplikasi tetap berjalan dengan semestinya walaupun tidak ada koneksi internet.

Tahap ini juga akan diimplementasikan beberapa metode pembayaran yang akan menjadi bagian dari sistem. Metode pembayaran yang direncanakan untuk digunakan pada sistem yakni, jika memungkinkan untuk menerapkan sistem pembayaran *online* (OVO, DANA, GOPAY, LINKAJA) maka akan diterapkan ke dalam sistem. Namun, jika hal tersebut tidak memungkinkan maka akan dibuatkan sistem yang bersifat seperti metode-metode pembayaran *online*. Kemudian, metode yang terakhir adalah dengan melakukan pembayaran secara tunai atau *cash* jika kedua metode sebelumnya tidak/belum bisa diterapkan.

3.5 Pengujian & Evaluasi Sistem

Metode pengujian dilakukan setelah sistem yang dibuat telah selesai, tujuan adanya metode pengujian adalah untuk menguji atau menemukan

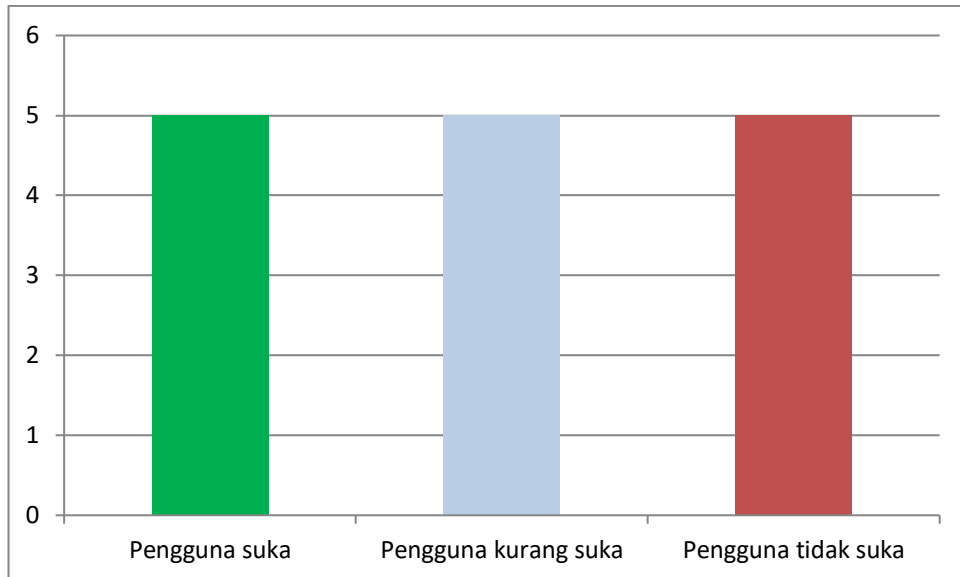
kesalahan-kesalahan dan memastikan bahwa hasil yang diberikan sesuai dengan hasil yang dibutuhkan. Pengujian penelitian ini menggunakan metode *white box* dan *black box*. Metode *white box* dilakukan dengan menguji apakah sistem pada program berjalan sesuai dengan yang diinginkan atau tidak. Pengujian ini dilakukan terhadap kode-kode aplikasi yang telah ditulis sebelumnya. Contoh dari pengujian *white box* adalah membuat suatu kondisi tertentu yang bertujuan untuk menemukan *bug* (cacat) pada aplikasi. Sedangkan metode *black box* digunakan untuk mengetahui apakah sistem pada program berfungsi dengan benar atau tidak. Metode *black box* akan menguji komponen dalam aplikasi sudah berfungsi dengan seharusnya. Contohnya, pengujian dengan melakukan penyimpanan suatu data inputan dari pengguna. Akan diuji apakah sistem berhasil melakukan penyimpanan terhadap data tersebut.

Saat melakukan pengujian, akan disebarkan juga angket yang berfungsi untuk menilai *user interface* (UI) dan *user experience* (UX) dari aplikasi. Angket yang disebarkan akan dibagi menjadi angket untuk pihak pengguna yakni penumpang, dan angket untuk yang mengoperasikan yakni supir dan kondektur.

Pengujian *black box* aplikasi ini akan dilakukan dengan implementasi langsung dibantu dengan pihak kondektur dan supir bus. Perusahaan yang direncanakan akan menjadi pihak pembantu dalam pengujian aplikasi ini adalah pegawai dari PO BUS BAGONG. Selain itu, akan dibagikan juga angket (setelah melakukan pengujian *black box*) kepada pihak penguji, yaitu kondektur dan supir.

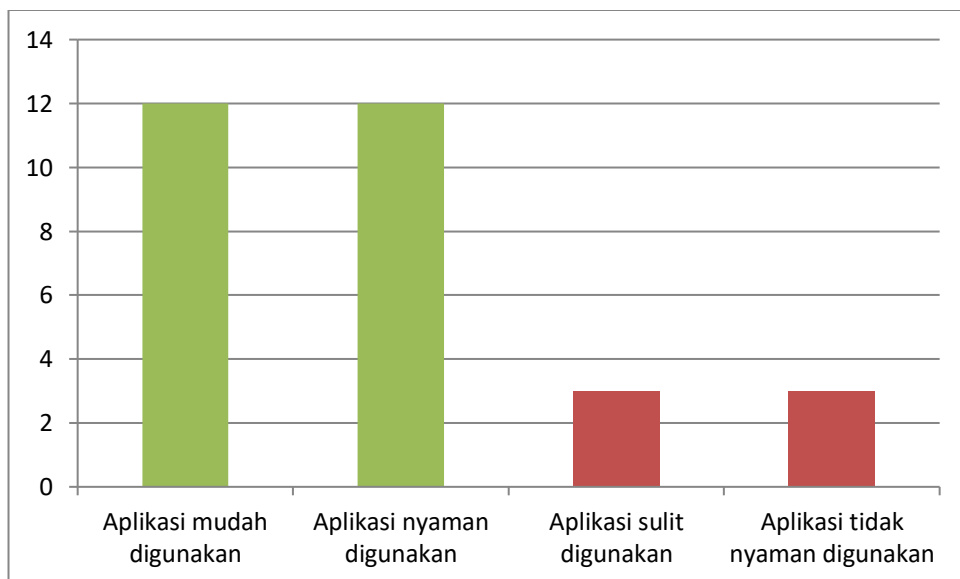
3.5.1 Hasil Pengujian

Setelah melakukan pengujian, maka hasil dari pengujian tersebut akan dinilai dengan menggunakan angket. Hasil dari angket-angket tersebut akan dijabarkan dengan menggunakan grafik seperti contoh berikut ini:



Gambar 3.16 Contoh Pengujian *User Interface*

Gambar 3.16 merupakan contoh tampilan grafik pengujian terhadap *user interface* aplikasi yang telah dibuat. Ada 3 poin yang menggambarkan nilai dari UI aplikasi, yaitu pengguna suka, kurang suka, dan tidak suka.



Gambar 3.17 Contoh Pengujian *User Experience*

Gambar 3.17 merupakan contoh tampilan grafik pengujian terhadap *user experience* aplikasi. Pada grafik ini akan ada 4 poin yang menggambarkan nilai dari UX aplikasi, yaitu aplikasi mudah digunakan, nyaman digunakan, sulit digunakan, dan tidak nyaman digunakan.

3.6 Pemeliharaan

Pada tahap ini, akan dilakukan proses pemeliharaan yang bertujuan untuk memperbaiki kesalahan pada sistem yang tidak ditemui pada tahap-tahap sebelumnya.

Bab IV

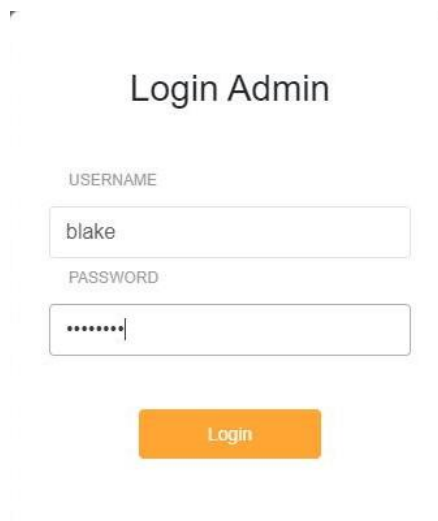
Hasil dan Pembahasan

4.1 Hasil Rancangan Aplikasi (*Back-End*)

Hasil rancangan aplikasi yang telah dibuat terbagi ke dalam dua bagian, yaitu bagian *back-end* (bagian admin) dan *front-end*. Pada bagian 4.1 ini akan membahas tentang antarmuka aplikasi *back-end* yang memiliki beberapa tampilan seperti antarmuka halaman *login*, antarmuka halaman *dashboard*, antarmuka halaman untuk menampilkan data-data (data *bus*, pegawai, jadwal, penumpang, dan lainnya), dan antarmuka untuk tambah data-data baru (data *bus*, pegawai, jadwal, penumpang, dan lainnya). Antarmuka pada *front-end* akan dibahas pada subbab 4.2.

4.1.1 Antarmuka Halaman *Login Admin*

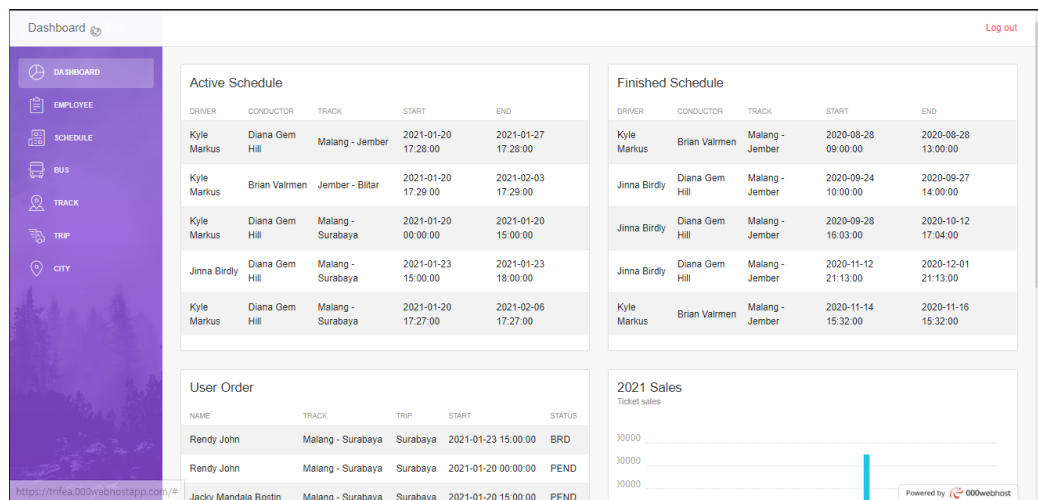
Pada halaman ini pihak yang berlaku sebagai admin akan melakukan *login* pada aplikasi. Hal pertama yang dilakukan adalah memasukkan *username* dan *password* pada kolom yang tersedia. setelah itu sistem akan memeriksa apakah *username* atau *password* yang dimasukkan sudah sesuai dan ada pada *database* atau tidak. Jika benar, maka sistem akan mengarahkan admin kepada halaman selanjutnya, yaitu halaman *dashboard*. Pada gambar 4.1 merupakan tampilan antarmuka halaman *login* pada pihak admin.



Gambar 4.1 Tampilan Antarmuka Halaman *Login Admin*

4.1.2 Antarmuka Halaman *Dashboard*

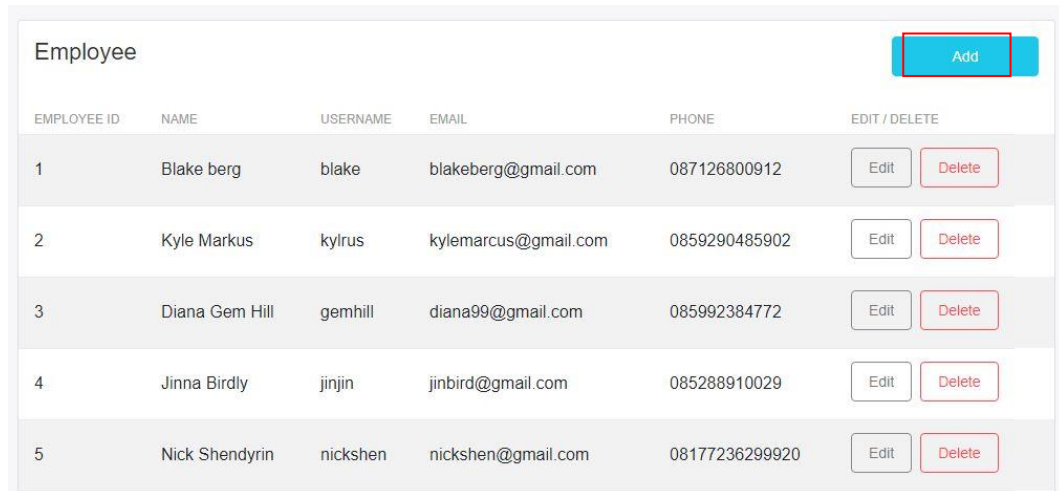
Halaman ini adalah halaman setelah admin berhasil melakukan *login*. Pada halaman ini terdapat beberapa bagian seperti, bagian navigasi untuk data-data yang ada pada aplikasi (data *bus*, pegawai, jadwal, dan lainnya), bagian yang menampilkan beberapa data-data pada aplikasi, dan juga tombol untuk melakukan *logout*. Pada gambar 4.2 merupakan tampilan antarmuka halaman *dashboard*.



Gambar 4.2 Tampilan Antarmuka Halaman *Dashboard*

4.1.3 Antarmuka Halaman *Data*

Pada halaman ini akan ditampilkan data-data (data *bus*, pegawai, jadwal, penumpang, dan lainnya) yang terdapat pada aplikasi. Data-data tersebut akan ditampilkan ke dalam bentuk tabel. Bentuk tabel dipilih untuk mempermudah admin dalam melihat informasi tiap data-data pada aplikasi. Terdapat juga tombol *Add* yang akan mengarahkan admin untuk menambahkan data baru (sesuai dengan tabel yang ditampilkan) pada aplikasi. Pada gambar 4.3 merupakan tampilan antarmuka halaman data.



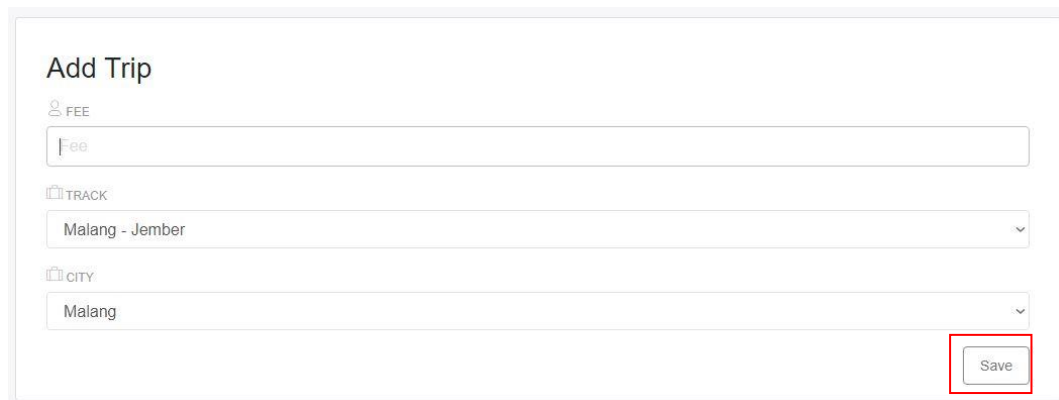
The screenshot shows a web interface for managing employees. At the top right, there is a blue 'Add' button with a red border. Below it is a table with the following columns: EMPLOYEE ID, NAME, USERNAME, EMAIL, PHONE, and EDIT / DELETE. The table contains five rows of employee data. Each row has 'Edit' and 'Delete' buttons in the last column.

EMPLOYEE ID	NAME	USERNAME	EMAIL	PHONE	EDIT / DELETE
1	Blake berg	blake	blakeberg@gmail.com	087126800912	<button>Edit</button> <button>Delete</button>
2	Kyle Markus	kylrus	kylemarcus@gmail.com	0859290485902	<button>Edit</button> <button>Delete</button>
3	Diana Gem Hill	gemhill	diana99@gmail.com	085992384772	<button>Edit</button> <button>Delete</button>
4	Jinna Birdly	jinjin	jinbird@gmail.com	085288910029	<button>Edit</button> <button>Delete</button>
5	Nick Shendyrin	nickshen	nickshen@gmail.com	08177236299920	<button>Edit</button> <button>Delete</button>

Gambar 4.3 Tampilan Antarmuka Halaman Data

4.1.4 Antarmuka Halaman Tambah Data Baru

Halaman ini akan tampil setelah admin menekan tombol *Add* pada halaman sebelumnya. Pada halaman ini akan muncul *field* yang menjadi tempat admin untuk mengisi informasi apa saja pada data baru yang akan ditambahkan pada aplikasi. Lalu pada bagian bawah halaman ini terdapat tombol *Save* yang akan menyimpan data baru ini. Pada gambar 4.4 merupakan tampilan antarmuka halaman tambah data baru.



The screenshot shows a web form titled 'Add Trip'. It contains three input fields: 'FEE' (a text box), 'TRACK' (a dropdown menu with 'Malang - Jember' selected), and 'CITY' (a dropdown menu with 'Malang' selected). At the bottom right, there is a 'Save' button with a red border.

Gambar 4.4 Tampilan Antarmuka Halaman Tambah Data Baru

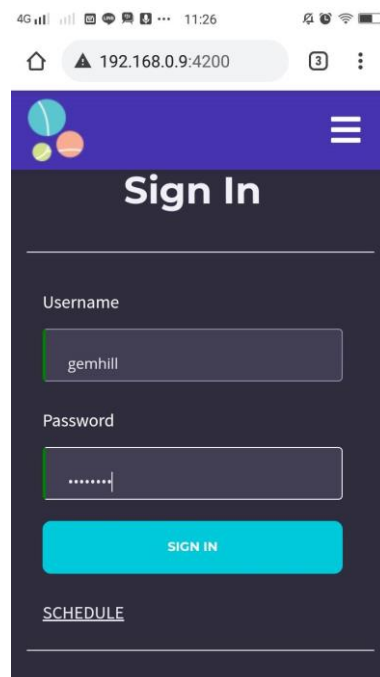
4.2 Hasil Rancangan Aplikasi (*Front-End*)

Untuk bagian *front-end* dibagi menjadi 3 bagian, yaitu bagian supir, kondektur, dan penumpang. Pada bagian supir akan memiliki tampilan aplikasi seperti, tampilan halaman *login*, tampilan halaman jadwal, dan tampilan halaman *map*. Bagian kondektur akan memiliki tampilan aplikasi seperti, tampilan halaman

login, tampilan halaman jadwal, dan tampilan halaman cetak tiket. Tampilan *front-end* yang terakhir adalah pada bagian penumpang. Bagian penumpang akan memiliki tampilan untuk halaman *login*, halaman registrasi aplikasi, dan halaman cari *bus*.

4.2.1 Antarmuka Halaman *Login* (Pegawai)

Antarmuka berikut merupakan halaman *login* untuk pegawai (supir dan kondektur). Pada halaman ini pegawai harus memasukkan *username* dan *password* (sama seperti pada halaman *login* admin) yang sesuai dengan data pada *database* aplikasi. Pada halaman ini juga pegawai bisa melihat jadwal mereka masing-masing dengan menekan tombol *burger* dan memilih *schedule* atau bisa menekan tombol yang berada pada bagian bawah tombol *sign in*. Pada gambar 4.5 merupakan tampilan antarmuka halaman *login* pegawai.

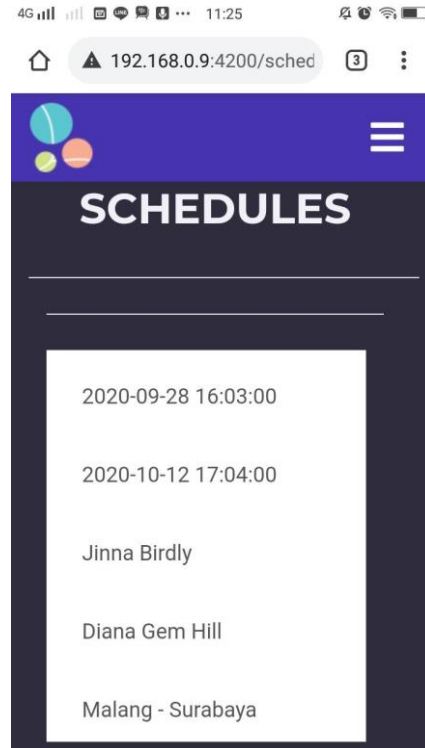


Gambar 4.5 Tampilan Antarmuka Halaman *Login* Pegawai

4.2.2 Antarmuka Halaman Jadwal (Pegawai)

Halaman ini bisa diakses sebelum pegawai masuk ke dalam aplikasi. Pada halaman ini pegawai bisa melihat jadwal bertugas yang sedang aktif (dijalankan). Halaman ini berisi informasi jadwal pegawai yang (sedang) bertugas seperti

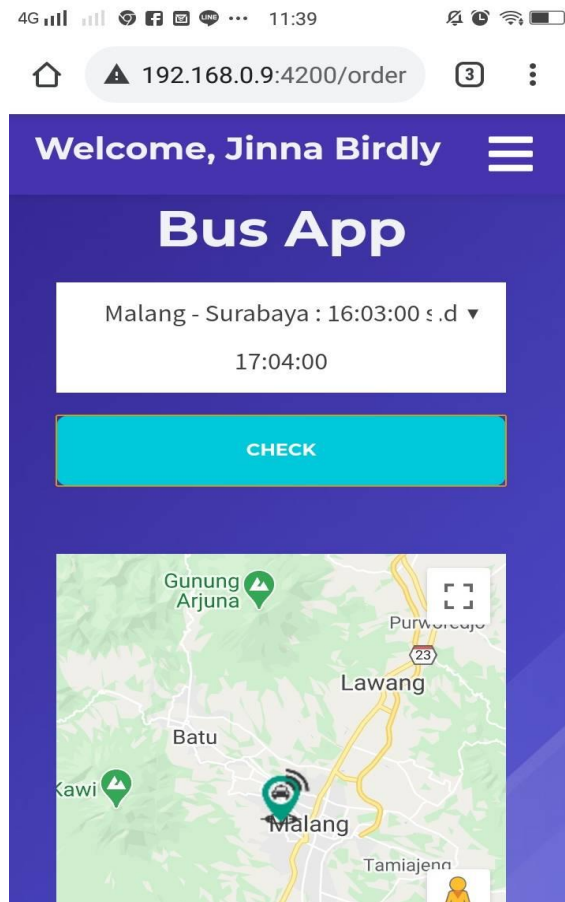
waktu (tanggal dan jam) dimana jadwal tersebut dimulai, waktu jadwal tersebut berakhir, nama supir, nama kondektur, dan juga jurusan dari *bus* tersebut. Pada gambar 4.6 merupakan tampilan antarmuka halaman jadwal.



Gambar 4.6 Tampilan Antarmuka Halaman Jadwal

4.2.3 Antarmuka Halaman *Map* (Supir)

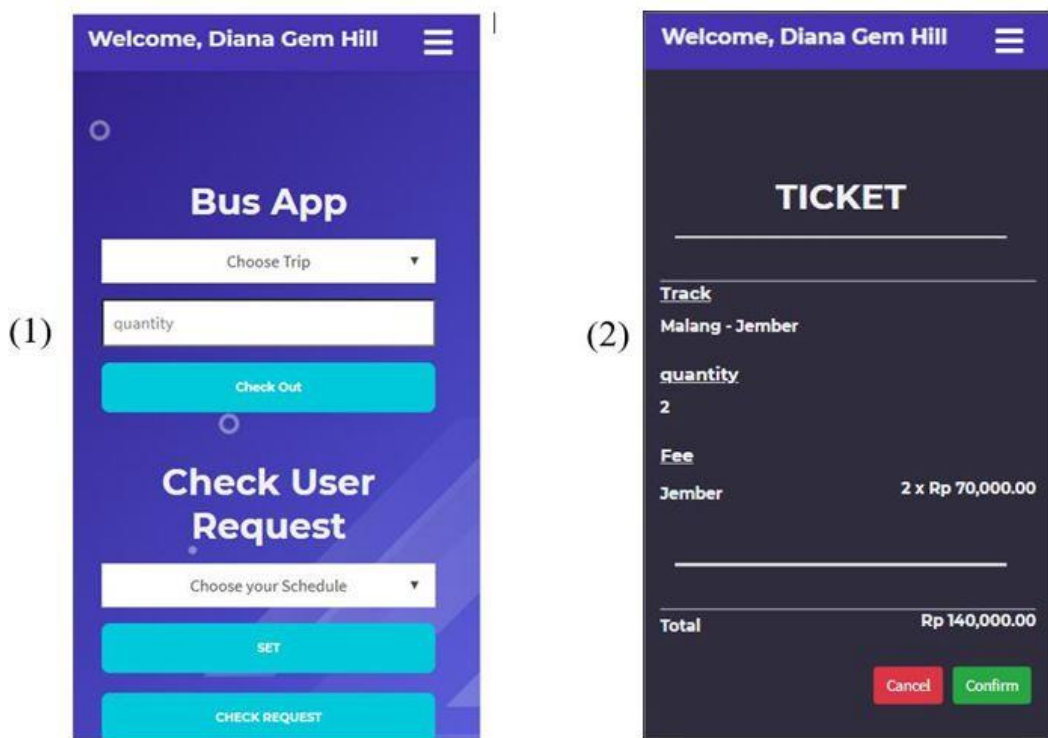
Setelah berhasil melakukan proses *login*, pegawai akan diarahkan ke halaman utama aplikasi masing-masing (pada antarmuka kali ini adalah halaman utama pada sisi supir). Supir akan mendapatkan pilihan jurusan yang sedang tersedia pada waktu tersebut. Setelah supir selesai memilih jadwal, supir harus menekan tombol *Check* untuk menampilkan *marker* dan *map* lokasi supir saat ini. Posisi *marker* akan diperbarui secara *realtime* sesuai dengan posisi *user* atau supir. Pada gambar 4.7 merupakan tampilan antarmuka halaman *map*.



Gambar 4.7 Tampilan Antarmuka Halaman Map

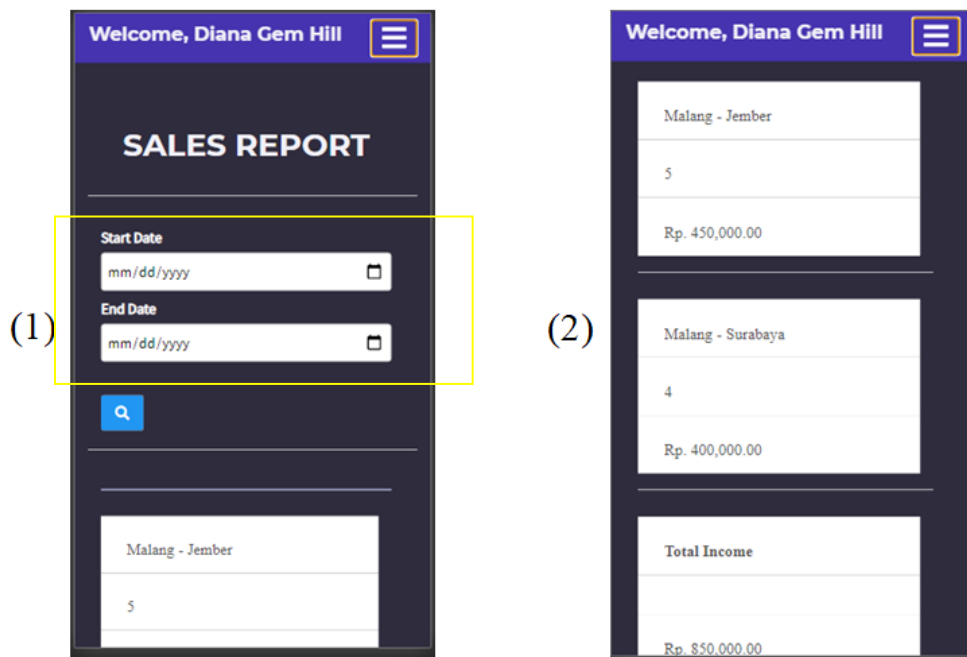
4.2.4 Antarmuka Halaman Tiket (Kondektur)

Antarmuka dibawah ini adalah halaman yang ditampilkan setelah proses *login* berhasil dilakukan oleh pihak kondektur. Pada halaman ini, kondektur akan bertugas untuk mendata dan mencetak tiket para penumpang. Kondektur akan diminta untuk memasukkan tujuan penumpang, dan jumlah tiket yang akan dicetak (1). Kemudian kondektur harus menekan tombol *check out* agar diarahkan ke halaman tampilan tiket yang akan dicetak (2). Setelah itu, kondektur bisa langsung mencetak tiket tersebut (menggunakan tombol *confirm*) atau jika mengalami salah dalam pengisian data, kondektur bisa kembali ke halaman pengisian data (dengan tombol *cancel*). Total harga tiket merupakan hasil dari *bus fee* ditambah *travel fee* dikali dengan *quantity*. Pada gambar 4.8 merupakan tampilan antarmuka halaman tiket.



Gambar 4.8 Tampilan Antarmuka Halaman Tiket

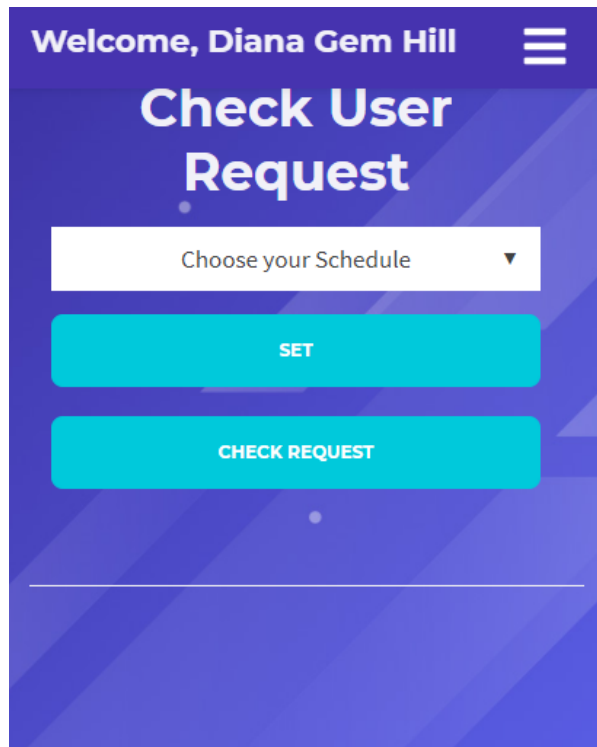
Kondektur bisa juga melihat hasil transaksi penjualan tiket pada halaman *report*. Halaman ini akan menampilkan hasil-hasil penjualan tiket bus. Pada halaman *report* ini akan ditampilkan hasil rekapitan penjualan tiket yang telah dilakukan oleh sang kondektur. Data hasil rekapitan dapat di *filter* oleh kondektur jika ingin menampilkan data rekapitan penjualan pada tanggal lainnya atau keseluruhan hasil penjualan(1). Pada halaman ini terdapat data-data seperti jurusan bus, jumlah tiket terjual, dan juga pemasukan, pemasukkan pada halaman ini terbagi atas 2 yakni pemasukkan tiap jurusan bus dan pemasukkan keseluruhan jurusan(2). Gambar 4.9 merupakan antarmuka halaman *report* pada kondektur.



Gambar 4.9 Antarmuka Halaman *Report*

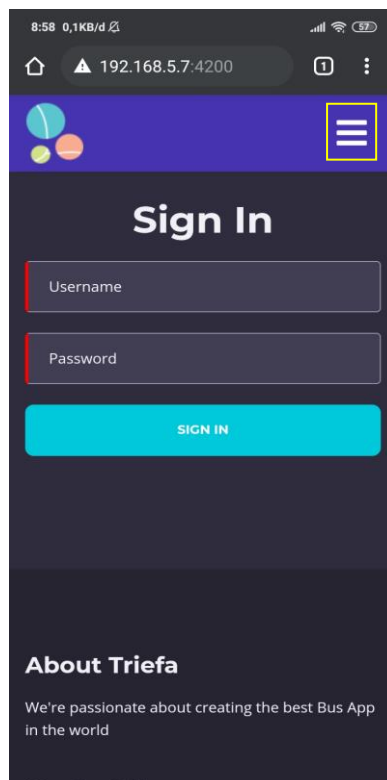
Selain mencetak tiket, kondektur dapat juga mengonfirmasi *request* dari pihak penumpang. Kondektur harus memilih jurusan yang tersedia pada waktu itu untuk memunculkan *map* dan *marker* dari penumpang. Setelah *map* dan *marker* berhasil tampil, kondektur dapat menekan *marker* penumpang pada *map* tersebut untuk mengonfirmasi atau menolak *request* yang telah dilakukan. Jika kondektur menolak *request* tersebut maka otomatis *marker request* tersebut akan hilang dari *map*. Selain untuk mengonfirmasi, kondektur juga dapat melihat informasi nama dan nomor telepon dari penumpang saat kondektur menekan *marker* di dalam *map*. Gambar 4.10 merupakan tampilan antarmuka halaman *confirm request*.

Pada *widget* ini kondektur juga dapat melakukan pengaturan jadwal. *Combo box* pada jadwal tersebut berisi *list* jadwal aktif dari kondektur tersebut. Jadwal yang telah diatur akan tersimpan dan secara otomatis menjadi jadwal aktif untuk kondektur tersebut.



Gambar 4.10 Tampilan Antarmuka Halaman *Confirm request*

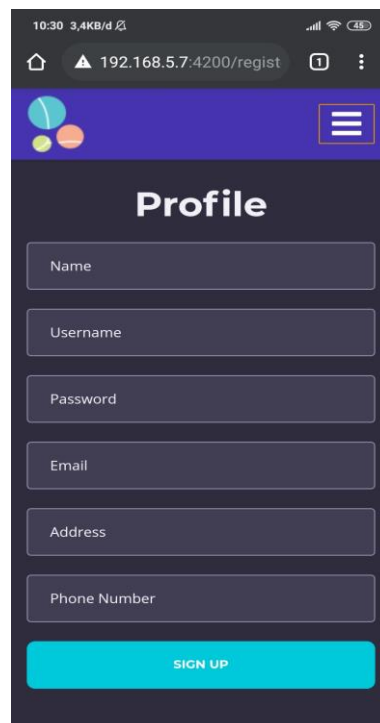
4.2.5 Antarmuka Halaman *Login* (Penumpang)



Gambar 4.11 Tampilan Antarmuka Halaman *Login* Penumpang

Antarmuka berikut merupakan halaman *login* untuk penumpang. Pada halaman ini penumpang harus memasukkan *username* dan *password* yang sesuai dengan data pada *database* aplikasi. Kemudian sistem akan memeriksa data tersebut sebelum penumpang bisa mengakses aplikasi. Jika penumpang belum memiliki *username* dan *password*, maka penumpang bisa mendaftarkan diri pada halaman registrasi yang bisa diakses oleh penumpang melalui menu *burger* di bagian atas aplikasi (kotak kuning). Pada gambar 4.11 merupakan tampilan antarmuka halaman *login* pada pihak penumpang.

4.2.6 Antarmuka Halaman Registrasi (Penumpang)



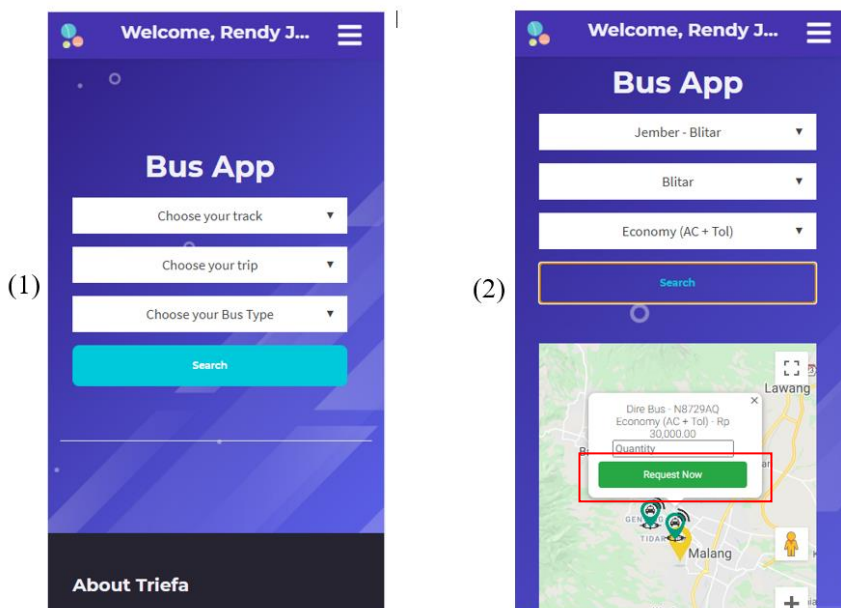
Gambar 4.12 Tampilan Antarmuka Halaman Registrasi

Halaman ini bisa diakses oleh penumpang untuk mendaftarkan diri mereka yang belum memiliki akses ke aplikasi (tidak memiliki *username* dan *password*). Pada halaman ini penumpang harus memasukkan nama lengkap, *username*, *password*, *email*, alamat rumah, dan nomor telepon sebagai syarat untuk mendapat akses pada aplikasi. Data tersebut kemudian akan masuk ke dalam *database* sistem dan bisa diakses oleh pihak admin. Setelah itu, penumpang bisa memasukkan *username* dan *password* yang telah dibuat sebelumnya untuk

melakukan *login* pada aplikasi (sudah memiliki akses untuk masuk ke dalam aplikasi). Pada gambar 4.12 merupakan tampilan antarmuka halaman registrasi.

4.2.7 Antarmuka Halaman Cari *Bus* (Penumpang)

Halaman ini akan muncul setelah proses *login* penumpang berhasil. Penumpang akan diarahkan pada halaman dimana penumpang diminta memasukkan *track*, pilihan kota penumpang turun, dan tipe *bus* untuk memunculkan *marker bus* yang sesuai dengan pilihan mereka(1). *User* dapat menekan *marker* pada salah satu *bus* untuk menampilkan *info window* dan melihat informasi seperti nomor plat, tipe *bus*, dan harga tiket *bus* itu sendiri. Jika sudah memilih bus yang ingin ditumpangi, pengguna dapat mengisi kolom *quantity* untuk menentukan jumlah tempat duduk yang diinginkan(2). *User* juga dapat melakukan *request* pada *info window* tersebut dengan menekan tombol *request now*. Pada gambar 4.13 merupakan tampilan antarmuka halaman cari *bus*.



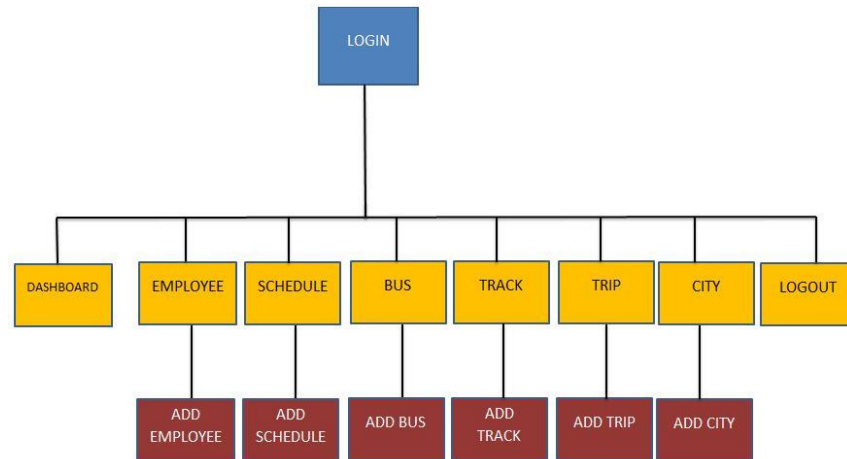
Gambar 4.13 Tampilan Antarmuka Halaman Cari *Bus*

4.3 Sitemap Aplikasi

Terdapat beberapa aplikasi yang terdiri dari aplikasi pegawai (supir dan kondektur) dan aplikasi *user* untuk bagian *front-end* dan admin di bagian *back-end*. Setiap aplikasi memiliki keunikan dan fitur yang berbeda dari satu aplikasi

ke aplikasi yang lainnya. Maka dari itu alur jalan setiap aplikasi juga berbeda. Berikut *sitemap* untuk menunjukkan alur jalan setiap aplikasi.

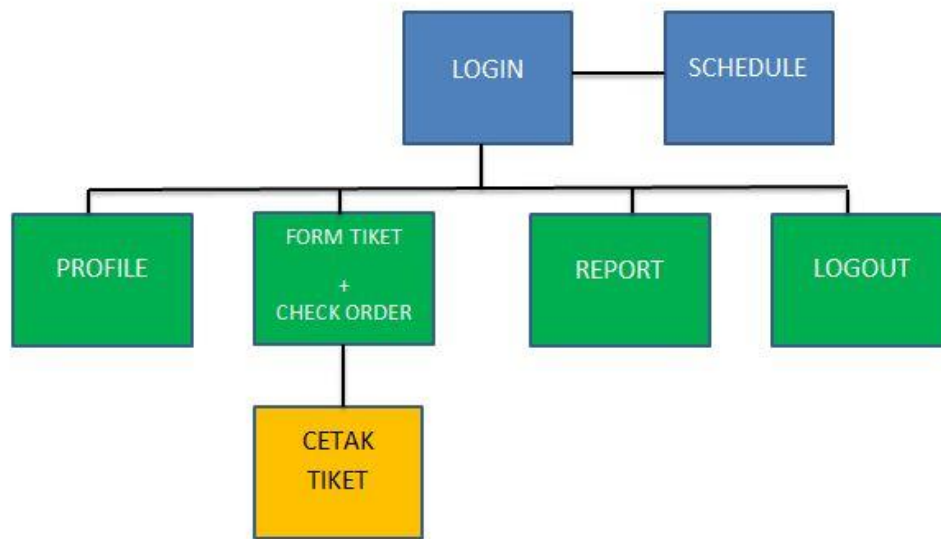
4.3.1 Sitemap Aplikasi Pada Sisi Admin



Gambar 4.14 Sitemap Aplikasi Pada Sisi Admin

Pada *sitemap* yang ditunjukkan pada gambar 4.14 tampilan awal aplikasi pada sisi admin adalah halaman *login*. Setelah memasukkan *username* dan *password* yang benar, pengguna akan berpindah ke tampilan halaman utama dan pengguna bisa memilih halaman untuk dikunjungi. Awalnya pengguna akan berada pada halaman *dashboard*, kemudian dari halaman ini pengguna bisa berpindah ke halaman lainnya melalui *navigation bar* yang tersedia. Terdapat beberapa halaman yang bisa dikunjungi oleh pengguna, yakni halaman *employee*, *schedule*, *bus*, *track*, *trip*, *city*, dan *logout*. Jika memilih halaman *employee*, *schedule*, *bus*, *track*, *trip*, *city* untuk dikunjungi, pengguna bisa memilih untuk mengubah (*create*, *read*, *update*, *delete*) data sesuai dengan halaman yang dikunjungi. Jika telah selesai pengguna bisa keluar dari aplikasi dengan tombol *logout* yang tersedia.

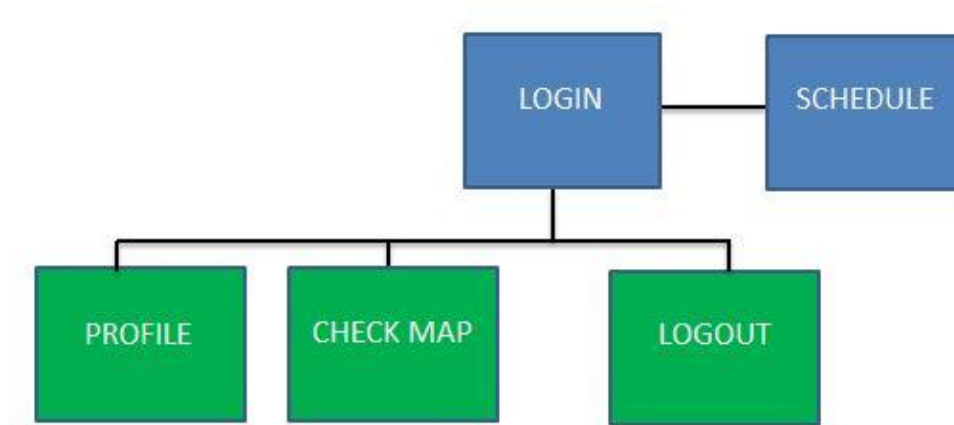
4.3.2 Sitemap Aplikasi Pada Sisi Kondektur



Gambar 4.15 Sitemap Aplikasi Pada Sisi Kondektur

Pada *sitemap* yang ditunjukkan pada gambar 4.15 tampilan awal aplikasi pada sisi kondektur adalah halaman *login*, terdapat halaman *schedule* yang berguna untuk menampilkan jadwal bertugas pegawai yang sedang aktif sekarang. Setelah memasukkan *username* dan *password* yang benar, pengguna akan berpindah ke tampilan halaman utama dan pengguna bisa memilih halaman untuk dikunjungi. Pada halaman utama, pengguna dapat langsung memasukkan data tiket penumpang pada *form* yang sudah tersedia. Kemudian, setelah memasukkan data-data tersebut, pengguna diarahkan ke halaman yang menampilkan tiket sebelum bisa dicetak. Selain halaman ini, pengguna dapat mengunjungi halaman *profile* jika ingin mengubah data-data pengguna. Jika telah selesai pengguna bisa keluar dari aplikasi dengan tombol *logout* yang tersedia.

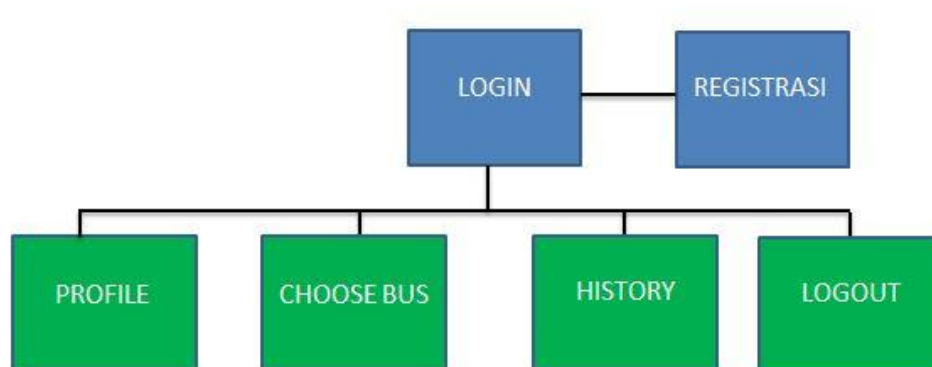
4.3.3 Sitemap Aplikasi Pada Sisi Supir



Gambar 4.16 Sitemap Aplikasi Pada Sisi Supir

Pada *sitemap* yang ditunjukkan pada gambar 4.16 tampilan awal aplikasi pada sisi kondektur adalah halaman *login*, terdapat halaman *schedule* yang berguna untuk menampilkan jadwal bertugas pegawai yang sedang aktif sekarang. Pada halaman utama, pengguna dapat memasukkan *marker* (sesuai dengan posisi pengguna sekarang) pada *map* yang tersedia dengan memasukkan jadwal tugas pengguna pada menu yang ada. Selain halaman ini, pengguna dapat mengunjungi halaman *profile* jika ingin mengubah data-data pengguna. Jika telah selesai pengguna bisa keluar dari aplikasi dengan tombol *logout* yang tersedia.

4.3.4 Sitemap Aplikasi Pada Sisi Penumpang

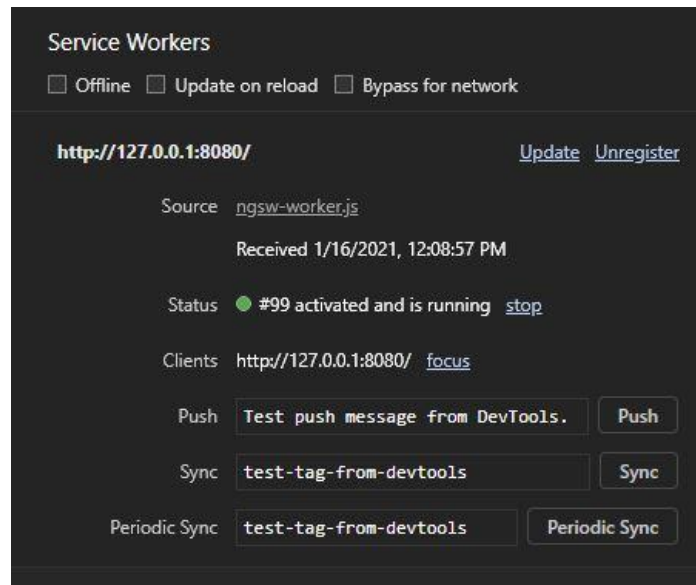


Gambar 4.17 Sitemap Aplikasi Pada Sisi Penumpang

Pada *sitemap* yang ditunjukkan pada gambar 4.17 tampilan awal aplikasi pada sisi penumpang adalah halaman *login*, terdapat halaman registrasi yang berguna untuk pengguna yang belum mendaftarkan diri sebagai penumpang pada aplikasi. Pada halaman utama, penumpang dapat memilih bus yang sesuai dengan keinginan dengan memilih jurusan bus pada menu yang tersedia. Kemudian, *marker* bus yang sesuai dengan jurusan pilihan pengguna akan tampil pada *map* yang tersedia. Selain halaman ini, pengguna dapat mengunjungi halaman *profile* jika ingin mengubah data-data pengguna. Jika telah selesai pengguna bisa keluar dari aplikasi dengan tombol *logout* yang tersedia.

4.4 Penerapan Sistem PWA (*Progressive Web App*) Dalam Aplikasi

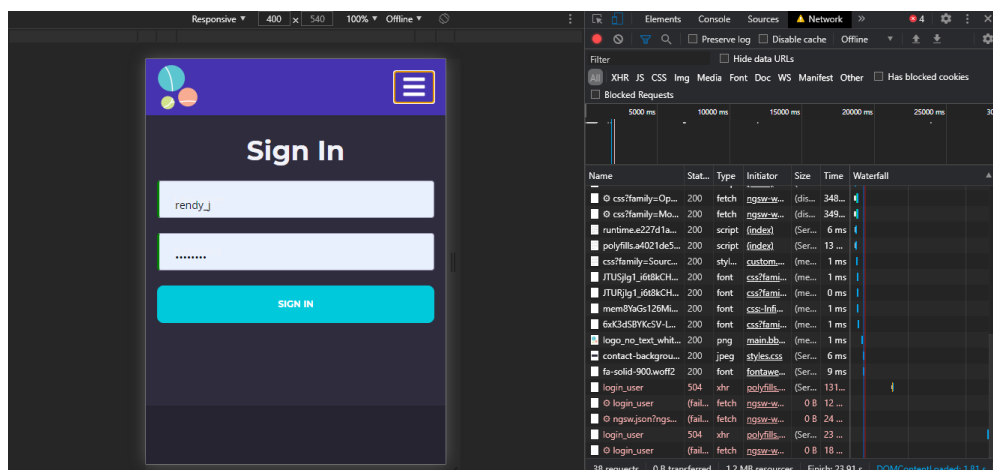
Dalam pembuatan aplikasi, telah diterapkan juga fitur PWA dalam aplikasi sehingga pengguna tetap bisa tersambung ke dalam aplikasi walaupun dalam kondisi internet yang tidak stabil ataupun saat tidak ada koneksi internet. Walaupun pengguna tidak dapat menggunakan seluruh fitur aplikasi, pengguna tidak akan terputus dari aplikasi dan tetap berada pada halaman yang diakses penumpang sebelum koneksi terputus. Pada subbab ini akan dibahas tampilan antarmuka aplikasi saat tidak ada koneksi internet (*offline*). Bagian aplikasi yang mendapat fitur PWA hanya pada pihak penumpang dan pegawai (konduktor dan supir). Semua hasil antarmuka aplikasi pada pihak penumpang dan pegawai diambil melalui *laptop* untuk menunjukkan *cache* dan *service worker* yang tersimpan pada *browser*. Pada gambar 4.18 berikut ini merupakan *service worker* yang sudah diterapkan pada aplikasi.



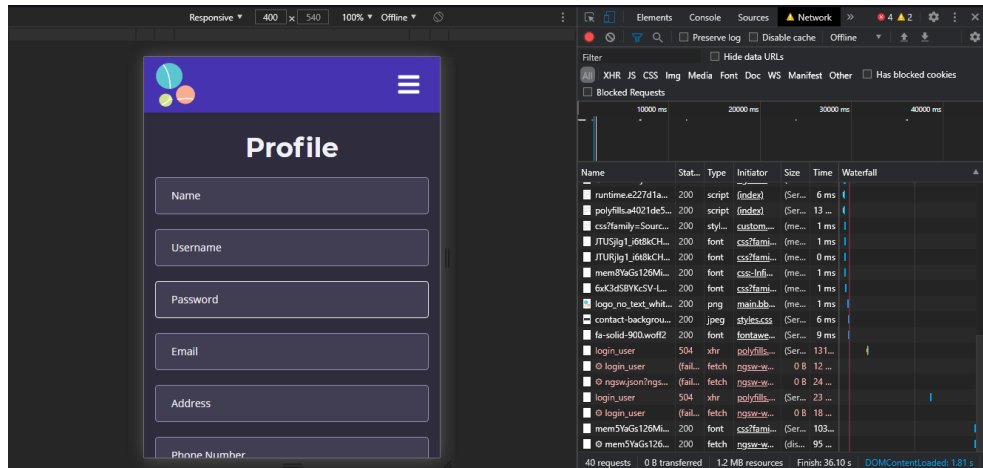
Gambar 4.18 Service Worker Pada Aplikasi

4.4.1 Antarmuka Aplikasi Pada Bagian Penumpang Dengan PWA

Pada aplikasi untuk pihak penumpang, masih akan muncul antarmuka untuk melakukan *login* atau daftar diri. Namun, penumpang tidak bisa melakukan proses *login* ataupun mendaftarkan diri. Penumpang hanya bisa mengisi *input box* yang tersedia saja. Pada gambar 4.19 merupakan antarmuka halaman *login* juga *cache* yang tersimpan pada *browser*. Kemudian, gambar 4.20 merupakan antarmuka halaman registrasi beserta *cache*.

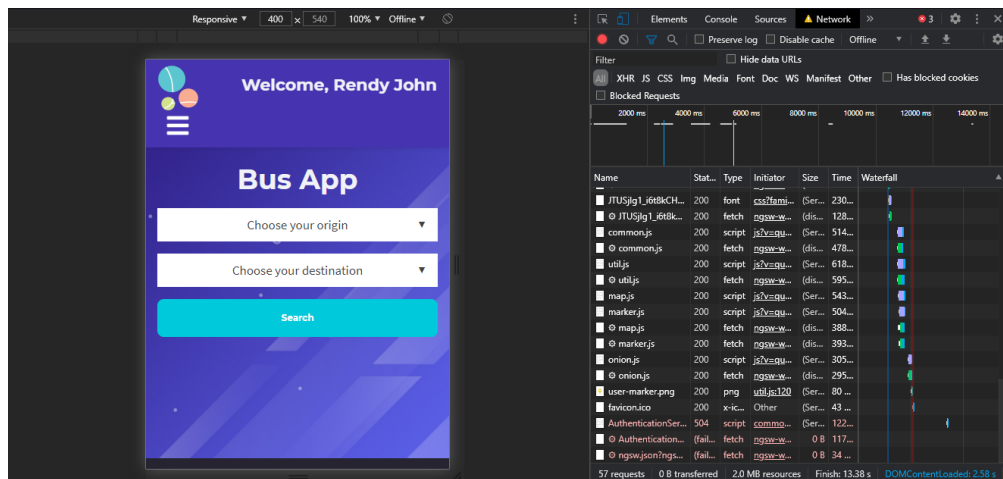


Gambar 4.19 Antarmuka Halaman *Login* Penumpang Dengan PWA

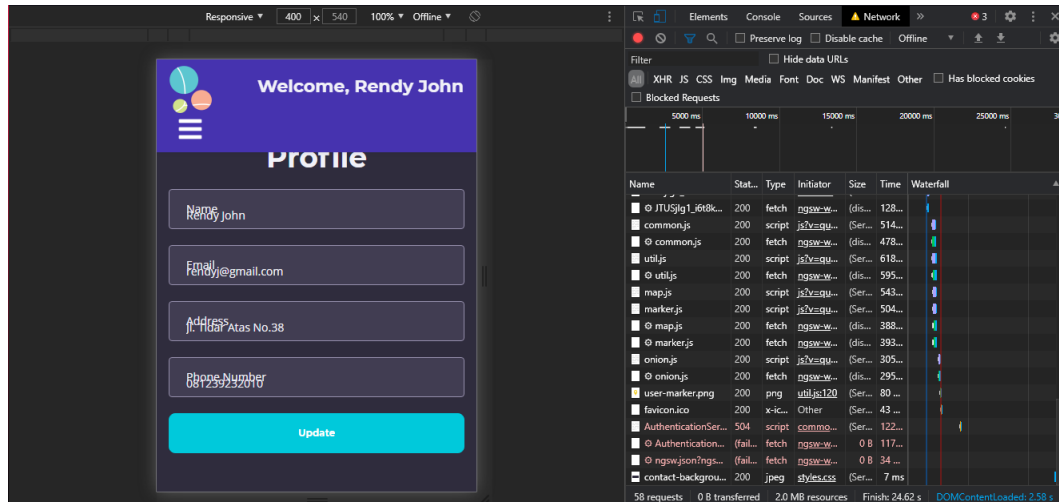


Gambar 4.20 Antarmuka Halaman *Register* Penumpang Dengan PWA

Pada halaman utama aplikasi, penumpang masih bisa mengakses tampilan aplikasi. Penumpang juga masih bisa mengakses menu *profile*. Namun, penumpang tidak bisa mengakses fungsi aplikasi seperti memilih jurusan bus yang diinginkan atau memperbaharui profil penumpang. Pada gambar 4.21 merupakan antarmuka halaman utama dan juga gambar 4.22 yang merupakan antarmuka halaman *profile*.



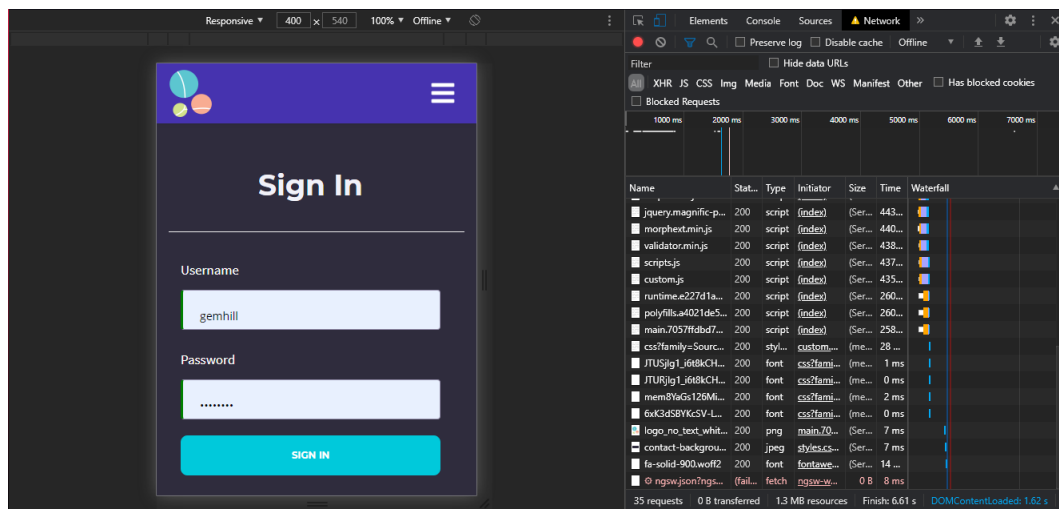
Gambar 4.21 Antarmuka Halaman Utama Penumpang Dengan PWA



Gambar 4.22 Antarmuka Halaman *Profile* Penumpang Dengan PWA

4.4.2 Antarmuka Aplikasi Pada Bagian Pegawai Dengan PWA

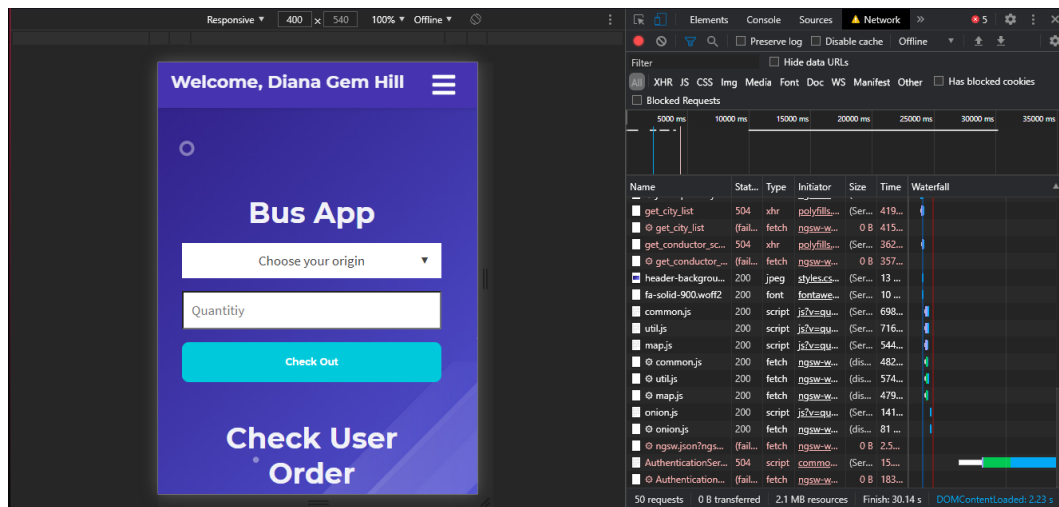
Pada aplikasi untuk pihak pegawai, masih akan muncul antarmuka untuk melakukan *login* atau melihat jadwal bertugas. Namun, pegawai tidak akan bisa melakukan proses *login* ataupun melihat jadwal. Gambar 4.23 merupakan antarmuka halaman *login* pegawai dengan PWA.



Gambar 4.23 Antarmuka Halaman *Login* Pegawai Dengan PWA

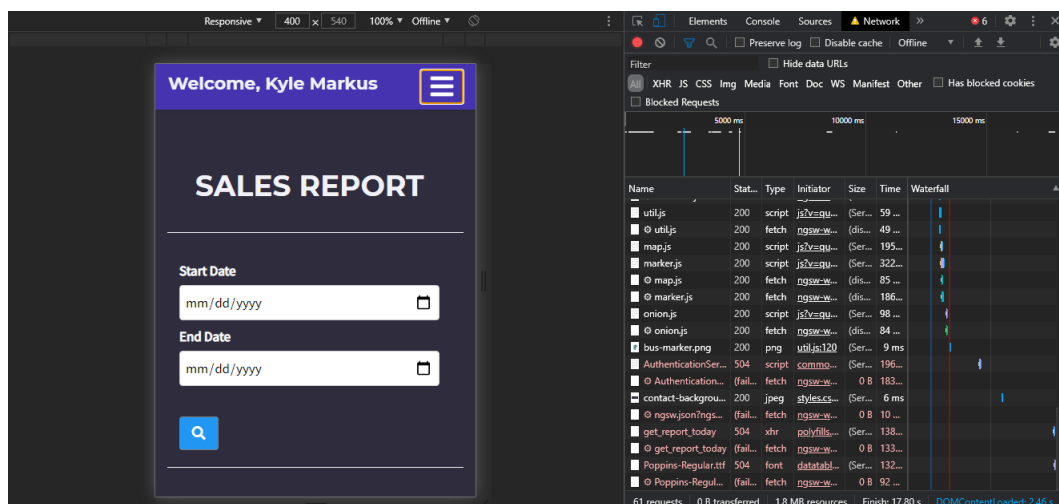
Halaman utama aplikasi pun masih bisa diakses oleh pegawai begitu juga dengan halaman *profile*. Selain itu, halaman *report* masih bisa diakses juga. Namun, untuk fungsi seperti menampilkan *marker bus* pada *map* (untuk supir) dan memasukkan data tiket penumpang (untuk kondektur) tidak bisa dilakukan.

Gambar 4.24 merupakan antarmuka halaman utama untuk kondektur dengan PWA.



Gambar 4.24 Antarmuka Halaman Utama Kondektur Dengan PWA

Halaman *report* pun masih dapat diakses oleh pegawai. Namun seperti halaman lain, pegawai hanya bisa melihat konten halaman yang ada sebelum koneksi internet terputus. Gambar 4.25 merupakan antarmuka halaman *report* dengan PWA.



Gambar 4.25 Antarmuka Halaman *Report* Pegawai Dengan PWA

4.5 Hasil Uji Coba Penggunaan Aplikasi

Pengujian aplikasi dilakukan melalui 2 tahap, yaitu pengujian *white box* dan survei. Pengujian *black box* dilakukan untuk menguji fitur-fitur pada aplikasi berjalan sesuai dengan yang diinginkan atau tidak. Untuk pengujian menggunakan

survei, dilakukan sebagai penilaian terhadap aplikasi yang telah dibuat. Penilaian tersebut meliputi tampilan dan juga jalannya aplikasi.

4.5.1 Hasil Pengujian *Black Box*

Berikut ini merupakan hasil pengujian *black box* yang telah dilakukan pada saat aplikasi dikembangkan. Hasil pengujian *black box* akan dijabarkan ke dalam tabel. Pada tabel tersebut terdapat fitur-fitur yang terdapat pada aplikasi, baik dari sisi admin maupun pegawai dan penumpang. Tabel 4.1 merupakan tabel hasil pengujian *black box* aplikasi.

Tabel 4.1 Tabel Hasil Pengujian *Black Box* Aplikasi

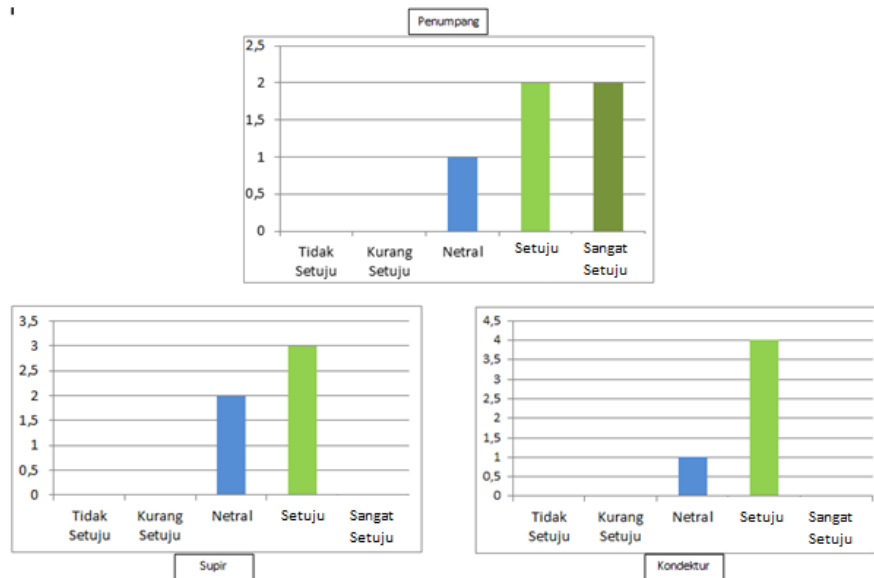
No.	Fitur	Berhasil	Gagal
1	Login & Logout	✓	
2	Registrasi	✓	
3	Cari Bus	✓	
4	Request Bus	✓	
5	Cetak Tiket	✓	
6	Confirm Request	✓	
7	Lihat Jadwal	✓	
8	Rekapitulasi Data	✓	
9	Cari Lokasi Penumpang	✓	
10	CRUD Data	✓	

4.5.2 Hasil Survei

Setelah aplikasi telah selesai dirancang dan dibuat, maka dilakukan uji coba penggunaan aplikasi terhadap beberapa responden, dalam hal ini responden tersebut adalah pihak masyarakat umum (selaku pengguna jasa transportasi *bus*) dan pihak pegawai *bus*, yakni supir dan kondektur (selaku yang mengoperasikan *bus*). Uji coba ini dilakukan dalam 2 tahap, yakni uji coba penggunaan aplikasi dan pengisian angket sebagai sarana untuk memberi penilaian terhadap aplikasi, baik dari segi tampilan maupun fungsionalitas.

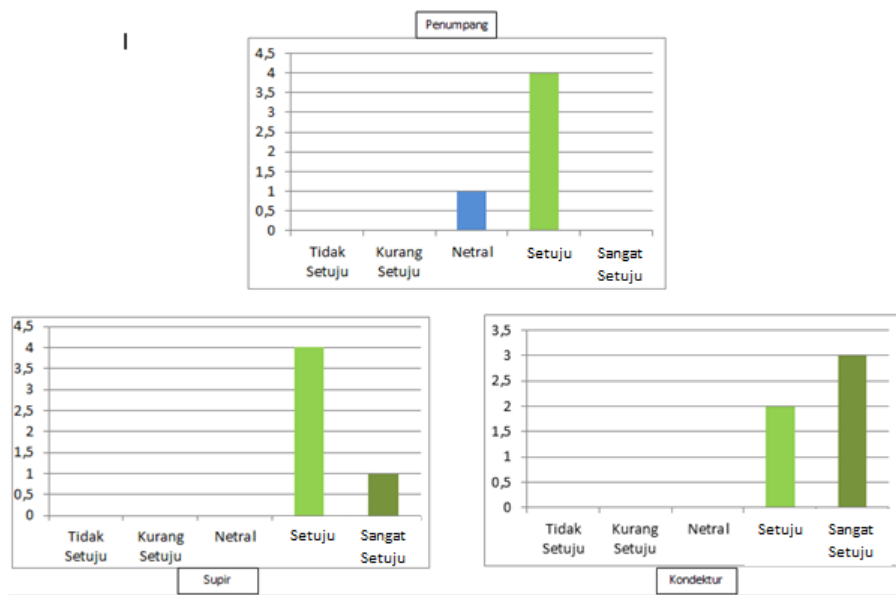
Pengguna aplikasi (masyarakat dan pegawai *bus*) harus mencoba menggunakan aplikasi terlebih dahulu, barulah kemudian angket akan diberikan kepada mereka. angket tersebut berisi pertanyaan-pertanyaan yang berkaitan dengan aplikasi, baik tampilan maupun pengoperasiannya. Masing-masing angket (untuk masyarakat dan pegawai *bus*) memiliki jumlah pertanyaan yang sama,

yakni 7 pertanyaan. Dari pertanyaan-pertanyaan tersebut, responden diberikan beberapa pilihan jawaban (Tidak Setuju / Kurang Setuju / Netral / Setuju / Sangat Setuju) untuk menjawabnya. Berikut ini merupakan grafik-grafik yang menunjukkan setiap pertanyaan beserta pilihan jawaban dari responden. Pertanyaan satu “Tampilan aplikasi menarik ?”



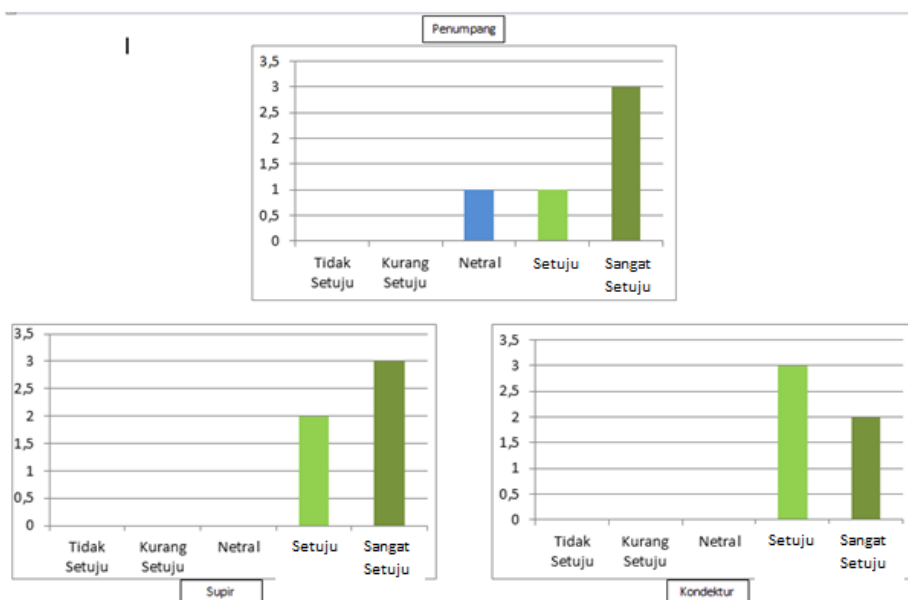
Gambar 4.26 Pilihan Jawaban Responden Terhadap Pertanyaan 1

Seperti ditunjukan pada gambar 4.26 yang diambil dari 15 orang responden (5 responden pengguna, 5 responden kondektur, dan 5 responden supir). Hasil penilaian melalui angket untuk pertanyaan ke 1 menunjukkan nilai yang cukup baik dalam tampilan aplikasi. Pertanyaan dua “Ukuran elemen-elemen (tombol, font, input box, dan lainnya) di aplikasi sesuai ?”



Gambar 4.27 Pilihan Jawaban Responden Terhadap Pertanyaan 2

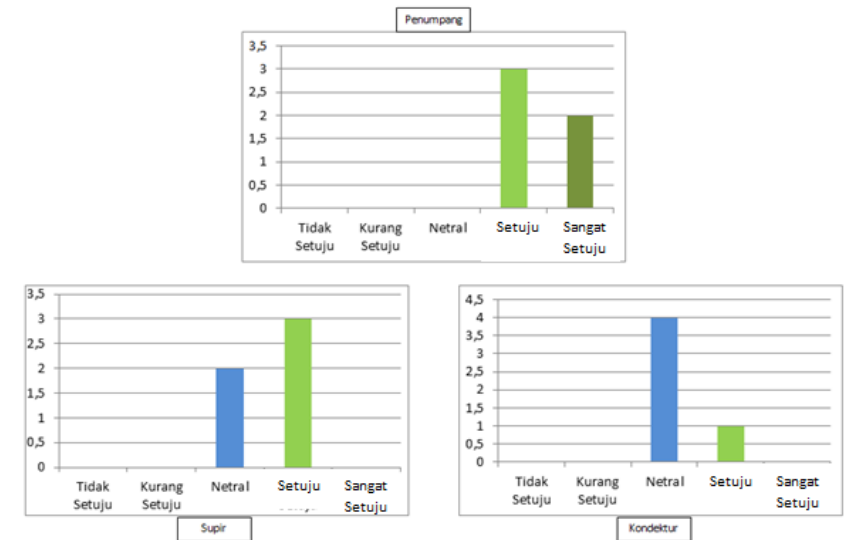
Seperti ditunjukkan pada gambar 4.27 yang diambil dari 15 orang responden (5 responden pengguna, 5 responden kondektur, dan 5 responden supir). Hasil penilaian melalui angket untuk pertanyaan ke 2 menunjukkan nilai yang cukup baik. Pertanyaan tiga “Pemilihan warna pada aplikasi sesuai ?”



Gambar 4.28 Pilihan Jawaban Responden Terhadap Pertanyaan 3

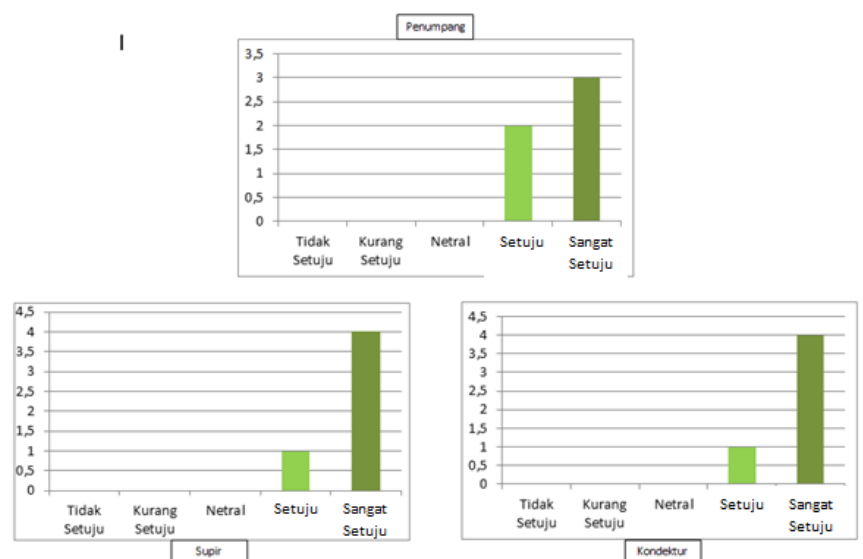
Seperti ditunjukkan pada gambar 4.28 yang diambil dari 15 orang responden (5 responden pengguna, 5 responden kondektur, dan 5 responden

supir). Hasil penilaian melalui angket untuk pertanyaan ke 3 menunjukkan nilai yang cukup baik. Pertanyaan empat “Aplikasi mudah untuk digunakan ? (tidak membingungkan dalam pengoperasian)”



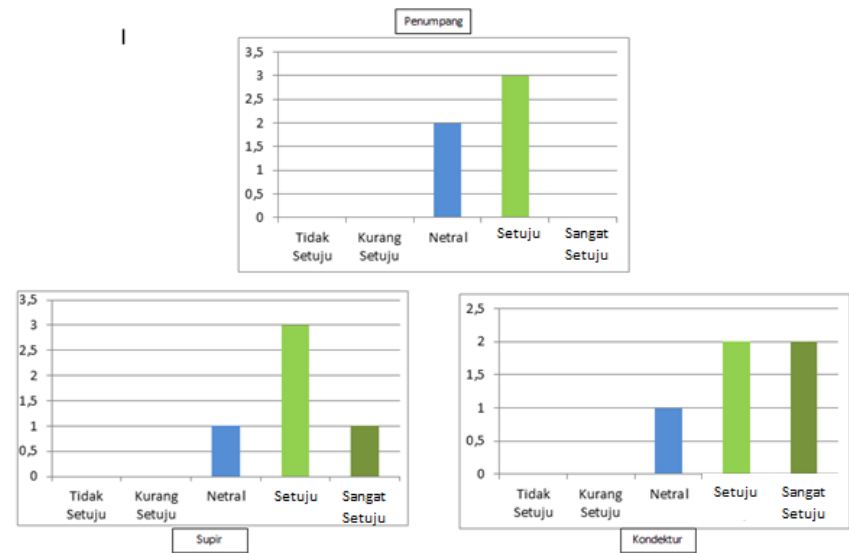
Gambar 4.29 Pilihan Jawaban Responden Terhadap Pertanyaan 4

Pada gambar 4.29 yang diambil dari 15 orang responden (5 responden pengguna, 5 responden kondektur, dan 5 responden supir). Hasil penilaian melalui angket untuk pertanyaan ke 4 menunjukkan nilai yang cukup baik. Pertanyaan lima “Fungsi elemen-elemen (tombol, input box, dan lainnya) bekerja dengan baik ?”



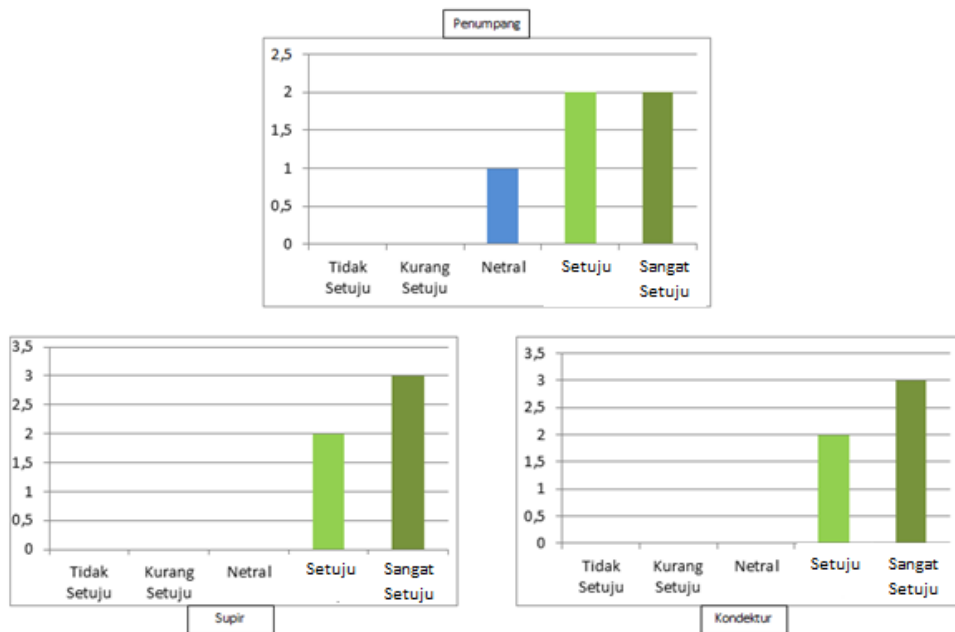
Gambar 4.30 Pilihan Jawaban Responden Terhadap Pertanyaan 5

Seperti ditunjukkan pada gambar 4.30 yang diambil dari 15 orang responden (5 responden pengguna, 5 responden kondektur, dan 5 responden supir). Hasil penilaian melalui angket untuk pertanyaan ke 5 menunjukkan nilai yang cukup baik. Pertanyaan enam “Aplikasi bermanfaat bagi pengguna ?”



Gambar 4.31 Pilihan Jawaban Responden Terhadap Pertanyaan 6

Seperti ditunjukkan pada gambar 4.31 yang diambil dari 15 orang responden (5 responden pengguna, 5 responden kondektur, dan 5 responden supir). Hasil penilaian melalui angket untuk pertanyaan ke 6 menunjukkan nilai yang cukup baik. Pertanyaan tujuh “Secara keseluruhan, apakah aplikasi membuat pekerjaan yang dilakukan menjadi lebih efisien ?”



Gambar 4.32 Pilihan Jawaban Responden Terhadap Pertanyaan 7

Pada gambar 4.32 yang diambil dari 15 orang responden (5 responden pengguna, 5 responden kondektur, dan 5 responden supir). Hasil penilaian melalui angket untuk pertanyaan ke 7 menunjukkan nilai yang cukup baik dalam hal kepuasan penggunaan aplikasi secara keseluruhan.

Keseluruhan hasil angket yang telah dijabarkan dalam bentuk grafik menunjukkan bahwa pengguna merasa aplikasi sudah cukup baik dari segi *User Interface* (UI) dan *User Experience* (UX). Aplikasi sudah dapat berjalan seperti yang diinginkan. Selain itu, pekerjaan pegawai pun terbantu dengan aplikasi ini. Pihak penumpang juga dibantu dalam hal *request* bus dan juga transaksi pembelian tiket.

Bab V

Penutup

5.1 Kesimpulan

Kesimpulan yang bisa didapat dari aplikasi yang telah dibuat dapat dilihat pada poin-poin berikut.

1. Aplikasi dapat membantu pekerjaan pegawai dan penumpang menjadi lebih efisien serta mempermudah dalam hal penjualan tiket sesuai dengan hasil ujicoba pada gambar 4.32 yang menunjukkan hasil 60% sangat setuju dan 40% setuju (supir dan kondektur) dan 40% sangat setuju dan 40% setuju, sisanya 20% menjawab netral pada penumpang.
2. Tampilan aplikasi sudah cukup menarik dan terstruktur sehingga mudah digunakan oleh penumpang dan pegawai. Hal ini sesuai dengan hasil ujicoba pada gambar 4.26 sampai gambar 4.29 yang menunjukkan hasil pada penumpang 35% sangat setuju, 50% setuju, dan 15% netral. Pada kondektur menunjukkan hasil 20% sangat setuju, 60% setuju, dan 20% netral. Pada supir menunjukkan hasil 50% sangat setuju, 25% setuju, dan 25% netral.
3. Proses pembelian tiket menjadi lebih terstruktur dan teratur sehingga data penjualan dapat diolah lebih mudah.

5.2 Saran

Pembuatan aplikasi ini masih memiliki beberapa kekurangan dan dapat dikembangkan lagi. Berikut beberapa saran yang dapat digunakan pengembangan selanjutnya.

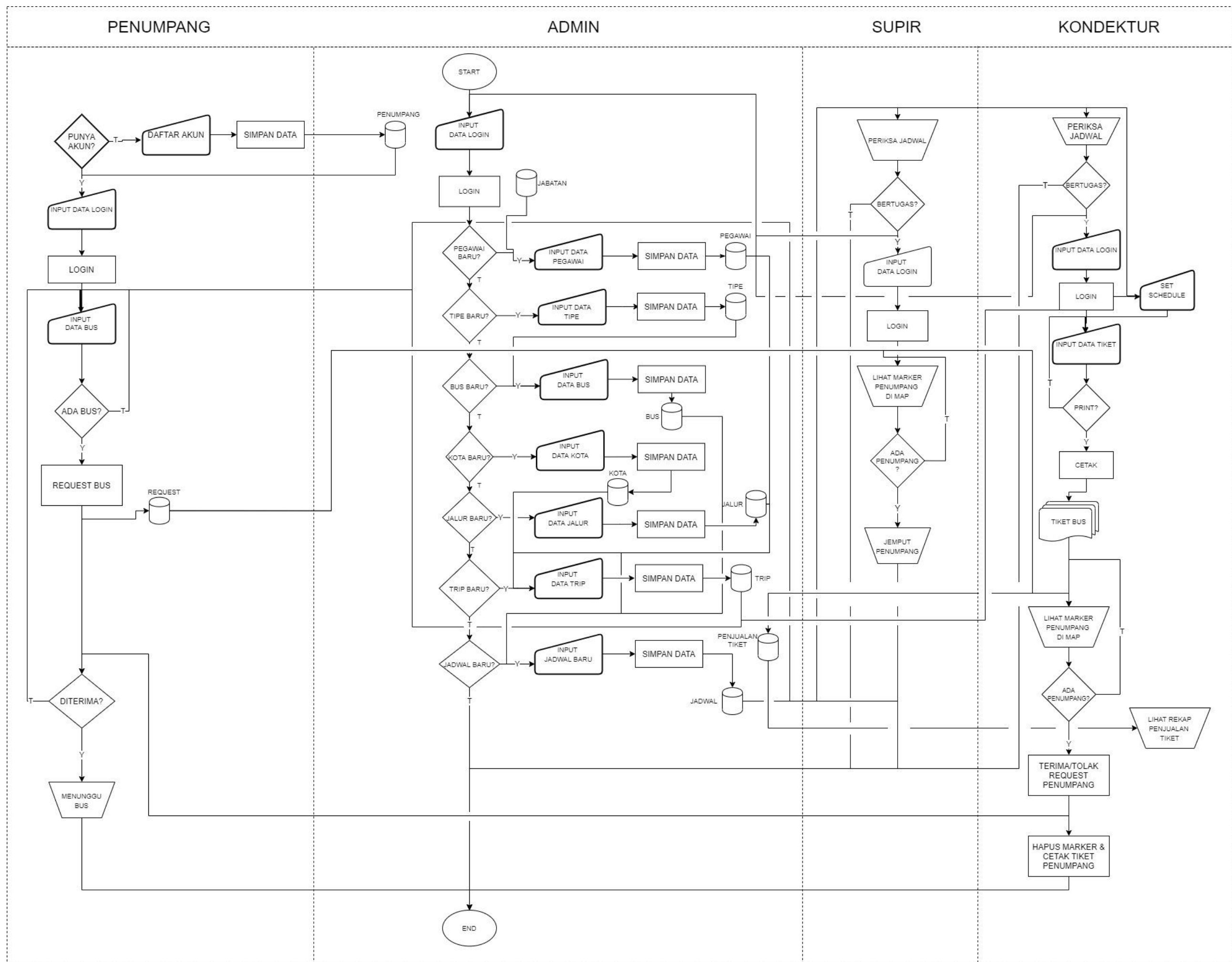
1. Penambahan fitur pembayaran *online* untuk lebih memudahkan penumpang dalam membayar tiket dengan menggunakan *payment gateway* seperti doku, iPaymu, atau Finpay.
2. Mengembangkan aplikasi sehingga dapat digunakan dengan tipe *bus* yang lain.
3. Menambahkan fitur notifikasi pada saat terjadi aksi seperti *order* penumpang atau kedatangan *bus* ke lokasi penumpang.

Daftar Pustaka

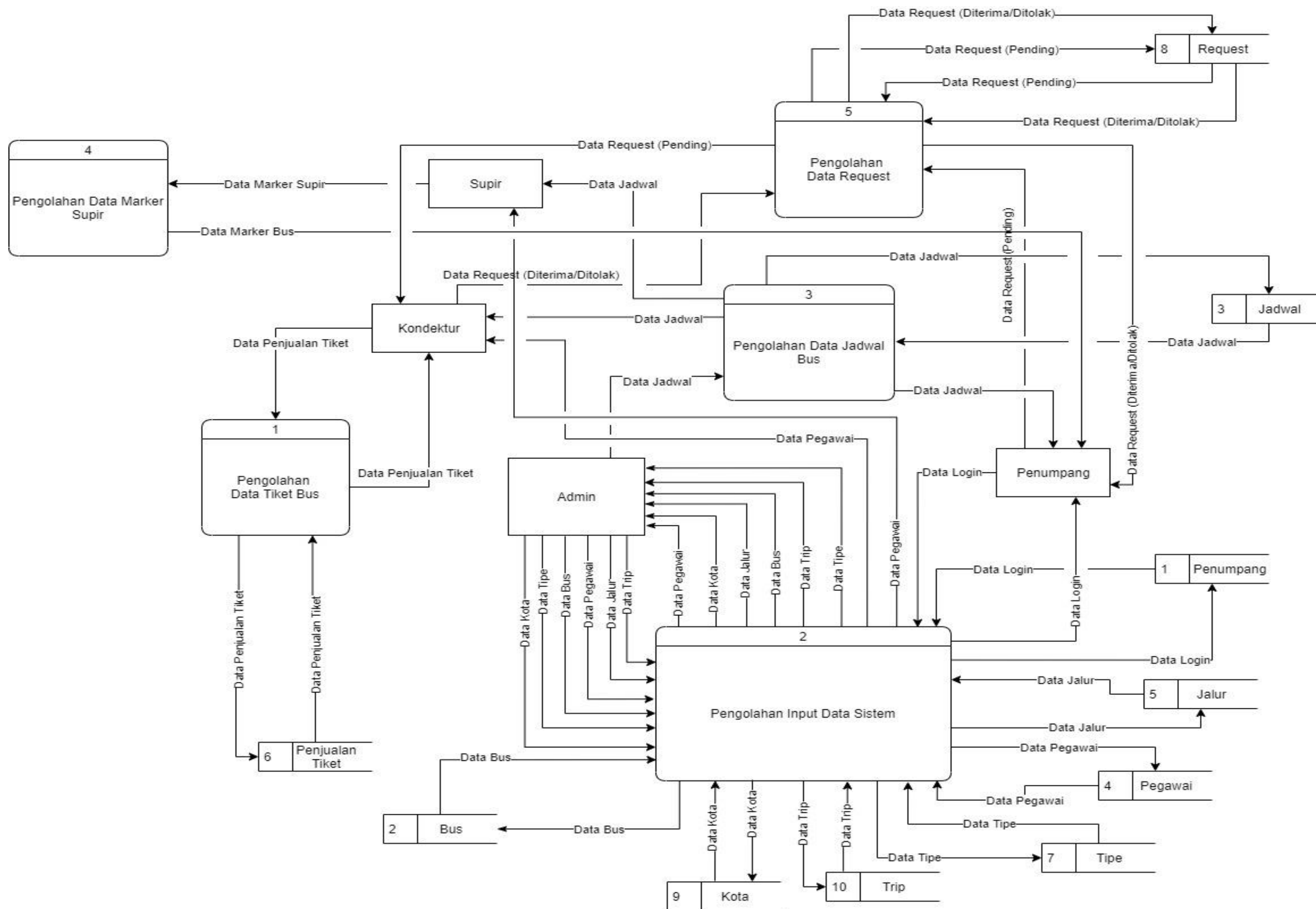
- Warpani, S. 2002. *Pengelolaan Lalu Lintas dan Angkutan Jalan*. Bandung: Penerbit ITB.
- Morlok. 1978. *Introduction To Transportation Engineering And Planning, US*: McGraw-Hill College.
- Miro, Fidel. 2012, *Pengantar Sistem Tranportasi Transportasi*, Jakarta: Erlangga.
- Miro, Fidel. 2004. *Perencanaan Transportasi Untuk Mahasiswa, Perencana, Dan Praktisi*. Jakarta: Erlangga.
- Bowersox, D.J., Calabro, P.J. and Wagenheim, G.D., 1981. *Introduction to transportation*. Macmillan College.
- Munawar, A., 2005, *Dasar-dasar Teknik Transportasi*, Beta Offset, Yogyakarta.
- Nugroho, B 2005, *Database Relasional dengan MySQL*, 1st penyunt, New York, Heinz Weinheimer.
- Peranginangin, Kasiman 2006, *Aplikasi WEB dengan PHP dan MySQL*, Andi, Yogyakarta.
- Suteja, R, Prijono, A & Agustaf, R 2006, *Mudah dan Cepat menguasai Pemrograman Web*, Informatika, Bandung.
- M. Adit Saputra. 2017. *Pengembangan Aplikasi Pemesanan Tiket Bus Pada Po. Puspa Jaya Berbasis Android*.
- E. Patrisia, W. Wikusna, P. Aji. 2019. *Aplikasi Penjualan Tiket Bus Secara Online Berbasis Web Di Cv Harum Prima Bandung. e-Proceeding of Applied Science : Vol.5, No.1*
- R. Rachmatullah. 2015. *Perancangan Sistem Pemesanan Tiket Bus Online Berbasis Web. Indonesian Journal on Networking and Security - Volume 4 No 1*
- A. R. A. Rahmat, A. Octaviano. 2016. *Aplikasi Pemesanan Tiket Bus Berbasis Web (Studi Kasus Pada Po. Harapan Jaya). JURNAL INFORMATIKA UNIVERSITAS PAMULANG Vol.1*
- P. Guntoro. 2007. *Sistem Informasi Pemesanan Tiket Bus Online (Studi Kasus P.O Safari Dharma Raya)*
- I. R. Pradana. 2015. *Perancangan Aplikasi Alat Pemesanan Tiket Bus Dengan Perangkat Telepon Genggam Berbasis J2me (Studi Kasus Pada Po. Bejeu)*

- E. Ngaga, F. Tedy. 2016. Rancang Bangun Sistem Pemesanan Tiket Bus Kupang-Atambua Berbasis SMS Gateway. Jurnal Edukasi dan Penelitian Informatika (JEPIN) Vol. 2, No. 2, 2016
- I. C. Suranta. 2012. Sistem Informasi Pemesanan Tiket Bus *Online* Pada PT Lorena Karina *Transport* Berbasis Web Dan Sms Gateway
- L. I. Idris. 2017. Sistem informasi Pemesanan Tiket Bus Berbasis Web Pada Po.Karunia Bakti.
- Sommerville, Ian. 2011. *Software Engineering*. Boston: Pearson.
- Dalma, M. Alif. 2020, Flowchart adalah – Pengertian, Jenis, Simbol, Contoh, diakses tanggal 13 Februari 2020, <https://dosenpintar.com/flowchart-adalah/>.

LAMPIRAN A

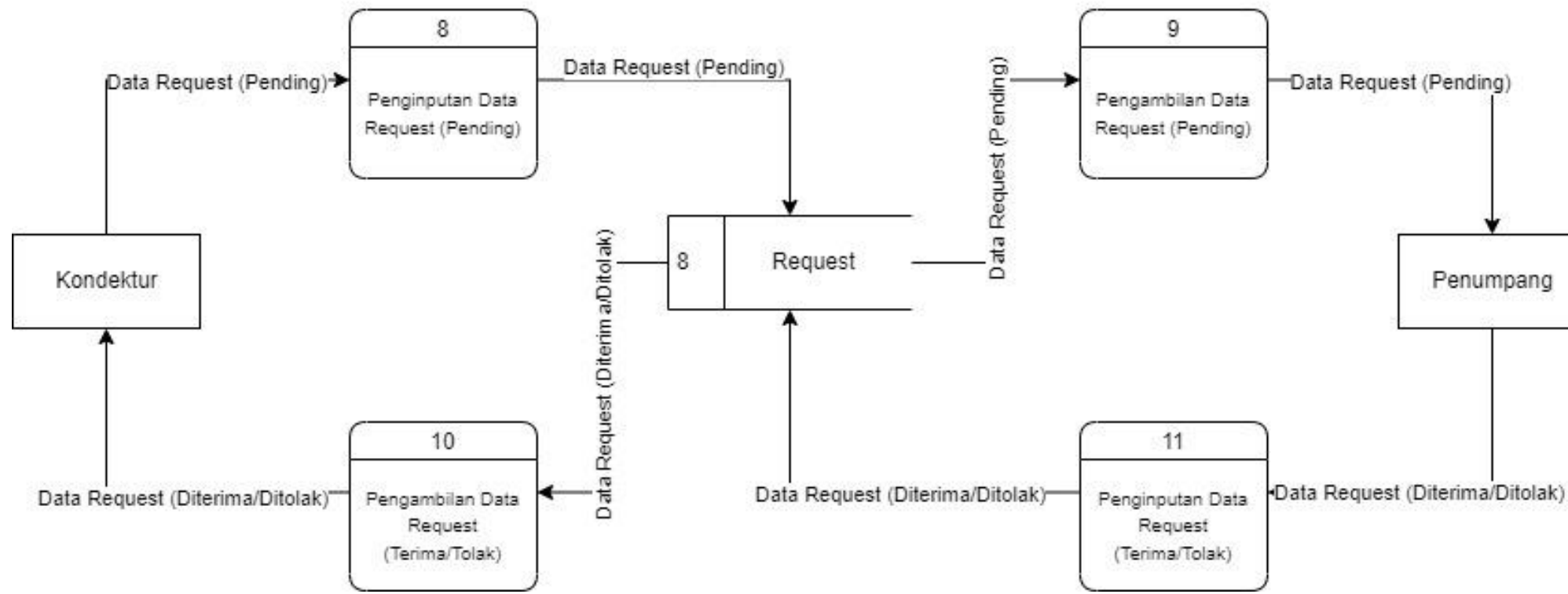


LAMPIRAN B

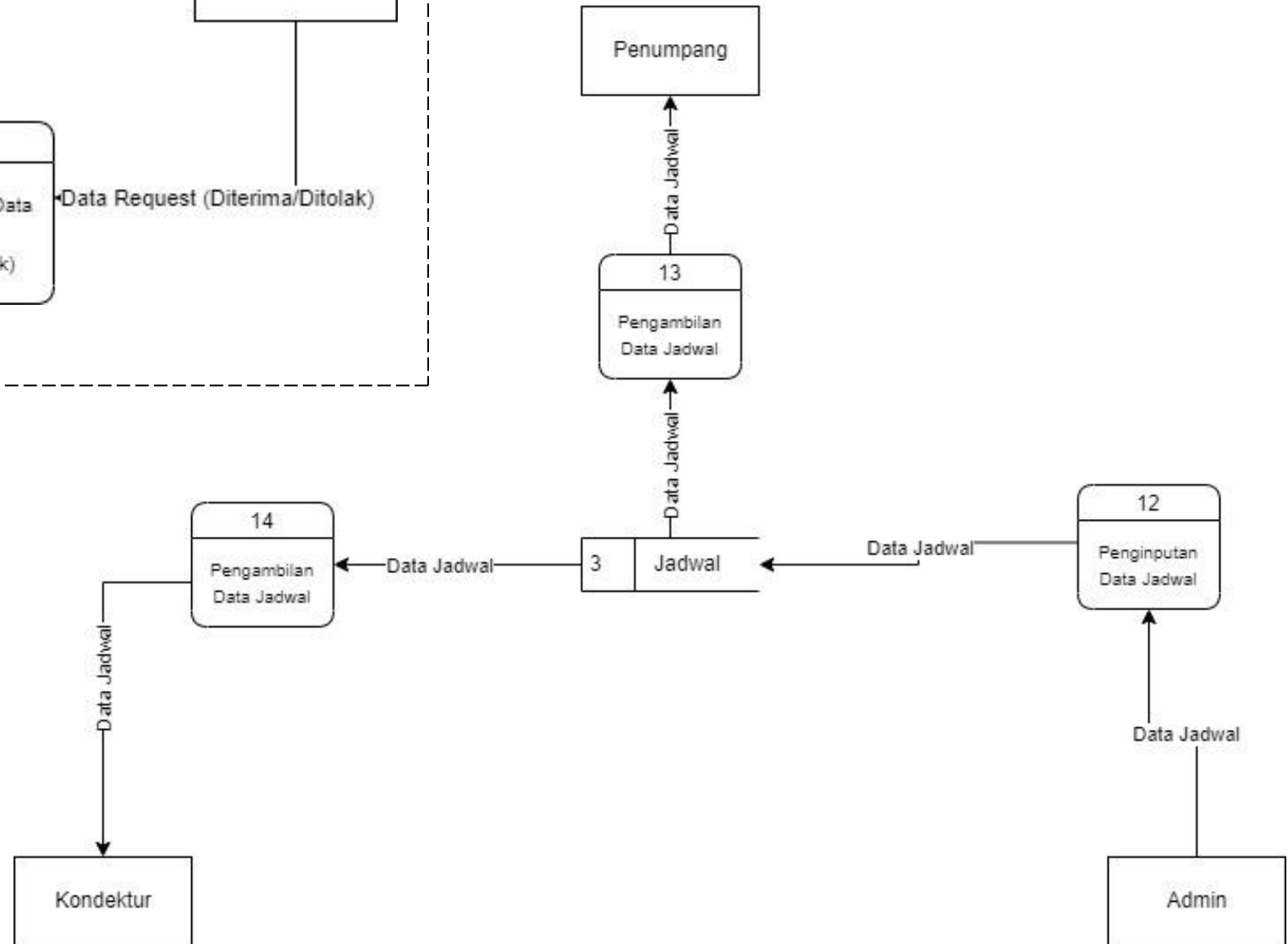


LAMPIRAN C

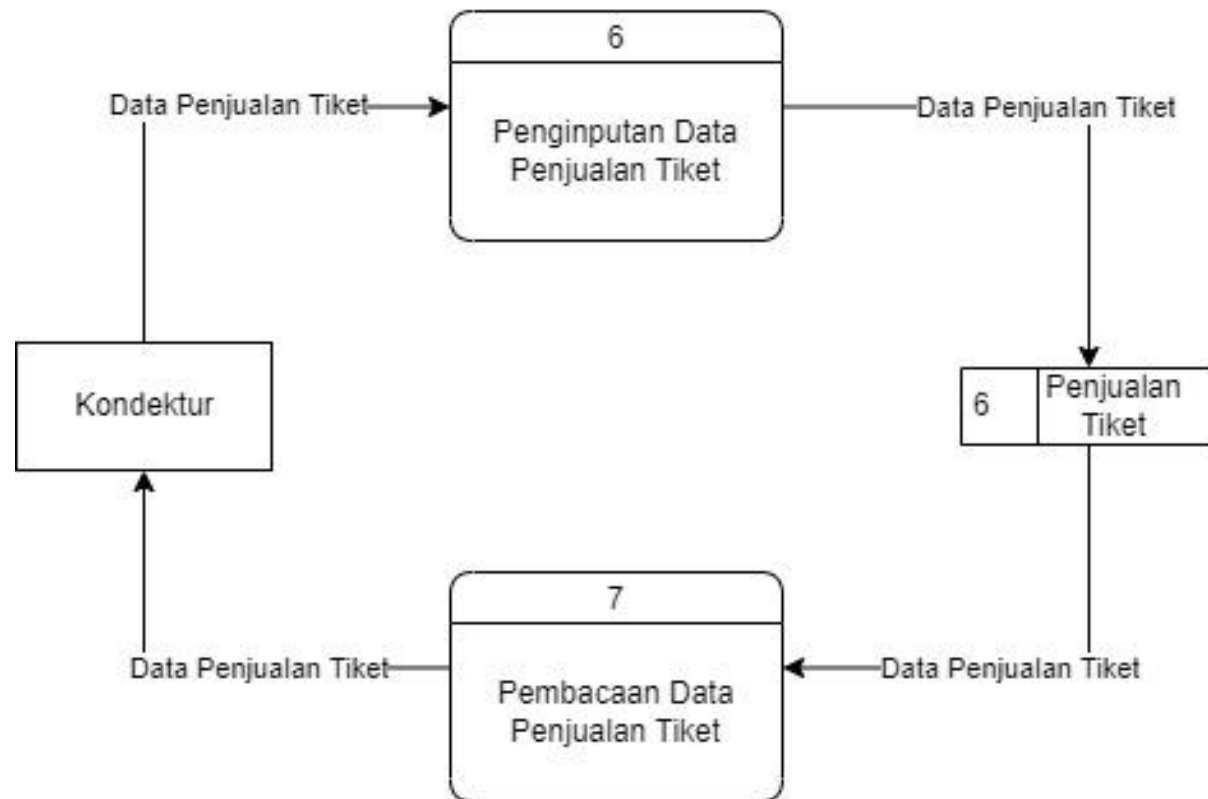
Pengolahan Data Request



Pengolahan Data Jadwal Bus



Pengolahan Data Tiket Bus



LAMPIRAN D

