



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KI091391

RANCANG BANGUN APLIKASI TRAVEL ONLINE BERBASIS ANDROID

PRASETYATI RISKI RAHMAWATI
NRP. 5109 100 016

Dosen Pembimbing I
Dwi Sunaryono, S.Kom, M.Kom

Dosen Pembimbing II
Sarwosri, S.Kom, M.T

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR - KI091391

RANCANG BANGUN APLIKASI TRAVEL ONLINE BERBASIS ANDROID

PRASETYATI RISKI RAHMAWATI
NRP. 5109 100 016

Dosen Pembimbing I
Dwi Sunaryono, S.Kom, M.Kom

Dosen Pembimbing II
Sarwosri, S.Kom, M.T

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]



UNDERGRADUATE THESIS - KI091391

APPLICATION DEVELOPMENT OF TRAVEL ONLINE ANDROID BASE

**PRASETYATI RISKI RAHMAWATI
NRP. 5109 100 016**

**Supervisor I
Dwi Sunaryono, S.Kom, M.Kom**

**Supervisor II
Sarwosri, S.Kom, M.T**

**DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Sepuluh Nopember Institute of Technology
Surabaya 2017**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

RANCANG BANGUN APLIKASI TRAVEL ONLINE BERBASIS ANDROID

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Manajemen Informasi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

PRASETYATI RISKI RAHMAWATI
NRP. 5109 100 016

Disetujui oleh Dosen Pembimbing Tugas Akhir:

1. Dwi Sunaryono, S.Kom., M.Kom.
NIP. 197205281997021001 (Pembimbing I)
2. Sarwosri, S.Kom., M.T.
NIP. 197608092001122001 (Pembimbing II)



SURABAYA
Januari, 2017

[Halaman ini sengaja dikosongkan]

RANCANG BANGUN APLIKASI TRAVEL ONLINE BERBASIS ANDROID

Nama Mahasiswa : PRASETYATI RISKI RAHMAWATI
NRP : 5109 100 016
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing I : Dwi Sunaryono, S.Kom., M.Kom
Dosen Pembimbing II : Sarwosri, S.Kom., M.T

Abstrak

Alat transportasi sudah menjadi kebutuhan yang tidak bisa ditinggalkan dalam kehidupan sehari-hari. Kebutuhan manusia yang terus meningkat dan keperluan untuk bepergian sudah menjadi bagian kegiatan setiap harinya. Hampir semua orang mempunyai kendaraan pribadi tapi tidak sedikit yang lebih memilih menggunakan kendaraan umum. Tetapi untuk mengakses kendaraan umum lebih merepotkan karena harus melalui terminal, stasiun, atau bandara.

Fasilitas kendaraan umum sekarang ini memang belum memadai. Dari segi kebersihan dan ketepatan waktu banyak yang mengeluhkan. Tetapi belakangan banyak muncul alternatif kendaraan yang disukai banyak orang, yaitu transportasi online. Mulai menjamur ojek-ojek online dan taksi online. Dengan Kemudahan dalam pemesanan yang hanya menggunakan aplikasi pada smartphone, kita tinggal menunggu pengemudi menjemput kita dan mengantarkan sampai tujuan. Hal ini tentu disambut hangat oleh masyarakat karena selain menghemat tenaga tanpa perlu menuju pangkalan ojek atau taksi, atau harus menuju jalan raya untuk mencari kendaraan, kita juga bisa menghemat waktu.

Pada tugas akhir ini transportasi yang digunakan adalah travel yang menjangkau wilayah yang lebih luas. Jika transportasi online yang marak sekarang hanya menjangkau dalam kota, tugas akhir ini memberikan pilihan dengan jangkauan antar kota dengan menggunakan smartphone Android

yang mayoritas masyarakat Indonesia gunakan saat ini. Kemudahan memasukkan alamat penjemputan dan alamat tujuan yang menggunakan Google Maps API yang memudahkan dalam penjemputan oleh armada. Harga yang digunakan juga berdasarkan jarak tempuh antara lokasi penjemputan dan lokasi tujuan sehingga tidak ada kecurangan oleh oknum pengemudi.

Kata Kunci : Android, Google Map API, Transportasi Online, Travel

APPLICATION DEVELOPMENT OF TRAVEL ONLINE ANDROID BASE

Student's Name : PRASETYATI RISKI RAHMAWATI
NRP : 5109 100 016
Major of Department : Teknik Informatika FTIF-ITS
Supervisor I : Dwi Sunaryono, S.Kom., M.Kom
Supervisor II : Sarwosri, S.Kom., M.T

Abstract

Transportation has become a necessity that can not be left out in everyday life. Human needs continue to increase and the need to travel has become part of daily activities. Almost everyone has a personal vehicle but not a few who prefer to use public transport. But to access the public transport is more troublesome because it must go through the terminals, stations, or airports.

Public transport facilities now there has not been adequate. In terms of cleanliness and timeliness of many who complained. But lately many emerging alternative vehicle favored by many people online transportation. Mushrooming online motorcycle-taxis and taxis online. With Ease in ordering that only use the app on a smartphone, we are just waiting driver picked us up and drove to the destination. It is certainly welcomed by the public because in addition to saving energy without the need for taxis or taxi to the base, or should be toward the road to search the vehicle, we can also save time.

In This undergraduate thesis, the transportation that be used is travel.. If the rampant online transportation now only reach the town, this undergraduate thesis provides choice with a range of between cities using Android smartphone the majority of Indonesian people use today. Ease of entering the pickup address and destination addresses using the Google Maps API that makes

it easy to pick-up by the fleet. Prices used are also based on the distance between the pickup location and destination location so no cheating by unscrupulous drivers.

Keywords: Android, Google Map API, Online Transportation, Travel

KATA PENGANTAR

Segala puji bagi Allah SWT yang senantiasa melancarkan dan memudahkan segala urusan terutama dalam penyelesaian tugas akhir yang berjudul: “RANCANG BANGUN APLIKASI TRAVEL ONLINE BERBASIS ANDROID”.

Atas selesainya tugas akhir ini, banyak pihak yang secara langsung maupun tidak langsung telah berkontribusi dalam membantu penyelesaian. Kepada pihak-pihak tersebut, penulis ingin menyampaikan penghormatan dan terimakasih. Diantaranya disebutkan sebagai berikut.

1. Allah SWT atas limpahan rahmat dan rezeki-Nya sehingga penulis dapat menyelesaikan tugas akhir.
2. Kedua orang tua penulis tugas akhir, Bapak Purwanto Adiprabowo dan Ibu Wahyu Indriati yang selalu memberikan dukungan, doa, perhatian dan kasih sayang, serta saudara-saudara Mas Yuda, Mbak Wulan.
3. Bapak Dwi Sunaryo selaku dosen pembimbing tugas akhir pertama dan Ibu Sarwosri selaku pembimbing tugas akhir kedua yang telah memberikan arahan dan dukungan dengan sabar hingga tugas akhir ini tuntas.
4. Bapak Darlis Herumurti selaku ketua jurusan dan seluruh dosen Teknik Informatika ITS, yang selama ini memberikan ilmu, wawasan baru, serta pengalaman kepada penulis.
5. Staf TU Teknik Informatika ITS yang senantiasa memudahkan segala urusan penulis di jurusan.
6. Rekan dan sahabat-sahabatku angkatan 2009 yang baik, kompak, dan tidak dapat disebutkan satu-persatu.
7. Bapak Radityo Anggoro, S.Kom., M.Sc selaku dosen wali dan koordinator Tugas Akhir jurusan Teknik Informatika ITS.
8. Pihak-pihak lain yang tidak sengaja terlewat dan tidak dapat penulis sebutkan satu-persatu.

Penulis telah berusaha maksimal dalam penyusunan tugas akhir ini. Harapannya, tugas akhir ini dapat bermanfaat bagi

kemajuan teknologi informasi dan dunia pendidikan di Indonesia. Penulis juga ingin menyampaikan permohonan maaf apabila terdapat kekurangan, kesalahan maupun kelalaian di dalamnya. Kritik dan saran yang membangun sangat diharapkan sebagai perbaikan selanjutnya.

Surabaya, Surabaya 2017

Prasetyati Riski Rahmawati

DAFTAR ISI

LEMBAR PENGESAHAN	v
Abstrak	vii
Abstract	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
BAB I	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah	3
1.4. Tujuan dan Manfaat	3
1.5. Metodologi Penulisan	4
1.5.2. Studi Literatur	4
1.5.3. Analisis dan Perancangan Perangkat Lunak	4
1.5.4. Implementasi Perangkat Lunak	5
1.5.5. Uji Coba dan Evaluasi	5
1.5.6. Penyusunan Buku Tugas Akhir	5
1.6. Sistematika Penulisan	6
BAB II TINJAUAN PUSTAKA	7
2.1. Travel	7
2.2. Transportasi Online	7
2.3. Android SDK(<i>Software Development Kit</i>)	7
2.4. MySQL	8
2.5. PHP (PHP Hypertext Preprocessor)	8
2.6. Retrofit	9
2.7. Google Maps Distance Matrix API	9
2.8. Referensi Aplikasi yang Sudah Ada	10
2.8.1. Gojek	10
2.8.2. Uber	10
2.8.3. Grab	10
BAB III ANALISIS DAN PERANCANGAN	13
3.1. Analisis	13

3.1.1 Analisis Permasalahan	13
3.1.2. Analisis Kebutuhan	14
3.1.3. Analisis Aktor	21
3.1.4. Arsitektur Sistem	22
3.2. Perancangan Antarmuka Pengguna	23
3.2.1. Rancangan Antarmuka Halaman Utama	23
3.2.2. Rancangan Antarmuka Halaman Jadwal	23
3.2.3. Rancangan Antarmuka Halaman Hasil	24
3.3. Perancangan Proses Sistem	25
BAB IV IMPLEMENTASI PERANGKAT LUNAK	27
4.1. Perangkat Bantu Implementasi Perangkat Lunak	27
4.2. Melihat Jadwal	27
4.3. Memesan Tiket	30
4.4. Menampilkan Rute, Jarak, dan Harga	30
BAB V UJI COBA DAN EVALUASI	33
5.1. Pengujian Fungsionalitas	33
5.1.1. Lingkungan Uji Coba Perangkat Lunak	33
5.1.2. Skenario Uji Coba Perangkat Lunak	33
5.1.3. Hasil Uji Coba Fungsionalitas	34
5.2. Pengujian Subjektif	38
5.2.1. Kriteria Penguji	38
5.2.3. Skenario Uji Coba	39
5.3. Evaluasi	42
5.3.2. Evaluasi Pengujian Subjektif	43
BAB VI PENUTUP	45
6.1. Kesimpulan	45
6.2. Saran	45
DAFTAR PUSTAKA	47
LAMPIRAN KODE SUMBER	49
LAMPIRAN FORMULIR PENILAIAN PENGUJIAN KEGUNAAN	57
BIODATA PENULIS	61

DAFTAR GAMBAR

Gambar 3. 1 Diagram Kasus Penggunaan	15
Gambar 3. 2 Diagram Aktivitas Melihat Jadwal	17
Gambar 3. 3 Diagram Aktivitas Memesan Tiket.....	19
Gambar 3. 4 Diagram Aktivitas Melihat Rute, Jarak, dan Harga	21
Gambar 3. 5 Arsitektur Sistem	22
Gambar 3. 6 Rancangan Antarmuka Halaman Utama	23
Gambar 3. 7 Rancangan Antarmuka Halaman Jadwal	24
Gambar 3. 8 Rancangan Antarmuka Halaman Hasil.....	24
Gambar 3. 9 Diagram Alur Kasus Penggunaan.....	25
Gambar 3. 10 Kelas Diagram Travel Online	26
Gambar 4. 1 Implementasi Membuat Tabel Jadwal	28
Gambar 4. 2 Implementasi Webservice Menampilkan Jadwal ...	28
Gambar 4. 3 Implementasi Mengambil Data dari <i>Database</i>	29
Gambar 4. 4 Implementasi Menampilkan Jadwal	29
Gambar 4. 5 Implementasi Membuat Tabel Penumpang	30
Gambar 4. 6 Implementasi Memesan Tiket	30
Gambar 4. 7 Implementasi Tampilan Peta	31
Gambar 4. 8 Implementasi Rute.....	31
Gambar 4. 9 Implementasi Jarak	32
Gambar 5. 1 Tampilan Antarmuka Jadwal.....	35
Gambar 5. 2 Tampilan Form Pemesanan	36
Gambar 5. 3 Tampilan Data Berhasil Masuk ke Database.....	37
Gambar 5. 4 Tampilan Rute, Jarak, dan harga	38

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 3. 1 Daftar Kode Diagram Kasus Penggunaan.....	15
Tabel 3. 2 Spesifikasi Kasus Melihat Jadwal	16
Tabel 3. 3 Spesifikasi Kasus Memesan Tiket.....	18
Tabel 3. 4 Spesifikasi Kasus Melihat Rute, Jarak, dan Harga.....	19
Tabel 5.1 Tabel Spesifikasi Lingkungan Uji Coba.....	33
Tabel 5. 2 Uji TO-001	34
Tabel 5. 3 Uji TO-002	36
Tabel 5. 4 Uji TO-003	37
Tabel 5. 5 Daftar Penguji Aplikasi	39
Tabel 5. 6 Penilaian Antarmuka Tampilan Awal Aplikasi.....	40
Tabel 5. 7 Penilaian Antarmuka Tampilan Hasil	41
Tabel 5. 8 Penilaian Tampilan Jadwal.....	41
Tabel 5. 9 Saran Perbaikan Aplikasi Selanjutnya	42
Tabel 5. 10 Evaluasi Pengujian Fungsionalitas.....	42
Tabel 5. 11 Rekapitulasi Hasil Pengujian Subjektif	44

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini dipaparkan garis besar tugas akhir yang meliputi latar belakang, tujuan dan manfaat pembuatan, rumusan dan batasan permasalahan, metodologi pembuatan tugas akhir, dan sistematika penulisan.

1.1. Latar Belakang

Transportasi adalah pemindahan manusia atau barang dari satu tempat ke tempat lainnya dengan menggunakan sebuah wahana yang digerakkan oleh tenaga manusia atau tenaga mesin. Transportasi digunakan untuk memudahkan manusia dalam melakukan aktivitas sehari-hari. Di negara maju, mereka biasanya menggunakan kereta bawah tanah dan taksi. Penduduk disana jarang yang mempunyai kendaraan pribadi karena mereka sebagian besar menggunakan angkutan umum sebagai transportasi mereka[1].

Alat transportasi merupakan salah satu kemajuan teknologi yang membantu manusia dalam berbagai kegiatan sehari-hari dalam menempuh perjalanan ke suatu tempat. Transportasi dapat berupa mobil pribadi, taksi, angkutan umum, kereta api, kapal laut bahkan pesawat terbang. Dan berbagai alat transportasi tersebut merupakan hasil dari pemikiran akal manusia dalam mempermudah seseorang mencapai tujuan yang jaraknya jauh dengan waktu yang relative lebih singkat atau cepat.

Mengamati fenomena angkutan umum, banyak manfaatnya, tapi ada pula kekurangannya. Transportasi online manfaatnya lebih terasa bagi penumpang sehingga transportasi online menjadi pilihan utama, bukan berarti mengesampingkan transportasi konvensional. Untuk di kota-kota besar memang seharusnya mengikuti jaman, yaitu secara online, jadi transportasi online lah solusinya. Untuk didaerah terpencil dan agak kepedalaman, atau desa terpencil transportasi pangkalan tetap masih diperlukan sebagai sarana transportasi.

Kebutuhan akan transportasi umum sekarang ini semakin meningkat. Banyak orang bepergian memilih menggunakan transportasi umum karena alasan kepraktisan tidak harus lelah menyetir. Masalah yang dihadapi dalam menggunakan transportasi umum antara lain kesulitan mendapatkan kendaraan umum apa yang dapat digunakan. Tidak mengetahui jadwal keberangkatan dan harus berangkat menuju tempat keberangkatan semisal terminal, stasiun, dan bandara yang memakan waktu dan biaya juga merupakan kendala dalam menggunakan transportasi umum. Travel merupakan salah satu transportasi umum yang memberikan kemudahan kepada penumpangnya. Bisa dijemput di alamat yang diinginkan dan diantarkan sampai tujuan. Hal ini memberikan efisiensi waktu dan tenaga.

Sekarang ini kita sudah berada di era yang segala sesuatunya telah terkoneksi dengan jaringan tanpa batas, internet. Sekarang ini internet tidak hanya sekedar berisi media sosial untuk saling menyapa dengan orang-orang yang terpisahkan jarak dan waktu, tapi jaringan internet yang telah berkembang begitu luas pun telah merambah hingga ke jasa transportasi.

Transportasi online sekarang ini semakin marak di masyarakat berupa ojek *online* dan taksi *online* seperti *gojek*, *uber*, *grab*, dll. Di Indonesia sendiri pengguna *smartphone* mencapai 69,4 juta pada tahun 2016 dengan pengguna OS Android sebesar 50,81% [2]. Kemudahan dalam memesan transportasi hanya dengan *smartphone* membuat masyarakat lebih tertarik menggunakan transportasi online dibandingkan dengan memesan transportasi secara konvensional. Perkembangan jasa transportasi online seperti ini tentunya bukan hal yang negatif tapi jauh lebih menguntungkan masyarakat kita sendiri karena kemudahan akses dan juga sistem yang jauh lebih transparan dan pastinya tepat sasaran. Dimulai dari kemudahan pemesanan hingga penghitungan tarif bahkan estimasi waktu dan jarak yang terhubung langsung dengan internet.

Tetapi aplikasi transportasi online yang tersedia sekarang ini hanya dengan jarak tempuh yang terbatas yaitu di dalam kota

sementara travel bisa menjangkau antar kota dalam provinsi dan antar kota luar provinsi.

Pada tugas akhir ini akan dibuat aplikasi Travel Online dengan sistem *point to point*, yaitu dengan sistem penjemputan di titik asal dan diantar sampai titik tujuan sehingga memudahkan masyarakat dalam bepergian. Pengguna dapat memesan pada *smartphone* android sesuai jadwal yang tersedia dan akan menampilkan biaya dari perhitungan jarak antara titik jemput sampai titik tujuan pada Google Map API.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, terdapat permasalahan yang dapat dirumuskan, yaitu sebagai berikut.

1. Bagaimana membuat aplikasi Travel Online?
2. Bagaimana mengintegrasikan aplikasi Travel Online dengan Google Map API?
3. Bagaimana menampilkan harga berdasarkan jarak yang ditempuh pada Google Map?

1.3. Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan yakni sebagai berikut.

1. Aplikasi ini berjalan pada platform Android untuk digunakan oleh pengguna.
2. *Server* aplikasi menggunakan PHP, sedangkan *client* menggunakan java serta SDK Android.
3. Kerangka kerja peta menggunakan Google Map API.
4. Uji Coba tim dilakukan di provinsi Jawa Timur.

1.4. Tujuan dan Manfaat

Tujuan tugas akhir ini yaitu mengembangkan aplikasi yang dapat memesan tiket travel secara online sehingga memberikan

kemudahan bagi para masyarakat dalam memesan travel menggunakan perangkat Android. Secara khusus, tujuan tugas akhir ini dijabarkan sebagai berikut.

1. Membuat Aplikasi Travel Online.
2. Mengintegrasikan aplikasi Travel Online dengan Google Map API
3. Dapat menampilkan harga berdasarkan jarak yang ditempuh pada Google Map.

1.5. Metodologi Penulisan

Metodologi yang digunakan dalam proses pelaksanaan tugas akhir kali ini adalah sebagai berikut.

1.5.1. Penyusunan Proposal Tugas Akhir

Tahap awal untuk memulai pengerjaan tugas akhir adalah penyusunan proposal tugas akhir. Pada proposal ini, diajukan gagasan rancang bangun aplikasi travel online berbasis android.

1.5.2. Studi Literatur

Pada studi literatur ini, akan dipelajari sejumlah referensi yang diperlukan dalam pembuatan aplikasi.

1.5.3. Analisis dan Perancangan Perangkat Lunak

Aktor dari aplikasi ini adalah calon penumpang travel. Fitur yang terdapat pada aplikasi ini adalah:

1. Melihat jadwal keberangkatan
2. Melihat informasi kursi yang tersedia
3. Memesan tiket
4. Mengonfirmasi pembayaran
5. Membatalkan tiket yang telah dipesan

1.5.4. Implementasi Perangkat Lunak

Implementasi merupakan tahap mengimplementasikan hasil rancangan yang dilakukan pada tahap sebelumnya. Pada tahap ini, setiap pustaka yang digunakan, digabungkan dan disesuaikan dengan kebutuhan perangkat lunak. Selain itu, setiap modul kelas, fungsi, antarmuka, dan sebagainya juga dibuat. Bahasa pemrograman yang digunakan dalam implementasi adalah Java.

1.5.5. Uji Coba dan Evaluasi

Pada tahap ini dilakukan uji coba terhadap aplikasi yang telah dikembangkan, dengan metode pengujian kotak hitam terhadap masukan dan keluaran yang dihasilkan berdasarkan skenario yang telah ditentukan.

1.5.6. Penyusunan Buku Tugas Akhir

Tahap terakhir merupakan penyusunan laporan yang memuat dokumentasi mengenai pembuatan serta hasil implementasi perancangan yang telah dibuat. Buku tugas akhir yang dibuat, terdiri dari beberapa bagian yaitu sebagai berikut.

1. Pendahuluan
 - 1.1 Latar Belakang
 - 1.2 Permasalahan
 - 1.3 Batasan Tugas Akhir
 - 1.4 Tujuan
 - 1.5 Metodologi Penulisan
 - 1.6 Sistematika Penulisan
2. Tinjauan Pustaka
3. Analisis dan Perancangan
4. Implementasi
5. Uji Coba dan Evaluasi
6. Penutup
7. Daftar Pustaka

1.6. Sistematika Penulisan

Sistematika penulisan buku tugas akhir dibagi menjadi beberapa bab sebagai berikut.

Bab 1 Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat dari pembuatan tugas akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika tugas akhir.

Bab 2 Tinjauan Pustaka

Bab ini membahas tentang teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini.

Bab 3 Analisis dan Perancangan

Bab ini membahas analisis dan perancangan perangkat lunak. Rancangan perangkat lunak meliputi rancangan data, arsitektur, dan proses.

Bab 4 Implementasi

Bab ini membahas cara mengimplementasikan hasil rancangan perangkat lunak ke dalam kode program dan pembuatan antarmuka.

Bab 5 Uji Coba dan Evaluasi

Bab ini membahas tentang uji coba subjektif dari aplikasi yang telah dikembangkan. Uji coba dan evaluasi langsung dilakukan oleh pengembang perangkat lunak menggunakan pengujian kotak hitam.

Bab 6 Penutup

Bab ini berisi kesimpulan dan saran dari hasil uji coba yang dilakukan.

BAB II

TINJAUAN PUSTAKA

Pada bab ini dibahas mengenai tinjauan pustaka yang menjadi dasar dari pembuatan tugas akhir. Berikut ini penjelasan secara khusus masing-masing tinjauan pustaka.

2.1. Travel

Travel (Point to Point) adalah sebuah layanan angkutan penumpang dari terminal / pool kota asal ke terminal / pool kota tujuan sesuai trayek / jurusan yang dilayani travel tersebut. Layanan travel memiliki beberapa keuntungan bagi konsumen antara lain lokasi cabang/ outlet berada pada lokasi strategis, cabang/outlet yang memiliki standar kenyamanan, keberangkatan terjadwal dan *on time, seat* penumpang yang terbatas, driver yang santun dengan melalui seleksi penerimaan yang ketat dan training yang berkelanjutan serta perlindungan asuransi kepada penumpang.

2.2. Transportasi Online

Transportasi online memiliki arti bahwa seluruh transaksi yang dilakukan melalui basis online, menggunakan media *smartphone*, aplikasi terkait dan penghubung internet. Contoh bisnis transportasi online yang ada saat ini adalah Go-Jek, Grabtaxi / Grabbike, Uber dan Jeger Taksi. Keuntungan dari menggunakan transportasi online antara lain, dapat menikmati jasa transportasi secara instan, pembayaran yang transparan dan pelayanan *door to door* yang tercepat [3].

2.3. Android SDK(*Software Development Kit*)

Android SDK adalah tools API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang

meliputi sistem operasi, *middleware* dan aplikasi kunci yang *direlease* oleh Google. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk memulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Android SDK juga berisi *librari* untuk mengikat aplikasi ke fitur inti Android seperti yang berhubungan dengan fungsi ponsel (membuat dan menerima panggilan), fungsi GPS, dan pesan teks [4].

2.4. MySQL

MySQL merupakan suatu *software* sistem manajemen database yang *open source*. MySQL adalah database *server* yang dibuat dan didistribusikan oleh perusahaan komersial yaitu MySQL AB. MySQL didistribusikan secara gratis di bawah lisensi *General Public License* (GPL). MySQL adalah sistem yang mendukung relational database. Artinya, dalam sebuah database memiliki beberapa table untuk menyimpan data–data dimana masing–masing tabel memiliki hubungan atau relasi satu sama lain sehingga dapat dilakukan kombinasi data dari beberapa tabel dalam satu saat. Sistem semacam ini sering disebut pula dengan RDBMS (*Relational DataBase Management System*). Sistem manajemen *database* seperti MySQL diperlukan untuk menambahkan, mengakses, memproses data yang disimpan di *server*[5].

2.5. PHP (PHP Hypertext Preprocessor)

PHP adalah *server-side embedded script language* yang didesain khusus untuk web[5], yaitu *script* berisi *syntax* perintah yang sepenuhnya dijalankan oleh *server* tetapi disertakan pada halaman HTML. PHP merupakan *script* yang terbilang baru dan tersedia secara bebas, dan masih memungkinkan untuk dikembangkan lebih lanjut. PHP dapat diintegrasikan (*embedded*) ke dalam *web server*, atau dapat berperan sebagai program CGI

yang terpisah. Ketika script PHP dijalankan, maka server akan melakukan hal-hal sebagai berikut:

1. Membaca permintaan dari client/browser
2. Mencari halaman/page di server
3. Melakukan instruksi yang diberikan oleh PHP untuk melakukan modifikasi pada halaman
4. Mengirim kembali halaman tersebut kepada client (browser) melalui internet atau intranet.

2.6. Retrofit

Retrofit adalah tipe *safe REST client library* modern pada Android dan Java yang dibuat oleh Square Inc. Retrofit menyediakan cara yang mudah untuk otentikasi dan berinteraksi dengan berbagai API dan memungkinkan mengirim permintaan jaringan dengan OkHttp atau HttpURLConnection. *Library*-nya memperoleh data JSON atau XML dari *RESTful web service* dan ketika respon diterima, akan diparsing sebagai *Object Plain Old Java* yang dispesifikasi untuk objek pada respon. Kegunaan *library* untyk memparsing JSON dan GSON didukung untuk serialisasi dan parsing otomatis. Retrofit bekerja dengan REST API menggunakan implementasi *interface* Java yang dapat dihasilkan dengan bantuan *RestAdapter*. Implementasi pada hal ini bertindak seperti contoh local dari *service* dan setiap panggilan sesuai dengan HTTP *request* [6].

2.7. Google Maps Distance Matrix API

Google Maps Distance Matrix API adalah layanan yang menyediakan waktu dan jarak perjalanan untuk matriks tempat asal dan tujuan. Informasi yang dikembalikan berdasarkan rute yang disarankan antara titik awal dan titik akhir, sesuai perhitungan Google Maps API, dan terdiri dari baris-baris berisi nilai *duration* dan *distance* untuk setiap pasang. Layanan ini tidak mengembalikan informasi rute detail. Informasi rute bisa

diperoleh dengan meneruskan satu tempat asal dan tujuan yang diinginkan ke Google Maps Directions API [7].

2.8. Referensi Aplikasi yang Sudah Ada

2.8.1. Gojek

GoJek adalah sebuah layanan booking ojek melalui aplikasi GoJek yang bisa didownload di Smartphone android & iPhone. GoJek menawarkan 4 (empat) jasa layanan yang bisa dimanfaatkan oleh para pelanggannya: Instant Courier (Pengantaran Barang), Transport (Jasa Angkutan), Shopping (Belanja) dan Corporate (Kerjasama dengan perusahaan untuk jasa kurir) yang menekankan keunggulan dalam Kecepatan, Inovasi dan Interaksi Sosial [8].

2.8.2. Uber

Uber merupakan layanan mobil *on-demand* (berbasis permintaan) yang memungkinkan meminta layanan sopir pribadi melalui aplikasi pada iPhone atau perangkat Android. Layanan ini menggunakan aplikasi khusus untuk mengirim sopir terdekat ke lokasi pengguna. Uber akan mengirimkan mobil pribadi dengan “sentuhan” taksi. Layanan ini menyediakan solusi pembayaran non-tunai, dengan biaya yang dikenakan langsung ke tagihan kartu kredit yang terhubung dengan akun penumpang [9].

2.8.3. Grab

Layanan Grab dikenal karena kemudahannya dalam memesan kendaraan yang akan Anda gunakan dalam perjalanan. Grab secara otomatis dapat mendeteksi lokasi pemesan jadi tinggal menentukan tujuan perjalanan. Layanan Grab dapat memantau secara langsung lokasi dari pengemudi Grab dan berkomunikasi langsung dari aplikasinya. Grab juga menunjukkan nomor polisi dari kendaraan yang akan mengantar. Pilihan pembayaran yang disediakan oleh Grab pun sangat

beragam. Anda dapat menggunakan uang tunai, kartu kredit, dan pembayaran via ponsel cerdas setelah sampai di tujuan. Grab juga menyediakan struk elektronik yang akan dikirimkan melalui surat elektronik yang terdaftar [10].

[Halaman ini sengaja dikosongkan]

BAB III

ANALISIS DAN PERANCANGAN

Bab ini membahas tahap analisis permasalahan dan perancangan dari system yang akan dibangun. Analisis permasalahan membahas permasalahan yang diangkat dalam pengerjaan Tugas Akhir. Analisis kebutuhan mencantumkan kebutuhan-kebutuhan yang diperlukan perangkat lunak. Selanjutnya dibahas mengenai perancangan sistem yang dibuat dalam perancangan ini adalah pendekatan berorientasi objek. Perancangan direpresentasikan dengan diagram UML.

3.1. Analisis

Tahap analisis dibagi menjadi beberapa bagian antara lain cakupan permasalahan, deskripsi umum sistem, kasus penggunaan sistem, dan kebutuhan perangkat lunak.

3.1.1 Analisis Permasalahan

Permasalahan utama yang diangkat dalam pembuatan Tugas Akhir ini adalah bagaimana membuat sebuah aplikasi pemesanan transportasi berupa travel secara online. Di era sekarang ini di mana masyarakat dituntut untuk melakukan aktivitas *mobile*, tentu transportasi merupakan kebutuhan yang penting. Banyak yang mempunyai kendaraan pribadi, tapi tidak sedikit pula yang tidak mempunyai sehingga untuk melakukan kegiatan bepergian harus menggunakan transportasi umum. Yang mempunyai kendaraan pribadi pun tidak selalu menggunakan kendaraan pribadinya sendiri. Kadang kala mereka pun memilih menggunakan transportasi umum dengan alasan lebih menghemat energi sehingga tidak perlu mengendarai kendaraan sendiri.

Dengan maraknya transportasi online saat ini tentu sangat membantu masyarakat dalam bepergian. Adanya ojek online dan taksi online yang menjamur memberi kemudahan masyarakat dengan menghemat waktu karena tinggal memesan lewat

smarthphone yang dimiliki, tinggal menunggu *driver* menjemput. Tetapi taksi online dan ojek online biasanya hanya mencakup dalam wilayah satu kota saja. Untuk bisa memanfaatkan transportasi online dengan jarak yang lebih luas seperti antar kota, transportasi online yang ada saat ini belum menyanggupi.

3.1.2. Analisis Kebutuhan

Aplikasi travel online merupakan aplikasi yang bisa melakukan pemesanan travel dan menampilkan jawal, rute, jarak dan harga. Dalam pembuatan aplikasi dilakukan analisis kebutuhan sistem melingkupi proses bisnis, kebutuhan fungsional, kebutuhan non-fungsional, dan arsitektur sistem.

3.1.2.1. Proses Bisnis

Proses bisnis sistem secara umum dapat dibagi menjadi 3 , antara lain:

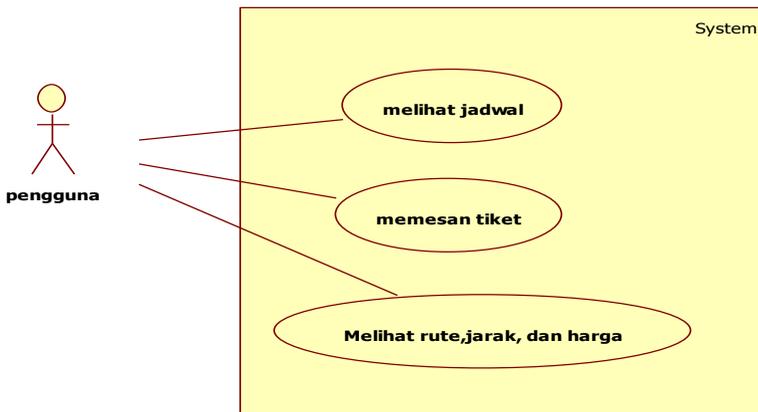
- i. Menampilkan jadwal
- ii. Memesan tiket
- iii. Menampilkan hasil berupa rute, jarak, waktu, dan harga.

Ketiga proses bisnis di atas merupakan fitur-fitur dari sistem. Fitur di atas dapat digunakan dengan beberapa kondisi dan tahapan proses awal. Pada saat pengguna memulai aplikasi, maka akan muncul tampilan awal tempat memesan tiket dan tombol untuk menampilkan jadwal.

Pada fitur menampilkan jadwal, pengguna bisa melihat jadwal yang terdiri dari rute kota asal dan kota tujuan, hari, dan jam keberangkatan. Pada fitur memesan tiket, pengguna harus memasukkan nama dan no hp sebagai data pemesan yang akan disimpan di *server*. Pada fitur menampilkan hasil, akan ditampilkan peta. Pada tampilan tersebut diminta memasukkan alamat asal dan alamat tujuan. Kemudian untuk meneruskan pesanan bisa mengklik tombol rute sehingga akan dimunculkan rute dari tempat berangkat menuju tempat tujuan, jarak, waktu, dan harga. Semua data sudah tersimpan di *database server*.

3.1.2.2. Kebutuhan Fungsional

Aplikasi dibangun berbasis Android. Aplikasi bisa digunakan siapa saja yang ingin menggunakan jasa travel. Kebutuhan fungsionalitas dari aplikasi travel online ini dideskripsikan pada kasus penggunaan Gambar 3.1. Pada kasus penggunaan, tiap kasus memiliki kondisi tertentu untuk memulai. Penjelasan kasus penggunaan dapat dilihat pada tabel 3.1.



Gambar 3. 1 Diagram Kasus Penggunaan

Tabel 3. 1 Daftar Kode Diagram Kasus Penggunaan

Kode Kasus Penggunaan	Nama	Deskripsi
TO-01	Melihat Jadwal	Pengguna dapat melihat jadwal keberangkatan travel berupa rute kota asal – kota tujuan, hari dan waktu keberangkatan
TO-02	Memesan tiket	Pengguna memesan tiket dengan memasukkan data diri yang terdiri dari nama dan nomor hp

Kode Kasus Penggunaan	Nama	Deskripsi
TO-03	Melihat rute, jarak, dan harga	Pengguna dapat melihat rute, jarak, dan harga jika sudah memasukkan alamat asal dan alamat tujuan

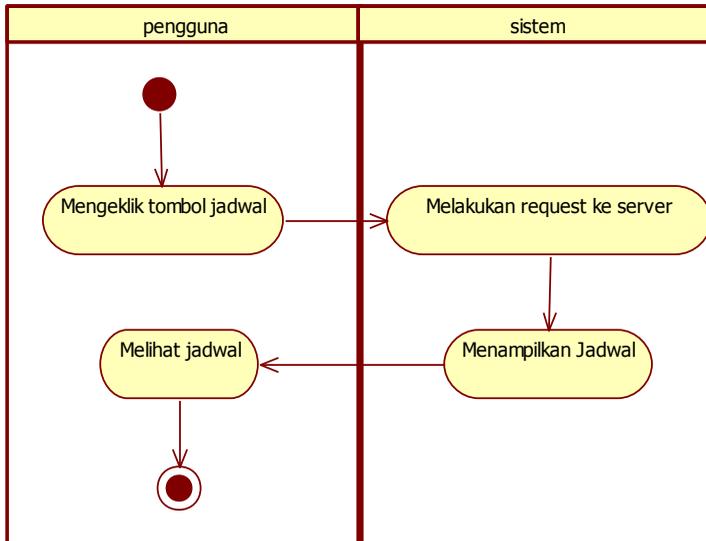
A. Melihat Jadwal

Pada kasus penggunaan ini, sistem menerima masukan berupa peneklikan tombol oleh pengguna. Kemudian sistem akan menampilkan jadwal yang telah disimpan di *server* dan ditampilkan pada halaman jadwal. Spesifikasi kasus melihat jadwal ini dapat dilihat pada Tabel 3.2. Diagram aktivitas dari kasus melihat jadwal bisa dilihat pada tabel 3.2.

Tabel 3. 2 Spesifikasi Kasus Melihat Jadwal

Nama	Melihat Jadwal
Kode	TO-01
Deskripsi	Sistem menampilkan jadwal dari server berupa rute kota asal – kota tujuan, hari dan waktu keberangkatan
Tipe	Fungsional
Pemicu	Sistem akan menerima respon jika pengguna menekan tombol jadwal
Aktor	Pengguna
Kondisi Awal	-
Aliran: -Kejadian Normal	A. Sistem menerima respon ketika tombol jadwal diklik B. Sistem akan mengirim <i>request</i> ke <i>server</i> C. <i>Server</i> merespon dan mengirimkan jadwal D. Sistem menampilkan jadwal
-Kejadian	-

alternatif	
Kondisi Akhir	Sistem menenampikan jadwal
Kebutuhan Khusus	Sistem tehubung koneksi internet



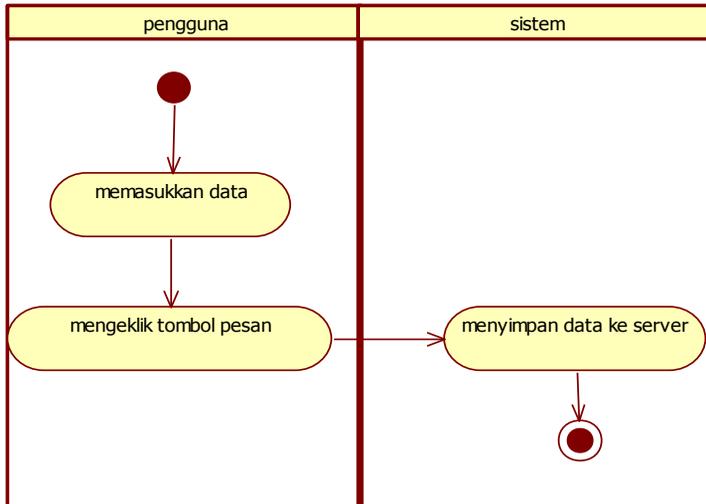
Gambar 3. 2 Diagram Aktivitas Melihat Jadwal

B. Memesan Tiket

Pada kasus penggunaan ini, sistem menerima data dari pengguna berupa nama dan nomor hp, kemudian sistem meneruskan agar idsimpak ke server. Spesifikasi kasus melihat jadwal ini dapat dilihat pada Tabel 3.3. Diagram aktivitas dari kasus melihat jadwal bisa dilihat pada tabel 3.3

.Tabel 3. 3 Spesifikasi Kasus Memesan Tiket

Nama	Memesan Tiket
Kode	TO-02
Deskripsi	Sistem menerima masukan data berupa nama dan nomor hp, kemudian menyimpan ke <i>server</i>
Tipe	Fungsional
Pemicu	Sistem akan menerima respon jika pengguna menekan tombol pesan
Aktor	Pengguna
Kondisi Awal	-
Aliran: -Kejadian Normal	A. Sistem menerima data yang dimasukkan pengguna setelah pengguna mengklik tombol pesan B. Sistem akan meneruskan data ke <i>server</i> untuk disimpan.
-Kejadian alternatif	-
Kondisi Akhir	Sistem menyimpan data
Kebutuhan Khusus	Sistem terhubung koneksi internet



Gambar 3. 3 Diagram Aktivitas Memesan Tiket

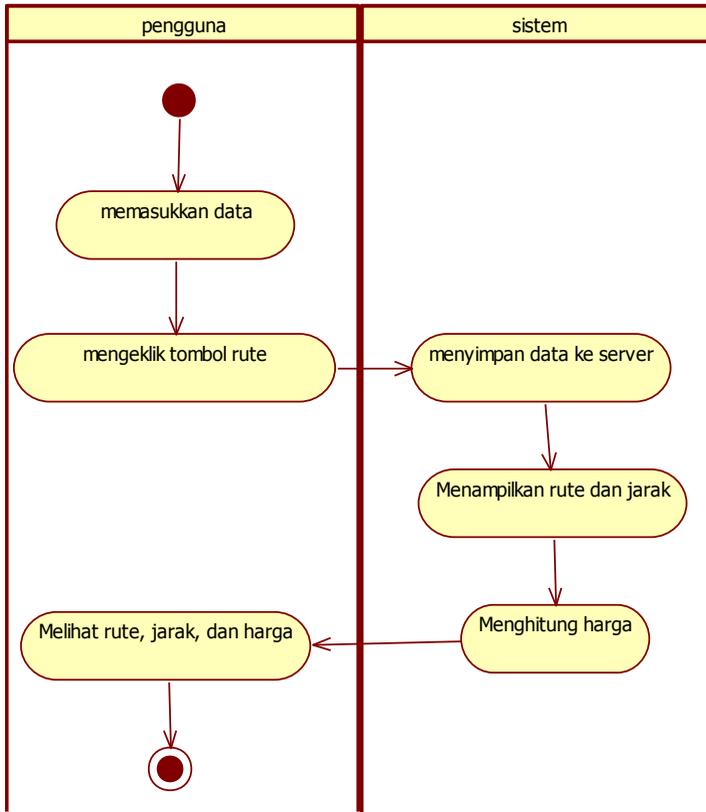
C. Melihat Rute, Jarak, dan Harga

Pada kasus penggunaan ini, sistem menerima data dari pengguna berupa nama dan nomor hp, kemudian sistem meneruskan agar idsimpak ke server. Spesifikasi kasus melihat jadwal ini dapat dilihat pada Tabel 3.4. Diagram aktivitas dari kasus melihat jadwal bisa dilihat pada tabel 3.4.

Tabel 3. 4 Spesifikasi Kasus Melihat Rute, Jarak, dan Harga

Nama	Melihat Rute, Jarak, dan Harga
Kode	TO-03
Deskripsi	Sistem menampilkan rute, jarak, dan harga setelah pengguna memasukkan alamat asal dan alamat tujuan. Harga didapatkan dari jarak antara alamat asal dan tujuan dikalikan harga per kilometer.

Tipe	Fungsional
Pemicu	Sistem akan menerima respon jika pengguna menekan tombol rute
Aktor	Pengguna
Kondisi Awal	-
Aliran: -Kejadian Normal	<p>A. Sistem menerima data berupa alamat asal dan tujuan yang dimasukkan pengguna setelah pengguna mengeklik tombol pesan</p> <p>B. Sistem akan meneruskan data ke <i>server</i> untuk disimpan.</p> <p>C. Sistem menampilkan rute dalam bentuk <i>path</i> pada <i>map</i>, titik asal dan titik tujuan terdapat penanda.</p> <p>D. Sistem menampilkan jarak dari rute</p> <p>E. Sistem menampilkan harga dari hasil perkalian jarak dan harga per kilometer</p>
-Kejadian alternatif	-
Kondisi Akhir	Sistem menampilkan rute, jarak, dan harga
Kebutuhan Khusus	Sistem terhubung koneksi internet



Gambar 3. 4 Diagram Aktivitas Melihat Rute, Jarak, dan Harga

3.1.3. Analisis Aktor

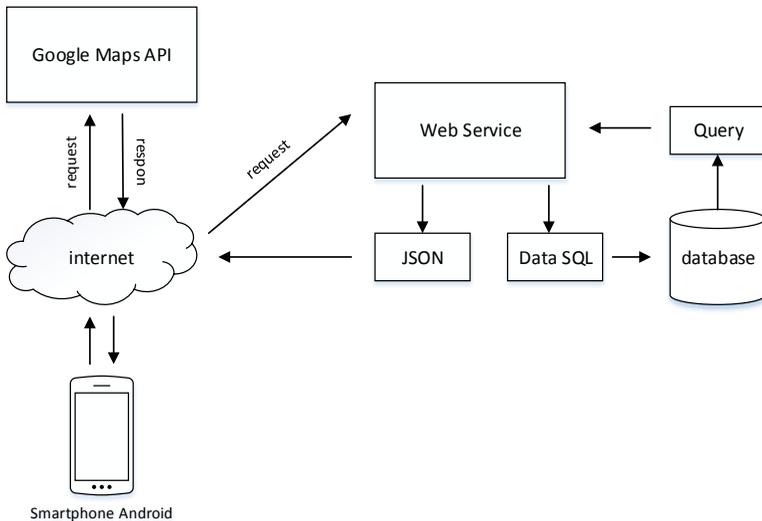
Aktor mendefinisikan entitas-entitas yang terlibat dan berinteraksi langsung dengan sistem. Entitas ini bisa berupa manusia maupun sistem atau perangkat lunak yang lain. Aktor

yang terdapat pada sistem ini hanya memiliki sebuah peran yaitu sebagai pengguna. Pengguna perangkat ini adalah orang yang menggunakan jasa travel online.

3.1.4. Arsitektur Sistem

Aplikasi travel online dibangun berbasis Android dan menggunakan *webservice* untuk mengirim dan menerima data dari aplikasi ke server. Untuk menjalankan aplikasi ini dibutuhkan koneksi internet untuk komunikasi antara klien dan *server*. Data yang dikirim dan diterima baik oleh Android maupun *server* tidak bisa dijalankan jika pada *smarthphone* tidak ada jaringan internet.

Pada Praktik menampilkan rute, jarak, dan harga, menggunakan Google Maps Distance Matrix API sehingga dibutuhkan juga koneksi internet agar fitur ini berjalan. Arsitektur yang digunakan pada sistem digambarkan pada Gambar 3.5.



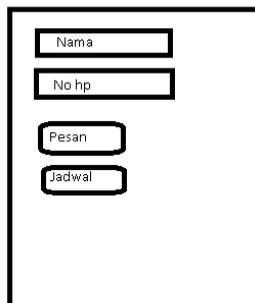
Gambar 3.5 Arsitektur Sistem

3.2. Perancangan Antarmuka Pengguna

Pada subbab ini akan dijelaskan mengenai beberapa rancangan antarmuka utama perangkat lunak yang memfasilitasi perancang atau produsen pakaian dalam berinteraksi dengan fitur-fitur pada perangkat lunak. Rancangan yang dibahas meliputi kendali pada antarmuka dan ketentuan masukan. Berikut ini penjabaran beberapa rancangan antarmuka tersebut.

3.2.1. Rancangan Antarmuka Halaman Utama

Halaman utama perangkat lunak tampak seperti ditampilkan pada Gambar 3.6. Terdapat dua menu yang dapat diakses yaitu melihat jadwal dan memesan tiket.



The diagram shows a rectangular frame containing four input fields stacked vertically. The first field is labeled 'Nama', the second 'No hp', the third 'Pesan', and the fourth 'Jadwal'. Each field is represented by a rounded rectangle with a thin border and a small gap between the text and the border.

Gambar 3. 6 Rancangan Antarmuka Halaman Utama

3.2.2. Rancangan Antarmuka Halaman Jadwal

Halaman utama perangkat lunak tampak seperti ditampilkan pada **Error! Reference source not found.** Gambar .7 berupa daftar jadwal.

Hari Rute Waktu berangkat
Hari Rute Waktu berangkat
Hari Rute Waktu berangkat

Gambar 3. 7 Rancangan Antarmuka Halaman Jadwal

3.2.3. Rancangan Antarmuka Halaman Hasil

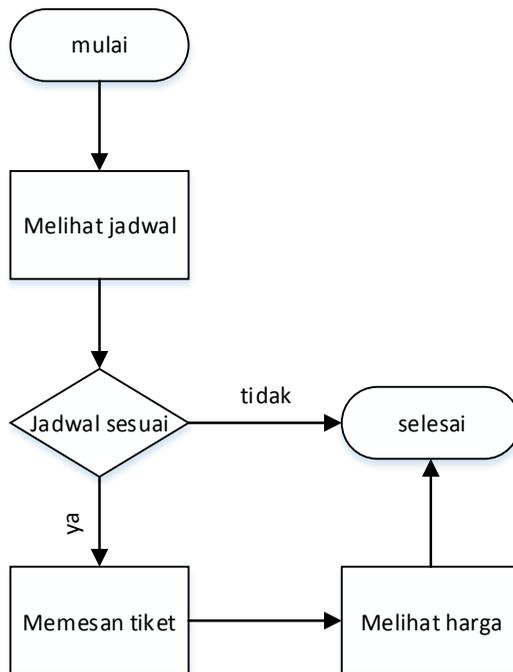
Halaman utama perangkat lunak tampak seperti ditampilkan pada Gambar 3.8, berupa daftar peta, inputan alamat, dan harga.

Alamat asal	
Alamat Tujuan	
Rute	Harga

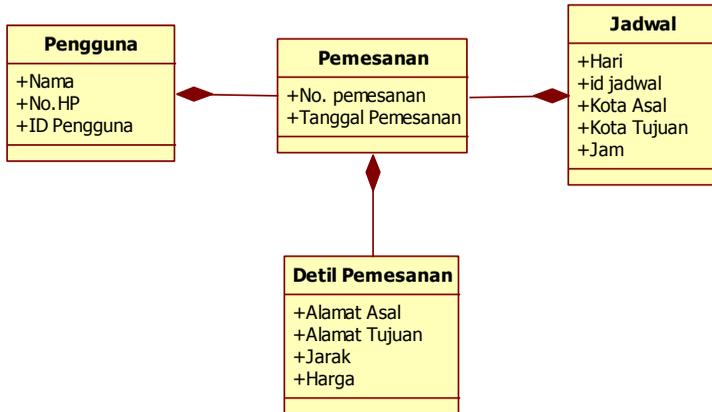
Gambar 3. 8 Rancangan Antarmuka Halaman Hasil

3.3. Perancangan Proses Sistem

Pada aplikasi travel online, proses untuk melakukan pemesanan tiket pertama kali adalah dengan melihat jadwal. Calon penumpang bisa melihat jadwal yang tersedia pada halaman jadwal. Jika ada jadwal yang sesuai dengan keinginan, penumpang bisa melanjutkan pemesanan dengan kembali ke halaman awal lalu memasukkan identitas berupa nama dan nomor hp. Jika sudah mengisikan data diri, pengguna bisa melanjutkan memasukkan alamat penjemputan dan alamat tujuan. Aplikasi akan menampilkan rute, jarak dan biaya berdasarkan alamat yang dimasukkan pengguna. Diagram alur bisa dilihat pada Gambar 3.9. dan kelas diagram pada gambar 3.10.



Gambar 3. 9 Diagram Alur Kasus Penggunaan



Gambar 3. 10 Kelas Diagram Travel Online

BAB IV

IMPLEMENTASI PERANGKAT LUNAK

Pada bab ini akan dijelaskan mengenai implementasi tugas akhir berdasarkan perancangan perangkat lunak, perangkat keras, dan integrasi. Fase perancangan menjadi tahap dasar dan acuan dalam implementasi tugas akhir namun tidak menutup kemungkinan terjadi perubahan-perubahan minor jika dirasa perlu.

4.1. Perangkat Bantu Implementasi Perangkat Lunak

Aplikasi diimplementasikan dan dikembangkan dalam lingkungan pemrograman dengan spesifikasi berikut ini:

1. Windows 10.
2. Android Studio sebagai IDE untuk pembuatan aplikasi pada Android.
3. Android Software Development Kit.
4. Android Development Tools.
5. Ponsel pintar Android Xiaomi Redmi 1S.
6. Emulator Android Studio
7. XAMPP sebagai Apache *server*.

4.2. Melihat Jadwal

Pada proses melihat jadwal, sistem akan menampilkan data yang diambil dari *database server*. Sistem menggunakan retrofit untuk mengubah REST API ke *Interface* Java. Untuk mengubah JSON string menjadi JAVA objek menggunakan *library* GSON. Gambar 4.1 adalah kode SQL untuk membuat tabel jadwal.

```
CREATE TABLE `coba_json` (
  `id` int(11) NOT NULL,
  `hari` varchar(10) NOT NULL,
  `kotaasal` varchar(25) NOT NULL,
  `kotatujuan` varchar(25) NOT NULL,
  `jam` time NOT NULL
)
```

Gambar 4. 1 Implementasi Membuat Tabel Jadwal

, Gambar 4.2 adalah implementasi *webservice* untuk mengirim data ke aplikasi. Gambar 4.3 adalah kode untuk membuat *class* RestManager dan Gambar 4.4 implementasi menampilkan jadwal pada Android menggunakan RecyclerView.

```
require_once('dbConnect.php');
$sql = "SELECT hari, kotaasal, kotatujuan,
DATE_FORMAT(jam, '%H:%i') as jam from coba_json";
$result = mysqli_query($con,$sql);
$rows = array();
while($row = mysqli_fetch_assoc($result)){
    $rows[] = $row;
}
echo json_encode($rows);
mysqli_close($con);
?>
```

Gambar 4. 2 Implementasi Webservice Menampilkan Jadwal

```

public class TravelRestManager {
    private TravelRestService mRestService;
    public TravelRestService getmRestService() {
        if(mRestService == null){
            Retrofit retrofit = new
Retrofit.Builder()
                .baseUrl(Constants.BASE_URL)
                .addConverterFactory(GsonConverterFactory.create
                ()).build();

            mRestService =
                retrofit.create(TravelRestService.class);
        }

        return mRestService;
    }
}

```

Gambar 4.3 Implementasi Mengambil Data dari Database

```

    public void
onBindViewHolder(RecyclerView.ViewHolder
holder, int position) {
    JadwalVH vh = (JadwalVH) holder;
    Jadwal item = mItems.get(position);
    vh.tv_hari.setText(item.hari);
    vh.tv_rute.setText(item.kotaAsal + " -
" + item.kotaTujuan);
    vh.tv_jam.setText(item.jam);
    vh.itemView.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {
        }
    });
}

```

Gambar 4.4 Implementasi Menampilkan Jadwal

4.3. Memesan Tiket

Pada fitur memesan tiket, pengguna memasukkan nama dan nomor hp kemudian akan disimpan ke *server* oleh sistem. Gambar 4.5 adalah implementasi membuat tabel penumpang dan Gambar 4.6 adalah implementasi penyimpanan data pemesan.

```
CREATE TABLE `penumpang` (
  `ID_PENUMPANG` varchar(10) NOT NULL,
  `NAMA_PENUMPANG` text NOT NULL,
  `NO_HP` varchar(12) NOT NULL
)
```

Gambar 4. 5 Implementasi Membuat Tabel Penumpang

```
simpan.setOnClickListener(new
View.OnClickListener()
{
    public void onClick(View v)
    {
        ArrayList<NameValuePair> postParameters =
new ArrayList<NameValuePair>();

        postParameters.add(new
BasicNameValuePair("nama",
namax.getText().toString());
        postParameters.add(new
BasicNameValuePair("nohp",
nohpx.getText().toString());
    }); }
```

Gambar 4. 6 Implementasi Memesan Tiket

4.4. Menampilkan Rute, Jarak, dan Harga

Pada fitur Menampilkan Rute, Jarak, dan Harga, pertama pada tampilan ditampilkan peta dengan fragmen ditunjukkan dalam kode seperti pada Gambar 4.7. Untuk menampilkan rute

ditunjukkan pada Gambar 4.8 dan untuk Perhitungan jarak ditunjukkan pada gambar 4.9.

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/a
ndroid"
    xmlns:map="http://schemas.android.com/apk/res-
auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
<fragment
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMa
pFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
</LinearLayout>
```

Gambar 4. 7 Implementasi Tampilan Peta

```
public void onDirectionFinderStart() {
    progressDialog = ProgressDialog.show(this,
    "Please wait.",
        "Finding direction..!", true);
    if (originMarkers != null) {
        for (Marker marker : originMarkers) {
            marker.remove();
        }
    }
    if (destinationMarkers != null) {
        for (Marker marker :
    destinationMarkers) {
            marker.remove();
        }
    }
    if (polylinePaths != null) {
        for (Polyline polyline:polylinePaths )
    {
            polyline.remove();
        }
    }
}
```

Gambar 4. 8 Implementasi Rute

```

public void onDirectionFinderSuccess(List<Route>
routes) {
    progressDialog.dismiss();
    polylinePaths = new ArrayList<>();
    originMarkers = new ArrayList<>();
    destinationMarkers = new ArrayList<>();
    int cost = 1000;
    for (Route route : routes) {
mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(ro
ute.startLocation, 16));
        ((TextView)
findViewById(R.id.tvDuration)).setText(route.duratio
n.text);
        ((TextView)
findViewById(R.id.tvDistance)).setText(route.distanc
e.text);
        ((TextView)
findViewById(R.id.tvCost)).setText("14000");
        originMarkers.add(mMap.addMarker(new
MarkerOptions()
        .icon(BitmapDescriptorFactory.fromResource(R.drawabl
e.start_blue))
        .title(route.startAddress)
        .position(route.startLocation)));
destinationMarkers.add(mMap.addMarker(new
MarkerOptions()
        .icon(BitmapDescriptorFactory.fromResource(R.drawabl
e.end_green))
        .title(route.endAddress)
        .position(route.endLocation)));
        PolylineOptions polylineOptions = new
PolylineOptions().
            geodesic(true).
            color(Color.BLUE).
            width(10);
        for (int i = 0; i < route.points.size(); i++)
polylineOptions.add(route.points.get(i));
        polylinePaths.add(mMap.addPolyline(polylineOptions)
);
    }
}

```

Gambar 4. 9 Implementasi Jarak

BAB V

UJI COBA DAN EVALUASI

Bab ini membahas uji coba dan evaluasi perangkat lunak Travelia. Proses pengujian akan dilakukan menggunakan metode kotak hitam berdasarkan skenario yang telah ditentukan. Pengujian akan dilakukan sesuai dengan perancangan kasus penggunaan dan implementasi yang telah dijelaskan pada Bab III.

5.1. Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan untuk mengetahui kesesuaian keluaran dari setiap tahapan atau langkah penggunaan fitur terhadap skenario yang dipersiapkan (*correctness*). Berikut ini penjabaran proses pengujian fungsionalitas terhadap aplikasi.

5.1.1. Lingkungan Uji Coba Perangkat Lunak

Dalam proses pengujian perangkat lunak, dibutuhkan lingkungan uji coba yang disesuaikan standar kebutuhan. Aplikasi diuji coba dalam lingkungan spesifikasi seperti pada Tabel 5.1 berikut ini.

Tabel 5.1 Tabel Spesifikasi Lingkungan Uji Coba

No	Spesifikasi
1	<i>Xiaomi Redmi 1S, OS Android 4.4, 1 GB RAM</i>
2	<i>Emulator dari Android Studio</i>
3	<i>Koneksi menggunakan mifi Smartfren</i>

5.1.2. Skenario Uji Coba Perangkat Lunak

Pada subbab ini akan dijabarkan beberapa skenario ujicoba perangkat lunak secara mandiri berdasarkan metode kotak hitam sebagai dasar tolak ukur keberhasilan dari tujuan tugas akhir ini. Pengujian fungsionalitas fitur-fitur yang terdapat pada perangkat lunak akan dijabarkan sebagai berikut.

- Uji coba melihat jadwal
- Uji coba memesan tiket.
- Uji coba sinkronisasi data pada ponsel.

Berdasarkan daftar pengujian di atas, akan dilakukan beberapa skenario di setiap pengujian tersebut. Penjelasan mengenai cara dan hasil pengujian fungsionalitas perangkat lunak akan dibahas pada subbab selanjutnya yaitu Hasil Uji Coba Fungsionalitas.

5.1.3. Hasil Uji Coba Fungsionalitas

Subbab ini membahas hasil uji coba aplikasi menggunakan metode kotak hitam. Beberapa skenario yang telah dijabarkan di atas akan dijabarkan sebagai berikut

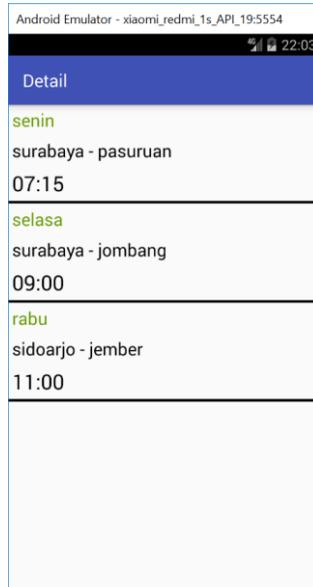
5.1.3.1. Melihat Jadwal

Tabel 5.2 menunjukkan proses uji dari menampilkan jadwal. Gambar 5.1 menampilkan tampilan jadwal pada aplikasi.

Tabel 5. 2 Uji TO-001

ID	UJI-TO-001
Nama	Uji coba menampilkan jadwal
Tujuan uji coba	Menampilkan jadwal
<i>Skenario 1</i>	<i>Aplikasi menampilkan tampilan jadwal dari server.</i>
Kondisi awal	Jadwal belum ditampilkan
Masukan	-
Keluaran yang diharapkan	<i>RecyclerView pada aplikasi android menampilkan jadwal</i>

ID	UJI-TO-001
Hasil uji coba	Berhasil.
Kondisi akhir	<i>RecyclerView pada aplikasi android berhasil menampilkan jadwal</i>



Gambar 5. 1 Tampilan Antarmuka Jadwal

. 5.1.3.2. Memesan Tiket

Tabel 5.3 menunjukkan proses uji dari memesan tiket, dan. Gambar 5.2 menampilkan tampilan untuk memasukkan data dan Gambar 5.3 menampilkan data berhasil masuk ke *database*

.Tabel 5. 3 Uji TO-002

ID	UJI-TO-002
Nama	Uji coba memesan tiket
Tujuan uji coba	Data berhasil masuk ke <i>database</i>
<i>Skenario 1</i>	<i>Aplikasi menampilkan tampilan memesan tiket</i>
Kondisi awal	Belum ada pemesanan
Masukan	Nama dan nomor hp
Keluaran yang diharapkan	<i>Data masuk ke database</i>
Hasil uji coba	Berhasil.
Kondisi akhir	<i>Data berhasil masuk ke database</i>

The image shows a screenshot of an Android emulator interface. At the top, it displays 'Android Emulator - samsung,redmi_1s_API_19-5554' and the time '22:18'. The main content is a form with two text input fields. The first field is labeled 'Nama' and the second is labeled 'No HP'. Below these fields, there are two buttons: a larger grey button labeled 'PESAN' and a smaller grey button labeled 'JADWAL'.

Gambar 5. 2 Tampilan Form Pemesanan

	id_penumpang	nama	password	no_hp
	1	lala	lalalala	081222333444

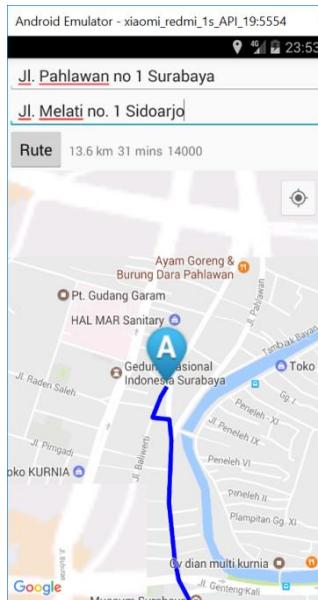
Gambar 5. 3 Tampilan Data Berhasil Masuk ke Database

5.1.3.2. Menampilkan Rute, Jarak, dan Harga

Tabel 5.4 menunjukkan proses uji dari menampilkan rute, jarak, dan harga, dan Gambar 5.4 menampilkan tampilan yang berhasil menampilkan rute, jarak, dan harga.

Tabel 5. 4 Uji TO-003

ID	UJI-TO-003
Nama	Uji coba menampilkan rute, jarak, dan harga,
Tujuan uji coba	Menampilkan rute, jarak, dan harga
<i>Skenario 1</i>	<i>Aplikasi menampilkan tampilan rute, jarak, dan harga</i>
Kondisi awal	Belum ada rute, jarak, dan harga
Masukan	Alamat asal dan alamat tujuan
Keluaran yang diharapkan	<i>Aplikasi menampilkan rute, jarak, dan harga</i>
Hasil uji coba	Berhasil.
Kondisi akhir	<i>Aplikasi berhasil menampilkan rute, jarak, dan harga</i>



Gambar 5. 4 Tampilan Rute, Jarak, dan harga

5.2. Pengujian Subjektif

Selain melakukan pengujian fungsionalitas dengan metode kotak hitam, juga perlu dilakukan pengujian subjektif untuk menilai *usability* perangkat lunak. Pengujian subjektif dilakukan untuk mengetahui penilaian dan tanggapan dari pengguna terhadap sejumlah aspek dari perangkat lunak ini (*usability*).

5.2.1. Kriteria Penguji

Pengujian subjektif dilakukan dengan 3 orang penguji.

5.2.2. Lingkungan Pengujian Perangkat Lunak

Penguji menggunakan perangkat ponsel pintar Android yang sama yaitu Xiaomi Redmi 1S..

5.2.3. Skenario Uji Coba

Dalam melakukan pengujian perangkat lunak, penguji diminta mencoba menggunakan perangkat lunak untuk menampilkan daftar semua berita, menampilkan daftar berita berdasarkan pada kota, menampilkan daftar berita berdasarkan pada tanggal. Jumlah penguji aplikasi ada 3 orang.

Untuk memberikan informasi tata cara penggunaan fitur aplikasi disediakan petunjuk penggunaan singkat yang tersedia pada aplikasi dan beberapa contoh gambar hasil penggunaan aplikasi ini.

Dalam memberikan penilaian dan tanggapan, penguji diberikan formulir pengujian perangkat lunak fitur ini. Formulir pengujian perangkat lunak fitur ini memiliki beberapa aspek penilaian dan pada bagian akhir terdapat saran untuk perbaikan aplikasi. formulir untuk pengujian subyektif aplikasi bisa dilihat pada lampiran A.

5.2.4. Hasil Uji Coba Pengujian Subyektif

Berikut ini adalah hasil uji coba perangkat lunak oleh beberapa para penguji. Pengujian yang meliputi aspek antarmuka *dashboard*, penampil antar muka penampil daftar berita, antarmuka penampil berita, aspek *update* berita, serta saran perbaikan aplikasi selanjutnya.

5.2.4.1. Daftar Penguji Perangkat Lunak

Pada Tabel 5.5 ditunjukkan perwakilan daftar penguji dari berbagai profesi.

Tabel 5. 5 Daftar Penguji Aplikasi

No	Nama	Profesi
1	Yati Kusuma Hartanti	Mahasiswa
2	Naylil Rohimah	Guru

No	Nama	Profesi
3	Juli Patricia	Pegawai Swasta
4	Abdulrahman Faruk	Programer
5	Yeye Choiriyah	Desain Grafis

5.2.4.2. Penilaian Antarmuka Tampilan Awal Aplikasi

Tabel 5.6 menunjukkan hasil rekapitulasi penilaian pengujian untuk aspek antarmuka tampilan menu. Berikut ini hasil rekapitulasi tersebut.

Tabel 5. 6 Penilaian Antarmuka Tampilan Awal Aplikasi

No.	Antarmuka Tampilan Menu Aplikasi	Penilaian				Rata-Rata
		1	2	3	4	
1	Kemudahan	0	2	3	0	2,5
2	Kejelasan menu	0	0	5	0	3
3	Kecepatan akses menu	0	0	0	5	4
Nilai Akhir						3,83

5.2.4.3. Penilaian Antarmuka Tampilan Hasil

Tabel 5.7 menunjukkan hasil rekapitulasi penilaian pengujian untuk aspek antarmuka tampilan hasil. Berikut ini hasil rekapitulasi tersebut.

Tabel 5. 7 Penilaian Antarmuka Tampilan Hasil

No.	Antarmuka Tampilan Hasil	Penilaian				Rata-Rata
		1	2	3	4	
1	Kejelasan menu	0	0	0	5	4
2	Kecepatan pengolahan data	0	0	2	3	3,5
3	Kejelasan menu	0	1	1	3	2,8
Nilai Akhir						3,43

5.2.4.3. Penilaian Antarmuka Tampilan Jadwal

Tabel 5.8Error! Reference source not found. enunjukkan hasil rekapitulasi penilaian penguji untuk aspek tampilan jadwal. Berikut ini hasil rekapitulasi tersebut. Sedangkan pada Tabel 5.9 menunjukkan saran dari penguji Berikut ini hasil rekapitulasi tersebut.

Tabel 5. 8 PenilaianTampilan Jadwal

No.	Antarmuka Tampilan Hasil	Penilaian				Rata-Rata
		1	2	3	4	
1	Kejelasan tulisan	0	0	0	5	4
2	Kecepatan akses menu	0	0	0	5	4
Nilai Akhir						4

Tabel 5. 9 Saran Perbaikan Aplikasi Selanjutnya

No.	Saran Perbaikan Aplikasi Selanjutnya
1	Tampilan jadwal warna background nya lebih cerah
2	Diberi keterangan pemesanan yang detail pada akhir pemesanan
3	Bisa diberi notif kalau travel sudah berangkat
4	Ada fitur <i>member</i>
5	Tampilan awal lebih diberi sentuhan desain

5.3. Evaluasi

Berdasarkan hasil uji coba fungsionalitas (*correctness*) dan uji coba subjektif (*usability*) di atas, maka dapat direkapitulasi dan disimpulkan hasil pengujian tersebut sebagai berikut.

5.3.1. Evaluasi Pengujian Fungsionalitas

Berikut ini rekapitulasi hasil pengujian fungsionalitas perangkat lunak yang dilakukan berdasarkan skenario uji coba yang telah dilakukan. Secara khusus, rekapitulasi tersebut dijelaskan pada Tabelberikut ini.

Tabel 5. 10 Evaluasi Pengujian Fungsionalitas

ID	Nama Pengujian		Hasil Pengujian
UJI-TO-001	Melihat jadwal	Skenario 1	Berhasil
UJI-TO-002	Memesan tiket	Skenario 1	Berhasil

UJI-TO-003	Menampilkan Rute, Jarak, dan Harga	Skenario 1	Berhasil
------------	------------------------------------	------------	----------

Melihat hasil evaluasi pada Tabel 5.10 **Error! Reference source not found.**, pengujian fungsionalitas telah berhasil dilakukan. Dari UJI-TO-001 sampai dengan UJI-TO-003. Hal itu menunjukkan bahwa implementasi aplikasi travel online berhasil dilakukan berdasarkan kriteria uji coba kebenaran fungsionalitas.

5.3.2. Evaluasi Pengujian Subjektif

Berikut ini rekapitulasi hasil pengujian subjektif secara keseluruhan mengenai penggunaan perangkat lunak yang dilakukan oleh 5 orang penguji dari berbagai kalangan. Sengaja untuk pengujian kali ini dipilih 5 orang penguji dengan latar belakang yang berbeda yaitu:

- 1) latar belakang desain,
- 2) latar belakang IT,
- 3) latar belakang pegawai swasta,
- 4) latar belakang guru,
- 5) latar belakang mahasiswa.

Pengujian sengaja dilakukan dengan latar belakang berbeda agar hasil pengujian lebih bersifat universal. Adapun penjelasan rekapitulasi pengujian tersebut dapat dilihat pada Tabel 5.11.

Tabel 5. 11 Rekapitulasi Hasil Pengujian Subjektif

No	Nama Pengujian		Rata-Rata	Nilai Akhir	Prosentase
1	Penilaian Antarmuka Tampilan Awal	Kemudahan	2,5	3,83	95,75%
		Kejelasan menu	3		
		Kecepatan akses menu	4		
2	Penilaian Antarmuka Tampilan Hasil	Kemudahan	4	3,43	85,75%
		Kejelasan menu	3,5		
		Kecepatan pengolahan data	2,8		
3	Penilaian Tampilan Jadwal	Kejelasan tulisan	4	4	100%
		Kecepatan akses menu	4		

BAB VI PENUTUP

Bab ini membahas kesimpulan yang dapat diambil dari tujuan pembuatan perangkat lunak serta hasil uji coba yang telah dilakukan untuk menjawab permasalahan yang telah dirumuskan. Selain itu, terdapat beberapa saran untuk pengembangan lebih lanjut.

6.1. Kesimpulan

Dari hasil pengamatan selama perancangan, implementasi, dan proses uji coba yang dilakukan, tugas akhir ini mengambil kesimpulan sebagai berikut:

1. Aplikasi bisa menampilkan jadwal dari *server*.
2. Aplikasi bisa menyimpam data pengguna.
3. Aplikasi bias menampilkan peta dan rute pemesanan dari alamat asal ke alamat tujuan, dan bias menampilkan jarak dan harga.

6.2. Saran

Di akhir penulisan buku Tugas Akhir ini, terdapat beberapa saran perbaikan terhadap perangkat lunak yang diimplementasikan. Diantaranya adalah sebagai berikut.

1. Layanan aplikasi ini dapat ditingkatkan lagi dengan menyediakan metode pembayaran pada sistem.
2. Aplikasi bisa menampilkan notifikasi untuk menampilkan pesan dari pihak travel.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] Wikipedia. Transportasi. 2016.
[Online]<https://id.wikipedia.org/wiki/Transportasi>
[diakses pada 1 Desember 2016]
- [2] S. Millward, Indonesia diproyeksi lampau 100 jutapengguna smartphone di 2018, keempat di dunia. 2014.
[Online] <https://id.techinasia.com/jumlah-pengguna-smartphone-di-indonesia-2018> [diakses pada 16 Oktober 2016]
- [3] Jenita. 2015. Transportasi Online.
[Online] <https://prezi.com/hqywaio5gn1k/transportasi-online> [diakses pada 1 Desember 2016]
- [4] DiMarzio, J.F. 2008. *Android A Programme's Guide*. USA:The McGraw-Hill
- [5] Welling, Luke dan Laura Thomson. 2009. *PHP and MySQL Web Development*. USA: Pearson Education
- [6] Retrofit: A type safe REST client for Android and Java [documentation online]. SanFrancisco, CA: Square Inc; 2014.
[Online] <http://square.github.io/retrofit> [dikases pada 1 Desember 2016]
- [7] Developer Google. Google Maps Distance Matrix API. 2016 [Online]
<https://developers.google.com/maps/documentation/distance-matrix/intro?hl=id> [diakses pada 6 Desember 2016]]
- [8] Bisnisojek. Apa itu Go-Jek. 2015.
[Online]]<http://bisnisojek.com/2015/09/17/apa-itu-go-jek/>
[diakses pada 1 Desember 2016]

- [9] Wikipediya. Cara Menggunakan Uber. 2016.
[Online]<http://id.wikihow.com/Menggunakan-Uber>
[diakses pada 1 Desember 2016]
- [10] Appnesia. Memudahkan Perjalanan Anda dengan Aplikasi Grab. 2016.
[Online]<http://app-nesia.com/articles/grab-taxi> [diakses
pada 3 Desember 2016]

LAMPIRAN KODE SUMBER

A. Kelas RecycleJadwalAdapter Untuk Mengatur Tampilan Jadwal

```

public class RecyclerJadwalAdapter extends
RecyclerView.Adapter<RecyclerView.ViewHolder> {

    private List<Jadwal> mItems;
    private Context mContext;

    public RecyclerJadwalAdapter(Context context){
        mContext = context;
        mItems = new ArrayList<>();
    }

    @Override
    public RecyclerView.ViewHolder
    onCreateViewHolder(ViewGroup parent, int viewType)
    {
        View v;
        RecyclerView.ViewHolder vh = null;

        v =
        LayoutInflater.from(mContext).inflate(R.layout.ite
        m_jadwal,parent,false);
        vh = new JadwalVH(v);

        return vh;
    }

    @Override
    public void
    onBindViewHolder(RecyclerView.ViewHolder holder,
    int position) {

        JadwalVH vh = (JadwalVH) holder;
        Jadwal item = mItems.get(position);
    }
}

```

```

        vh.tv_hari.setText(item.hari);
        vh.tv_rute.setText(item.kotaAsal + " - " +
item.kotaTujuan);
        vh.tv_jam.setText(item.jam);
        vh.itemView.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                //saat diklik
            }
        });
    }

    @Override
    public int getItemCount() {
        return mItems.size();
    }
    public void addItem (Jadwal item){
        mItems.add(item);
    }

    public void addListItem (List<Jadwal>
itemList){

        mItems.clear();
        mItems.addAll(itemList);
    }
}

```

B. Kelas TravelRestManager Untuk Mengatur REST Data

```

public class TravelRestManager {

    private TravelRestService mRestService;

```

```

public TravelRestService getmRestService() {
    if(mRestService == null){
        Retrofit retrofit = new
Retrofit.Builder()
        .baseUrl(Constants.BASE_URL)
        .addConverterFactory(GsonConverterFactory.create()
)
        .build();

        mRestService =
retrofit.create(TravelRestService.class);
    }

    return mRestService;
}
}

```

C. Kelas DetailActivity untuk Tampilan Jadwal

```

private void setupView() {
    mRvList = (RecyclerView)
findViewById(R.id.rv_list);
    mRvList.setLayoutManager(new
LinearLayoutManager(DetailActivity.this,
LinearLayoutManager.VERTICAL, false));

    mAdapter = new
RecyclerViewJadwalAdapter(DetailActivity.this);
    mRvList.setAdapter(mAdapter);

    mRestManager = new TravelRestManager();

    startFetching();
}
private void startFetching() {
    (item3);
    Call<List<Jadwal>> getJadwal =
mRestManager.getmRestService().getListJadwal();
    getJadwal.enqueue(new Callback<List<Jadwal>>()
{

```

```

        @Override
        public void onResponse(Call<List<Jadwal>>
call, Response<List<Jadwal>> response) {
            if(response.isSuccessful()){
                List<Jadwal>      resultList      =
response.body();
                mAdapter.addItem(resultList);
                mAdapter.notifyDataSetChanged();
            }
        }

        @Override
        public void onFailure(Call<List<Jadwal>>
call, Throwable t) {

        }
    });
}
}

```

D. Kelas ActivityMap untuk Menampilkan Peta dan Menentukan Jarak

```

public class MapsActivity extends FragmentActivity
implements OnMapReadyCallback,
DirectionFinderListener {

    private GoogleMap mMap;
    private Button btnFindPath;
    private EditText etOrigin;
    private EditText etDestination;
    private List<Marker> originMarkers = new
ArrayList<>();
    private List<Marker> destinationMarkers = new
ArrayList<>();
    private List<Polyline> polylinePaths = new
ArrayList<>();
    private ProgressDialog progressDialog;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {

```

```

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        // Obtain the SupportMapFragment and get
        notified when the map is ready to be used.
        SupportMapFragment mapFragment =
        (SupportMapFragment) getSupportFragmentManager()
            .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);

        btnFindPath = (Button)
        findViewById(R.id.btnFindPath);
        etOrigin = (EditText)
        findViewById(R.id.etOrigin);
        etDestination = (EditText)
        findViewById(R.id.etDestination);

        btnFindPath.setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View v) {
                sendRequest();
            }
        });
    }

    private void sendRequest() {
        String origin = etOrigin.getText().toString();
        String destination =
        etDestination.getText().toString();
        if (origin.isEmpty()) {
            Toast.makeText(this, "Please enter origin
            address!", Toast.LENGTH_SHORT).show();
            return;
        }
        if (destination.isEmpty()) {
            Toast.makeText(this, "Please enter
            destination address!", Toast.LENGTH_SHORT).show();
            return;
        }

        try {
            new DirectionFinder(this, origin,
            destination).execute();
        } catch (UnsupportedEncodingException e) {

```

```

        e.printStackTrace();
    }
}

@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    LatLng hcmus = new LatLng(-
    7.2882728, 112.7984043);

    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(
    hcmus, 18));
    originMarkers.add(mMap.addMarker(new
    MarkerOptions()
        .title("Đại học Khoa học tự nhiên")
        .position(hcmus)));

    if (ActivityCompat.checkSelfPermission(this,
    Manifest.permission.ACCESS_FINE_LOCATION) !=
    PackageManager.PERMISSION_GRANTED) &&
    ActivityCompat.checkSelfPermission(this,
    Manifest.permission.ACCESS_COARSE_LOCATION) !=
    PackageManager.PERMISSION_GRANTED) {
        // TODO: Consider calling
        // ActivityCompat#requestPermissions
        // here to request the missing
    permissions, and then overriding
        //
        // public void
    onRequestPermissionsResult(int requestCode,
    String[] permissions,
        //
        int[] grantResults)
        // to handle the case where the user
    grants the permission. See the documentation
        // for ActivityCompat#requestPermissions
    for more details.
        return;
    }
    mMap.setMyLocationEnabled(true);
}

@Override
public void onDirectionFinderStart() {

```

```
        progressDialog = ProgressDialog.show(this,
        "Please wait.",
            "Finding direction..!", true);

        if (originMarkers != null) {
            for (Marker marker : originMarkers) {
                marker.remove();
            }
        }

        if (destinationMarkers != null) {
            for (Marker marker : destinationMarkers) {
                marker.remove();
            }
        }

        if (polylinePaths != null) {
            for (Polyline polyline:polylinePaths ) {
                polyline.remove();
            }
        }
    }

    @Override
    public void onDirectionFinderSuccess(List<Route>
    routes) {
        progressDialog.dismiss();
        polylinePaths = new ArrayList<>();
        originMarkers = new ArrayList<>();
        destinationMarkers = new ArrayList<>();
        int cost = 1000;

        for (Route route : routes) {

            mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(
            route.startLocation, 16));
                ((TextView)
            findViewById(R.id.tvDuration)).setText(route.durat
            ion.text);
                ((TextView)
            findViewById(R.id.tvDistance)).setText(route.dista
            nce.text);
                ((TextView)
            findViewById(R.id.tvCost)).setText("14000");
```

```
        originMarkers.add(mMap.addMarker(new
MarkerOptions()

        .icon(BitmapDescriptorFactory.fromResource(R.drawable.start_blue))
            .title(route.startAddress)
            .position(route.startLocation));
        destinationMarkers.add(mMap.addMarker(new
MarkerOptions()

        .icon(BitmapDescriptorFactory.fromResource(R.drawable.end_green))
            .title(route.endAddress)
            .position(route.endLocation));

        PolylineOptions polylineOptions = new
PolylineOptions().
            geodesic(true).
            color(Color.BLUE).
            width(10);

        for (int i = 0; i < route.points.size();
i++)

        polylineOptions.add(route.points.get(i));

        polylinePaths.add(mMap.addPolyline(polylineOptions
));
    }
}
}
```

LAMPIRAN FORMULIR PENILAIAN PENGUJIAN KEGUNAAN

A. Formulir Uji Coba Subjektif

Tanggal : ____-____-2017
 Nama Penguji : _____
 Profesi Penguji : _____

1. Antarmuka halaman awal aplikasi

No.	Antarmuka Tampilan Awal Aplikasi	Penilaian			
		1	2	3	4
1	Kemudahan				
2	Kejelasan menu				
3	Kecepatan akses menu				

Keterangan:

- 1) Penilaian Kemudahan:
 - 1 – Keseluruhan antarmuka tidak mudah digunakan
 - 2 – Sebagian kecil antarmuka mudah digunakan
 - 3 – Sebagian besar antarmuka mudah digunakan
 - 4 – Keseluruhan antarmuka mudah digunakan
- 2) Penilaian kejelasan menu:
 - 1 – Tidak mengetahui menu dari antarmuka
 - 2 – Mengetahui menu dari antarmuka
 - 3 – Mengetahui menu dari antarmuka tetapi tidak mengetahui tujuannya
 - 4 – Mengetahui menu dan tujuan dari antarmuka
- 3) Penilaian Kecepatan akses menu:
 - 1 – kecepatan akses menu sangat lambat
 - 2 – kecepatan akses menu terasa lambat
 - 3 – Kecepatan akses menu dalam tahap wajar

4 – Kecepatan akses menu cepat

2. Antarmuka Tampilan Hasil

No.	Antarmuka Tampilan Hasil	Penilaian			
		1	2	3	4
1	Kemudahan				
2	Kejelasan menu				
3	Kecepatan pengolahan data				

Keterangan:

1) Penilaian Kemudahan:

- 1 – Keseluruhan antarmuka tidak mudah digunakan
- 2 – Sebagian kecil antarmuka mudah digunakan
- 3 – Sebagian besar antarmuka mudah digunakan
- 4 – Keseluruhan antarmuka mudah digunakan

2) Penilaian kejelasan menu:

- 1 – Tidak mengetahui menu dari antarmuka
- 2 – Mengetahui menu dari antarmuka
- 3 – Mengetahui menu dari antarmuka tetapi tidak mengetahui tujuannya
- 4 – Mengetahui menu dan tujuan dari antarmuka

3) Penilaian Kecepatan pengolahan data:

- 1 – kecepatan pengolahan data sangat lambat
- 2 – kecepatan pengolahan data terasa lambat
- 3 – Kecepatan pengolahan data dalam tahap wajar
- 4 – Kecepatan pengolahan data cepat

3. Tampilan Jadwal

No.	Tampilan Jadwal	Penilaian			
		1	2	3	4
1.	Kejelasan tulisan				
2.	Kecepatan akses menu				

Keterangan:

1) Penilaian Kemudahan:

- 1 – tulisan jadwal tidak dapat dibaca
- 2 – tulisan jadwal sulit untuk dibaca
- 3 – tulisan jadwal dapat dibaca
- 4 – tulisan jadwal mudah dibaca dan bagus

2) Penilaian Kecepatan akses menu:

- 1 – kecepatan akses menu sangat lambat
- 2 – kecepatan akses menu terasa lambat
- 3 – Kecepatan akses menu dalam tahap wajar
- 4 – Kecepatan akses menu cepat

4. Saran Dan Kritik Perbaikan Fitur

No.	Saran dan Kritik Perbaikan Aplikasi
1	
2	

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Penulis, Prasetyati Riski Rahmawati, lahir di Jombang, Jawa Timur, 2 Januari 1991. Penulis adalah putra dari pasangan Purwanto Adiprabowo dan Wahyu Indriati. Penulis menyelesaikan pendidikan pendidikan formal di SDN Jombatan 3 Jombang (1997-2003), SMP Negeri 1 Jombang (2003-2006), dan SMA Negeri 2 (2006-2009). Pada pertengahan 2009, penulis melanjutkan studi di Institut Teknologi Sepuluh Nopember Jurusan S-1 Teknik Informatika, Surabaya, Jawa Timur.

Selama berada dalam bangku perkuliahan, penulis aktif sebagai anggota himpunan HMTC di Jurusan Teknik Informatika. Bidang minat yang diambil oleh penulis pada saat kuliah di Teknik Informatika ITS adalah Manajemen Informasi (MI).